FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multipath Video Transport for Edge-Supported Cooperative Perception Services

Bernardo Carrilho

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Ana Aguiar

Second Supervisor: Pedro Santos

June 30, 2023

Resumo

Com o número crescente de utentes da estrada a adoptarem modos de transporte ambientalmente conscientes, como andar a pé e de bicicleta, garantir a segurança rodoviária torna-se cada vez mais crítico. Infelizmente, estes utilizadores vulneráveis da estrada estão frequentemente envolvidos em acidentes graves que poderiam ser evitados com a utilização de tecnologias como Cooperative Perception, um serviço que permite que entidades de tráfego próximas partilhem informações sobre o ambiente circundante, melhorando a perceção espacial. No entanto, o processamento de vídeo para esses serviços consome recursos significativos. Para responder a este desafio, foi introduzida a computação na edge para permitir um processamento de vídeo mais rápido. Consequentemente, a transmissão do vídeo para o nó de edge torna-se uma tarefa que consome muitos recursos. Assim, o objetivo desta dissertação é conceber um método para a distribuição de tráfego de vídeo entre múltiplas redes de acesso, com o objetivo de melhorar a latência ao nível da aplicação através da utilização de computação na edge. Para o efeito, foi criado um ambiente de teste, composto por um emissor, um recetor, e dois caminhos disponíveis: WiFi e 5G. O MA-RTP, solução do Instituto de Telecomunicações para multipath RTP, foi utilizado por se adequado a fluxos de dados em tempo real e tolerantes a perdas. Adicionalmente, o ambiente de teste inclui uma estação de monitorização de performance da rede que armazena dados históricos essenciais relacionados com localizações geográficas específicas, permitindo decisões de escalonamento informadas. Depois de definir as métricas de desempenho e as experiências, foram desenvolvidos, testados e avaliados vários escalonadores. Os resultados demonstram a possibilidade de distribuição eficiente de vídeo através da tomada de decisões de escalonamento informadas considerando as condições da rede. Esta dissertação contribui para o avanço de Cooperative Perception ao explorar o seu potencial para aumentar a segurança rodoviária dos utilizadores vulneráveis. Os resultados destacam a importância de considerar factores como a latência, a perda de pacotes e a comutação de caminhos para melhorar a transmissão de vídeo e melhorar a latência ao nível da aplicação. Ao abordar o desafio da distribuição de vídeo de ligação uplink entre várias redes de acesso, esta investigação fornece informações valiosas sobre as vantagens e limitações da integração da computação edge neste contexto.

Abstract

With the growing number of road users adopting environmentally conscious modes of transportation, such as walking and cycling, ensuring road safety becomes increasingly critical. Unfortunately, these vulnerable road users are often involved in severe accidents that could be prevented with the use of technologies like Cooperative Perception, a service that allows traffic entities in close proximity to share information about the surrounding environment, enhancing situational awareness. However, video processing for such services consumes significant resources. To address this challenge, edge computing has been introduced to enable faster video processing. Consequently, transmitting the video to the edge node becomes a resource-intensive task. Thus, the goal of this dissertation is to design a method for uplink distribution of video traffic between multiple access networks, with the purpose of enhancing the application level latency through the use of computation at the edge. For this purpose, a testing environment was established, comprising a sender, a receiver, and two available paths: WiFi and 5G. MA-RTP, Instituto de Telecomunicações' solution for multipath RTP, was employed due to its suitability for real-time, loss-tolerant data streams. Additionally, the testing environment comprises a network performance monitoring station that stores essential historic data related to specific geographical locations, enabling informed scheduling decisions. After defining the performance metrics and the experiments, various schedulers were developed, tested, and evaluated. The results demonstrate the possibility of efficient video distribution by making informed scheduling decisions considering network conditions. This dissertation contributes to the advancement of Cooperative Perception by exploring its potential in enhancing road safety for vulnerable road users. The findings highlight the significance of considering factors such as latency, packet loss, and path switching to improve video transmission and improve application-level latency. By addressing the challenge of uplink video distribution between multiple access networks, this research provides valuable insights into the benefits and limitations of edge computing integration within this context.

Keywords: Packet scheduling, decision making, multipath transport, video transmission, 5G, WiFi, edge computing, cooperative perception, offloading

Agradecimentos

à minha orientadora, professora Ana Aguiar, por me guiar no projeto mais complexo que já fiz

à equipa do Instituto de Telecomunicacões, por estar sempre disposta a ajudar no que eu precisar

ao Instituto de Telecomunicacões, por fornecer todo o material e espaço necessário para trabalhar

aos meus pais, pelo amor dado todos estes anos

aos meus irmãos, por estarem sempre presentes

ao Pedro, por finalmente descobrir quando faço anos (acho eu)

ao Diogo, por nunca se esquecer de mostrar as novas adições à playlist

ao Miguel, por saber sempre onde jantar (tirando aquela vez...)

ao Guilherme, porque o cinema não se paga sozinho

à Tânia, por me aturar durante 5 anos de aulas práticas (e algumas teóricas...)

E aos restantes amigos e familia por todo o apoio dado ao longo da minha vida

This work is a result of project FLOYD (POCI-01-0247-FEDER-045912), funded by the European Regional Development Fund (FEDER), through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) and by Portuguese National Funds (OE), through Fundação para a Ciência e Tecnologia, I.P.; and UIDB/50008/2020, funded by the applicable financial framework (FCT/MCTES) (PIDDAC).

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Problem Statement . . . 2
1.3 Document Structure . . . 4

2 Literature Review **5**
2.1 Background and Fundamental Aspects . . . 5
2.1.1 Edge Computing . . . 5
2.1.2 RTP and RTCP . . . 5
2.1.3 Multipath Transport . . . 6
2.2 Related Work . . . 9
2.2.1 Multipath Scheduling . . . 9
2.2.2 Performance Metrics . . . 10
2.2.3 Methodology . . . 11
2.3 Contributions . . . 13

3 Multipath Video Transport **15**
3.1 MA-RTP . . . 15
3.1.1 Protocol Description . . . 16
3.1.2 Implementation . . . 17
3.1.3 Limitations . . . 18
3.2 Network Performance Monitoring Station . . . 19
3.2.1 Implementation . . . 19
3.2.2 Populating the Station . . . 19
3.2.3 Limitations . . . 19
3.3 Work Approach . . . 20
3.4 Schedulers . . . 20

4 System Implementation **23**
4.1 Test Environment . . . 23
4.1.1 System Overview . . . 23
4.1.2 System Components . . . 25
4.1.3 Network Configuration . . . 26
4.1.4 Limitations . . . 28
4.2 Performance Metrics . . . 29
4.2.1 Measurements . . . 29
4.2.2 Data Logging . . . 30
4.2.3 File Structure . . . 31

4.2.4 Performance Analysis Methodology 32
4.2.5 Limitations 34

5 Experiments **35**
5.1 Experiment Rounds 36
5.2 Limitations 39

6 Results and Discussion **41**
6.1 Establish Baseline 42
6.2 Improving Delivery by Frame Type 45
6.3 Maximizing Round-Robin Efficiency 47
6.4 Finding Anomalies in User-Defined and Random Scheduling 50
6.5 Minimizing Packet Loss and Late Arrivals 52

7 Final Remarks **55**
7.1 Conclusions 55
7.2 Contributions 55
7.3 Future Work 56

References **59**

List of Figures

1.1 Cooperative Perception use case example. [1, Fig. 4.2] 2
1.2 Object detection delay. [1, Fig. 5.3] 3

2.1 Multipath usage scenario. Adapted from [2, Fig. 2] 7
2.2 Comparison of standard TCP and MP-TCP protocol stacks [3, Fig. 1] 7
2.3 Comparison of standard DCCP and MP-DCCP protocol stacks. [2, Fig. 1] 8
2.4 Comparison of standard RTP and MP-RTP protocol stacks. [4, Fig. 1] 8
2.5 Applications, schedulers, and their overall goals. [5, Fig. 3] 11

3.1 MA-RTP protocol stack 16
3.2 MA-RTP connections 17
3.3 MA-RTP architecture 18
3.4 Round-robin frame distribution 21
3.5 Round-robin based on GoP Size frame distribution 22
3.6 Round-robin based on Frame Size frame distribution 22

4.1 Setup scheme 23
4.2 Photographs of the experimental setup 24
4.3 Network layout scheme 27
4.4 Clock drift illustration 30
4.5 Comparison of latency without and with NTP synchronization, respectively . . . 30
4.6 Screenshot of logs 31
4.7 Depiction of the file structure used for data storage 31
4.8 Flowchart illustrating the data processing and analysis pipeline from logging to result presentation 32

5.1 Testing route 35

6.1 Redundant scheduler results 42
6.2 Round-robin scheduler results 43
6.3 5G-only scheduler results 43
6.4 WiFi-only scheduler results 44
6.5 I Frame to WiFi scheduler results 46
6.6 I Frame to 5G scheduler results 46
6.7 Round-robin based on GoP Size scheduler results 48
6.8 Round-robin based on Frame Size scheduler results 49
6.9 Random scheduler results 50
6.10 Zone based scheduler results 51
6.11 Packet Loss Rate scheduler results 52

6.12 Timely Arrival scheduler results 53

List of Tables

2.1 Summary of various packet scheduling solutions proposed from the public releases of MP-TCP Linux kernel (2013). Adapted from [5, Tab. I] 12

4.1 Metric calculation variables' descriptions 33

5.1 Testing rounds' schedule 38
5.2 Identifier / Scheduler pairs 39

6.1 Average results of each scheduler - Establishing Baseline 45
6.2 Average results of each scheduler - Improving Delivery by Frame Type 47
6.3 Average results of each scheduler - Maximizing Round-Robin Efficiency 49
6.4 Average results of each scheduler - Finding Anomalies in Arbitrary and Random Scheduling 51
6.5 Average results of each scheduler - Minimizing Packet Loss and Late Arrivals . . 53
6.6 Average results for all schedulers 54

Abbreviations

AP	Access Point
CPM	Cooperative Perception Message
DCCP	Datagram Congestion Control Protocol
HoL	Head of Line
LL	Lowest Latency
MP	Multipath
RP	Raspberry Pi
RTP	Real-time Transport Protocol
RTT	Round Trip Time
TCP	Transmission Control Protocol
V2X	Vehicle-to-Everything
VRU	Vulnerable Road User

Chapter 1

Introduction

The increasing population of vulnerable road users (VRUs), such as pedestrians and bicycles, has led to a growing concern for their safety on the road. These individuals lack protective structures and are highly vulnerable to severe accidents, often resulting in fatalities. However, advancements in technology present an opportunity to address this issue. Services like Cooperative Perception enable the sharing of environmental information among entities in close proximity, enhancing their awareness of the surroundings. Additionally, the integration of sensors in vehicles provides a foundation for the necessary infrastructure. By leveraging these technologies, it is possible to prevent accidents and improve the safety of vulnerable road users.

A possible use case for Cooperative Perception is the protection of VRUs. For instance, a roadside unit can actively monitor a specific area, like a crosswalk, by gathering information from both road users and roadside sensors. This data is transmitted through local or cellular wireless links to the management entity. Utilizing this information, the management entity creates a comprehensive situational representation of the target area and identifies potentially dangerous situations. Subsequently, these findings are communicated to the relevant road users, enabling them to take necessary actions to ensure their safety. Fig. 1.1 provides an illustration of such a scenario.

1.1 Motivation

To maximize the impact of Cooperative Perception in various scenarios, it is essential to ensure efficient and fast transmission and processing of information, particularly when dealing with video streams. However, as a relatively new field, there is ample room for improvement. For instance, the integration of uplink distribution of video traffic between multiple access networks with the aim of improving application-level latency through edge computation is still an emerging area with limited implementations.

In addition, comprehensive evaluations of full stack integration, encompassing the entire system from application to wireless technologies, are crucial. These evaluations help demonstrate the

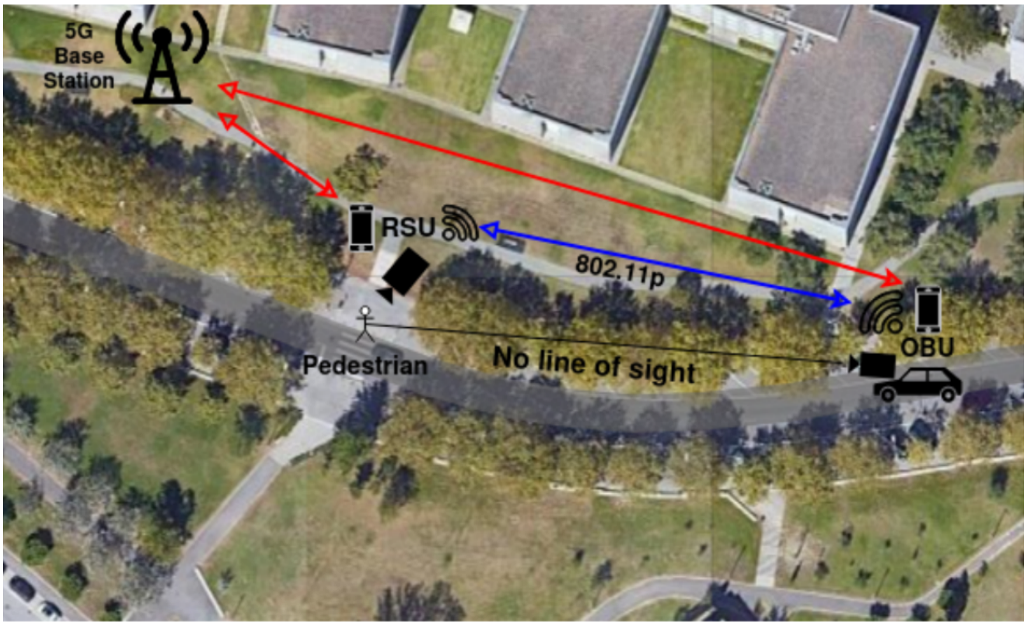

Figure 1.1: Cooperative Perception use case example. [1, Fig. 4.2]

feasibility and supportability of such use cases, shed light on the necessary infrastructure and technologies, and identify potential limitations. Conducting these evaluations in both small-scale and large-scale scenarios is important to understand the system's performance and scalability.

By addressing these challenges and pushing for advancements in information transmission, processing, and system integration, Cooperative Perception can make a significant difference in improving safety and efficiency in various real-world applications.

1.2 Problem Statement

The Cooperative Perception system described in [1] was originally developed using 802.11p technology for inter-vehicular communication. To improve the system's performance and capabilities, 5G technology was introduced. The purpose of this integration was to empirically evaluate the performance of 802.11p and 5G within the Cooperative Perception context. A series of experiments were conducted in scenarios where both technologies could be evaluated. The system operates in the following manner:

1. Video data is streamed from the camera
2. The video stream is transmitted to the local processor for object detection
3. CPMs[1] are generated based on the video processing and sent to relevant road users

[1] Cooperative Perception Message, defined by the European Telecommunications Standards Institute (ETSI). The goal of these messages is to give vehicles awareness of each other

Results show high latency in the object detection operation. These results are presented in Fig. 1.2. The object detection time significantly contributes to the overall end-to-end delay, with the remaining time components (except for packet latency) being negligible in comparison, with durations of less than 1 ms.

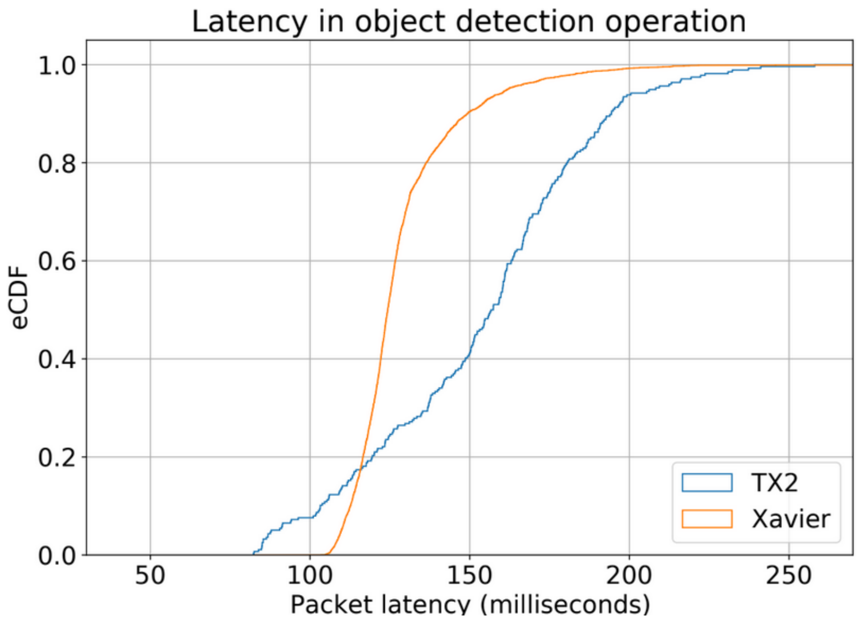

Figure 1.2: Object detection delay. [1, Fig. 5.3]

In order to shorten the object detection time, a new scenario incorporating edge computing is proposed, where the video stream is transmitted to an edge node for faster processing. This, however, introduces some different types of delay into the system, namely video transmission time. This new system operates in the following manner:

1. Video is streamed from the camera
2. The video stream is sent to an edge node, for object detection
3. CPMs are generated based on the video processing and sent to relevant road users

With object detection now occurring at the edge node, the video transmission becomes the primary contributor to time delays in the system, significantly impacting the overall delay.

Keeping this in mind, the goal of this dissertation is to design a method for uplink distribution of video traffic between multiple access networks, with the purpose of enhancing the application level latency through the use of computation at the edge, by focusing on designing an efficient packet scheduling algorithm to determine the appropriate channel for data transmission at any given moment.

1.3 Document Structure

Chapter 1 provides the necessary context and motivation for the work. It presents the problem and outlines the goals of the dissertation.

Chapter 2 discusses the theoretical aspects and fundamental knowledge relevant to the research area. It includes an overview of the current state-of-the-art, references to relevant reports and literature, and explores solutions found in the literature that may impact the work. Furthermore, this chapter highlights the contributions made by this research.

Chapter 3 contains the work approach, and the schedulers used in this work.

Chapter 4 contains the implementation of the testing environment. This includes its components, the necessary configurations, and a description of the network setup. Additionally, it contains the implementation of the performance metrics used, as well as how data is measured and processed.

Chapter 5 provides a detailed description of the experiment process, and outlines the executed experiments.

Chapter 6 presents the results obtained from the conducted experiments. It includes a thorough analysis and interpretation of the data, discussing the performance of the schedulers and their impact on the system.

Chapter 7 contains the conclusions made from the work, the contributions provided, and suggestions for future work.

Chapter 2

Literature Review

2.1 Background and Fundamental Aspects

2.1.1 Edge Computing

Running computationally demanding applications on user equipment is constrained by limited processing power, resulting in severe delays as shown in Sec. 1.2. A solution is to offload the applications to a conventional centralized cloud. However, this option introduces significant execution delays of its own, consisting of delivery of the offloaded applications to the cloud and back plus time of the computation at the cloud. Such a delay is inconvenient and makes the offloading unsuitable for real-time applications. Edge computing brings computation to the edge of the network, enabling it to run the highly demanding applications on user equipment while meeting strict delay requirements [6].

