

UNIVERSIDADE FEDERAL FLUMINENSE

Debora Guimarães de Oliveira

**Uma Arquitetura para Roteamento Dinâmico de Vídeos por Multicaminhos em Redes Sem Fio**

NITERÓI

2024

Debora Guimarães de Oliveira

# **Uma Arquitetura para Roteamento Dinâmico de Vídeos por Multicaminhos em Redes Sem Fio**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Orientador:

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

Coorientador:

Prof. Fabiano Pereira Bhering, D.Sc.

NITERÓI

2024

Ficha catalográfica automática - SDC/BEE  
Gerada com informações fornecidas pelo autor

048a Oliveira, Debora Guimarães de  
Uma Arquitetura para Roteamento Dinâmico de Vídeos por  
Multicaminhos em Redes sem Fio / Debora Guimarães de  
Oliveira. - 2024.  
74 f. : il.

Orientador: Célio Vinicius Neves de Albuquerque.  
Coorientador: Fabiano Pereira Bhering.  
Dissertação (mestrado)-Universidade Federal Fluminense,  
Instituto de Computação, Niterói, 2024.

1. Roteamento por Multicaminhos. 2. Redes sem Fio. 3.  
Monitor de Topologia. 4. Qualidade de Experiência do  
Usuário. 5. Produção intelectual. I. Albuquerque, Célio  
Vinicius Neves de, orientador. II. Bhering, Fabiano Pereira,  
coorientador. III. Universidade Federal Fluminense. Instituto  
de Computação. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

Debora Guimarães de Oliveira

Uma Arquitetura para Roteamento Dinâmico de Vídeos por Multicaminhos  
em Redes Sem Fio

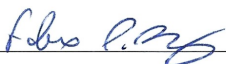
Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação.

Aprovada em março de 2024.

BANCA EXAMINADORA



Prof. Célio Vinicius Neves de Albuquerque, Ph.D. - Orientador, UFF



Prof. Fabiano Pereira Bhering, D.Sc. - Coorientador, CEFET-MG



Prof. Igor Monteiro Moraes, D.Sc. - UFF



Prof. Luís Henrique Maciel Kosmalski Costa, Dr. - COPPE/UFRJ

Niterói

2024

*Aos meus familiares e amigos.*

# Agradecimentos

Agradeço aos meus familiares pelo constante incentivo e apoio ao longo da vida.

Ao Prof. Célio Albuquerque, pela cordialidade e orientações valiosas desde o início do mestrado, sempre empenhado em oferecer as melhores condições para o meu progresso acadêmico e o desenvolvimento desta pesquisa.

Ao Prof. Fabiano Bhering, pela disponibilidade, atenção e pelos significativos ensinamentos compartilhados.

Ao Prof. Diego Passos e à Prof.<sup>a</sup> Katia Obraczka, pelas contribuições essenciais para o sucesso deste trabalho.

Aos membros da banca examinadora, Prof. Igor Moraes e Prof. Luís Kosmalski, pela gentileza em se disporem a colaborar com esta dissertação.

Ao Instituto de Computação (IC) da UFF e, em particular, ao Laboratório Mídia-Com, pela infraestrutura disponibilizada ao longo de todo o processo de elaboração deste trabalho.

À Marinha do Brasil, e especialmente à Diretoria de Sistemas de Armas da Marinha, pela oportunidade de me proporcionar uma experiência enriquecedora que contribuiu significativamente para o meu crescimento profissional e acadêmico.

Por fim, a todos que de alguma forma contribuíram, meu sincero agradecimento por fazerem parte desta conquista.

# Resumo

As aplicações de vídeo na Internet das Coisas através de redes sem fio multissalto se beneficiam de estratégias de roteamento por multicaminhos para atender aos requisitos de Qualidade de Experiência. Os experimentos conduzidos até o momento avaliaram a qualidade dos vídeos apenas com os fluxos de vídeo transmitidos, desconsiderando a presença dos fluxos de controle da topologia, necessários para o funcionamento dos mecanismos. No entanto, em cenários realistas, os fluxos de controle da topologia devem ser transmitidos simultaneamente com os fluxos de vídeo, a fim de manter a topologia constantemente atualizada, e permitir que possíveis alterações de rotas sejam processadas pelo mecanismo de seleção de multicaminhos diante de mudanças significativas na topologia ao longo do tempo, como por exemplo, eventuais degradações tempestivas de qualidade e/ou quebra dos enlaces.

Este trabalho propõe uma arquitetura de rede sem fio descentralizada, onde cada nó mantém uma visão completa da topologia da rede, semelhante a implementação de protocolos de estado dos enlaces. O objetivo é possibilitar um roteamento dinâmico que se adapte às mudanças da topologia para proporcionar uma melhor Qualidade de Experiência ao usuário. Os resultados das simulações indicam que a arquitetura proposta consegue recuperar a qualidade do vídeo em caso de degradação dos enlaces. No entanto, observou-se que a presença dos fluxos de controle, essenciais para a manutenção da topologia, pode impactar na qualidade final dos vídeos.

Esta pesquisa também contribuiu com a análise de duas abordagens de provisionamento das taxas dos fluxos de controle da topologia. Essas estratégias de provisionamento mostraram uma tendência de melhoria na Qualidade de Experiência do usuário em comparação aos cenários que não empregam nenhuma técnica de provisão. Por fim, a análise dos resultados evidenciou a baixa correlação entre a métrica de roteamento baseada na qualidade dos enlaces e a métrica de Qualidade de Experiência do usuário nas transmissões de múltiplos fluxos por multicaminhos.

**Palavras-chave:** Roteamento por Multicaminhos, Redes sem Fio, Monitor de Topologia, Aplicações de Vídeo, Métricas de Roteamento, Qualidade de Experiência do Usuário.

# Abstract

Internet of Things video applications in multihop wireless networks benefit from multipath routing strategies to meet Quality of Experience requirements. The experiments conducted so far have assessed video quality only with video streams, disregarding the presence of topology control streams. However, in realistic scenarios, the topology control flows shall be transmitted simultaneously with the video flows in order to keep the topology constantly updated, and allow possible route changes to be automatically established by the multipath selection mechanism in the face of significant changes in the topology over time, such as occasional degradation in quality and/or breakage of links.

This work proposes a decentralized wireless network architecture where each node maintains a complete view of the network topology, similar to the implementation of link state protocols. The objective is to enable dynamic routing that adapts to changes in the topology to provide a better Quality of Experience for the user. The simulation results indicate that the proposed architecture can recover video quality in case of link degradation. However, it was observed that the presence of control flows, essential for maintaining the topology, can reduce the final quality of the videos.

This research also contributed to the analysis of two approaches to provisioning topology control flow rates. These provisioning strategies showed an improvement trend in the user's Quality of Experience compared to scenarios that did not employ any provisioning technique. Finally, the analysis of the results highlighted the lack of correlation between the routing metric based on link quality and the User Quality of Experience metric in the transmission of multiple flows over multipath.

**Keywords:** Multipath Routing, Wireless Networks, Topology Monitor, Video Applications, Routing Metrics, Quality of Experience.



# Lista de Figuras

2.1	Cenário de um sistema de videovigilância com múltiplas câmeras transmitindo por multicaminhos. Adaptado de [9]. . . . .	21
3.1	Diagrama geral do FITPATH [9]. . . . .	28
4.1	Arquitetura proposta com um plano de controle descentralizado. . . . .	31
4.2	Fluxograma simplificado de processos para a transmissão de vídeos com roteamento dinâmico por multicaminhos. . . . .	33
4.3	Formato do pacote HELLO. . . . .	43
4.4	Formato do pacote TC. . . . .	44
4.5	Formato do pacote de requisição. . . . .	45
4.6	Formato do pacote dos fluxos de vídeo. . . . .	46
5.1	Impacto dos fluxos de controle da topologia na QoE do usuário em transmissões de cenas de vídeo de baixa mobilidade, com codificação LC. . . . .	53
5.2	Impacto dos fluxos de controle da topologia na QoE do usuário em transmissões de cenas de vídeo de alta mobilidade, com codificação MDC. . . . .	54
5.3	QoE do usuário em diversos níveis de carga ofertada e diferentes estratégias de provisão das taxas dos fluxos de controle da topologia ao FITPATH. . . . .	56
5.4	Configuração da topologia de 60 nós com transmissão de vídeo por duas fontes para um nó destinatário (estação de monitoramento). . . . .	57
5.5	Soluções de caminhos geradas pelo FITPATH na topologia de 60 nós. . . . .	58
5.6	Desempenho da QoE média dos últimos 30 segundos. . . . .	59
5.7	Soluções de caminhos geradas pelo FITPATH após falha do nó 11. . . . .	59
5.8	Evolução das métricas de roteamento registrada no nó destinatário a cada 30 segundos quando há tráfego dos vídeos. . . . .	60

---

5.9	Evolução das métricas de roteamento registrada no nó destinatário a cada 30 segundos apenas com o tráfego de controle da topologia. . . . .	61
-----	---	----

# Lista de Tabelas

1	Parâmetros do protocolo do Monitor de topologia. . . . .	42
2	Parâmetros das Simulações. . . . .	50
3	SSIM nas transmissões de cenas de vídeo de baixa mobilidade por caminhos selecionados pelo FITPATH. . . . .	72
4	SSIM nas transmissões de cenas de vídeo de baixa mobilidade por caminhos selecionados pelo QSOpt. . . . .	73
5	SSIM nas transmissões de cenas de vídeo de alta mobilidade por caminhos selecionados pelo FITPATH. . . . .	73
6	SSIM nas transmissões de cenas de vídeo de alta mobilidade por caminhos selecionados pelo QSOpt. . . . .	73
7	SSIM em estratégias distintas de provisão das taxas dos fluxos de controle da topologia. . . . .	74

# Lista de Abreviaturas e Siglas

**ACK** do inglês, *Acknowledgement*

**ANSI** do inglês, *American National Standards Institute*

**AVC** do inglês, *Advanced Video Coding*

**CLMR** do inglês, *Cross-Layer Multipath Routing*

**ERVT** do inglês, *Error Resilient Video Transmission*

**ETX** do inglês, *Expected Transmission Count*

**FMO** do inglês, *Flexible Macroblock Ordering*

**GOP** do inglês, *Group Of Pictures*

**GPLv2** do inglês, *General Public License, version 2*

**HD** do inglês, *High Definition*

**HEVC** do inglês, *High Efficiency Video Coding*

**IEEE** do inglês, *Institute of Electrical and Electronic Engineers*

**ILS** do inglês, *Iterator Local Search*

**IoT** do inglês, *Internet of Things*

**IP** do inglês, *Internet Protocol*

**LC** do inglês, *Layered Coding*

**LQ** do inglês, *Link Quality*

**MAPE** do inglês, *Multimedia-Aware Performace Estimator*

**MDC** do inglês, *Multiple Description Coding*

**MPEG** do inglês, *Motion Picture Expert Group*

**NLQ** do inglês, *Neighbor Link Quality*

**NS-3** do inglês, *Network Simulator 3*

**Q-MMTP** do inglês, *QoS-oriented Multipath Multimedia Transmission Planning*

**QoE** do inglês, *Quality of Experience*

**QoS** do inglês, *Quality of Service*

**QSOpt** do inglês, *QoE-aware Sub-Optimal routing*

**RAM** do inglês, *Random Access Memory*

**RTVP** do inglês, *Real-Time Video streaming routing Protocol*

**SHVC** do inglês, *Scalable High-definition Video Coding*

**SL** do inglês, *Single Layer*

**SNR** do inglês, *Signal-to-Noise Ratio*

**SSIM** do inglês, *Structural Similarity Index Measure*

**SVC** do inglês, *Scalable Video Coding*

**TC** do inglês, *Topology Control*

**UDP** do inglês, *User Datagram Protocol*

**VoD** do inglês, *Video on-Demand*

**VVC** do inglês, *Versatile Video Coding*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Objetivos . . . . .	15
1.2	Questões da Pesquisa . . . . .	16
1.3	Organização . . . . .	16
<b>2</b>	<b>Aplicações de Vídeo em IoT</b>	<b>18</b>
2.1	Saúde . . . . .	18
2.2	Indústria . . . . .	19
2.3	Meio Ambiente . . . . .	19
2.4	Cidades Inteligentes . . . . .	20
2.4.1	Videovigilância com Transmissões por Multicaminhos . . . . .	21
<b>3</b>	<b>Fundamentos Teóricos</b>	<b>23</b>
3.1	<i>Streaming</i> de Vídeo . . . . .	23
3.1.1	<i>Streaming</i> em Tempo Real . . . . .	23
3.1.2	<i>Streaming</i> Sob Demanda . . . . .	24
3.2	Compressão de Vídeo . . . . .	24
3.2.1	Codificação por Camadas . . . . .	25
3.2.1.1	Camada Única . . . . .	26
3.2.1.2	Camadas Escaláveis . . . . .	26
3.2.2	Codificação por Múltiplos Descritores . . . . .	27
3.3	Mecanismos de Seleção de Multicaminhos . . . . .	27

---

3.3.1	FITPATH . . . . .	28
3.3.2	<i>QoE-aware Sub-Optimal routing</i> (QSOpt) . . . . .	29
<b>4</b>	<b>Arquitetura Proposta</b>	<b>30</b>
4.1	Visão Geral da Arquitetura . . . . .	30
4.2	Monitor de Topologia . . . . .	32
4.3	Roteamento . . . . .	39
4.4	Implementação . . . . .	40
4.4.1	Formato dos Pacotes . . . . .	43
4.4.1.1	Pacote HELLO . . . . .	43
4.4.1.2	Pacote de Controle de Topologia (TC) . . . . .	44
4.4.1.3	Pacote de Requisição . . . . .	45
4.4.1.4	Pacotes de Dados . . . . .	46
<b>5</b>	<b>Avaliações Experimentais</b>	<b>48</b>
5.1	Metodologia de Avaliação . . . . .	49
5.1.1	Modelo de Tráfego . . . . .	50
5.1.2	Métrica de Avaliação da QoE do Usuário . . . . .	51
5.2	Resultados . . . . .	51
5.2.1	Impacto dos Fluxos de Controle na QoE do Usuário . . . . .	51
5.2.1.1	Cenas de Vídeo de Baixa Mobilidade . . . . .	52
5.2.1.2	Cenas de Vídeo de Alta Mobilidade . . . . .	53
5.2.2	Desempenho da QoE com Provisão das Taxas de Controle . . . . .	54
5.2.3	Desempenho da QoE . . . . .	57
5.2.4	Evolução do ETX no Roteamento Dinâmico . . . . .	60
<b>6</b>	<b>Conclusão</b>	<b>63</b>
6.1	Trabalhos Futuros . . . . .	64

---

<b>REFERÊNCIAS</b>	<b>65</b>
<b>Apêndice A</b>	<b>72</b>
A.1 SSIM de Cenas de Vídeo de Baixa Mobilidade . . . . .	72
A.2 SSIM de Cenas de Vídeo de Alta Mobilidade . . . . .	73
<b>Apêndice B</b>	<b>74</b>
B.1 SSIM em Estratégias de Provisão das Taxas de Controle . . . . .	74



# 1 Introdução

Nos últimos anos, os avanços das tecnologias de câmeras IP (do inglês, *Internet Protocol*) e das redes de comunicação sem fio viabilizaram uma ampla variedade de aplicações de vídeo para Internet das Coisas (ou IoT, do inglês *Internet of Things*) em áreas como cidades inteligentes, indústria, meio ambiente e saúde [11, 50, 51, 26]. Nas cidades inteligentes, mais especificamente no campo da segurança pública, diversas aplicações de vídeo IoT são empregadas na vigilância de vias públicas, estacionamentos e de transportes públicos [50, 56, 59], captando e transmitindo os vídeos por redes sem fio multissaltos.

Face à natureza de alta demanda de tráfego nas transmissões de vídeo, as aplicações IoT podem experimentar congestionamentos e atrasos na rede, em virtude de restrições de banda e perda de pacotes, resultando em impactos adversos na qualidade da mídia entregue [37]. Neste contexto, os desafios impostos são significativos, devido aos requisitos rigorosos de garantia da Qualidade de Serviço (ou QoS, do inglês *Quality of Service*) para a rede subjacente em termos de vazão, atraso e perda de pacotes, principalmente no que tange às aplicações de tempo real [33].

O roteamento por multicaminhos, multissalto sem fio, é uma técnica que utiliza múltiplos caminhos entre uma fonte e o seu destinatário. O uso desse tipo de roteamento para a distribuição do tráfego de vídeo tem se apresentado como uma boa estratégia para suprir os requisitos de QoS por oferecer reduções de atrasos e de perdas contínuas de pacotes, devido ao balanceamento de cargas [3].

Atualmente, há diversos mecanismos de seleção de multicaminhos para a transmissão de vídeos em redes sem fio multissaltos [9, 29, 1, 4, 35, 54, 12]. No entanto, para garantir a eficiência na alocação das rotas, esses mecanismos adotam critérios distintos, baseados na topologia corrente da rede.

Neste trabalho, serão considerados dois mecanismos de seleção de multicaminhos, o FITPATH [9] e o QSOpt (do inglês, *QoE-aware Sub-Optimal routing*) [54], pois estes se destacaram no trabalho de [9], respectivamente, com a melhor e a segunda melhor QoE

(do inglês, *Quality of Experience*) do usuário alcançada, mensurada em SSIM (do inglês *Structural Similarity Index Measure*). Embora o FITPATH e o QSOpt sejam baseados na visão corrente da topologia da rede, é importante destacar que nos experimentos dos estudos publicados em [9] e [54], após a geração da solução de caminhos pelo mecanismo de seleção, apenas os fluxos de vídeo foram transmitidos, sem a presença dos fluxos de controle da topologia. Contudo, em um cenário realista, os fluxos de controle da topologia devem ser transmitidos juntamente com os fluxos de vídeo, de modo a manter a topologia constantemente atualizada, e permitir que possíveis alterações de rotas sejam automaticamente estabelecidas pelo mecanismo de seleção de multicaminhos, caso ocorram mudanças significativas na topologia ao longo do tempo, como por exemplo, eventuais degradações significantes de qualidade e/ou quebra dos enlaces, ou falha nos nós intermediários.

## 1.1 Objetivos

Motivado por oferecer suporte dinâmico ao roteamento por multicaminhos, este trabalho propõe um monitor de topologia em uma arquitetura de rede sem fio, descentralizada, do tipo *ad hoc*. Neste cenário, cada nó mantém a visão completa da topologia da rede, através do monitoramento e a disseminação dos estados dos enlaces, de modo similar ao implementado pelos protocolos de roteamento proativos. O objetivo é permitir que o mecanismo de seleção de multicaminhos estabeleça, de modo automático, novas rotas para os fluxos de vídeo, caso ocorra uma mudança significativa na qualidade de quaisquer enlaces contidos nas rotas preestabelecidas. A finalidade da arquitetura proposta é aumentar a resiliência da QoE do usuário em situações de mudanças tempestivas da topologia.

Ressalta-se ainda que para manter a topologia constantemente atualizada, é necessário o envio periódico de pacotes de controle da topologia. Tais pacotes concorrem com os fluxos de vídeo no canal e podem causar efeitos adversos na QoE do usuário. Visto que o tráfego de controle da topologia não foi considerado nos estudos de [9] e [54], este trabalho também tem por objetivo avaliar, sob uma condição de rede mais realista, o impacto dos fluxos de controle da topologia na QoE do usuário, bem como propor soluções de provisionamento que visem mitigar este impacto.

