UNIVERSIDADE FEDERAL FLUMINENSE

FABIANO PEREIRA BHERING

A CROSS-LAYER MULTIPATH SELECTION APPROACH FOR IoT VIDEO TRANSMISSION

NITERÓI

2022

UNIVERSIDADE FEDERAL FLUMINENSE

FABIANO PEREIRA BHERING

A CROSS-LAYER MULTIPATH SELECTION APPROACH FOR IoT VIDEO TRANSMISSION

Thesis presented to the Graduate Program in Computing at the Fluminense Federal University as a requirement for obtaining a Doctoral Degree in Computing. Concentration area: Computer Science.

Advisor:
Célio Vinicius Neves de Albuquerque

Co-advisor:
Diego Gimenez Passos

NITERÓI

2022

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

B575c Bhering, Fabiano Pereira
A CROSS-LAYER MULTIPATH SELECTION APPROACH FOR IoT VIDEO TRANSMISSION / Fabiano Pereira Bhering. - 2022.
120 f.: il.

Orientador: Célio Vinicius Neves De Albuquerque.
Coorientador: Diego Gimenez Passos.
Tese (doutorado)-Universidade Federal Fluminense, Instituto de Computação, Niterói, 2022.

1. Roteamento por Multicaminhos. 2. Transmissão de Vídeos sem Fio. 3. Mecanismo de Seleção Multicaminhos. 4. Aplicações de Vídeo IoT. 5. Produção intelectual. I. Neves De Albuquerque, Célio Vinicius, orientador. II. Gimenez Passos, Diego, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

FABIANO PEREIRA BHERING

A CROSS-LAYER MULTIPATH SELECTION APPROACH FOR IoT VIDEO TRANSMISSION

Thesis presented to the Graduate Program in Computing at the Fluminense Federal University as a requirement for obtaining a Doctoral Degree in Computing. Concentration area: Computer Science.

Approved in October 2022.

EXAMINATION BOARD

Prof. Célio Vinicius Neves de Albuquerque – Advisor, UFF

Prof. Diego Gimenez Passos – Co-advisor, UFF

Profa. Katia Obraczka, University of California, Santa Cruz

Prof. Débora Christina Muchaluat Saade, UFF

Prof. Igor Monteiro Moraes, UFF

Prof. Luís Henrique Maciel Kosmalski Costa, UFRJ

Niterói

2022

Understand well as I may, my comprehension can only be an infinitesimal fraction of all I want to understand. (Ada Lovelace)

Acknowledgements

This thesis would not be possible without assistance, support and guidance of many individuals and institutions. First, I would like to thank God and my family, especially my wife, Liana Lima, who have continuously encouraged and enthusiastically assisted me in any way. All the professors at IC-UFF, particularly my advisors Célio Albuquerque, Diego Passo, and Katia Obraczka from UCSC, for their consistent support and guidance during this research. I also would like to thank the colleagues, employees, and professors of CEFET-MG, Laboratório MidiaCom - IC-UFF, and i-NRG - UCSC for the support, the infrastructure, the companionship, and the advice. Finally, I would like to acknowledge CEFET-MG for supporting me during the whole study period and CAPES for providing the opportunity for the sandwich period at UCSC.

Resumo

Os avanços nas tecnologias das câmeras de vídeo e das redes de comunicação sem fio possibilitaram uma variedade de aplicações de vídeo na Internet. Em aplicações de vídeo para Internet das Coisas (*Internet of Things*- IoT), como em monitoramento de trânsito e transporte público, o vídeo é geralmente transmitido por uma rede sem fio de múltiplos saltos e possui requisitos rigorosos de Qualidade de Serviço (*Quality of Service* - QoS). No entanto, atender estes requisitos apresenta desafios significativos para a rede subjacente e tem atraído a atenção da comunidade de pesquisa em redes. Em particular, a transmissão de vídeo multicaminho sem fio, onde cada fluxo gerado é transmitido por um caminho selecionado, tem sido proposta como uma estratégia eficaz para fornecer largura de banda adequada e garantias de atraso para aplicações de vídeo. Esta tese tem como objetivo apresentar uma nova abordagem de otimização iterativa baseada em heurística que estima as condições da rede em tempo real, considerando os diferentes requisitos de taxa de transmissão dos fluxos de vídeo da aplicação. Para isso, realizamos uma revisão detalhada do atual estado da arte em transmissão de vídeo multicaminhos com foco em aplicações IoT, introduzimos uma taxonomia para classificar as abordagens existentes conforme seus mecanismos específicos de aplicação e das técnicas de rede, e apresentamos um *framework* para seleção de multicaminhos baseado em abordagem *cross-layer* para transmissão de vídeo em IoT, no qual são utilizadas as informações dos requisitos dos fluxo de vídeo da camada de aplicação para identificar melhores caminhos para esses fluxos. A principal contribuição é o FITPATH, um mecanismo de seleção de caminhos eficiente e adaptável para roteamento multicaminhos que pode acomodar cenários onde múltiplas fontes de vídeo podem transmitir simultaneamente fluxos de vídeo heterogêneos. Uma das contribuições adicionais é o desenvolvimento do *Multimedia-Aware Performance Estimator* (MAPE), o primeiro estimador da performance da rede baseado em simulação determinística que fornece estimativas de vazão, perda de pacotes e atrasos, considerando interferência entre fluxos e fluxos com taxa de transmissão específica, típico dos tráfegos de vídeo. O MAPE é capaz de fornecer, em tempo quase real, estimativas de desempenho da rede que podem ser usadas pelo FITPATH para informar a seleção de multicaminhos na transmissão de vídeo em IoT. Nossos resultados mostram que o MAPE atinge alta precisão em uma fração do tempo de execução quando comparado aos simuladores de rede estocásticos, e o FITPATH supera vários mecanismos de seleção de caminho existentes, tanto em termos de Qualidade de Experiência (*Quality of Experience* - QoE) do usuário quanto de desempenho da rede. Por fim, demonstramos como o FITPATH pode ser utilizado na prática para gerar rapidamente soluções com melhor qualidade de transmissão de vídeo.

Palavras-chave: Roteamento por Multicaminhos, Transmissão de Vídeos sem Fio, Estimador de Performance, Mecanismo de Seleção Multicaminhos, Aplicações de Vídeo IoT.

Abstract

Advances in video cameras and wireless communication technologies have enabled a variety of video applications over the Internet. In Internet of Things (IoT) video applications, such as public transportation and traffic monitoring, video is often transmitted over a multi-hop wireless network and has stringent Quality of Service (QoS) requirements. However, meeting these requirements poses significant challenges to the underlying network and has attracted attention from the networking research community. In particular, wireless multipath video transmission, where each generated flow is transmitted through a selected path, has been proposed as an effective strategy to provide adequate bandwidth and delay guarantees for video applications. This thesis aims to present a novel heuristic-based iterative optimization approach that estimates the conditions of the underlying network in real time while accounting for the different bitrate requirements of the application video flows. For that, we carry out a thorough review of the current state-of-the-art in multipath video transmission focusing on IoT applications, introduce a taxonomy to classify existing approaches based on their application-specific mechanisms as well as networking-specific techniques, and present a multipath selection framework based on a cross-layer approach for IoT video transmission, in which information of video flow requirements from the application layer is used to find better paths for these flows. The main contribution is the FITPATH, an efficient and adaptable path selection mechanism for multipath routing that can accommodate scenarios where multiple video sources can simultaneously transmit heterogeneous video flows. An additional contribution is the development of the Multimedia-Aware Performance Estimator (MAPE), the first deterministic simulation-based estimator that provides per-flow throughput, packet loss and delay estimates while considering inter-flow interference and flows with specific transmission rates, typical of video traffic. MAPE is able to provide, in *quasi* real-time, network performance estimates that can be used by FITPATH to inform multipath selection in IoT video transmission. Our results show that MAPE yields higher accuracy at a fraction of the execution time when compared to stochastic network simulators, and FITPATH outperforms various existing path selection mechanisms both in terms of user Quality of Experience (QoE) and network performance. Finally, we demonstrate how FITPATH can be used in practice to quickly generate solutions with better video transmission quality.

Keywords: Multipath Routing, Wireless Video Transmission, Performance Estimator, Multipath Selection Mechanism, IoT video applications.

List of Figures

List of Tables

List of Acronyms

Contents

Chapter 1

Introduction

In the last few years, we have witnessed the rapid proliferation of IoT multimedia deployments and applications. According to recent Internet projections from Cisco [31], nearly half (47%) of all devices and connections will be video capable by 2023. This is likely a result of the proliferation of end-user devices equipped with low-cost, yet relatively powerful video technology combined with the ever increasing availability of wireless communication infrastructure, which in turn, has contributed to the unprecedented growth of a variety of video applications and services. In this thesis, we consider such applications, including public transportation, traffic monitoring, parking and surveillance systems in smart cities [8, 24, 70] collect and transmit video over a multi-hop wireless network.

As the number and diversity of video applications increase and as IoT becomes more heterogeneous incorporating legacy as well as new technology (e.g., 5G and beyond, IEEE 802.11ac), it is imperative to gain a solid understanding of the interplay between the characteristics and requirements of video applications and the core functions and features of the underlying network in order to adequately support existing and emerging Internet and IoT video systems.

Like other services that transmit video streams, IoT video applications have stringent QoS requirements in terms of throughput, delay and packet loss [53]. Mutipath routing, a routing strategy that tries to find multiple paths between the source and destination for distributing generated flows, has been proposed as an effective strategy to help meet QoS requirements of video transmission by providing adequate bandwidth and delay guarantees [104] as well as reliability and resilience [90, 53].

Several recent studies have been presented to support the required high efficiency of wireless video transmission, which in turn can be applied in a variety of multipath video applications [12, 13, 35, 1, 94, 76, 29, 48, 61, 101, 33, 54, 1, 56, 108, 7, 10, 11]. However, no consensus solution has emerged thus far, as multiple issues remain open, mainly about routing metrics and cross-layer approaches integrating the application and network layers. Further, there is no current one-size-fits-all solution that can cater to the wide variety of IoT video application scenarios.

In order to fully leverage its ability to find multiple paths for transmitting video flows, multipath routing must be able to, in a timely manner, select routes that meet the requirements of the driving video application (e.g., multiple video sources transmitting flows at specific bitrates [53]), while accounting for the current conditions of the underlying network. By highlighting the interplay between IoT application features and the underlying network, this thesis introduces an efficient wireless multipath selection mechanism using a cross-layer approach that estimates the conditions of the underlying network in real time while accounting for the application requirements.

1.1 Application Scenarios

This thesis intends to cover a wide variety of IoT systems that demand live streams from multiple sources to a sink. It targets multihop IEEE 802.11 based networks since it is evolving towards even higher speeds and richer features for supporting video transmission. In addition to showing better resilience under natural disaster situations, wireless multihop networks facilitate scalability by providing flexible deployment when there is no infrastructure, or the existing infrastructure is inconvenient or expensive. Our application scenarios assume that all network nodes can work as video sources as well as relay nodes, are powered by a continuous energy source, and have resources and processing capacity to deploy the routing mechanisms and video services.

Live streaming is an emerging video service of IoT systems, in which source cameras capture live videos and stream them to the destination. In IoT video systems, this destination is usually a monitoring control center where the staff watches videos in real time. This is particularly challenging due to the real-time system requirements such as throughput and delay, where packet retransmission or buffering is less acceptable. In this case, applications are influenced by several design aspects such as limited hardware resources, network topology, and the routing protocol [8].

Multiple video sources enable several video services which have been used in different IoT application scenarios, such as Connected Health [5, 52], Industrial IoT [5, 125], Smart

City [36, 8, 70] and Environmental Monitoring [70, 40]. In these cases, the number of active video sources depends on the requirements of the monitoring center. Usually, sources are enabled by segments that require attention, and the video resolution is set according to the viewer's needs.

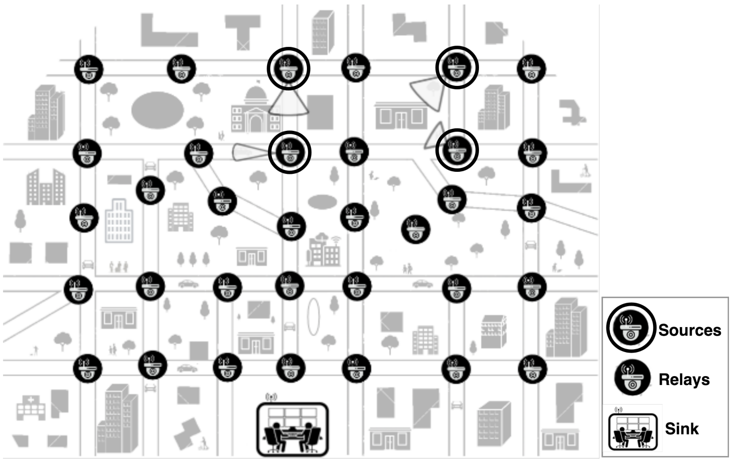


Figure 1.1: Example of a wireless video surveillance system with multiple video sources transmitting to a monitoring center.

A hypothetical case that illustrates the prototypical scenario considered in this thesis is illustrated in Figure 1.1. It represents an urban area with cameras deployed along the streets of a neighborhood of a smart city, where the video sources send live video streams to a monitoring center over a multihop wireless network, without any type of other concurrent flows. Those multiple sources can transfer flows simultaneously according to the demands of the monitoring center. For example, the monitoring center demands a set of network nodes to act as video sources with determined levels of video quality, such as video services to track an event or a target in a street or a block of the city region. In this case, each source may generate a known number of video flows with specific bitrates, according to the video resolution and encoder used. These flows are transmitted from each source node through selected paths. At the destination, i.e., the sink node, the video decoder is responsible for synchronizing and merging the received flows to render the received video flows. Depending on the application scenario, the system can also handle multiple sinks by specifying the sink for each source. This scenario is not restricted to outdoor environments. IoT video systems are also usually deployed in indoor environments considering specific topologies.

1.2 Problem Statement

Efficient transmission of multimedia traffic in multihop wireless networks poses significant challenges mainly due to their more stringent QoS requirements (e.g., throughput and delay), especially in the case of real-time applications [53]. There are several types of live streams services where different sources can use different numbers of flows and bitrates. The number of sources and their bitrates can influence users' QoE, especially if the multipath routing protocol ignores applications' QoS requirements [113]. In addition to satisfying applications' QoS requirements, a routing protocol for video transmission should target network reliability and delay constraints as well as improving users' QoE.

Applications with multiple video sources transmitting at specific flow bitrates, for example, require more complex path selection approaches to satisfy user QoE because the probability of paths sharing nodes and links can be significantly higher. Additionally, multihop wireless communication is inherently more prone to losses and congestion; for instance, the performance of a single wireless link can vary due to the link-layer bitrate, its Signal to Noise Ratio (SNR), and complex propagation phenomena. Uncontrolled transmissions of multiple flows can also cause heavy congestion, as well as flow interference, medium access contention, and collisions [63]. And, in the specific case of multimedia traffic, even though compression techniques use a pre-defined average data rate as a target, the actual data rate of the compressed flow may vary considerably depending on scene complexity, flow resolution, and the different types of frames [102]. Depending on the number of simultaneous flows, the likelihood of losing or delaying video frames that are particularly loss- and delay-intolerant may increase as a consequence of flow interference, mainly due to collision and queuing.

During video transmission, distortion in received video largely depends on the position of packet losses or delay in the transmitted video flow. To identify whether the packet losses will distort the received video or not is a difficult task. Video traffic requires high bitrate and the compression method uses a bitrate target that may vary according to the video resolution. Depending on how dynamic a video scene is, it can generate irregular flow rates even when considering relatively long interval times, mainly due to the different types of video frames [102]. Therefore, the routing protocols need to be adapted according to the video coding, in terms of video compression techniques and flow rates.

Traditional multipath routing metrics to select optimal paths for the video flows, such as shortest paths and hop count, are not necessarily the best options because they analyze links and paths individually and do not have a global network view. In addition, they increase the probability that two or more paths share a link, which causes self-interference. The disjoint path selection mechanism is adopted in several proposals and can provide better performance and reliability, but disjoint paths are rarely available for all video flows from multiple sources [15]. As the routing protocols need to provide an efficient interplay between video coding and video transmission to adequately support wireless multipath transmission, the challenge is to efficiently solve the multipath selection problem maximizing the users' QoE. It can be formulated as an optimization problem that aims to find multiple paths from each source with the highest possible level of QoE, subject to the inter-flow interference, QoS constraints, and video services demands.

1.3 Research Questions

According to the problem statement, it is related to a cross-layer approach integrating application and network layers which can provide a multipath selection based on users' QoE estimates in IoT application scenarios with multiple video sources. Hence, this thesis is guided by the following research questions:

- **Q1**: How to select the optimal paths for each video flow that maximizes the users' QoE?
- **Q2**: How to estimate the users' QoE from multiple video sources transmitting simultaneously, considering the inter-flow interference, QoS constraints and video services demands?
- **Q3**: How to provide an efficient interplay between sources' video coding and multipath routing?

1.4 Goals

In order to answer the presented research question, the aim of this thesis is to investigate the wireless multipath video transmission for IoT applications scenarios and to provide a cross-layer multipath routing scheme for IEEE 802.11 based networks to maximize the users' QoE, considering the characteristics of video traffic and the inter-flow interference from multiple video sources transmitting simultaneously. The proposed solution consists of an efficient and adaptable path selection scheme for multipath routing that can accommodate environments where multiple video sources can simultaneously transmit heterogeneous video flows, i.e., flows of specific bitrates.

This broad goal can be further divided into the following more specific objectives:

- Providing a thorough review of the current state-of-the-art in multipath video transmission focusing on IoT applications;
- Proposing a taxonomy that maps the design space of wireless multipath video communication by exploring the synergy between IoT application scenarios, their features and requirements, and core networking functions such as routing and forwarding;
- Analyzing the characteristics of video traffic by extracting a standard behavior from the wide variety of IoT video services requirements;
- Providing a cross-layer routing design for deploying multipath video transmission in both centralized and decentralized network architectures;
- Proposing a multimedia-aware performance estimator that provides real-time QoS estimates while considering inter-flow interference and specific flow rates, typical of multimedia traffic; and
- Proposing a heuristic-based optimization approach that selects flows' paths while accounting for the conditions of the underlying network and the video application requirements.

1.5 Thesis Outline

The remainder of this document is organized as follows:

- Chapter 2 presents a survey on wireless multipath video transmission, introducing a taxonomy to classify existing approaches based on their application-specific mechanisms as well as networking-specific techniques;
- Chapter 3 presents a multipath selection framework for IoT video transmission, including the hypothesis that underlines this thesis;
- Chapter 4 presents the proposed multimedia-aware performance estimator, including experiments and results;
- Chapter 5 presents the proposed wireless multipath selection mechanism, including experiments and results; and
- Chapter 6 presents the conclusions, including contributions and directions for future work.

Chapter 2

Wireless Multipath Video Transmission: Background and Related Work

This chapter provides a thorough review of the current state-of-the-art in wireless multipath video transmission focusing on IoT applications. It introduces a taxonomy to classify existing approaches based on their application-specific mechanisms (e.g., video coding techniques) as well as networking-specific techniques, considering the interplay between core networking functions, notably routing and forwarding, and the wide spectrum of IoT application scenarios and their requirements.

2.1 IoT Video Application Scenarios

There is a wide variety of IoT applications that involve video transmission. In this section, we discuss a number of IoT scenarios and their specific features and requirements related to video transmission, as illustrated in Figure 2.1.

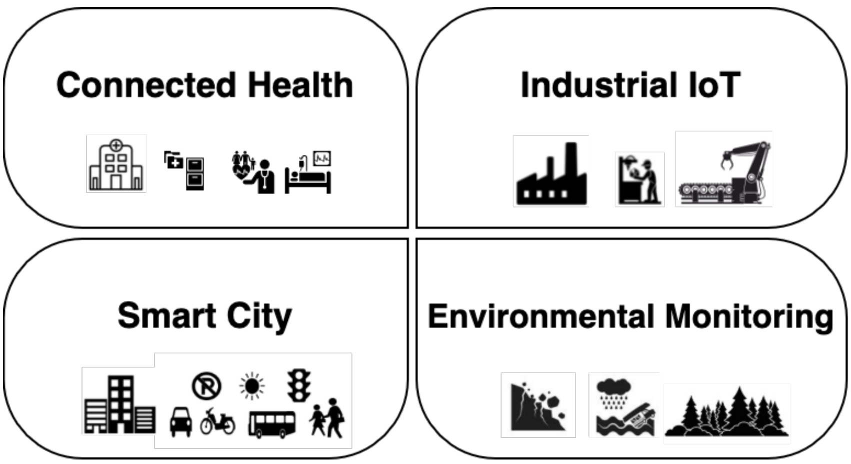


Figure 2.1: Examples of IoT video application scenarios

Table 2.1 lists common IoT video applications, their networking requirements, and references to prior work that focus on video transmission in the context of these applications. Note that all references mention reliability as one of the requirements for these applications. Reliability is defined as the probability of successful transmission of video frames to guarantee maximum users' QoE. As it depends on the paths through which the video are transmitted, several studies consider multipath routing strategies as one of the best ways to achieve reliable video delivery [27, 44, 100, 38, 101, 41, 65, 4, 57, 54].

In IoT video applications, it is important to provide security mechanisms to protect the information being transmitted, the identity of the user, etc. Security of video transmission is becoming increasingly more challenging because standard encryption techniques like AES/DES may not be efficient [103]. To address this gap, cryptography and steganography techniques have been used in scenarios where secure multimedia transmission is critical [67, 84, 69, 103, 46].

Table 2.1: IoT video applications and their QoS requirements.

References	IoT Scenarios	Video Application	Network Requirements
[8, 5, 52, 4, 40]	Connected Health	Remote Patient Observation	Security, Reliability
		Real-Time Examination	Throughput, Delay, Security, Reliability
[5, 52, 125, 4]	Industrial IoT	Process Control Systems	Delay, Security, Reliability
		Product Inspections	Throughput, Delay, Security, Reliability
[36, 4, 104, 40]	Smart City	Person Tracking	Throughput, Delay, Security, Reliability
		Vehicle Traffic Management	Security, Reliability
[70, 40]	Environmental Monitoring	Forest Monitoring	Reliability, Energy
		Natural Disaster Detection	Throughput, Delay, Reliability, Energy

2.1.1 Connected Health

Healthcare systems have been increasingly relying on video for applications such as real-time diagnostics and real-time patient monitoring [5], both in healthcare facilities or at home. Additionally, video from patients can be stored for offline analysis [52].

Telemedicine is another important application that allows patients to be examined remotely by doctors (e.g., specialists) who may not be available where the patient resides. For remote examination, data from other sensors such as body temperature, blood pressure, and breathing activity can also be examined [52]. Telemedicine applications typically require real-time video transmission with low delay and high bandwidth.

In order to support connected health services, IoT nodes are deployed indoors (e.g., in hospitals, clinics, patient homes), and, as such, communication may be affected by obstacles, walls, as well as interference from other equipment.

2.1.2 Industrial IoT

In industrial applications, video can be used in the inspection and control of industrial equipment that requires high accuracy, availability, and reliability to support high precision manufacturing processes such as those used in semiconductor chips, automobiles, food or pharmaceutical products [5].

In industrial IoT deployments, cameras are usually installed at different points in industrial plants and may need to operate under challenging conditions. Yet, they may be required to meet stringent quality-of-service requirements such as high bandwidth in order to deliver high video resolution, high reliability, and low data transmission delay [125].

2.1.3 Smart City

One of the most common video applications in smart cities is security and surveillance of outdoor or indoor spaces to control access to an area, detect unauthorized visitors, among other purposes [36].

Large urban centers have been adopting camera networks to help manage public transportation, traffic, road conditions, and parking [8]. Video from multiple cameras can be analyzed in real time or offline using computer vision techniques to provide valuable information regarding peak hours, traffic and road conditions/incidents, routes, and more [70].

Note that the network requirements vary with application-specific requirements. For example, traffic management only needs enough reliability to get details to distinguish vehicle contours from the surroundings and track their movement. On the other hand, person tracking applications require high network throughput because they depend on high-definition video to enable identifying and tracking facial features [36].

Network nodes are usually deployed on poles on the streets [8]. As in all urban environments, wireless communication in smart cities can suffer from large obstacles (such as buildings), noise and interference. Thus, the systems' performance is also highly dependent on the distribution of the camera nodes, as well as the other aforementioned factors.

2.1.4 Environmental Monitoring

Environmental monitoring is another important IoT application domain that uses video as well as other types of sensor data to monitor wildlife habitats, forests, and oceans. Such deployments usually target remote, hard to access locations [70] and therefore must be energy efficient to maximize their operational lifetime independent of human intervention (e.g., battery replacement). In general, multiple cameras are deployed in wide outdoor areas where long range communication may be needed [40].

In applications for disaster management, like wildfires, floods, or landslides, in addition to reliability and energy efficiency, low latency and high throughput are also required for timely event detection and real-time event tracking [70].

2.2 Taxonomy

We propose a novel taxonomy that charts the design space of wireless multipath video transmission systems by exploring the interplay between IoT application features and requirements and core networking functions. Based on our taxonomy, we classify and discuss a broad range of existing wireless multipath video transmission approaches for IoT applications. We examine how aspects specific to IoT video services influence the design of core network protocols as well as how the underlying networking issues play a role in video transmission application design.

Examples of application-specific aspects for video transmission include techniques that seek to improve coding efficiency and mechanisms to enhance video compression ratio. Key networking services include efficient routing and forwarding, and reliable encoded/compressed video stream transmission.

To better capture the close interaction between application and networking factors specific to video transmission in IoT scenarios, we opted for using Unified Modeling Language (UML) [99]) notation to represent this taxonomy. UML was proposed as a way to standardize how software system design and development processes are represented visually. While UML provides a wide range of representations and symbols, for our taxonomy, we use *classes*, *composition* and *specialization* relations. Composition is represented by a filled diamond and a solid line, indicating that one class is composed of others. Specialization is represented by a hollow triangle close to the more generic class in its connection with the more specialized classes.

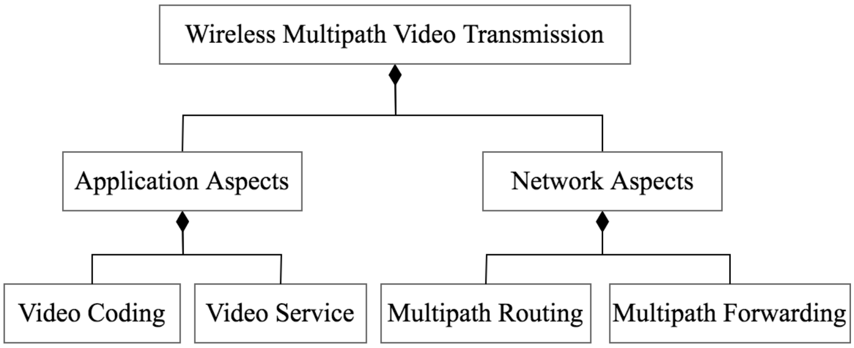


Figure 2.2: Wireless multipath video transmission taxonomy for IoT applications.

Our taxonomy was created based on an extensive review of the literature on IoT video transmission, in particular wireless multipath approaches. As will become clear in the remainder of this Chapter, not every work in the literature tackles all components identified in the taxonomy. Indeed, several proposals focus on a single feature or on a subset thereof. And in fact, one of the important takeaways from our study and taxonomy is what areas of wireless multipath video transmission targeting IoT applications can benefit from further investigation. Figure 2.2 illustrates the top layers of our taxonomy showing that existing wireless multipath video transmission proposals typically consist of application and networking components. In turn, the application component may comprise video coding techniques and other video services, such as real-time or on-demand video streaming. On the other hand, network aspects can be divided into multipath routing and multipath forwarding strategies.

2.2.1 Application Aspects

The main application aspects identified in the multipath video transmission literature focusing on IoT applications were various video coding techniques as well as other video services. Video coding includes video compression as well as error correction techniques. Video services refer to the types of video transmission supported by the application, such as streaming (live or on-demand) as well as different video processing services including computer vision algorithms for target tracking, situation awareness, and multi-camera coordination. As will become clear in this study, these application features and services can affect both the quality of the video being transmitted as well as how the video flow is transmitted through the network.