The inclusion of edge computing is essential to our use case, presented in Sec. 1.2, as 1) it reduces video processing delays, therefore constituting an addition with a lot of potential to increase time gain, and 2) given the required connectivity and high degree of cooperation, both between vehicles, and between vehicles and transportation infrastructure, which supports high throughput and low latency communications and processing [7, p. 4]. Furthermore, 5GAA (5G Automotive Association) is backing the vision of 5G combined with edge computing capabilities to support V2X[1] applications [8].

2.1.2 RTP and RTCP

RTP is a protocol used for transmitting real-time multimedia data. It is designed to support the efficient and timely delivery of audio, video, and other time-sensitive data streams. RTP provides mechanisms for packetization, transport, and synchronization of media streams, making it suitable for applications that require low-latency and continuous delivery, such as video streams.

[1] Vehicle-to-everything communication, such as Vehicle-to-Vehicle, Vehicle-to-Infrastructure or Vehicle-to-Pedestrian

RTP packets are typically transported over UDP. RTP also supports payload-specific features and extensions, allowing for the inclusion of additional information, such as codec types, timestamp accuracy, and transmission control information.

RTCP works alongside RTP as a companion protocol to provide control and feedback mechanisms for real-time multimedia communication sessions. RTCP enables participants in a session to exchange control packets containing valuable information about the quality, statistics, and control parameters of the media streams being transmitted. These RTCP packets are periodically sent by participants to report on various metrics, including packet loss, round-trip time, jitter, and participant identification. These reports assist in monitoring network conditions, adjusting transmission strategies, and detecting potential issues affecting the quality of the media streams.

By working in tandem, RTP and RTCP provide a robust framework for real-time multimedia communication. RTP handles the actual transport and delivery of time-sensitive media streams, while RTCP focuses on control, monitoring, and signaling aspects to maintain the quality and synchronization of the communication sessions.

2.1.3 Multipath Transport

Multipath (MP) provides the ability to simultaneously use multiple paths between peers [2]. For protocols like TCP, DCCP, and RTP, it is implemented as an extension. MP extensions work similarly, which is as follows (as described in [2], and illustrated in Fig. 2.1):

- A connection between 2 hosts is established
- In case extra paths are available, additional subflows are created
- Existing additional subflows are attached to the existing MP session, which continues to appear as a single connection to the applications at both ends

An important part of MP is the scheduler, which is where the focus of this work resides. With multiple subflows established, the scheduler's job is to select the best one available to send each packet. The concept of best subflow depends on the scheduler's purpose, and thus, the scheduling decision plays a key role on the improvement of MP performance [5, p. 1445]. Schedulers can be classified into 2 general categories: single criterion, and multicriteria [5, p. 1448]. As the names suggest, a scheduler fits into one of these categories according to the number of metrics it uses to make scheduling decisions, being single criterion when decisions are based on only 1 metric, and multicriteria when decisions are based on more than 1 metric. Different types of schedulers are explored in Sec. 2.2.1.

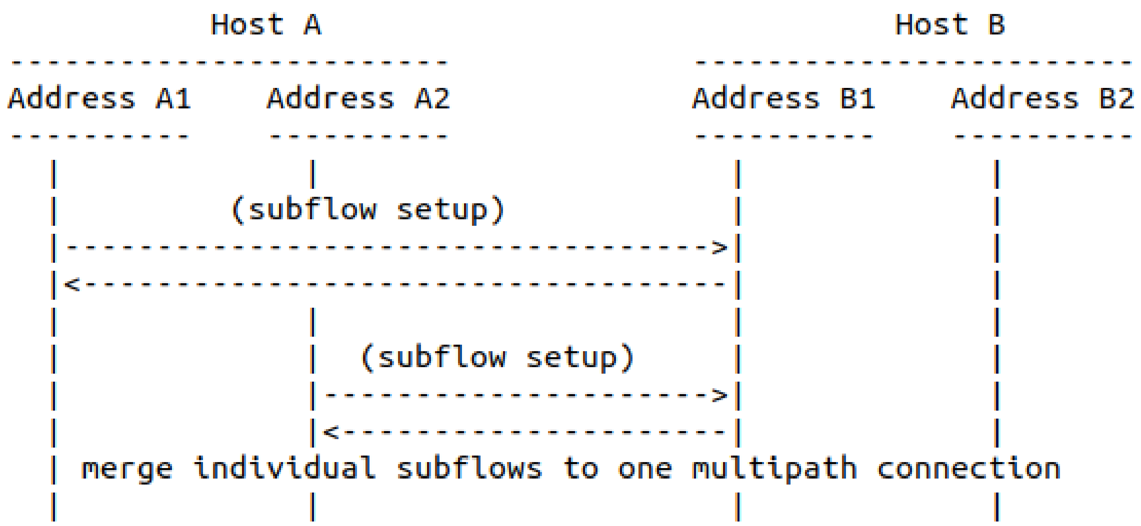

Figure 2.1: Multipath usage scenario. Adapted from [2, Fig. 2]

2.1.3.1 MP-TCP

MP-TCP is the MP extension of TCP [9]. It provides the following major benefits: better network resource utilization, higher throughput from multipath bandwidth aggregation, and greater communication failure tolerance when available subflows can replace faulty ones [5, p. 1445]. A comparison of standard TCP and MP-TCP protocol stacks is illustrated in Fig. 2.2, where we can see an additional layer (MP-TCP), which provides abstraction to the application, showing only a single socket, and that separates and merges traffic. However, like TCP, MP-TCP is not adequate for real-time traffic that can tolerate some amount of losses.

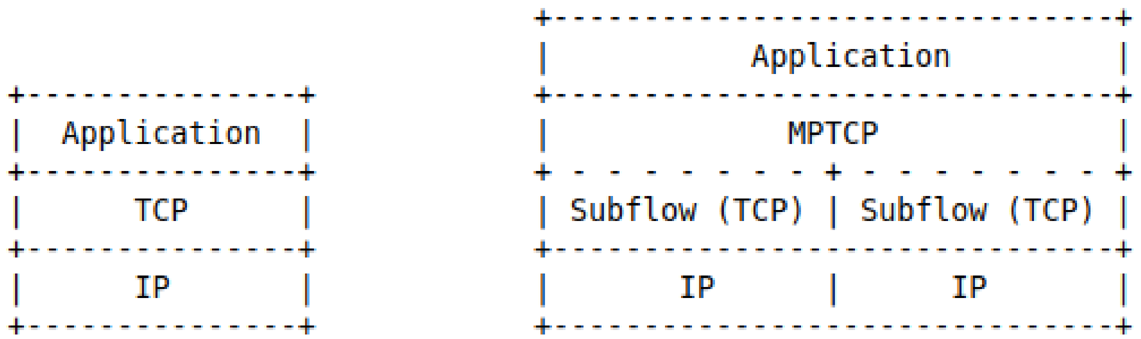

Figure 2.2: Comparison of standard TCP and MP-TCP protocol stacks [3, Fig. 1]

2.1.3.2 MP-DCCP

Contrary to TCP, DCCP [10] provides connections of congestion-controlled unreliable datagrams, making it suitable for applications that transfer fairly large amounts of data and that can benefit from control over the tradeoff between timeliness and reliability [10]. MP-DCCP is the MP extension of DCCP. Fig. 2.3 illustrates a comparison of standard DCCP and MP-DCCP protocol stacks. Like MP-TCP, MP-DCCP provides abstraction to the application, as well as merging and separation of traffic.

```
                          +-------------------------------+
                          |          Application          |
+---------------+         +-------------------------------+
|  Application  |         |            MP-DCCP            |
+---------------+         + - - - - - - - + - - - - - - - +
|     DCCP      |         |Subflow (DCCP) |Subflow (DCCP) |
+---------------+         +-------------------------------+
|      IP       |         |      IP       |      IP       |
+---------------+         +-------------------------------+
```

Figure 2.3: Comparison of standard DCCP and MP-DCCP protocol stacks. [2, Fig. 1]

Given the available control over the timeliness and reliability tradeoff, making it possible to support several congestion control algorithms, MP-DCCP becomes advantageous when compared to MP-TCP, for the purposes of this work.

2.1.3.3 MP-RTP

RTP [11] provides loss tolerant end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data [11, pag. 1]. MP-RTP is the MP extension of RTP. Fig. 2.4 illustrates a comparison of standard RTP and MP-RTP protocol stacks.

```
+----------------------+     +-----------------------------+
|     Application      |     |         Application         |
+----------------------+     +-----------------------------+
|                      |     |            MPRTP            |
+         RTP          +     +- - - - - - - -+- - - - - - -+
|                      |     | RTP subflow   | RTP subflow |
+----------------------+     +---------------+-------------+
|         UDP          |     |      UDP      |     UDP     |
+----------------------+     +---------------+-------------+
|          IP          |     |      IP       |      IP     |
+----------------------+     +---------------+-------------+
```

Figure 2.4: Comparison of standard RTP and MP-RTP protocol stacks. [4, Fig. 1]

Yet, this proposal had several limitations, mostly due to the bandwidth estimation formula for making traffic splitting decisions. It depended on the existence of traffic in all links, and it was very sensitive to short-term fluctuations [12].

2.2 Related Work

In this section, several reports that contain proposed solutions, surveys, work methodologies, and performance metrics that are related to this work will be analyzed.

2.2.1 Multipath Scheduling

2.2.1.1 Scheduling Problems

As noted in [5], there are two main scheduling problems that may lead to high delays and throughput degradation, which are head-of-line (HoL) blocking and bufferbloating. HoL blocking occurs when packets that are sent through faster subflows reach the destination before their predecessors. When this happens, the received data cannot be processed until the remaining out-of-order packets arrive. On the other hand, bufferbloating occurs due to large buffers in bottleneck routers, which results in long queue times when the paths are under congestion.

LL-based policies, that choose the lowest-latency subflows, help to mitigate HoL blocking and bufferbloating. However, other issues like blinder decision and biased-feeding phenomenon might come up, as stated in [5]. Relying on path latency as the single criterion, policies like LL fail to consider other criteria, and consequently make uninformed/blind decisions. On the other hand, biased-feeding stems from the sticky nature of LL, that is it keeps selecting the same path as long as it is available. Such behaviour does not take advantage of the multipath capabilities.

Although most of the literature only mentions HoL blocking and bufferbloating, as they are generally the most prominent issues, there is a clear need for a balance between these problems and others like blinder decision and biased-feeding. While some scheduling solutions might prevent some of the issues, they might also bring out others more prominently. Therefore, scheduling solutions should be thought out with all these problems in mind, to impose the needed balance.

2.2.1.2 Scheduling Solutions

Literature on scheduling solutions shows a clear advantage of multicriteria solutions over single criterion solutions.

In [13], several schedulers are proposed in an attempt to improve video streaming through MP-TCP. Here, it is proposed that video streaming playout may be affected by path switches between paths with different characteristics. After building a model for the impact of path switch on video streaming playout events that drive quality of service, schedulers to minimize the number of path switches were proposed. They were evaluated against the minimum RTT scheduler and the round robin scheduler. Results show that path switch reduction may improve video streaming quality, or at least maintain it in the worse case.

In [14], five state-of-the-art multipath schedulers are tested. These include four traditional non-learning-based schedulers - round-robin (RR), minimum round-trip time (min-RTT), blocking estimation (BLEST) [15], and earliest completion first (ECF) [16] - and a learning-based scheduler - Peekaboo [17]. The following are descriptions of these schedulers, as presented in [14]:

- RR cyclically sends packets over each path, which performs well for homogeneous paths, but otherwise performs poorly, as it does not consider the characteristics of individual paths. This means it is not able to prevent out-of-order packet arrival, which leads to head-of-line (HoL) blocking. For use cases such as the one being explored in this work, HoL blocking constitutes a severe performance issue, as data can only be processed when head of the data stream has been received.
- minRTT is the default scheduler in MP-TCP. Here, packets are sent through the path with lower RTT. This reduces out-of-order packet arrivals, which lowers the chance of HoL blocking, consequently achieving higher throughput.
- BLEST and ECF try to provide both high throughput and low latency. They behave like minRTT when more than one path is available. Otherwise, they make decisions on whether to send packets on the path with larger RTT or wait for the path with lower RTT to become available again. Finally, both BLEST and ECF further alleviate the out-of-order packet arrival compared to minRTT.
- Peekaboo also tries to leverage a minRTT-based logic, since it aims at learning how to optimally exploit the path with lower RTT. However, when the lower RTT path is not available, Peekaboo learns at runtime the decision to make. It achieves higher aggregation compared to other multipath schedulers in scenarios with high path heterogeneity and its performance converges to the performance provided by min-RTT, BLEST and ECF when the available paths get more homogeneous in terms of RTT.

In [5], several solutions are presented. Fig. 2.5 shows an arrangement of several schedulers, as well as their applications and goals. Additionally, Table 2.1 summarizes the schedulers presented in Fig. 2.5, highlighting the algorithms and their approach to make scheduling decisions.

2.2.2 Performance Metrics

Literature on performance metrics shows that throughput is the most, and almost always the only, used metric. Some literature also acknowledges FMSE (Foveated Mean Squared Error) as the best video quality metric.

In [5], improving throughput is one of the main goals of MP-TCP, and thus it is used as a base for the performance metrics. The used metric aggregates throughput from all of the paths into one single value, to assess throughput improvement.

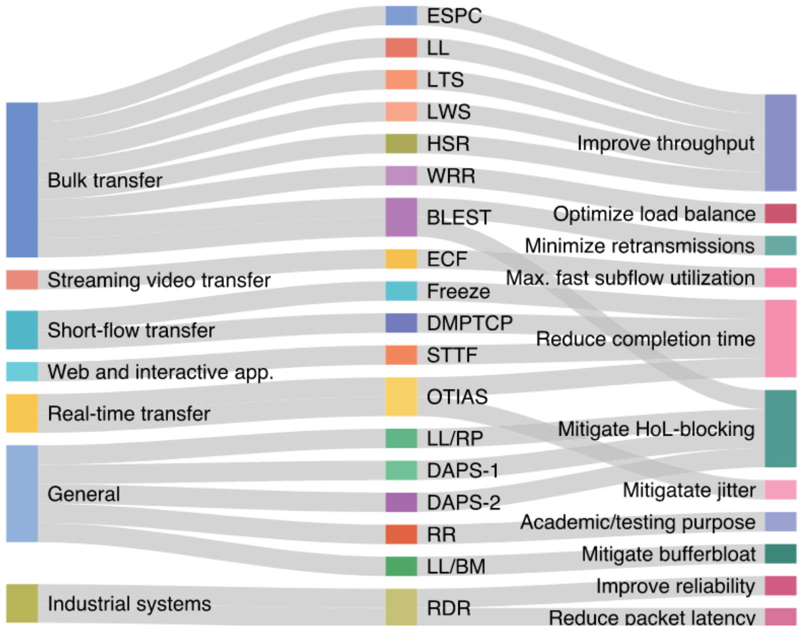

Figure 2.5: Applications, schedulers, and their overall goals. [5, Fig. 3]

In [13], picture discards (the number of frames discarded by the video receiver), buffer underflow (the number of times playout buffer has emptied out), and subflow throughput were used as performance measurements.

For video quality metrics, results from [18] show that metrics that generally achieve a high correlation with subjective results are MOtion-based Video Integrity Evaluation (MOVIE) and FMSE, but MOVIE has significantly higher complexity than FMSE.

2.2.3 Methodology

The methodologies used in related literature are very similar. Indeed, a good example of their structure is described in [5], where an experimental methodology is used. This methodology is composed of 3 main parts:

- Deployment of Multipath Test Environments
- Design of Multipath Experiments
- Definition of Performance Metrics

In [14], the aim was to provide a realistic evaluation of multipath schedulers. For that, link characteristics were derived from real measurements, with both network traces and statistical values. These values are then applied to the emulation environment, based on Mininet.

In [5], three types of test environments are discussed: simulation, emulation, and real environments. For simulations, it is pointed out that while available tools allow for the creation of realistic

Table 2.1: Summary of various packet scheduling solutions proposed from the public releases of MP-TCP Linux kernel (2013). Adapted from [5, Tab. I]

Scheduler	Algorithm	Approach
LL	Lowest-latency first.	Default MPTCP scheduler that chooses the available subflow of lowest sRTT.
RR	Round-robin.	Alternates the use of available subflows in round-robin fashion.
DAPS-1	Forward delay aware.	Schedules a chunk of packets based on their sequence numbers and the path delays for in-order receiving.
DAPS-2	Maximum receive buffer blocking time.	Same approach of DAPS-1.
ESPC	Estimated subflow path capacity.	Estimates subflow capacity and keeps it longer at the saturation point, while considering the path congestion.
LL/RP	LL with retransmission and penalization.	LL extension to retransmit HoL packets and penalize subflows that cause HoL-blocking.
LL/BM	LL with bufferbloat mitigation.	LL extension to identify subflows in bufferbloat condition bounding their congestion windows.
OTIAS	Out-of-order transmission for in-order arrival.	Estimates the packet arrival time over all available subflows, and chooses the ones of earliest time.
RDR	Redundant Data Reinjection.	LL extension to send the packet by replicating it throughout the other existing subflows.
FPS	Temporary freezing of slow subflows.	LL extension to freeze temporarily slower subflows and prioritize the fastest ones.
BLEST	Blocking estimation.	Estimates whether a subflow will cause HoL-blocking and dynamically adapts scheduling to prevent blocking.
ECF	Earliest completion first.	Identifies whether scheduling a slow subflow will cause faster subflows to become idle.
LTS	Lowest time/space first.	Chooses subflows of lowest ratio of latency and window space whenever possible, otherwise, reduces to LL.
LWS	Largest window space first.	Chooses subflows of largest congestion window space whenever possible, otherwise, reduces to LL.
HSR	Highest send rate first.	Chooses subflows of highest send rate.
WRR	Weighted round-robin.	RR extension with subflow weight to provide optimal load-balance.
DMPTCP	Application workload aware.	Finds the set of the subflows that can complete the applications transmission within RTT.
STTF	Shortest transmission time first.	Estimates transfer time of a data segment over the subflows and selects the one of the shortest time.

scenarios, which are then used to validate the target communication algorithm, a rework is always going to be needed to reimplement the algorithm to be suitable for the real Internet. With emulators, this problem is eliminated, as container-based emulation tools allow for the validation of these algorithms without the need of reimplementation. However, these scenarios are more limited than the ones provided from simulations. Lastly, real environments provide both the benefits from both simulation and emulation, with its flexibility and absence of need for reimplementation. The downside here is the coexistent cross traffic, which makes it difficult to identify, understand and explain the events that impact the experimental results. This could be entirely avoided with an isolated network infrastructure, if the cost of such a test environment were to be tolerable.

Furthermore, in [19], the maintainers of MP-TCP proposed a single-hop network between two multihomed nodes to test its performance limits of MP-TCP. As noted in [5], while this setup may be appropriate for data center networks, it is not realistic for other scenarios. To fix this, an addition of bottleneck links is proposed, to provide a more general purpose test environment. This way, each link can be programmed to have to desired characteristics, to reproduce any network scenario.

In [5], experimental design approaches are used to evaluate how MP-TCP implementations behave in different environments, as well as a tool for revealing performance issues. Here, sets of factors - such as throughput capacity, packet loss rate, and queue delay - and their domains of values are combined to generate a space of experiments. Issues arise when the number of experiments grows to an unfeasible quantity to conduct fully and properly. For that, a way to provide full space coverage with a minimal number of experiments is proposed in [20], using heuristics to randomly generate the experiments to cover the space of experiments.