## 1.2 Questões da Pesquisa

Com o propósito de avaliar a qualidade dos vídeos transmitidos por multicaminhos em IoT, em um cenário mais próximo do realista, esta pesquisa é orientada pelas seguintes questões:

- **Q1:** Como a presença dos fluxos de controle da topologia impacta na QoE do usuário?
- **Q2:** Uma vez que as soluções de caminhos dos mecanismos de seleção são baseadas nas taxas dos fluxos de vídeo, provisionar as taxas dos fluxos de controle da topologia poderia mitigar o impacto na QoE do usuário, uma vez que seriam selecionados, na medida do possível, caminhos compatíveis a todos os fluxos da rede?
- **Q3:** A arquitetura proposta, com o monitor de topologia, é capaz de restaurar a QoE do usuário em caso de interrupção de um dos fluxos de vídeo, por motivo de falha em um dos nós intermediários? No contexto das soluções de caminhos otimizadas, a arquitetura também é capaz de melhorar a QoE do usuário?
- **Q4:** Ao longo da transmissão dos fluxos de vídeo, há alguma correlação entre as métricas de roteamento (baseadas na qualidade dos enlaces) obtidas durante a transmissão dos fluxos de vídeo e a QoE do usuário?

## 1.3 Organização

O texto deste trabalho está organizado da seguinte forma:

- O Capítulo 2 apresenta alguns dos principais domínios das aplicações de vídeo em IoT e descreve o cenário de um sistema de videovigilância com transmissões por multicaminhos;
- O Capítulo 3 apresenta uma visão geral sobre o *streaming* de vídeo, aborda seus desafios nas transmissões em tempo real e descreve os fundamentos teóricos sobre as técnicas de compressão de vídeos. Adicionalmente é descrita uma revisão sobre os dois mecanismos de seleção de multicaminhos, FITPATH e QSOpt, utilizados na avaliação da arquitetura proposta neste trabalho;
- O Capítulo 4 apresenta a visão geral de uma arquitetura proposta para sistemas de videovigilância em IoT, e introduz um monitor de topologia para esta arquitetura.

- 
- O Capítulo 5 apresenta a metodologia e os resultados obtidos por meio de experimentos conduzidos com o objetivo de responder as questões da Seção 1.2; e
  - Por fim, o Capítulo 6 contém as considerações finais, resumindo as contribuições, bem como sugestões para trabalhos futuros.

## 2 Aplicações de Vídeo em IoT

Atualmente, os sistemas de vídeo em IoT estão cada vez mais integrados com tecnologias avançadas como aprendizagem profunda, *blockchain*, computação de borda e computação em nuvem [50], o que vem gerando melhorias significativas em vários domínios, como por exemplo, saúde, indústria, meio ambiente e cidades inteligentes. Neste capítulo, as principais características das aplicações de vídeo em IoT nesses domínios são apresentadas.

### 2.1 Saúde

Os sistemas de saúde têm cada vez mais adotado aplicações de vídeo para o diagnóstico e monitoramento de pacientes em tempo real, tanto em instalações de saúde, quanto em *homecare* [49, 34].

A telemedicina é outra aplicação importante que possibilita o atendimento e a avaliação remota de pacientes por médicos especialistas, mesmo que estes não estejam disponíveis na localidade do paciente. Durante os exames remotos, dados provenientes de sensores, tais como: frequência cardíaca, temperatura corporal, pressão arterial e atividade respiratória, podem ser avaliados pelo especialista [49, 48]. As aplicações de telemedicina geralmente demandam transmissões de vídeo em tempo real, caracterizadas por baixa latência e alta largura de banda. Para apoiar serviços de saúde conectados, os nós em IoT são implantados em ambientes internos (por exemplo, em hospitais, clínicas e residências de pacientes) [11].

O reconhecimento de comportamento também é um elemento técnico chave em diversas aplicações clínicas. Em terapia intensiva, resgate de emergência, reabilitação e outros cenários, é possível monitorar o comportamento dos pacientes por meio de gravação remota de vídeo para identificar efetivamente os movimentos dos pacientes e fornecer uma base para a avaliação e o tratamento. Atualmente, os principais métodos de reconhecimento de comportamento utilizam redes neurais de convolução ou redes neurais recorrentes [48].

Um grande desafio nas aplicações de vídeo em IoT para cuidados da saúde está na aceitação, pelos pacientes, da monitorização constante do comportamento diário, com medidas equilibradas de privacidade e segurança [18].

## 2.2 Indústria

Em aplicações industriais, o processamento de vídeo IoT em tempo real pode ser utilizado para a detecção e classificação de defeitos de produtos, correção de movimento do trabalhador ou verificação de erros na montagem de componentes de equipamentos [53].

Devido à crescente acessibilidade dos sensores de visão artificial e aos componentes associados de *software* e *hardware* significativamente mais capazes, espera-se que as aplicações de vídeo em IoT desempenhem um papel cada vez mais importante na segurança e monitoramento inteligente dos processos [18].

As câmeras em IoT geralmente são instaladas em diferentes pontos das plantas industriais e podem operar sob condições desafiadoras. Portanto, elas podem ter requisitos rigorosos de qualidade de serviço, como alta largura de banda para fornecer alta resolução de vídeo, alta confiabilidade e baixo atraso na transmissão de dados [18, 69, 11].

## 2.3 Meio Ambiente

O monitoramento ambiental é outro domínio importante de aplicação de IoT que utiliza o vídeo, bem como outros tipos de dados de sensores, para monitorar habitats de vida selvagem, florestas e oceanos. Tais implantações geralmente visam locais remotos e de difícil acesso e, portanto, devem ser energeticamente eficientes para maximizar sua vida útil operacional, independentemente da intervenção humana (por exemplo, substituição de bateria) [11, 51].

As câmeras de alta resolução equipadas com recursos de infravermelhos e de visão noturna são usadas para o monitoramento visual e a detecção de chamas e fumaça, mesmo em condições de baixa visibilidade. Os algoritmos de aprendizagem profunda também podem integrar dados de vários sensores IoT, de modo a criar uma compreensão abrangente do ambiente monitorado [2].

Em aplicações para gerenciamento de desastres, como incêndios florestais, inundações ou deslizamentos de terra, além de confiabilidade e eficiência energética, baixa latência e

alto rendimento também são necessários para detecção oportuna de eventos e rastreamento de eventos em tempo real [11, 40].

## 2.4 Cidades Inteligentes

Uma das aplicações de vídeo mais comuns em cidades inteligentes é na área de segurança e vigilância, sendo utilizada para o controle de acesso e para a identificação de atividades tais como roubo, vandalismo e movimentos suspeitos em multidões, entre outros fins [50, 23].

Até alguns poucos anos atrás, os sistemas de vídeo tradicionais dependiam apenas da supervisão humana para monitorar e analisar os vídeo em tempo real. No entanto, os sistemas de vídeo recentes evoluíram com tecnologias inteligentes que gravam e salvam automaticamente dados de vídeo em *blockchain* e, em seguida, analisam e interpretam os dados de vídeo usando técnicas de visão computacional baseadas em aprendizagem profunda que podem ser realizadas pela borda ou em nuvem [55].

Os sistemas de videovigilância inteligentes visam detectar e rastrear objetos, reconhecer rostos, identificar anomalias e prever potenciais incidentes ou emergências. Portanto, as aplicações demandam funções como detecção, classificação e rastreamento de objetos, reconhecimento de ação humana, detecção de anomalias e gerenciamento de armazenamento de vídeo [50].

Os grandes centros urbanos também têm adotado redes de câmeras para ajudar a gerenciar o transporte público, o tráfego, as condições das estradas e os estacionamentos [5]. O vídeo obtido por múltiplas câmeras pode ser analisado em tempo real ou *offline* usando técnicas de visão computacional para fornecer informações valiosas sobre horários de pico, tráfego e condições/incidentes das estradas e rotas, dentre outros [40].

Nota-se que os requisitos de rede variam de acordo com os requisitos específicos da aplicação. Por exemplo, a gestão do tráfego só necessita de confiabilidade suficiente para obter detalhes que permitam distinguir os contornos dos veículos dos arredores e monitorizar o seu movimento. Por outro lado, aplicações de rastreamento de pessoas requerem alto rendimento de rede, pois dependem de vídeo de alta definição para permitir a identificação e rastreamento de características faciais [23].

Os nós da rede geralmente são implantados em postes nas ruas [5]. Como em todos os ambientes urbanos, a comunicação sem fio em cidades inteligentes pode sofrer com

grandes obstáculos (como edifícios), ruído e interferência. Assim, o desempenho dos sistemas também é altamente dependente da distribuição dos nós das câmeras, bem como dos demais fatores citados [11].

A Seção 2.4.1 a seguir apresenta um cenário de um sistema de videovigilância que utiliza a estratégia de transmissão dos fluxos de vídeo por multicaminhos.

### 2.4.1 Videovigilância com Transmissões por Multicaminhos

O cenário de aplicação considerado neste trabalho é de um sistema de videovigilância em área urbana, conforme ilustrado na Figura 2.1, no qual foi assumido que os nós da rede são fontes de vídeo (câmeras) pertencentes a um conjunto  $S$ , bem como também podem fazer o papel de nós intermediários.

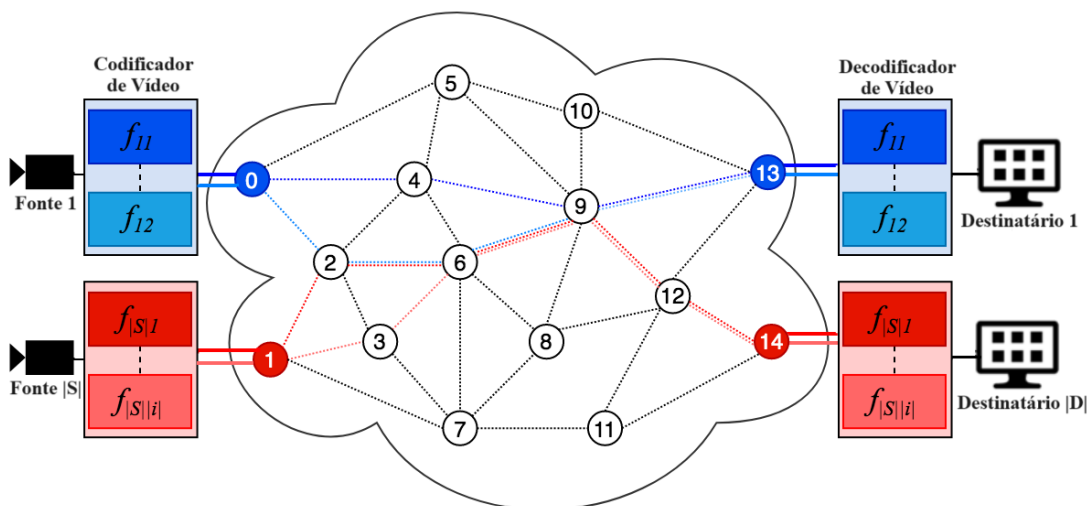


Figura 2.1: Cenário de um sistema de videovigilância com múltiplas câmeras transmitindo por multicaminhos. Adaptado de [9].

Cada fonte  $s \in S$  pode gerar um conjunto de fluxos de vídeo  $f_s = \{f_{s_i}\}$ , onde  $0 < i \leq |f_s|$ . Cada fluxo  $f_{s_i}$  transmitido possui uma taxa de bits específica, determinada pela fonte, de acordo com a resolução do vídeo e o codificador empregado, e utiliza um caminho selecionado  $p_{s_i}$  de uma lista de  $P$  caminhos possíveis para alcançar o destinatário.

Em cada nó destinatário, o decodificador de vídeo é responsável por sincronizar e combinar os fluxos recebidos para a renderização do vídeo. Neste cenário, os nós destinatários são estações de vigilância, onde os vídeos são observados por operadores em tempo real e, de acordo com as demandas estabelecidas na aplicação, múltiplas fontes poderão transferir fluxos simultaneamente para um ou mais destinatários.

Em contraste com um codificador de mídia convencional que gera um único fluxo de



---

bits, este trabalho assume o uso de vídeo escalável. A codificação em camadas (ou LC, do inglês *Layered Coding*) e a codificação em múltiplos descritores (ou MDC, do inglês *Multiple Description Coding*) são técnicas de codificação escaláveis tradicionalmente utilizadas em transmissões de vídeo por multicaminhos [45], pois codificam uma fonte em dois ou mais subfluxos com diferentes taxas de bits que podem ser transmitidos pelos caminhos selecionados. Assim, com o objetivo de maximizar a QoE do usuário, os mecanismos de seleção de multicaminhos devem considerar as taxas de bits de cada fluxo concorrente, a fim de balancear a carga oferecida e minimizar os atrasos e as perdas de pacotes. Essas técnicas de codificação são detalhadas na Seção 3.2.

# 3 Fundamentos Teóricos

Este capítulo apresenta uma visão geral sobre o *streaming* de vídeo, aborda seus desafios nas transmissões em tempo real e descreve os fundamentos teóricos sobre as técnicas de compressão de vídeo. Adicionalmente é descrita uma revisão sobre os dois mecanismos de seleção de multicaminhos, FITPATH e QSOpt, utilizados na arquitetura proposta neste trabalho.

## 3.1 *Streaming* de Vídeo

O *streaming* de vídeo pode ocorrer em serviços sob demanda ou em tempo real.

### 3.1.1 *Streaming* em Tempo Real

No contexto das aplicações IoT, o vídeo ao vivo é um serviço de *streaming* onde as câmeras capturam e transmitem as imagens de vídeo, em tempo real, aos destinatários. Nestas aplicações, o destinatário geralmente é uma central de monitoramento, onde os operadores fazem a vigilância por meio das imagens. O *streaming* de vídeo é particularmente desafiador, devido aos rigorosos requisitos de vazão e atraso. As retransmissões de pacotes ou o armazenamento em *buffer* de grande capacidade, nestes casos, não são opções adequadas [11]. Uma aplicação de vídeo em tempo real impõe um prazo de decodificação. Desta forma, os pacotes atrasados não podem ser tratados no decodificador, mesmo que cheguem com sucesso após o prazo [67]. Por isso, várias abordagens foram propostas para a melhoria da QoS do *streaming* de vídeo, dentre estas, a transmissão de vídeo por multicaminhos.

Cabe ressaltar que suprir os requisitos de QoS não garante necessariamente uma maior QoE do usuário. Sistema operacional dos dispositivos, *hardware*, bateria e pessoas ao redor do usuário são alguns exemplos de fatores que também impactam a experiência dos usuários [62, 7].

### 3.1.2 *Streaming* Sob Demanda

O serviço de vídeo sob demanda (ou VoD, do inglês *Video on Demand*) permite ao usuário assistir a vídeos pré-gravados no momento desejado. Esse tipo de serviço é utilizado em uma ampla variedade de domínios IoT, como por exemplo, em aplicações de análise de vídeos para segurança pública, e no planejamento de tráfego em cidades inteligentes [11]. Nestas aplicações, os vídeos são capturados, codificados e pré-armazenados em servidores, para serem assistidos futuramente, conforme a necessidade do usuário [3].

O servidor onde o vídeo é armazenado também pode realizar alguns processamentos de filtragens e extração de informações a respeito das cenas, como por exemplo, a análise de atividades. Além disso, resumos de vídeo podem ser gerados para ajudar os usuários a navegar por grandes sequências de vídeo e recuperar aquelas que são mais relevantes para a análise. Para tanto, diferentes técnicas de sumarização de vídeos utilizando métodos baseados em clusterização e aprendizagem profunda são apresentados em [41, 42, 43]. Ao contrário do *streaming* em tempo real, os serviços de Vídeo sob Demanda (VoD) não estão sujeitos às mesmas restrições de atraso de fim a fim. Os usuários têm a liberdade de avançar, retroceder e reproduzir o vídeo conforme necessário, o que torna a utilização de um *buffer* com maior capacidade de armazenamento mais facilmente tolerável.

## 3.2 Compressão de Vídeo

As técnicas de compressão de vídeo segmentam cada quadro (também conhecido como *frame*) em unidades chamadas de macroblocos, que são então compactadas para reduzir a largura de banda necessária para a transmissão do vídeo, ao mesmo tempo que mantém uma qualidade visual aceitável. A compressão explora as correlações espaciais e temporais dos macroblocos, conforme especificado pelos padrões de codificação de vídeo.

A compressão de vídeo *Motion Picture Expert Group* (MPEG) [44], por exemplo, segmenta o vídeo em *Group Of Pictures* (GOP), que determina a organização dos quadros entre sucessivos *frames* I. Cada GOP pode ter até três tipos de *frames*: (i) *frames* I, também conhecidos como *intra coded picture*, (ii) *frames* P, também conhecidos como *predictive coded picture*, e (iii) *frames* B, também conhecidos como *bipredictive coded picture*. Os *frames* I são compactados de forma independente, e não necessitam de informações adicionais para a sua reconstrução. Eles são usados como referência para a descompactação dos *frames* P e B.

A compressão de vídeo MPEG padroniza estruturas e taxas de bits do decodificador para permitir o desenvolvimento de algoritmos de codificação eficientes. Ele oferece suporte a uma ampla gama de parâmetros específicos de aplicações, e fornece um *framework* para estender a codificação em camadas e a codificação por múltiplos descritores como uma forma de oferecer suporte a aplicações de vídeo emergentes.

O padrão H.264/AVC (do inglês, *Advanced Video Coding*) tem sido amplamente utilizado por aplicações de vídeo HD (do inglês, *High Definition*). Em seguida, surgiu o padrão de codificação de vídeo de alta eficiência H.265/HEVC (do inglês, *High Efficiency Video Coding*), cuja eficiência maior de codificação permitiu o *streaming* de vídeo 4K com maior fidelidade. Mais recentemente, o padrão H.266/VVC (do inglês, *Versatile Video Coding*) [14] foi concebido não só para proporcionar uma redução substancial da taxa de bits em comparação com o seu antecessor HEVC, mas também para ser altamente versátil, ou seja, para cobrir todas as necessidades de mídia atuais e emergentes, incluindo resolução ainda mais alta (até 8K ou maior).

A fim de mitigar deficiências induzidas pelo canal, a codificação em camadas (ou LC, do inglês *Layered Coding*) e a codificação por múltiplos descritores (ou MDC, do inglês *Multiple Description Coding*) foram propostas como técnicas robustas de codificação de fonte contra erros inevitáveis de transmissão [30, 32]. Em contraste com um codificador de mídia convencional que gera um único fluxo de bits, os codificadores LC e MDC codificam uma fonte de mídia em dois ou mais subfluxos que podem ser transmitidos por multicaminhos.

Na codificação LC, os subfluxos são mapeados em camadas hierárquicas, onde o fluxo da camada base é o mais importante e fornece um nível de qualidade básico, que pode ser melhorado com camadas de aprimoramento adicionais. Na codificação MDC, todos os subfluxos têm a mesma importância, pois diferentes níveis de qualidade podem ser obtidos com diferentes combinações de subfluxos [45].

### 3.2.1 Codificação por Camadas

A codificação LC tornou-se parte dos padrões estabelecidos de compressão de vídeo [36]. Ela fornece fluxos de bits incorporados em camadas, que são gerados em diferentes taxas de bits, codificando uma sequência de vídeo em múltiplas camadas sem comprometer a qualidade do vídeo. As representações em camadas fornecem uma maneira conveniente de realizar o controle de taxa para mitigar o congestionamento da rede, e podem usar camada única ou camadas escaláveis [17].