2.2.2 Network Aspects

Network aspects underlying multipath video transmission for IoT applications have attracted considerable attention from the research community. Most of the work in the literature focuses on multipath routing — i.e., finding multiple possible routes in the network from source to destination — and forwarding — i.e., once multiple paths are found, how to forward video traffic using these routes.

This thesis, like most studies on wireless multipath video transmission, is focused on the requirements of IoT applications that need to transmit multimedia content with a particular level of QoS [6].

Meeting the requirements of IoT applications is challenging due to factors such as resource constraints at participating nodes, e.g., limited processing storage, and communication capabilities, as well as battery lifetime.

In video transmission, multipath routing is a fundamental mechanism to determine the network paths that will be used to send video traffic. For example, different routing metrics have been explored with the goal of improving video transmission quality [9]. Moreover, path quality also depends on route discovery and selection mechanisms used by the routing protocol. In this thesis, we identify the main building blocks used by existing multipath routing approaches, namely: Route Discovery and Maintenance, Routing Metrics, Multipath Selection, and Multipath Forwarding. We should point out that, in most routing protocols, these components are tightly coupled and their functions may overlap with each other. As discussed in Sections 2.5 and 2.6, conceptually decoupling them facilitates the understanding of the state-of-the-art in wireless multipath routing for video transmission focusing on IoT applications.

Additionally, cross-layer approaches, which allow the exchange of information across multiple protocol layers in order to improve the performance of multipath video transmission, are also discussed. For instance, video frames can be marked at the application layer according to their importance, allowing the network layer to distinguish them and possibly forwarding them through different paths in order to improve video delivery quality [1].

2.3 Video Services

As illustrated in Figure 2.3, we consider two different types of video services, namely streaming and processing. Video streaming refers to real-time video transmission where the video is transmitted without any processing other than video encoding, while video processing provides additional services to improve performance, increase energy efficiency, or decrease bandwidth requirements.

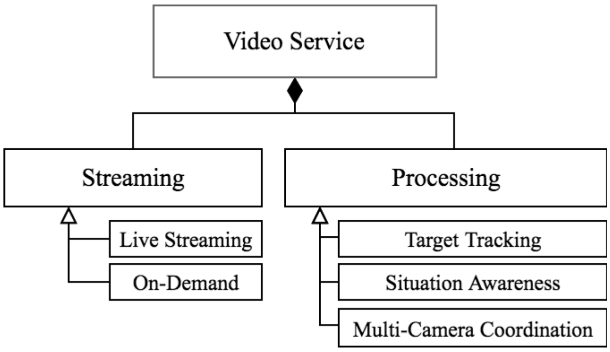


Figure 2.3: Video services commonly associated with multipath video transmission.

Table 2.2 summarizes the main video service components and their definitions as commonly found in the literature.

Table 2.2: Types of Video Services for Wireless Multipath Video Transmission.

Video Services		Definitions	References
Streamming	Live	Cameras collect and transmit live video coverage of an event	[111, 14, 12, 13, 34, 35, 100, 57, 106, 2, 128, 41, 95, 94, 76, 49, 123, 39, 29, 48, 61, 101, 116, 90, 124, 33, 54, 1, 56, 108, 7, 10, 11]
	On-Demand	Video is stored so that it can be later requested by users	[28, 43, 42, 51, 27, 129, 119]
Processing	Target Tracking	Recognition and tracking of people or objects	[4, 65, 70, 80, 16]
	Situation Awareness	Aggregation of perception information to reduce data transmission	[64, 6, 70, 80, 18]
	Multi-Camera Coordination	Coordination between neighboring cameras to share or fuse their videos	[88, 82, 117]

2.3.1 Video Streaming

Video streaming is commonly used for both live transmission or on-demand services.

2.3.1.1 Live Streaming

In the context of IoT applications, live video is a streaming service in which source cameras capture live videos and stream them to the destination in real time. In live streaming applications, the destination is usually a monitoring center where staff may need to watch videos in real time. Live streaming is particularly challenging due to their stringent delay and throughput requirements as packet retransmission or larger buffering size are typically not an option. Hence, a number of approaches have been proposed to address live streaming's quality-of-service needs, which can be supported by multipath video transmission.

2.3.1.2 On-Demand

On-demand video services allow users to watch pre-recorded videos at a desired time. They are used in a variety of IoT domains such as video analytics applications for public safety and traffic planning in smart cities. In these applications, videos are captured and stored for future on-demand viewing. The server where the video is stored may also perform some processing to filter and extract information about the scenes — for example, activity analysis. In addition, video summaries can be generated to help users navigate through large video sequences and retrieve the ones that are most relevant to the analysis. To this end, different video summarization techniques using methods based on clustering and deep learning are presented in [73, 72, 75, 74, 71]. Note that, unlike real-time streaming, on-demand video service does not have the same constraints, such as a maximum end-to-end delay. Users can fast-forward, rewind and play back the video as many times as necessary, and thus, techniques like buffering can be used more aggressively. Some notable efforts that consider exclusively real-time video transmission include [28, 43, 42, 51, 27, 129, 119].

2.3.2 Video Processing

Video processing services are typically used by video applications to reduce transmission of redundant information, perform preliminary video processing such as combining data originated from multiple views, on different media, or with different resolutions, as well as filtering and extracting semantically relevant information from the environment. Additionally, video processing algorithms can be performed by network nodes along the transmission paths. Figure 2.4 illustrates the main video processing services identified in the literature.

2.3.2.1 Target Tracking

Target tracking, which is the main focus of the work reported in [4, 65, 70, 80, 16, 75], aims to identify a person or object in motion, enabling cameras to follow the target. Access

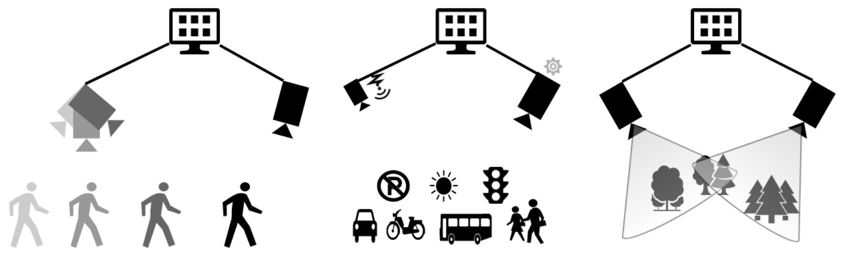


Figure 2.4: Different types of video processing services.

control, monitoring traffic and surveillance applications as well as habitat monitoring are some of the applications that employ target tracking.

Target tracking usually consists of three phases: target detection, recognition, and tracking. *Detection* refers to identifying the presence of a new target whenever it enters the monitored environment. *Recognition* is used to determine if the object is of interest and whether further processing is warranted. Finally, *tracking* consists of enabling cameras to follow the target as it moves through the environment. Target tracking has raised some interesting research issues. A notable example is coordinating multiple cameras as the target object moves through the region. Some works have incorporated object localization to accurately detect the moving object and predict the trajectory by estimating the movement pattern of the object [70, 88]. In recent years, deep learning-based methods have been developed with the goal of facilitating visually tracking objects [86].

2.3.2.2 Situation Awareness

Many applications do not need a raw video to be transmitted all the time [64]. For example, in surveillance or event tracking applications, as long as there are events detected, video can be compressed to a simple scalar value or not be transmitted altogether. In environmental monitoring applications, weather data can be used to enable video cameras, as well as distributed filtering techniques can be applied to create time-lapse images to produce some forecast patterns that can anticipate future disasters [5].

These types of video processing rely on the perception of certain elements in the environment to decide whether or not to transmit raw video, snippets of the video, or simply a scalar value. As such, they can dramatically reduce the amount of data transmitted over the network, therefore optimizing network resource utilization and energy efficiency.

SensEye [70] is a notable example of an approach that uses a hierarchical architecture for energy-efficient event tracking. They propose a two-tier camera deployment where the bottom tier consists of low power cameras which trigger higher resolution cameras at the top tier in an on-demand fashion. Another hierarchical deployment-based approach for critical-event surveillance was proposed in [18] which uses densely deployed low-cost audio sensor nodes as the bottom tier while the second tier is equipped with high-cost sparsely deployed rotational video nodes. The first tier performs the preliminary audio event detection task that sends an alarm message to activate the rotational video nodes to cover the event. Another example of a system that uses video processing was proposed in [80] where a processing proxy server analyzes all video flows and only alerts the monitoring center if relevant events are detected.

2.3.2.3 Multi-Camera Coordination

This is a service typically used by applications such as surveillance, target tracking, and control of automated systems where multiple cameras fuse their videos. Natarajan *et al.* [88] present a survey of various techniques for multi-camera coordination and control that have been adopted in surveillance systems.

Multi-Camera Coordination can also provide support to free-viewpoint applications. Free-viewpoint video is a technology that enables 3D visualization of a scene by freely changing the viewpoints. Liu *et al.* [82] proposed a system for multipath transmission of free-viewpoint video with joint recovery capability for both inter-view (left and right views) and temporal description. The proposed approach correlates viewpoint similarities from two nearby cameras. This technology enables motion parallax — a viewer's head movement triggers a corresponding shift in the viewing perspective of the observed scene. Depending on the intermediate virtual view currently requested by the client, texture and depth maps from the two nearest camera viewpoints are encoded for transmission.

A framework for transmitting real-time surveillance video with a wide viewing angle and high resolution using multi-cameras was proposed in [117]. They consider a video surveillance scenario in which the signals from two rotatable monitoring cameras can be combined to obtain a wider viewing angle. A video mosaicing method is adopted to merge the videos generated by the two cameras into one seamless video, which is transferred to the destination in real-time.

2.3.3 Summary

Video processing services usually complement video streaming services. In this section, we intended to identify the video streaming and additional processing services indicated for the proposed multipath video transmission found in the literature. Table 2.3 summarizes the different video streaming and video processing services found in the literature. We observe that while most studies focus mainly on video streaming services, in particular live streaming, video processing is also tackled by a significant number of references. This observation indicates that one needs to consider the type of video service used by IoT applications when designing multipath video transmission because of their different quality of service requirements.

Table 2.3: State-of-the-art in video Streaming and video processing.

References	Video Streaming		Processing		
	Live	On-Demand	Target Tracking	Situation Awareness	Multi-camera coordination
[1, 2, 14, 12, 13, 34, 35, 41, 7, 10, 11, 29, 33, 39] [49, 48, 54, 56, 61, 57, 76, 90, 94]	✓				
[95, 100, 123, 101, 116, 124, 108, 111, 106, 128] [28, 43, 42, 113, 129, 51, 27, 95, 119]		✓			
[80]	✓		✓		
[70]		✓	✓		
[18]	✓		✓		✓
[4]	✓	✓		✓	
[65, 16]	✓			✓	
[64]	✓	✓		✓	
[82, 117]	✓				✓

2.4 Video Coding

As illustrated in Figure 2.5, the main video coding techniques explored in the literature are compression and error correction. Video coding technologies have been extensively used as a way to increase video compression ratio as well as improve the efficiency and visual video quality. Many studies explore video compression techniques that split the video stream into sub-streams to transmit over different paths. In addition, some of them adopt error correction techniques to provide greater reliability.

Video compression techniques fall into two basic types, namely Layered Coding (LC) and Multiple Description Coding (MDC). Both generate multiple sub-streams that can be transmitted over multiple paths. In LC, the sub-streams are mapped into layers where the base layer stream is the most important and provides a basic quality level that can be improved with additional enhancement layers [58]. LC techniques can be classified as Single Layer (SL) and Scalable Video Coding (SVC), where SL generates a single video stream for transmission, while SVC encodes the video as a set of hierarchically layered sub-streams. In MDC, all sub-streams have equal importance, as different quality levels can be obtained with different combinations of sub-streams.

Error correction techniques can be categorized according to the roles performed by the encoder and decoder. In Error Resilience (ER), the encoder adds redundancy to the video streams, allowing certain levels of errors to be recovered at the decoder. Error Concealment (EC), on the other hand, adds post-processing methods to the decoding process to improve the perceived quality of the reproduced stream in face of errors.

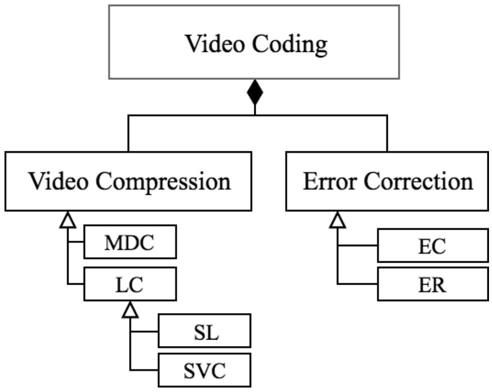


Figure 2.5: Classification of video coding techniques for multipath video transmission.

2.4.1 Video Compression

Video compression techniques segment each frame into processing units called *macroblocks* which are then compressed to reduce the required bandwidth to transmit the video while maintaining acceptable visual quality. Compression exploits macroblocks spatial and temporal correlations as specified by video coding standards.

For example, Motion Picture Expert Group (MPEG) video compression [78] segments the video sequence into Groups Of Pictures (GOP) that determine the organization of frames. Each group includes three types of frames: Intra (I-frame), Predictive (P-frame), and Bidirectional (B-frame). I-frames are compressed independently and do not require additional information to be reconstructed. They are used as references for forward and/or backward prediction to decompress the P and B-frames.

MPEG video compression standardizes decoder structures and bitrates to enable the development of efficient encoding algorithms. It supports a wide range of application-specific parameters and provides a framework for extending layered coding and multiple description coding as a way to support emerging video applications.

The H.264/AVC Advanced Video Coding standard has been widely used by High Definition (HD) video applications [57]. More recently, the H.265/HEVC High-Efficiency Video Coding standard has emerged to support Ultra-High Definition (UHD) video transmissions and provide lower bitrate. While more complex, H.265 results in almost 50% bandwidth requirement reduction compared to H.264/AVC at the same reproduction quality [113]. The main advantage of H.265/HEVC is the use of more flexible macroblocks – a fundamental unit of the video coding process, allowing encoding predictive macroblocks of different sizes.

Some early multipath video transmission approaches have adopted the H.264/AVC and H.265/HEVC video coding standards with SL techniques. However, more recent studies have proposed scalable extensions using SVC and MDC techniques. These different approaches are discussed in more detail below and are illustrated in Figure 2.6.

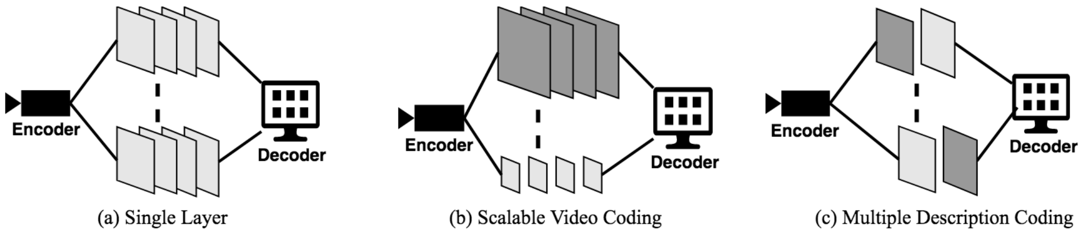


Figure 2.6: Video compression techniques.

2.4.1.1 Layered Coding

LC has become part of the established video compression standards [28, 58]. It provides layered embedded bitstreams that are generated at different bitrates, encoding a video sequence into multiple layers without compromising video quality. Layered representations provide a convenient way to perform rate control to mitigate network congestion and can use single or scalable layers.

- **Single Layer** SL techniques perform video compression using only one layer. Approaches to multipath video transmission with SL have adopted the H.264/AVC and H.265/HEVC standards [1, 100, 4, 16, 80, 57, 27, 106, 111, 2, 128, 41, 3]. In general, the video is segmented into GOP which are encoded and transmitted on different paths. The better paths – selected based on routing metric – are dedicated to I-frames since video decoding relies heavily on I-frames and I-frames require higher bandwidth than P- and B-frames. P- and B-frames can be transmitted using alternative, lower bandwidth paths. B-frames typically have the lowest bandwidth requirement and B-frame encoding requires the acquisition of the corresponding I-frame or P-frame, which introduces delay. For this reason, some proposals only use I-frame and P-frame encoding modes [80, 100].

- **Scalable Video Coding** In SVC, the video streams are divided into a base layer and one or more enhancement layers. SVC layers are hierarchical: a given layer cannot be decoded unless all of its lower layers have been received correctly. Enhancement layers complement the base layer to improve visual quality in terms of temporal, spatial and SNR scalable modalities. The MPEG standards include extension tools to support these SVC modes. The scalable extensions for the H.264/AVC and H.265/HEVC standards are called H.264/SVC and H.265/SHVC (Scalable High-definition Video Coding), respectively.

  Generally, multipath proposals based on H.264/SVC transmit the base layer over the better paths and the enhancement layers over the marginal paths [129, 95, 35, 94, 43, 76]. Further, the H.265/SHVC encoding was adopted in [113, 12, 13] due to the better compression, reducing latency and bitrate. With H.265/SHVC, the control of the bitrate of each layer can be handled according to the bandwidth available for each path.

  SVC can be extended to address application-specific requirements. As an example, a reference-frame-cache-based surveillance video transmission system (RSVTS) was proposed in [117]. It is an H.264/SVC extension that implements a method for merging multiple rotatable cameras to deliver wide-viewing-angle and high-definition video. In addition, it implements a reference frame cache on both the sender and receiver sides to improve video quality by increasing the probability to achieve adequate decoded video quality. The Scalable High-efficiency Inter-layer Prediction based Video coding (SHIPVC) was proposed in [61]. It extends H.265/SHVC to implement two-layer predictions such as texture color and motion with different quantization parameters.

2.4.1.2 Multiple Description Coding

MDC has been proposed as an alternative to layered coding for video streaming [28]. Each description alone can guarantee a basic level of reconstruction quality at the decoder, while additional descriptions further improve quality.

The key idea of MDC is to partition the video stream into two or more independently decodable and mutually refinable descriptions. Several techniques to generate video descriptions using MDC are discussed in [50]. The number of descriptions can be defined according to application requirements, and the partitioning may be in the spatial or temporal domains. In the spatial domain, the descriptions are generated by a process performed at the pixel level. In temporal domain MDC, the descriptions are generated by a process performed at the frame level. Each description's individual packets can be transmitted separately through different paths. If packets are lost, the video may still be successfully decoded using packets carrying the other descriptions, albeit with lower fidelity. As such, MDC provides a solution to mitigate video quality degradation in the presence of packet losses, bit errors, and burst errors during transmission [49].

The combination of MDC with multipath routing has been proposed to reduce network congestion by exploring path diversity to balance traffic load [14, 119, 123, 49, 39, 64, 29, 20]. Some studies propose to classify descriptions in order to define packet priorities and thus improve MDC video streaming robustness [34, 39].

A texture-plus-depth format of free-viewpoint video was proposed in [82]. It employs MDC with H.264/AVC for a multi-view representation where multiple texture maps from closely spaced capturing cameras are encoded into one bitstream.

One way to generate MDC is to explore the GOP or macroblock structure of MPEG video coding standards. In this context, studies suggest the Flexible Macroblock Ordering (FMO) of the H.264/AVC standard as more appropriate [123, 82, 48]. FMO refers to rearranging macroblocks in groups where each group is a description of the video according to specific standards.

Hybrid approaches combining the advantages of both SVC and MDC techniques have also been proposed [116, 101, 48]. In these approaches, the descriptions of each layer are generated in the FMO format. Then, macroblocks of each layer are sent over disjoint paths.

Video coding can also generate variable bitrate during packet forwarding to maintain the quality of video transmission when facing limited available bandwidth. In this way, some studies have proposed adaptive video coding to reduce the number of video frames or enhancement layers in order to reduce the required bitrate. Cross-layered designs have been proposed to control the bitrate [113, 95, 128, 2]. In these designs, each layer of an encoded video stream is handled according to the status of the available bandwidth. In [129], the available bandwidth of the different paths is estimated in order to determine the number of enhancement layers and select optimal paths to transmit each layer.

In adaptive video coding forwarding, video packets are encoded at the source node, but intermediate nodes can re-encode them before forwarding. This type of coding presented in [35] is called intra-session network coding or transcoding and can adapt to the available network bandwidth and improve video streaming quality.

2.4.2 Error Correction Techniques

Error correction techniques have been proposed for multipath video transmission to mitigate the effect of packet losses due to network congestion and transmission errors and thus limit their impact of video distortion and video quality deterioration. Typically, error correction techniques are integrated into video compression to improve the reliability and robustness of video decoding. In the specific case of multipath video transmission, the main techniques adopted in prior work are ER and EC.

2.4.2.1 Error Resilience

In ER, the encoder also adds redundant packets at the transmission source to enable error detection and correction. This well-known technique is also known as Forward Error Correction (FEC).

Error Resilience techniques aim to minimize the transmission errors' impact upon video decoding. ER can be combined with Layered Coding techniques. For example, it can set optimal video compression parameters in SL to generate packet flows that respect the capacities of the network. Recommendations for setting the coding parameters can be provided by the physical layer and sent from destination to source. This way, each source node can adopt different parameter settings according to the channel state, such as link reliability and stability. Thus, some studies have proposed that the destination periodically sends the recommended video compression parameters after analyzing the video received during the previous period in different states of the wireless channel [128, 2].

An overview of ER techniques for SVC referred to as inter-layer FEC is presented in [58]. Most approaches combine ER with video compression to protect the base layer using the enhancement layers [48, 35]. In [48], the base layer is duplicated and transmitted as redundant packets over different paths. In [35], packets with the highest priority (e.g., I-frames) are re-encoded with redundant data during generation.

ER can also be combined with MDC [119, 82, 124]. In [82], redundant packets are generated for all MDC descriptions. The work reported in [124] proposes adaptive ER algorithms that dynamically generate redundant packets based on current network conditions.

2.4.2.2 Error Concealment

In Error Concealment techniques, the decoder tries to compensate for missing information so that the visual quality of the reproduced video is not severely compromised. EC can exploit spatial and temporal redundancy in video compression in order to recover from network losses.

A survey of EC techniques applied to LC [112] grouped existing approaches into *intra-layer EC* and *inter-layer EC*. Intra-layer EC explores the spatial domain based on pixels, using either texture features or edge/object-shape information to conceal lost frames. Inter-layer EC uses temporal information to estimate the lost motion vectors, which are used to recover damaged macroblocks by considering a reference frame. Usually, only one of those methods is suitable for a whole-frame, although it is possible to use both or different methods for each macroblock.

When SVC is used for coding, it can improve the EC by exploiting the similarity among the layers. In this approach when a macroblock, slice or frame of the enhancement layer is lost, the corresponding part of the frame in the base layer can be used to conceal the lost data [48].

In [113], a cross-layer framework for video streaming with inter-layer EC was proposed. It applies EC based on feedback messages aiming to improve the end-to-end QoS and to provide smooth video streaming. If some video frames are missing, then an SVC with EC scheme is used to estimate and recover the missing video frames. This scheme is executed on the base layer to maintain a minimum level of QoE.

Another inter-layer EC method is proposed in [61]. Missing video frames are identified based on the sequence number in the packet header information. The missing frames are concealed by computing a motion vector extrapolation based on two consecutively received frames.

2.4.3 Summary

Table 2.4 summarizes the different video coding techniques discussed in this section. There are basically two types of video compression techniques – multiple descriptions and layered video coding, both of them encode a video sequence in a way that multiple levels of quality can be obtained depending on the parts that are received. In LC, although many studies adopt only a single layer, SVC has been widely used offering special protection for the base layer. MDC has been explored considering the diversity of systems where each description has an equal probability of decoding. In addition, some studies have proposed a combination of MDC and SVC.

While EC techniques can improve video coding robustness, they can also increase video compression complexity, generate additional data (increasing bandwidth requirements for video transmission), worsen network congestion, and increase processing delays [113]. Thus, while the literature has contemplated different combinations of compression and error correction techniques, there are still many challenges to be addressed in the context of multipath video transmission.

Table 2.4: Summary of prior work using video coding techniques classified as video compression and error correction

References	**Compression**				**Error Correction**	
	SL	**SVC**	**MDC**	**Standard**	**ER**	**EC**
[1, 100, 4, 16, 80, 57, 27, 106, 111, 41]	✓			H.264/AVC		
[2, 128]	✓			H.264/AVC	✓	
[129, 95, 94, 43, 42, 76]		✓		H.264/SVC		
[35]		✓		H.264/SVC	✓	
[113, 12, 13]		✓		H.265/SHVC		✓
[14, 123, 29, 3]			✓	H.263-		
[49, 39, 82, 90, 64]			✓	H.264/AVC		
[119, 124]			✓	H.264/AVC	✓	
[34]			✓	H.264/SVC		
[101]		✓	✓	H.264/SVC		
[116, 48]		✓	✓	H.264/SVC	✓	✓
[117]		✓		H.264/RSVTS		
[61]		✓		H.265/SHIPVC		✓

2.5 Multipath Routing

Multipath routing protocols for wireless video transmission are surveyed in [104, 53, 6, 30]. They are classified based on the reliability requirement and QoS constraints of multimedia applications. Reliability has emerged as an important aspect of multipath routing protocols motivated by various multimedia applications, notably services that require reliable monitoring. Existing multipath routing protocols that seek to satisfy reliability requirements include Multipath Multi-SPEED (MMSPEED) [44], Reliable Information Forwarding Multiple Paths (ReInForM) [38], and Network Coding-Reliable Multipath Routing (NC-RMR) [126], while the main protocols that satisfy multipath QoS constraints are Sequential Assignment Routing (SAR) [107] and Stateless Protocol for Real-time Communications in Sensor Networks, called SPEED [55].

More recent studies propose extensions to traditional wireless ad-hoc routing approaches that incorporate multipath mechanisms for video transmission [101, 76, 57, 64, 90].

Considering the state-of-the-art in multipath routing for wireless video transmission, we classify multipath routing protocols based on their fundamental building blocks, which include their route discovery mechanism, routing metric(s), and path selection strategy. It is worth noting that, in most existing routing protocols, these basic functions are usually intertwined and not easily decoupled. One of the goals of our classification is to disentangle them in order to better understand the fundamental differences between proposed existing routing approaches and how they can be improved. Figure 2.7 illustrates the proposed classification of multipath routing building blocks for wireless video transmission aiming at IoT applications.

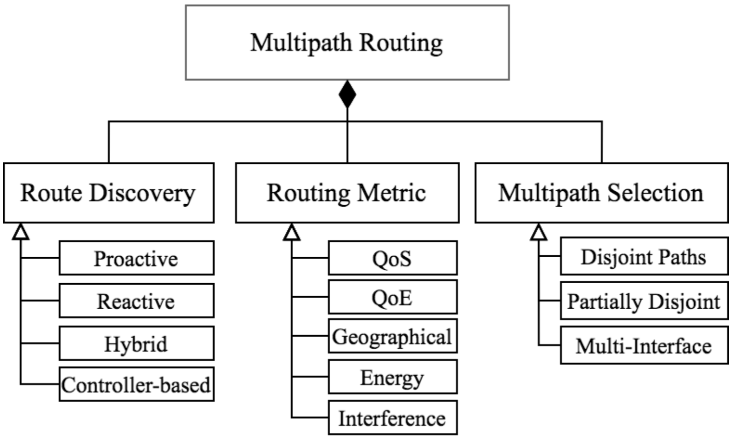


Figure 2.7: Multipath routing classification for wireless video transmission in IoT scenarios.

2.5.1 Route Discovery

Route discovery is the first phase of the routing protocol in which a source node tries to find the available paths to a specific destination. In this procedure, each node performs neighbor discovery by exchanging control messages between them. In the same way, nodes also perform the route maintenance by identifying broken paths that have been found. Our literature review indicates that existing multipath routing approaches have not proposed novel route discovery and maintenance mechanisms, in particular techniques that consider application-specific requirements for multipath video transmission. In general, variants of traditional routing protocols have been adapted to provide routing over multiple paths.

Route discovery mechanisms are traditionally classified as Reactive, Proactive, or Hybrid. In this taxonomy, we also consider controller-based approaches in which a controller provides core networking functions, including route discovery. These approaches were inspired and motivated by the Software Defined Networking (SDN) paradigm [33, 111, 12, 13].

