

Cross-layer Optimization of MAC Scheduling for Multi-User Virtual Reality over 5G

Zheng Du



This document was typeset by L^AT_EX.

Cover image © Artem Podrez, Pexels, 2021

Artem Podrez - Person using virtual reality goggles (<https://www.pexels.com/>)

Cross-layer Optimization of MAC Scheduling for Multi-User Virtual Reality over 5G

Author:
Zheng Du

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in Electrical Engineering

at the Delft University of Technology,
to be defended publicly on June 09, 2022 at 13:00

Responsible professor: Dr. R. Litjens MSc
Daily supervisor: Prof. dr. J.L. van den Berg

Student Number: 5098920
Project duration: March 15, 2021 – June 09, 2022

THESIS COMMITTEE

Dr. R. (Remco) Litjens MSc, TU Delft, TNO
Prof. dr. J.L. (Hans) van den Berg, University of Twente
Prof. dr.ir. F.A. (Fernando) Kuipers, TU Delft

Abstract

As a result of a global pandemic, there has been an increasing interest in tools for remote video conferencing and collaboration. One of these new innovations is social eXtended Reality (XR). By combining Virtual Reality (VR) and Augmented Reality (AR) technologies, social XR can provide a more immersive experience than any other VR application by giving users at different locations the chance to virtually gather in real-time. But such applications impose enormous requirements on computational and communication resources. 5th Generation (5G) mobile networks are targeted as solution to provide ultra-low latency and ultra-high throughput for social XR. In current research, many optimisations are aimed at VR applications such as on-demand streaming, while there is a lack of solutions for real-time user-interactive applications like social XR. In this graduation project we develop and assess cross-layer solutions for optimised scheduling of social XR applications in 5G networks. An existing framework for simulating social XR conference applications serves as the basis for our modelling approach. We devise different schedulers, that utilise cross-layer information in the form of the video frametype and frame-level End-to-End (E2E) latency budgets rather than packet-level latency budgets purely within the Radio Access Network (RAN). In contrast to previous work, we create the VR traffic based on real video data and develop tools to model the packet dispersion caused by multi-hop transmission over the internet towards the RAN. We study the effect of various system and traffic parameters on the Quality of Service (QoS) and perceived Quality of Experience (QoE) in the context of social XR applications through an extensive sensitivity analysis. Herein we also assess the performance impact of different types of cross-layer packet schedulers. Further, we gain insights into the correlation between the network QoS and perceived QoE by end users which are the key in future cross-layer implementations for social XR.

Contents

1	Introduction	6
1.1	Motivation	6
1.2	Technical Background	7
1.2.1	Video Frame Types	7
1.2.2	5G Mobile Networks	8
1.3	Objective and Approach	9
1.4	Outline	10
2	Literature Review	11
2.1	Review of Cross-Layer Scheduling	11
2.2	Contributions	12
3	System Description and Modelling	14
3.1	Overview of a Multi-User Wireless VR Application Scenario	14
3.2	Social XR Conference Model	14
3.2.1	Social XR Setting	15
3.2.2	Propagation Environment	16
3.2.3	Radio Access Network	16
3.2.4	Packet Scheduling	18
3.3	Traffic Model	20
3.3.1	Creating Packet Traces from Recorded Video	20
3.3.2	Modelling Packet Dispersion in IP Networks	23
4	Simulation Scenarios and Results	26
4.1	Simulation Scenarios and Approach	26
4.2	Results of Sensitivity Analysis	29
4.2.1	Comparison of Schedulers for Default Scenario	29
4.2.2	Impact of Scenario Parameters	31
4.3	Feasible Load and Application Configurations	41
4.3.1	Scenarios and Settings	41
4.3.2	Results and Observations	41
4.4	Analysing Quality of Experience	44
4.4.1	Interpreting PSNR and SSIM Values	44
4.4.2	Analysing PSNR and SSIM Results	44
5	Conclusion	48
5.1	Conclusion	48
5.2	Future Work	50

A	BLER Curves	55
B	Choosing Number of Hops with Traceroute	56
C	Tuning Bandwidth and RAN Budget	59
C.1	Tuning the Default Carrier Bandwidth	59
C.2	Tuning the Default RAN Latency Budget for Different E2E Requirements	59
D	Video Quality Results (PSNR/SSIM) of Sensitivity Analysis	61