### 3.2.1.1 Camada Única

As técnicas de camada única (ou SL, do inglês *Single Layer*) realizam a compactação do vídeo utilizando apenas uma camada. As abordagens para transmissão de vídeo por multicaminhos com SL adotaram os padrões H.264/AVC e H.265/HEVC [1, 4, 57]. Em geral, o vídeo é segmentado em GOP que são codificados e transmitidos em diferentes caminhos. Os melhores caminhos – selecionados com base na métrica de roteamento – são dedicados aos *frames* I, uma vez que a decodificação do vídeo é muito dependente deste tipo de *frame*, além de exigirem largura de banda maior do que os *frames* P e B, que podem ser transmitidos usando caminhos alternativos, com largura de banda menor. Os *frames* B normalmente têm o menor requisito de largura de banda e a codificação destes requer a aquisição do *frame* I ou do *frame* P correspondente, o que introduz atraso. Por esse motivo, algumas propostas utilizam apenas os modos de codificação com *frames* I e P [57].

### 3.2.1.2 Camadas Escaláveis

Na codificação de vídeo escalável (ou SVC, do inglês *Scalable Video Coding*), os fluxos de vídeo são divididos em uma camada base, e uma ou mais camadas de aprimoramento. As camadas SVC são hierárquicas. Portanto, uma determinada camada não pode ser decodificada, a menos que todas as suas camadas inferiores tenham sido recebidas corretamente. As camadas de aprimoramento complementam a camada base para melhorar a qualidade visual em termos de modalidades escaláveis temporais, espaciais e de relação sinal-ruído (ou SNR, do inglês *Signal-to-Noise Ratio*). Os padrões MPEG incluem ferramentas de extensão para suportar estes modos SVC [64]. As extensões escaláveis para os padrões H.264/AVC e H.265/HEVC são respectivamente chamadas de H.264/SVC e H.265/SHVC (do inglês, *Scalable High-definition Video Coding*) [13].

Propostas de multicaminhos baseadas em H.264/SVC geralmente transmitem a camada base pelos melhores caminhos, e as camadas de aprimoramento pelos caminhos marginais [15, 22, 27]. Além disso, a codificação H.265/SHVC foi adotada em [63, 6] devido à melhor compactação, reduzindo a latência e a taxa de bits. Com o H.265/SHVC, o controle da taxa de bits de cada camada pode ser feito de acordo com a largura de banda disponível para cada caminho.

### 3.2.2 Codificação por Múltiplos Descritores

A codificação MDC foi proposta como uma alternativa à codificação em camadas para *streaming* de vídeo [17]. Cada descritor pode, por si só, garantir um nível básico de qualidade de reconstrução no decodificador, enquanto descritores adicionais melhoram ainda mais a qualidade.

A idéia central do MDC é particionar o fluxo de vídeo em dois ou mais descritores decodificáveis independentemente e refinados mutuamente. Várias técnicas para gerar descritores de vídeo usando MDC são discutidas em [32]. O número de descritores pode ser definido de acordo com os requisitos da aplicação, e o particionamento pode ser no domínio espacial ou temporal. No domínio espacial, os descritores são gerados por um processo realizado no nível do pixel, enquanto que no domínio temporal o processo é realizado no nível do *frame*. Os pacotes individuais de cada descritor podem ser transmitidos separadamente por diferentes caminhos. Se os pacotes forem perdidos, o vídeo ainda poderá ser decodificado com sucesso usando pacotes que transportam os outros descritores, embora com menor fidelidade. Para tal, o MDC fornece solução para mitigar a degradação da qualidade de vídeo na presença de perdas de pacotes, erros de bits e erros de rajadas durante a transmissão [31].

A combinação do MDC com o roteamento por multicaminhos foi proposta para reduzir o congestionamento da rede, explorando a diversidade de caminhos para equilibrar a carga de tráfego [12, 19, 38]. Alguns estudos propõem classificar os descritores para definir prioridades de pacotes e assim melhorar a robustez da transmissão [60, 21].

Uma maneira de gerar MDC é explorar o GOP ou a estrutura do macrobloco dos padrões de codificação de vídeo MPEG. Nesse contexto, estudos sugerem a ordenação de macroblocos flexíveis (ou FMO, do inglês *Flexible Macroblock Ordering*) do padrão H.264/AVC como mais apropriada [66, 46]. A FMO refere-se à reorganização de macroblocos em grupos, onde cada grupo é um descritor do vídeo, de acordo com padrões específicos.

## 3.3 Mecanismos de Seleção de Multicaminhos

O FITPATH e o QSOpt são mecanismos de seleção de multicaminhos, baseados na topologia corrente, que possuem como foco as redes sem fio multissaltos do padrão IEEE 802.11, e destinam-se aos sistemas IoT que demandem *streaming* de vídeo ao vivo. A seguir será

apresentada uma visão geral do princípio de funcionamento de ambos os mecanismos.

### 3.3.1 FITPATH

Recentemente, um mecanismo eficiente de seleção de multicaminhos para transmissão de vídeos IoT em redes sem fio multissaltos, chamado FITPATH, foi publicado em [9]. Naquele trabalho, o cenário específico de estudo foi um sistema de videovigilância para cidades inteligentes, onde foi assumido que a maior parte dos nós da rede eram fontes de vídeo (câmeras) que transferiam fluxos de vídeo simultaneamente, de acordo com as demandas das centrais de controle e monitoramento. O desempenho do FITPATH foi avaliado em comparação a outros seis mecanismos que adotam diferentes métricas e métodos para a seleção de multicaminhos, a saber: (i) ERVT (do inglês, *Erro Resilient Video Transmission*) [29], (ii) CLMR (do inglês, *Cross-Layer Multipath Routing*) [1], (iii) RTVP (do inglês, *Real-Time Video streaming routing Protocol*) [4], (iv) Q-MMTP (do inglês, *QoS-oriented Multipath Multimedia Transmission Planning*) [35], (v) QSOpt (do inglês, *QoE-aware Sub-Optimal routing*) [54] e (vi) ILS-MDC (do inglês, *Iterator Local Search - Multiple Description Coding*) [12]. Dentre estes, o FITPATH e o QSOpt se destacaram, respectivamente, com a melhor e a segundo melhor métrica de QoE (do inglês, *Quality of Experience*) do usuário alcançada.

O FITPATH [9] é um mecanismo baseado na metaheurística ILS (do inglês, *Iterated Local Search*) [47] que seleciona caminhos para cada fluxo de vídeo gerado a partir de múltiplas fontes. Os principais componentes do FITPATH são apresentados na Figura 3.1. O FITPATH requer dois tipos de informações na entrada: (i) a topologia da rede, que consiste nos enlaces disponíveis e seus respectivos custos, computados pela métrica ETX (do inglês, *Expected Transmission Count*) [25], e (ii) os requisitos dos fluxos de vídeo, compostos pelos identificadores das fontes e seus respectivos destinatários finais, e a taxa alvo de bits do codificador para cada fluxo.

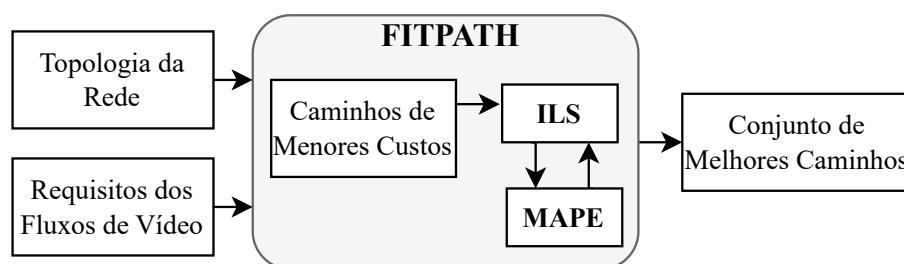


Figura 3.1: Diagrama geral do FITPATH [9].

Em linhas gerais, o FITPATH gera, para cada fluxo, um conjunto de caminhos can-

didatos de menores custos ETX com base no Algoritmo de Yen [70]. A seguir, o ILS aplica seu algoritmo de busca iterada e avalia cada solução, dentre os caminhos candidatos, utilizando o MAPE (do inglês, *Multimedia-Aware Performance Estimator*) [10], um estimador de desempenho da rede baseado em simulação determinística que fornece, em tempo real, estimativas de vazão, atraso e perda de pacotes considerando a interferência entre os fluxos. Deste modo, o FITPATH toma decisões de roteamento que propiciem maiores vazões aos fluxos de vídeo heterogêneos, ou seja, fluxos com diferentes taxas de bits que são simultaneamente transmitidos.

### 3.3.2 *QoE-aware Sub-Optimal routing (QSOpt)*

O QSOpt [54] formula o problema de seleção de múltiplos caminhos usando programação linear inteira mista, e implementa uma heurística para encontrar uma solução que maximize a QoE. Para avaliar a QoE, uma função discreta é proposta para determinar um mapeamento entre a perda de pacotes e a métrica de QoE. Embora o algoritmo centralizado forneça uma solução viável para melhorar a utilização de recursos e a QoE, o modelo do QSOpt não é escalável e requer recursos computacionais significativos à medida que o número de fluxos aumenta. O QSOpt também assume que todos os fluxos possuem a mesma taxa de bits, independentemente das características de codificação do vídeo, o que restringe a aplicabilidade, impactando na avaliação do nível de interferência entre os fluxos e, conseqüentemente, na capacidade de encontrar as melhores soluções.



# 4 Arquitetura Proposta

Este capítulo apresenta a visão geral de uma arquitetura proposta para o roteamento dinâmico em um sistema de videovigilância que requer a transmissão de múltiplos fluxos de vídeo em ambiente IoT. A arquitetura inclui um monitor de topologia com o propósito de assegurar a visão global da rede aos mecanismos de seleção de multicaminhos que são baseados na qualidade dos enlaces.

## 4.1 Visão Geral da Arquitetura

Nos sistemas de videovigilância em IoT aplicados às cidades inteligentes, os nós são tipicamente estacionários e alimentados por fontes de energia contínua. Portanto, não é esperado que ocorram mudanças frequentes na topologia por mobilidade dos nós, nem tão pouco que haja impacto significativo de escassez de energia. No entanto, nas redes em malha sem fio, as transmissões de rádio são suscetíveis a interferências, o que pode levar a mudanças de qualidade e/ou quebra de enlaces. Em situações em que as mudanças da topologia precisam ser consideradas, as informações sobre as condições da rede podem ser obtidas de diferentes formas, dependendo se o plano de controle da rede é centralizado ou descentralizado.

No controle centralizado, uma possível solução consiste do uso de um controlador SDN (*Software Defined Network*) [68] que toma as decisões de roteamento assistido por um mecanismo de seleção de caminhos, e baseado nos requisitos de tráfego dos vídeos e na topologia atualizada da rede. O monitor da topologia pode ser instalado no controlador, que recebe as informações diretamente de cada nó. Em seguida, o controlador atualiza as tabelas de roteamento dos nós de encaminhamento para que estes transmitam os fluxos de acordo com os caminhos selecionados. Contudo, o controle centralizado requer recursos de rede adicionais e introduz um ponto de falha crítico.

No controle descentralizado, cada nó mantém a visão completa da topologia da rede por um monitor. Cada estação de vigilância toma as decisões de roteamento assistida por

um mecanismo de seleção de caminhos, e baseado nos requisitos de tráfego dos vídeos e na topologia corrente da rede. Em seguida, a estação de vigilância informa os caminhos dos fluxos de vídeo para cada fonte, por meio de um pacote de requisição.

A Figura 4.1 ilustra a arquitetura proposta de uma rede *ad hoc* sob controle descentralizado, onde o nó destinatário (em cinza) é uma estação de vigilância e o nó fonte (em preto) é a câmera de vídeo. Em cada nó da rede há um módulo de seleção de multicaminhos, um monitor de topologia e a aplicação de vídeo.

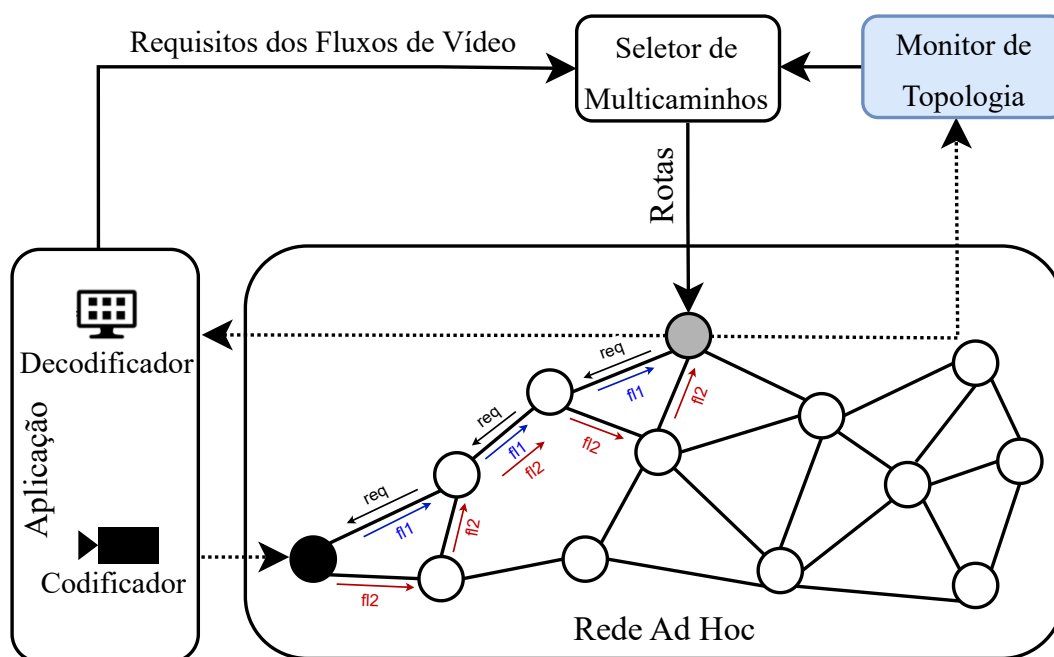


Figura 4.1: Arquitetura proposta com um plano de controle descentralizado.

A Figura 4.2 apresenta um fluxograma simplificado de processos para a transmissão de vídeo na arquitetura da Figura 4.1. O fluxograma está representado no padrão ANSI (do inglês, *American National Standards Institute*), onde os retângulos representam os processos, os paralelogramos indicam as entradas e os losangos apresentam pontos de decisão.

No controle descentralizado, objeto desta proposta, (1) todos os nós proativamente fazem *broadcast* periódico de pacotes de controle da topologia, com uma pequena diferença aleatória de tempo entre eles, permitindo que cada nó identifique seus vizinhos e os respectivos custos. Essas informações são disseminadas para garantir que todos os nós tenham a visão global da topologia da rede. (2) No instante em que a estação de vigilância gera uma demanda por vídeos; (3) a aplicação de vídeo e o monitor de topologia da estação de vigilância informam ao seletor de multicaminhos desta estação, os requisitos

dos fluxos de vídeo e a topologia atualizada. (4) O seletor de multicaminhos da estação de vigilância então seleciona as melhores rotas para os fluxos de vídeo e, em seguida; (5) essas rotas juntamente com os requisitos das taxas dos fluxos de vídeo são informadas à aplicação cliente/servidor da estação de vigilância; (6) A estação de vigilância envia a mensagem de requisição para cada uma das fontes demandadas, no sentido inverso do melhor caminho selecionado para cada fonte. As taxas requeridas de codificação e as listas completas e ordenadas dos nós através dos quais cada pacote dos fluxos de vídeo deve passar são informadas no cabeçalho de uma mensagem de requisição. (7) Os nós intermediários encaminham a mensagem de requisição até a fonte e; (8) quando a requisição chega à fonte demandada, o vídeo é codificado conforme as taxas requeridas, e os pacotes dos fluxos são transmitidos pelos caminhos selecionados usando a técnica de roteamento pela fonte, incluindo no cabeçalho de cada pacote a informação do caminho completo até o nó destinatário, evitando assim a formação de *loops* de roteamento. (9) Uma vez que o monitoramento da topologia é atualizado periodicamente na estação de vigilância, poderão ser estabelecidos gatilhos para a busca de novas soluções de caminhos, como por exemplo, a falha de um dos enlaces das rotas, ou a disponibilidade de novos enlaces na rede. A depender dos serviços de vídeos, os requisitos dos fluxos também poderão ser alterados, como por exemplo, as câmeras poderão ter suas resoluções alteradas, ou serem desabilitadas ou habilitadas. Além disso, também é possível provisionar as taxas dos fluxos de controle da topologia, em conjunto com as taxas dos fluxos de vídeo, de modo que o seletor de multicaminhos leve em consideração todos esses fluxos no processo de geração das rotas. Duas abordagens de provisionamento são discutidas na Seção 5.2.2. (10) Caso não ocorram mudanças na topologia ou nos requisitos do vídeo, as transmissões dos vídeos serão mantidas pelos caminhos preestabelecidos.

Visto que ambos os seletores de multicaminhos, FITPATH e QSOpt, consideram a interferência entre os fluxos na solução, caso haja demanda de vídeo por mais de uma estação de vigilância, é necessário que os requisitos de vídeo de cada uma das estações de vigilância sejam considerados pelas demais estações. Para tal, os requisitos de vídeo podem ser disseminados de forma semelhante ao mecanismo de disseminação da topologia apresentado na Seção 4.2.

## 4.2 Monitor de Topologia

O monitor de topologia apresentado neste trabalho é baseado no protocolo OLSR (*Optimized Link State Routing Protocol*) [20], sem o uso da técnica de MPR (*Multi-Point*

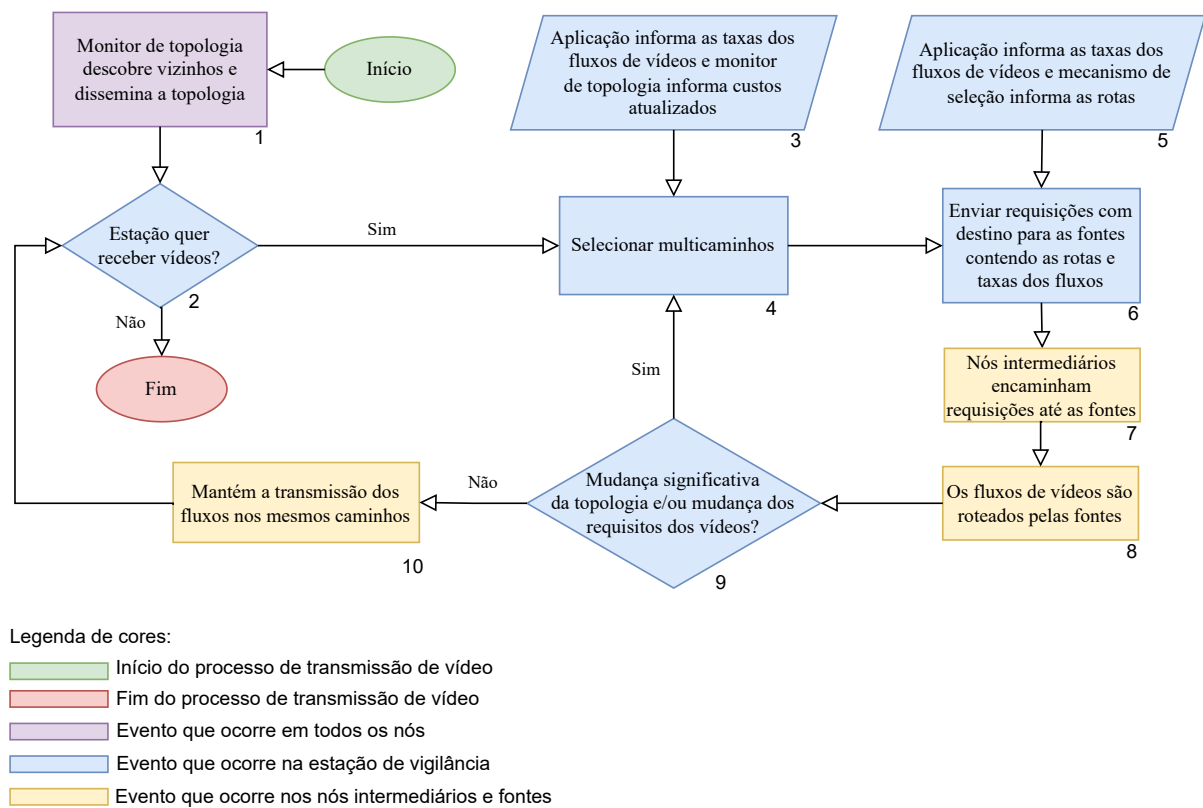


Figura 4.2: Fluxograma simplificado de processos para a transmissão de vídeos com roteamento dinâmico por multicaminhos.