2.5.1.1 Proactive

In proactive routing approaches, routes for every destination are proactively discovered so that, when a source has data to send to a destination, a route has already been discovered. As such, routes are periodically updated, e.g., by periodically updating routing tables in each node. Topology updates are usually gathered by means of control messages exchanged periodically between nodes. The bandwidth overhead generated by topology updates can be reduced in application scenarios that consider fixed monitoring cameras because topology changes are less frequent. For example, in video surveillance applications nodes are typically stationary and powered by continuous power sources. Therefore, video surveillance applications have used proactive routing approaches since they yield reduced packet delays [80].

Some studies proposed proactive multipath routing in order to improve QoS for video applications [14, 123, 57, 101]. A multipath extension to the traditional Destination Sequenced Distance Vector (DSDV) is evaluated in [101]. In DSDV, each node maintains a routing table listing all the other nodes they have known either directly or through some neighbors. The proposal extends DSDV by adding new routing table fields to store information about multiple disjoint paths. This mechanism demonstrated good performance for video transmission in terms of loss rate and network load for large networks.

Based on the original Optimized Link State Routing (OLSR) protocol, a QoS multipath routing approach for wireless video transmission was proposed in [57]. OLSR constitutes a more organized and efficient way to manage routes between nodes. It performs a shortest path algorithm (e.g., Dijkstra) over its complete view of the topology. This extension modifies OLSR's route discovery to estimate link delay between nodes and then performs a multipath Dijkstra algorithm using delay — instead of hop count — to calculate multiple shortest paths.

2.5.1.2 Reactive

In reactive routing approaches, route discovery is triggered on-demand when a source has data to send, in this case, when the source wants to start video transmission. Typically, if the source does not have route information for the intended destination in its local routing table, it initiates the route discovery mechanism by sending out special messages, usually referred to as "route requests". Existing multipath video transmission approaches that use reactive routing have adopted extensions to the traditional single path using source routing algorithms or routing tables, such as Dynamic Source Routing (DSR) [60] and Ad hoc On-Demand Distance Vector (AODV) [93], seeking to improve video transmission QoS.

In source routing, the entire path from source to destination is maintained at the source node, which includes path information in the data packet headers. Despite higher header overhead, this mechanism allows each source node to select and control the routes used in forwarding its packets. An evaluation of a QoS-aware multipath extension to the DSR protocol, presented in [64], shows good throughput performance for high data rate video traffic. Also based on DSR, an adaptive-multipath protocol is proposed in [3]. It aims to improve end-to-end performance of video services providing dynamic self-configuration depending on the state of the network, which is used by the source nodes to make a proper multipath selection. In the same way, [100] proposes a path selection mechanism for the Split Multipath Routing (SMR) source routing protocol, reducing the frequency of route discovery processes and control message overhead.

In routing table-driven approaches, nodes maintain a routing table with the next hop to each destination. Most studies focusing on multipath video transmission adopt the Ad hoc On-demand Multipath Distance Vector (AOMDV) routing algorithm [90, 76, 39, 2, 41, 128]. It is an extension of the traditional AODV [85] protocol for finding multiple, loop-free and link disjoint paths.

There are other AODV extensions proposed for multipath video transmission. One of these extensions is evaluated in [64] and incorporates a multi-criteria decision approach which is shown to outperform video transmission using the original AOMDV in terms of throughput, delay, and reliable delivery.

Another multipath variant of AODV that aims at improving video transmission QoS is evaluated in [101]. It demonstrates the advantages of reactive routing for video transmission in terms of throughput and network load in scenarios subject to high mobility.

2.5.1.3 Hybrid

Hybrid route discovery tries to combine the advantages of both proactive- and reactive routing. A hybrid real-time video stream routing protocol is proposed in [4]. It divides the complete network into coronas with the data sink at the center. Proactive route discovery maintains a corona identification for each candidate node based on its distance to the sink, and then a reactive mechanism computes hop-by-hop the optimal forwarding choice from a source that has high corona identification to the zero corona that is the sink node.

Motivated by achieving an adequate balance between scalability and efficiency, in [108, 10, 11, 65], a proactive mechanism periodically assigns the node geographic location to its directly connected neighbors, and then each node decides on-demand which paths should be considered when forwarding packets.

2.5.1.4 Controller-Based

SDN [68] has been proposed to decouple the network control plane from the data plane. According to the SDN paradigm, a controller centralizes routing decisions, while network nodes perform only forwarding. Several studies for multipath video transmission follow this paradigm, centralizing route information at the controller which then computes routes and adds corresponding routing information to routing tables at the forwarding network elements. As the controller has a global view of the network topology and conditions, routing decisions can be simplified. In the context of multipath video transmission, there are proposals where links that do not have enough bandwidth for real-time video streams are not considered when computing routes [111], as well as efforts that periodically monitor the quality of links for video streaming [33].

In [12, 13], an approach that combines controller assistance and source routing is used for multipath video transmission in SDN-based networks. This approach implements a source routing using a segment routing paradigm in which the source chooses a path and encodes it in the packet header as an ordered list of segments. It increases routing efficiency by improving the capability of selecting paths and thus speeds up video transmission.

While video transmission approaches that leverage controller assistance in SDN-based networks can benefit from SDN's network control flexibility and efficiency, there are still open issues on how the route discovery should be designed to consider the interplay with video applications.

2.5.2 Routing Metric

Routing metrics adopted for multipath selection are essential to improve video transmission. Our taxonomy considers the most commonly found metrics for multipath video transmission. However, these classes are not mutually exclusive since several studies integrate multiple metrics into a single solution. These metrics are described as follows.

2.5.2.1 Quality of Service

This is the main routing metric adopted in almost all multipath video transmission proposals due to the video streaming constraints. There are different QoS measures concerning aspects of delay, throughput, and packet loss associated with application requirements. As such, some studies [1, 54] have presented metrics that try to balance several QoS parameters seeking high throughput, low delay, and proper packet delivery. On the other hand, some studies have considered the hop count and geographic distance as factors strongly correlated to QoS [108, 56, 61].

End-to-end delay is also usually adopted because the delay constraint is precisely necessary for critical IoT video applications, as is the case of health and surveillance systems. In [57, 29, 100], end-to-end delay of paths are simply measured by the sum of hop delays, without considering other factors such as link congestion that comes from multiple video sources.

Although QoS is the main metric used in studies for multipath video transmission, care has to be taken as optimizing one QoS metric can compromise performance in other aspects. For instance, minimizing end-to-end delay by means of increasing transmission power may consume more energy.

2.5.2.2 Quality of Experience

This is a metric adopted to evaluate and present subjective study results. However, some studies have also used QoE estimates based on QoS parameters as routing metrics [49, 12, 13, 113, 94, 95]. Hence, these studies present a QoE model to estimate the Mean Opinion Score (MOS) considering the bitrate of the transmitted video and other QoS parameters. MOS is an indicator of QoE that can be used to assess video quality, which is divided into five levels corresponding to the users' perception. In this case, QoE-based protocols seek to maximize the MOS. It is a great routing metric for multipath video transmission, but it requires a lot of resources and cannot be obtained automatically [94].

To achieve perceptual quality evaluation in real-time, some studies have adopted the Pseudo-Subjective Quality Assessment (PSQA) [49], which is a tool based on statistical learning using Random Neural Network (RNN). The idea is to train the RNN to learn the mapping between QoE scores and QoS parameters. In this case, the route discovery and the multipath selection mechanisms can be aware of any episodes of QoE degradation.

2.5.2.3 Geographical

In these approaches, the video source maintains geographical information about its neighbors and the destination to select the best relays. Based on the geographical distances between each possible next-hop and the destination node, they perform an efficient route selection [16, 113, 10, 11, 7, 108]. Implicitly, they assume that the Euclidian distance between nodes is a good indicator of energy-efficient paths with good QoS. However, we cannot assume this for IoT application scenarios, especially the smart cities scenario where there are large communication blocking obstacles such as walls or buildings.

In order to optimize QoS with the minimum geographic distance transmission over the network, a multipath genetic algorithm is proposed in [61]. It seeks to maximize fault tolerance with minimum communication delay during video transmission. The fault tolerance of the path is estimated by the links quality estimation and neighbor distance. The geographical metric associated with fault tolerance showed improvement with different strategies of video coding and multipath routing. Although the authors affirm that it can be adjusted for a variety of related applications, they also do not consider obstacles in these application scenarios.

To provide delay and energy balance in video transmission, Hossain *et al.* [56] propose a multipath routing generation algorithm based on geographical information. It defines a model on the basis of traditional spline path generation planning to generate a set of paths uniformly distributed in multiple spaces between source and destination nodes. In fact, the algorithm generates paths with lower interference which is uniformly distributed. However, it depends on a highly dense network in addition to not being designed for multiple video sources.

2.5.2.4 Energy-Efficient

This routing metric is essential in IoT application scenarios where nodes are battery-powered. In this context, some studies have proposed this routing metric which aims to minimize the energy consumption of each node [128, 4, 2, 41]. They have recommended that a cost function considers the remaining energy to dynamically choose the video transmission power — and consequently, the communication range — of each node to maximize the network lifetime. Besides these, some studies have proposed a model to estimate the energy consumption on the basis of geographical distance [18, 7, 1, 10, 11, 61]. In [18], this is combined with an event detection approach where a minimum number of nodes remain awaken for transmission according to the event, while others get to sleep to maximize energy-efficiency. These models aim to minimize energy without considering video application requirements, such as end-to-end delay. The balance between energy and delay, for example, is fundamental for multipath video transmission [56].

Other studies attempt to balance energy usage of all nodes [56, 65]. They use the end-to-end delay required by video applications in their models. As these energy consumption models are based on simple geographical distance, they are also not indicated for IoT application scenarios, as mentioned before.

2.5.2.5 Interference

Video transmission may be degraded by the interference caused by network flows that share the same nodes or links. Node-disjoint multipath selection mechanisms are used as a simple solution in [4, 1], but they have topological constraints that may not be met in all IoT application scenarios. As an alternative, some proposals relax those constraints by allowing partially-disjoint paths with low levels of interference [10, 11, 108, 123, 16]. To this end, they estimate interference levels based on the Link Quality Indicator (LQI), SNR, or simply the distance between nodes.

A multicommodity network flows model which considers inter-flow interference is adopted in [94, 95, 96]. Based on this model, the optimization problem to maximize the quality of paths is formulated as a mixed-integer linear problem under link capacity constraint. The link capacity constraint is defined to optimize the number of video flows over links, however, it assumes flows with the same destination as only one commodity, regardless of their sources.

Another interference metric that aims to maximize the aggregated network throughput considering inter-flow interference is proposed in [20]. It is implemented using the Algorithmic Framework for Throughput EstimatoRs (AFTER) [91], an on-the-fly algorithm for real-time throughput estimate that considers a global view of the network. In this case, it proposes a multipath selection mechanism for wireless video surveillance systems assuming flows between multiple sources and a destination.

2.5.3 Multipath Selection

A multipath selection mechanism is used for choosing the best paths among the discovered routes. While the route discovery process finds possibly all available paths between source and destination, route selection chooses a subset of those to construct a multipath route.

In proactive route discovery, just one additional multipath selection mechanism must be added to the processing of routing update messages to obtain other possible paths at each node. In reactive route discovery, a multipath selection mechanism must be added to collect and store more than one path between the source and the destination. For this, it usually does not discard the duplicate of received route messages at a node to find additional paths.

In general, approaches found in the literature have considered the disjoint and partially-disjoint paths techniques for multipath selection. Furthermore, other proposals assume each node possesses multiple radio interfaces — often of different technologies — each of which provides a feasible path to the destination.

2.5.3.1 Disjoint Paths

This technique is prevalent in the literature, because of the independence and resilience of the paths it discovers. Two paths are said to be disjoint if they have no nodes or links in common. Several studies assume there are multiple node-disjoint paths available between each pair of nodes [108, 10, 11, 48, 16, 65, 56, 54, 123, 7, 113, 95, 100]. The existence of node-disjoint paths, however, is topology-dependent and, thus, such paths may not exist in all cases. Because of that, some studies have considered link-disjoint paths [100, 57], which are more likely to occur. Disjoint paths may be less affected by interference, but that does not necessarily guarantee the optimal path selection in terms of other performance criteria.

2.5.3.2 Partially Disjoint Paths

Path diversity can be explored if the concept of disjointness is made flexible by considering partially disjoint paths. Therefore, this technique is proposed in several studies [29, 94, 4, 18, 61] motivated by the fact that even link-disjoint paths may be unavailable when nodes are deployed randomly. Further, it can optimize multipath selection evaluating all available paths according to routing metrics, without the strict disjointedness constraint [14, 33, 1]. Another approach of partially disjoint paths is the idea of braided paths, where intermediate nodes process the traffic to select the next nodes that create better multipath [126]. It uses the disjoint paths technique to build actual paths, which are called main paths. Some braided paths on each main path are built according to a braided multipath algorithm.

2.5.3.3 Multi-Interface

Another approach to multipath transmission is to use multiple radio interfaces such as cellular and wi-fi networks. It can simplify the multipath selection mechanism by requiring just evaluating the interfaces that meet the routing metrics to select the better paths. For this, the studies [27, 12, 13, 124, 43, 42, 106] have proposed the use of the Multipath Transmission Control Protocol (MPTCP) [45]) transport protocol. MPTCP provides support for simultaneous multipath transmission, multiplexing the data transmitted by each one. Thus, a multipath selection mechanism can be implemented to evaluate the paths of each radio interface and then perform multipath allocations to the video sub-streams. However, it only seems suitable for scenarios with client-server communication in fixed topologies. In addition, for live streaming services, it is necessary to evaluate the path asymmetry in different access networks and the disadvantages of the data retransmission mechanism in MPTCP.

2.5.4 Summary

The literature has contemplated several multipath routing mechanisms, spanning different aspects of routing. However, no consensus solution has emerged thus far, as multiple issues remain open. Table 2.5 summarizes the different multipath routing proposals found in the literature according to the three components discussed in this section: route discovery mechanism, routing metric, and multipath selection.

2.6 Multipath Forwarding

A multipath forwarding mechanism determines the packet forwarding strategy of the video streams over the multiple paths selected by routing. At the most basic level, strategies can be divided into concurrent and non-concurrent multipath forwarding, as illustrated in

Table 2.5: Summary of prior work according to their approach to multipath routing.

References	Routing Discovery	Multipath Selection	Routing Metric
[57, 101]	Proactive	Disjoint Paths	QoS
[29]	Proactive	Partially Disjoint	
[27, 43, 42, 106]	Proactive	Multi-Interface	
[100, 101, 64, 90, 3, 48]	Reactive	Disjoint Paths	
[124]	Hybrid	Multi-Interface	
[33, 111]	Controller-based	Partially Disjoint	
[12, 13]	Controller-based	Multi-Interface	QoE
[18]	Proactive	Partially Disjoint	QoS, Energy
[2, 128, 41, 54, 48]	Reactive	Disjoint Paths	
[65]	Hybrid	Disjoint Paths	
[20]	Proactive	Partially Disjoint	QoS, Interference
[123]	Reactive	Disjoint Paths	
[95, 96, 94]	Proactive	Partially Disjoint	QoE, Interference
[108, 10, 11]	Hybrid	Disjoint Paths	QoS, Geographical, Interference, Energy
[56, 7]	Reactive	Disjoint Paths	QoS, Geographical, Energy
[61]	Reactive	Partially Disjoint	
[1]	Reactive	Partially Disjoint	QoS, Energy
[4]	Hybrid	Partially Disjoint	
[16]	Reactive	Disjoint Paths	QoS, QoE, Geographical, Interference
[113]	Reactive	Disjoint Paths	QoS, QoE, Geographical,Energy

Figure 2.8. Within those categories, however, proposals found in the literature can differ quite a bit, originating several specializations.

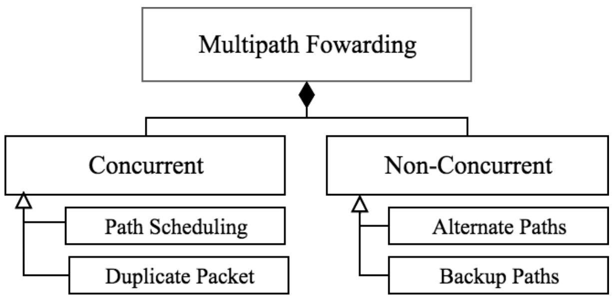


Figure 2.8: Classification of multipath forwarding strategies.

2.6.1 Concurrent Multipath Forwarding

Concurrent forwarding is the main strategy used to improve reliability. It implements video transmission using multiple paths simultaneously. In general, concurrent forwarding approaches propose cross-layer designs to integrate video coding at the application layer with packet transmissions between the network and physical layers. The video coding is responsible for performing packet segmentation or generates duplicated packets according to concurrent forwarding approaches. As illustrated in the taxonomy presented in Figure 2.8, in this work we classify the approaches for concurrent forwarding into Path Scheduling and Duplicate Packet.

2.6.1.1 Path Scheduling

In this concurrent forwarding approach, the packets can be scheduled for forwarding according to their priority, path capacity or schedule policies.

In packet priority, packets are identified at the application layer that communicates with the network layer to define the forwarding according to their priority. Proposals usually define path schedules according to their cost — as specified by some routing metric — and packet importance [18, 65, 56, 54, 123, 27, 3]. Data packets also can be classified as less important than video packets [41]. In addition, video packets can be prioritized according to the type of frames or layers generated by video coding at the application layer. In this case, some studies have proposed a cross-layer design that checks the type of each packet and the current network conditions to ensure the transmission of the prioritized packets [7, 35, 4, 10, 11, 94, 125]. In general, these studies assume a fixed amount of a maximum of three types of packets, regardless of the capacity of the paths. However, it can be impractical when there are not enough paths for all types of packets.

When the capacity of the main path alone is not enough to forward the whole video stream, the packets can be decomposed so that each flow matches the capacity of the available paths. Thet *et al.* [111] present a splitting method for video streaming over multiple concurrent paths to decompose the packet rate to match the capacity of available paths. This method can be applied in multipath video transmission; however, it is not recommended when the main path capacity already suffices to forward the whole video stream. In addition, it does not consider that path capacity can vary over time.

2.6.1.2 Duplicate Packet

A simple approach to increase the reliability of video delivery is to duplicate video frame packets to forward over different paths. Hence, copies of the same packet are transmitted over selected paths [48, 119]. However, this creates redundant packets that can occupy useful bandwidth. Moreover, to generate duplicate packets at the source and to filter out these duplicate packets at the destination, a special arrangement is required. For example, two agents are used in [119]: one to generate duplicate packets at the transmitter and another to filter received duplicates. Although it provides error-resilient video transmission, it can overload the network.

2.6.2 Non-Concurrent Multipath Forwarding

This strategy has been presented in the literature to improve the reliability of video transmission using alternative paths. As illustrated in the taxonomy of Figure 2.8, studies have proposed the use of backup paths and alternate paths as non-concurrent forwarding approaches. These approaches seek to provide fault-tolerance and load-balancing, respectively.

2.6.2.1 Backup Paths

It is one of the most popular multipath forwarding techniques to provide fault tolerance. In general, the idea is to use only a path for packet forwarding, while keeping some alternative paths ready to use in case of necessity. For example, if a path fails because of a broken link, then packets can be retransmitted through another path. In general, fault tolerance approaches select two paths so that whenever the primary path fails, the transmission falls back to the secondary path [61, 16, 33].

2.6.2.2 Alternate Paths

This strategy exploits multiple paths to achieve a better distribution of traffic and to provide load-balancing. Some studies have used these strategies to balance the energy consumption among network nodes [18, 90]. They propose different models that perceive the path load and node energy consumption and, accordingly, control the packet forwarding.

A weighted round-robin scheduling policy strategy can be a simple and effective alternative to distribute video traffic. This strategy was implemented in [100, 90] in which the traffic flows are forwarded proportionally over alternate paths. Furthermore, Wu *et al.* [124] propose a multipath forwarding mechanism that implements a packet distribution based on the path quality to adjust video traffic load and minimize total distortion.

2.6.3 Summary

Table 2.6 summarizes the different multipath forwarding strategies found in the literature in order to optimize the multipath video transmission.

Multipath forwarding is one of the mechanisms that presents the most promising strategies for multipath video transmission. The literature has contemplated interesting

Table 2.6: Multipath Classification of the proposals for Wireless Multipath Video transmission found on the literature according to the characteristics of their forwarding mechanisms found on the literature.

References	**Multipath Forwarding**	**Mechanism**
[111, 35, 95, 41, 94] [123, 56, 10, 108, 7, 90, 54, 11] [27, 65, 4, 18, 3]	Concurrent	Path Scheduling
[119, 48, 96]	Concurrent	Duplicate Packet
[61, 16, 33]	Non-Concurrent	Backup Paths
[100, 18, 90, 124]	Non-Concurrent	Alternate Paths

alternatives, mainly with regard to concurrent paths, which is the focus of this study. In this case, we can notice most studies have focused on path scheduling, where there are still many challenges to implementing an ideal mechanism to forward video packets over multipath while still considering video coding techniques from IoT application aspects.

2.7 Discussion

Wireless multipath video transmission targeting IoT applications has attracted considerable attention from the research and practitioner communities with the goal of improving the quality of service and reliability of video services. This chapter presented a thorough review of the current state-of-the-art in multipath video transmission for IoT applications such as Smart City, Industrial IoT, Telehealth, etc. To this end, we introduced a full taxonomy (see Figure 2.9) that explores the synergy between IoT applications, their features and requirements, and core networking services such as routing and forwarding. We used our classification to discuss a range of wireless multipath video transmission approaches for IoT applications.

The development of efficient applications for wireless multipath video transmission continues to be an open research area. By highlighting the interplay between IoT application features and requirements and the underlying network, we identified open research opportunities in exploring cross-layer support for IoT video services, for instance, exploring new multipath forwarding strategies using cross-layer information (e.g., video coding information) and proposing new routing metrics for multipath selection. Thus, we choose the six most relevant mechanisms as the state-of-the-art, namely: Error Resilient Video Transmission (ERVT) [48], Cross-Layer Multipath Routing (CLMR) [1], Real-Time Video streaming routing Protocol (RTVP) [4], QoS-oriented Multipath Multimedia Transmission Planning (Q-MMTP) [56], QoE-aware Sub-Optimal routing (QSOpt) [94] and ILS-

MDC [20]. They represent different multipath selection strategies and metrics for our evaluation study, as will be discussed in more detail in Section 5.4.3.

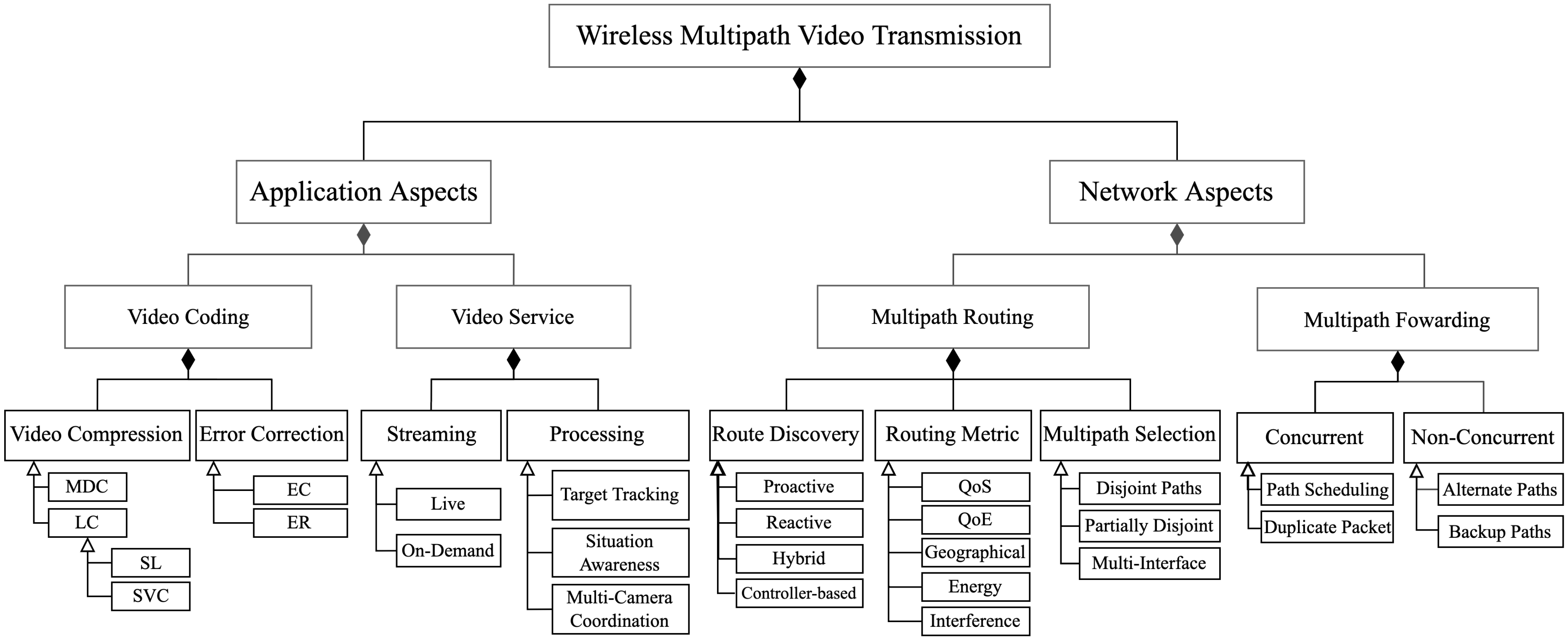


Figure 2.9: Full wireless multipath video transmission taxonomy for IoT applications.

Chapter 3

Multipath Selection Framework for IoT Video Transmission

The central hypothesis that underlines this thesis is that a cross-layer routing framework integrating video coding with multipath routing mechanisms can improve the user's QoE in IoT application scenarios with multiple video sources. Moreover, information of video flows requirements as well as the network status provided by topology monitoring can be used to estimate the network performance of the video transmission in real time to make the routing decisions. Hence, a heuristic-based algorithm can use a global view of the network to select good paths considering inter-flow interference. Thus, multiple video flows could be forwarded over multiple paths, meeting video services requirements and network dynamics as well as improving the QoE.

The proposed solution consists of an efficient multipath selection mechanism for IoT video transmission that provides multi-layer metrics for better routing decisions enhancing the network performance and the users' QoE. A real time performance estimator is incorporated in the multipath selection mechanism to evaluate candidate paths. It works as a routing metric to select multiple paths that maximize QoE. While the application coordinates video flows requirements, a network topology monitoring module evaluates the links' state. The quality of links is used by the mechanism to find candidate paths that meet flows' requirements. Hence, both the network conditions and the video services demands are jointly evaluated to select multiple paths for concurrent video flows.

The architecture raised by the proposed framework is described in Section 3.1. The remaining sections provide an overview of each component of the proposed framework.

3.1 Proposed Architectures

As discussed in Chapter 1, the proposed solution focuses on IoT applications scenarios such as surveillance systems in smart cities which require a multipath selection mechanism for forwarding multiple videos flows simultaneously. In these scenarios, all network nodes can work as video sources as well as relay nodes and are typically stationary, powered by continuous energy sources, and equipped with sufficient storage and computing power. As such, frequent topology changes caused by node mobility, energy depletion, etc. are not expected to play a significant role. In scenarios where topology changes need to be considered, information about network conditions can be obtained using different mechanisms depending on the type of network control plane architecture as discussed below.

3.1.1 Decentralized Network Control

Figure 3.1 illustrates an architecture under decentralized network control, e.g., multi-hop wireless ad-hoc networks using distributed routing protocols. In these scenarios, each node maintains a complete view of the network topology utilizing topology state monitoring and dissemination mechanisms similar to those implemented by proactive link-state routing, e.g., OLSR [32], which also update topology information periodically and whenever nodes detect "significant" changes, e.g., link failures, new nodes/links, changes in link quality, etc. Current topology state information is fed as input to proposed mechanism.