2.3 Contributions

Taking into account the focus of existing related work on MP-TCP rather than other protocols like MP-DCCP and MP-RTP, as well as the scarcity of research on multipath transport for video transmission, this work presents unique findings in a distinct use case. Additionally, this research contributes valuable insights into the usage of 5G in multipath transport, which is an area that has received limited attention. Furthermore, several new schedulers were designed for multipath video transport. Lastly, the results of full-stack integration show that these use cases can be supported, how they can be supported, and where the limitations of current technologies reside.

Chapter 3

Multipath Video Transport

This work is part of a bigger project named Floyd. This project was created in the context of a mobility industry currently facing a significant increase in the use of autonomous functionalities with the objective to achieve the concept of full self-driving cars. Artificial Intelligence, Machine Learning or Advanced Networking are some examples of the concepts that have progressed rapidly and that help to develop autonomous vehicles. There are several benefits that can result from this technology such as the reduction of car accidents, reduced cost and environmental impact of transport, accessibility for a wider range of users or even the liberation of time used for driving.

Floyd project aims to devise a full-stack computation-networking solution to support autonomous vehicles, more specifically platoons, with the efficiency and reliability required to work properly. Vehicular platooning has the potential to become an important part of daily life in several areas such as industrial/logistic premises (e.g., factory, port, airport) or public roads. It involves several demanding applications (e.g., autonomous driving, precise tracking, safety) that require strong connectivity requirements and, for that, needs to leverage networking and cloud/edge computing technologies, as vehicles generate large amounts of data from sensors that need to be processed at edge and cloud nodes. This project has the purpose to be a part of the complex solution for the autonomous driving concept, which involves several other tools or technologies such as LIDAR or GPS to become efficient and reliable.

3.1 MA-RTP

Real-time data streams, such as video, require efficient transport protocols to ensure timely delivery and minimize data loss. Existing multipath transport protocols like MP-TCP, MP-DCCP, and MP-RTP, discussed in Sec. 2.1.3, offer benefits such as improved resource utilization, higher throughput, and fault tolerance through multipath bandwidth aggregation. However, these protocols have limitations when it comes to handling real-time traffic that can tolerate a certain level of loss.

The motivation behind developing MA-RTP stems from the need for a multipath transport protocol specifically designed to cater to the requirements of real-time, loss-tolerant data streams. Traditional protocols like MP-TCP and MP-DCCP prioritize reliability and congestion control over timely delivery, which may not be suitable for applications that are not bytestream oriented, where a certain level of loss can be tolerated. MP-RTP, while designed for real-time data, has limitations in its bandwidth estimation formula and sensitivity to short-term fluctuations. Therefore, there is a need for a new multipath transport protocol that addresses these limitations and offers improved performance for loss-tolerant real-time data streams.

The MA-RTP protocol, one of the main contributions to Floyd from Instituto de Telecomunicações, and developed before this work in [21], aims to fill this gap by providing a dedicated multipath transport protocol tailored for real-time, loss-tolerant data streams. The main application envisioned for this protocol is the transfer of data heavy sensor streams on the uplink, targeting processing at edge nodes to reduce overall processing latency. It leverages the advantages of existing multipath protocols while introducing enhancements in areas such as bandwidth estimation, traffic splitting decisions, and connection management. By focusing on the specific requirements of real-time data streams, MA-RTP aims to offer reduced latency, and efficient utilization of available network resources. Its design and architecture take into account the unique characteristics of real-time data transmission, enabling applications to benefit from multipath transport while maintaining the necessary tolerance for loss.

3.1.1 Protocol Description

MA-RTP follows a similar structure to other multipath protocols, having one RTP session over each aggregated network interface. It also includes an additional layer that provides a single interface to the application. This protocol stack is depicted in Fig. 3.1.

```
+---------------+
|  Application  |
+---------------+
|     MA-RTP    |
+-------+-------+
|  RTP  |  RTP  |
+---------------+
|  UDP  |  UDP  |
+---------------+
|   IP  |  IP   |
+-------+-------+
```

Figure 3.1: MA-RTP protocol stack

The MA-RTP layer assumes several responsibilities, including managing multiple RTP sessions, making scheduling decisions, and monitoring available paths for data transmission. It leverages RTCP reports associated with the RTP sessions to collect application-level per session performance data that can be useful for enhancing traffic splitting and connection management. These connections are illustrated in Fig. 3.2. Additionally, MA-RTP can request data from an external database to further improve traffic splitting.

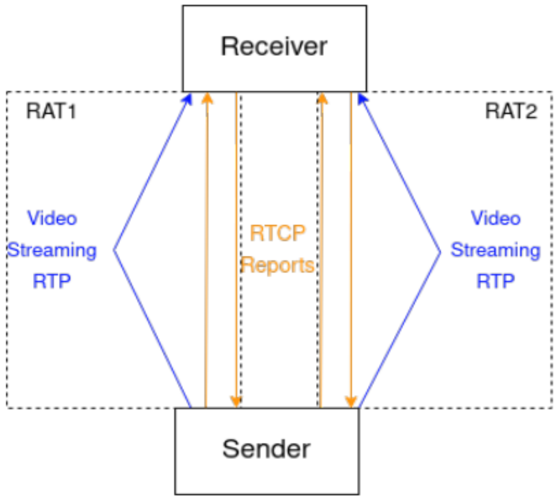

Figure 3.2: MA-RTP connections

Furthermore, MA-RTP incorporates a dedicated module designed for handling video streams. This module takes care of essential operations such as fragmenting, assembling, and reordering video frames, ensuring smooth and uninterrupted video data transmission. It also includes a module for updating GPS coordinates to determine the current geographical location.

3.1.2 Implementation

The implementation of the MA-RTP protocol involves several key components that work together to facilitate its operation and functionality. These key components and their interactions are illustrated in Fig. 3.3.

Overseer: This is the most important component. It manages the remaining classes and the data flow. This class receives frames to be sent, calls the Scheduler asking for a decision on which interface to use, and then passes the frame and the decision to the RTPSender. It is also the Overseer that acquires GPS position and retrieves from an external database relevant data to be used for decision making.

Scheduler: It is the Scheduler that decides through which interface the next video frame should be sent.

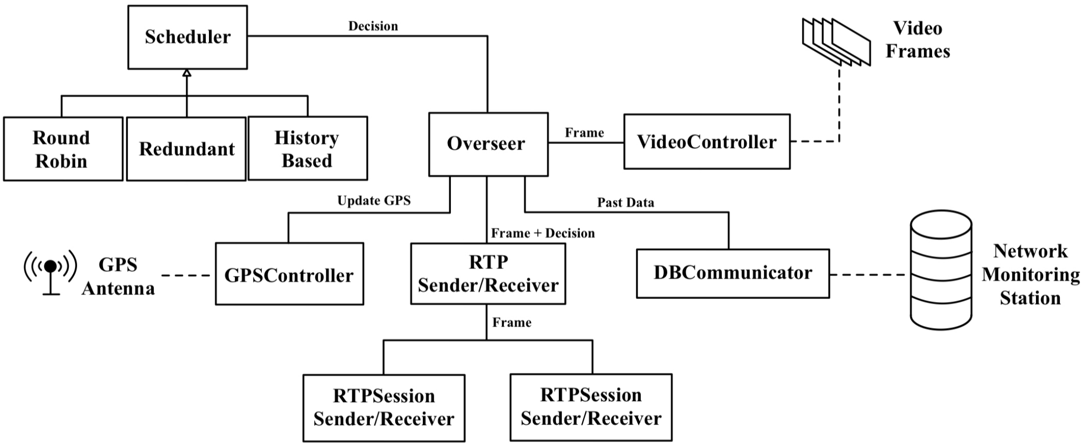

Figure 3.3: MA-RTP architecture

RTPSender: The RTPSender fragments (whenever necessary) and sends a video frame, through the indicated interface(s).

RTPReceiver: The RTPReceiver receives RTP packets through either RTP session. RTP packets contain video frames or fragments of video frames, and they are re-assembled into frames upon reception.

Lastly, RTP and RTCP are implemented through imported JRTPLIB libraries. The protocol can run as a Sender or a Receiver.

3.1.3 Limitations

During the course of the project, it was discovered that the RTCP sender and receiver reports were not being sent as initially planned. As a result, alternative options had to be explored. The chosen alternative involved collecting the necessary data and populating a database (the network performance monitoring station, addressed in Sec. 3.2) with it. The data is then requested as needed. However, this solution is suboptimal since the information provided is not in real-time and cannot accurately reflect the current state of the network.

Furthermore, it is important to acknowledge that certain features were not fully implemented during [21] due to time constraints. One such limitation pertains to the video reconstruction process, which currently relies on finishing the video transmission before reconstruction can occur. Consequently, this approach does not account for late arrivals of frames. Additionally, in the event of losing an RTP packet, there is no mechanism in place to replace it or the frame it belongs to; instead, the frame is skipped. To mitigate the limitation of video reconstruction, a statistic was devised to assess late frame arrivals and determine whether a frame would be considered in the analysis. However, this is still not an ideal solution as it only provides an estimation of the resulting video quality rather than a comprehensive analysis of the received video.

3.2 Network Performance Monitoring Station

To enhance the performance and reliability of the multimedia communication system, MA-RTP incorporates essential data from a network performance monitoring station. This station stores essential data related to various geographical locations within the designated testing area. The collected data plays a crucial role in making informed scheduling decisions regarding the assignment of each frame to its respective path. By leveraging this historical information, the system may be able to improve the allocation of frames and ensure efficient transmission.

3.2.1 Implementation

The network performance monitoring station consists of a PostgresSQL database containing two tables: 'locations' and 'history'. The 'locations' table establishes associations between location IDs and corresponding latitude/longitude pairs. This information serves to identify specific geographical positions within the testing area. The 'history' table contains data associated with each location ID. Other than the respective location ID, this data includes timestamps, bandwidth, number of transmitted bits, packet loss rate, jitter, signal strength, RAT choice, user speed, user orientation, and user movement status.

3.2.2 Populating the Station

Data collection is carried out using TEMS Pocket [22], a portable mobile network testing tool, which allows capturing data from the network. The captured data is then extracted into a file using TEMS Investigation [23]. To facilitate the data collection process, a script consisting of two actions is created in TEMS Pocket: one action to initiate the recording of the logfile, and another action to perform an HTTP GET request for a Raspbian image. The choice of a Raspbian image is made to ensure continuous data collection without frequent interruptions, considering its larger file size. Additionally, the positioning feature is enabled to capture the geographical information associated with the collected data. To appropriately populate the database, a Python program developed in [21] is used to extract relevant information from the TEMS Investigation file and insert it into the the network performance monitoring station.

3.2.3 Limitations

The network performance monitoring station had limited data available, which posed challenges when making scheduling decisions. The effectiveness of these decisions relies heavily on the availability of comprehensive and up-to-date network information. Unfortunately, the network performance monitoring station in this study did not contain as much data as desired, possibly hindering the accuracy and efficiency of the scheduling process.

3.3 Work Approach

To accomplish the goal of this dissertation, an experimental methodology was adopted. The aim was to conduct tests on multiple schedulers, assess their performance, and develop new schedulers based on the findings.

To conduct these tests, a suitable test environment is needed. This environment operated on the following concept: a mobile node streams a video to an edge node, with multiple paths available for the transmission, while physically moving through different positions. As such, this mobile node needs to be easily carried and cannot be physically attached to any fixed infrastructure. This allowed for the evaluation of schedulers in dynamic conditions, considering the varying network characteristics and path availability as the mobile node moved. Thus, it is possible to assess the performance of the schedulers in a real-world setting. Nevertheless, the environment needs to be controlled, so that conditions do not change much and variability is limited to enable understanding what happens.

Additionally, the selection of appropriate performance metrics is essential to highlight relevant aspects of each scheduler. Each metric has been chosen based on its relevance to the overall performance assessment and its ability to provide meaningful insights into system behavior. The selected performance metrics are throughput, latency, frame loss, and video quality. Throughput is an important metric to assess the efficiency of the system, as higher throughput indicates better data transfer rates. It is a way to verify the individual and combined usage of both paths. Low latency is crucial for real-time applications such as this, as it ensures minimal delay in data transmission. This is important, as it determines how soon the data can be used. Frame loss is essential metric for assessing the quality of video transmission, as it may greatly affect the received video. High frame loss can result in visible artifacts or interruptions in the video stream. Video quality assessment provides an objective measure of the visual fidelity of the transmitted video. It helps to ensure the received video can be appropriately processed in the later stages of this use case's pipeline.

3.4 Schedulers

A variety of schedulers were employed in the experiments conducted for this work. Some of these schedulers were implemented in [21]. These include:

WiFi-only: Send all frames through WiFi.

5G-only: Send all frames through 5G.

Redundant: Send all frames through both paths.

Round-robin: Send each frame through only one path in an alternating fashion. This scheduler is illustrated in Fig. 3.4.

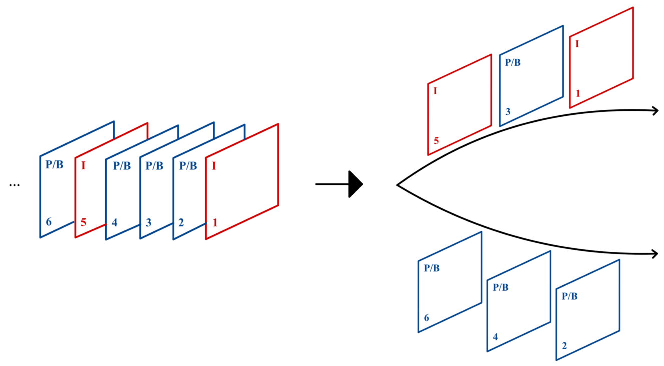

Figure 3.4: Round-robin frame distribution

Context-based average throughput: Send frames via the path with higher average throughput, according to historic data associated with the current geographical position. This scheduler was developed in [21].

In addition to the above schedulers, several new schedulers were designed and implemented specifically for this work. These schedulers include:

I Frame to WiFi: Send all I frames via WiFi and the rest via 5G. This scheduler was developed in an effort to provide stability for the B and P frames, as the 5G path appeared to be the more consistent in terms of QoS. It was expected that the received video would be smooth when the I frames were successfully delivered.

I Frame to 5G: Send all I frames via 5G and the rest via WiFi. This scheduler was developed in an effort to provide reliability for delivering I frames, as the 5G path appeared to be the more consistent in terms of QoS. It was expected that even if the video displayed some artifacts, they would be quickly fixed.

Random: Send frames via a randomly chosen interface. The idea behind this scheduler was to identify existing unpredictable aspects, useful when developing other schedulers.

Round-robin based on GoP Size: Similar to round-robin, but frames are sent in alternating interfaces at the granularity of the GoP. Each GoP is sent through one interface before changing to the other, ensuring a more balanced distribution compared to the original round-robin scheduler where most I frames would go through one interface, due to the GoP size of 4 frames. This scheduler is illustrated in Fig. 3.5.

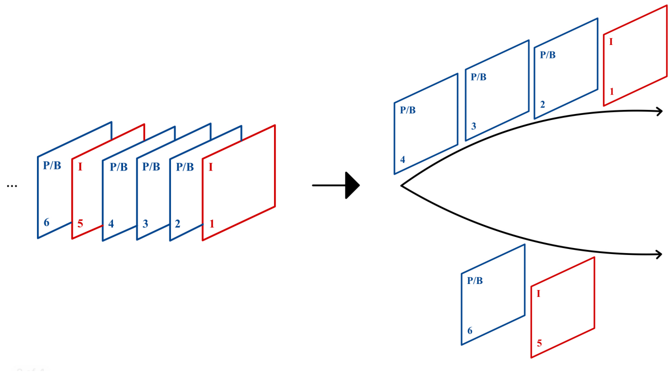

Figure 3.5: Round-robin based on GoP Size frame distribution

Round-robin based on Frame Size: Similar to round-robin, but I frames and other frames are sent separately. I frames are sent in an alternating manner, and the rest of the frames (P and B frames) also follow a separate alternating pattern. This scheduler aims to achieve an even more balanced distribution. This scheduler is illustrated in Fig. 3.6.

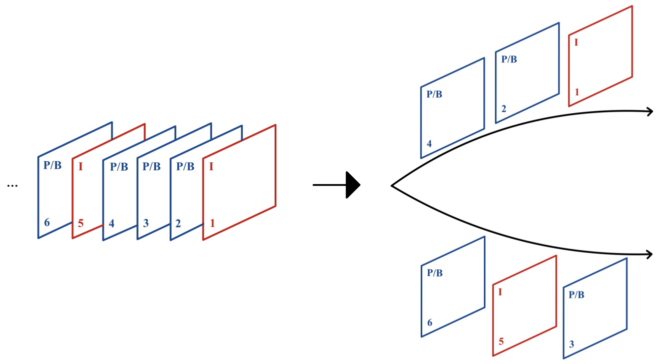

Figure 3.6: Round-robin based on Frame Size frame distribution

Zone based: Frames are sent according to user defined standard. Each zone is assigned a specific interface choice by the user. This scheduler can act as a "debugger" to test theories and rule out or confirm certain assumptions.

Packet Loss Rate: Send frames via the path with lower packet loss rate, according to historic data associated with the current geographical position.

Timely Arrival: Send frames through the path with better RTT and packet loss rate values, in an attempt to get frames to the receiver and on time.

Chapter 4

System Implementation

4.1 Test Environment

4.1.1 System Overview

The schedulers will be evaluated experimentally in FEUP's campus by uplink video transmission over 5G and WiFi using the setup shown in Fig. 4.1.

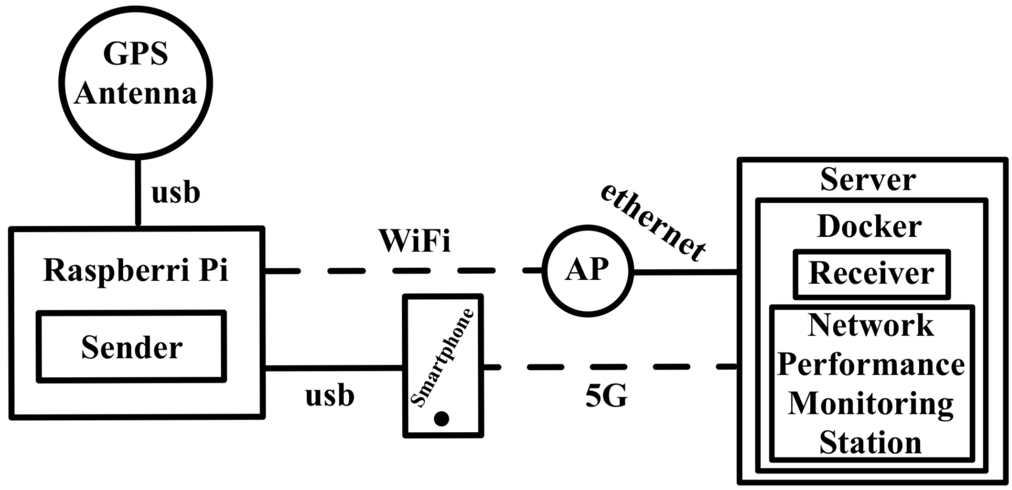

Figure 4.1: Setup scheme

On the sender side, a raspberry pi, chosen for its portability, serves as the primary device running the ma-rtp protocol as sender. It is connected to a GPS antenna, a 5G-capable smartphone, and a power bank, all via USB. The small size of these components allows for easy transportation within the testing area. A power bank is included to serve as a power supply, to meet the flexibility needs of the setup. Additionally, a WiFi AP is setup to provide WiFi connectivity to the raspberry pi. The WiFi and 5G paths are provided through the AP and smartphone, respectively. Consequently, frames selected for the WiFi subflow are transmitted via the AP, while frames designated for the 5G subflow are sent through the smartphone. For a visual representation, refer to Fig. 4.2.