*Relay*). O MPR consiste na seleção de um conjunto de nós através dos quais os nós a dois saltos são alcançáveis. Essa técnica visa a redução da inundação de pacotes na rede. No entanto, optou-se por não utilizar dessa técnica, pois em caso de perdas de pacotes de controle da topologia, a redundância das retransmissões oferecerá uma garantia maior da disponibilidade da visão total da topologia.

O monitoramento da topologia é realizado através de três processos:

- (i) a descoberta dos nós vizinhos alcançáveis por um enlace, o cômputo dos respectivos custos e o armazenamento dessas informações em uma estrutura de dados da topologia local;
- (ii) a disseminação das informações locais para os demais nós da rede, e o armazenamento dessas informações em uma estrutura de dados da topologia global; e
- (iii) o monitoramento das condições da topologia que possam levar a degradação da qualidade dos enlaces.

A descoberta dos vizinhos alcançados por um enlace é realizada através do envio

periódico de pacotes HELLO, em *broadcast*, por todos os nós. O Algoritmo 1 apresenta o procedimento de geração do tráfego dos pacotes HELLO.

---

**Algoritmo 1:** Gerador de Tráfego HELLO.

---

```

// Estrutura de dados da topologia local
1: localTopology(oneHopNeighId, LQ)
2: begin
3:   Function generateTrafficHello(socket, packet_size, seqNum, hello_interval):
4:     // Manutenção da topologia local
5:     localTopologyMaintenance(hello_interval);
6:     // Monta o pacote HELLO
7:     Create(packetHello);
8:     packetHello → addHeader(senderNodeId, seqNum, localTopology.size());
9:     for  $0 \leq i < \text{localTopology.size}()$  do
10:      | packetHello → addHeader(localTopology[i].oneHopNeighId, localTopology[i].LQ);
11:     end
12:     RemoveAtEnd(packetHello → GetSize() - packet_size);
13:     // Envia pacote HELLO no socket
14:     socket → Send(packetHello);
15:     // Agendamento de generateTrafficHello a cada hello_interval
16:     Schedule(hello_interval, &generateTrafficHello, socket, packet_size, seqNum + 1,
17:             hello_interval);
18: end

```

---

O Algoritmo 2 apresenta o procedimento de recepção do tráfego de HELLO. Quando um nó recebe um pacote HELLO, este identifica o nó de origem e verifica se ele já é reconhecido como um vizinho na sua topologia (linha 8 do Algoritmo 2), caso não seja, este nó é adicionado na topologia local (linha 23 do Algoritmo 2).

Os custos dos enlaces foram implementados com a métrica ETX. Para isto, é necessário primeiramente realizar o cômputo da qualidade do enlaces, conforme apresentado no Algoritmo 3. A cada recebimento de um pacote HELLO de um nó específico, o nó receptor computa a qualidade do enlace LQ (do inglês, *Link Quality*) como um percentual de pacotes HELLO recebidos de seu vizinho, dentro de uma janela deslizante, designada no Algoritmo 3 como *window\_size*, dos últimos  $p$  pacotes enviados por este vizinho (linha 17 do Algoritmo 3).

Para detectar o recebimento ou perda de pacotes HELLO, o nó receptor monitora o número de sequência dos pacotes (linha 12 do Algoritmo 3). Ao transmitir um pacote HELLO, o nó também informa uma lista dos nós vizinhos já descobertos, e seus respectivos LQ (linha 7 do Algoritmo 1). Os nós vizinhos, ao receberem o pacote HELLO, tratam os LQ informados pelo nó de origem como NLQ (*Neighbor Link Quality*) (linha 11 do Algoritmo 2).

O ETX [25] é definido como o número estimado de transmissões necessárias para que

**Algoritmo 2:** Recepção de Tráfego HELLO.

---

```

// Estrutura de dados da topologia local
1: localTopology(oneHopNeighId, LQ, twoHopNeighId[], NLQ[], lastSeqNum, lastTimeRcvHello,
   window[], etx)
2: begin
3:   Function receiveTrafficHello(socket):
   // Manutenção da topologia global em caso de falhas
4:   localTopologyMaintenance(hello_interval);
   // Se nó remetente já for conhecido, atualiza informações de Id e LQ dos
   vizinhos do remetente na topologia
5:   senderExists ← false;
6:   senderIndex ← 0;
7:   for 0 ≤ i < localTopology.size() do
8:     if localTopology[i].oneHopNeighId == header.senderNodeId then
9:       for 0 ≤ j < header.senderNumNeigh do
10:        localTopology[i].twoHopNeighId[j] ← header.oneHopNeighId[j];
11:        localTopology[i].NLQ[j] ← header.LQ[j];
   // Cômputo do ETX
12:        if localTopology[i].twoHopNeighId[j] == receiverNodeId then
13:          localTopology[i].etx ← 10000/(localTopology[i].LQ ×
   localTopology[i].NLQ);
14:        end
15:      end
16:      senderExists ← true;
17:      break;
18:    end
19:    senderIndex++;
20:  end
   // Se nó remetente não for conhecido, adiciona na topologia local
21:  if !senderExists then
22:    localTopology newSender;
23:    localTopology.push_back(newSender);
24:    senderIndex ← localTopology.size() - 1;
25:  end
   // Atualiza o instante do último pacote recebido do nó remetente
26:  localTopology[senderIndex].lastTimeRcvHello ← Now().GetSeconds();
   // Cômputo do LQ na janela deslizante (Algoritmo 3)
27:  countLQ (header.seqNum, senderIndex);
28: end

```

---

um quadro seja corretamente recebido pelo seu vizinho. No entanto, além da transmissão dos dados, o sucesso da transmissão na camada de enlace depende também que o quadro de reconhecimento (ou ACK) seja corretamente recebido. Logo, a métrica ETX da transmissão de um nó  $A$  para um nó  $B$  é calculada pelo nó  $B$  como:

$$ETX_{AB} = \frac{1}{LQ_{AB} \times NLQ_{BA}}, \quad (4.1)$$

onde  $LQ_{AB}$  é computado pelo nó  $B$  como o percentual de pacotes HELLO recebidos de  $A$ , dentro do período da janela deslizante, e  $NLQ_{BA}$  é igual ao  $LQ_{BA}$  computado pelo

**Algoritmo 3:** Cômputo da Qualidade dos Enlaces na Janela Deslizante.

---

```

// Estrutura de dados da topologia local
1: localTopology(oneHopNeighId, LQ, twoHopNeighId[], NLQ[], lastSeqNum, lastTimeRcvHello,
   window[], etc)
2: begin
3:   Function countLQ(header.seqNum, senderIndex):
4:     // computa o pacote recebido na janela deslizante
5:     indexRecPackets  $\leftarrow$  (header.seqNum-1)%window_size;
6:     localTopology[senderIndex].window[indexRecPackets]  $\leftarrow$  1;
7:     // computa os pacotes não recebidos na janela deslizante
8:     diffPackets  $\leftarrow$  header.seqNum - localTopology[senderIndex].lastSeqNum;
9:     if diffPackets > 1 then
10:      for 0 < i  $\leq$  diffPackets do
11:        | localTopology[senderIndex].window[(header.seqNum-i)%window_size]  $\leftarrow$  0;
12:      end
13:     end
14:     // Atualiza número de sequência do último pacote recebido do nó
15:     remetente
16:     localTopology[senderIndex].lastSeqNum  $\leftarrow$  header.seqNum;
17:     // total de pacotes recebidos na janela deslizante
18:     lqCount  $\leftarrow$  0;
19:     for 0  $\leq$  i < window_size do
20:       | lqCount  $\leftarrow$  lqCount + localTopology[senderIndex].window[i];
21:     end
22:     localTopology[senderIndex].LQ  $\leftarrow$  (lqCount  $\times$  100) / window_size;
23:   end

```

---

nó  $A$  como o percentual de pacotes HELLO recebidos de  $B$ , dentro do período da janela deslizante, e posteriormente informado ao nó  $B$ , através da mensagem HELLO (linha 13 do Algoritmo 2). Portanto, a grandeza de  $ETX_{AB}$  representa o número provável de tentativas necessárias para que um quadro de dados do nó  $A$  seja recebido corretamente pelo nó  $B$  e o quadro de reconhecimento do nó  $B$  também seja corretamente recebido pelo nó  $A$ . Considerando-se uma rota completa, o número total de transmissões na camada de enlace é estimado como a soma dos valores de ETX de todos os enlaces da rota.

À medida que as informações locais são conhecidas, os nós disseminam sua visão da rede para os demais nós, por meio de pacotes periódicos de controle de topologia TC (do inglês, *Topology Control*), conforme apresentado no Algoritmo 4.

O Algoritmo 5 mostra o processo de recepção dos pacotes TC. Quando um pacote TC é recebido por um nó, este armazena as informações em uma estrutura de dados da topologia global (linhas 24 a 25 do Algoritmo 5), e o pacote em seguida é retransmitido em *broadcast* (linha 27 do Algoritmo 5).

Para evitar sobrecarga na rede e manter a topologia atualizada, é realizado o controle do número de sequência dos pacotes TC. Deste modo, um nó somente armazena

**Algoritmo 4:** Gerador de Tráfego TC.

---

```

// Estrutura de dados da topologia local
1: localTopology(oneHopNeighId, etx)
2: begin
3:   Function generateTrafficTC(socket, packet_size, seqNum, tc_interval):
4:     // Manutenção da topologia global
     localTopologyMaintenance(tc_interval);
     // Monta o pacote TC
5:     Create(packetTC);
6:     packetTC → addHeader(senderNodeId, seqNum, localTopology.size());
7:     for  $0 \leq i < \text{localTopology.size}()$  do
8:       | packetTC → addHeader(localTopology[i].oneHopNeighId, localTopology[i].etx);
9:     end
10:    RemoveAtEnd(packetTC → GetSize() - packet_size);
     // Envia pacote TC no socket
11:    socket → Send(packetTC);
     // Agendamento de generateTrafficTC a cada tc_interval
12:    seqNum ← seqNum + 1;
13:    Schedule(tc_interval, &generateTrafficTC);
14: end

```

---

e retransmite as informações de um pacote TC, caso o número de sequência do pacote recebido seja maior que o número de sequência anteriormente armazenado (linha 20 do Algoritmo 5).

Caso ocorra uma falha na interface de um nó, o nós vizinhos que eram alcançados por um enlace deixarão de receber os pacotes HELLO do nó falhante, e conseqüentemente deixarão de atualizar a janela deslizante do LQ na sua topologia local. Por esse motivo foi estabelecido um limite de perdas consecutivas menor que  $n$  pacotes HELLO para a atualização da janela deslizante, e menor que  $m$  pacotes TC para a atualização da topologia global. Caso o período de perdas consecutivas de  $n$  pacotes HELLO, designado no Algoritmo 6 como *hello\_loss\_tolerance*, seja atingido, os nós vizinhos que eram alcançados por um enlace do nó falhante excluirão, de sua topologia local, o nó falhante e demais dados referentes a este, incluindo a sua janela deslizante (linha 7 do Algoritmo 6), e no caso de qualquer nó computar  $m$  perdas consecutivas de pacotes TC do nó falhante, designada no Algoritmo 7 como *tc\_loss\_tolerance*, esses nós excluirão o nó falhante da sua topologia global (linha 7 do Algoritmo 7).

O Algoritmo 8 apresenta o processo do monitor de topologia. Esse algoritmo possibilita a geração de novas rotas pelo mecanismo de seleção de caminhos (linha 5 do Algoritmo 8) sempre que uma mudança na topologia é identificada (linha 4 do Algoritmo 8). Dessa forma, o roteamento pode ser adaptado de acordo com as novas rotas (linha 6 do Algoritmo 8), garantindo a QoE mais apropriada para a atual configuração



**Algoritmo 5:** Recepção de Tráfego TC.

---

```

// Estrutura de dados da topologia global
1: globalTopology(nodeTC, lastSeqNumTC, neighborsTC[], etxTC[], lastTimeRcvTC)
2: begin
3:   Function receiveTrafficTC(socket):
4:     // Manutenção da topologia global em caso de falhas
5:     globalTopologyMaintenance(tc_interval);
6:     // Verifica se nó remetente já é conhecido na topologia global
7:     senderExists ← false;
8:     senderIndex ← 0;
9:     for  $0 \leq i < \text{globalTopology.size}()$  do
10:      if globalTopology[i].nodeTC == header.senderNodeId then
11:        senderExists ← true;
12:        break;
13:      end
14:      senderIndex++;
15:    end
16:    // Se nó remetente não for conhecido, adiciona na topologia global
17:    if !senderExists then
18:      globalTopology newSender;
19:      globalTopology.push_back(newSender);
20:      senderIndex ← globalTopology.size() - 1;
21:    end
22:    // Se o pacote TC for mais recente, atualiza a topologia global e
23:    // retransmite o pacote
24:    diffPackets = header.seqNum - globalTopology[senderIndex].lastSeqNumTC;
25:    if diffPackets > 0 then
26:      globalTopology[senderIndex].lastSeqNumTC ← header.seqNum;
27:      globalTopology[senderIndex].lastTimeRcvTC ← Now().GetSeconds();
28:      for  $0 < i < \text{header.senderNumNeigh}$  do
29:        globalTopology[senderIndex].neighborsTC[i] ← header.oneHopNeighId[i];
30:        globalTopology[senderIndex].extTC[i] ← header.etx[i];
31:      end
32:      socket → Send(packetTC);
33:    end
34:  end
35: end

```

---

**Algoritmo 6:** Manutenção da Topologia Local.

---

```

// Estrutura de dados da topologia local
1: localTopology(lastTimeRcvHello)
2: begin
3:   Function localTopologyMaintenance(hello_interval):
4:     // Exclui vizinhos que ultrapassam o limite de pacotes perdidos
5:     hello_loss_tolerance ← n;
6:     for  $0 \leq i < \text{localTopology.size}()$  do
7:       if (Now().GetSecond() - localTopology[i].lastTimeRcvHello) ≥
8:         hello_loss_tolerance × hello_interval then
9:         localTopology.erase(localTopology[i]);
10:      end
11:    end
12:  end
13: end

```

---

**Algoritmo 7:** Manutenção da Topologia Global.

---

```

// Estrutura de dados da topologia global
1: globalTopology(lastTimeRcvTC)
2: begin
3:   Function localTopologyMaintenance(tc_interval):
4:     // Exclui vizinhos que ultrapassam o limite de pacotes perdidos
5:     tc_loss_tolerance ← m;
6:     for  $0 \leq i < \text{globalTopology.size}()$  do
7:       if (Now()).GetSecond() - globalTopology[i].lastTimeRcvTC ≥ tc_loss_tolerance
8:         × tc_interval then
9:           globalTopology.erase(globalTopology[i]);
10:        end
11:   end

```

---

da topologia. Considerando que a topologia é atualizada em intervalos regulares definidos como *tc\_interval*, o monitoramento da topologia ocorre com a mesma periodicidade desse intervalo.

**Algoritmo 8:** Monitor de Topologia.

---

```

// Estrutura de dados da topologia global
1: globalTopology(lastTimeRcvTC)
2: begin
3:   Function topologyMonitor(lastTCSize, sources, numNodes, containerClients):
4:     // Chama mecanismo de seleção de caminhos se houver mudança no tamanho
5:     // da topologia global
6:     if globalTopology.size() ≠ lastTCSize then
7:       RecallMultipathSelection (sources, numNodes);
8:       ScheduleNow(&EvalvidClient::Recall, containerClients);
9:     end
10:    // Atualiza o tamanho da topologia para controle
11:    lastTCsize ← globalTopology.size();
12:    // Monitora a topologia global a cada tc_interval
13:    Schedule(tc_interval, &TopologyMonitor);
14:  end

```

---

## 4.3 Roteamento

Após a seleção das rotas, é necessário que a estação de vigilância informe a cada fonte, por meio de um pacote de requisição, os caminhos para a transmissão do vídeo e a taxa alvo de codificação. A função `handleRead()` no Algoritmo 9 foi desenvolvida para gerenciar o roteamento pela fonte.

O caminho utilizado para o envio do pacote de requisição é o melhor caminho gerado pelo mecanismo de seleção, porém no sentido inverso, ou seja, da estação de vigilância para a fonte.

Quando um nó intermediário recebe a requisição, ele verifica se seu identificador (Id) corresponde ao Id da fonte demandada. Para tal, o nó verifica se seu Id é igual ao último elemento do primeiro caminho indicado no cabeçalho do pacote. Se o Id do último elemento for diferente do Id do nó intermediário (linha 4 do Algoritmo 9), o pacote de requisição é encaminhado para o nó subsequente ao seu no cabeçalho (linha 8 do Algoritmo 9).

Ao atingir a fonte demandada, as informações do cabeçalho da requisição são passadas para uma estrutura de dados e o vídeo é preparado para o envio por meio da função `videoSetup()`. Os pacotes dos fluxos de vídeo recebem os caminhos dos nós em seu cabeçalho e são encaminhados de acordo com as técnicas de codificação LC ou MDC apresentadas na Seção 3.2. A função `handleRead()` gerencia o roteamento de forma semelhante ao pacote de requisição, porém no sentido da fonte para a estação de vigilância.

## 4.4 Implementação

A arquitetura proposta nesta dissertação foi implementada no NS-3 (do inglês, *Network Simulator 3*) [16, 39], um simulador de redes de eventos discretos, direcionado principalmente para a pesquisa e uso educacional. Esse simulador foi escolhido por ser uma ferramenta de código aberto, licenciado pela GNU GPLv2, e mantido por uma comunidade mundial. Além disso, o *software* é modular e possui bom desempenho para simulações de rede.

A programação do NS-3 é feita em linguagem C++. Seu funcionamento básico consiste na criação de nós, seus canais de comunicação e a geração de tráfego na rede. No NS-3, a abstração básica para um programa de usuário que gera alguma atividade a ser simulada é a aplicação. Esta por sua vez itera com o protocolo de transporte por meio de soquetes.

Para o desenvolvimento do monitor de topologia, foram primeiramente implementadas duas aplicações, conforme descrito a seguir:

- (i) Uma aplicação para gerar a topologia local, em uma estrutura de dados, através de transmissões periódicas do pacote HELLO; e
- (ii) Uma aplicação para gerar a topologia global, em uma estrutura de dados, através de transmissões periódicas do pacote TC.