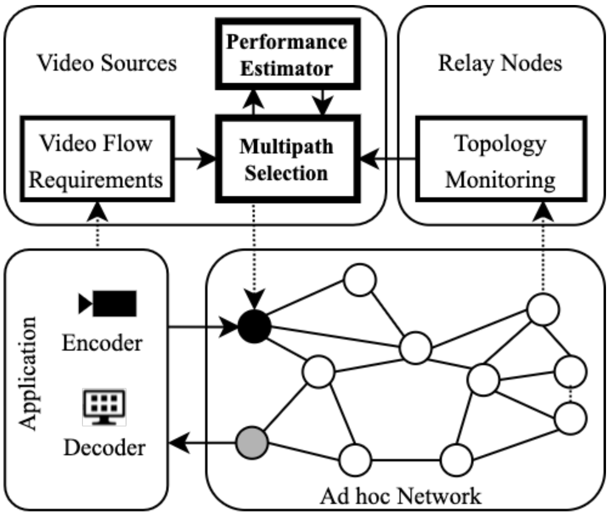


Figure 3.1: Decentralized network control architecture.

At each video source, the mechanism makes routing decisions based on the current network state obtained from the topology monitoring module. It finds a set of paths over which required video flows can be transmitted. The video source is then responsible for forwarding the flows through the selected paths. In order to avoid routing loops, a simple solution is to implement a source routing technique in which the source includes complete path information in the packet header.

3.1.2 Centralized Network Control

The proposed mechanism can also be deployed under a centralized network control plane *a la* SDN [68] as shown in Figure 3.2. The SDN controller makes routing decisions based on its global view of the network and informed by the multipath selection mechanism, while network nodes perform data forwarding based on the routing rules they receive from the controller.

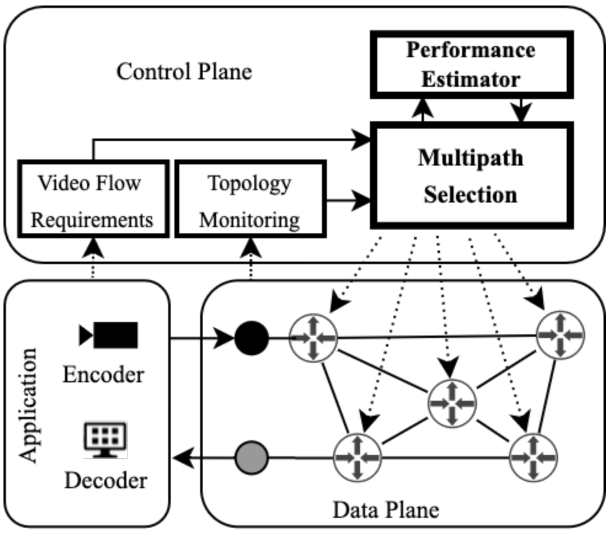


Figure 3.2: Centralized network control architecture.

Note that in centralized network control environments, the mechanism will be used to assist the controller in its routing decisions based on current network topology knowledge and video traffic requirement information. The controller then updates the forwarding nodes' routing tables to forward flows according to the selected paths.

3.2 Topology Monitoring

Topology monitoring keeps a snapshot of the network with necessary information for multipath selection. This snapshot is updated whenever a node identifies a link failure, a link cost change, or a new link available. In this proposal, The link cost is represented by the Expected Transmission Count (ETX) metric. Each node periodically broadcasts probes to measure ETX by using the delivery ratio of probes sent on the forward and reverse directions. With this, the cost of a path is obtained by the sum of the costs of its links which correlates well with the path throughput [91, 37].

As illustrated in Figures 3.1 and 3.2, the deployment of topology monitoring depends on the type of network control plane. In decentralized networks, it can be implemented as a link-state monitoring based on a proactive link-state algorithm, which uses probe and control packets to discover and then disseminate link-state information throughout the network. In this case, it is executed at each network node measuring costs of each node's links to its neighbors. Thus, the link-state monitoring disseminates this information within the network state, which is used by a multipath selection mechanism at each video source. In a centralized network, this task is relatively simpler since the topology monitoring can be installed into the controller which receives direct information from each node. Hence, this component maintains a global view of all link costs to update the multipath selection mechanism. Since the network state information and video requirements from all sources need to be synchronized for the selection of paths, decentralized and centralized network control deployments present different impacts on routing performance due to overhead and lack of synchronism.

3.3 Video Flow Requirements

Considering the different video compression techniques presented in Chapter 2, there are several open issues, such as what is the ideal technique for adaptive video coding: MDC, LC, or a combination of them, and what is the ideal number of video partitions (layers or descriptions). However, answering those questions in general is outside the scope of this thesis. Instead, we simply focus on evaluating which technique is appropriated for the proposed solution.

To inform the multipath selection mechanism about the number of flows and their respective rates, the video requirements component works similarly to the topology monitoring. Although video flows are generated at variable bitrates, this component can inform the bitrate targeted by video encoders at the application. Just like the usual link-state information, changes in the transmission status of flows trigger an update whenever a video transmission starts, updates or stops.

According to the dynamics of the IoT application scenarios, the video sources can change their transmission state. For example, cameras can be enabled/disabled or have their resolution changed depending on the video services. Thus, the multipath forwarding mechanism needs to be adaptive following the application demands. In this way, this proposed solution intends to integrate this mechanism with multipath selection to update the paths whenever the transmission state changes, as follows:

- Transmission starts: Video sources start transmission by shortest paths while running multipath selection mechanism searching for better solutions;
- Transmission updates: Video coding can be adapted — updating the number of flows and bitrate — based on the video service or network condition. In this case, the video requirements trigger the paths update; and
- Transmission stop: Similar to the transmission updates, the selection mechanism needs to be aware of state changes of the sources. Thus, it must also identify this interruption to synchronize the video requirements information.

The proposed solution implements multipath routing strategies that select paths estimating current available network capacity as well as deciding whether the network can fulfill each flow's requirements. In addition, video sources could use feedback of the selected paths to help video compression perform the rate adaptation, seeking to optimize the QoE. Furthermore, adaptive video coding could control the average bitrate and keyframe interval so that the resultant bitrate is adequate to the throughput of the selected paths. Notice that this throughput may vary dynamically due, for example, to the inter-flow interference.

3.4 Performance Estimator

Estimating network performance is an effective mechanism to address the challenges raised by multimedia traffic as a way to achieve QoS-aware admission control, resource provisioning and allocation in multihop wireless networks [97]. Accurate multimedia performance estimates are useful for routing and video coding decisions [24, 53, 122]. Thus, we propose the MAPE[23] mechanism, which estimates, in real time, network performance for multi-rate multimedia flows using their video coding rate as input.

As will be discussed in more detail in Chapter 4, the FITPATH multipath selection mechanism invokes MAPE with an up-to-date network snapshot and video flow requirements as input. Then, based on MAPE's performance estimates, it performs path selection accordingly. MAPE provides per-flow throughput, packet loss rate and end-to-end delay estimates, which can be correlated to define a function as a routing metric that maximizes the user's QoE. Hence, MAPE can guide route selection and, as a result, improve video transmission quality.

3.5 Multipath Selection

The multipath selection mechanism is the core component of this thesis. Aiming at providing an efficient and adaptable path selection scheme for multipath routing, we propose a mechanism called FITPATH [22]. Once invoked, FITPATH is responsible for finding a better path for each video flow required and providing multipath forwarding. It must occur periodically or whenever the components of topology monitoring and the video flow requirements change, according to application and network dynamics.

As will be discussed in more detail in Chapter 5, FITPATH is able to accommodate typical IoT applications scenarios where multiple video sources can simultaneously transmit heterogeneous video flows, i.e., flows of different bitrates. FITPATH addresses these issues by using an efficient algorithm [19, 21] to: (1) find a set of candidate paths and (2) given the requirements of the flows to be transmitted by the different video sources (e.g., number of flows, flow bitrates), rank the candidate paths based on MAPE's performance estimates.

Chapter 4

MAPE: Multimedia-Aware Performance Estimator

There is a wide variety of IoT multimedia applications that can benefit from a real-time network performance estimator to route selection [24]. As will be discussed in more detail in Section 4.1, different performance estimators have been proposed but do not fulfill the needs of IoT multimedia applications which require estimators to account for multi-rate flows as well as inter-flow interference, while being able to provide their estimates in a timely and resource-efficient manner.

In this chapter, we introduce MAPE, which estimates network performance for multi-rate multimedia flows using their video coding rate as input. To the best of our knowledge, MAPE is the first estimator that is able to provide throughput, packet loss and delay estimates in real time considering rate-heterogeneous flows and accounting for inter-flow interference.

Experiments using different IoT multimedia application scenarios demonstrate that MAPE is able to provide real time network performance estimates, i.e., throughput, delay, and packet loss, with savings of over two orders of magnitude in execution time when compared to the *ns-3* [98] network simulator.

4.1 Network Performance Estimators

IEEE 802.11 networks have offered several attractive rate-capable amendments that serve various multimedia application scenarios [53]. Providing performance estimates is critical to meet QoS guarantees in such networks. Existing approaches to network performance estimation in IEEE 802.11 networks can be classified in three main categories, namely:

- **Mathematical models** Estimators based on mathematical models typically make simplifying assumptions to make modeling tractable. For instance, most existing proposals target one-hop flows [118, 47, 81]. Moreover, they make additional simplifications, such as perfect links and identical transmission rates for all nodes. In the context of per-flow performance estimation, Laufer and Kleinrock [77] present a more complete model for analyzing the throughput of CSMA/CA networks. This model estimates the maximum throughput for each flow by modeling the network behavior as a system of non-linear equations and solving the resulting optimization problem. That approach can become prohibitively expensive for larger networks, as the size of the system of equations grows exponentially with the number of network nodes.

- **Online Estimators** While mathematical models for performance estimation are useful to understand the limits of contention-based medium access protocols, approaches that can be operated online are required in practice, e.g., for real-time applications such as adaptive video streaming [17, 59, 110, 121, 25, 62, 87, 122, 115] and routing protocols [26]. In particular, performance estimation for adaptive video streaming is discussed in [121, 62]. These studies also consider buffer occupancy information for predicting performance to improve video streaming quality of experience (QoE). The work reported in [115] proposes a method to reduce the impact of inaccurate throughput prediction on QoE by controlling the buffer occupancy within a safe range. In turn, routing metrics provide indirect information that is expected to correlate well with throughput [26], but they usually fail to evaluate the interference between flows.

- **Discrete-Event Simulators** Discrete event simulators can be stochastic or deterministic. Stochastic simulators use pseudo-random number generators to determine the outcomes of events that have some level of randomness (e.g., the choice of backoff intervals for medium access), while deterministic simulators replace pseudo-random generation with deterministic values (e.g., a fixed average backoff interval).

Network performance estimation performed by stochastic simulators like *ns-3* [98] and OMNET++ [114] is commonly used to either conduct an *a-priori* evaluation of a certain network and its protocols, guide network provisioning, deployment or operational tasks. Because of their random nature, they usually require a large enough number of runs for every experimental configuration in order to obtain statistically meaningful results, which adds to their inherent scalability limitations, long execution times, and high computational resource needs. On the other hand, deterministic estimators provide an adequate accuracy with identical results no matter how many times they run. However, they must be designed to perform in real-time while the network operates to help dynamically adjust operational parameters. For that, simplifications (e.g., replacing random network events for their expected outcome) are necessary to reach a steady state in a short period of time.

One notable example of this latter class of performance estimators is AFTER [91]. It was proposed to tackle the problem of real-time throughput estimation for multihop IEEE 802.11 networks. AFTER simulates the behavior of the link- and network layers to quickly converge to steady state behavior that allows it to estimate the long term average throughput of each flow for a given set of application flows and corresponding routes. To this end, it maintains in memory a complete view of the network topology and performs a deterministic simulation of the network dynamics, generating simulated virtual packets (*v_packets*) for each flow at their respective virtual source nodes, triggering a number of other relevant simulation events, such as wireless medium access, queue management (*v_packets* being added, removed and discarded from buffers) and, eventually, the delivery of *v_packets* to their virtual destination nodes. In particular, AFTER takes into account inter-flow interference, employing a set of deterministic rules to deal with nodes competing to access the wireless medium. However, AFTER cannot handle arbitrary traffic models because it seeks to estimate the maximum achievable network throughput by considering each flow to have an infinite backlog at the source. This means that AFTER provides no support for scenarios in which multimedia applications themselves limit the transmission rate of each flow.

Although a number of performance estimation approaches have been proposed, none of them is able to provide real-time performance estimates that account for both inter-flow interference and rate-heterogeneous flows. The proposed MAPE tries to fill this gap and uses a deterministic simulation-based approach to estimate the long term average throughput, packet loss and end-to-end delay for all (multi-rate) flows considering inter-flow interference. Note that considering the specific rate of each flow is essential to more realistically reproduce the behavior of multimedia applications. For instance, in video applications, transmission rates are determined by video coding at each source and, therefore, each flow can be generated at a lower or higher transmission rate than what is supported by the network.

4.2 Design and Operation

Algorithm 1 illustrates MAPE's overall operation, which is divided in three steps: **Step 1 -** MAPE starts with a complete snapshot of the current network state as input consisting of a representation of the network topology that includes link quality estimates (i.e., link frame delivery probability), list of currently active flows along with the respective paths, and each flow's data rate; **Step 2 -** MAPE then uses the initial network snapshot to simulate the network as it operates until reaching steady state, which is used to compute long term throughput, packet loss, and end-to-end delay estimates in **Step 3**. Note that we employ the term *steady state* in the same sense as in [91], i.e., as a finite cycle of states that repeats themselves.

Algorithm 1: MAPE's pseudo-code.

```
{Step 1: Initialization}
networkTopology ←graph representing the network
flowsPath ← list of paths of all flows
flowsRate ← list of bitrate of all flows
{Step 2: Simulation}
while no Steady State do
    foreach flow f ∈ flowsPath do
        Update the number of v_packets received by flow f;
        Schedule the queuing of new packet of flow f to its queue according to
          flowsRate;
    end
    Next network state;
end
{Step 3: Estimation}
Compute long term per-flow performance.
```

At the end of each iteration, MAPE stores a snapshot of the current network state, which consists of currently ongoing transmissions with their respective remaining times, the content of the queues and the backoff counter of the wireless medium access for all nodes that are traversed by any flow on the evaluated flow set, and the current medium access priority list. To decide whether the steady state has been achieved, the current state is compared to all previous ones. Whenever a duplicate state is found, MAPE declares that steady state has been reached and computes the average throughput, packet loss rate and end-to-end delay for each flow. A heuristic stop criterion is also used to guarantee low execution time and adequate real-time performance independent of application scenarios. When duplicated states are not found, MAPE computes the average cycle performance of events within which at least one packet from each flow has been delivered to its final destination as an attempt to approximate steady state performance.

Unlike stochastic simulators that study network behavior over a predefined period of time, MAPE aims at estimating the performance of the network, e.g., throughput, packet loss, and end-to-end delay at steady-state. This can be especially useful for QoS provisioning and, as previously noted, for route selection in real-time multimedia applications. Additionally, as discussed in Section 4.1, deterministic simulators that assume rate-homogeneous flows may result in severely inaccurate estimates for a number of reasons. First of all, the performance of a flow is necessarily limited by its transmission rate. Thus, such simulators may grossly overestimate performance in scenarios where network capacity is much larger than the aggregate demand of the active flows. Furthermore, severe underestimates may also occur for individual flows because interfering flows may be transmitted at a higher rate, consuming more network resources than they would in reality, reducing the achievable performance of other flows. MAPE overcomes these limitations by explicitly accounting for both multi-rate flows and inter-flow interference and thus attains more accurate performance estimates in more realistic multimedia application scenarios.

While MAPE builds on "traditional" deterministic estimators such as AFTER [91], unlike these estimators, MAPE relaxes the assumption that all flows have infinite backlogs and instead generates *v_packets* according to the rate of each flow — which can be specified as an input, based on the flows' video coding rate, for instance. Whenever invoked, MAPE receives flow rate arguments as input and uses them deterministically to simulate the network dynamics by: (1) generating simulated *v_packets* for each flow at their respective source nodes, (2) triggering a number of other relevant simulation events, such as wireless medium access transmission, queue management (*v_packets* being added, removed and discarded from buffers), and, (3) eventually, delivering *v_packets* to their destination nodes. As such, inter-flow interference happens as a result of buffer overflow, link-layer transmission losses, and medium access conflicts.

4.3 Implementation

MAPE's current implementation uses AFTER [91] as the underlying deterministic performance estimator. As shown in Figure 4.1, MAPE starts by initializing the simulation state with its input arguments. In this phase, the first packet of each flow is added to the queue of the respective source node and the simulation time is initialized to keep track of the events that are used to generate scheduled *v_packets*. Thus, the main loop of the simulation starts with the advance of the simulation according to the time of next possible events. This loop also handles packet receptions and eventually generates new transmission events until it detects that the network has reached a steady state — a state when the same sequence of events starts to repeat itself — which informs MAPE that it can then compute the estimated performance of each flow.

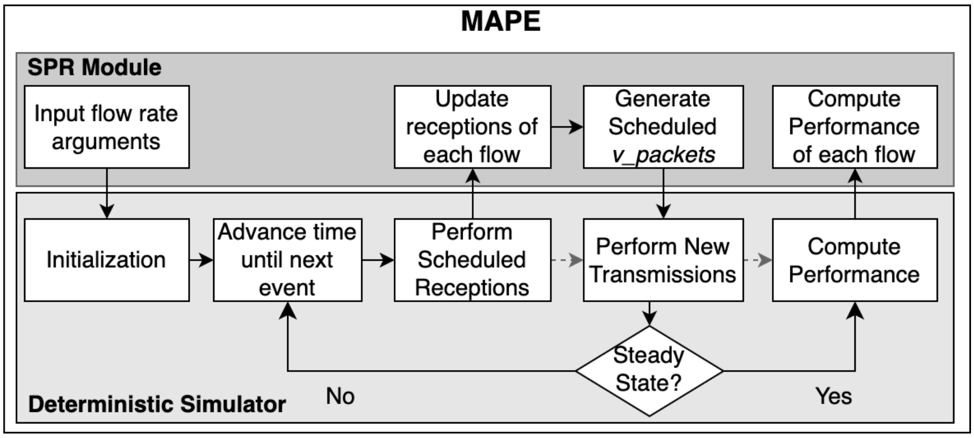


Figure 4.1: MAPE's implemention

MAPE's functionality is implemented as a module (called Specific Per-flow Rate (SPR)) that interfaces with the deterministic simulation engine to (1) provide flow rate information as part of simulation initialization, (2) update each flow when their *v_packets* are received, (3) generate new *v_packets* according to the stipulated flow rates, and (4) provide per-flow performance measurements.

MAPE uses a representation of the current simulation state, which includes information about all received *v_packets*. Furthermore, the SPR module implements a procedure to schedule the next packet generation for each flow according to the specified rate and keeps track of the number of *v_packets* received per flow, which is used to calculate performance estimates for each flow. The number of received *v_packets* is also used to determine the steady state cycle – i.e. the shortest sequence of network states that repeats itself indefinitely on the steady state of the simulation. While in the original AFTER the network state is updated when a *v_packet* from any flow arrives at the destination, in the SPR module the condition was modified to update it only when all flows receive at least one *v_packet*.

To simulate *v_packet* transmissions, MAPE starts by placing the initial *v_packet* of each flow on the queue of the respective source node. It then iterates through all nodes that have at least one *v_packet* on their queues and triggers events for dequeuing a *v_ packet* and adding this *v_ packet* to a transmission buffer, where the *v_ packet* is stored while waiting for an opportunity to be transmitted.

Note that MAPE's SPR Module introduces a mechanism to schedule the next *v_ packet* generation for each flow according to the specified rate. Once per-flow rates have been specified, the scheduler uses them to place new *v_ packets* in each source node's queue until the steady state is detected. Thus, *v_ packets* of each flow are generated according to the intervals of the simulation time. To keep track of the simulated time, MAPE uses a time variable that is updated according to the end time of a link layer transmission attempt and the backoff procedure.

Once the simulation reaches steady state, MAPE computes the average throughput of each flow as the ratio between the total number of *v_ packets* delivered within the last steady state cycle — i.e., the period between two repeating simulation events — and the steady state cycle length. In addition, the SPR Module computes packet loss and end-to-end delay by tracking all *v_ packets* from the instant when they are generated at their source nodes until they are received at their destinations. MAPE is then able to estimate the average per-flow packet loss rate and end-to-end delay. Such metrics account for the data link layer transmission attempts and queuing delays.

4.4 Assumptions

While MAPE makes assumptions about network events and convergence to steady state, our experimental evaluation (see Section 4.6) shows that MAPE is still able to estimate per-flow performance with adequate accuracy in *quasi* real time.

In its current implementation, MAPE assumes that flows are transmitted at constant bitrate. However, multimedia applications typically employ variable bitrate transmission. One way to address this is to simply have MAPE use the flow's average bitrate, which can be determined during transmission. Another approach to handle dynamic traffic patterns is to provide MAPE with updated data rate information whenever significant transmission rate changes are detected in the video coding process. In this work, we use the average bitrate of each video trace as input to MAPE. As part of future work, we plan to add support to variable bit rate flows.

4.5 Evaluation Methodology

We evaluated MAPE against two types of discrete event simulators, stochastic (*ns-3*) and deterministic (AFTER). We chose *ns-3* because it is widely used by the network researchers and practitioners since it provides an adequate model of the network and, thus, provides reliable estimates of network performance. We use AFTER as the example of a deterministic simulator and demonstrate that MAPE can achieve better accuracy by being able to model specific per-flow rates, i.e., it simulates each flow transmitting at specified multimedia bitrates.

In this section, we describe the experimental methodology we use to evaluate MAPE, including the topologies and traffic models considered, as well as how the experiments were carried out.

4.5.1 Experimental Topologies

We evaluate MAPE using two different IoT wireless network topologies akin of IoT scenarios and whose parameters are summarized in Table 4.1. The *Random Indoor* topology aims to replicate smart building scenarios and was generated by placing nodes randomly within an indoor environment. The *Grid Outdoor* topology tries to simulate smart city scenarios of neighborhood blocks and streets in an urban region, represented by a grid. More specifically, we reproduced a region of the city of Niterói, in the state of Rio de Janeiro, Brazil using an 8×7 grid of nodes in which two consecutive nodes are placed 60 m and 70 m apart on a given line and column of the grid, respectively, as illustrated in Figure 4.2.

Table 4.1: Simulation parameters.

Parameters	Topologies	
	Random Indoor	Grid Outdoor
Deployment area:	100 m x 100 m	360 m x 490 m
Number of nodes:	30	55
PHY/MAC technology:	802.11g	802.11g
Link speed:	18 Mb/s	18 Mb/s
Mac Queue Size:	10 p	10 p
Packet Lifetime in the Queue:	500 ms	500 ms
Traffic Control Queue Size:	1 p	1 p
Network Queue Size:	1 p	1 p
Propagation Model:	Shadowing	Cost231

For a fair comparison between all three simulators, we set up the same link speeds,

Figure 4.2: Scenario of *Grid Outdoor* topology simulating wireless video transmission in urban area.

queue sizes and packet lifetime policy on *ns-3*, AFTER and MAPE. The *Shadowing* and *Cost231* propagation models [109] were chosen to more realistically reproduce indoor and urban environments. All simulations use the same MAC and PHY technology and the same link speed, which was chosen to support multimedia application scenarios.

In order to estimate link quality (an information that is required by MAPE), a series of preliminary simulations were performed using the *ns-3* simulator. For all nodes in each topology, we executed a simulation transmitting $20,000$ packets to extract the long term quality of each link.

4.5.2 Traffic Models

In addition to link quality, MAPE requires per-flow transmission rate information. In our experiments, we use a mix of three different rates (as shown in Table 4.2) to represent different levels of video quality. The EvalVid framework [66] was used to generate traces of the same video clip with these three rates, and the resulting average bitrate of each video trace was used as input to MAPE. Additional traffic generation parameters and their values used in our simulation experiments are listed in Table 4.2.

Experiments which used multimedia (MM) traffic employ a publicly available and commonly used video clip, namely "Hall Monitor" [102], which was converted to H.264 format with a rate of 30 frames per second. Considering real-time transmission delay and human tolerance, the play-out buffer is set to 300 ms to mitigate potential out-of-order

Table 4.2: MM and CBR traffic parameters

Parameters	Values
MM-Traffic	
Video:	Hall Monitor
Encoding:	H.264/MPEG-4 AVC
Frame rate:	30 fps
Format:	YUV CIF, 352 x 288
Number of frames:	3600
Target Bitrate:	256, 512, 1024 kb/s
Packet size:	1024 $bytes$
CBR-Traffic	
Bitrate:	261, 485 and 836 kb/s
Packet size:	1024 $bytes$

packets; packets with delay longer than 300 ms are discarded at the decoder.

In video traffic, transmission rates may vary according to the coding technique used. For example, more important video frames (e.g. MPEG I frames) increase the instantaneous flow rate because the frame size is larger than less important frames (e.g. MPEG P frames and B frames). In our experiments, multimedia (MM) traffic target bitrates used by MAPE are based on long-term average bitrates calculated at the video source encoder. Because MAPE currently models variable bit rate flows using their long-term average rates, we also ran experiments with CBR traffic in our *ns-3* simulations in order to assess how short-term fluctuations of the video traffic bitrate affect MAPE's estimates. In those experiments, we adopt the same bitrates used as input for AFTER and MAPE as listed in Table 4.2, where 261, 485 and 836 kb/s represent CBR long-term average by targeting 256, 512, 1024 kb/s at video encoder, respectively. As part of our future work (see Section 6.2), we will modify MAPE's current variable bit rate traffic model to be able to account for shorter-term transmission rate variations.

4.5.3 Experiments

Simulation experiments were conducted as follows. For each topology, we computed the 5 best paths (based on the quality of their links) for 500 source-destination pairs generated randomly. Selecting one path for each pair, out of their 5 best, we generated random instances for scenarios with 3, 6, 9, and 12 pairs (or flows), which are used to transmit concurrent video flows with 3 different levels of quality – a third of the flows use each of the three transmission rates listed in Table 4.2. For instance, in a scenario with 6 flows, we have 2 sources transmitting at 256 kb/s, 2 sources transmitting at 512 kb/s, and 2 sources transmitting at 1024 kb/s. We left out the evaluations of scenarios with more than 12 flows because the networks become saturated. These scenarios do not provide satisfactory support for video applications, so they are not relevant for this work.

Finally, all scenarios were also executed in the *ns-3* simulator for both the CBR and MM traffic models using a simulation time of $120s$. For each scenario, execution time, per-flow throughput, end-to-end delay, and packet loss were computed by averaging results over all runs.

4.5.4 Evaluation Metrics

We evaluate MAPE's performance according to execution time and prediction accuracy. Since *ns-3* is a highly reliable and complete open source packet-level simulation platform, we are using its throughput, packet loss and end-to-end delay statistics as the baseline in our performance study. Throughput is calculated as the ratio between the number of packets delivered to the destination and simulation time. End-to-end delay is the time interval between when a packet is transmitted by the source node and when that packet is delivered at the destination, averaged over all packets received, and packet loss is calculated as the percentage of packets transmitted that were not delivered to the destination.

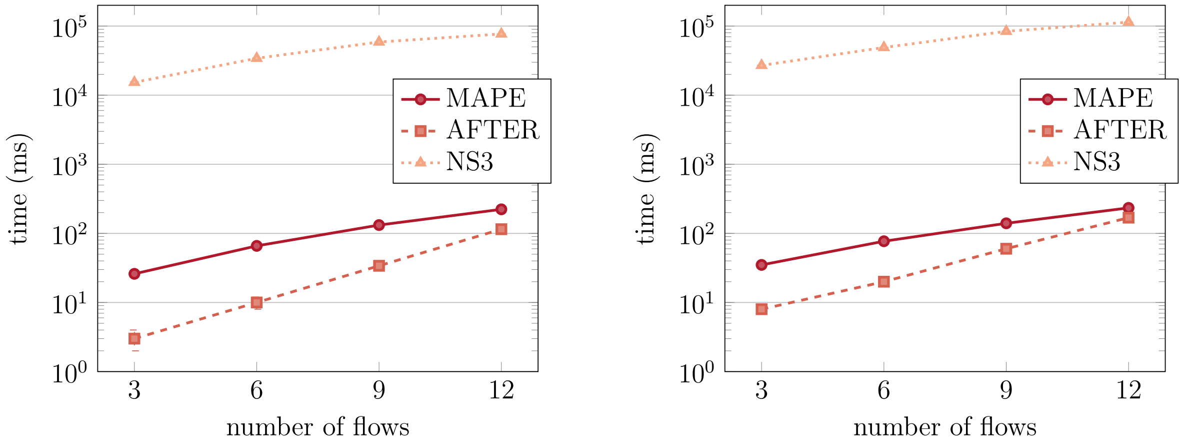
We expect MAPE to achieve predictions close to those of *ns-3*, but in a reproducible manner and at a fraction of the required execution time. We also use Structural Similarity Index Measure (SSIM) [120] and another metric called *classification inversions* — as defined in [91], and further explained in Subsection 4.7.2 — to evaluate video quality and demonstrate the practical suitability of MAPE to the problem of route selection for multimedia applications.