The AP is always positioned on top of the same 2m structure, similarly to a road-side unit, which is placed on top of a marking on the floor, to be in the same position. Alongside the AP, additional components are mounted on the structure. These include a laptop, a power extension, and a Power over Ethernet (PoE) injector. The laptop serves as an interface for establishing SSH connections to both the raspberry pi and the server, for experiment control and configuration. Through the laptop, necessary operations can be performed to execute the experiments effectively. Both sender and receiver can be reached via the AP, so the laptop is connected to the AP to establish the SSH connections. The power extension provides electrical power to both the laptop and the AP. For the latter, this is achieved by using the PoE injector, which enables power transmission over Ethernet cables. For a visual depiction of the setup, refer to Fig. 4.2.

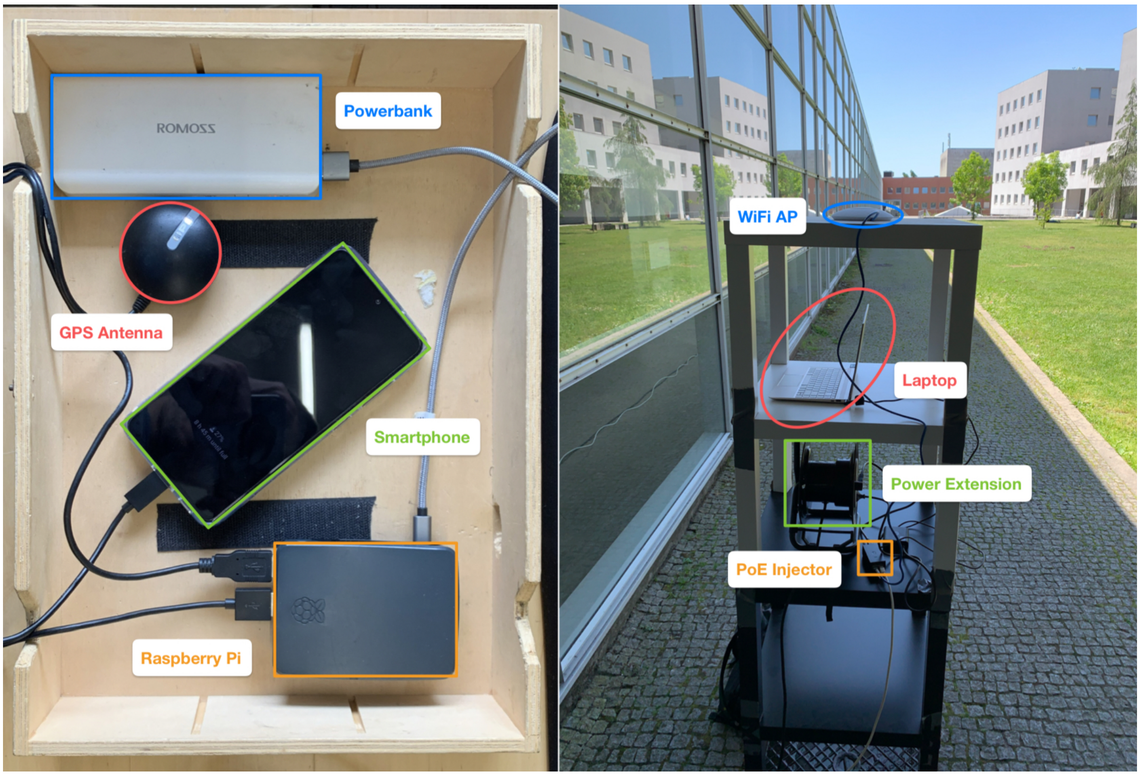

Figure 4.2: Photographs of the experimental setup

The receiver side of the experimental setup comprises a server running the ma-rtp protocol as receiver. The server is also running the network performance monitoring station, isolated from the receiver. This station stores essential data related to various geographical locations within the designated testing area. The collected data plays a crucial role in making informed scheduling decisions regarding the assignment of each frame to its respective path. By leveraging this historical information, the system can improve the allocation of frames and ensure efficient transmission.

4.1.2 System Components

Server: The server is running Ubuntu 18.04.6 LTS, with kernel version 4.15.0-112-generic. The ma-rtp protocol and the network performance monitoring station both run isolated in Docker containers. The Docker version installed on the server is 20.10.2.

Raspberry Pi: The Raspberry Pi used is the Raspberry Pi 4 Model B Rev 1.1. It is powered by a Broadcom BCM2711 quad-core Cortex-A72 (ARMv8) 64-bit system-on-chip (SoC). Each core of the SoC runs at a clock speed of up to 1.5 GHz, providing efficient processing power. The Raspberry Pi is equipped with 2GB of RAM, offering the necessary memory for its tasks. In terms of connectivity, the Raspberry Pi supports dual-band 2.4 GHz and 5 GHz IEEE 802.11ac wireless LAN. Additionally, it has two USB 2.0 ports and two USB 3.0 ports, allowing for convenient connectivity with peripherals such as the GPS antenna and the smartphone. For stable and reliable operation, the Raspberry Pi requires a USB-C power supply with a capacity of 5V and a recommended current rating of at least 3A. The Raspberry Pi runs Ubuntu Server 22.04.1 LTS, with kernel version 5.15.0-1027-raspi.

Access Point: The Ubiquiti UniFi AP AC Lite is a feature-rich access point designed for seamless wireless network connectivity. With a 2.4 GHz radio rate of 300 Mbps and a 5 GHz radio rate of 867 Mbps, it delivers fast and reliable wireless connections. Equipped with 2x2 MIMO technology for both frequency bands and a maximum TX power of 20 dBm, it ensures efficient data transmission and wide coverage. The access point features a 10/100/1000 Ethernet port for easy integration into wired networks, dual-band antennas with 3 dBi gain each, and support for multiple wireless standards, including 802.11ac.

Smartphone: The Samsung Galaxy S21 FE 5G runs on the Android 13 operating system and is powered by a high-performance CPU. The phone comes with 6GB RAM. The device features a large Li-Ion 4500mAh battery, supporting fast wired charging at 25W. It offers versatile network connectivity options, including GSM, CDMA, HSPA, EVDO, LTE, and 5G, allowing for cellular connectivity in various environments. The phone's kernel version is 5.4.147. The phone was connected to the Raspberry Pi via USB tethering, enabling the Raspberry Pi to access the server using the phone's mobile data 5G connection.

GPS Antenna: The Globalsat BU-353S4 USB GPS Receiver is a GPS antenna that uses the SiRF Star IV chipset, known for its high performance and sensitivity of -163dBm to track GPS signals even in areas with weak signal strength. It quickly establishes a GPS fix, thanks to its fast Time To First Fix (TTFF) capability. The antenna supports the NMEA 0183 protocol, which allows it to communicate with other navigation devices. It has internal memory that helps it re-track its position faster by using previous location information. With its USB to Serial bridge, it can convert signals between USB and Serial formats. The antenna can receive signals from up to 48

GPS channels simultaneously, providing accurate positioning with a 2-meter accuracy. It supports SBAS, which improves GPS accuracy using additional satellites or ground-based stations. The antenna uses the WGS-84 standard to represent positions on Earth's surface. In case of signal loss, it can re-establish a GPS fix in just 0.1 seconds on average. From a cold start, it takes around 35 seconds to acquire a GPS fix, while from a hot start, it takes approximately 8 seconds. The antenna can operate at high speeds of up to 515 meters per second and can withstand accelerations of up to 4g. It is powered by a 5V DC supply connected via a USB port and consumes around 60mA of power. The antenna outputs GPS data in various formats, including GGA, GSA, GSV, RMC, VTG, and GLL v2.2, following the NMEA 0183 standard. The chosen GPS daemon was gpsd.

Power Bank: The ROMOSS Sense 4s power bank is utilized to power the Raspberry Pi, the smartphone, and the GPS antenna in the experimental setup. With a capacity of 11000mAh, it provides an ample power supply for these components, enough to support the operation of the equipment throughout at least one round of experiments. This ensures uninterrupted functionality and reliable performance. It features one input port with a rating of DC5V-2.1A MAX and two output ports delivering DC5V-2.1A MAX and DC5V-1A MAX respectively.

Smartphone for 5G Measurements: The Xiaomi Mi 10 5G features 8GB of RAM, allowing for smooth multitasking and efficient performance and an octa-core CPU, clocked at a maximum speed of 2.84GHz, that ensures fast and responsive operation. The device runs on Android 11, and its kernel version is 4.19.113.

4.1.3 Network Configuration

The experimental configuration required two different paths between sender and receiver: WiFi and 5G. A layout of this network is shown in Fig. 4.3.

For the WiFi connection, the RP is connected to the WiFi AP, which is directly linked to the edge node. The edge node resides within the DMZ of Instituto de Telecomunicações in FEUPNet. The AP is connected to a network port in a building at FEUP using an ethernet connection. This network port has been assigned an IP address from the DMZ.

Regarding the 5G connection, the RP is connected to a 5G capable smartphone via USB tethering. The smartphone acts as a 5G modem. The data travels from the RP to the 5G antenna. It then passes through the cellular service provider, MEO. From there, it reaches the Internet and ultimately arrives at FEUPNet, where it is directed to the edge node.

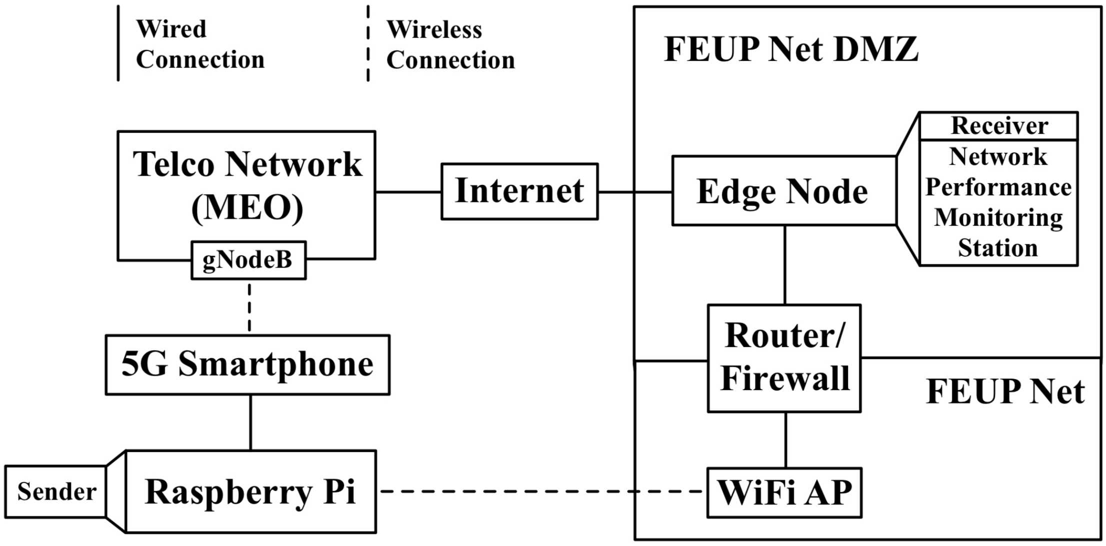

Figure 4.3: Network layout scheme

Several configurations had to be performed in the RP and the edge node. On the edge node, specific ports were opened using iptables to allow traffic for RTP, RTCP, and network performance monitoring station.

```
iptables -A INPUT -i em1 -p <udp/tcp> –dport <port_number> -j ACCEPT
```

Port forwarding was established using iptables to direct traffic to the respective Docker containers.

```
iptables -A PREROUTING -i em1 –dport (8080/8090/8091) -j DNAT –to <docker container IP>
```

On the RP, two routing tables were configured, one for WiFi and another for 5G, with associated marks for each table.

```
ip rule add fwmark 70 table wifi_t
ip rule add fwmark 71 table fiveg_t
```

Routes were added to each table to specify the gateway and interface for WiFi and 5G connections.

```
ip route add default <wifi/5g-gateway> dev <wifi/5g_interface> table <table_name>
```

To mark packets for the respective tables, iptables was used to identify packets based on the port-base.

```
iptables -t mangle -A OUTPUT -p udp –sport <8000/8001/8002/8003> -j MARK –set-mark 70/71
```

4.1.4 Limitations

An interesting finding, common to all measurements, was the observation of high and consistent variation in latency within the WiFi path, especially in the farther zones. This suggests that the RP's hardware may struggle to maintain a stable and reliable connection at such distances or signal strengths. However, considering the remaining time constraints for the work, the focus was put on improving the current setup rather than introducing changes.

Additionally, as mentioned in Sec. 3.2.3, the network performance monitoring station had limited data available, which posed challenges when making scheduling decisions. The effectiveness of these decisions relies heavily on the availability of comprehensive and up-to-date network information. Unfortunately, the network performance monitoring station in this study did not contain as much data as desired, possibly hindering the accuracy and efficiency of the scheduling process.

4.2 Performance Metrics

Throughput, latency, and packet loss will be quantitatively measured using the developed Python program, which will analyze the experiment data. Video quality will be assessed using PSNR.

In addition to these quantitative metrics, video quality will be assessed through a visual examination using a naked eye approach. This subjective analysis allows for the detection of artifacts or visual distortions resulting from the transmission process. This method is particularly suitable as it aligns with the subsequent application of computer vision techniques on the received video. By incorporating human perception, visible anomalies can be identified and evaluated.

By combining quantitative analysis of throughput, latency, and packet loss with the qualitative assessment of video quality, a comprehensive evaluation of the system's performance can be obtained. This multi-faceted approach provides a holistic understanding of the impact of different schedulers on the overall video transmission process.

4.2.1 Measurements

All measurements are done at the application level. For each test, data was collected from both the sender and the receiver in the form of logs. These logs captured essential information for each sent and received RTP packet, including:

- Sequence Number
- Fragment Number
- Total Number of Fragments
- Used Interface
- Timestamp
- Number of Bytes

These logs provide insights into the transmission process and allow for detailed analysis of the system's performance. To process and analyze these logs, a Python program was developed, which will be described in Sec. 4.2.4. This program facilitated the aggregation, interpretation, and visualization of the collected logs, enabling a thorough examination of the experimental results.

One of the challenges encountered was obtaining precise timestamps across all devices, particularly for measuring small time intervals, due to desynchronisation caused by clock drift, depicted in Fig. 4.4. To overcome this challenge, a synchronization mechanism was implemented, where the server acted as an NTP server and the RP synchronized with it periodically. This ensured the

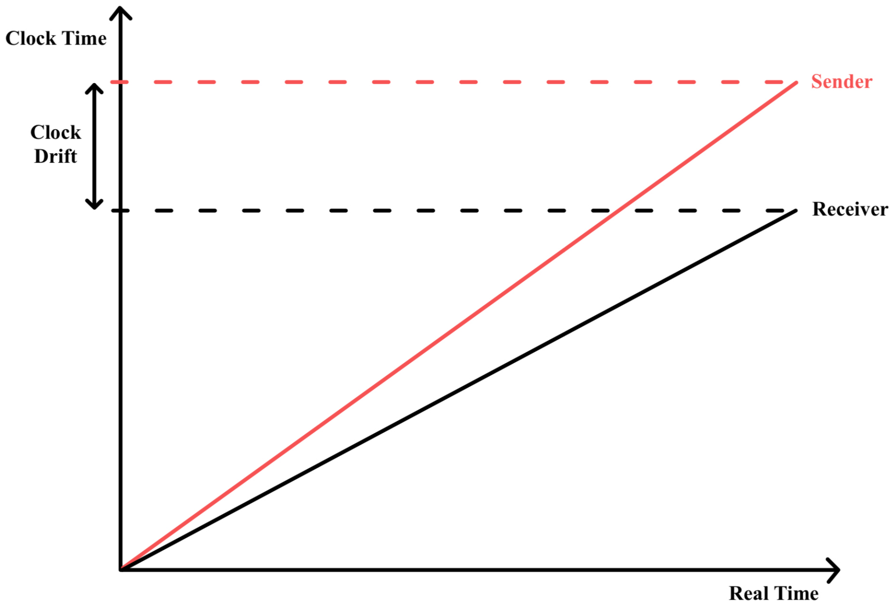

Figure 4.4: Clock drift illustration

alignment of the RP's clock with the server's reference time, minimizing clock drift and enabling accurate timestamping for reliable performance measurement. The difference between these 2 cases is illustrated in Fig. 4.5.

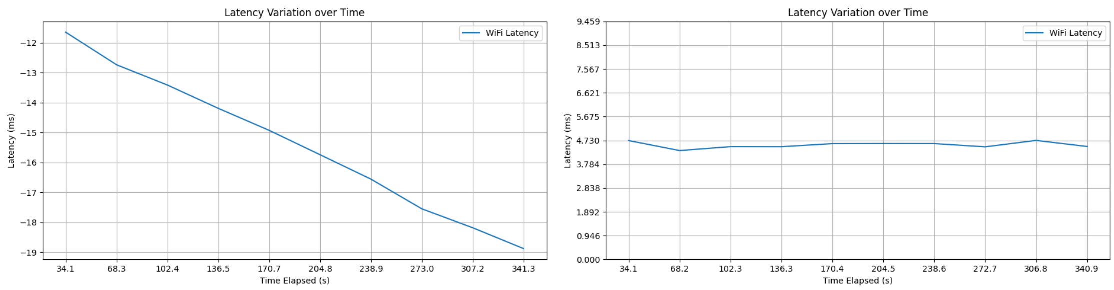

Figure 4.5: Comparison of latency without and with NTP synchronization, respectively

4.2.2 Data Logging

The MA-RTP protocol already features a logging system, to facilitate the collection and extraction of this data, making it a logical choice to incorporate the required metrics into the logs metrics.