O protocolo do monitor de topologia possui alguns parâmetros configuráveis, cujas

**Algoritmo 9:** Roteamento pela Fonte

---

```

// Estrutura de dados dos caminhos
1: pathInfo(firstPathSize, firstPath, secondPathSize, secondPath, bitrate)
2: begin
3:   Function handleRead(socket):
      // Se o Id não corresponder ao Id do destinatário final, identifica o Id
      // do próximo salto
4:   if nodeId != header.firstPath[header.firstPathSize - 1] then
5:     for 0 ≤ i < header.firstPathSize do
6:       if nodeId == header.firstPath[i] then
7:         nextHop ← header.firstPath[i+1];
8:         socket → Send(packet);
9:       end
10:    end
11:  end
      // Se o Id for igual ao destinatário final, adiciona informações do
      // cabeçalho na estrutura de dados e prepara o vídeo para o envio
12:  else
13:    pathInfo.clear();
14:    newPath;
15:    newPath.firstPathSize = header.firstPathSize;
16:    newPath.secondPathSize = header.secondPathSize;
17:    newPath.bitrate = header.bitrate;
      // Inverte o sentido do primeiro caminho
18:    for 0 ≤ i < header.firstPathSize do
19:      | newPath.firstPath[i] ← header.firstPath[header.firstPathSize - 1 - i];
20:    end
21:    for 0 < i < header.secondPathSize do
22:      | newPath.secondPath[i] ← header.secondPath[i];
23:    end
      // Adiciona informações na estrutura de dados
24:    pathInfo.push_back(newPath);
      // Prepara o vídeo para ser enviado pelos caminhos do cabeçalho
25:    videoSetup();
26:  end
27: end

```

---

descrições e os valores<sup>1</sup> utilizados nos experimentos deste trabalho estão apresentados na Tabela 1.

O *framework* EvalVid [24] possibilita a transmissão de vídeos a partir de um modelo cliente/servidor em ambiente de rede. Ele gera o tráfego correspondente a um videoclipe, o que confere às simulações características mais realistas. Além disso, o EvalVid oferece um conjunto ferramentas para codificação, decodificação, geração dos arquivos de rastreamento (*trace*) e mensuração da qualidade do vídeo entregue ao receptor.

Para preparar o arquivo de vídeo para a transmissão, um arquivo de vídeo original, no formato YUV, é codificado para o formato MPEG-4. A partir do arquivo de vídeo em

---

<sup>1</sup>Ressalta-se que estes são valores típicos usados por protocolos proativos como o OLSR [20] e adequados para as topologias utilizadas neste trabalho.

Tabela 1: Parâmetros do protocolo do Monitor de topologia.

Parâmetro	Descrição	Valor
window_size	Janela deslizante com tamanho de $p$ pacotes HELLO	100
hello_interval	Intervalo de transmissão dos pacotes HELLO em segundos	2
tc_interval	Intervalo de transmissão dos pacotes TC em segundos	5
hello_loss_tolerance	Limite de $n$ pacotes HELLO perdidos consecutivamente	20
tc_loss_tolerance	Limite de $m$ pacotes TC perdidos consecutivamente	8
packet_size	Tamanho dos pacotes HELLO e TC em bytes	360

MPEG-4, é gerado um arquivo de *trace* do vídeo por meio de um *parser*.

O Evalvid utiliza o arquivo de *trace* do vídeo para realizar as transmissões, via UDP, no NS-3. Um arquivo de *trace* do vídeo contém informações essenciais, incluindo o número do quadro, o tipo de quadro (I, P ou B), o tamanho do quadro em bytes, o número de pacotes e o tempo necessário para a transmissão do quadro.

Durante as simulações no NS-3, o Evalvid envia os pacotes por multicaminhos, de acordo com a técnica de codificação de transmissão LC ou MDC, e produz dois arquivos *trace* de saída, um do remetente e outro do receptor. Os registros contidos nestes arquivos incluem as informações de cada pacote como: *timestamp*, identificador do pacote e tamanho de *payload* do pacote. À medida que a rede introduz atrasos e eventuais perdas, o arquivo *trace* de saída do receptor apresentará pacotes desordenados e/ou faltando, o que pode refletir em perda de qualidade na recepção do vídeo.

Para avaliar a qualidade do vídeo recebido, a ferramenta `etmp4` do Evalvid reconstrói o vídeo no formato MPEG-4, a partir dos arquivos de *trace* de saída. Posteriormente a ferramenta `FFmpeg` [28] converte o vídeo do formato MPEG-4 para o YUV. A ferramenta `psnr` é um utilitário do Evalvid que compara o arquivo de vídeo YUV original com o recebido na simulação, e gera a métrica de qualidade para cada quadro do vídeo.

Para que a estação de vigilância solicite vídeos às fontes e que estas transmitam os pacotes de dados dos vídeos, foi implementada uma aplicação, no módulo do Evalvid, que roteia os pacotes pela fonte, seguindo os caminhos indicados no cabeçalho dos pacotes. Os formatos dos cabeçalhos de todos os pacotes implementados estão detalhados na

Seção 4.4.1.

### 4.4.1 Formato dos Pacotes

Esta seção detalha o formato dos pacotes implementados para assegurar o funcionamento adequado do monitor de topologia, ou seja, os pacotes HELLO e TC. Além disso, são apresentados os formatos dos pacotes indispensáveis tanto para o roteamento da requisição quanto para o roteamento dos dados dos vídeos.

#### 4.4.1.1 Pacote HELLO

O formato do pacote HELLO, ilustrado na Figura 4.3, é composto por 5 campos, cujas funcionalidades são descritas a seguir:

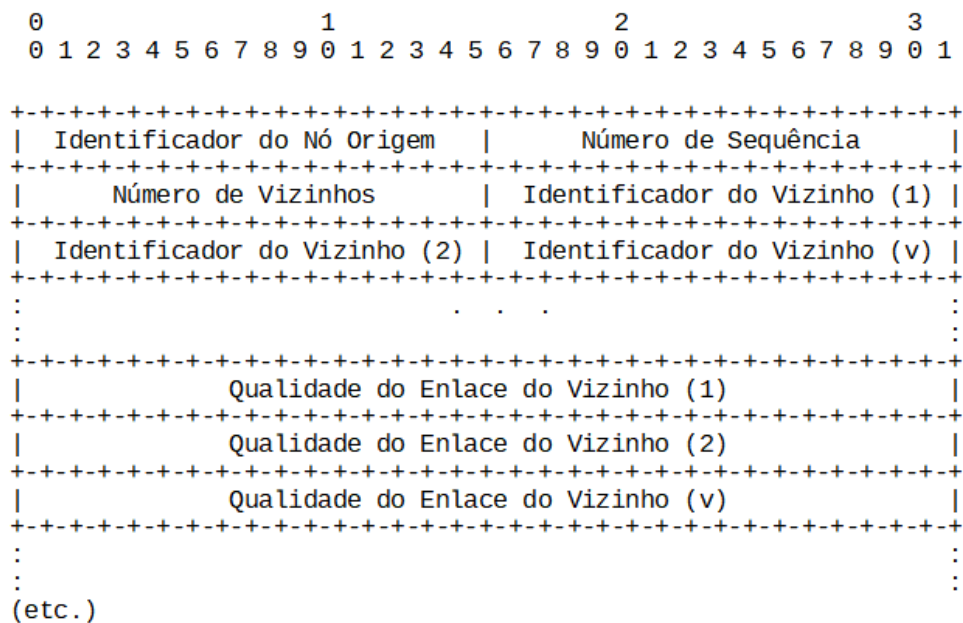


Figura 4.3: Formato do pacote HELLO.

- **Identificador do Nó Origem:** Campo de 16 bits que contém o identificador do nó que transmite o pacote. Para as topologias utilizadas neste trabalho, o tamanho deste campo é suficiente.
- **Número de Sequência:** Campo de 16 bits que contém o número de sequência do pacote. Este campo junto com o identificador do nó de origem, possibilita o cálculo do LQ da janela deslizante de cada vizinho.



- **Número de Vizinhos:** Campo de 16 bits que contém o valor da quantidade total de vizinhos computados pelo nó na topologia local. Permite a deserialização de todos os identificadores dos nós vizinhos que são informados nos campos subsequentes.
- **Identificador do Vizinho:** Campo de 16 bits que contém o identificador de um nó vizinho computado na estrutura de dados da topologia local. A depender do número de vizinhos, esse campo se repete em mais 16 bits para cada um dos  $v$  vizinhos.
- **ETX:** Campo de 32 bits que se repete para cada um dos  $v$  vizinhos. Este campo contém o cômputo de ETX, que pode assumir valores a partir de 1 para cada vizinho.

#### 4.4.1.3 Pacote de Requisição

O formato do pacote de requisição, ilustrado na Figura 4.5, é composto por 3 campos, cujas funcionalidades são descritas a seguir:

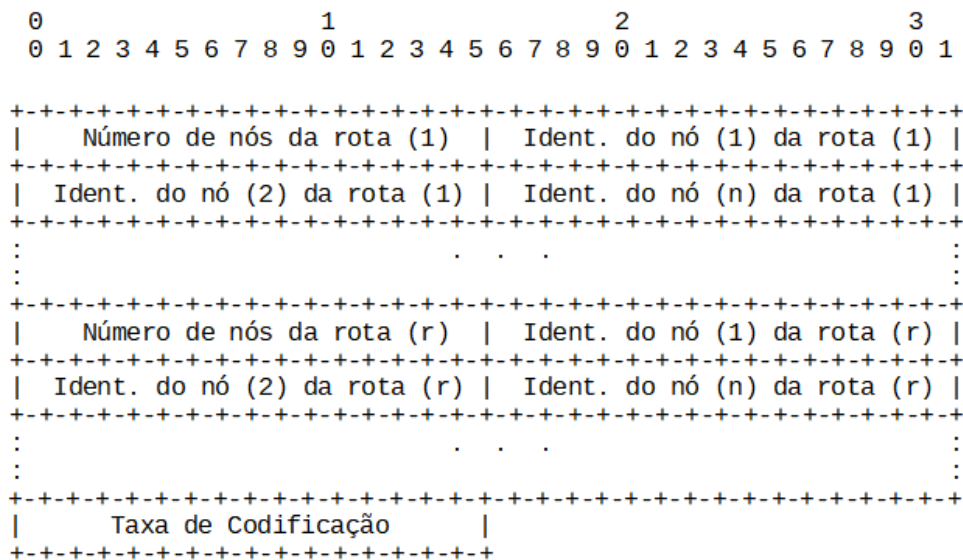


Figura 4.5: Formato do pacote de requisição.

- **Número de nós da rota  $r$ :** Campo de 16 bits que contém o valor correspondente à quantidade total de nós que compõem cada rota  $r$ . Permite a deserialização de todos os identificadores dos nós que são informados nos campos subsequentes. O número de rotas é determinado pelo número de fluxos de vídeo requeridos a uma fonte.
- **Identificador do nó  $n$  da rota  $r$ :** Campo de 16 bits que contém o identificador do nó  $n$  que compõe a rota  $r$ . A depender do número de nós da rota, esse campo se repete em mais 16 bits nos campos subsequentes. Os nós são serializados com as



rotas selecionadas pelo mecanismo de seleção de multicaminhos, porém a primeira rota é serializada no sentido inverso ao sentido dos fluxos de vídeo, ou seja, no sentido da estação de vigilância para a fonte. O pacote de requisição é transmitido através da primeira rota. As demais rotas também são informadas à fonte para a posterior transmissão dos múltiplos fluxos de vídeo.

- **Taxa de Codificação:** Campo de 16 bits que informa a taxa alvo da codificação em kb/s. A taxa dependerá da resolução requerida na aplicação da estação de vigilância.

#### 4.4.1.4 Pacotes de Dados

O formato do pacote de dados, ilustrado na Figura 4.6, é composto por 4 campos, cujas funcionalidades são descritas a seguir:

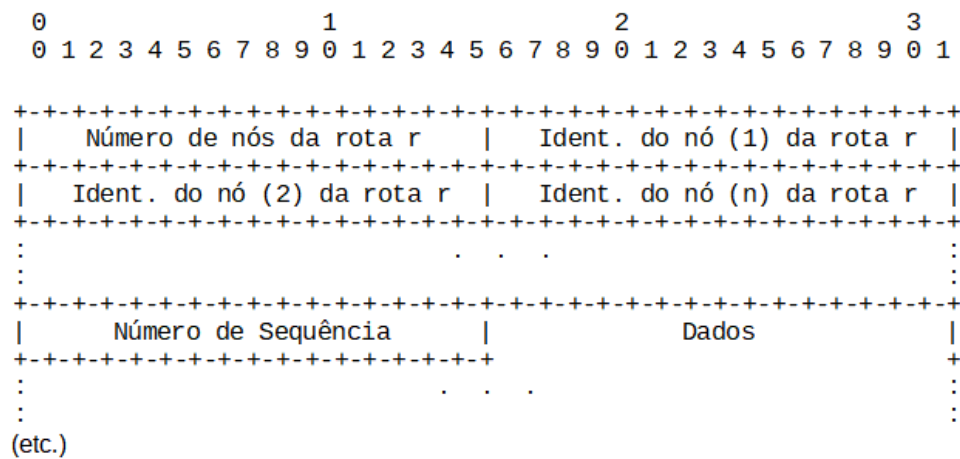


Figura 4.6: Formato do pacote dos fluxos de vídeo.

- **Número de nós da rota  $r$ :** Campo de 16 bits que contém o valor da quantidade total de nós que compõem a rota de um fluxo de vídeo. Permite a deserialização de todos os identificadores dos nós que são informados nos campos subsequentes.
- **Identificador do nó  $n$  da rota  $r$ :** Campo de 16 bits que contém o identificador do nó  $n$  que compõe a rota de um fluxo de vídeo. A depender do número de nós da rota, esse campo se repete em mais 16 bits nos campos subsequentes. Cada fluxo serializa os nós com a sua respectiva rota informada no pacote da requisição.
- **Número de Sequência:** Campo de 16 bits que contém o número de sequência do pacote transmitido. Este campo serve para o rastreamento dos pacotes nos dois arquivos *trace* de saída do remetente e do receptor.

- 
- **Dados:** Campo que contém o pacote de dados do vídeo. Neste trabalho, o pacote do vídeo possui até 1024 bytes.

# 5 Avaliações Experimentais

Este capítulo apresenta a metodologia e os resultados obtidos por meio de experimentos que foram conduzidos com o objetivo de responder as questões delineadas na Seção 1.2.

Primeiramente, foram realizadas simulações que visavam avaliar o impacto dos fluxos de controle da topologia na qualidade do vídeo, em cenários com diferentes cargas ofertadas, diferentes níveis de mobilidade de cena, diferentes técnicas de codificação e utilizando diferentes mecanismos de seleção de multicaminhos.

Também foram realizadas simulações utilizando duas abordagens simples de provisão das taxas dos fluxos de controle da topologia, juntamente com as taxas dos fluxos de vídeo, ao mecanismo de seleção de caminhos FITPATH. Estes experimentos foram realizados em cenários com diferentes cargas ofertadas, com o objetivo de avaliar uma possível redução das perdas de QoE de usuário, quando as rotas selecionadas levam em conta os fluxos de controle da topologia na rede.

Por fim, foi realizada uma simulação utilizando o mecanismo FITPATH, a título de exemplo, para demonstrar a capacidade da arquitetura proposta restaurar a QoE do usuário, através do estabelecimento de novas rotas após a falha de um nó intermediário, bem como na condição de otimização das soluções de caminhos pelo mecanismo de seleção, ao longo do tempo. Neste exemplo, também foi analisada a evolução da métrica de roteamento ETX, ao longo do tempo, pelos caminhos iniciais e otimizados - tanto antes quanto após a falha, a fim de verificar se o aumento do ETX necessariamente reflete em degradação da QoE do usuário, e vice-versa. A métrica utilizada para a avaliação da QoE do usuário foi o Índice de Similaridade Estrutural (ou SSIM, do inglês *Structural Similarity Index Measure*).

## 5.1 Metodologia de Avaliação

Os experimentos foram realizados no simulador de redes NS-3 utilizando um servidor dedicado com sistema operacional Ubuntu 16.04.6 LTS, processador Intel i7-860 de 2,8 GHz e 32 GB de RAM (do inglês, *Random Access Memory*). O cenário de aplicação modela um sistema IoT de videovigilância típico de áreas urbanas, conforme descrito na Seção 2.4.1, onde as fontes de vídeo transmitem os fluxos de vídeo simultaneamente para uma única estação de vigilância e monitoramento.

Em cada simulação, 59 nós foram alocados aleatoriamente em uma região de 300 m  $\times$  300 m, separados uns dos outros por uma distância mínima de 5 metros, e com um alcance de transmissão de 100 metros. Em complemento, um nó destinatário foi fixado em um vértice da área, na coordenada em metros (300, 300), totalizando 60 nós. As fontes geradoras dos vídeos foram posicionadas dentre as dez seguintes coordenadas em metros: (0, 0), (0, 50), (0, 100), (0, 150), (0, 200), (50, 0), (100, 0), (150, 0), (200, 0) e (50, 50).

Na camada MAC (*Medium Access Control*) foi adotado o padrão IEEE 802.11g, que embora seja capaz de transmitir a 54 Mb/s, foi adotada a taxa de transmissão fixa de 18 Mb/s para avaliar cenários com níveis mais desafiadores de congestionamento da rede. No canal sem fio foi utilizado o modelo de propagação Cost231 [61] em 2,4 GHz, que reproduz as condições mais realistas de um ambiente urbano, com obstáculos. Conforme visto na Seção 4.1, nos sistemas de videovigilância em IoT, os nós são tipicamente estacionários, e por isso o modelo de mobilidade adotado foi o de posição fixa dos nós. Esses parâmetros descritos são similares aos que foram adotados na avaliação de desempenho do FITPATH e do QSOpt em [9].

O tempo necessário para o cômputo completo das LQ nas janelas deslizantes dos nós vizinhos é de  $window\_size \times hello\_interval$ , ou seja, 200 segundos, conforme os parâmetros estabelecidos na Tabela 1. Além disso, é necessário um tempo adicional para a posterior disseminação da topologia local. Nas topologias empregadas neste trabalho, 200 segundos foram suficientes para permitir que o nó destinatário tenha a visão global da rede. Por esta razão, o instante de 400 segundos, a partir do início da simulação, foi estabelecido para a inicialização do mecanismo de seleção de multicaminhos e iniciar a transmissão dos vídeos.

### 5.1.1 Modelo de Tráfego

Os experimentos empregaram videoclipes comumente utilizados e disponíveis publicamente<sup>1</sup>, conhecidos como *Hall Monitor*, *Bus* e *Coastguard*, que representam cenas de diferentes níveis de complexidade de movimento, ou seja, baixo, médio e alto nível de movimento, respectivamente, na tentativa de reproduzir a diversidade de tráfego de vídeo em ambiente de cidades inteligentes [58]. Os videoclipes foram convertidos para o formato H.264/AVC com taxa de 30 quadros por segundo.

Os tráfegos de vídeo foram gerados usando uma combinação de cinco diferentes taxas alvo de codificação de vídeo: 256 kb/s, 512 kb/s, 1 Mb/s, 1,5 Mb/s e 2 Mb/s, para representar diferentes níveis de resolução de vídeo. Para cada fonte foram gerados 2 fluxos de vídeo com taxas semelhantes, que foram encaminhados de acordo com as técnicas de codificação LC ou MDC. Na codificação LC, os pacotes dos *frames* I foram transmitidos pelo caminho de melhor qualidade, enquanto que os pacotes dos demais *frames* foram transmitidos pelo segundo melhor caminho. Na codificação MDC, os pacotes foram distribuídos de modo que os ímpares foram transmitidos pelo melhor caminho, e os pares seguiram pelo segundo melhor caminho. Além disso, foi definido nas simulações um *playout buffer* de  $t$  ms para mitigar potenciais pacotes fora de ordem; pacotes com atraso superior a  $t$  ms foram descartados no decodificador.