4.6 MAPE's Accuracy Evaluation

Our experimental evaluation aims at demonstrating MAPE's ability to accurately estimate per-flow throughput, delay and packet loss in a timely manner when compared to estimates provided by existing stochastic (*ns-3*) and deterministic (AFTER) simulators.

4.6.1 Execution Time

We measured average execution time for AFTER, MAPE, and *ns-3* for each scenario considering the 95% confidence intervals. All mean times are in milliseconds and simulations were performed on a dedicated server with an Intel i7-860 processor running at 2.8 GHz and 32 GB of RAM. As shown in Figure 4.3, MAPE and AFTER report execution times that are at least 2 orders of magnitude lower than those of *ns-3* for different scenarios.



(a) *Random Indoor* topology.

(b) *Grid Outdoor* topology.

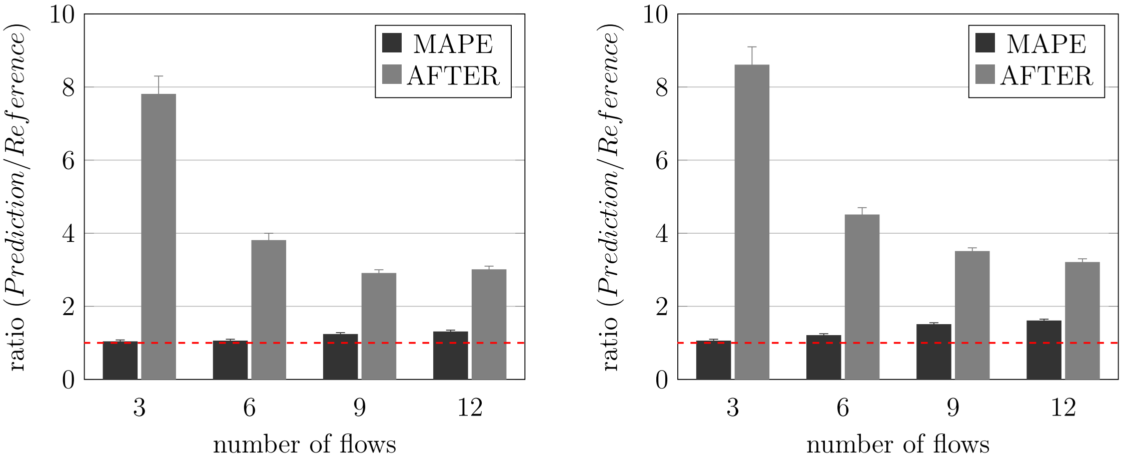
Figure 4.3: Execution time (log scale) for different numbers of flows.

Note that execution times for *ns-3* vary from tens to hundreds of seconds for the scenarios considered. While we observe a slight increase in MAPE's time complexity when compared to AFTER's for scenarios with only a few flows, that difference becomes negligible when the number of flows increases. It demonstrates that MAPE is able to compute per-flow network performance estimates in real time which can be used to inform core network services such as routing. MAPE and AFTER are fast because, unlike stochastic simulators, they do not need to simulate nearly as many events to reach steady state.

As expected, execution times increase with the number of flows. However, AFTER and MAPE's execution times increase more significantly with the number of path hops because that increases the number of transmission and reception events needed to deliver the flows' *v_packets* to the destination node. This explains the slightly higher times measured with the *Grid Outdoor* topology, which typically requires paths with more hops because of the greater distances between nodes.

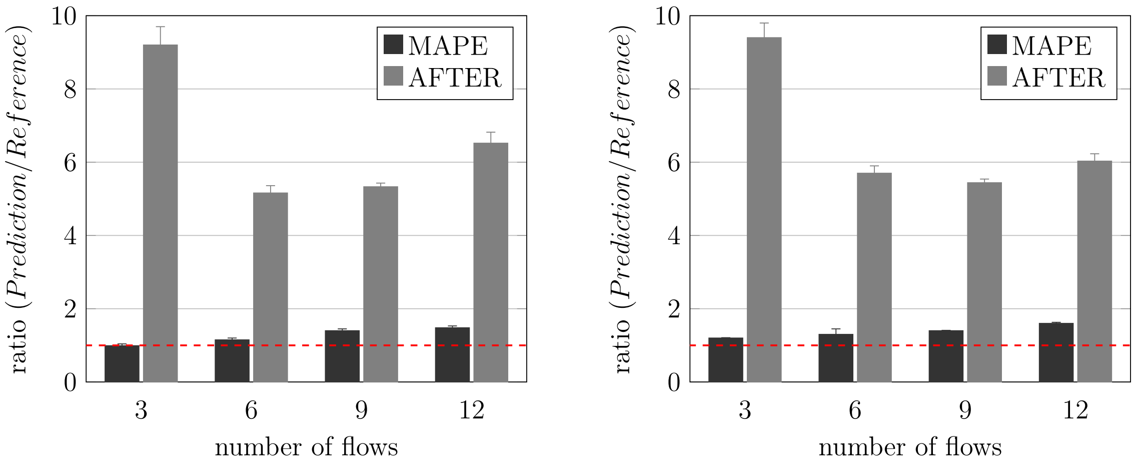
4.6.2 Estimated Throughput

We measure the throughput estimate accuracy as the mean ratio between the per-flow estimate returned by AFTER or MAPE and the per-flow throughput obtained by *ns-3*. Differently from other common ways to measure accuracy, such as the mean squared error, the way we evaluate accuracy conveys whether the estimate is an underestimate or overestimate of the reference value, which is the value reported by *ns-3*. Figures 4.4 and 4.5 show MAPE's and AFTER's throughput estimate accuracy for CBR and multimedia traffic in both the *Random Indoor* and the *Grid Outdoor* topologies, respectively. The red line represents the "ideal" ratio of 1, i.e., a perfect match between the estimates and *ns-3*'s measured throughput.



(a) CBR-Traffic (b) MM-Traffic

Figure 4.4: Estimated throughput accuracy relative to *ns-3* in *Random Indoor* topology.



(a) CBR-Traffic (b) MM-Traffic

Figure 4.5: Estimated throughput accuracy relative to *ns-3* in *Grid Outdoor* topology.

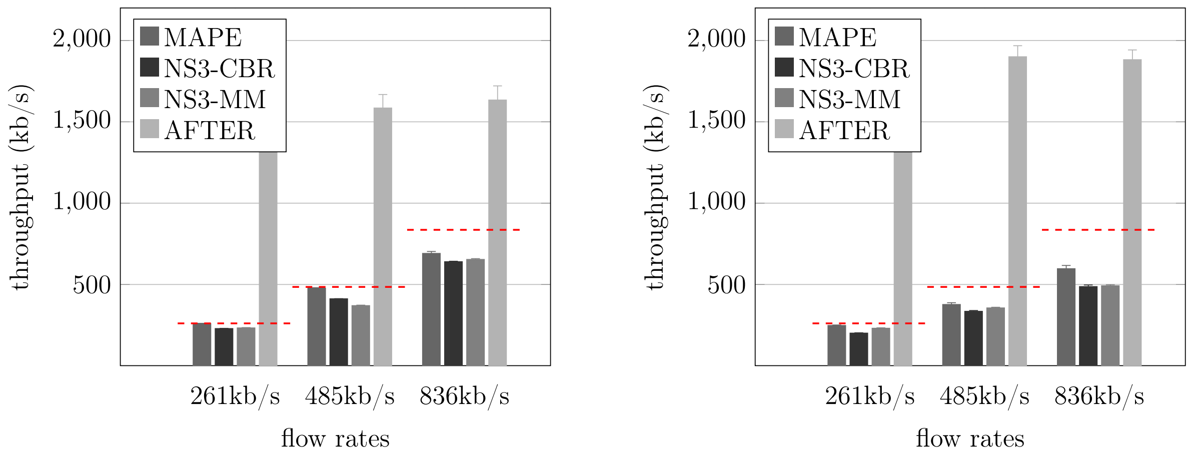
We observe that AFTER's throughput estimates are significantly less accurate when compared to MAPE because AFTER's simulated flows attempt to transmit at the highest supported rate, typically resulting in overestimates. This is particularly pronounced for scenarios with few flows in which there is low inter-flow interference and, consequently, more residual network capacity to support higher transmission rates. As more flows are added, AFTER's prediction improves because, with more flows sharing the network's capacity, there is less room for each flow's transmission rate to increase beyond the real transmission rate.

This prediction discrepancy between AFTER and MAPE also quantitatively demonstrates the impact that not accounting for specific flow transmission rates may have. It also illustrates that MAPE is able to significantly improve prediction accuracy for scenarios with few flows (in our experiments, 3- and 6-flow scenarios). MAPE's accuracy decreases in scenarios with more flows — with a bias toward overestimates due to some simplifications inherited from AFTER. For instance, AFTER does not take into account packet losses due to collision, which may influence network throughput when there are more flows transmitting simultaneously. Instead, in its inter-flow interference model, AFTER implements a medium access scheduler based on an interference graph of the topology. In future work, we plan to address this issue by improving how flow interference is modeled.

Note that MAPE yields higher accuracy for CBR traffic (Figures 4.4a and 4.5a). That is because its scheduler also generates *v_packets* at constant rates. For multimedia (MM) traffic scenarios (Figures 4.4b and 4.5b), however, transmission rate variations cause MAPE to overestimate the throughput. This is because bursts of the more important video packets cause losses due to buffer overflow and packet collisions, while less important video packets which have lower transmission rates are delivered more reliably.

We also evaluate the per-flow throughput prediction accuracy considering the different classes of flows based on their transmission rates. Figure 4.6 summarizes the results for 6 flows using CBR and MM traffic in both the indoor and outdoor topologies. We also ran these experiments for 3, 9 and 12 flows, but we omit those results since they show similar trends. The red reference lines represent the ideal throughput based on the average bitrate generated for each video trace.

The figure shows that AFTER tends to overestimate all three classes of flows. Moreover, as the source-destination pair is chosen randomly regardless of the transmission rate of the flow, the average throughput estimated by AFTER tends to be roughly the same for all three classes. Conversely, by knowing the transmission rate of each flow, MAPE is able to more accurately estimate per-flow throughput. Note, however, that it slightly overestimates MM's throughput. This issue, which is more pronounced in the outdoor topology due to its higher link reliability, is due to the fact that MAPE's current implementation uses AFTER, and thus it inherits the mechanism used by AFTER to estimate packet loss. It considers two possible sources of packet loss: buffer overflow and link-layer



(a) *Random Indoor* Topology.

(b) *Grid Outdoor* Topology.

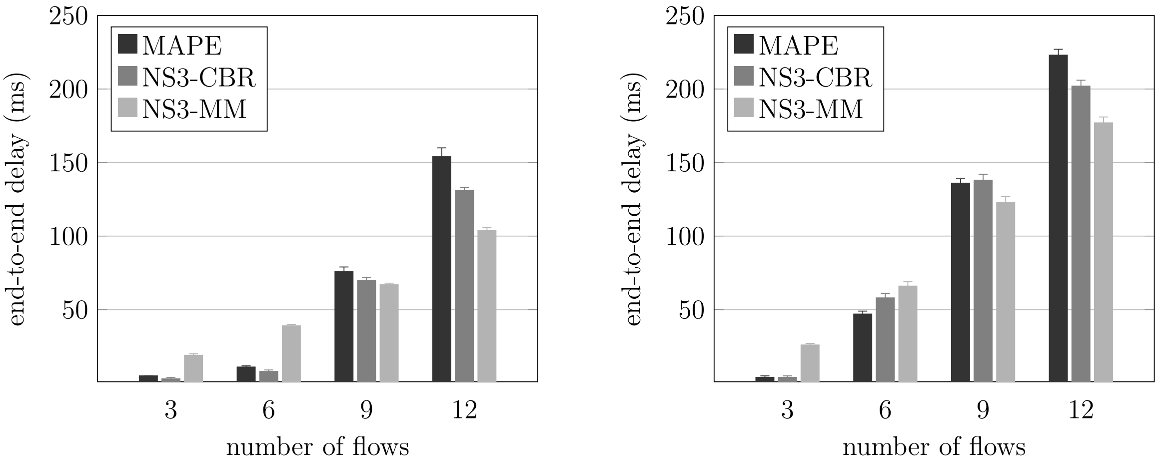
Figure 4.6: Average throughput for scenarios with 6 flows.

transmission losses. If all links that compose a path have perfect delivery rates, losses computed by AFTER are only due to buffer overflow. In practice, however, there are other sources of losses, such as collisions, and as a result, MAPE and AFTER tend to overestimate flows' throughputs.

4.6.3 Estimated Delay and Packet Loss

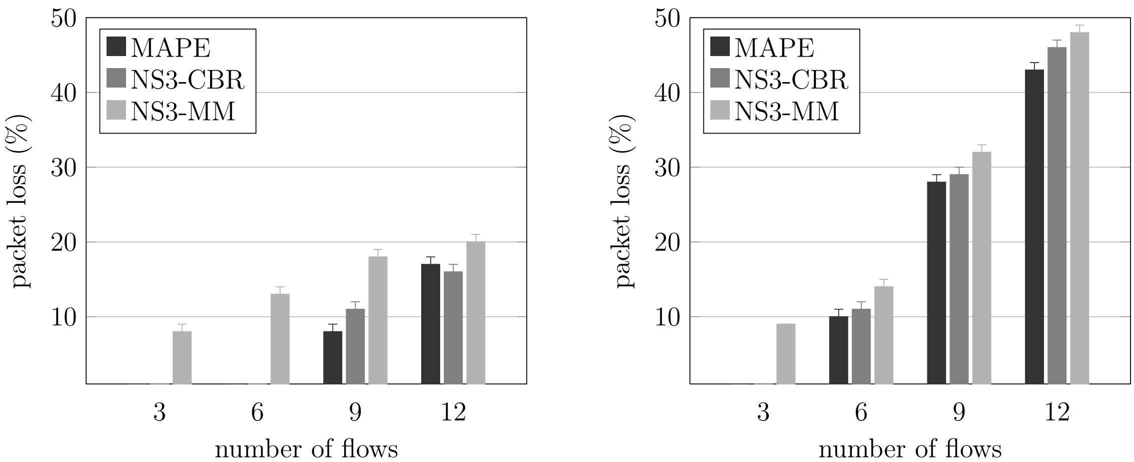
We also evaluate MAPE's delay and packet loss estimates. Figure 4.7 shows the average end-to-end delay, considering the 95% confidence intervals for different number of flows in both experimental topologies. When compared to the results obtained by *ns-3*, MAPE shows similar delay increase trend as the number of flows increases. Note that MAPE overestimates the end-to-end delay for scenarios with 12 flows in both topologies. This is due to MAC layer congestion as more flows share the same nodes/links increasing contention and consequently increasing MAPE's time to reach steady state, which, in turn, may cause MAPE's execution to end before reaching steady state. Although MAPE's estimate is less accurate compared to *ns-3* when it does not reach steady state, we will demonstrate in the Section 4.7 that these results are still useful to inform the route selection process ahead of multimedia flow transmission.

Figure 4.8 plots the average packet loss rate. It also shows a discrepancy between MAPE's and *ns-3*'s estimates in both topologies. But here, instead of overestimating, losses are generally underestimated by MAPE. The culprit is the absence of a collision packet loss counter in MAPE, which causes it to be more prone to estimate lower overall loss rates. These results also help explain the reason for instances in which MAPE overestimates the throughput — a consequence of fewer packets being discarded at the MAC layer. Furthermore, as expected, packet losses for MM traffic were even more impacted by the bursty nature of the video packet flows. As part of our future work, we plan to improve how MAPE models packet losses due to collision.



(a) *Random Indoor* Topology. (b) *Grid Outdoor* Topology.

Figure 4.7: Average end-to-end delay for different number of flows.



(a) *Random Indoor* Topology. (b) *Grid Outdoor* Topology.

Figure 4.8: Average packet loss for different number of flows.

Despite those discrepancies, the results shown in Figures 4.6, 4.7 and 4.8 demonstrate MAPE's ability to capture the overall trend in throughput, delay and packet loss for multimedia flows in different application scenarios. Furthermore, we note that the discrepancies for 9 and 12 flows are mostly caused by network congestion and MAPE estimates being generated before steady state is achieved.

In order to confirm this hypothesis, in Figure 4.9 we show a scatter plot for the 9-flow runs using the *Random Indoor* topology representing which instances did and did not reach the steady state and their respective delays discrepancies when comparing MAPE to *ns-3* — i.e., the difference between MAPE's and *ns-3*'s average delay estimates. Note that we show results for the 9-flow *Random Indoor* topology experiments because, with 9 flows (and above), the network gets more congested and consequently the number of instances that do not reach steady state increases, which, as previously discussed, results in higher delay and packet loss discrepancies.

As the plot shows, when steady state is reached, MAPE yields adequate estimation accuracy, with discrepancies concentrating around less than 100 *ms*. However, MAPE tends to overestimate end-to-end delay for instances that do not reach steady state, causing higher discrepancies.

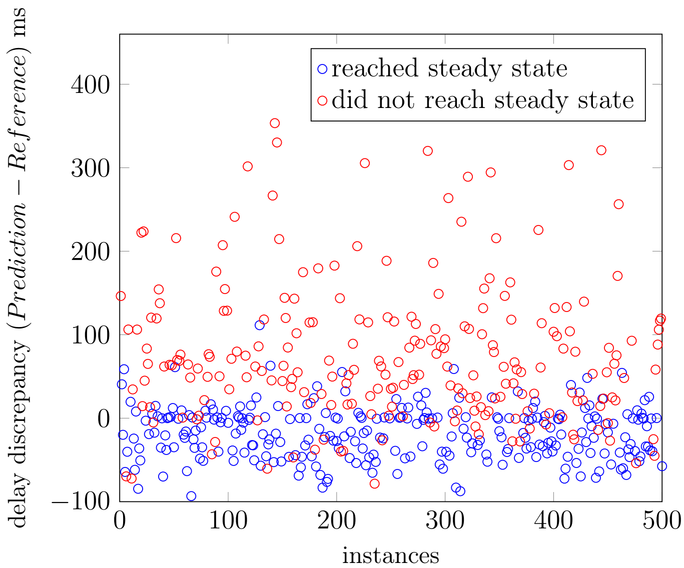


Figure 4.9: Difference between MAPE's and *ns-3*'s average delay predictions considering MAPE's steady and non-steady instances for scenarios with 9 flows in the *Random Indoor* Topology.

Figure 4.10 shows a scatter plot of the path average throughput according to *ns-3* for instances that reached the steady state and those that did not as a function of the delay estimate discrepancies for the 12-flow experiments in the *Random Indoor* Topology (since they showed the highest discrepancies). As the plot shows, higher throughput paths are concentrated around the smallest discrepancies, while the largest discrepancies happen with paths that present poor network performance and are, thus, not suitable for multimedia flows. Note that instances with higher throughput were those in which MAPE was able to reach steady state, unlike runs that exhibit higher discrepancies, which, again, are the ones where steady state was not reached.

As a tool to guide real-time route selection decisions for IoT multimedia applications, low throughput routes —likely because of congestion — are generally undesirable, as they are often unable to meet the requirements of multimedia flows. As such, overestimating delay for those paths should not negatively impact path selection. That is, MAPE's delay overestimates when compared to *ns-3*'s correspond to paths that are undesirable for video traffic anyway and therefore should not be selected by routing.

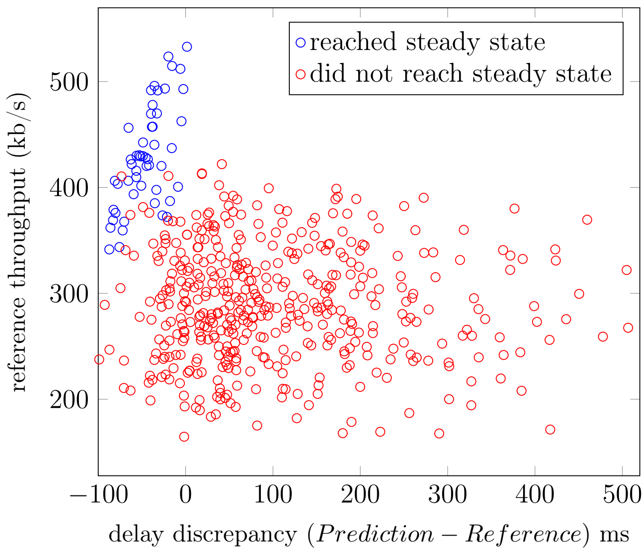


Figure 4.10: Path average throughput (according to *ns-3*) as a function of the difference between MAPE's and *ns-3*'s average delay predictions for scenarios with 12 flows in the *Random Indoor* Topology.

4.7 Video Quality Evaluation

The ability to estimate network performance is essential to ensure adequate network support for many IoT multimedia applications. In the case of applications involving video transmission, for instance, timely and fresh estimates of the current state of the network can significantly help routing protocols to rapidly identify paths that satisfy QoS constraints, as well as promote load balancing and better network resource utilization. To examine how MAPE's performance estimates can be used to improve overall video quality, we use a well-known QoE metric called Structural SSIM [120] measured by the EvalVid video transmission and quality evaluation framework [66]. In the second part of this section, we evaluate the quality of the video transmitted using the route selected based on MAPE's estimates.

4.7.1 Video Structural Similarity

The Structural Similarity Index Measure, or SSIM, measures video structural distortion which is known to correlate with video quality as perceived by the end user [120]. This metric combines luminance, contrast, and structural similarity of the frames to compute the correlation between the original frame and the (possibly distorted) displayed one. SSIM values vary between 0 and 1, with higher values meaning better quality.

To show how MAPE estimates can be used to improve video quality, we run experiments transmitting the "Hall Monitor" video clip (as described in Section 4.5.2) and compute the SSIM by comparing all transmitted and received video frames.

Figure 4.11 plots the average SSIM of the instances for different numbers of flows in the *Random Indoor* topology. According to the delay discrepancy ranges observed in Figure 4.9, we group experimental run instances in two classes, where the first class exhibits delay discrepancies greater than 100 *ms* (called *larger delay discrepancy*) and the second class exhibits delay discrepancies equal to or less than 100 *ms* (called *smaller delay discrepancy*) when compared to the results obtained with *ns-3*. From Figure 4.11, we observe that *larger delay discrepancy* instances were not found in scenarios with 3 flows; however, for the 6-, 9-, and 12-flow experiments, *larger delay discrepancy* instances resulted in lower video quality. Consequently, as previously discussed in Section 4.6.3, accuracy of MAPE's delay estimates could be useful to discard paths which result in a maximum threshold of the flow's delay to provide adequate video quality.

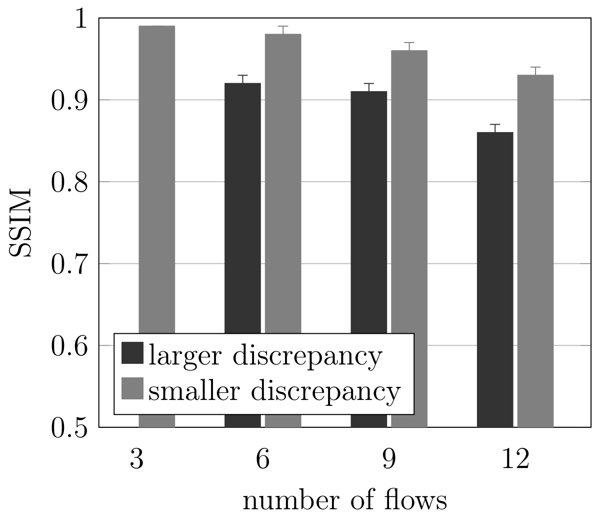
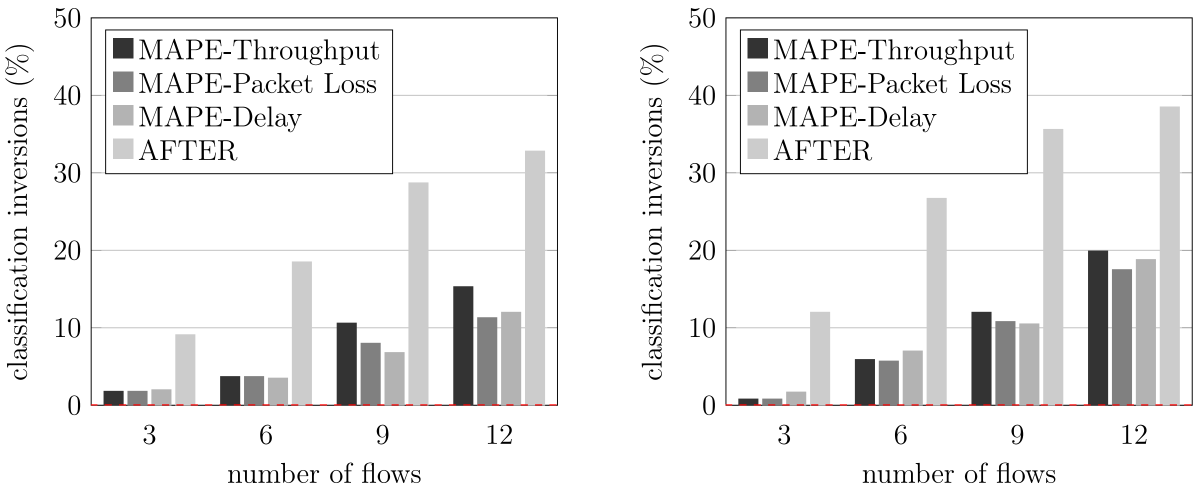


Figure 4.11: Average SSIM according to the instances with larger and smaller delay discrepancies for scenarios with different number of flows in *Random Indoor* Topology.

4.7.2 Classification Inversions

To evaluate how MAPE can be used to inform path selection for video transmission, we use the concept of *classification inversions* [91] defined as follows. Consider two different paths a and b. Suppose that video transmissions using path a yield higher SSIM than if path b was used. If MAPE's throughput estimate indicates that path b will outperform path a, that constitutes a *classification inversion* using throughput as metric; otherwise, if path a is selected, there is no inversion.



(a) *Random Indoor* Topology

(b) *Grid Outdoor* Topology

Figure 4.12: Percentage of classification inversions in terms of SSIM for different numbers of flows.

To better understand how classification inversions can be used in practice, let us consider the route selection problem in multipath routing, where a set of paths needs to be selected for the transmission of multiple flows. In this example, the most important aspect is to get the relative ranking of the path sets correctly in order to make adequate flow-to-path assignments.

Figure 4.12 shows a comparison between MAPE and AFTER in terms of classification inversions for both the *Random Indoor* and the *Grid Outdoor* topologies as a function of the number of flows. We evaluate the (baseline) quality of paths using SSIM metric. For MAPE, we consider three possible scenarios to classify paths: using the predicted average throughput, packet loss or end-to-end delay as metrics to compare the set of paths of all instances. Since AFTER does not estimate delay or packet loss, we only show results when throughput is used to calculate *classification inversions* based on AFTER estimates. MAPE results in lower percentages of classification inversions (lower than 20%) in all scenarios. AFTER, on the other hand, yields 3 to 6 times higher classification inversion rates, as it ignores flow transmission rates.

Note that MAPE's packet loss and delay estimates result in lower *classification inversions* for 9 and 12 flows when compared to *classification inversions* based on throughput. This demonstrates that both delay and loss should be considered when selecting paths for video transmission, especially when the network becomes saturated. These results are relevant because they confirm that MAPE's estimates, which can be computed in *quasi* real time, can be used to select paths that improve user QoE in terms of perceived video quality.

Chapter 5

FITPATH: Multipath Selection Mechanism

In this chapter, we introduce FITPATH, an efficient and adaptable path selection scheme for multipath routing that can accommodate environments where multiple video sources can simultaneously transmit heterogeneous video flows, i.e., flows of different bitrates. FITPATH uses a novel heuristic-based iterative optimization approach that estimates in real time the conditions of the underlying network while accounting for the different bitrate requirements of the application video flows.