To ensure the effective organization and accessibility of the logged data, a structured approach was adopted. The logs were organized in a patterned format as follows: "Data1Name:Data1: Data2Name:Data2:...". This structured format allows for easier data extraction and analysis. An illustration of this data organization pattern is depicted in Fig. 4.6.

```
1683811536:Metrics:Sender:SequenceNumber:9642:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811536685:Bytes:365
1683811536:Metrics:Sender:SequenceNumber:9643:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811536720:Bytes:16978
1683811536:Metrics:Sender:SequenceNumber:9644:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811536755:Bytes:363
1683811536:Metrics:Sender:SequenceNumber:9645:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811536789:Bytes:321
1683811536:Metrics:Sender:SequenceNumber:9646:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811536823:Bytes:312
1683811536:Metrics:Sender:SequenceNumber:9647:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811536858:Bytes:15717
1683811536:Metrics:Sender:SequenceNumber:9648:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811536893:Bytes:366
1683811536:Metrics:Sender:SequenceNumber:9649:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811536927:Bytes:307
1683811536:Metrics:Sender:SequenceNumber:9650:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811536962:Bytes:297
1683811536:Metrics:Sender:SequenceNumber:9651:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811536998:Bytes:17138
1683811537:Metrics:Sender:SequenceNumber:9652:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811537033:Bytes:503
1683811537:Metrics:Sender:SequenceNumber:9653:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811537067:Bytes:284
1683811537:Metrics:Sender:SequenceNumber:9654:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811537102:Bytes:339
1683811537:Metrics:Sender:SequenceNumber:9655:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811537137:Bytes:17327
1683811537:Metrics:Sender:SequenceNumber:9656:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811537172:Bytes:1215
1683811537:Metrics:Sender:SequenceNumber:9657:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811537206:Bytes:502
1683811537:Metrics:Sender:SequenceNumber:9658:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811537240:Bytes:707
1683811537:Metrics:Sender:SequenceNumber:9659:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811537275:Bytes:17641
1683811537:Metrics:Sender:SequenceNumber:9660:FragmentNumber:0:TotalFragments:1:Interface:0:Timestamp:1683811537310:Bytes:2229
1683811537:Metrics:Sender:SequenceNumber:9661:FragmentNumber:0:TotalFragments:1:Interface:1:Timestamp:1683811537344:Bytes:1022
```

Figure 4.6: Screenshot of logs

4.2.3 File Structure

Logs and videos resulting from experiments are stored in accordance with the file structure depicted in Fig. 4.7. This structure serves as a systematic organization scheme, allowing for the efficient storage and retrieval of the relevant data. By adhering to this structure, the logs and video files from each test can be easily located and referenced, facilitating further analysis and comparison of the experimental results.

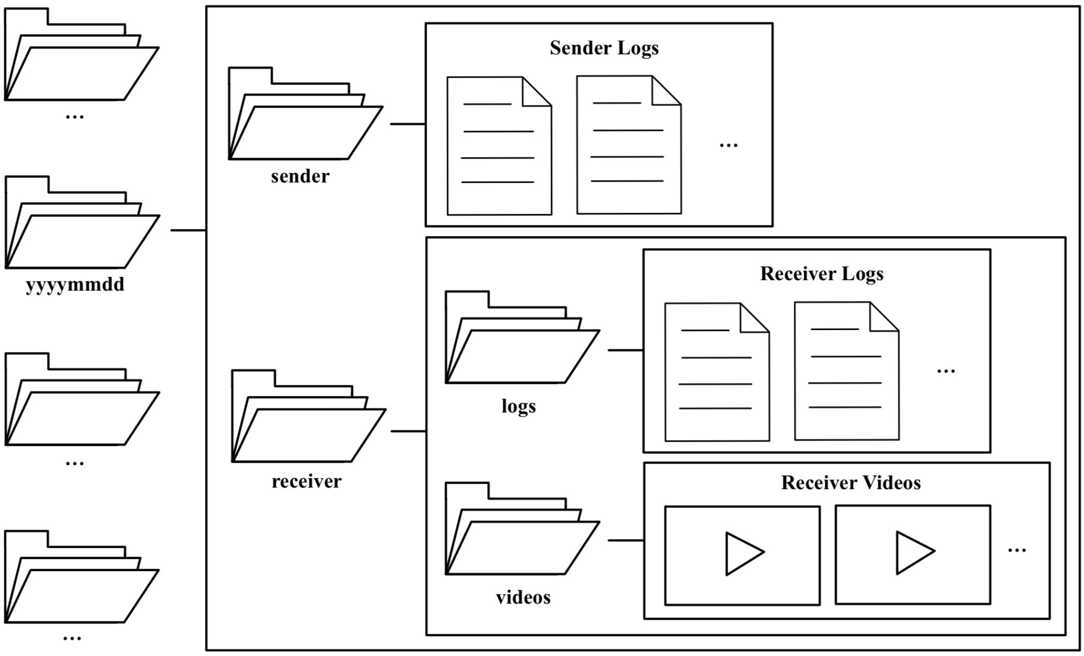

Figure 4.7: Depiction of the file structure used for data storage

4.2.4 Performance Analysis Methodology

An integral part of the performance metrics analysis is the methodology used to analyze the collected data. In order to obtain the performance metrics, the relevant information is extracted from the logs and subsequently processed using a Python program. By following a structured approach, the program provides a systematic and efficient means of analyzing and presenting the performance metrics, enabling an in-depth evaluation of the schedulers.

This program takes the sender and receiver log files as input from the command line, and an optional parameter for specifying the number of time intervals for graphing. The script extracts relevant information from the log files, such as sequence numbers, timestamps, interface type, and bytes sent/received. It then matches sent and received RTP packets using sequence numbers and interface type.

Throughput is calculated for WiFi, 5G, and both interfaces combined. Latency is calculated separately for WiFi, 5G, and both interfaces combined. Packet loss analysis is performed by determining the number and percentage of sent, received, and lost RTP packets for both WiFi and 5G interfaces. The mathematical expressions used to calculate these values are shown in Sec. 4.2.4.1.

The program also divides the transmission time into intervals. Within each interval, the number of RTP packets and bytes sent for WiFi, 5G, and both interfaces combined are calculated. This information provides a more detailed understanding of how RTP packet transmission is distributed over time. The metrics are presented in an organized format using the matplotlib library. The data processing and analysis pipeline from logging to result presentation is illustrated in Fig. 4.8.

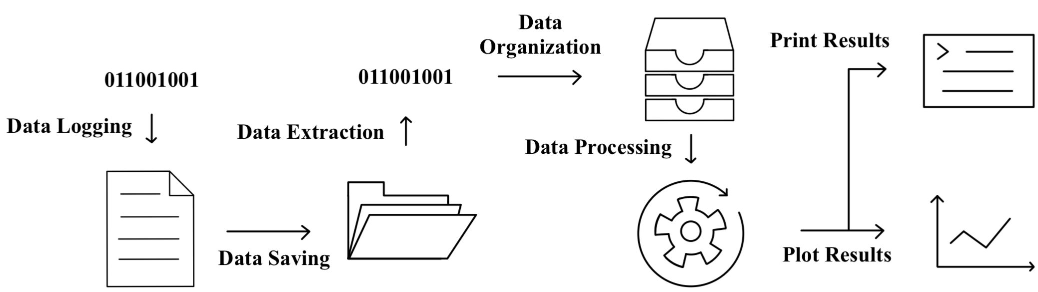

Figure 4.8: Flowchart illustrating the data processing and analysis pipeline from logging to result presentation

4.2.4.1 Metric Calculation Formulas

$$Transmission\ Time = t_{transmission} = timestamp_{received_{last}} - timestamp_{received_{first}}$$

$$Throughput = \frac{n_{bits_{received}}}{t_{transmission}}$$

$$Latency = \frac{\sum_i^n (timestamp_{received_i} - timestamp_{sent_i})}{n}$$

$$Lost\ Packets = \frac{n_{packets_{lost}}}{n_{packets_{sent}}} \qquad Lost\ Bits = \frac{n_{bits_{lost}}}{n_{bits_{sent}}}$$

$$Late\ Frame\ Arrivals = \frac{n_{late\_frames_{zone}}}{n_{sent\_frames_{zone}}} \quad , \quad zone = 1,2,3,4$$

$$File\ Size = \frac{file\_size_{receiver}}{file\_size_{original}}$$

Table 4.1: Metric calculation variables' descriptions

Variable	Description
$timestamp_{sent_i}$	Timestamp from when RTP packet *i* was sent
$timestamp_{received_i}$	Timestamp from when RTP packet *i* was received
$n_{packets_{sent}}$	Number of sent RTP packets
$n_{packets_{received}}$	Number of received RTP packets
$n_{packets_{lost}}$	Number of lost RTP packets
$n_{sent\_frames_i}$	Number of frames sent in zone *i*
$n_{late\_frames_i}$	Number of late arriving frames in zone *i*
$file\_size_{original}$	File size of the original video
$file\_size_{receiver}$	File size of the received video

4.2.5 Limitations

During the testing and result analysis phase of this work, it became evident that the chosen video quality metrics were inadequate for evaluating the performance of the schedulers. This limitation stemmed from the current state of the MA-RTP protocol, discussed in Sec. 3.1.3, which presented challenges in assessing video quality accurately. One notable constraint was the reliance on completing the entire video transmission before initiating the reconstruction process. Consequently, this approach did not consider the possibility of late arrivals of frames, leading to incomplete analysis. As such, relying solely on the obtained video quality data would not yield reliable and precise results for evaluating the scheduler's performance.

Furthermore, the absence of a mechanism to replace frames in the event of RTP packet loss posed additional challenges. Instead of recovering from RTP packet loss, the protocol simply skips the affected frame. This is an issue because the system should be close to what a real system will do, and error concealment is a basic function of a media receiver.

To address the limitation of video reconstruction, a statistical method was developed to evaluate late frame arrivals and determine their inclusion in the analysis. However, it is important to note that this solution only provides an estimation of the resulting video quality, rather than a comprehensive analysis of the received video. Therefore, while efforts were made to mitigate the limitations, it is essential to consider the potential impact on the accuracy and reliability of the evaluation.

Chapter 5

Experiments

The tests consist of streaming a pre-recorded video while walking through four distinct pre-defined locations (zones), represented in green in Fig. 5.1. This allows the transmission under different positions and different wireless environments, and the use of data from the network performance monitoring station that is associated with these different points. The walking route follows the sequence of zones 4, 1, 2, 3, 4. The duration of stay in each zone is set at 1 minute and 30 seconds, except for the first and last zones, both of which are zone 4. In the first zone, the subject immediately proceeds to the next zone without any additional delay, while in the last zone, more time may be required to complete the video transmission. The time taken to walk to each zone is considered in the analysis of the results. Fig. 5.1 also contains the placement of the WiFi AP and the 5G gNodeB, respectively marked W (in blue) and F (in red).

The pre-recorded video consists of a street scene capturing passing cars and pedestrians. It was paced to be sent as a stream rather than a file. It has a duration of 5 minutes and 28 seconds, equivalent to 328 seconds. The video has a resolution of 640x480 pixels and utilizes the H.264 codec with the High 4:2:2 Profile. It has a frame rate of 30 frames per second, with a bitrate of 1676 kbps. Furthermore, the video follows a Group of Pictures (GoP) of size 4.

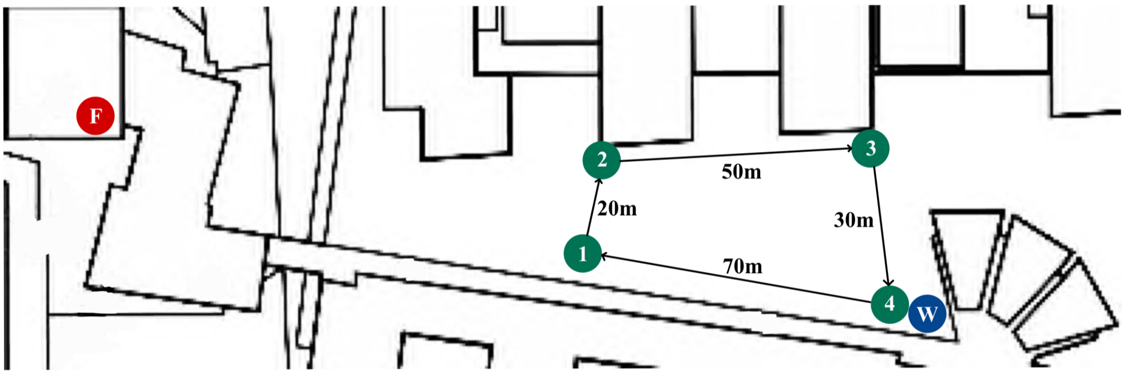

Figure 5.1: Testing route

5.1 Experiment Rounds

For the first round of testing, the following schedulers were used:

- Redundant
- Round-robin
- 5G-only
- WiFi-only

The purpose of this round of testing is to assess the conditions within the testing environment, specifically focusing on aspects such as delays and errors associated with each interface. By employing the four different schedulers mentioned earlier, we aim to evaluate how each scheduler performs and ascertain their impact on the overall system performance. This assessment will provide valuable insights into the behavior and effectiveness of the schedulers in managing delays and errors encountered during video transmission. By considering these factors, we can gain a comprehensive understanding of the performance characteristics and limitations of each interface, aiding in the improvement of video transmission in this scenario.

These schedulers, although basic in nature, serve as effective tools for studying the environment and analyzing various aspects of the system. It is worth noting that these schedulers have already been implemented in the ma-rtp protocol, which further justifies their selection for this round of testing. Their simplicity enables us to isolate and evaluate specific factors, such as delays and errors, without the complexities introduced by more advanced scheduling algorithms. By leveraging the existing implementation, we can focus on examining the behavior and performance of each scheduler within the testing environment. This choice allows us to gather meaningful insights into the system's performance characteristics and lays the foundation for future improvements and enhancements.

For the second round of testing, the following schedulers were used:

- Redundant
- Round-robin
- WiFi-only
- 5G-only
- Context-based average throughput
- I Frame to WiFi
- I Frame to 5G

In the previous round of testing, an issue was identified with timestamp recording on the sender side. Time synchronization was not occurring, leading to significant clock drifting on the RP, as previously addressed in Sec. 4.2.1. The encountered issue was due to a configuration in the receiver's NTP settings, which restricted requests to the local network. To resolve this, the configuration was modified to allow requests from the sender's IP address. Lastly, to ensure accurate results, all previously tested schedulers were retested.

Fortunately, certain results, such as the received video file, were unaffected by the timestamp issue since the timestamps were used for logging purposes. Upon analyzing each video, it became evident that I frames encountered difficulties during WiFi transmission. For this reason, two additional schedulers were introduced to further investigate this issue: 'I Frame to 5G' and 'I Frame to WiFi'.

Additionally, the 'Context-based average throughput' scheduler was included in this round of testing. Although it had already been implemented in the ma-rtp protocol before this work, the testing couldn't be conducted earlier due to incomplete setup of both the GPS antenna and network performance monitoring station. Unfortunately, due to a fault in the code, the protocol stopped very soon after beginning transmission, and no usable data was collected for this scheduler. We were not able to explain why this happened.

For the third round of testing, the following schedulers were used:

- Random
- Round-robin based on GoP Size
- Round-robin based on Frame Size
- Zone based
- Packet Loss Rate

For this round, two additional schedulers, 'Random' and 'Zone based', were employed to explore any unique aspects that may not have been captured or easily detectable with the other schedulers. The 'Random' scheduler aimed to identify any unforeseen factors or patterns, while the 'Zone based' scheduler served to confirm or refute assumptions and conclusions previously made.

An issue observed in the previous round was related to the 'Round-robin' scheduler. Due to the GoP size of 4 and the nature of the scheduler, the distribution of bits per path is asynchronous, with all I frames being sent through the same interface. This effect can be visualized in Fig. 3.4. To address this issue, two additional round-robin schedulers were implemented: 'Round-robin based on GoP Size' and 'Round-robin based on Frame Size'. The 'Round-robin based on GoP Size' scheduler, illustrated in Fig. 3.5, considers the entire GoP as a unit and only changes the interface after the complete transmission of a GoP. The 'Round-robin based on Frame Size' scheduler, depicted in Fig. 3.6, treats I frames and the remaining frames separately, essentially functioning as two independent round-robin schedulers. These approaches ensures a more balanced distribution across both interfaces.

Lastly, the remaining 'Packet Loss Rate' scheduler accounts solely for packet loss. It uses historical data from the network performance monitoring station to determine which path is less likely to lose the frames.

For the fourth round of testing, the following scheduler was used:

- Timely Arrival

In this final round of testing, the primary focus is on evaluating the performance of the 'Timely Arrival' scheduler. Throughout the previous rounds, it has become evident that packet loss is a significant performance indicator. This scheduler takes this into account, similar to the 'Packet Loss Rate' scheduler. However, it also incorporates consideration of RTT as an additional factor. Since the video is streamed, it is crucial for the scheduler to make every effort to deliver every frame on time. Therefore, the scheduler selectively sends frames through interfaces with sufficiently low RTT, if available. Only after considering the RTT does it assess the packet loss rates.

It was due to the unavailability of RTCP receiver reports during the initial testing phase that RTT has not been accounted for in the previous rounds. The network performance monitoring station was used as an alternative for this purpose. It is not ideal, however, as it does not provide real-time feedback on the network conditions, only historical data.

The testing schedules for all rounds are illustrated in Table 5.1. Table 5.2 shows the correspondence between the identifiers used and its respective schedule, as well as the number of tests executed per scheduler.

Table 5.1: Testing rounds' schedule

Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Round 1	rd	rr	fg	wf	rd	rr	fg	wf	-	-	-	-	-	-
Round 2	rd	at	rr	fg	wf	iw	i5	rd	at	rr	fg	wf	iw	i5
Round 3	rand	rrg	rrs	zb	pl	rand	rrg	rrs	zb	pl	-	-	-	-
Round 4	ta	ta	ta	-	-	-	-	-	-	-	-	-	-	-

5.2 Limitations

While the path from the AP to the receiver is exclusively used for these experiments, the 5G path is subject to potential variations due to shared infrastructure usage. Given the dynamic nature of network traffic, it is expected that the network usage fluctuates over time. Because of that, the experiments are conducted with no interruption, to minimize variation between tests.

While the WiFi path remains consistent in terms of usage throughout the experiments, the 5G path, being subject to shared infrastructure utilization, introduces a variable that may impact the experimental results. It is important to acknowledge that the dynamic nature of the network and varying conditions may lead to fluctuations in performance and results. By acknowledging this limitation, we can approach the analysis and interpretation of the experimental outcomes with a realistic understanding of the potential impact of the network state on the observed variations.

Table 5.2: Identifier / Scheduler pairs

Identifier	**Scheduler**	**No. of Tests Executed**
rd	Redundant	4
rr	Round-robin	4
at	Context-based average throughput	2
wf	WiFi-only	4
fg	5G-only	4
iw	I Frame to WiFi	2
i5	I Frame to 5G	2
rand	Random	2
rrg	Round-robin based on GoP Size	2
rrs	Round-robin based on Frame Size	2
zb	Zone based	2
pl	Packet Loss Rate	2
ta	Timely Arrival	3

Chapter 6

Results and Discussion

The following graphics depict the variation in throughput and latency, as well as the RTP packets and bits lost over time. Each graph includes red vertical limits that delineate different zones, along with grey vertical limits indicating the transition areas between zones.

The throughput graphics presented in the analysis feature a horizontal line representing the bitrate of the video file used in the experiments. Additionally, four throughput measurements are included: WiFi, 5G, and total throughput, observed at the receiver, and sender throughput observed at the sender. The latency graphics depict the latency measurements for WiFi, 5G, and total latency. The packet loss graphics illustrate the RTP packet loss measurements for WiFi, 5G, and total RTP packet loss. Lastly, the bit loss graphics present the bit loss measurements for WiFi, 5G, and total bit loss.

The tables present the average results obtained for each scheduler. Green values indicate the best performance, while red values indicate the worst. Any adjusted value is marked with an asterisk (*). Each scheduler is represented by its identifier, which can be translated using Table 5.2. The late frame arrivals statistic indicates the percentage of RTP packets that arrived late for each interface and zone. The file size statistic displays the relative size of the received file compared to its original size.

Due to the issues with the reassembly of the video file, discussed in Sec. 4.2.5, the quality of the received video could not be considered when assessing the respective scheduler. To circumvent this issue, the number of late arriving frames for each interface and each zone was collected and is displayed in the tables present in this chapter. This way, we can at least have an estimated idea of the state of the received video.

Finally, it is important to note that the presented results correspond to a single run of each scheduler, which is considered representative of the remaining runs.