List of Abbreviations

3GPP	3rd Generation Partnership Project
4G	4th Generation
5G	5th Generation
AR	Augmented Reality
AVC	Advanced Video Coding (H.264)
B-frame	Bidirectional Predicted Frame
BLER	Block Error Rate
BS	Base Station
CQI	Channel Quality Indicator
CSI	Channel State Information
CSV	comma-separated values
DL	Downlink
DPI	Deep Packet Inspection
E2E	End-to-End
EDD	Earliest-Due-Date
EXP/PF	Exponential/Proportional Fair
FIFO	First-In-First-Out
F-OFDM	Flexible Orthogonal Frequency Division Multiplexing
FOV	Field Of View
GoP	Group of Pictures
HEVC	High Efficiency Video Coding (H.265)
HMD	Head Mounted Display
HOL	Head Of Line
I-frame	Intra-coded Frame
IoT	Internet of Things
IP	Internet Protocol

LoS	Line-of-Sight
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MCU	Multi-point Control Unit
MI-ESM	Mutual Information Effective SINR Mapping
MIMO	Multiple-Input Multiple-Output
M-LWDF	Maximum-Largest Weighted Delay First
mMTC	massive Machine-Type Communications
mmWave	Millimetre Wave
MSE	Mean Squared Error
NALU	Network Abstraction Layer Unit
PDR	Packet Drop Rate
PF	Proportional Fair
P-frame	Predicted Frame
PRB	Physical Resource Block
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
QuaDRiGa	QUAsi Deterministic RadIo channel GenerAtor
RAN	Radio Access Network
RTP	Real-time Transport Protocol
SINR	Signal-to-Interference-plus-Noise Ratio
SSIM	Structural Similarity Index Measure
TB	Transport Block
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
URLLC	Ultra-Reliable Low-Latency Communications
VR	Virtual Reality
XR	eXtended Reality

Chapter 1

Introduction

1.1 Motivation

With the evolution and evermore deployment of the 5G network in the past years, more and more applications have emerged that can profit from the new mobile communication standard. Besides bringing higher bandwidths and data rates, 5G also enables novel applications that require high reliability and ultra low latencies such as autonomous driving, and makes technologies like Internet of Things (IoT) and massive Machine-Type Communications (mMTC) possible, where high numbers of devices are connected.

One type of application that can highly benefit from the new supply of extreme high data rates are VR-based applications. With the global pandemic happening in the past few years, people have been forced to come up with new ways to communicate and collaborate remotely. One novel solution for this is social XR. It leverages XR technologies, which includes AR and VR, to create immersive virtual experiences.

To facilitate such a high-quality and immersive experience, many stringent requirements have to be fulfilled, in particular very low latency and very high throughputs. Further, unlike more traditional VR or AR applications, the real-time aspect and user-interactive nature of social XR impose different demands than for example for VR gaming or on-demand streaming of VR content. Novel cross-layer solutions, where information is shared among layers for more efficient use of available resources and the ability for high adaptivity are the key. Clemm et al. [1] emphasises a need for new cross-layer solutions to deal with real-time user interactivity as a new challenge, which limits the utilisation of caching/buffering [2] or movement prediction techniques applied in traditional VR [3].

With current Head Mounted Displays (HMDs) having higher resolutions, higher throughputs may be needed - up to 100 Mbps for a compressed stream and potentially up to 500 Mbps for a raw (uncompressed) stream [4]. The added real-time interactivity limits the possibility to buffer content and imposes strict latency requirements to create an immersive experience in social XR. Ideally required limits in the order of 1 ms for the E2E latency are still impossible in the near future due to physical and technical constraints. The preference is thus to achieve a latency as low as possible to avoid motion sickness and improve immersiveness, with different targets given in literature like 50ms [5] and 30ms [6].

For more traditional, non-interactive applications, plenty of strategies exist to optimise VR delivery [7], with many solutions focused specifically on improving the efficiency of video processing in the application layer which in turn benefits the latency and bandwidth usage. One technique for example is tiling, where the video is divided into multiple separately stored and processed sections.

By focusing the available resources only on what is inside the Field Of View (FOV) of a user, it results in savings of bandwidth and computation power [8]. Related to it is using prediction strategies for the user’s head or eye movement while streaming VR content. With this method good radio channel conditions can be exploited in the application layer to choose the predicted parts of a video in a cross-layer fashion and transmit the corresponding tiles with higher resolution [9]. By using edge computing the video processing can be brought closer to the user from the source in the network cloud when streaming VR. Alternatively, if computations were previously done on the user device, it can be offloaded to the edge depending on the edge load conditions and user throughput, since raw (and hence bigger) data needs to be transmitted from the device to the 5G Base Station (BS) [2, 10]. Other possibilities with edge servers, most noticeably caching content in advance to reduce the transmission latency, may prove to be difficult for social XR given the real-time nature of the interaction.