We conduct an extensive comparative performance study considering application scenarios that mimic realistic IoT deployments and show that FITPATH outperforms various existing path selection mechanisms, delivering superior end-user QoE and achieving higher network performance. We evaluate FITPATH's convergence characteristics and show that it is able to, relatively quickly, generate path selection solutions that yield high QoE. We also demonstrate that FITPATH can be used in practice by video transmission sources to quickly generate feasible intermediate solutions that result in adequate video quality, while incrementally improving QoE as the algorithm iterates.

5.1 Problem Formulation

We focus on the problem of choosing paths for each video flow. As illustrated in Figure 5.1, we consider a known set S of network nodes that act as video sources. According to the video encoder used, each source $s \in S$ generates a set $f_s = \{f_{s_i}\}$ of video flows, where $0 < i \leq |f_s|$. Each flow f_{s_i} is transmitted from each source node through a selected path p_{s_i} within a list P of paths. At the destination, i.e., the sink node, the video decoder is responsible for synchronizing and merging the received flows to render the video stream. Depending on the application scenario, FITPATH can also handle multiple sinks by specifying the sink for each source s.

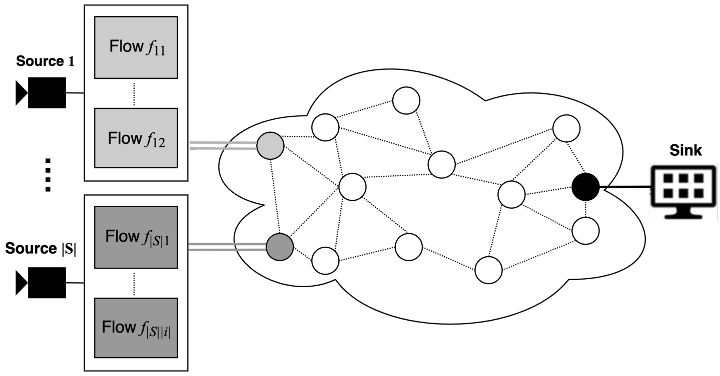


Figure 5.1: Multi-source, multi-path video streaming scenario.

In some applications, video sources generate multiple flows with different video compression techniques [8]. For instance, traditional video compression techniques adopted in multipath video transmission, such as LC and MDC, generate flows with different bitrates. As such, the multipath selection mechanism must account for the flow's bitrate to balance offered load and reduce packet losses and delays. Therefore, it needs to be aware of the application's characteristics to maximize the user's QoE.

5.2 Overview

FITPATH is a multipath selection mechanism based on the Iterated Local Search (ILS) metaheuristic [83] for selecting candidate paths for each video flow generated from multiple sources. Figure 5.2 shows FITPATH's main components. As input, FITPATH uses information about the network topology and about the video flows from the different video sources (e.g., number of flows and flow bitrate).

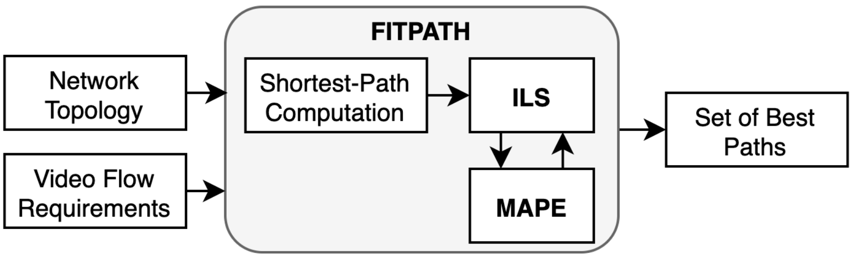


Figure 5.2: FITPATH overview

FIPATH starts by performing an all-pairs least-cost (or shortest) path computation to generate a list of candidate paths P, which is used by ILS as input. The list P of candidate paths are generated based on Yen's algorithm [127] by choosing, for each flow, a list of K least cost paths between the source and the sink using the ETX. In order to rank these K candidate paths, ILS applies its iterated search algorithm and evaluates each solution using MAPE estimates. MAPE, which was introduced in Chapter 4, estimates the network performance of candidate paths, i.e., throughput, delay and loss, which FITPATH's ILS algorithm uses to select the best candidate path for each flow.

5.3 Iterated Local Search

Iterated Local Search (ILS) algorithms have been proposed as one of the best heuristic-based approaches to solve a variety of combinatorial optimization problems of the NP-complete class [83]. They have also been used to solve network routing problems [92, 79, 105]. Additionally, as it tries to find a list of best paths for each video flow among candidate solutions, FITPATH is well suited for ILS whose iterative nature continuously seeks to improve a set of solutions.

Algorithm 2 shows FITPATH's ILS pseudo-code and Table 5.1 summarizes the algorithm's notations. FITPATH's ILS computation consists of the following steps: (i) $initSolution(P)$ initializes p with the first candidate solution in the list of candidate solutions P generated by an all-pairs least-cost path algorithm. P's calculation is based on the graph $G(N, L, F, R)$ representing the underlying network topology, where N is the set of nodes $\{0, 1, \ldots, n\}$, L is the set of links $\{0, 1, \ldots, l\}$, F is a list of flows associated with their sources — notice that $F = \bigcup_{s \in S} f_s$ and $|p| = |F|$ — and R is a list of bitrates corresponding to each flow. As discussed in Chapter 3, information about the underlying network topology, i.e., G, can be obtained in practice using different approaches, such as proactive link-state routing protocols (e.g. OLSR [32]), or from an SDN controller in SDN-based networks. Note that the initial solution p contains a single path for each pair source/sink, through which all flows are transmitted; (ii) then $MAPE$ is invoked and calculates the cost of the candidate solution p. The best solution p and its cost are used to try to find better solutions; (iii) $LocalSearch$ performs local searches to refine p seeking a better solution; (iv) $Perturbation$ generates intermediate solutions $p\prime$ and $p\prime\prime$ by applying pertubations to p; and (v) $AcceptanceCriterion$ evaluates and determines if a new solution can be accepted according to the objective function.

The output of FITPATH's ILS computation is the list of best paths p represented by a list. A possible solution p obtained for the topology in Figure 5.3 is illustrated in Figure

Algorithm 2: Iterated Local Search (ILS)

Input : P - List of candidate solutions;
Output: p - List of best paths;
begin
  $p \leftarrow initSolution(P)$;
  $\Gamma(p), \Delta(p) \leftarrow MAPE(p)$;
  $\Gamma_{best} \leftarrow \Gamma(p)$;
  $\Delta_{best} \leftarrow \Delta(p)$;
  $p \leftarrow LocalSearch(p, P, \Gamma_{best}, \Delta_{best})$;
  repeat
    $p\prime \leftarrow Perturbation(p, P, H)$;
    $p\prime\prime \leftarrow LocalSearch(p\prime, P, \Gamma_{best}, \Delta_{best})$;
    $p, \Gamma_{best}, \Delta_{best} \leftarrow AcceptanceCriterion(p, p\prime\prime, H)$;
  until *termination condition met*;
end

Table 5.1: ILS algorithm notations.

Notation	Definition
P	list of candidate solutions
p	current solution (list of best paths)
$p\prime$	perturbation solution
$p\prime\prime$	local search solution
c	candidate local solution
H	solution history list
Γ_{best}	estimated throughput gap of the current solution
Δ_{best}	estimated delay of the current solution
$\Gamma_{min}(c)$	minimum throughput gap of candidate solution c
$\Gamma(c)$	estimated throughput gap of candidate solution c
$\Delta(c)$	estimated delay of candidate solution c

5.4, where each list represents a path p_{s_i} for each flow f_{s_i}. In this example, the selected paths share some links, which may cause inter-flow interference and, consequently, lower network performance. Thus, the next step attempts to maximize this metric.

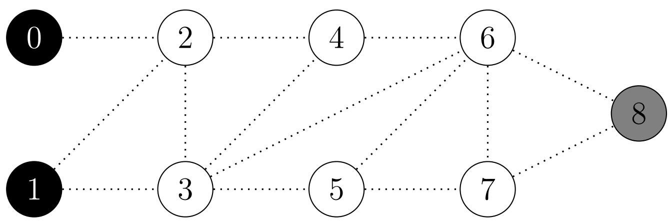


Figure 5.3: Example network topology used to illustrate FITPATH.

LocalSearch, *Perturbation* and *AcceptanceCriterion* iterate to traverse potentially good alternative solutions with the goal of accepting a solution that improves network

$$\boxed{p_{1_1}} \rightarrow \boxed{0}\boxed{2}\boxed{4}\boxed{6}\boxed{8}$$
$$\boxed{p_{1_2}} \rightarrow \boxed{0}\boxed{2}\boxed{3}\boxed{6}\boxed{8}$$
$$\boxed{p_{2_1}} \rightarrow \boxed{1}\boxed{3}\boxed{4}\boxed{6}\boxed{8}$$
$$\boxed{p_{2_2}} \rightarrow \boxed{1}\boxed{3}\boxed{5}\boxed{7}\boxed{8}$$

Figure 5.4: Example of a candidate solution p.

performance while meeting the requirements of the video flows. Based on MAPE's performance estimates, the ILS algorithm calculates the objective function given by Equations 5.1 and 5.2 which evaluate the candidate solutions using two network performance metrics: (1) the throughput gap $\Gamma(c)$ measures the normalized network throughput gap of a candidate solution c, defined as the sum of the percentage difference between the estimated throughput $\overline{Th_f}$ and bitrate λ_f for all flows f; and (2) end-to-end delay $\Delta(c)$ which is the estimated average delay $\overline{\Delta_f}$ averaged over all flows f of a candidate solution c, where $|F|$ is the number of flows. The end-to-end delay is used only as a tie-breaking criteria between solutions that estimate the same throughput gap. We choose to use a normalized function for the throughput gap because flows may have different bitrates.

$$\Gamma(c) = \sum_{f \in F} \frac{\lambda_f - \overline{Th_f}}{\overline{Th_f}} \tag{5.1}$$

$$\Delta(c) = \frac{\sum_{f \in F} \overline{\Delta_f}}{|F|} \tag{5.2}$$

The ILS algorithm stops when the termination condition is met establishing a limit on the number of iterations without significant improvement to the solution and on the algorithm's total execution time (often based on the application response time requirements). The termination condition can be used to control the balance between *intensification*, i.e., performing local search and *diversification*, i.e., exploring perturbations in the solution space of that search.

5.3.1 Local Search

At each iteration, *LocalSearch* evaluates solutions that are "neighbors" of the current best solution p. Here, two feasible solutions are considered neighbors if they differ by exactly one path. We use a simple local search technique to minimize computation and storage overhead. As shown in Algorithm 3, *LocalSearch* uses paths of the next solution in P to generate and evaluate all possible neighbor candidates of the current solution, including the same path for all source's flows, as if it were a single path transmission. Before estimating the network performance of each neighbor solution using MAPE, *LocalSearch* computes a pruning threshold using the *minimum throughput gap* $\Gamma_{min}(c)$, which is calculated based on the ETX. $\Gamma_{min}(c)$ is given by Equation 5.1, replacing $\overline{Th_f}$ with the maximum throughput given by the inverse of the sum of the ETX of all the links of flow f's path. Hence, solutions $c's$ that cannot meet the throughput estimate of the current best solution p are pruned to reduce computation resource usage and execution time.

Solutions that are not pruned are evaluated by MAPE, which returns estimates for the throughput and delay of each flow. A new candidate solution is considered better than the current best solution if it either has a lower throughput gap or if it results in the same throughput, but in lower delay.

Algorithm 3: Local Search

```
Input  : p, P, Γ_best, Δ_best;
Output: p;
begin
    p′ ← nextSolution(P);
    for new candidate c ∈ neighbors(p, p′) do
        if Γ_min(c) ≤ Γ_best then
            Γ(c), Δ(c) ← MAPE(c);
            if Γ(c) < Γ_best then
                Γ_best ← Γ(c);
                Δ_best ← Δ(c);
                p ← c;
            else if Γ(c) = Γ_best then
                if Δ(c) < Δ_best then
                    p ← c;
                    Δ_best ← Δ(c);
    end
end
```

5.3.2 Perturbation

To avoid converging too soon and getting "stuck" in local optimal solutions, FITPATH applies perturbations to the current best solution. Our current strategy is to generate a new candidate solution by randomly replacing one path for each source in the best currently known solution p. This procedure, shown in Algorithm 4, uses paths in P and the history solution list H which may lead to visiting new portions of the solution space not previously visited by the initial all-pairs least cost path (*initSolution*) or nearest neighbor (*LocalSearch*).

Algorithm 4: Perturbation

Input : p, P, H;

Output: $p\prime$;

begin

  $p\prime \leftarrow p$;

  foreach *source* $s \in p\prime$ **do**

    $i \leftarrow random(1, |f_s|)$;

    $j \leftarrow random(1, |f_s|)$;

    $p\prime_{s_i} \leftarrow random(p_{s_j} \in (P \cup H))$;

  end

end

5.3.3 Acceptance Criterion

FITPATH's quest for the set of best paths terminates when a pre-specified acceptance criterion is met (see Algorithm 2). FITPATH compares the costs of solutions p and $p\prime\prime$ considering the same criterion used in *LocalSearch* and accepts the best one. Whenever the best solution is accepted, Γ_{best} is updated and the previous solution is added to the solution history list H that is consulted upon future perturbations.

FITPATH's convergence time depends on the network topology and application scenarios, as the solution space increases with the number of sources and the number of flows. For more complex scenarios that require longer convergence times, video transmission could start using FITPATH's output after the first few iterations; FITPATH can continue to execute in the background, looking for better solutions.

5.4 Evaluation Methodology

We evaluate the performance of FITPATH in the same experimental environment used in Chapter 4, using the ns-3 [98] network simulator performed on a dedicated server with an Intel i7-860 processor running at 2.8 GHz and with 32 GB of RAM. We consider application scenarios that mimic typical surveillance deployments in urban regions where video sources transmit flows simultaneously to a single monitoring center [88]. We use a variety of video sequences that contain scenes of different levels of motion in an attempt to reproduce video traffic diversity in complex environments, such as smart cities [102].

In each simulation run, nodes were randomly placed in a 300 m $\times$ 300 m region separated from each other by a minimum distance. In our current experiments, we used a minimum separation distance of 5 meters. At the Medium Access Control (MAC) layer, we adopt the 802.11g standard. Although it is capable of transmitting 54 Mb/s, a fixed link rate of 18 Mb/s was used to evaluate scenarios with more challenging network utilization levels. At the physical layer, links are simulated using the Cost231 propagation model [109] at 2.4 GHz, reproducing a more realistically urban environment.

In our experiments, we use the maximum execution time of 60 sec as ILS' termination condition but that can be adjusted accordingly depending on the driving application. Additionally, as discussed in Section 5.3.3, intermediate solutions can be used during the convergence process, and, as our results show (see Section 5.5.4.4), they yield adequate performance. For P, the list of candidate solutions used by ILS as input, we set its size to $100 \times |S|$, where $|S|$ is the number of sources. P's size was determined empirically: given the computational environment used in our experiments, $100 \times |S|$ was the maximum candidate solution size that could be evaluated within FITPATH's stipulated maximum execution time of 60 sec. Table 5.2 lists the parameters and their values used to simulate a variety of network and application scenarios.

Table 5.2: Simulation Parameters

Network		Application	
Network area (size)	300 m × 300 m	Application type	Variable Bit Rate (VBR)
Network topology	Random	Video Frame Rate	30 fps
No. of nodes	30, 45, 60	Video Scene Motion Level	Low, Medium and High
Min. node distance	5 m	Packet Size	1500 B
MAC layer	802.11g	Video Encoder Rate	256 kb/s, 512 kb/s, 1 Mb/s, 1.5 Mb/s and 2 Mb/s
Physical type	Wifi Phy		
Channel type	Wifi Channel	No. of sink nodes	1
Frequency band	2.4 GHz	No. of source nodes	4
Data rate	18 Mb/s	No. of flows	8 (2 from each source)
Propagation Model	Cost231	Offered Load	1 Mb/s, 2 Mb/s, 3 Mb/s, 4 Mb/s, 6 Mb/s and 8 Mb/s

5.4.1 Traffic Models

The Evalvid [66] framework was used to generate realistic video traffic for the simulations. It generates traffic corresponding to a given video clip and evaluates the quality of the video delivered at the receiver.

Our experiments employed publicly available and commonly used video clips, namely “Hall Monitor”, “Paris” and “CoastGuard” [102], which represent different motion complexity levels, i.e., low-, medium- and high-motion, respectively. The video clips were generated in a loop of 1 and 2 minutes in duration and converted to H.264 format at a rate of 30 frames per second. Considering real-time transmission delay and human tolerance, the play-out buffer was set to 300 ms to mitigate potential out-of-order packets; packets delayed longer than 300 ms were discarded at the decoder.

Video traffic was generated using a mix of five different target video encoder rates, namely 256 kb/s, 512 kb/s, 1 Mb/s, 1.5 Mb/s and 2 Mb/s to represent different levels of video quality. At the video source, flows were forwarded according to the multipath forwarding strategies specified by each of the evaluated proposals. Both LC or MDC video compression techniques were used.

5.4.2 Evaluation Metrics

Our evaluation considers both network performance and quality of experience. For that, we experiment with different network densities and offered load (see Table 5.2). Network density refers to the number of nodes in an area, i.e, topologies generated with 30, 45 and 60 nodes. Offered load is defined as the sum of all flows bitrate transmitted simultaneously, considering the average rate used as target for generating video traffic. We vary video traffic for each source generating offered loads between 1 and 8 Mb/s. Furthermore, we evaluate the heuristic convergence time varying the time used as stopping criterion from 5 to 60 seconds. It aims to demonstrate the feasibility of implementing FITPATH in terms of speed to obtain a solution that results in better video quality.

- **Network performance:** We evaluate network performance using per-flow throughput, end-to-end delay, video frame loss, and network utilization. Per-flow throughput is calculated as the ratio between the number of bits delivered at the destination and simulation time per flow, averaged over each flow bitrate. End-to-end delay is the time interval between when a packet is transmitted by the source node and when that packet is delivered at the destination, averaged over all received packets. Video frame loss is calculated as the percentage of frames transmitted that were not delivered at the destination. Network utilization is based on the nodes' queue occupancy averaged over the simulation time and is measured as the highest average queue occupancy among all nodes.

- **Quality of experience:** We use the SSIM and the Peak Signal To Noise Ratio (PSNR) to evaluate received video quality. Both metrics have been widely used to measure user's QoE [89]. As detailed in Section 4.7. SSIM values range from 0 to

1, where 1 means maximum quality. PSNR, measured in dB, is calculated as the frame-by-frame error between the original and the delivered video. The PSNR value tends to increase as the video quality improves. While we provide PSNR results in Table 5.5, our analysis focuses on the SSIM metric as it has been reported to better correlate with QoE [120].

5.4.3 Evaluation Baselines

We evaluate FITPATH by comparing its performance against six different mechanisms. As introduced in Chapter 2, the mechanisms ERVT [48], CLMR [1], RTVP [4], Q-MMTP [56], QSOpt [94] and ILS-MDC [20] adopt different multipath selection strategies and metrics.

The ERVT [48] implements an extended version of the AOMDV [85] routing protocol where the two best disjoint paths according to the minimum hop-count metric are selected for each source, while the others are considered backup paths. This approach assumes the existence of paths with disjoint nodes for each source. However, this assumption is not always true, particularly for scenarios with multiple video sources. In terms of video coding, ERVT combines LC and MDC techniques, where two descriptions of each layer are generated in the FMO format of the H.264/AVC standard. Macroblocks of each layer are divided into two paths. Hence, in bursty error conditions, the error is smoothly spread over all layers. For better protection and more network compatibility, the data in the base layer is partitioned and the layer's most important part is repeated in both paths. However, as the proposed error resilience is based on duplicate packets, there is no preference of one path over the other, which could result in an overloading of the network.

The CLMR [1] proposes a reactive protocol that starts with the flooding of a route request to obtain the total cost for each path. It aims to improve QoS and minimize energy consumption by measuring the path cost with a parameterized weight to control the importance of delay, packet delivery ratio and energy metrics. However, these parameters are defined offline and configured statically which makes this method not suitable for dynamic video services. CLMR classifies the multimedia traffic into three flows based on the LC technique. So, at least three paths are needed from the source node to the sink node. These three paths are not necessarily node disjoint paths: rather, some paths may have common nodes, according to the acceptance rules. When the joint nodes receive two flows, the forwarding priority will be given to the higher priority flows.

The RTVP [4] is a path selection approach based on a geographic structure in the form of *coronas*, where the network is divided into regions. The coronas are delimited by a set of circumferences of certain radii centered at the destination node. Each node is assigned a corona level according to its position in these regions. Thus, paths are selected by forwarding packets hop-by-hop from a source that has a high corona level to the destination node, which has corona level zero, seeking to avoid nodes that share the same paths. This approach benefits mobility as the corona levels are changed according to the nodes' position, but, depending on the corona structure, it might select paths with a high number of hops. RTVP also uses LC strategies that allow highest priority packets to use the most reliable path. When the common nodes of multiple paths receive more than one type of packet, the higher priority packet is sent through the best path and the common node sends a feedback control packet to the neighbor that forwarded the lower priority packet. When a node receives this feedback control packet, it changes the selected route to the next best node. This strategy can improve the reliability and the load balancing, however these proposals do not consider interference between flows from multiple sources. In addition, RTVP presents an adaptive traffic shaping in order to minimize the burstiness of the video flow considering the path capacity. It uses a token bucket algorithm that allows some burstiness in the outgoing traffic, but limits the maximum burst size.

Q-MMTP [56] proposes a multipath selection mechanism using the Spline mathematical model to estimate a set of ideal positions for the nodes. As a result, it generates a series of interpolations between the source and the destination. Thus, the paths are selected according to the distance between the nodes and the ideal positions. Although Q-MMTP provides adequate load balance for multiple flows, it does not support multiple sources and its strategy depends on the nodes' geographical positions. To support the real-time transmission through concurrent paths, Q-MMTP also includes a path scheduling to distribute the flows according to the energy consumption of each path. The flows of the path with the highest load are redirected to the least loaded path seeking to ensure the overall throughput of the system and avoid the failure of a single path because of overload. This mechanism enhances the system reliability, but does not integrate with video coding at the application layer.

QSOpt [94] formulates the multipath selection problem using mixed integer linear programming and implements a heuristic to find a solution that maximizes QoE. To evaluate QoE, a discrete function is proposed to determine a mapping between packet loss and QoE metrics. While the centralized algorithm provides a viable solution to improve resource utilization as well as QoE, the QSOpt model is not scalable and requires significant computational resources as the number of flows increases. QSOpt considers application scenarios with multiple sources and multiple destinations where each flow is defined as a commodity taking into account inter-flow interference. However, it also assumes identical transmission rate for all flows, regardless of video coding characteristics, which impacts the level of interference and consequently their ability to find good solutions.

In our previous work, we proposed a mechanism called ILS-MDC [20] based on the Iterator Local Search (ILS) metaheuristic for transmitting flows generated by the MDC technique, where even and odd frames are transmitted through different paths. The heuristic performs iterative searches for a list of paths that maximize the aggregate network throughput. The objective function was implemented using the AFTER [91], a real-time throughput estimator that estimates the throughput of each flow for a given list of paths. However, flow bitrates are not specified in the ILS-MDC because AFTER does not support this. Instead, AFTER assumes a TCP-like behavior in which flows transmit at the maximum capacity achieved by their respective paths. Because multimedia flows have well-defined bitrates, this may lead to significant inaccuracies.

FITPATH addresses the above issues by using MAPE which accounts for video transmission requirements, flow interference and multiple sources transmitting flows of potentially different bitrate. It enables FITPATH to handle both LC and MDC video coding. In addition, FITPATH implements new local search strategies that maximize the number of candidate solutions evaluated – to identify a list of paths that maximizes the users' QoE. Table 5.3 shows a comparison between the mechanisms considered in our performance evaluation study.

Table 5.3: Summary of the related proposals according to their multipath selection approaches.

Mechanism	Route Discovery	Multipath Selection	Type of Algorithm	Routing Metric
FITPATH	**Proactive**	**Partially Disjoint**	**Heuristic-based**	**QoS (throughput and delay estimates) and Interference**
ERVT	Reactive	Disjoint-Path	Hop-based	Hop count
CLMR	Reactive	Disjoint-Path	Hop-based	QoS (delay and packet loss) and Energy
RTVP	Hybrid	Partially Disjoint	Position-based	Geographical, QoS (delay and packet loss) and Energy
Q-MMTP	Reactive	Partially Disjoint	Position-based	Geographical, QoS(delay) and Energy
QSOpt	Proactive	Partially Disjoint	Heuristic-based	QoE model based on QoS (packet loss) and Interference
ILS-MDC	Proactive	Partially Disjoint	Heuristic-based	QoS (throughput estimates) and Interference

Table 5.4 compares the video coding techniques and application scenarios originally adopted by these mechanisms. Most of the proposals consider a fixed number of flows. In contrast, QSOpt evaluates multiple sources and a variety of flows. However, similar to Q-MMTP, QSOpt assumes CBR traffic for all sources.

We implement all these mechanisms to run on ns-3. Once validating the implementation using their original scenarios presented in Table 5.4, we evaluate them by using the same scenarios as specified in Table 5.2.

Table 5.4: Summary of the related proposals classified according to their evaluation scenarios.

References	Video Coding	Number of Sources	Number of Flows/Source	Number of Sink	Per-flow Rate
FITPATH	**LC/MDC**	**1 - 4**	**2**	**1**	**256 kb/s - 2 Mb/s**
ERVT	LC/MDC	1	2	1	1 Mb/s
CLMR	LC	1	3	1	250 kb/s
RTVP	LC	1	2	1	250 kb/s
Q-MMTP	-	1	5	1	100 kb/s
QSOpt	-	4	1 - 8	1	2 Mb/s
ILS-MDC	MDC	1 - 4	2 - 4	1	1 Mb/s

5.5 Simulation Results

We divide the results from our evaluation study into four broad categories: FITPATH's average performance compared against the other multipath path selection mechanisms considered in Chapter 5.4.3, the impact of the network density, the impact of offered load, and an analysis of heuristic performance in terms of the objective function, pruning, video coding and convergence time.

5.5.1 FITPATH's Average Performance

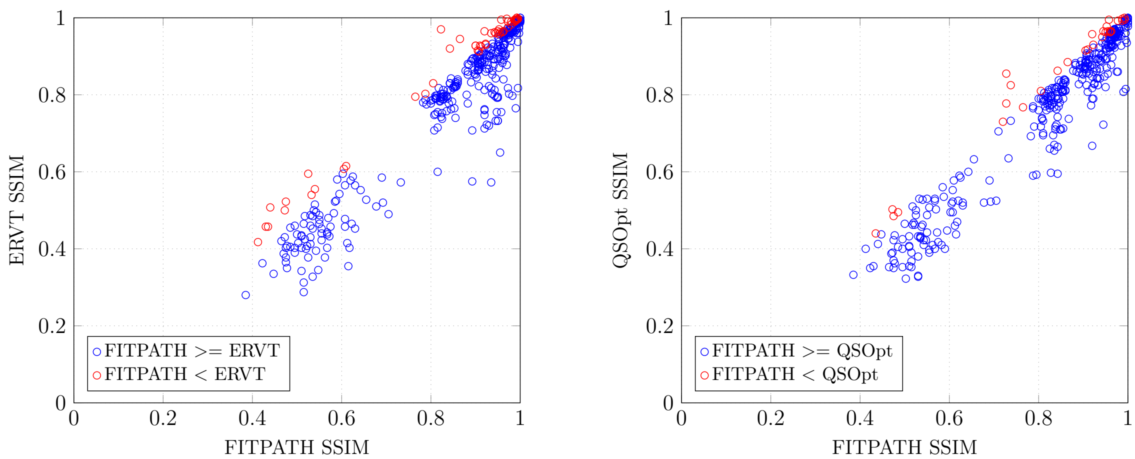
Table 5.5 summarizes FITPATH's average performance as well as the performance of the multipath path selection mechanisms considered in our study. These experiments were executed for all scenarios (different number of nodes and coding rates) using 4 Mb/s offered load. Value ranges shown for the different metrics denote 95% confidence intervals around the mean. As expected, the network performance of each mechanism — throughput, delay and frame loss — correlates well with their video quality performance. Overall, FITPATH performs better in terms of per-flow throughput and video frame losses, resulting in superior video quality as reflected in the PSNR and SSIM metrics.