Os parâmetros de rede e da aplicação utilizados nas simulações estão relacionados na Tabela 2.

Tabela 2: Parâmetros das Simulações.

Rede		Aplicação	
Área	300 × 300 m	Tipo de tráfego	VBR ( <i>Variable Bit Rate</i> )
Topologia da rede	Aleatória	Codificação	H.264/MPEG-4 AVC
Nº de nós	60	Formato	YUV CIF, 352 x 288
Dist. mín. entre nós	5 m	Taxa de quadros	30 fps ( <i>frames/s</i> )
Alcance de Tx dos nós	100 m	Mov. da cena	Baixa, média e alta
Camada de transporte	UDP	<i>Payload</i>	1024 B
Camada MAC/PHY	802.11g	Nº de nós coletores	1
Banda de frequência	2.4 GHz	Nº de nós fontes	até 4
Taxa de dados	18 Mb/s	Nº de fluxos	até 8 (2 por fonte)
Modelo de Propagação	Cost231	Taxa de codificação	256 kb/s, 512 kb/s, 1 Mb/s, 1.5 Mb/s e 2 Mb/s
Mobilidade	Estática	Carga oferecida	1 Mb/s, 2 Mb/s, 4 Mb/s, 6 Mb/s e 8 Mb/s

<sup>1</sup>Site: <https://media.xiph.org/video/derf/>, acessado em fevereiro de 2024.

### 5.1.2 Métrica de Avaliação da QoE do Usuário

A qualidade do vídeo recebido foi avaliada através do Índice de Similaridade Estrutural (ou SSIM, do inglês *Structural Similarity Index Measure*) [65], que é uma métrica objetiva amplamente utilizada para mensurar a QoE do usuário [52]. O SSIM mede a distorção estrutural do vídeo. Para isso, combina luminância, contraste e semelhança estrutural dos quadros para comparar cada quadro original com o correspondente entregue (possivelmente distorcido). Os valores do SSIM variam entre 0 e 1, onde 1 significa a qualidade máxima.

## 5.2 Resultados

Esta seção apresenta os resultados de 3 experimentos realizados com o objetivo de:

- (i) avaliar o impacto dos fluxos de controle da topologia na qualidade dos vídeos recebidos;
- (ii) analisar duas possíveis soluções de provisionamento para minimizar o impacto dos fluxos de controle da topologia na qualidade dos vídeos recebidos;
- (iii) analisar a evolução da qualidade dos vídeos recebidos, através de um exemplo, no contexto de otimização da solução de multicaminhos e na mudança da topologia em caso de falha de um nó intermediário; e
- (iv) analisar a evolução da métrica de roteamento ETX, ao longo do tempo, a fim de verificar uma possível correlação entre a métrica de roteamento ETX e a QoE do usuário.

### 5.2.1 Impacto dos Fluxos de Controle na QoE do Usuário

Neste experimento, as transmissões de vídeo foram realizadas por 4 fontes, simultaneamente, em cenários de diferentes cargas ofertadas. Cada carga ofertada consiste na soma das taxas de bits alvo de todos os fluxos de vídeo transmitidos. Os cenários também foram avaliados em transmissões de vídeo com cenas de baixa e alta mobilidade, com codificações LC e MDC, respectivamente, porque na pesquisa de [8], o desempenho da QoE do usuário foi avaliado nas transmissões dos fluxos de vídeo por rotas selecionadas pelo FITPATH, em cenas de alta e baixa mobilidade, nas codificações LC e MDC. Os

resultados apresentaram um desempenho da QoE do usuário superior nas transmissões de cenas de vídeo de baixa mobilidade que empregaram a codificação LC, em comparação com a codificação MDC. Por outro lado, nas transmissões de vídeo com cenas de alta mobilidade, a codificação MDC gerou melhores resultados de QoE do usuário quando comparados com as transmissões em LC. Em virtude disto, neste trabalho, as transmissões de cenas de vídeo de baixa mobilidade utilizaram a técnica de codificação LC, ao passo que as transmissões de cenas de vídeo de alta mobilidade foram realizadas com a codificação MDC.

Para cada carga ofertada, foram simuladas 30 instâncias de topologia em duas condições: (i) a **realista** - onde os fluxos de controle da topologia são transmitidos simultaneamente com os fluxos de vídeos, e (ii) a **não-realista** - onde ocorre a interrupção das transmissões dos fluxos de controle após o cômputo da topologia e o início da transmissão dos vídeos <sup>2</sup>.

### 5.2.1.1 Cenas de Vídeo de Baixa Mobilidade

O gráfico da Figura 5.1 apresenta os resultados de QoE nas duas condições de transmissão dos fluxos de controle da topologia (não-realista e realista), com cargas ofertadas de 1, 2, 4, 6 e 8 Mb/s, durante o tráfego de 300 quadros do videoclipe **Hall Monitor** (cenas de baixa mobilidade) a partir de cada fonte, com codificação LC e *playout buffer* de 480 ms nas soluções de caminhos do FITPATH e QSOpt. O resultado da QoE foi computado para cada carga ofertada como o valor médio de SSIM com um intervalo de confiança de 95%.

Quando as transmissões dos vídeos foram realizadas em concorrência com os fluxos de controle, a média de QoE dos vídeos recebidos pelas soluções do FITPATH e do QSOpt sofreram, respectivamente, uma redução de 3,14% e 2,00% em comparação com a média de QoE dos vídeos recebidos sem a concorrência dos fluxos de controle. Os resultados do SSIM médio, do desvio-padrão amostral, dos valores máximo e mínimo de SSIM e percentuais de impacto dos fluxos de controle no SSIM para cada carga ofertada, nas cenas de vídeo de baixa mobilidade, encontram-se sumarizados no Apêndice A.1.

---

<sup>2</sup>Os experimentos de [9] e [54] foram realizados na condição não-realista.

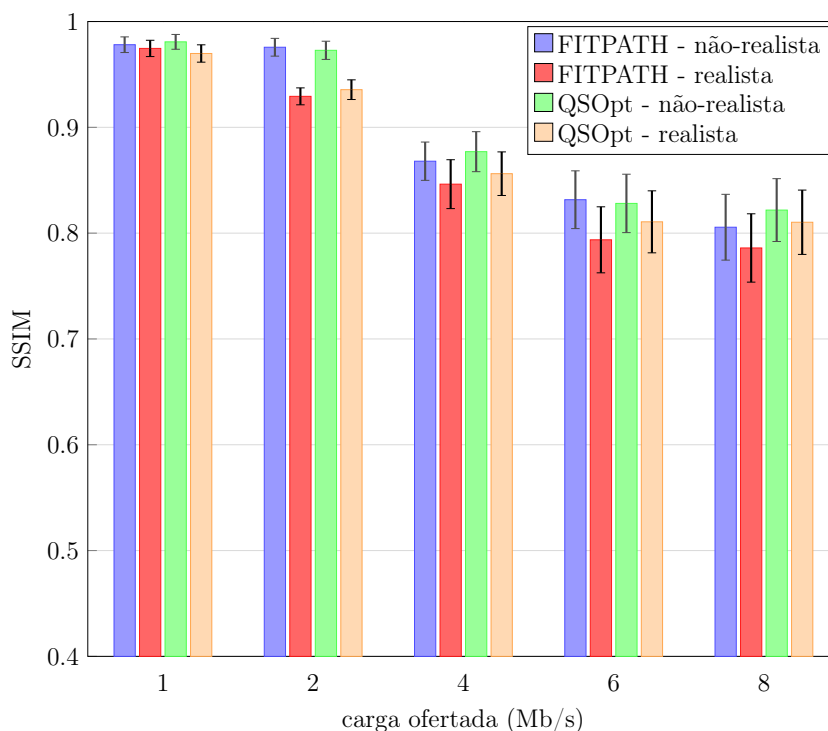


Figura 5.1: Impacto dos fluxos de controle da topologia na QoE do usuário em transmissões de cenas de vídeo de baixa mobilidade, com codificação LC.

### 5.2.1.2 Cenas de Vídeo de Alta Mobilidade

O gráfico da Figura 5.2 apresenta os resultados das simulações realizadas de modo similar às simulações apresentadas na Seção 5.2.1.1, porém com cenas de alta mobilidade do videoclipe *Coastguard*, em codificação MDC e cargas ofertadas de 1, 2 e 3 Mb/s. Quando as transmissões dos vídeos foram realizadas em concorrência com os fluxos de controle da topologia, a média de QoE dos vídeos recebidos pelas soluções de caminhos do FITPATH e do QSOpt atingiram, respectivamente, uma redução de 11,41% e 14,48% em comparação com os vídeo recebidos sem a concorrência dos fluxos de controle. Os resultados do SSIM médio, do desvio-padrão amostral, dos valores máximo e mínimo de SSIM e percentuais de impacto dos fluxos de controle no SSIM para cada carga ofertada, nas cenas de vídeo de alta mobilidade, encontram-se sumarizados no Apêndice A.2.

Destaca-se que, nos cenários realistas, as transmissões de cenas de vídeo de baixa mobilidade pelas soluções dos mecanismos de seleção de multicaminhos FITPATH e o QSOpt alcançaram, respectivamente, um SSIM médio de 0,87 e 0,88. Nas cenas de alta mobilidade, o FITPATH e o QSOpt obtiveram, respectivamente, um SSIM médio de 0,68 e 0,66. Assim, podemos concluir que os dois mecanismos de seleção de multicaminhos apresentaram soluções competitivas no que diz respeito à QoE do usuário, quando em



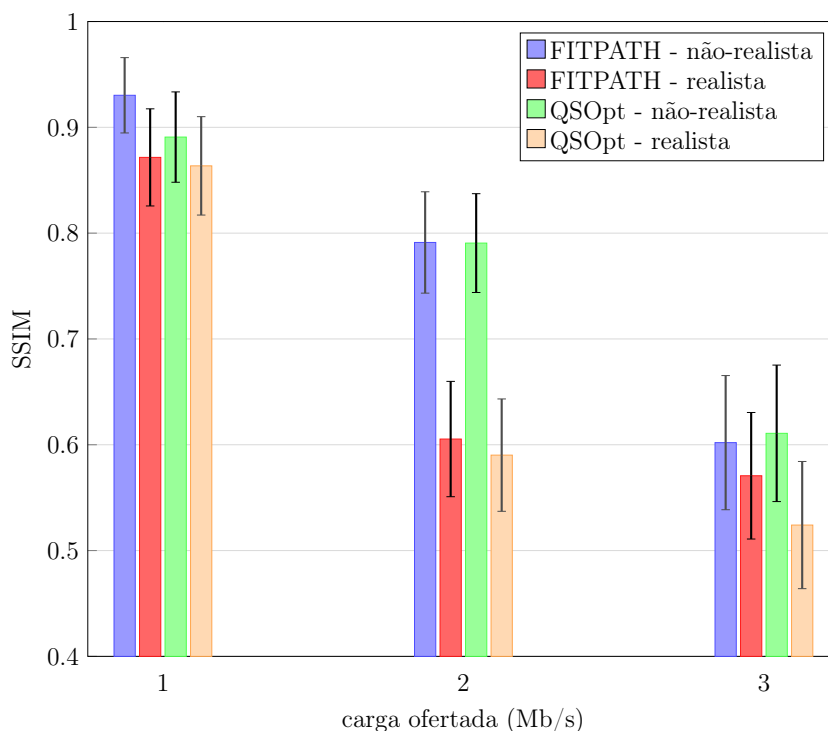


Figura 5.2: Impacto dos fluxos de controle da topologia na QoE do usuário em transmissões de cenas de vídeo de alta mobilidade, com codificação MDC.

cenários que mantêm o monitoramento da topologia através dos fluxos de controle.

Em resposta à questão Q1 (Como a presença dos fluxos de controle da topologia impacta na QoE do usuário?): A QoE do vídeo recebido diminui não somente com o aumento da carga ofertada e da mobilidade das cenas no vídeo, mas também com a concorrência dos fluxos de controle da topologia junto aos fluxos de vídeo, independentemente do mecanismo de seleção de caminhos utilizado - FITPATH ou QSOpt, haja vista que os dois mecanismos apresentaram soluções competitivas quando em cenários que mantêm o monitoramento da topologia. O impacto ocorre especialmente para vídeos com cenas de maiores níveis de mobilidade, uma vez que as rajadas geradas pelas mudanças de cenas resultam em maiores contenções e, conseqüentemente, mais colisões e atrasos nas filas.

### 5.2.2 Desempenho da QoE com Provisão das Taxas de Controle

Embora na Seção 5.2.1 tenha sido constatada a ocorrência de degradação na qualidade dos vídeos quando os fluxos de controle da topologia concorrem com os fluxos de vídeo no canal, uma possível abordagem para mitigar essa degradação é o provisionamento das taxas dos fluxos de controle da topologia, juntamente com as taxas dos fluxos de vídeo nas entradas dos mecanismos de seleção de multicaminhos, de forma que todos os

fluxos possam ser considerados no processo de escolha das rotas. Neste trabalho foram examinadas duas abordagens simples de provisionamento, conforme descritas a seguir:

- **Provisão A** - Incrementa as taxas de bits alvo dos fluxos de vídeo em 5%; e
- **Provisão B** - Estima uma taxa média dos fluxos de controle da topologia para todos os nós.

O experimento foi simulado de modo análogo ao descrito na Seção 5.2.1.1, porém com a solução de caminhos gerada apenas pelo mecanismo FITPATH nas estratégias de provisão A e B. Para o caso da provisão B, a taxa média de bits dos fluxos de controle da topologia foi estimada pela parcela da taxa dos pacotes HELLO somada à parcela da taxa dos pacotes TC. Foi observado que no instante de inicialização do mecanismo de seleção de caminhos, o cômputo das topologias experimentadas gerava aproximadamente 20 nós vizinhos para cada nó. Logo, como os pacotes TC são retransmitidos, a taxa total dos fluxos de controle foi estimada em 12,96 kb/s por nó, de acordo com a Equação 5.1, onde  $n\_neighbour$  é o número médio de vizinhos por nó, e os demais parâmetros encontram-se definidos na Tabela 1.

$$\text{Taxa (b/s)} = (\text{packet\_size} \times 8) \times \left[ \left( \frac{1}{\text{hello\_interval}} \right) + \left( \frac{n\_neighbour}{\text{tc\_interval}} \right) \right] \quad (5.1)$$

Os seguintes resultados da QoE do usuário são apresentados na Figura 5.3, conforme os cenários descritos a seguir:

- **Não-realista** - sem a transmissão dos fluxos de controle da topologia durante o tráfego dos vídeos;
- **Realista e sem provisão** - com a transmissão dos fluxos de controle da topologia durante o tráfego dos vídeos, porém sem a provisão das taxas dos mensagens de controle;
- **Realista e provisão A** - com a transmissão dos fluxos de controle da topologia e com a estratégia de provisão A; e
- **Realista e provisão B** - com a transmissão dos fluxos de controle da topologia e com a estratégia de provisão B.

No gráfico da Figura 5.3, observa-se que as duas estratégias de provisionamento analisadas apresentaram uma tendência de melhora da QoE do usuário em comparação ao

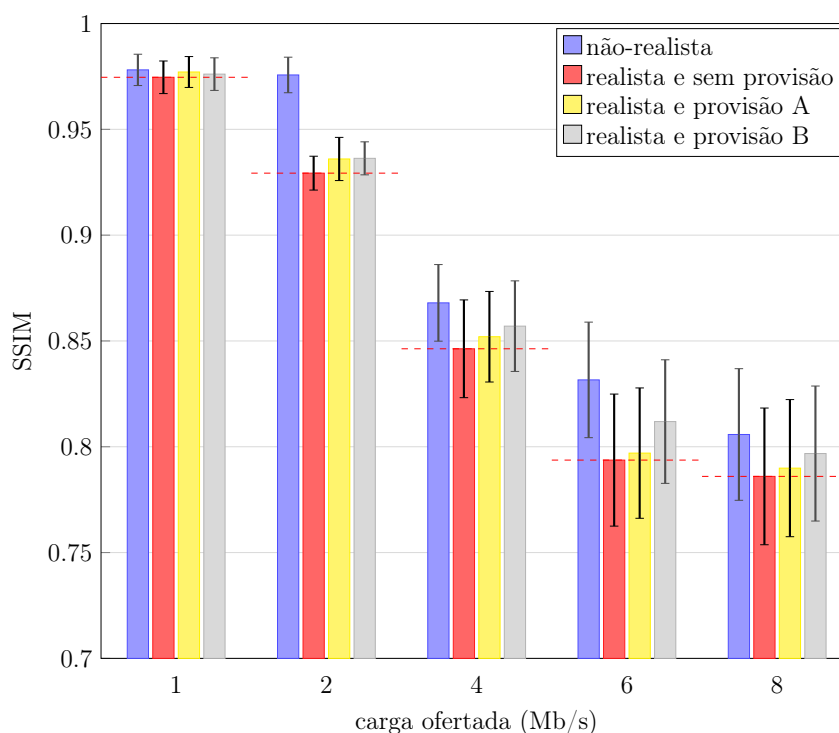


Figura 5.3: QoE do usuário em diversos níveis de carga ofertada e diferentes estratégias de provisão das taxas dos fluxos de controle da topologia ao FITPATH.

cenário realista que não fez uso de nenhuma técnica provisão das taxas de controle da topologia. Na provisão A, registrou-se uma recuperação média de 28,15% da QoE perdida devido ao impacto dos fluxos de controle da topologia, enquanto na provisão B, a recuperação média atingiu 42,08% da QoE perdida. Os resultados do SSIM médio, do desvio-padrão amostral, e dos valores máximo e mínimo de SSIM para cada carga ofertada nas duas estratégias de provisionamento (A e B) das taxas dos fluxos de controle encontram-se sumarizados no Apêndice B.1. Contudo, é importante destacar que, devido à sobreposição dos intervalos de confiança, é necessário que um número maior de amostras seja futuramente simulado, a fim de validar essa tendência de forma mais robusta.

Vale também ressaltar que nas cargas ofertadas maiores que 2 Mb/s, a provisão A sobrestima as taxas dos fluxos de controle da topologia, o que pode restringir o número de caminhos disponíveis na rede que atendam aos requisitos das taxas e, conseqüentemente, refletir em uma menor recuperação da QoE quando comparada à estratégia de provisão B.

Em resposta à questão Q2 (Uma vez que as soluções de caminhos dos mecanismos de seleção são baseadas nas taxas dos fluxos de vídeo, provisionar as taxas dos fluxos de controle da topologia poderia mitigar o impacto na QoE do usuário, uma vez que seriam selecionados, na medida do possível, caminhos compatíveis a todos os fluxos da

rede?): Duas estratégias de provisionamento, aqui apresentadas, poderão ser potenciais técnicas de melhoria da QoE do usuário, uma vez que demonstraram uma tendência de melhora da média da QoE do usuário em comparação ao cenário realista que não fez uso de nenhuma técnica de provisão das taxas de controle da topologia. Contudo, ainda é necessário validar essa tendência em trabalhos futuros.

O método de provisionamento também poderá ser futuramente aperfeiçoado, se forem provisionados os valores exatos das taxas dos fluxos de controle da topologia de cada nó. Esta melhoria poderá ser implementada de forma simples no simulador, através da Equação 5.1, com o uso do número exato de vizinhos computados para cada nó, na variável  $n\_neighbour$ , no momento prévio de chamada do mecanismo de seleção de multicaminhos.