One cross-layer optimisation in form of a multi-user Medium Access Control (MAC) scheduling scheme for VR services in 5th Generation (5G) is proposed in [11]. It exploits video frame type information from the application layer encapsulated in video packets to prioritise the ones more crucial for maintaining a good video quality. To implement such a cross-layer scheduling scheme for social XR in practice, the only premise is the functionality to extract the application-level frame type information at the packet scheduler in the BS which is covered in the following section.

For streaming services, schedulers in base stations are usually latency-aware and take into account the delay packets experience in the RAN [12]. The RAN delay is only a small part of the total E2E latency budget of a frame, which is a QoE requirement set by the application. The E2E latency experienced by individual packets comprises multiple parts such as latency due to video processing and networking latency. It has been shown that considering the E2E deadlines in the packet scheduling can improve the resulting video quality [13]. Hence, in this thesis, a cross-layer solution in the context of social XR applications will be modelled and assessed. We take an existing framework for simulating social VR conference applications [14] as a basis for our modelling approach and devise different schedulers, that utilise the video frame type and the remaining E2E latency budget of video packets as cross-layer information.

In the following parts of this chapter, we provide a high-level overview of the video frametypes and the 5G RAN, which are the key components of this study. Then, we state the goals of the thesis and the research approach followed to develop and assess the targeted cross-layer solutions for the enhancement of social XR performance. Lastly, we give the structure for the rest of this thesis.

1.2 Technical Background

In this section we provide a brief overview of the main technical aspects that are featured in the thesis. First, we explain the functionality of video frame types in the context of video processing and the mechanism of exploiting this application layer information in the scheduling at the BS. Then, we present the key novel features of 5G, the transmission technology used in this study.

1.2.1 Video Frame Types

Video frame types are not only used for social XR but for video applications in general. There are three different types of frames used in video compression, the Intra-coded Frame (I-frame), the Predicted Frame (P-frame), and the Bidirectional Predicted Frame (B-frame) [15]. The reason to take advantage of frame type information stems from their distinct characteristics. An I-frame contains a complete image, and is the least compressible out of the three, but also the only frame

where the decoder does not require other video frames to decode it. A P-frame contains only the changes in the image relative to that represented by the previous frame. This lets the encoder reduce the transmission payload by not needing to store unchanged pixels in the frame. The other optional B-frame uses differences from both the previous and following frames in the encoding, making them the most compressible type of frame.

A Group of Pictures (GoP) then contains a single I-frame followed by multiple P- and optionally B-frames arranged in a specified order. This means for example that decoding errors due to packet loss in an I-frame may lead to error propagation and potentially quality degradation through the whole GoP [16].

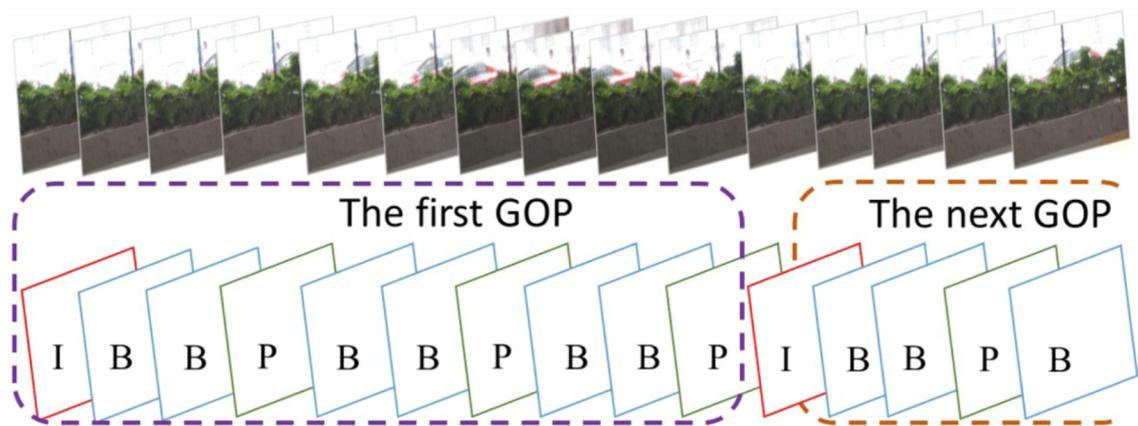


Figure 1.1: Example GoP structure of size 10 [17].