To compare FITPATH against the second and third top performers in terms of SSIM, i.e., QSOpt and ERVT, we plot the SSIM results of these mechanisms for all simulation runs. Each point in the graphs of Figure 5.5 represents a simulation run, or *instance*, and its coordinates correspond to the SSIM obtained by FITPATH (horizontal axis) and the SSIM for ERVT or QSOpt (vertical axis). In particular, blue points denote instances where FITPATH reached higher or equal SSIM, while the red ones are instances for which

Table 5.5: Network and video quality performance for the different path selection mechanisms considering different number of nodes and coding rates with 4 Mb/s offered load. Value ranges denote the lower and upper limits of the 95% confidence intervals around the mean.

Mechanism	Per-flow Throughput (kb/s)	End-to-End Delay (ms)	Video Frame Loss (%)	PSNR (dB)	SSIM
FITPATH	**[420,427]**	**[98,100]**	**[29,30]**	**[30,31]**	**[0.85,0.86]**
ERVT	[406,413]	[91,93]	[33,34]	[29,30]	[0.80,0.81]
CLMR	[373,380]	[121,123]	[39,40]	[26,27]	[0.74,0.75]
RTVP	[360,366]	[102,104]	[40,41]	[26,27]	[0.75,0.76]
Q-MMTP	[267,270]	[206,208]	[58,59]	[18,19]	[0.54,0.55]
QSOpt	[409,417]	[105,107]	[34,35]	[28,29]	[0.81,0.82]
ILS-MDC	[392,398]	[109,111]	[39,40]	[27,28]	[0.77,0.78]

it yielded worse performance. In both graphs, there is clearly a significantly larger number of blue points (over 85%), indicating that FITPATH yields superior video quality compared to the other two mechanisms in most instances. We notice FITPATH underperforms in instances with paths containing lower quality links according to their estimated ETX. As future work, we intend to explore different mechanisms to estimate links performance as well as investigate alternate link quality metrics, e.g., the Received Signal Strength Indicator (RSSI). Moreover, we observe that in scenarios with higher offered load and higher motion levels, and thus lower SSIM, FITPATH is less likely to be outperformed by ERVT or QSOpt.



(a) SSIM for FITPATH and ERVT. (b) SSIM for FITPATH and QSOpt.

Figure 5.5: QoE performance

Our simulations confirmed that performance is strongly influenced by the characteristics of the video. In particular, Figure 5.6 which plots mean network utilization (95% confidence intervals are very small and not visible in the graph) for different number of nodes, shows that, as expected, network utilization increases for all path selection algorithms as video motion levels increase. This happens because of the transmission bursts resulting from changes in scenes. Thus, videos with more motion induce more bursts which put more pressure on the network. Even though all path selection approaches are affected by motion levels, FITPATH is the least affected one and results in lower network utilization for all three types of video, indicating it chooses more efficient paths overall. This result demonstrates that, since network utilization influences video quality, proposals that better distribute the load between paths are able to deliver superior video quality. FITPATH yields lower network utilization for all scenarios considered. Although ERVT and QSOpt present similar results in terms of video quality as shown in Table 5.5, ERVT and QSOpt tend to incur higher network overhead by, respectively, transmitting duplicate packets to improve error resilience and by not selecting paths that are not able to meet flow bitrates.

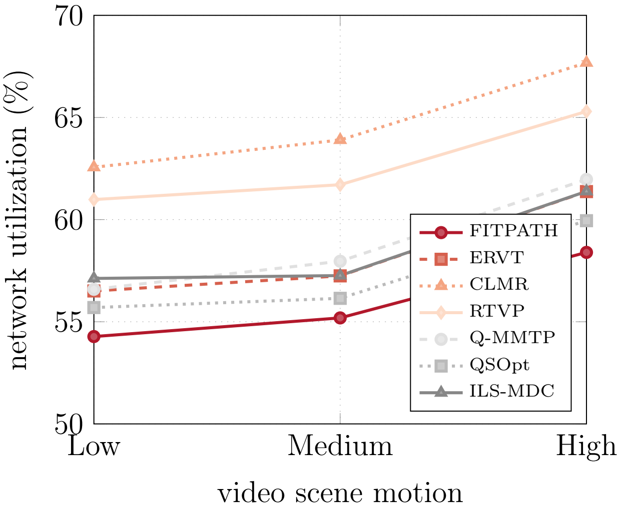
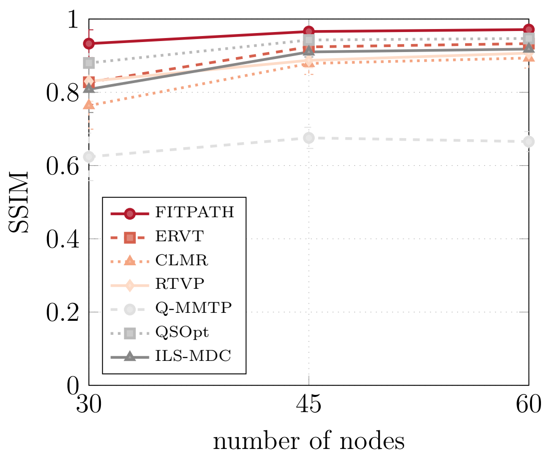


Figure 5.6: Average network utilization for videos with different scene motion levels.

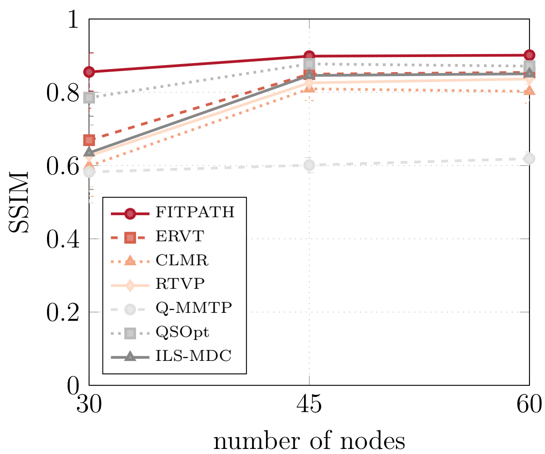
Table 5.6 summarizes SSIM results for all path selection mechanisms studied considering different scene motion levels in scenarios with 60 nodes and 4 Mb/s offered load. We also ran experiments for the different number of nodes and offered loads and observed that they exhibit similar trends. We again see that FITPATH obtains higher video quality, regardless of the level of motion in the videos. We also highlight that SSIM's variance is always low, indicating that FITPATH is consistently able to find good solutions. While QSOpt and ERVT presented results that were relatively close on average, their variances are much larger, suggesting they more often result in inferior solutions. FITPATH's efficiency is even more evident when we look at lower SSIM values: for low-motion videos, for example, FITPATH's lowest SSIM is 0.82, while QSOpt and ERVT both have at least one instance with an SSIM as low as 0.52. FITPATH SSIM's low variance also demonstrates that FITPATH can identify good paths in a wide range of scenarios.

Table 5.6: SSIM for different video scene motion levels with 4 Mb/s offered load and 60 nodes.

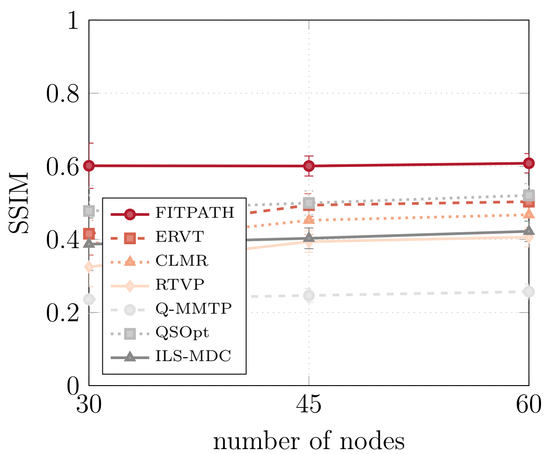
Mechanism	Low-Motion				Medium-Motion				High-Motion			
	SSIM	σ^2	max	min	SSIM	σ^2	max	min	SSIM	σ^2	max	min
FITPATH	**0.97**	**0.04**	**0.99**	**0.82**	**0.90**	**0.05**	**0.93**	**0.79**	**0.61**	**2.01**	**0.82**	**0.30**
ERVT	0.93	0.31	0.99	0.51	0.85	0.25	0.93	0.61	0.50	3.22	0.82	0.06
CLMR	0.89	2.35	0.99	0.50	0.80	3.22	0.93	0.25	0.47	4.90	0.86	0.05
RTVP	0.91	0.61	0.99	0.51	0.84	0.64	0.91	0.43	0.41	2.30	0.78	0.12
Q-MMTP	0.66	12.50	0.99	0.00	0.62	10.20	0.90	0.00	0.25	2.37	0.82	0.00
QSOpt	0.95	0.50	0.99	0.52	0.87	0.32	0.93	0.69	0.52	3.27	0.85	0.21
ILS-MDC	0.92	0.57	0.98	0.52	0.85	0.31	0.92	0.70	0.42	2.42	0.77	0.16



(a) Low-Motion video scene.

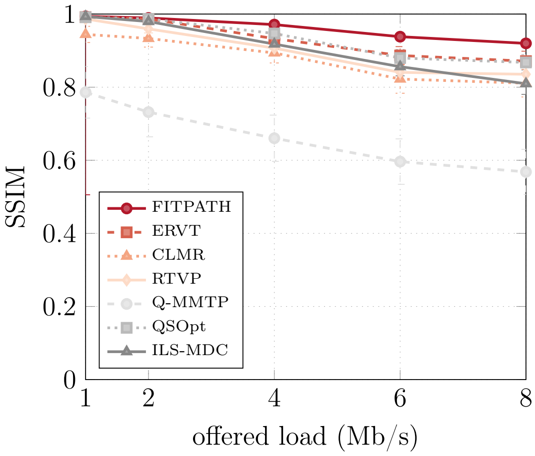


(b) Medium-Motion video scene.

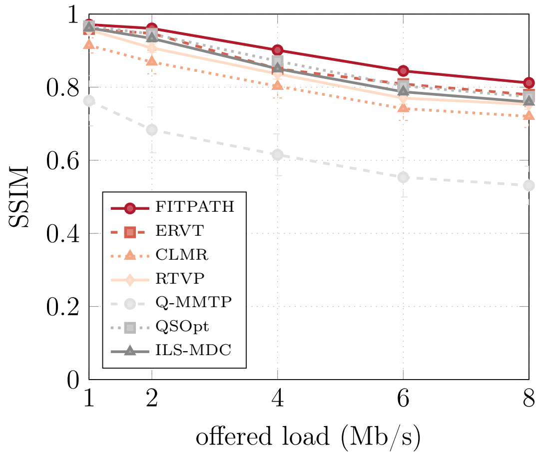


(c) High-motion video scene.

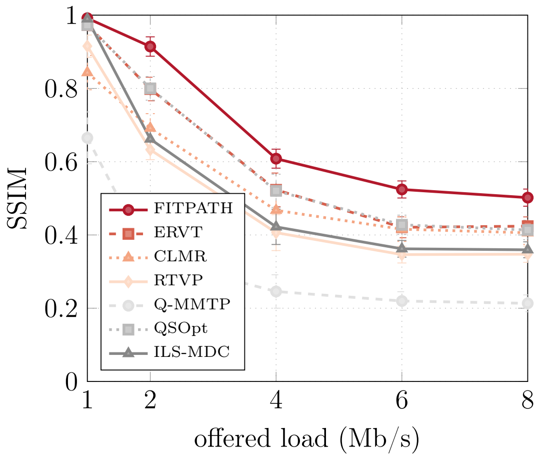
Figure 5.7: SSIM for different network densities and different video scene motion levels under 4 Mb/s offered load.



(a) Low-Motion video scene.



(b) Medium-Motion video scene.



(c) High-motion video scene.

Figure 5.8: SSIM for different offered loads in topologies with 60 nodes according to the level of video motion.

5.5.2 Network Density

Since network density has direct impact on path diversity and thus can influence multipath selection mechanisms, we ran experiments varying the number of deployed nodes using the same 300 $m \times 300$ m area. Figure 5.7 shows the average SSIM (mean and 95% confidence intervals) obtained by each mechanism as a function of the number of nodes under 4 Mb/s offered load for different motion levels. In general, all mechanisms' SSIM tend to improve as the number of nodes increases because of the higher path diversity. Position-based approaches, in particular, leverage high network density but might not accommodate multiple video sources, which explains the difficulty of RTVP and Q-MMTP in finding paths for all flows, resulting in lower performance. FITPATH and QSOpt, on the other hand, are relatively immune to the effects of network density since their approaches are based on the current conditions of the underlying network, regardless of the position of the nodes.

5.5.3 Offered Load

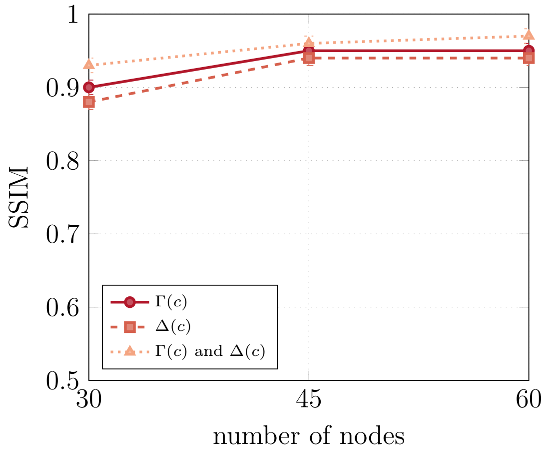
Results (mean and 95% confidence intervals) showing the impact of the total offered load on video quality are presented in Figure 5.8. In these experiments, we varied the offered load from 1 Mb/s to 8 Mb/s for all scenarios with 60 nodes. As expected, video quality decreases with increased offered load, especially for videos with higher scene motion levels, since it results in higher contention and consequently more collisions and longer queuing delays. Once again, the results show that FITPATH outperforms the other path selection mechanisms in all evaluated scenarios. Moreover, the performance gap between FITPATH and the other mechanisms generally grows for scenarios with higher offered load and higher motion levels. This can be explained by the fact that FITPATH uses MAPE's network performance estimation strategy which accounts for flow interference. As such, it is able to select paths that are better suited to handle the offered load and, therefore, maximize video quality. This result is also consistent with our previous observation that FITPATH is able to deliver adequate performance under challenging network conditions and more stringent application requirements.

5.5.4 Heuristic Evaluation

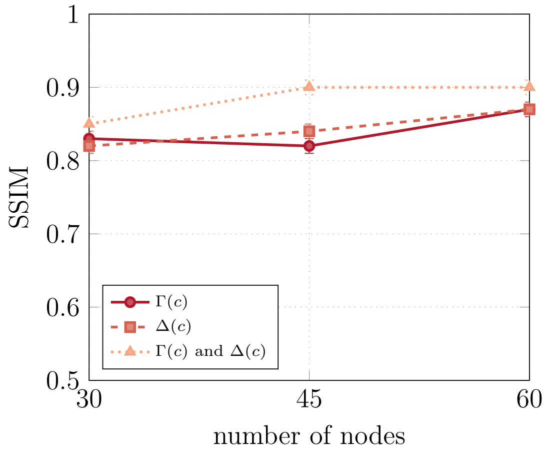
In order to define the FITPATH's parameters, we evaluated the heuristic using different strategies for the objective function, pruning, video coding and convergence time, as described below.

5.5.4.1 Objective Function

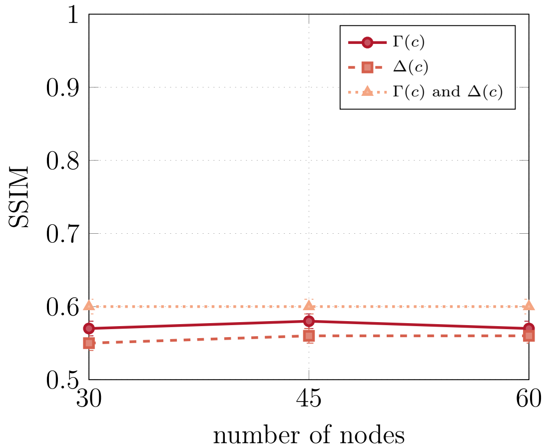
Defining an objective function as a routing metric, we evaluated the correlation between throughput and delay (given by Equations 5.1 and 5.2), metrics estimated by MAPE that most influence video quality. Figure 5.9 shows that using throughput with end-to-end delay as a tie-breaking criteria was more effective in the evaluated scenarios. This demonstrates that the video quality is more sensitive to packet losses — which manifest themselves on the throughput —, but delay must also be considered when the paths have a similar throughput.



(a) Low-Motion video scene.



(b) Medium-Motion video scene.

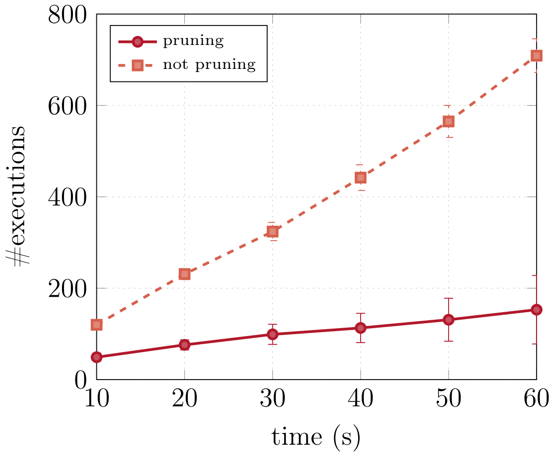


(c) High-motion video scene.

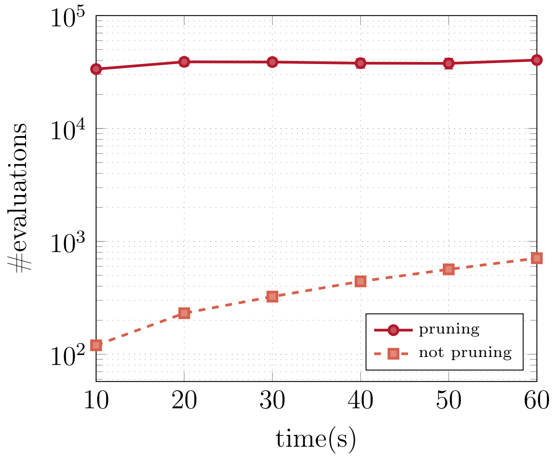
Figure 5.9: Objective function evaluation considering SSIM for different network densities and different video scene motion levels under 4 Mb/s offered load.

5.5.4.2 Pruning

As discussed in Section 5.3.1, FITPATH implements a pruning strategy to cover the largest possible fraction of the search space, considering an acceptable convergence time as MAPE executions correspond to most of the computation time during the heuristic's iterations. To minimize the unnecessary number of MAPE runs, FITPATH prunes all solutions with a throughput estimated by ETX lower than the best solution throughput estimated by MAPE. Figure 5.10a shows the number of MAPE executions while Figure 5.10b shows the number of candidate solutions in P evaluated by pruning and not pruning. These results demonstrate that it is possible to evaluate thousands of solutions more when applying the pruning strategy of pre-evaluating solutions by considering only paths ETX, thus minimizing MAPE runs.



(a) MAPE executions.



(b) Solution Evaluations.

Figure 5.10: Pruning performance.

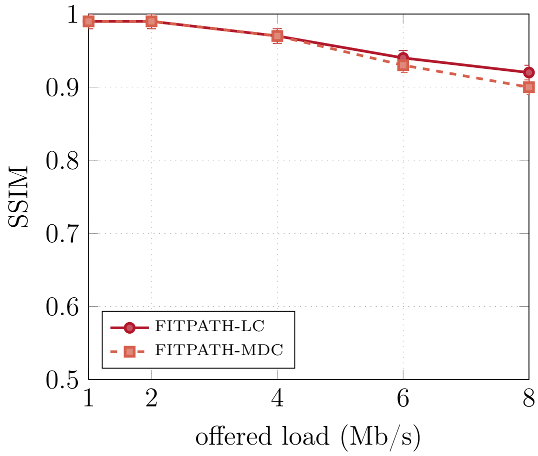
5.5.4.3 Video Coding

As FITPATH selects paths based on the transmission bitrate at which the flows are generated, it is able to deal using different video compression techniques – independent of the video encoding. Figure 5.11 demonstrates the similarity between video qualities for FITPATH using both LC and MDC techniques, considering the different scenarios evaluated.

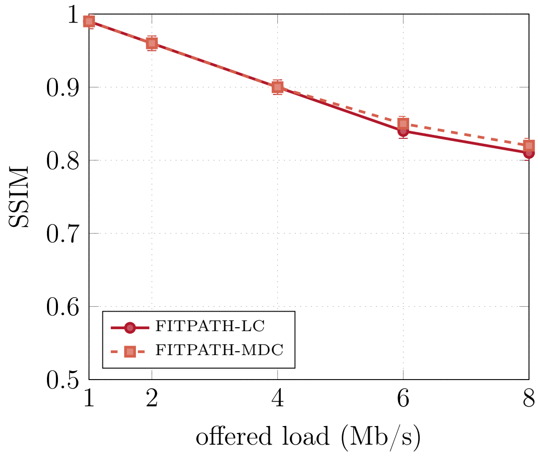
We also run the same experiments for related works and observed that proposals that were originally designed for working with LC, such as ERVT, CLMR and RTVP, may have worse performance by using MDC. This is because such mechanisms prioritize I-frame flows when selecting primary and secondary paths. This performance loss is more pronounced for high-motion video as the secondary paths tend not to withstand the bursts of scene changes.

5.5.4.4 Convergence Time

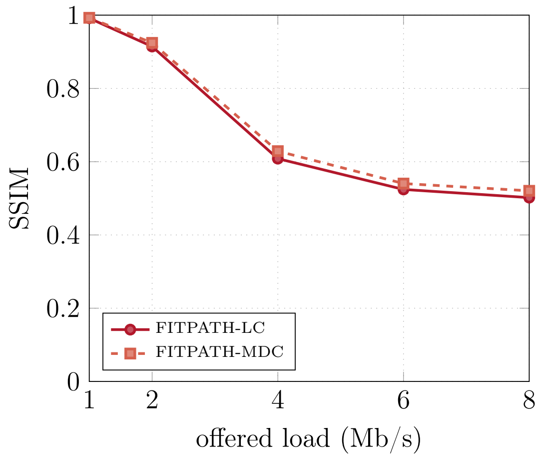
Since FITPATH uses a heuristic-based iterative algorithm, it is important to study how long it takes to converge and how convergence impacts video quality. To that end, we analyze how the SSIM of the solutions found by FITPATH evolve over its execution period — limited to 60 seconds. This kind of analysis can also be used to help define the ideal termination criterion. In fact, Figure 5.12 shows SSIM's mean and 95% confidence intervals of the current solution found by FITPATH at different execution moments for all scenarios with 4 Mb/s offered load and 60 node. As expected, FITPATH's solutions



(a) Low-Motion video scene.



(b) Medium-Motion video scene.



(c) High-motion video scene.

Figure 5.11: Video coding evaluation considering SSIM for different offered loads in topologies with 60 nodes according to the level of video motion.

generally improve with time, especially for videos with higher scene motion levels. However, in this particular scenario with 60 nodes and 4 Mb/s offered load, we observe that relatively little improvement is achieved after 40 seconds of execution regardless of the level of motion. Furthermore, after 5 seconds, FITPATH already found solutions whose average SSIM is similar to the one obtained by QSOpt, as shown by the reference blue lines in the graph. Note that QSOpt was the second best performer, behind only FITPATH, in all previous experiments. Moreover, QSOpt itself is a heuristic-based approach and takes between 5 to 10 seconds to obtain a solution for this scenario with 60 nodes — its execution time increases with the number of nodes and flows. However, differently from FITPATH, QSOpt does not provide intermediate solutions, only its final solution when its execution terminates, which is why we see a constant SSIM value for it in Figure 5.12.

FITPATH must run whenever network topology or video requirements change, ac-

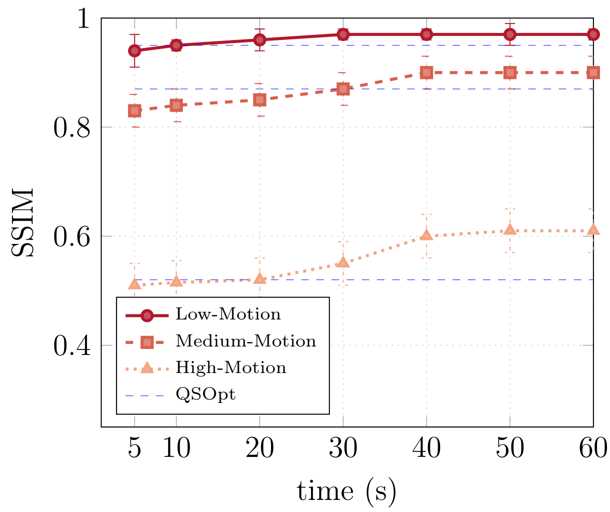


Figure 5.12: FITPATH convergence: SSIM evolution over time for scenarios with 4 Mb/s offered load and 60 nodes.

cording to the application and network dynamics. However, as discussed in Section 5.3.3, solutions obtained by FITPATH in its initial iterations can still be used by the video sources to start transmission and paths can be updated as new solutions are found.

Chapter 6

Conclusion

Multipath video transmission has been fundamental for improving the user's QoE in multihop wireless networks. However, the multipath selection mechanism requires approaches to find solutions that maximize the users' QoE considering flow interference from multiple video sources.

This thesis presents a thorough review of the current state-of-the-art in wireless multipath video transmission and provides solutions in order to answer the research questions raised in Section 1.3.

Considering Question **Q3** (How to provide an efficient interplay between sources' video coding and multipath routing?), we provide a cross-layer multipath routing framework considering different network architectures. This framework includes the proposal of the MAPE and the FITPATH mechanisms which use information of application and network layer for multipath selection.

In order to answer Question **Q2** (How to estimate the users' QoE from multiple video sources transmitting simultaneously, considering the inter-flow interference, QoS constraints and video services demands?), we propose the MAPE, a per-flow performance estimator based on a deterministic discrete event simulation approach. MAPE estimates the throughput, packet loss and end-to-end delay of individual flows using their average transmission rate as input. To the best of our knowledge, MAPE is the first performance estimator that is able to both account for inter-flow interference and accommodate rate-heterogeneous flows, which is essential to more realistically model the behavior of multimedia traffic.

We evaluated MAPE in terms of execution time, prediction accuracy and ability to classify sets of paths according to the video quality at the receiver. Results indicate that MAPE yields comparable throughput, packet loss and delay estimate accuracy when compared to stochastic network simulators such as *ns-3* at a fraction of the execution time. When compared to AFTER, through its ability to consider specific per-flow rates, MAPE yields higher accuracy at comparable execution times. We also show in practice that by adopting video coding rates as input, MAPE is able to obtain estimates similar to the ones obtained by ns-3 when driven by multimedia (MM) traffic. We also demonstrate how MAPE's accurate real-time throughput and delay predictions can be used to make routing decisions for multimedia applications. In particular, we show that MAPE makes correct path selection decisions for over 80% of the cases, including saturated network scenarios.

Finally, considering Question **Q1** (How to select the optimal paths for each video flow that maximizes the users' QoE?), we proposed FITPATH, an efficient and adaptable multipath routing path selection scheme that can accommodate environments where multiple video sources may simultaneously transmit flows of different bitrates. FITPATH uses a novel heuristic-based iterative optimization approach that estimates in real time the conditions of the underlying network while accounting for the different bitrate requirements of the application video flows.

Our experimental results that consider application scenarios that resemble realistic IoT deployments show that FITPATH outperforms various existing path selection mechanisms both in terms of user QoE and network performance. We also evaluated FITPATH's convergence behavior to demonstrate that it can be used in practice by video transmission sources to quickly generate feasible solutions that result in adequate video quality while incrementally improving QoE as the algorithm iterates.