6.1 Establish Baseline

The objective of the following tests is to gain insights into the behavior of the system. Although the schedulers employed in these tests are relatively simple, lacking feedback responsiveness and contextual awareness, they serve as a valuable starting point for the subsequent analysis. The 'Redundant' scheduler provides an understanding of how the system responds to redundancy, while the 'Round-robin' scheduler offers insights into the system's behavior when the selected path changes. Additionally, the 'WiFi-only' and '5G-only' schedulers shed light on the specific characteristics of each path and how the system operates when a particular path is consistently selected.

For the 'Redundant' scheduler, the measurements indicate significantly lower throughput than expected in zones 1 and 2. It is worth noting that the WiFi and 5G curves appear to exhibit interdependence, although the underlying cause of this behavior remains unclear. The latency curve, on the other hand, follows a consistent pattern across all tests. In the case of 5G, the latency remains constant throughout the entire data transfer, while WiFi latency achieves much higher values than 5G in zones 1 and 2, but lower values in zones 3 and 4. Regarding RTP packet loss, both interfaces exhibit considerable variability. Notably, WiFi RTP packet loss demonstrates a consistent trend across all tests, with minimal RTP packet loss near the AP in zone 4. Lastly, a peak in RTP packet loss is occasionally displayed near the transmission's end. Despite our efforts, we could not understand why it happened. These findings are visually represented in Fig. 6.1.

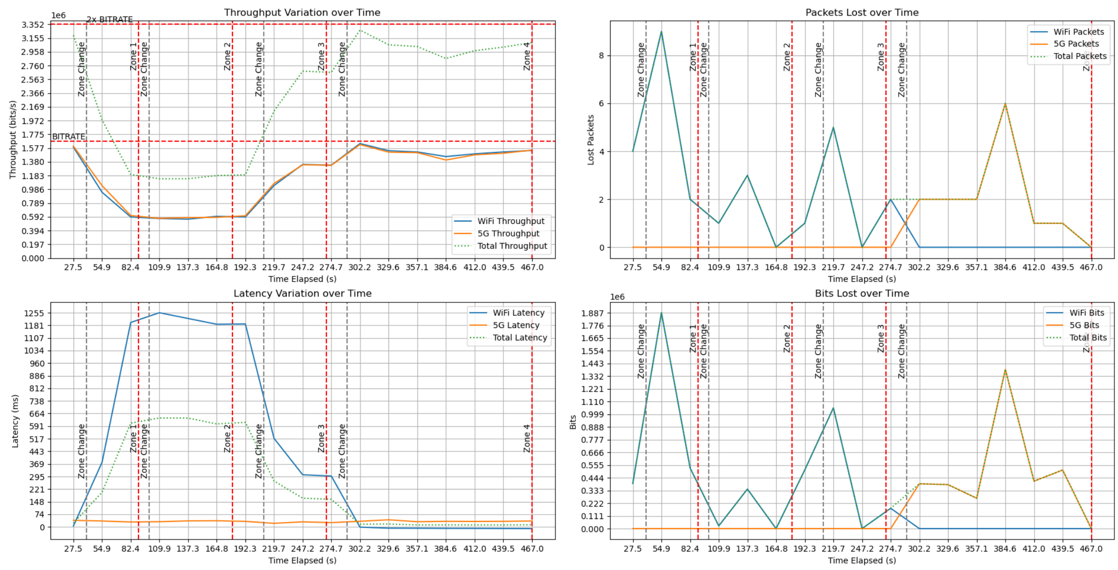

Figure 6.1: Redundant scheduler results

For the 'Round-robin' scheduler, the measurements again indicate interdependence between both throughput values. However, unlike 'Redundant', this occurrence can be attributed to the nature of the scheduler, combined with a GoP size of 4. In this setup, the larger I frames are consistently transmitted via the 5G interface. Moreover, towards the end of the transmission, some I frames occur more irregularly, leading to a switching of the relationship between the throughput values a few times. Latency follows a similar pattern observed in the previous scheduler. The 5G interface exhibits a relatively constant level of RTP packet loss, with a noticeable peak towards the end, similar to the 'Redundant' scheduler. The WiFi interface displays RTP packet loss following the previously mentioned trend. These findings are presented in Fig. 6.2.

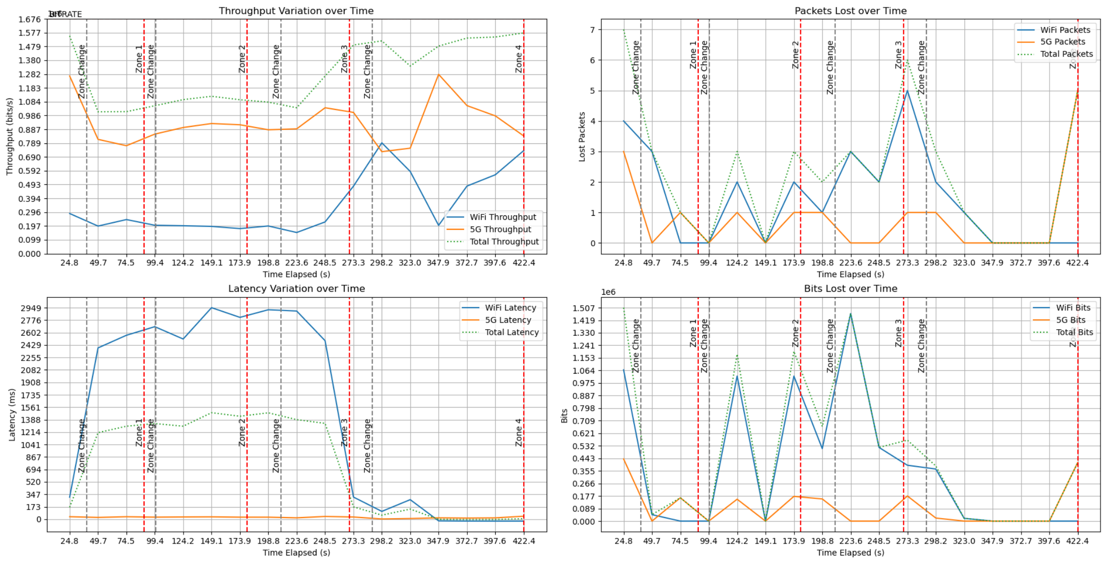

Figure 6.2: Round-robin scheduler results

The '5G-only' scheduler demonstrates consistent throughput throughout the testing period. Similarly, the latency remains relatively constant, as observed in the other schedulers. The RTP packet loss also remains constant, except for the peak trend observed towards the end of the transmission. These findings are visually represented in Fig. 6.3.

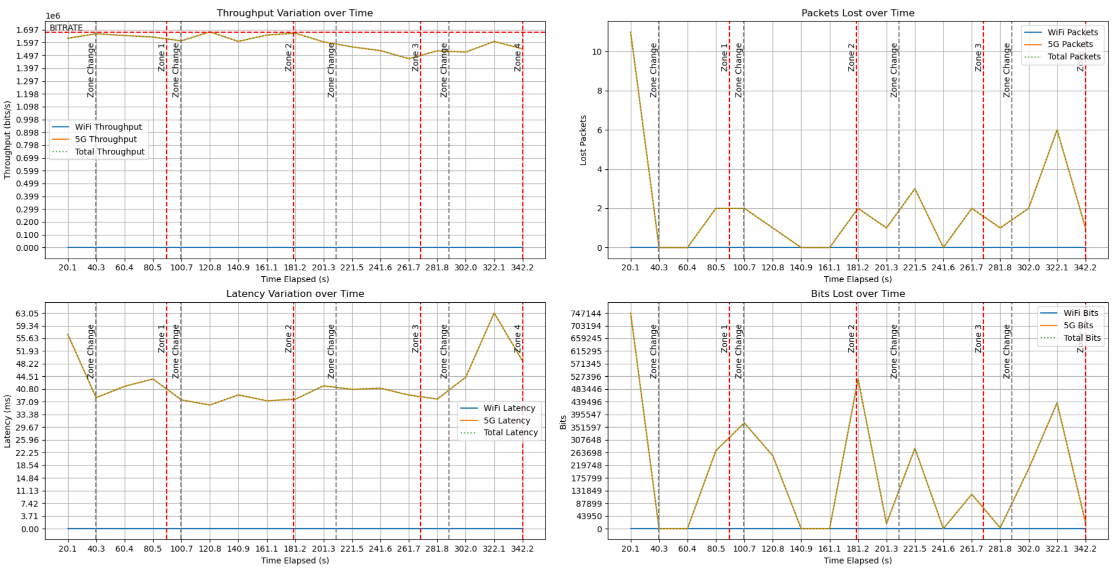

Figure 6.3: 5G-only scheduler results

The behavior observed in the 'WiFi-only' scheduler aligns with the findings from the 'Redundant' scheduler. Both schedulers exhibit similar patterns and characteristics. The details of these findings are visually presented in Fig. 6.4.

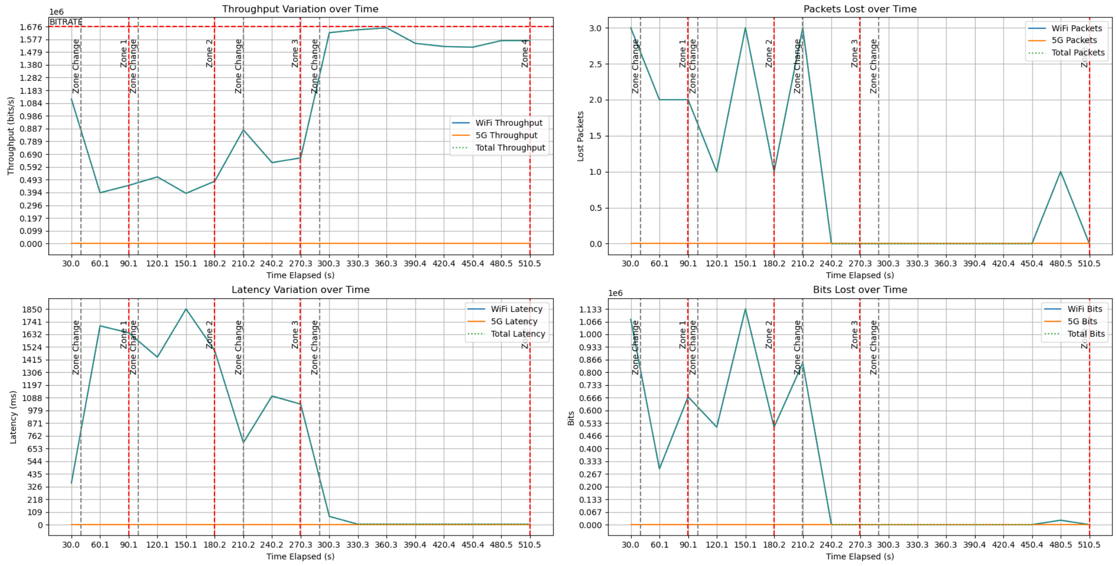

Figure 6.4: WiFi-only scheduler results

Table 6.1 presents the average results obtained for these schedulers. (*) In the case of 'Redundant', the total average throughput has been halved to account for the transmission of twice as much data.

Redundant: Applying redundancy seems to be a poor strategy when applied to every packet, mostly due to the drastic increase in transmission time. That means that it uses twice as much bandwidth while yielding poorer results. RTP packet loss seems to have no effect on file size, which is to be expected when transmitting every frame twice.

Round-robin: This scheduler stands out with the lowest file size, which aligns with the rates of RTP packet and bit loss. This scheduler's results seem to be a combination of the results from '5G-only' and 'WiFi-only', with a couple of differences: the latency and RTP packet loss in the WiFi path are much higher on this scheduler; this suggests that changing paths might have a negative effect on performance.

5G-only: This scheduler performs well in terms of transmission time. However, it exhibits higher rates of late frames in all zones, which suggests that more frames remain unused due to their delay.

WiFi-only: This scheduler has the weakest performance in transmission time. Most frames arrive late in all zones except for zone 4, closer to the AP. Packet loss seems to be lower, which further validates the suggestion that path changes are detrimental to performance. Conversely, bit loss is disproportionately higher, which suggests most of the lost RTP packets correspond to I frames.

Table 6.1: Average results of each scheduler - Establishing Baseline

Scheduler	-	rd	rr	fg	wf
Transmission Time (s)	-	466.96	422.417	342.242	510.486
Average Throughput (bits/s)	WiFi	1,168,814.237	347,990.332	-	1,068,869.806
	5G	1,172,164.228	936,455.569	1,599,684.177	-
	Total	1,170,489.233*	1,284,445.901	1,599,684.177	1,068,869.806
Average Latency (ms)	WiFi	612.037	2,156.162	-	339.627
	5G	32.744	32.614	42.653	-
	Total	204.791	918.069	42.653	339.627
Lost Packets (%)	WiFi	0.14	0.25	-	0.16
	5G	0.07	0.14	0.34	-
	Total	0.21	0.39	0.34	0.16
Lost Bits (%)	WiFi	0.45	1.17	-	0.92
	5G	0.31	0.31	0.59	-
	Total	0.75	1.48	0.59	0.92
Late Frame Arrivals (WiFi) (%)	Zone 1	45.06	74.15	-	74.98
	Zone 2	99.47	99.57	-	99.29
	Zone 3	84.34	76.88	-	100.00
	Zone 4	0.00	9.67	-	0.05
Late Frame Arrivals (5G) (%)	Zone 1	39.27	29.40	62.11	-
	Zone 2	46.46	44.25	48.76	-
	Zone 3	26.69	41.34	66.95	-
	Zone 4	36.39	20.81	88.51	-
File Size (%)	-	100.00	96.62	98.88	98.50

6.2 Improving Delivery by Frame Type

Results from the previous experiments, from comparing RTP packet and bit loss values and from the visualization of the received video, show that I frames were more prone to errors or loss than other types of frames. Given this, the objective of these tests is to improve delivery based on frame type. Two schedulers were employed for this end: 'I Frame to WiFi' and 'I Frame to 5G'.

The measurements obtained for the 'I Frame to WiFi' scheduler reveal unexpectedly lower throughput values in zones 1 through 3. This is most likely the result of losing the I frames in the WiFi path, as throughput in the 5G path remains constant. Furthermore, the absence of RTP packet loss on the 5G path is expected, as smaller frames have a lower likelihood of being lost. Another noteworthy observation is the lower latency values in the WiFi path compared to 5G in zone 4, accompanied by no RTP packet loss. These results align with the previous findings, reinforcing the overall trend. These findings are visually represented in Fig. 6.5.

The measurements obtained for the 'I Frame to 5G' scheduler reveal a more consistent throughput, aligning closer with the expected values. Notably, there is a low but noticeable level of RTP packet loss observed for the 5G path, which now carries the I frames, while the RTP packet loss for WiFi

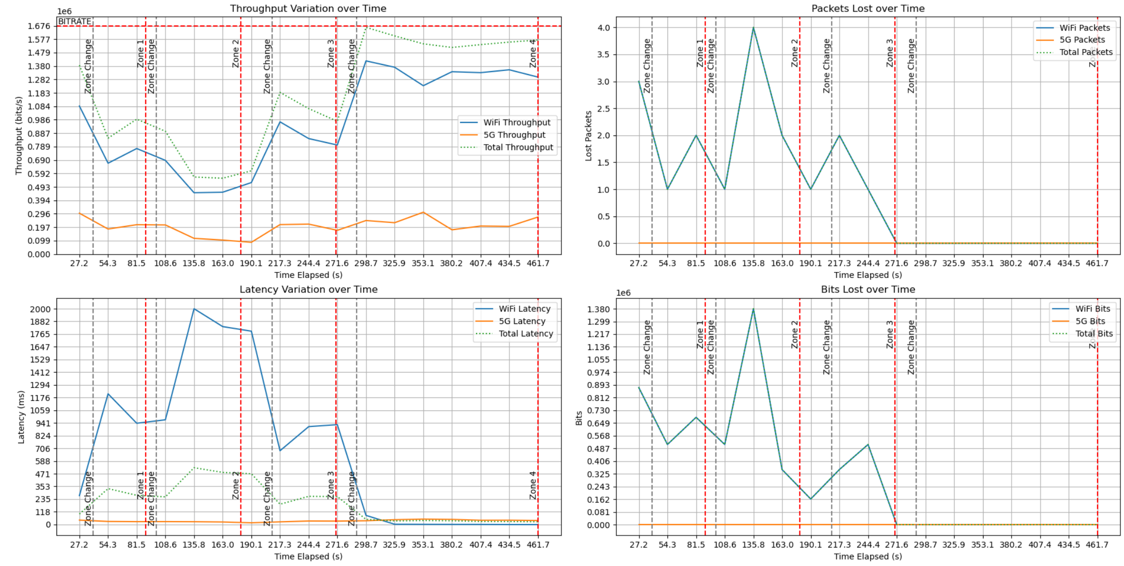

Figure 6.5: I Frame to WiFi scheduler results

is significantly lower. Regarding latency behavior, it remains consistent with previous findings, except for a peak in latency observed in the 5G path towards the end of the transmission. These findings are visually represented in Fig. 6.6.

Table 6.2 presents the average results obtained for these schedulers.

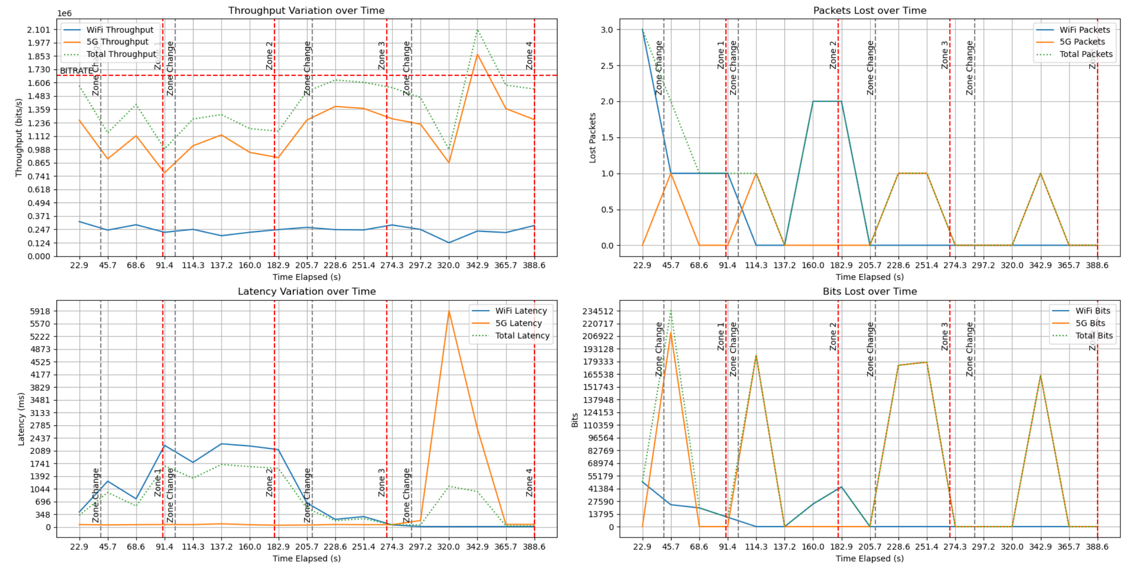

Figure 6.6: I Frame to 5G scheduler results

Table 6.2: Average results of each scheduler - Improving Delivery by Frame Type

Scheduler	-	iw	i5
Transmission Time (s)	-	461.697	388.604
Average Throughput (bits/s)	WiFi	977,107.809	242,078.105
	5G	204,123.189	1,172,291.155
	Total	1,181,230.998	1,414,369.260
Average Latency (ms)	WiFi	542.324	728.785
	5G	35.722	572.840
	Total	150.481	686.476
Lost Packets (%)	WiFi	0.17	0.10
	5G	0.00	0.05
	Total	0.17	0.15
Lost Bits (%)	WiFi	0.97	0.03
	5G	0.00	0.17
	Total	0.97	0.20
Late Frame Arrivals (WiFi) (%)	Zone 1	84.75	60.54
	Zone 2	96.81	99.81
	Zone 3	99.28	54.70
	Zone 4	0.00	0.10
Late Frame Arrivals (5G) (%)	Zone 1	30.45	99.20
	Zone 2	10.70	89.22
	Zone 3	35.14	97.83
	Zone 4	63.34	99.76
File Size (%)	-	94.37	96.76

I Frame to WiFi: Evaluating the performance of the 'I Frame to WiFi' scheduler reveals certain drawbacks. The transmission time and file size are notably poor compared to other schedulers. The high bit loss observed can be attributed to the fact that all frames transmitted via WiFi are I frames. Moreover, except for zone 4, the WiFi path exhibits high rates of late frame arrivals. In contrast, the 5G path benefits from transmitting smaller frames, resulting in relatively lower rates of late frame arrivals.