### 5.2.3 Desempenho da QoE

Neste experimento, as transmissões de vídeo foram realizadas simultaneamente por duas fontes codificadas em 512 kb/s e alocadas nas coordenadas (0, 50) e (50, 0), conforme a topologia ilustrada na Figura 5.4.

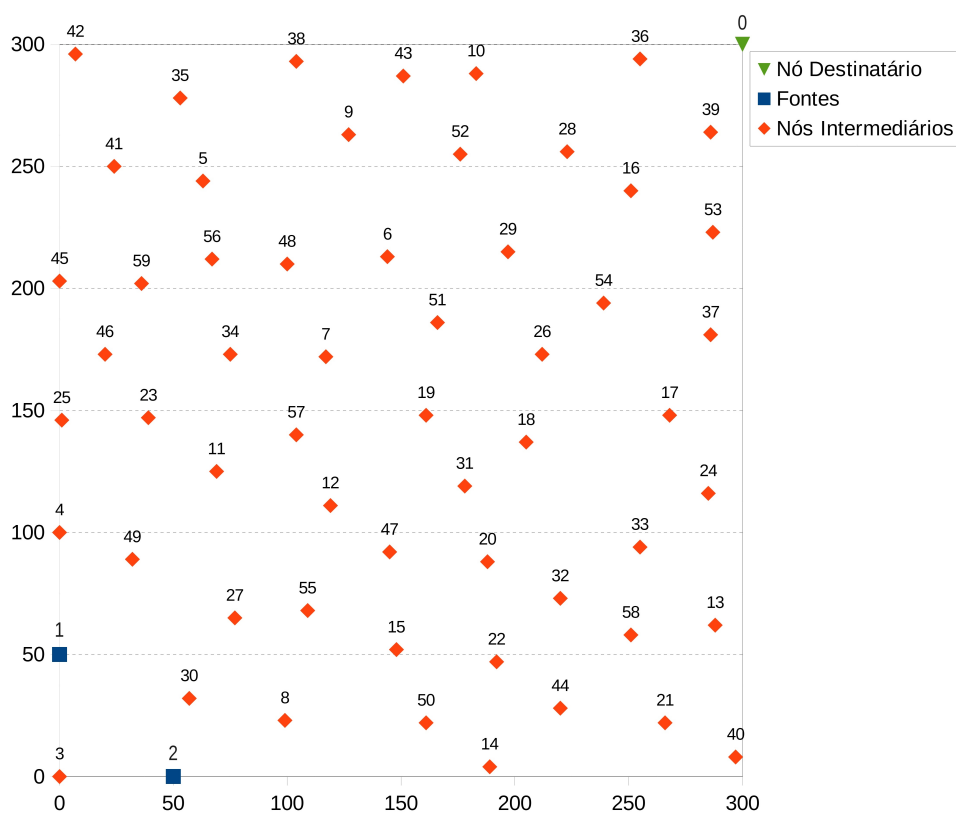


Figura 5.4: Configuração da topologia de 60 nós com transmissão de vídeo por duas fontes para um nó destinatário (estação de monitoramento).

Foram transmitidos 9000 quadros do videoclipe Bus, a partir de cada fonte, utilizando

a codificação LC e um *playout buffer* de 1000 ms. A topologia foi computada até o instante de 400 segundos e, em seguida, a transmissão foi realizada através da solução inicial de caminhos gerada pelo FITPATH. Após o tempo limite de execução da heurística de 60 segundos, a solução final foi obtida e, a partir do instante de 460 segundos, os fluxos de vídeo foram roteados pela nova solução.

Os caminhos gerados pelo FITPATH, para a topologia de 60 nós, são apresentados na Figura 5.5. Observa-se que ao contrário da solução final, que leva em conta a interferência entre os fluxos, a solução inicial não considera a interferência entre os fluxos, o que acarreta na escolha do mesmo caminho (o caminho de menor ETX) para todos os fluxos de vídeo transmitidos pela mesma fonte, ou seja, na fonte 1 o caminho  $p_{1_1}$  é igual ao caminho  $p_{1_2}$ , e na fonte 2 o caminho  $p_{2_1}$  é igual ao caminho  $p_{2_2}$ .



Figura 5.5: Soluções de caminhos geradas pelo FITPATH na topologia de 60 nós.

A Figura 5.6 ilustra a média da QoE do usuário dos quadros de vídeo transmitidos nos últimos 30 segundos, ao longo do tempo. No instante de 460 segundos, a QoE média do vídeo transmitido pela fonte 2, através da solução inicial, foi de 0,93, enquanto a QoE média do vídeo transmitido pela fonte 2, através da solução final, nos instantes de 490 e 520 segundos foi de 0,94, e portanto 1% maior que na solução anterior. A respeito da QoE média dos vídeos transmitidos pela fonte 1, esta permaneceu em 0,94 para ambas as soluções (inicial e final). No instante de 540 segundos ocorreu uma falha no nó 11, acarretando então em uma redução abrupta da QoE do vídeo transmitido pela fonte 1, em virtude do nó 11 ser um nó intermediário do caminho principal  $p_{1_1}$ , que transmite os quadros I na codificação LC. A falha foi identificada pela estação de vigilância no instante de 575 segundos, pois o limite de tempo para a atualização das informações da topologia, por meio das mensagens de TC, foi atingido. Com a mudança da topologia, o mecanismo de seleção de caminhos inicializa a busca de novas soluções e em seguida reestabelece o roteamento através de uma nova solução inicial.

As novas soluções, após a mudança da topologia em decorrência da falha do nó 11, são ilustradas na Figura 5.7. Nota-se que na solução inicial pós-falha, todos os caminhos gerados utilizam o enlace  $16 \rightarrow 0$ . No entanto, na solução final pós-falha, os fluxos de

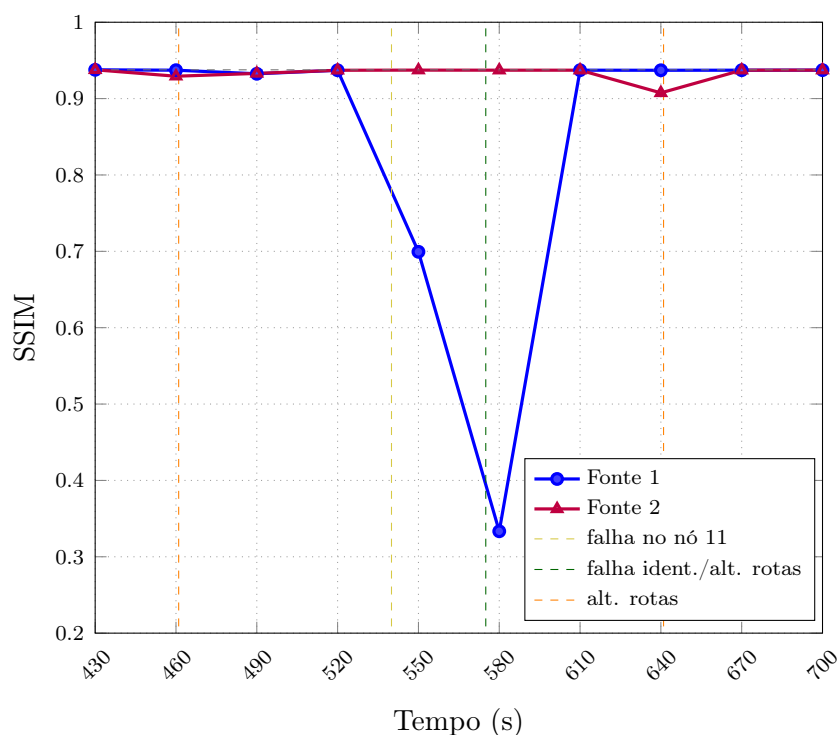


Figura 5.6: Desempenho da QoE média dos últimos 30 segundos.

vídeo da fonte 2 tornam-se totalmente disjuntos dos fluxos da fonte 1, uma vez que foram consideradas as interferências entre eles na otimização das rotas, e portanto afastada a possibilidade de gargalo no referido enlace.



Figura 5.7: Soluções de caminhos geradas pelo FITPATH após falha do nó 11.

Verifica-se na Figura 5.6 que a QoE média do vídeo transmitido pela fonte 2, por meio da solução inicial pós-falha, foi de 0,91 no instante de 640 segundos, enquanto a QoE média do vídeo transmitido pela fonte 2, através da solução final pós-falha, nos instantes de 670 e 700 segundos foi de 0,94, e portanto 3% maior que na solução anterior. A respeito da QoE média do vídeo transmitido pela fonte 1, esta permaneceu em 0,94 para ambas as soluções após a falha (inicial e final).

Em resposta à questão Q3 (A arquitetura proposta, com o monitor de topologia, é capaz de restaurar a QoE do usuário em caso de interrupção de um dos fluxos de vídeo, por motivo de falha em um dos nós intermediários? No contexto das soluções de

caminhos otimizadas, a arquitetura também é capaz de melhorar a QoE do usuário?): Foi apresentado um experimento, de uma forma didática, contendo outras situações de falhas foram simuladas, resultando em comportamentos semelhantes, o que demonstra a habilidade da arquitetura de se adaptar a eventuais falhas, e restabelecer a QoE do usuário a níveis competitivos aos anteriores às falhas. A arquitetura também se mostrou capaz de melhorar a QoE do usuário, ao longo do tempo, através da otimização das soluções de rotas pelo FITPATH.

### 5.2.4 Evolução do ETX no Roteamento Dinâmico

A Figura 5.8 ilustra a evolução das métricas de roteamento ETX dos caminhos de cada fluxo de vídeo (somatórios dos ETX de todos os enlaces de uma rota) registradas a cada 30 segundos no nó destinatário (estação de vigilância) do mesmo experimento apresentado na Seção 5.2.3.

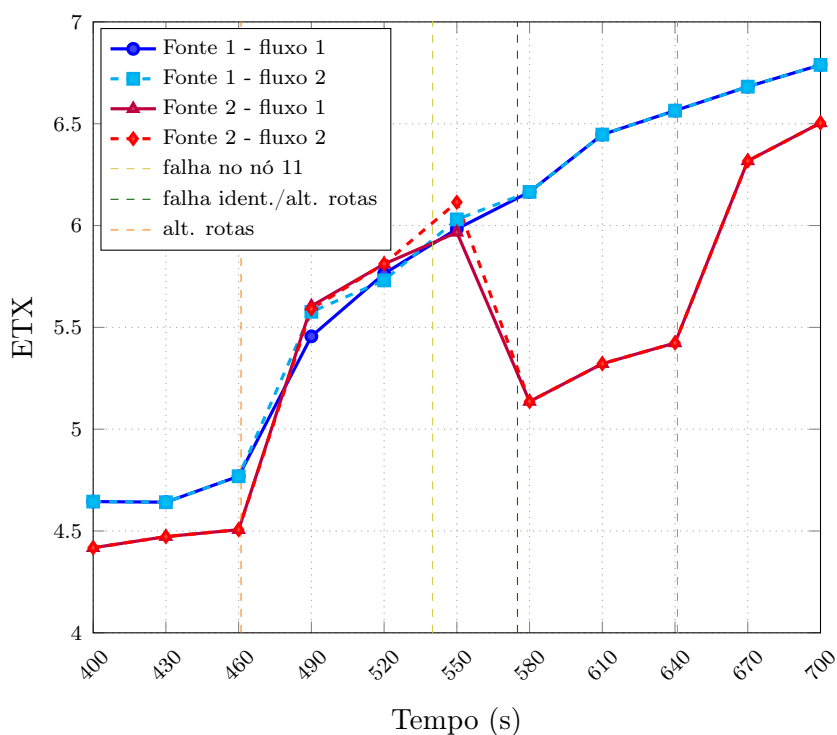


Figura 5.8: Evolução das métricas de roteamento registrada no nó destinatário a cada 30 segundos quando há tráfego dos vídeos.

É possível observar que apesar das curvas dos fluxos de vídeo da fonte 2 apresentarem aumento de seus coeficientes com as alterações das rotas da solução inicial para a solução final (tanto antes, quanto depois da falha), a tendência de aumento do ETX não se refletiu em degradação da QoE do vídeo recebido (vide gráfico da Figura 5.6). Ao contrário, a

QoE melhorou com as transmissões realizadas pela solução final de caminhos otimizados pelo FITPATH, visto que esta busca afastar possíveis gargalos na rede ao considerar a interferência entre os fluxos.

Apesar da falha do nó 11 ter impactado negativamente a QoE do vídeo transmitido pela fonte 1, desde o instante em que a falha se manifesta até a detecção da mesma na estação de vigilância, também é possível observar que após o reestabelecimento de novos caminhos, o aumento do ETX das novas rotas não causou prejuízos à QoE do vídeo recebido.

É importante ressaltar que quando a falha do nó 11 é identificada pelo nó destinatário, embora o ETX da solução inicial da fonte 2 tenha diminuído, não houve melhora da QoE do vídeo. Ao contrário, a QoE do vídeo sofreu uma redução de 3%, uma vez que esta solução não considera a interferência entre os fluxos de vídeo.

A Figura 5.9 ilustra a evolução do ETX ao longo do tempo nas mesmas rotas, porém apenas com os fluxos de controle. Como neste caso não há fluxos de vídeo nas rotas, o ETX tende a se manter estável em cada rota.

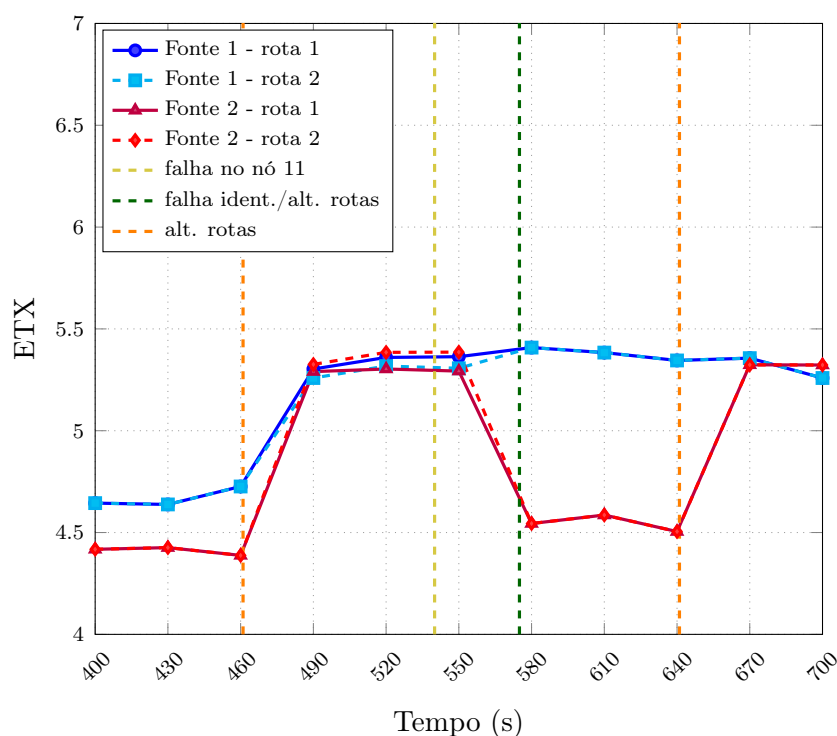


Figura 5.9: Evolução das métricas de roteamento registrada no nó destinatário a cada 30 segundos apenas com o tráfego de controle da topologia.

Nota-se que na Figura 5.8 há uma tendência de aumento do ETX em cada rota em virtude da transmissão dos fluxos de vídeo. Esse aumento pode impactar a determinação



de novos caminhos caso o mecanismo de seleção sofra uma rechamada. Isso ocorre porque os enlaces, inicialmente submetidos exclusivamente ao tráfego de controle, tinham um ETX com maior predileção pelo mecanismo de seleção. Entretanto, após o aumento do ETX, esses enlaces podem ser desconsiderados pelo mecanismo de seleção de caminhos durante a solução de novas rotas.

Em resposta à questão Q4 (Ao longo da transmissão dos fluxos de vídeo, há alguma correlação entre as métricas de roteamento (baseada na qualidade dos enlaces) obtidas durante a transmissão dos fluxos de vídeo e a QoE do usuário?): Não foi evidenciada correlação entre a métrica de roteamento ETX (dos caminhos) e a métrica SSIM de QoE do usuário nas transmissões por multicaminhos. Isto ocorre porque nas soluções iniciais, a concorrência de diversos fluxos em enlaces não-disjuntos - *a priori* de boa qualidade, e portanto baixo ETX, pode resultar na degradação da QoE do usuário, em virtude de possíveis gargalos gerados na rede pela interferência entre esses fluxos. Em contrapartida, o aumento do ETX nas transmissões pelas soluções otimizadas não necessariamente reflete em piora da QoE do usuário, pois essas soluções buscam distribuir os fluxos de forma a obter melhores vazões, e conseqüentemente melhores resultados de QoE do usuário.

## 6 Conclusão

Este trabalho propôs uma arquitetura de rede sem fio, descentralizada, que viabiliza a execução de testes dos mecanismos de seleção de multicaminhos para vídeos em ambiente IoT próximo do realista, que requeiram a visão global da topologia para estabelecimento das melhores rotas. Essa arquitetura se mostrou capaz de oferecer suporte dinâmico ao roteamento, diante de mudanças significativas da topologia, tais como eventuais falhas, e restabelecer a QoE do usuário a níveis competitivos aos anteriores às falhas. A arquitetura também se mostrou capaz de melhorar a QoE do usuário, ao longo do tempo, através da otimização das soluções de rotas pelo FITPATH.

Foi demonstrado que a qualidade do vídeo recebido diminui não somente com o aumento da carga ofertada e da mobilidade das cenas no vídeo, mas também com a concorrência dos fluxos de controle da topologia junto aos fluxos de vídeo, independentemente de qual mecanismo de seleção de caminhos tenha sido utilizado - FITPATH ou QSOpt, haja vista que os dois mecanismos apresentaram soluções competitivas quando em cenários que mantêm o monitoramento da topologia. O impacto ocorre especialmente para vídeos com cenas de maiores níveis de mobilidade, uma vez que as rajadas geradas pelas mudanças de cenas resultam em maiores contenções e, conseqüentemente, mais colisões e atrasos nas filas.

Esta pesquisa também contribuiu com a análise de duas abordagens de provisão das taxas dos fluxos de controle da topologia, juntamente com as taxas dos fluxos de vídeo nas entradas dos mecanismos de seleção de multicaminhos, de forma que todos os fluxos pudessem ser considerados no processo de escolha das rotas. As duas estratégias de provisionamento demonstraram uma tendência de melhora da média da QoE do usuário em comparação ao cenário que não faz uso de nenhuma técnica de provisão. Contudo, foi verificada a necessidade de um maior número de amostras simuladas no futuro, a fim de reduzir o intervalo de confiança e validar essa tendência de forma mais robusta.

Por fim, não foi evidenciada correlação entre a métrica de roteamento ETX e a métrica

SSIM de QoE do usuário nas transmissões por multicaminhos. Isto ocorre porque nas soluções iniciais, a concorrência de diversos fluxos em enlaces não-disjuntos - *a priori* de boa qualidade, e portanto baixo ETX, pode resultar na degradação da QoE do usuário, em virtude de possíveis gargalos gerados na rede pela interferência entre esses fluxos. Em contrapartida, o aumento do ETX nas transmissões pelas soluções otimizadas não necessariamente reflete em piora da QoE do usuário, pois essas soluções buscam distribuir os fluxos de forma a obter melhores vazões, e conseqüentemente melhores resultados de QoE do usuário.