6.1 Contributions

In summary, this thesis includes the proposal of an efficient and adaptable cross-layer multipath selection approach for IoT video transmission, presenting the following contributions:

- A survey in wireless multipath video transmission focusing on IoT applications, introducing a taxonomy to classify existing approaches based on their application-specific mechanisms as well as networking-specific techniques;
- A cross-layer multipath routing framework for IoT video transmission integrating the

video coding to the selection mechanism for centralized and decentralized network architecture;

- MAPE, the first deterministic simulation-based estimator that provides per-flow throughput, packet loss and delay estimates while considering inter-flow interference and flows with specific transmission rates, typical of video traffic; and
- FITPATH, a novel heuristic-based iterative path selection scheme for multipath routing that can accommodate environments where multiple video sources can simultaneously transmit heterogeneous video flows.

6.2 Future Work

As future work, we plan to refine MAPE's packet loss and delay models which will help improve its estimation accuracy. While our current implementation uses IEEE 802.11, we also plan to extend MAPE so that it can also be used with other IoT communication technologies such as IEEE 802.15.4 networks. We also intend to further explore the correlation between routing metrics and video quality (e.g., based on the SSIM) and to incorporate into MAPE multicast transmission as well as alternate ways to model variable bitrate, including traffic patterns representative of prominent adaptive bitrate streaming, e.g., by simulating video frame packets bursts. Additionally, we plan to evaluate MAPE's accuracy compared to a real-world testbed.

Furthermore, we intend to evaluate different FTIPATH's objective functions, e.g., accounting for other network performance metrics such as packet loss, as well as other heuristics, e.g., more elaborate local search approaches based on level of path disjointness. We also plan to propose objective functions to assess solutions that adjust to dynamic coding requirements. FITPATH can provide feedback in support of adaptive video coding and prioritize the paths to forward the video packets. Thus, the video compression bitrates could be adapted according to the selected paths' capacity and the flows forwarded by proper paths considering video packets' characteristics.

Finally, we plan to implement a proactive link-state routing protocol to test the behavior of the proposed framework under dynamic topology and network conditions and deploy it in a real wireless video network testbed.

References

[1] Abazeed, M.; Faisal, N.; Ali, A. Cross-layer multipath routing scheme for wireless multimedia sensor network. *Wireless Networks 6* (2018).

[2] Abd El Kader, M. E. E. D.; Youssif, A. A.; Ghalwash, A. Z. Energy Aware and Adaptive Cross-Layer Scheme for Video Transmission Over Wireless Sensor Networks. *IEEE Sensors Journal 16*, 21 (2016), 7792–7802.

[3] Aguilar Igartua, M.; Carrascal Frías, V. Self-configured multipath routing using path lifetime for video-streaming services over Ad Hoc networks. *Computer Communications 33*, 15 (2010), 1879–1891.

[4] Ahmed, A. A. A real-time routing protocol with adaptive traffic shaping for multimedia streaming over next-generation of Wireless Multimedia Sensor Networks. *Pervasive and Mobile Computing 40* (2017), 495–511.

[5] Akyildiz, I. F.; Melodia, T.; Chowdury, K. R. Wireless multimedia sensor networks: A survey, dec 2007.

[6] Al-Ariki, H. D. E.; Swamy, M. N. A survey and analysis of multipath routing protocols in wireless multimedia sensor networks. *Wireless Networks 23*, 6 (2017), 1823–1835.

[7] Al-quran, F. Energy-Efficient and QoS-Aware Multi-Path Geographic Routing Protocol for WMSN. *2018 IEEE Globecom Workshops (GC Wkshps)* (2018), 1–6.

[8] Alvi, S. A.; Afzal, B.; Shah, G. A.; Atzori, L.; Mahmood, W. Internet of multimedia things: Vision and challenges. *Ad Hoc Networks 33* (2015), 87–111.

[9] Aswale, S.; Ghorpade, V. R. Survey of QoS Routing Protocols in Wireless Multimedia Sensor Networks. *Journal of Computer Networks and Communications 2015* (2015), 1–29.

[10] Aswale, S.; Ghorpade, V. R. EEMGR: Energy and ETX aware Multipath Geographic Routing in WMSN. *Proceedings - 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018* (2018), 1–6.

[11] Aswale, S.; Ghorpade, V. R. Geographic Multipath Routing based on Triangle Link Quality Metric with Minimum Inter-path Interference for Wireless Multimedia Sensor Networks. *Journal of King Saud University - Computer and Information Sciences* (2018), 1–12.

[12] Barakabitze, A. A.; Mkwawa, I. H.; Sun, L.; Ifeachor, E. QualitySDN: Improving Video Quality using MPTCP and Segment Routing in SDN/NFV. *2018 4th*

IEEE Conference on Network Softwarization and Workshops, NetSoft 2018 (2018), 10–18.

[13] Barakabitze, A. A.; Sun, L.; Mkwawa, I. H.; Ifeachor, E. A Novel QoE-Centric SDN-Based Multipath Routing Approach for Multimedia Services over 5G Networks. *IEEE International Conference on Communications 2018-May* (2018), 1–7.

[14] Begen, A. C.; Altunbasak, Y.; Ergun, O. Fast heuristics for multi-path selection for multiple description encoded video streaming. *Proceedings - IEEE International Conference on Multimedia and Expo 1* (2003), I517–I520.

[15] Begen, A. C.; Altunbasak, Y.; Ergun, O. Multi-path selection for multiple description encoded video streaming. In *IEEE International Conference on Communications (ICC)* (2003), vol. 3, IEEE, pp. 1583–1589.

[16] Bennis, I.; Fouchal, H.; Piamrat, K.; Zytoune, O.; Aboutajdine, D. A realistic multipath routing for Ad Hoc networks. *2015 IEEE Global Communications Conference, GLOBECOM 2015* (2015), 1–6.

[17] Bentaleb, A.; Timmerer, C.; Begen, A. C.; Zimmermann, R. Bandwidth prediction in low-latency chunked streaming. In *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (2019), pp. 7–13.

[18] Bhatt, R.; Datta, R. A two-tier strategy for priority based critical event surveillance with wireless multimedia sensors. *Wireless Networks 22*, 1 (2016), 267–284.

[19] Bhering, F.; Albuquerque, C.; Passos, D.; Obraczka, K. Iamvt: Mecanismo de seleção de rotas multicaminhos para aplicações de vídeo monitoramento. In *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos* (2021), SBC, pp. 294–307.

[20] Bhering, F.; Albuquerque, C.; Passos, D.; Ochi, L. S. Multipath selection mechanism for wireless video-surveillance systems. *9th Latin American Network Operations and Management Symposium, LANOMS* (2019).

[21] Bhering, F.; Passos, D.; Albuquerque, C.; Obraczka, K. Efficient multipath selection for iot video transmission. In *IEEE Global Internet Symposium* (2022), IEEE CloudNet Conference.

[22] Bhering, F.; Passos, D.; Albuquerque, C.; Obraczka, K. Fitpath: Multipath selection mechanism for iot video transmission. *Multimedia Systems* (2022), In prep.

[23] Bhering, F.; Passos, D.; Albuquerque, C.; Obraczka, K. Network performance estimator with applications to route selection for iot multimedia applications. *SIMULATION* (2022), Under review.

[24] Bhering, F.; Passos, D.; Ochi, L. S.; Obraczka, K.; Albuquerque, C. Wireless multipath video transmission: when iot video applications meet networking—a survey. *Multimedia Systems* (2022), 1–20.

[25] Biernacki, A. Improving quality of adaptive video by traffic prediction with (f) arima models. *Journal of communications and networks 19*, 5 (2017), 521–530.

[26] Campista, M. E.; Esposito, P. M.; Moraes, I.; Costa, L. H.; Duarte, O. C.; Passos, D.; Albuquerque, C.; Muchaluat-Saade, D. C.; Rubinstein, M. G. Routing metrics and protocols for wireless mesh networks. *IEEE network 22*, 1 (2008), 6–12.

[27] Cao, Y.; Liu, Q.; Luo, G.; Yi, Y.; Huang, M. PR-MPTCP+: Context-aware QoE-oriented multipath TCP partial reliability extension for real-time multimedia applications. *VCIP 2016 - 30th Anniversary of Visual Communication and Image Processing* (2017), 1–4.

[28] Chakareski, J.; Han, S.; Girod, B. Layered coding vs. multiple descriptions for video streaming over multiple paths. *Proceedings of the ACM International Multimedia Conference and Exhibition 285* (2003), 422–431.

[29] Chenna, R. B.; Singhal, C.; De, S. MDTC video transmission using channel aware multipath routing in wireless mesh networks. *ACM International Conference Proceeding Series* (2018).

[30] Chikh, A.; Lehsaini, M. Multipath routing protocols for wireless multimedia sensor networks: a survey. *International Journal of Communication Networks and Distributed Systems 20*, 1 (2018), 60–81.

[31] Cisco. New cisco annual internet report forecasts 5g to support more than 10% of global mobile connections by 2023. *The News Room* (2020).

[32] Clausen, T.; Jacquet, P.; Adjih, C.; Laouiti, A.; Minet, P.; Muhlethaler, P.; Qayyum, A.; Viennot, L. Optimized link state routing protocol (olsr). *RFC Editor* (2003).

[33] Colombo, C.; Lepage, F.; Kopp, R.; Gnaedinger, E. SHERPA: A SDN multipath approach to eliminate resilience impact on video streams. *International Conference on Communication Technology Proceedings, ICCT 2019-Octob* (2019), 1357–1362.

[34] Correia, P.; Ferreira, L.; Assuncao, P. A.; Cruz, L.; Silva, V. Optimal priority MDC video streaming for networks with path diversity. *2012 International Conference on Telecommunications and Multimedia, TEMU 2012* (2012), 54–59.

[35] Cui, H.; Sun, Q.; Zhang, B.; Zhang, X.; Wu, W. Video-Aware Multipath Network Coding over Multi-Hop Wireless Networks. *International Conference on Information Networking 2019-Jan* (2019), 171–175.

[36] Dao, N. Q.; Le-Dang, Q.; Morawski, R.; Dang, A. T.; Le-Ngoc, T. Management of video surveillance for smart cities. *Handbook of Smart Cities: Software Services and Cyber Infrastructure* (2018), 285–310.

[37] De Couto, D. S.; Aguayo, D.; Bicket, J.; Morris, R. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking* (2003), pp. 134–146.

[38] Deb, B.; Bhatnagar, S.; Nath, B. ReInForM: Reliable information forwarding using multiple paths in sensor networks. *Proceedings - Conference on Local Computer Networks, LCN 2003-jan* (2003), 406–415.

[39] E., S.; V. P. Sundararajan, T. Improved Video Transmission over Mobile Adhoc Networks using AOMDV and MDC with Path Diversity. *International Journal of Computer Applications 62*, 17 (2013), 35–42.

[40] Ehsan, S.; Hamdaoui, B. A survey on energy-efficient routing techniques with QoS assurances for wireless multimedia sensor networks. *IEEE Communications Surveys and Tutorials 14*, 2 (2012), 265–278.

[41] El Dien, M. E.; Youssif, A. A. A.; Ghalwash, A. Z. Energy Efficient and QoS Aware Framework for Video Transmission over Wireless Sensor Networks. *Wireless Sensor Network 08*, 03 (2016), 25–36.

[42] Elgabli, A.; Felemban, M.; Aggarwal, V. GroupCast: Preference-Aware Cooperative Video Streaming with Scalable Video Coding. In *IEEE/ACM Transactions on Networking* (2019), vol. 27, IEEE, pp. 1138–1150.

[43] Elgabli, A.; Liu, K.; Aggarwal, V. Optimized preference-aware multi-path video streaming with scalable video coding. *IEEE Transactions on Mobile Computing 9* (2018).

[44] Felemban, E.; Lee, C. G.; Ekici, E. MMSPEED: Multipath Multi-SPEED Protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing 5*, 6 (2006), 738–753.

[45] Ford, A.; Raiciu, C.; Handley, M.; Barre, S.; Iyengar, J., et al. Architectural guidelines for multipath tcp development. *IETF, Informational RFC 6182* (2011), 2070–1721.

[46] Gadicha, A. B.; Gupta, V. B. B.; Gadicha, V. B.; Kumar, K.; Ghonge, M. M. Multimode approach of data encryption in images through quantum steganography. In *Multidisciplinary Approach to Modern Digital Steganography.* IGI Global, 2021, pp. 99–124.

[47] Garetto, M.; Salonidis, T.; Knightly, E. W. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. *IEEE/ACM Transactions on Networking 16*, 4 (2008), 864–877.

[48] Ghahremani, S.; Ghanbari, M. Error resilient video transmission in ad hoc networks using layered and multiple description coding. *Multimedia Tools and Applications 76*, 6 (2017), 9033–9049.

[49] Ghareeb, M.; Viho, C. Hybrid QoE assessment is well-suited for multiple description coding video streaming in overlay networks. In *Eighth Annual Communication Networks and Services Research Conference (CNSR)* (2010), IEEE, pp. 327–333.

[50] Goyal, V. K. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine 18*, 5 (2001), 74–93.

[51] Guo, Y.; Chen, Y.; Wang, Y. K.; Li, H.; Hannuksela, M. M.; Gabbouj, M. Error resilient coding and error concealment in scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology 19*, 6 (2009), 781–795.

[52] Harjito, B.; Han, S. Wireless multimedia sensor networks applications and security challenges. *Proceedings - 2010 International Conference on Broadband, Wireless Computing Communication and Applications, BWCCA 2010* (2010), 842–846.

[53] Hasan, M. Z.; Al-Rizzo, H.; Al-Turjman, F. A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Wireless Multimedia Sensor Networks. *IEEE Communications Surveys and Tutorials 19*, 3 (2017), 1424–1456.

[54] Hasan, M. Z.; Al-Turjman, F.; Al-Rizzo, H. Optimized Multi-Constrained Quality-of-Service Multipath Routing Approach for Multimedia Sensor Networks. *IEEE Sensors Journal 17*, 7 (2017), 2298–2309.

[55] He, T.; Stankovic, J. A.; Lu, C.; Abdelzaher, T. Speed: A stateless protocol for real-time communication in sensor networks. Tech. rep., Virginia Univ. Charlottesville Dept of Computer Science, 2003.

[56] Hossain, M. S.; You, X.; Xiao, W.; Lu, J.; Song, E. QoS-oriented multimedia transmission using multipath routing. *Future Generation Computer Systems 99* (2019), 226–234.

[57] Hu, Y.; Jianguang, S.; Fangfang, S.; Zhuoxian, Z. Delay constraint multipath routing for wireless multimedia ad hoc networks. *International Journal of Communication Systems* (2016).

[58] Huo, Y.; Hellge, C.; Wiegand, T.; Hanzo, L. A tutorial and review on inter-layer fec coded layered video streaming. *IEEE Communications Surveys and Tutorials 17*, 2 (2015), 1166–1207.

[59] Jiang, J.; Sekar, V.; Zhang, H. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking 22*, 1 (2014), 326–340.

[60] Johnson, D. B.; Maltz, D. A.; Broch, J., et al. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking 5*, 1 (2001), 139–172.

[61] K, P. K.; Govindaraj, E. Quality enhancement with fault tolerant embedding in video transmission over WMSNs in 802.11e WLAN. *Ad Hoc Networks 88* (2019), 18–31.

[62] Karn, N. K.; Zhang, H.; Jiang, F.; Yadav, R.; Laghari, A. A. Measuring bandwidth and buffer occupancy to improve the qoe of http adaptive streaming. *Signal, Image and Video Processing 13*, 7 (2019), 1367–1375.

[63] Ke, C.-H.; Lin, K.-W.; Huang, C.-A. Inter-flow fairness support and enhanced video quality delivery over multi-hop wireless networks. *Computers & Electrical Engineering 56* (2016), 193–204.

[64] Khan, K.; Goodridge, W. Multi-criterion decision making and adaptation for multi-path video streaming in wsns. *International Journal of Advanced Networking and Applications 9*, 2 (2017), 3376–3381.

[65] Khernane, N.; Couchot, J. F.; Mostefaoui, A. Maximum network lifetime with optimal power/rate and routing trade-off for Wireless Multimedia Sensor Networks. *Computer Communications 124*, March (2018), 1–16.

[66] Klaue, J.; Rathke, B.; Wolisz, A. Evalvid–a framework for video transmission and quality evaluation. In *International conference on modelling techniques and tools for computer performance evaluation* (2003), Springer, pp. 255–272.

[67] Koppanati, R. K.; Kumar, K.; Qamar, S. E-moc: An efficient secret sharing model for multimedia on cloud. In *International Conference on Deep Learning, Artificial Intelligence and Robotics* (2019), Springer, pp. 246–260.

[68] Kreutz, D.; Ramos, F.; Verissimo, P.; Rothenberg, C. E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *arXiv preprint arXiv:1406.0440* (2014).

[69] Krishna, R.; Kumar, K. P-mec: polynomial congruence based multimedia encryption technique over cloud. *IEEE Consumer Electronics Magazine* (2020).

[70] Kulkarni, P.; Ganesan, D.; Shenoy, P.; Lu, Q. Senseye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia* (2005), ACM, pp. 229–238.

[71] Kumar, K. Text query based summarized event searching interface system using deep learning over cloud. *Multimedia Tools and Applications 80*, 7 (2021), 11079–11094.

[72] Kumar, K.; Shrimankar, D. D. F-des: Fast and deep event summarization. *IEEE Transactions on Multimedia 20*, 2 (2017), 323–334.

[73] Kumar, K.; Shrimankar, D. D.; Singh, N. Equal partition based clustering approach for event summarization in videos. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (2016), IEEE, pp. 119–126.

[74] Kumar, K.; Shrimankar, D. D.; Singh, N. Eratosthenes sieve based keyframe extraction technique for event summarization in videos. *Multimedia Tools and Applications 77*, 6 (2018), 7383–7404.

[75] Kumar, K.; Sinha, S.; Manupriya, P. D-pnr: Deep license plate number recognition. In *Proceedings of 2nd International Conference on Computer Vision & Image Processing* (2018), Springer, pp. 37–46.

[76] Lal, C.; Laxmi, V.; Gaur, M. S.; Conti, M. Enhancing QoE for video streaming in MANETs via multi-constraint routing. *Wireless Networks 24*, 1 (2018), 235–256.

[77] Laufer, R.; Kleinrock, L. The capacity of wireless csma/ca networks. *IEEE/ACM Transactions on Networking 24*, 3 (2015), 1518–1532.

[78] Le Gall, D. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM 34*, 4 (1991), 46–58.

[79] Lemamou, E. A.; Galinier, P.; Chamberland, S. A hybrid iterated local search algorithm for the global planning problem of survivable 4g wireless networks. *IEEE/ACM Transactions on Networking (TON) 24*, 1 (2016), 137–148.

[80] Licandro, F.; Lombardo, A.; Schembra, G. Multipath routing and rate-controlled video encoding in wireless video surveillance networks. *Multimedia Systems 14*, 3 (2008), 155–165.

[81] Liew, S. C.; Kai, C. H.; Leung, H. C.; Wong, P. Back-of-the-envelope computation of throughput distributions in csma wireless networks. *IEEE Transactions on Mobile Computing 9*, 9 (2010), 1319–1331.

[82] Liu, Z.; Cheung, G.; Chakareski, J.; Ji, Y. Multiple description coding and recovery of free viewpoint video for wireless multi-path streaming. *IEEE Journal of Selected Topics in Signal Processing 9*, 1 (2015), 151–164.

[83] Lourenço, H. R.; Martin, O. C.; Stützle, T. Iterated local search: Framework and applications. In *Handbook of metaheuristics*. Springer, 2010, pp. 363–397.

[84] Manupriya, P.; Sinha, S.; Kumar, K. V see: Video secret sharing encryption technique. In *2017 conference on information and communication technology (CICT)* (2017), IEEE, pp. 1–6.

[85] Marina, M. K.; Das, S. R. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings Ninth International Conference on Network Protocols. ICNP 2001* (2001), IEEE, pp. 14–23.

[86] Marvasti-Zadeh, S. M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems* (2021).

[87] Nabi, S.; Farooq, M. U.; Hussain, F. Shanz algorithm for qoe enhancement of http based adaptive video streaming. In *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)* (2019), pp. 393–400.

[88] Natarajan, P.; Atrey, P. K.; Kankanhalli, M. Multi-camera coordination and control in surveillance systems: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 11*, 4 (2015), 57.

[89] Orosz, P.; Skopkó, T.; Nagy, Z.; Varga, P.; Gyimóthi, L. A case study on correlating video qos and qoe. In *2014 IEEE Network Operations and Management Symposium (NOMS)* (2014), IEEE, pp. 1–5.

[90] Pan, C.; Liu, B.; Zhou, H.; Gui, L. Multi-path routing for video streaming in multi-radio multi-channel wireless mesh networks. *2016 IEEE International Conference on Communications, ICC 2016 11* (2016), 73–92.

[91] Passos, D.; Albuquerque, C. AFTER: Algorithmic Framework for Throughput EstimatoRs for IEEE 802.11 networks. *Simulation Modelling Practice and Theory 84* (2018), 143–160.

[92] Penna, P. H. V.; Subramanian, A.; Ochi, L. S.; Vidal, T.; Prins, C. A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. *Annals of Operations Research* (2017), 1–70.

[93] Perkins, C. E.; Royer, E. M. Ad-hoc on-demand distance vector routing. In *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications* (1999), IEEE, pp. 90–100.

[94] Pham, T. A. Q.; Singh, K. D.; Rodríguez-Aguilar, J. A.; Picard, G.; Piamrat, K.; Cerquides, J.; Viho, C. AD 3 -GLAM: A cooperative distributed QoE-based approach for SVC video streaming over wireless mesh networks. *Ad Hoc Networks 80* (2018), 1–15.

[95] Quang, P. T. A.; Piamrat, K.; Deep Singh, K.; Viho, C. Q-SWiM: QoE-based routing algorithm for SVC video streaming over wireless mesh networks. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC* (2016), 1–6.

[96] Quang, P. T. A.; Piamrat, K.; Singh, K. D.; Viho, C. Video Streaming over Ad Hoc Networks: A QoE-Based Optimal Routing Solution. *IEEE Transactions on Vehicular Technology 66*, 2 (2017), 1533–1546.

[97] Reddy, T. B.; Karthigeyan, I.; Manoj, B.; Murthy, C. S. R. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad Hoc Networks 4*, 1 (2006), 83–124.

[98] Riley, G. F.; Henderson, T. R. The ns-3 network simulator. In *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[99] Rumbaugh, J.; Jacobson, I.; Booch, G. *Unified modeling language reference manual, the*. Pearson Higher Education, 2004.

[100] Santos, M. A.; Porras, D. E.; Silveira, R. M.; Margi, C. B. Multipath source routing strategies for video transmission in ad hoc wireless networks. *Wireless Networks 21*, 3 (2015), 859–869.

[101] Sedrati, M.; Benyahia, A. Multipath Routing to Improve Quality of Service for Video Streaming Over Mobile Ad Hoc Networks. *Wireless Personal Communications 99*, 2 (2018), 999–1013.

[102] Seeling, P.; Reisslein, M. Video transport evaluation with h. 264 video traces. *IEEE Communications Surveys & Tutorials 14*, 4 (2011), 1142–1165.

[103] Sharma, S.; Kumar, K. Guess: Genetic uses in video encryption with secret sharing. In *Proceedings of 2nd international conference on computer vision & image processing* (2018), Springer, pp. 51–62.

[104] Shen, H.; Bai, G. Routing in wireless multimedia sensor networks: A survey and challenges ahead. *Journal of Network and Computer Applications 71* (2016), 30–49.

[105] Silva, M. M.; Subramanian, A.; Ochi, L. S. An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research 53* (2015), 234–249.

[106] Singh, V.; Ahsan, S.; Ott, J. MPRTP: Multipath considerations for real-time media. *Proceedings of the 4th ACM Multimedia Systems Conference, MMSys 2013* (2013), 190–201.

[107] Sohrabi, K.; Gao, J.; Ailawadhi, V.; Pottie, G. J. Protocols for self-organization of a wireless sensor network. *IEEE personal communications 7*, 5 (2000), 16–27.

[108] Sreeram, K.; Unnisa, A.; Poornima, P.; Chaudhari, S. QoS aware Multi-Constrained Node Disjoint Multipath Routing for Wireless Sensor Networks. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019* (2019), 382–385.

[109] Stoffers, M.; Riley, G. Comparing the ns-3 propagation models. In *2012 IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2012), pp. 61–67.

[110] Sun, Y.; Yin, X.; Jiang, J.; Sekar, V.; Lin, F.; Wang, N.; Liu, T.; Sinopoli, B. Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 272–285.

[111] Thet, P. M.; Panwaree, P.; Kim, J.; Aswakul, C. Design and functionality test of chunked video streaming over emulated multi-path OpenFlow network. *ECTI-CON 2015 - 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* (2015), 1–6.

[112] Usman, M.; He, X.; Xu, M.; Lam, K. M. Survey of Error Concealment techniques: Research directions and open issues. *2015 Picture Coding Symposium, PCS 2015 - with 2015 Packet Video Workshop, PV 2015 - Proceedings* (2015), 233–238.

[113] Usman, M.; Yang, N.; Jan, M. A.; He, X.; Xu, M.; Lam, K. A joint framework for QoS and QoE for video transmission over wireless multimedia sensor networks. *IEEE Transactions on Mobile Computing 17*, 4 (April 2018), 746–759.

[114] Varga, A. The omnet++ discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001), Bd* (2001), vol. 9, p. 185.

[115] Wang, B.; Ren, F. Improving robustness of dash against network uncertainty. In *2019 IEEE International Conference on Multimedia and Expo (ICME)* (2019), pp. 748–753.

[116] Wang, H.; Ortega, A. Robust video communication by combining scalability and multiple description coding techniques. *Image and Video Communications and Processing 2003 5022*, May 2003 (2003), 111.

[117] Wang, M.; Cheng, B.; Yuen, C. Joint Coding-Transmission Optimization for a Video Surveillance System with Multiple Cameras. *IEEE Transactions on Multimedia 20*, 3 (2018), 620–633.

[118] Wang, X.; Kar, K. Throughput modelling and fairness issues in csma/ca based ad-hoc networks. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies.* (2005), vol. 1, Ieee, pp. 23–34.

[119] Wang, Y.; Reibman, A. R.; Lin, S. Multiple description coding for video delivery. *Proceedings of the IEEE 93*, 1 (2005), 57–69.

[120] Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612.

[121] Wei, B.; Kawakami, W.; Kanai, K.; Katto, J.; Wang, S. Trust: A tcp throughput prediction method in mobile networks. In *2018 IEEE Global Communications Conference (GLOBECOM)* (2018), IEEE, pp. 1–6.

[122] Wei, B.; Song, H.; Wang, S.; Kanai, K.; Katto, J. Evaluation of throughput prediction for adaptive bitrate control using trace-based emulation. *IEEE Access 7* (2019), 51346–51356.

[123] Wei, W.; Zakhor, A. Interference aware multipath selection for video streaming in wireless ad hoc networks. *IEEE Transactions on Circuits and Systems for Video Technology 19*, 2 (2009), 165–178.

[124] Wu, J.; Yuen, C.; Cheng, B.; Yang, Y.; Wang, M.; Chen, J. Bandwidth-Efficient Multipath Transport Protocol for Quality-Guaranteed Real-Time Video over Heterogeneous Wireless Networks. *IEEE Transactions on Communications 64*, 6 (2016), 2477–2493.

[125] Yang, C.; Wu, C.; Yang, Z.; Liu, T.; Yin, Z.; Liu, Y.; Mao, X. Enhancing industrial video surveillance over wireless mesh networks. *2016 25th International Conference on Computer Communications and Networks, ICCCN 2016* (2016).

[126] Yang, Y.; Zhong, C.; Sun, Y.; Yang, J. Network coding based reliable disjoint and braided multipath routing for sensor networks. *Journal of Network and Computer Applications 33*, 4 (2010), 422–432.

[127] Yen, J. Y. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics 27*, 4 (1970), 526–530.

[128] Youssif, A. A.; Ghalwash, A. Z.; Abd El Kader, M. E. E. D. ACWSN: an adaptive cross layer framework for video transmission over wireless sensor networks. *Wireless Networks 21*, 8 (2015), 2693–2710.

[129] Zhang, Y.; Zhan, A.; Dai, G.; Lin, Z.; Yang, Y. An Adaptive Multi-Path Video Streaming Scalable Video Coding Algorithm. *Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017 2* (2017), 261–264.