I Frame to 5G: The latency experienced on the 5G path is considerably higher compared to previous schedulers. This suggests RTP packet size might play a significant role in determining latency. On the other hand, late frame arrivals via WiFi show either similar or lower rates compared to the 'I Frame to WiFi', while the 5G path experiences much higher rates of late frame arrivals.

6.3 Maximizing Round-Robin Efficiency

The used 'Round-robin' scheduler was not suitable for this particular system. To properly evaluate this type of scheduler, modifications were made to ensure equal frame allocation to both paths. As a result, two new schedulers were introduced: 'Round-robin based on GoP Size' and 'Round-robin based on Frame Size', designed to distribute the workload evenly between the WiFi and 5G paths.

'Round-robin based on GoP Size' exhibits a lot of the behaviors displayed by 'Round-robin'. The biggest difference is the equal throughput distribution between both paths, even if the total throughput values are still significantly lower for zones 1 and 2. Latency follows the same pattern observed in previous schedulers. In terms of RTP packet loss, the 5G interface exhibits no loss for the first 2 zones and a noticeable increase in the last 2 zones. The WiFi interface displays RTP packet loss following the previously observed trend. These findings are visually presented in Fig. 6.7.

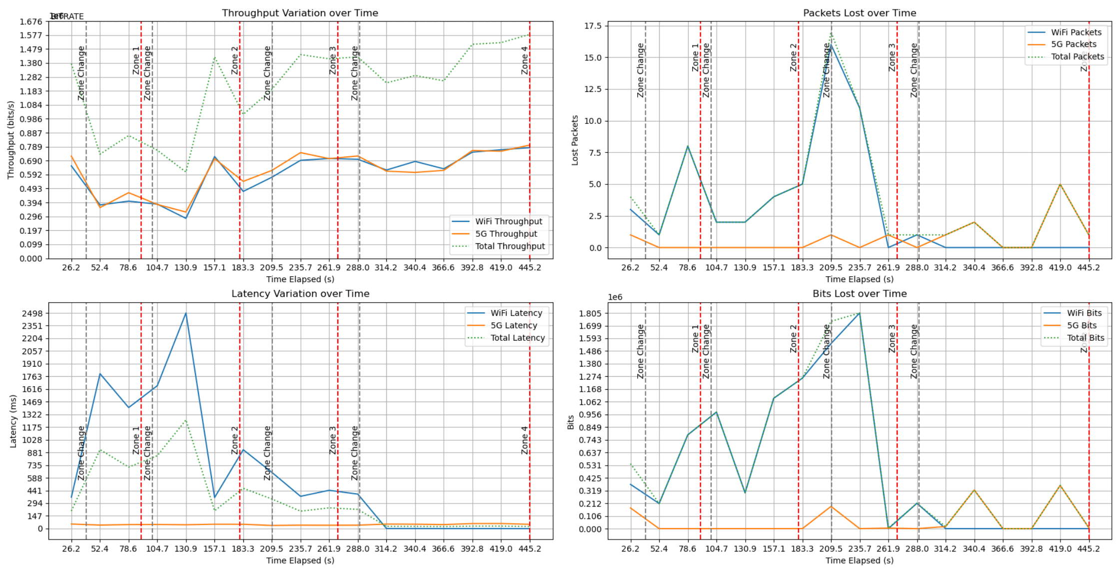

Figure 6.7: Round-robin based on GoP Size scheduler results

Similarly to 'Round-robin based on GoP Size', 'Round-robin based on Frame Size' exhibits a lot of the behaviors displayed by 'Round-robin'. Once again, there is an equal throughput distribution between both paths, even if the total throughput values are still significantly lower for zones 1 and 2. However, latency values are now more regular, with 5G being constant and lower, and WiFi being high initially and being nonexistent near the end. These findings are visually presented in Fig. 6.8.

Table 6.3 provides evidence that the system does not benefit from an even distribution of paths. It is apparent that the 5G path demonstrates greater capability in delivering all frames across most zones. Furthermore, the latency values indicate that frequent path switches have a negative impact on performance. Notably, the 'Round-robin' scheduler with the highest number of path changes exhibits the highest latency, while the 'Round-robin based on GoP Size' scheduler with the least path changes shows lower latency. Overall, it becomes evident that this type of scheduler is not suitable for improving performance in this particular system.

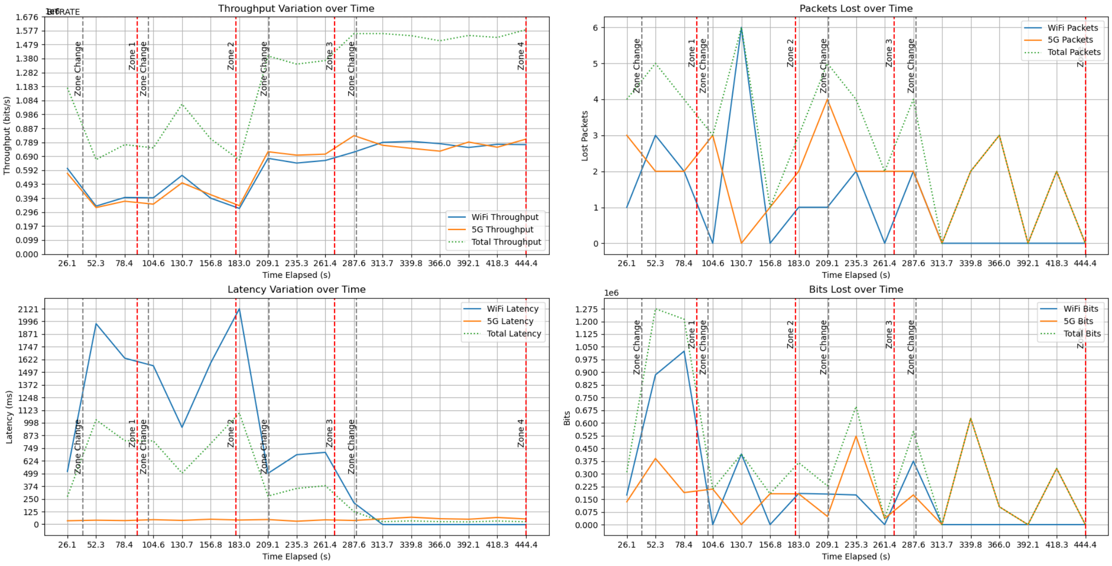

Figure 6.8: Round-robin based on Frame Size scheduler results

Table 6.3: Average results of each scheduler - Maximizing Round-Robin Efficiency

Scheduler	-	rr	rrg	rrs
Transmission Time (s)	-	422.417	445.159	444.407
Average Throughput (bits/s)	WiFi	347,990.332	600,307.036	610,137.536
	5G	936,455.569	615,176.510	614,340.746
	Total	1,284,445.901	1,215,483.546	1,224,478.282
Average Latency (ms)	WiFi	2,156.162	619.430	882.540
	5G	32.614	47.151	47.933
	Total	918.069	290.700	357.276
Lost Packets (%)	WiFi	0.25	0.54	0.18
	5G	0.14	0.12	0.30
	Total	0.39	0.66	0.49
Lost Bits (%)	WiFi	0.37	1.55	0.62
	5G	0.31	0.20	0.57
	Total	1.48	1.75	1.19
Late Frame Arrivals (WiFi) (%)	Zone 1	74.15	72.33	69.79
	Zone 2	99.57	70.77	78.97
	Zone 3	76.88	88.30	91.54
	Zone 4	9.67	0.05	0.00
Late Frame Arrivals (5G) (%)	Zone 1	29.40	65.43	53.25
	Zone 2	44.25	80.15	70.72
	Zone 3	41.34	47.54	54.97
	Zone 4	20.81	72.96	72.99
File Size (%)	-	96.62	94.87	95.08

6.4 Finding Anomalies in User-Defined and Random Scheduling

To further validate previous assumptions, the 'Zone based' scheduler was implemented, which allows the user to manually assign paths to each zone based on their perception of the system. This approach helps identify any flaws in the user's understanding of the system and its performance. In the conducted tests, zones 1 and 2 were assigned the 5G path, while zones 3 and 4 were assigned the WiFi path. Additionally, the 'Random' scheduler was introduced. This scheduler aims to uncover previously unnoticed aspects and potentially provide additional evidence to support any existing beliefs. By randomly selecting paths for transmission, this scheduler offers a different perspective on the system's behavior and performance.

The 'Random' scheduler does not introduce any new information and largely reinforces the findings observed in previous tests. The throughput values remain significantly below the expected levels, which is a common occurrence. Similarly, the latency and RTP packet loss patterns align with the typical trends observed in other schedulers. These findings are presented in Fig. 6.9.

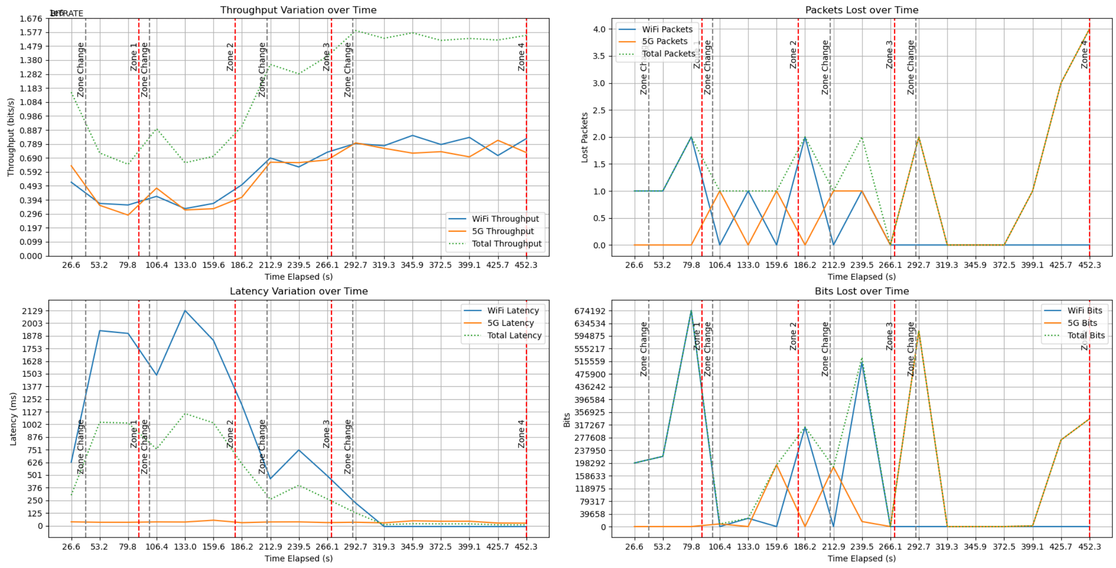

Figure 6.9: Random scheduler results

The 'Zone based' scheduler demonstrates the expected throughput performance, with dips observed at the beginning and during path switch moments. These dips correspond to latency spikes specifically in the WiFi path. RTP packet loss rates are negligible for WiFi, while occasional spikes are observed for 5G. These behaviors highlight the potential for leveraging the distinctive characteristics of both paths to achieve improved results. These findings are presented in Fig. 6.10.

Table 6.4 further validates the claims so far. Transmission time, file size, throughput, and latency values considerably better for schedulers with limited path changes, and considering path characteristics.

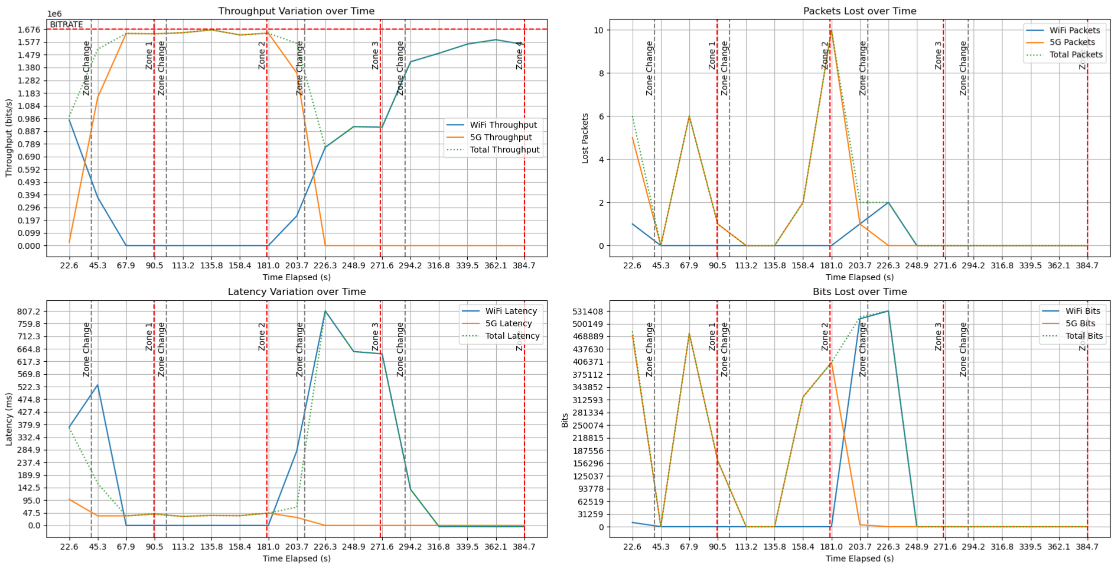

Figure 6.10: Zone based scheduler results

Table 6.4: Average results of each scheduler - Finding Anomalies in Arbitrary and Random Scheduling

Scheduler	-	rand	zb
Transmission Time (s)	-	452.309	384.729
Average Throughput (bits/s)	WiFi	617,521.703	694,580.091
	5G	592,168.221	729,340.809
	Total	1,209,689.924	1,423,920.900
Average Latency (ms)	WiFi	1,112.064	531.479
	5G	35.481	37.660
	Total	387.938	185.509
Lost Packets (%)	WiFi	0.08	0.04
	5G	0.14	0.25
	Total	0.22	0.29
Lost Bits (%)	WiFi	0.35	0.19
	5G	0.29	0.33
	Total	0.65	0.53
Late Frame Arrivals (WiFi) (%)	Zone 1	74.44	56.62
	Zone 2	94.54	0.00
	Zone 3	92.28	99.20
	Zone 4	0.04	0.18
Late Frame Arrivals (5G) (%)	Zone 1	37.37	44.34
	Zone 2	34.42	50.49
	Zone 3	32.50	0.00
	Zone 4	45.21	0.00
File Size (%)	-	94.79	99.29

6.5 Minimizing Packet Loss and Late Arrivals

In this final set of tests, the focus is placed on prioritizing timely arrivals and minimizing packet loss. These two aspects have been identified as crucial in ensuring a high-quality video stream. The objective is to guarantee that all RTP packets reach the receiver within the specified time limit, thereby ensuring improved video quality. 'Packet Loss Rate' is designed to solely consider the packet loss rates in each zone. By prioritizing zones with lower packet loss rates, the aim with this scheduler is to maximize the chances of delivering all RTP packets successfully and minimizing any loss. On the other hand, 'Timely Arrival' takes into account both the RTT and packet loss rates in each zone, with a higher priority given to RTT. The aim with this scheduler is to ensure that RTP packets not only reach the receiver but also arrive within the required time frame.

'Path Loss Rate' is able to achieve similar results to 'Zone based', while having no prior user input into path selection. These findings are visually presented in Fig. 6.11.

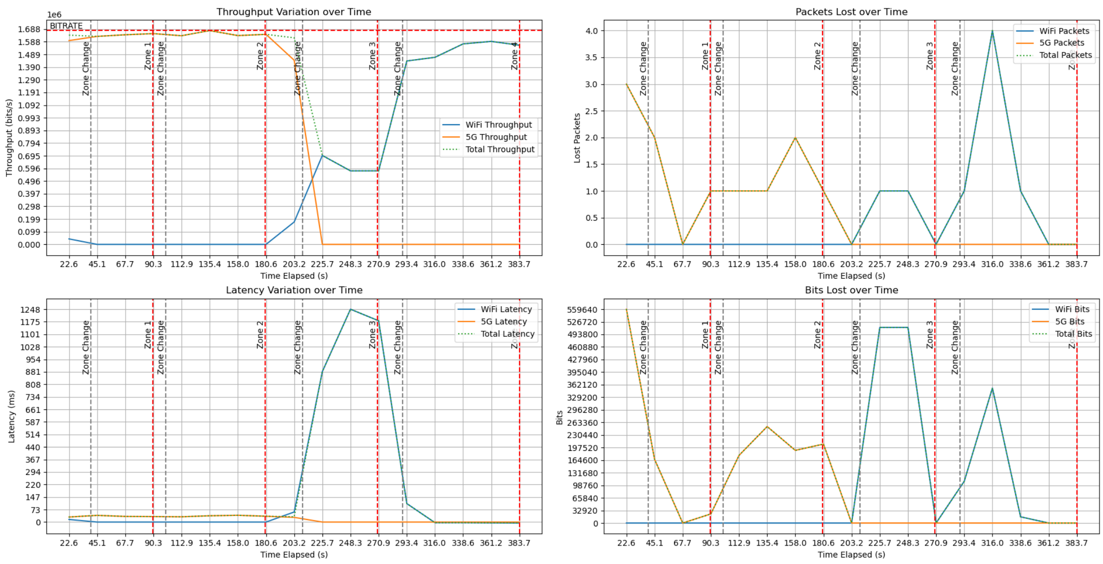

Figure 6.11: Packet Loss Rate scheduler results

The 'Timely Arrival' scheduler exhibits consistently high throughput, with a minor decrease observed during path changes. Latency remains consistently low, except for a peak observed during the path change. RTP packet loss rates are consistently low for the 5G path, with a peak observed for WiFi during path changes. These findings are visually presented in Fig. 6.12.

Table 6.5 shows that the 'Timely Arrival' scheduler demonstrates improved performance compared to the 'Packet Loss Rate' scheduler. It achieves reduced transmission time, lower latency, and improved late arrival values, while maintaining an identical file size. Lastly, Table 6.6 presents the results from all schedulers, providing a comprehensive comparison. Among the tested schedulers, it is evident that the 'Timely Arrival' scheduler is the most suitable for the system, demonstrating superior performance across multiple metrics.