## 6.1 Trabalhos Futuros

Estudos futuros a serem desenvolvidos incluem:

- A execução de testes exaustivos dos mecanismos de controle FITPATH e QSOpt em cenários de roteamento dinâmico, tais como em condições de falha de nós, inclusão de novos nós na rede, e em serviços que demandem a alteração da resolução do vídeo ou do status de habilitação das câmeras;
- A implementação de uma arquitetura com o plano de controle centralizado, conforme a proposta sugerida na Seção 4.1;
- A avaliação da performance da arquitetura implementada com outras configurações de parâmetros do protocolo de monitoramento da topologia, tais como o tamanho da janela deslizante, o limite de pacotes HELLO e TC consecutivamente perdidos, e o intervalo de transmissão dos pacotes HELLO e TC;
- A simulação de um maior número de instâncias de topologia para validar o potencial de melhora da QoE do usuário quando da utilização das estratégias de provisionamento das taxas dos fluxos de controle da topologia; e
- A implementação do provisionamento dos valores exatos das taxas dos fluxos de controle da topologia de cada nó, conforme sugerido na Seção 5.2.2.

# REFERÊNCIAS

- [1] Mohammed Abazeed, Norshiela Faisal e Adel Ali. “Cross-layer multipath routing scheme for wireless multimedia sensor network”. Em: *Wireless Networks* 25 (2019), pp. 4887–4901.
- [2] N Achyutha Prasad et al. “AN EPHEMERAL INVESTIGATION ON FOREST FIRE, WILDLIFE PRESERVATION, AND TRIBAL PREPAREDNESS USING DEEP LEARNING AND IOT”. Em: *International Journal For Technological Research in Engineering* 11 (2023), pp. 35–40.
- [3] Samira Afzal et al. “A holistic survey of multipath wireless video streaming”. Em: *Journal of Network and Computer Applications* 212 (2023), p. 103581.
- [4] Adel A Ahmed. “A real-time routing protocol with adaptive traffic shaping for multimedia streaming over next-generation of Wireless Multimedia Sensor Networks”. Em: *Pervasive and Mobile Computing* 40 (2017), pp. 495–511.
- [5] Sheeraz A Alvi et al. “Internet of multimedia things: Vision and challenges”. Em: *Ad Hoc Networks* 33 (2015), pp. 87–111.
- [6] Alcardo Alex Barakabitze et al. “QualitySDN: Improving video quality using MPTCP and segment routing in SDN/NFV”. Em: *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE. 2018, pp. 182–186.
- [7] Sabina Baraković, Lea Skorin-Kapov et al. “Survey and challenges of QoE management issues in wireless networks”. Em: *Journal of Computer Networks and Communications* 2013 (2013).
- [8] Fabiano Bhering. “A Cross-Layer Multipath Selection Approach for IoT Video Transmission”. Tese de dout. Instituto de Computação - UFF, 2022.
- [9] Fabiano Bhering et al. “Efficient Multipath Selection for IoT Video Transmission”. Em: *2022 IEEE 11th International Conference on Cloud Networking (CloudNet)*. IEEE. 2022, pp. 73–78.
- [10] Fabiano Bhering et al. “Network performance estimator with applications to route selection for IoT multimedia applications”. Em: *Simulation* 100.1 (2024), pp. 23–37.

- 
- [11] Fabiano Bhering et al. “Wireless multipath video transmission: when IoT video applications meet networking—a survey”. Em: *Multimedia Systems* 28.3 (2022), pp. 831–850.
- [12] Fabiano Pereira Bhering et al. “Multipath Selection Mechanism for Wireless Video-Surveillance Systems.” Em: *LANOMS*. 2019.
- [13] Jill M Boyce et al. “Overview of SHVC: Scalable extensions of the high efficiency video coding standard”. Em: *IEEE Transactions on Circuits and Systems for Video Technology* 26.1 (2015), pp. 20–34.
- [14] Benjamin Bross et al. “Overview of the versatile video coding (VVC) standard and its applications”. Em: *IEEE Transactions on Circuits and Systems for Video Technology* 31.10 (2021), pp. 3736–3764.
- [15] Jiling Cai et al. “SVC-based and Flow Assignment Strategy for Multi-path Concurrent Video Transmission with Low-Latency”. Em: *2022 3rd Asia Conference on Computers and Communications (ACCC)*. IEEE. 2022, pp. 109–115.
- [16] Lelio Campanile et al. “Computer network simulation with ns-3: A systematic literature review”. Em: *Electronics* 9.2 (2020), p. 272.
- [17] Jacob Chakareski, Sangeun Han e Bernd Girod. “Layered coding vs. multiple descriptions for video streaming over multiple paths”. Em: *Proceedings of the eleventh ACM international conference on Multimedia*. 2003, pp. 422–431.
- [18] Chang Wen Chen. “Internet of video things: Next-generation IoT with visual sensors”. Em: *IEEE Internet of Things Journal* 7.8 (2020), pp. 6676–6685.
- [19] Raga Bharani Chenna, Chetna Singhal e Swades De. “MDTC video transmission using channel aware multipath routing in wireless mesh networks”. Em: *Proceedings of the 19th International Conference on Distributed Computing and Networking*. 2018, pp. 1–9.
- [20] Thomas Clausen e Philippe Jacquet. *Optimized link state routing protocol (OLSR)*. Rel. técn. 2003.
- [21] Pedro Correia et al. “Optimal priority mdc video streaming for networks with path diversity”. Em: *2012 International Conference on Telecommunications and Multimedia (TEMU)*. IEEE. 2012, pp. 54–59.
- [22] Huali Cui et al. “Video-aware multipath network coding over multi-hop wireless networks”. Em: *2019 International Conference on Information Networking (ICOIN)*. IEEE. 2019, pp. 171–175.

- [23] Nhat-Quang Dao et al. “Management of video surveillance for smart cities”. Em: *Handbook of Smart Cities: Software Services and Cyber Infrastructure* (2018), pp. 285–310.
- [24] M. Sheik Dawood et al. “Performance analysis of efficient video transmission using EvalSVC, EvalVid-NT, EvalVid”. Em: *Materials Today: Proceedings* 46 (2021), pp. 3848–3850.
- [25] Douglas De Couto et al. “A high-throughput path metric for multi-hop wireless routing”. Em: *Proceedings of the 9th annual international conference on Mobile computing and networking*. 2003, pp. 134–146.
- [26] Samina Ehsan e Bechir Hamdaoui. “A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks”. Em: *IEEE communications surveys & tutorials* 14.2 (2011), pp. 265–278.
- [27] Anis Elgabli, Ke Liu e Vaneet Aggarwal. “Optimized preference-aware multi-path video streaming with scalable video coding”. Em: *IEEE Transactions on Mobile Computing* 19.1 (2018), pp. 159–172.
- [28] *FFmpeg*. <https://ffmpeg.org/>. Acesso em: Fevereiro, 2024.
- [29] Shahram Ghahremani e Mohammad Ghanbari. “Error resilient video transmission in ad hoc networks using layered and multiple description coding”. Em: *Multimedia Tools and Applications* 76 (2017), pp. 9033–9049.
- [30] Mohammed Ghanbari. “Two-layer coding of video signals for VBR networks”. Em: *IEEE Journal on Selected Areas in Communications* 7.5 (1989), pp. 771–781.
- [31] Majd Ghareeb e Cesar Viho. “Hybrid qoe assessment is well-suited for multiple description coding video streaming in overlay networks”. Em: *2010 8th Annual Communication Networks and Services Research Conference*. IEEE. 2010, pp. 327–333.
- [32] Vivek K Goyal. “Multiple description coding: Compression meets the network”. Em: *IEEE Signal processing magazine* 18.5 (2001), pp. 74–93.
- [33] Mohammed Zaki Hasan, Hussain Al-Rizzo e Fadi Al-Turjman. “A survey on multi-path routing protocols for QoS assurances in real-time wireless multimedia sensor networks”. Em: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1424–1456.
- [34] M Shamim Hossain. “Patient status monitoring for smart home healthcare”. Em: *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE. 2016, pp. 1–6.

- [35] M Shamim Hossain et al. “QoS-oriented multimedia transmission using multipath routing”. Em: *Future Generation Computer Systems* 99 (2019), pp. 226–234.
- [36] Yongkai Huo et al. “A tutorial and review on inter-layer FEC coded layered video streaming”. Em: *IEEE Communications Surveys & Tutorials* 17.2 (2015), pp. 1166–1207.
- [37] Amulya Karaadi, Lingfen Sun e Is-Haka Mkwawa. “Multimedia communications in internet of things QoT or QoE?” Em: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2017, pp. 23–29.
- [38] Koffka Khan e Wayne Goodridge. “Multi-criterion decision making and adaptation for multi-path video streaming in WSNS”. Em: *International Journal of Advanced Networking and Applications* 9.2 (2017), pp. 3376–3381.
- [39] Jirka Klaue, Berthold Rathke e Adam Wolisz. “Evalvid—a framework for video transmission and quality evaluation”. Em: *Computer Performance Evaluation. Modelling Techniques and Tools: 13th International Conference, TOOLS 2003, Urbana, IL, USA, September 2-5, 2003. Proceedings 13*. Springer. 2003, pp. 255–272.
- [40] Purushottam Kulkarni et al. “SensEye: a multi-tier camera sensor network”. Em: *Proceedings of the 13th annual ACM international conference on Multimedia*. 2005, pp. 229–238.
- [41] Krishan Kumar. “Text query based summarized event searching interface system using deep learning over cloud”. Em: *Multimedia Tools and Applications* 80.7 (2021), pp. 11079–11094.
- [42] Krishan Kumar e Deepti D Shrimankar. “F-DES: Fast and deep event summarization”. Em: *IEEE Transactions on Multimedia* 20.2 (2017), pp. 323–334.
- [43] Krishan Kumar, Deepti D Shrimankar e Navjot Singh. “Equal partition based clustering approach for event summarization in videos”. Em: *2016 12th International conference on signal-image technology & internet-based systems (SITIS)*. IEEE. 2016, pp. 119–126.
- [44] Didier Le Gall. “MPEG: A video compression standard for multimedia applications”. Em: *Communications of the ACM* 34.4 (1991), pp. 46–58.

- [45] Yen-Chi Lee et al. “Performance comparisons of layered and multiple description coded video streaming over error-prone networks”. Em: *IEEE International Conference on Communications, 2003. ICC’03*. Vol. 1. IEEE. 2003, pp. 35–39.
- [46] Zhi Liu et al. “Multiple description coding and recovery of free viewpoint video for wireless multi-path streaming”. Em: *IEEE Journal of Selected Topics in Signal Processing* 9.1 (2014), pp. 151–164.
- [47] Helena Ramalhinho Lourenço, Olivier C Martin e Thomas Stützle. “Iterated local search: Framework and applications”. Em: *Handbook of metaheuristics* (2019), pp. 129–168.
- [48] Zhao-xia Lu et al. “Application of AI and IoT in clinical medicine: summary and challenges”. Em: *Current medical science* 41 (2021), pp. 1134–1150.
- [49] Priyanka Mishra e Ghanshyam Singh. “Internet of medical things healthcare for sustainable smart cities: current status and future prospects”. Em: *Applied Sciences* 13.15 (2023), p. 8869.
- [50] Yanjinkham Myagmar-Ochir e Wooseong Kim. “A Survey of Video Surveillance Systems in Smart City”. Em: *Electronics* 12.17 (2023), p. 3567.
- [51] Ali Nauman et al. “Multimedia Internet of Things: A comprehensive survey”. Em: *Ieee Access* 8 (2020), pp. 8202–8250.
- [52] Peter Orosz et al. “A case study on correlating video QoS and QoE”. Em: *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE. 2014, pp. 1–5.
- [53] Tie Qiu et al. “Edge computing in industrial internet of things: Architecture, advances and challenges”. Em: *IEEE Communications Surveys & Tutorials* 22.4 (2020), pp. 2462–2488.
- [54] Pham Tran Anh Quang et al. “Video streaming over ad hoc networks: A qoe-based optimal routing solution”. Em: *IEEE Transactions on Vehicular Technology* 66.2 (2016), pp. 1533–1546.
- [55] Albert Rego et al. “An intelligent system for video surveillance in IoT environments”. Em: *IEEE Access* 6 (2018), pp. 31580–31598.
- [56] Abderahman Rejeb et al. “The big picture on the internet of things and the smart city: a review of what we know and what we need to know”. Em: *Internet of Things* 19 (2022), p. 100565.
- [57] Mateus Santos et al. “Multipath source routing strategies for video transmission in ad hoc wireless networks”. Em: *Wireless Networks* 21 (2015), pp. 859–869.



- [58] Patrick Seeling e Martin Reisslein. “Video transport evaluation with H. 264 video traces”. Em: *IEEE Communications Surveys & Tutorials* 14.4 (2011), pp. 1142–1165.
- [59] Yutian Sha et al. “Smart City Public Safety Intelligent Early Warning and Detection”. Em: *Scientific Programming* 2022 (2022).
- [60] E Shalini e TVP Sundararajan. “Improved video transmission over mobile ad-hoc networks using AOMDV and MDC with path diversity”. Em: *International Journal of Computer Applications* 62.17 (2013).
- [61] Mirko Stoffers e George Riley. “Comparing the ns-3 propagation models”. Em: *2012 IEEE 20th international symposium on modeling, analysis and simulation of computer and telecommunication systems*. IEEE. 2012, pp. 61–67.
- [62] Ramona Trestian, Ioan-Sorin Comsa e Mehmet Fatih Tuysuz. “Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?” Em: *IEEE Communications Surveys & Tutorials* 20.2 (2018), pp. 945–977.
- [63] Muhammad Usman et al. “A joint framework for QoS and QoE for video transmission over wireless multimedia sensor networks”. Em: *IEEE Transactions on Mobile Computing* 17.4 (2017), pp. 746–759.
- [64] Zhao Wang, YiFeng Yang e Hao Zhang. “Error Concealment for Scalable Video Coding Based on Deformable Convolution Network”. Em: *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2023, pp. 2455–2459.
- [65] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. Em: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.
- [66] Wei Wei e Avidesh Zakhor. “Interference aware multipath selection for video streaming in wireless ad hoc networks”. Em: *IEEE Transactions on Circuits and Systems for Video Technology* 19.2 (2008), pp. 165–178.
- [67] Jiyan Wu et al. “Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks”. Em: *IEEE Transactions on Mobile Computing* 15.9 (2015), pp. 2345–2361.
- [68] Wenfeng Xia et al. “A survey on software-defined networking”. Em: *IEEE Communications Surveys & Tutorials* 17.1 (2014), pp. 27–51.
- [69] Chaofan Yang et al. “Enhancing industrial video surveillance over wireless mesh networks”. Em: *2016 25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2016, pp. 1–9.

- 
- [70] Jin Y Yen. “An algorithm for finding shortest routes from all source nodes to a given destination in general networks”. Em: *Quarterly of applied mathematics* 27.4 (1970), pp. 526–530.

# APÊNDICE A

Este apêndice sumariza os resultados do SSIM médio, do desvio-padrão amostral, e dos valores máximo e mínimo de SSIM para as transmissões de cenas de vídeo de baixa e alta mobilidade, por caminhos selecionados pelo FITPATH e pelo QSOpt, em cenários não-realista (sem os fluxos de controle da topologia) e realista (com os fluxos de controle da topologia), em diferentes níveis de carga ofertada na rede, assim como apresenta os percentuais de impacto dos fluxos de controle da topologia no SSIM, conforme experimento descrito na Seção 5.2.1.

## A.1 SSIM de Cenas de Vídeo de Baixa Mobilidade

Tabela 3: SSIM nas transmissões de cenas de vídeo de baixa mobilidade por caminhos selecionados pelo FITPATH.

Carga Ofertada (Mb/s)	Sem Fluxos de CTRL				Com Fluxos de CTRL				Impacto no SSIM
	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	
1	0,98	0,04	1,00	0,85	0,97	0,04	1,00	0,85	-1,02%
2	0,98	0,05	1,00	0,84	0,93	0,04	1,00	0,77	-5,10%
4	0,87	0,10	0,99	0,52	0,85	0,13	0,95	0,52	-2,30%
6	0,83	0,15	0,93	0,52	0,79	0,17	0,93	0,52	-4,82%
8	0,81	0,17	0,93	0,52	0,79	0,18	0,93	0,52	-2,47%
SSIM Médio:					0,87	Impacto SSIM méd.:			-3,14%

Tabela 4: SSIM nas transmissões de cenas de vídeo de baixa mobilidade por caminhos selecionados pelo QSOpt.

Carga Ofertada (Mb/s)	Sem Fluxos de CTRL				Com Fluxos de CTRL				Impacto no SSIM
	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	
1	0,98	0,04	1,00	0,82	0,97	0,05	1,00	0,85	-1,02%
2	0,97	0,05	1,00	0,77	0,94	0,05	1,00	0,70	-3,09%
4	0,88	0,11	1,00	0,52	0,86	0,12	0,99	0,52	-2,27%
6	0,83	0,15	0,93	0,52	0,81	0,16	0,93	0,52	-2,41%
8	0,82	0,17	0,93	0,52	0,81	0,17	0,93	0,51	-1,22%
SSIM Médio:					0,88	Impacto SSIM méd.:			-2,00%

## A.2 SSIM de Cenas de Vídeo de Alta Mobilidade

Tabela 5: SSIM nas transmissões de cenas de vídeo de alta mobilidade por caminhos selecionados pelo FITPATH.

Carga Ofertada (Mb/s)	Sem Fluxos de CTRL				Com Fluxos de CTRL				Impacto no SSIM
	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	
1	0,93	0,20	1,00	0,35	0,87	0,26	1,00	0,36	-6,45%
2	0,79	0,27	1,00	0,20	0,61	0,30	1,00	0,20	-22,78%
3	0,60	0,35	1,00	0,06	0,57	0,33	1,00	0,06	-5,00%
SSIM Médio:					0,68	Impacto SSIM méd.:			-11,41%

Tabela 6: SSIM nas transmissões de cenas de vídeo de alta mobilidade por caminhos selecionados pelo QSOpt.

Carga Ofertada (Mb/s)	Sem Fluxos de CTRL				Com Fluxos de CTRL				Impacto no SSIM
	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	
1	0,89	0,24	1,00	0,36	0,86	0,26	1,00	0,36	-3,37%
2	0,79	0,26	1,00	0,07	0,59	0,30	1,00	0,06	-25,32%
3	0,61	0,36	1,00	0,06	0,52	0,34	1,00	0,06	-14,75%
SSIM Médio:					0,66	Impacto SSIM méd.:			-14,48%

## APÊNDICE B

Este apêndice sumariza os resultados de SSIM médio, de desvio-padrão amostral e dos valores máximo e mínimo de SSIM para as transmissões de cenas de vídeo de baixa mobilidade, por caminhos selecionados pelo FITPATH, em diferentes níveis de carga ofertada na rede, utilizando estratégias distintas de provisionamento das taxas dos fluxos de controle, conforme experimento descrito na Seção 5.2.2.

### B.1 SSIM em Estratégias de Provisão das Taxas de Controle

Tabela 7: SSIM em estratégias distintas de provisão das taxas dos fluxos de controle da topologia.

Carga Ofertada (Mb/s)	Provisão A				Provisão B			
	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.	SSIM médio	$\sigma^2$	SSIM máx.	SSIM mín.
1	0.98	0.04	1.00	0.85	0.98	0.04	1.00	0.82
2	0.94	0.05	1.00	0.54	0.94	0.04	1.00	0.77
4	0.85	0.12	0.96	0.52	0.86	0.12	0.97	0.52
6	0.80	0.17	0.93	0.45	0.81	0.16	0.93	0.52
8	0.79	0.18	0.93	0.52	0.80	0.18	0.93	0.52