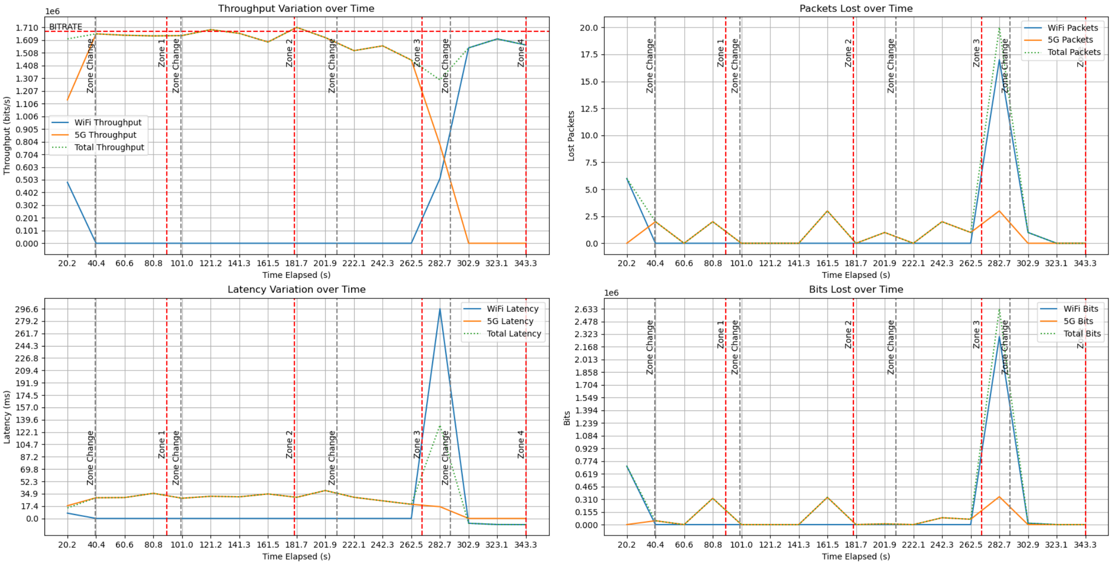

Figure 6.12: Timely Arrival scheduler results

Table 6.5: Average results of each scheduler - Minimizing Packet Loss and Late Arrivals

Scheduler	-	pl	ta
Transmission Time (s)	-	383.731	343.26
Average Throughput (bits/s)	WiFi	570,455.866	337,114.817
	5G	856,678.538	1,254,954.821
	Total	1,427,134.404	1,592,069.638
Average Latency (ms)	WiFi	613.576	219.349
	5G	33.752	29.132
	Total	155.899	36.421
Lost Packets (%)	WiFi	0.08	0.24
	5G	0.11	0.14
	Total	0.19	0.38
Lost Bits (%)	WiFi	0.27	0.55
	5G	0.29	0.22
	Total	0.56	0.77
Late Frame Arrivals (WiFi) (%)	Zone 1	54.03	33.01
	Zone 2	0.00	0.00
	Zone 3	99.71	0.00
	Zone 4	0.78	0.06
Late Frame Arrivals (5G) (%)	Zone 1	35.30	28.09
	Zone 2	41.53	29.07
	Zone 3	0.00	16.30
	Zone 4	0.00	0.00
File Size (%)	-	99.17	99.14

Scheduler	-	rd	rr	fg	wf	iw	i5	rand	rrg	rrs	zb	pl	ta
Transmission Time (s)	-	466.96	391.045	342.242	510.486	461.697	388.604	452.309	445.159	444.407	384.729	383.731	343.26
Average Throughput (bits/s)	WiFi	1,168,814.237	388,225.703	-	1,068,869.806	977,107.809	242,078.105	617521.703	600307.036	610137.536	694580.091	570455.866	337114.817
	5G	1,172,164.228	1,014,497.728	1,599,684.177	-	204,123.189	1,172,291.155	592168.221	615176.510	614340.746	729340.809	856678.538	1254954.821
	Total	1,170,489.233*	1,402,723.431	1,599,684.177	1,068,869.806	1,181,230.998	1,414,369.260	1,209,689.924	1,215,483.546	1,224,478.282	1,423,920.900	1,427,134.404	1,592,069.638
Average Latency (ms)	WiFi	612.037	997.502	-	339.627	542.324	728.785	1112.064	619.430	882.540	531.479	613.576	219.349
	5G	32.744	34.685	42.653	-	35.722	572.840	35.481	47.151	47.933	37.660	33.752	29.132
	Total	204.791	433.175	42.653	339.627	150.481	686.476	387.938	290.700	357.276	185.509	155.899	36.421
Lost Packets (%)	WiFi	0.14	0.10	-	0.16	0.17	0.10	0.08	0.54	0.18	0.04	0.08	0.24
	5G	0.07	0.03	0.34	-	0.00	0.05	0.14	0.12	0.30	0.25	0.11	0.14
	Total	0.21	0.13	0.34	0.16	0.17	0.15	0.22	0.66	0.49	0.29	0.19	0.38
Lost Bits (%)	WiFi	0.45	0.30	-	0.92	0.97	0.03	0.35	1.55	0.62	0.19	0.27	0.55
	5G	0.31	0.10	0.59	-	0.00	0.17	0.29	0.20	0.57	0.33	0.29	0.22
	Total	0.75	0.40	0.59	0.92	0.97	0.20	0.65	1.75	1.19	0.53	0.56	0.77
Late Frame Arrivals (WiFi) (%)	Zone 1	45.06	63.49	-	74.98	84.75	60.54	74.44	72.33	69.79	56.62	54.03	33.01
	Zone 2	99.47	99.47	-	99.29	96.81	99.81	94.54	70.77	78.97	0.00	0.00	0.00
	Zone 3	84.34	61.64	-	100.00	99.28	54.70	92.28	88.30	91.54	99.20	99.71	0.00
	Zone 4	0.00	0.14	-	0.05	0.00	0.10	0.04	0.05	0.00	0.18	0.78	0.06
Late Frame Arrivals (5G) (%)	Zone 1	39.27	64.28	62.11	-	30.45	99.20	37.37	65.43	53.25	44.34	35.30	28.09
	Zone 2	46.46	37.85	48.76	-	10.70	89.22	34.42	80.15	70.72	50.49	41.53	29.07
	Zone 3	26.69	30.83	66.95	-	35.14	97.83	32.50	47.54	54.97	0.00	0.00	16.30
	Zone 4	36.39	31.50	88.51	-	63.34	99.76	45.21	72.96	72.99	0.00	0.00	0.00
File Size (%)	-	100.00	96.62	98.88	98.50	94.37	96.76	94.79	94.87	95.08	99.29	99.17	99.14

Table 6.6: Average results for all schedulers

Chapter 7

Final Remarks

7.1 Conclusions

Based on the results and analysis presented in this dissertation, it can be concluded that the goal of designing a method for uplink distribution of video traffic between multiple access networks has been achieved. The utilization of various schedulers and the evaluation of different network paths and parameters have provided valuable insights into the behavior and performance of video distribution in this specific use case.

The findings demonstrate that efficient video distribution is possible by leveraging computation at the edge and making informed scheduling decisions based on network conditions. The results highlight the importance of considering factors such as latency, packet loss, throughput, frame type, and path switching in order to improve video transmission and enhance application-level latency.

By addressing the specific challenges and limitations of the multipath network environment, this dissertation contributes to the body of knowledge in the field of uplink video distribution and provides a foundation for further research and development in this area.

7.2 Contributions

This work provides several valuable contributions to the field, which are outlined below:

- **Exploration of Multipath Transport Protocols**: While existing research has primarily focused on MP-TCP, this work expands the scope by investigating another protocol. The results obtained shed light on the effectiveness and suitability of multipath RTP applications.
- **Advancement in Multipath Video Transmission**: The research on the usage of multi-path transport for video transmission is limited. This work addresses this gap by providing valuable insights and results for uplink video distribution using multipath transport. The

findings contribute to the understanding of the benefits and challenges associated with multipath video transmission. Additionally, with the emergence of 5G technology, its potential in multipath transport remains relatively unexplored. This work investigates the utilization of 5G within multipath transport, offering insights into its performance, capabilities, and limitations in the context of multipath distribution.

- **Design of Multiple Multipath Schedulers**: Throughout this work, several multipath schedulers were developed and implemented, allowing for a comprehensive evaluation and comparison of their performance. These schedulers contribute to the existing body of knowledge by offering new approaches and strategies for optimizing multipath video transmission.

- **Full-Stack Integration Results**: The research includes the evaluation of full-stack integration for this specific use case. By demonstrating the feasibility and operational aspects, this work provides valuable knowledge on how it can be supported and identifies potential technological limitations. The findings contribute to the advancement of practical implementation and understanding of the possibilities and constraints of current technologies. This contribution was achieved by:

    1. Deploying a functional experimental environment for multipath video transmission
    2. Defining a method for result analysis and interpretation
    3. Designing and implementing several schedulers to evaluate their performance and effectiveness

7.3 Future Work

While this work has made progress, there are several limitations that remained unaddressed due to time constraints. Future work can focus on overcoming these limitations and further improving the results. The following list outlines these limitations and includes suggestions on how to address them:

- **Fix RTCP Sender and Receiver Reports**: The RTCP reports encountered issues during the experimentation, and their proper functioning was not achieved. While the MA-RTP protocol is capable of retrieving historic data from the network performance monitoring station to make better informed scheduling decisions, real-time feedback from the RTCP reports would enable more accurate and adaptive scheduling decisions, especially in response to unpredictable changes in the network conditions.

- **Implement Missing Features**: The current implementation of MA-RTP lacks certain important features. Specifically, the reconstruction of video frames only occurs after the transmission process is complete, and so late frames not being properly handled. To address this limitation, future work should focus on implementing mechanisms to reconstruct video frames in real-time and handle late frames. Additionally, incorporating error correction mechanisms to replace lost frames would help ensure the integrity and quality of the transmitted video.

- **Fix WiFi Connection**: The WiFi path experienced high latency variation, particularly in the farther zones. This suggests that the RP's hardware may struggle to maintain a stable and reliable connection at such distances or signal strengths. Future work should explore options such as using a suitable external antenna with high gain or considering alternative sender devices to improve the stability of the WiFi connection in various conditions.

- **Introduce External Load**: The current setup provided sufficient bandwidth, resulting in the network not being fully utilized. To simulate more realistic and diverse scenarios where network usage plays a significant role, future work can introduce external load. This would enable the evaluation of the system's performance under more challenging conditions and provide a deeper understanding of its capabilities.

- **Get Stable 5G Network Usage**: The inconsistency between the WiFi and 5G paths, with the 5G path being shared with other devices, can lead to unpredictable fluctuations in the network performance. Future work should focus on finding ways to control and stabilize the 5G network usage. This would ensure consistent and reliable performance evaluation of both paths, enabling more accurate comparisons and analysis.

- **Populate the Network Performance Monitoring Station**: The network performance monitoring station had limited data available, which posed challenges when making scheduling decisions. Having access to a richer database within the network performance monitoring station would enhance the ability to make better informed scheduling decisions, ultimately improving the overall performance of the multipath communication system.

- **New Schedulers**: Expanding the range of schedulers used in the experiments can provide valuable insights and potentially improve the results. The current study focused on a specific set of schedulers, but there are other parameters and factors that could be considered for scheduling decisions. Future work should explore and implement a more comprehensive list of schedulers, taking into account additional parameters. By evaluating the performance of these new schedulers, a better understanding can be gained regarding their effectiveness in improving video transmission over multipath networks.

References

[1] J. Neves, "Empirical evaluation of 5G and 802.11p for supporting Cooperative Perception Services," M.S. thesis, FEUP, UP, Porto, Portugal, 2021. [Online]. Available: https://hdl.handle.net/10216/135554.

[2] M. Amend, A. Brunstrom, A. Kassler, V. Rakocevic, V. Johnson, "DCCP Extensions for Multipath Operation with Multiple Addresses," Internet Engineering Task Force, 2022. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-tsvwg-multipath-dccp/06/.

[3] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," Internet Engineering Task Force, 2022. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6824.

[4] V. Singh, T. Karkkainen, J. Ott, S. Ahsan, "Multipath RTP (MPRTP) draft-singh-avt-mprtp-00," Internet Engineering Task Force, 2010. [Online]. Available: https://datatracker.ietf.org/doc/id/draft-singh-avt-mprtp-00.html#rfc.authors.

[5] B. Y. L. Kimura, D. C. S. F. Lima, A. A. F. Loureiro, "Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities," *IEEE SYSTEMS JOURNAL*, vol. 15, no. 1, Mar. 2021, doi: 10.1109/JSYST.2020.2965471.

[6] P. Mach, Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628-1656, Mar. 2017, doi: 10.1109/COMST.2017.2682318.

[7] N. Sprecher *et al.*, "Harmonizing standards for edge computing - A synergized architecture leveraging ETSI ISG MEC and 3GPP specifications," ETSI, Jul 2020. [Online]. Available: https://www.etsi.org/images/files/ETSIWhitePapers/ETSI_wp36_Harmonizing-standards-for-edge-computing.pdf.

[8] D. Sabella *et al.*, "Toward fully connected vehicles: Edge computing for advanced automotive communications," 5GAA, Dec 2017. [Online]. Available: https://5gaa.org/wp-content/uploads/2017/12/5GAA_T-170219-whitepaper-EdgeComputing_5GAA.pdf.

[9] Transmission Control Protocol (TCP), RFC9293, Internet Engineering Task Force, 2022. [Online]. Available: https://www.ietf.org/rfc/rfc9293.html.

[10] Datagram Congestion Control Protocol (DCCP), RFC4340, Internet Engineering Task Force, 2006. [Online]. Available: https://www.rfc-editor.org/rfc/rfc4340.

[11] RTP: A Transport Protocol for Real-Time Applications, RFC3550, Internet Engineering Task Force, 2003. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3550.

[12] F. Fontes, R. Calé, M. Freitas, M. Araújo, J. Silva, D. Martins, B. Fonseca, P. Santos, A. Aguiar, A. Azevedo, F. Alves, "5G/SDN Intelligent Systems For Low latency V2X Communications in cross-Domain mobility applications," Altice Labs, CapGemini Engineering, Instituto de Telecomunicações, CISTER, VORTEX, Deliverable FLOYD_D3.2 of Project FLOYD, Dez. 2022. Confidential.

[13] S. Nagayama, D. Cavendish, D. Nobayashi, T. Ikenaga, "Path Switching Schedulers for MPTCP Streaming Video," *2019 IEEE PACRIM*, Victoria, BC, Canada, Aug. 2019, pp: 1-6, doi: 10.1109/PACRIM47961.2019.8985114.

[14] H. Wu, G. Caso, S. Ferlin, Ö. Alay, A. Brunstrom, "Multipath Scheduling for 5G Networks: Evaluation and Outlook," *IEEE Communications Magazine*, vol. 59, no. 4, pp. 44-50, Apr. 2021, doi: 10.1109/MCOM.001.2000881.

[15] S. Ferlin, Ö. Alay, O. Mehani, R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, Vienna, Austria, Jun. 2016, pp: 431-439, doi: 10.1109/IFIPNetworking.2016.7497206.

[16] Y. Lim, E. Nahum, D. Towsley, R. Gibbens, "ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths," *CoNEXT '17: Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, Nov. 2017, pp: 147–159, doi: 10.1145/3143361.3143376.

[17] H. Wu, Ö. Alay, A. Brunstrom, S. Ferlin, G. Caso, "Peekaboo: Learning-Based Multipath Scheduling for Dynamic Heterogeneous Environments," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2295-2310, Jun. 2020, doi: 10.1109/JSAC.2020.3000365.

[18] M. Vranješ, S. Rimac-Drlje, K. Grgić, "Review of objective video quality metrics and performance comparison using different databases," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 1-19, Jan. 2013, doi: http://dx.doi.org/10.1016/j.image.2012.10.003.

[19] C. Paasch, G. Detal, S. Barré, F. Duchêne, O. Bonaventure. "The fastest TCP connection with Multipath TCP." multipath-tcp.org. http://multipath-tcp.org/pmwiki.php/Main/50Gbps (accessed Jan. 1, 2023).

[20] C. Paasch, R. Khalili, O. Bonaventure, "On the Benefits of Applying Experimental Design to Improve Multipath TCP," *CoNEXT '13: Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, Dec. 2013, pp: 393-398, doi: 10.1145/2535372.2535403.

[21] A. Azevedo, "Edge computing for VRU protection: empirical evaluation of 5G and 802.11 communication technologies," M.S. thesis, FEUP, UP, Porto, Portugal, 2021. [Online]. Available: https://hdl.handle.net/10216/135424.

[22] Infovista. "TEMS Pocket - Portable Mobile Network Testing | Infovista." https://www.infovista.com/tems/pocket.

[23] Infovista. “TEMS Investigation - Mobile Network Drive Testing | Infovista.” https://www.infovista.com/tems/investigation.

[24] O. Bonaventure, M. Handley, C. Raiciu, "An Overview of Multipath TCP," *;login:*, vol. 37, no. 5, May. 2012. [Online]. Available: https://www.usenix.org/system/files/login/articles/login1210_bonaventure.pdf.

[25] C. Paasch, O. Bonaventure, "Multipath TCP," *acmqueue*, vol. 12, no. 2, Mar. 2014. [Online]. Available: https://queue.acm.org/detail.cfm?id=2591369.

[26] X. Jiang, F. Yu, T. Song,V. Leung, "A Survey on Multi-Access Edge Computing Applied to Video Streaming: Some Research Issues and Challenges," *IEEE Communications Surveys Tutorials*, vol. 23, no. 2, pp. 871-903, Mar. 2021, doi: 10.1109/COMST.2021.3065237.

[27] Y. Mao, C. You, J. Zhang, K. Huang, K. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322-2358, Aug. 2017, doi: 10.1109/COMST.2017.2745201.

[28] B. Kimura, D. Lima, A. Loureiro, "Alternative Scheduling Decisions for Multipath TCP," *IEEE Communications Letters*, vol. 21, no. 11, Nov. 2017, doi: 10.1109/LCOMM.2017.2740918.

[29] V. Singh, S. Ahsan, J. Ott, "MPRTP: multipath considerations for real-time media," *MMSys '13: Proceedings of the 4th ACM Multimedia Systems Conference*, Feb. 2013, pp: 190-201, doi: 10.1145/2483977.2484002.

[30] F. Yang, P. Amer, N. Ekiz, "A Scheduler for Multipath TCP," *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, Oct. 2013, pp: 1-7, doi: 10.1109/ICCCN.2013.6614091.

[31] T. Li, L. Li, X. Wang, X. Zhang, F. Zhang, K. Wan, "An In-depth Analysis of Subflow Degradation for Multi-path TCP on High Speed Rails," *Conference: 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Jun. 2022, pp: 231-240, doi: 10.1109/WoWMoM54355.2022.00042.

[32] C. Paasch, S. Barre, et al. "Multipath TCP in the Linux Kernel." multipath-tcp.org. http://multipath-tcp.org/ (accessed Jan. 3, 2023).

[33] C. Paasch, S. Ferlin, O. Alay, O. Bonaventure, "Experimental Evaluation of Multipath TCP Schedulers," *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, Aug. 2014, pp: 27–32, doi: 10.1145/2630088.2631977.