As our scope is to develop solutions for social XR applications over 5G networks, the findings in [11] raise an incentive to prioritise I-frames in the social XR traffic handling and optimise the packet scheduler at the 5G BS. Traditionally, the BS would not be able to read Internet Protocol (IP) packets' payload content, and thus is not able to know information such as the video frame type for individual packets. A solution for this is using a Deep Packet Inspection (DPI) function directly at the BS or another related node, like the gateway to monitor the VR packet stream and read out the frame type information from every IP packet.

In the video formats Advanced Video Coding (H.264) (AVC) or its successor High Efficiency Video Coding (H.265) (HEVC) that are the most commonly used in streaming [18], the encoded data of a video frame is organised into so-called Network Abstraction Layer Units (NALUs) [19]. These units make up the payload of the Real-time Transport Protocol (RTP), the application-layer protocol used for real-time streaming applications for delivery over the IP network. Typically, it is used in conjunction with the User Datagram Protocol (UDP), since it is more suited as transport-layer protocol for latency-sensitive applications than Transmission Control Protocol (TCP) [1]. For routing across the internet, the data is then encapsulated in the IP payload. Figure 1.2 shows how these NAL units are encapsulated in the different protocols. The DPI function would then for every packet remove the headers to extract the bytes belonging to the video frame and read out a 5-bits type field in the NALU header to determine the frame type [11].

1.2.2 5G Mobile Networks

In the following, we present several key technologies in 5G, which allow for far higher throughputs and lower latencies compared to its predecessor, the 4th Generation (4G) standard, and enable demanding applications such as social XR.

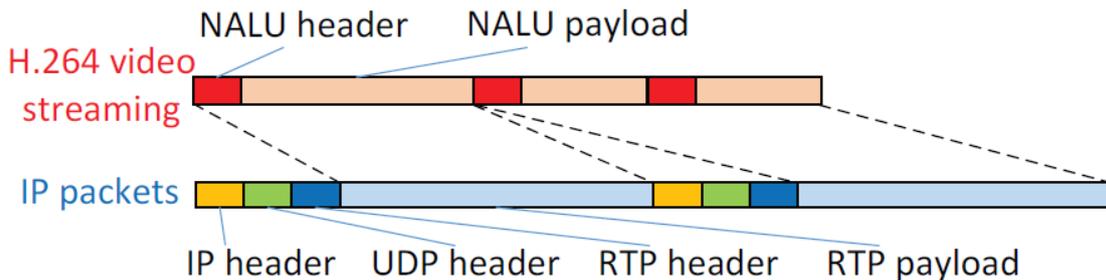


Figure 1.2: Protocol stack structure of VR video streaming [11].

One upgrade in 5G consists of using frequencies in the range of 24-300 GHz, which is commonly referred to as the Millimetre Wave (mmWave) spectrum, which allows for higher bit rates and cell capacity. Further, 5G employs Flexible Orthogonal Frequency Division Multiplexing (F-OFDM) that enables the network operator to adapt the numerology depending on factors like the service requirements of the application or the propagation aspects of the cell environment. By configuring a wider subcarrier spacing and hence shorter symbol duration in the fraction of a millisecond, shorter Round-Trip Times (RTTs) of a few milliseconds are now achievable. With higher frequency bands, 5G depends upon massive MIMO and beamforming to increase the spectral efficiency. Massive MIMO beamforming is necessary to provide sufficient coverage at higher frequencies and to enhance the Signal-to-Interference-plus-Noise Ratio (SINR) for enabling higher data rates. Driven by Ultra-Reliable Low-Latency Communications (URLLC)-type applications, it is imperative for packet schedulers in 5G to incorporate a latency component. This directly relates to the main aspect explored in our thesis, that is optimising the scheduling functionality by exploiting cross-layer information.

1.3 Objective and Approach

The main goal of this thesis is to propose and assess various cross-layer optimisations of MAC scheduling of social XR traffic in a 5G network. More specifically, we aim to assess the benefits of utilising cross-layer information of the VR application for the packet scheduling in the RAN by exploiting knowledge about E2E frame latency budgets and the video frame type incoming IP packets belong to. We investigate the potential benefits compared to the latency-aware non-cross-layer EDD [20] and M-LWDF [12] packet schedulers in terms of both network QoS and the perceived QoE.

For our investigations we focus solely on the effects of the cross-layer scheduler. To this end, we modify an existing framework for simulating indoor social XR conferences [14]. The framework comprises modelling aspects for the RAN and its physical configurations and resource management mechanisms, the user behaviour, the propagation environment and physical setting. We use real video data to create a suited VR traffic stream. Further, we model the effects of packet dispersion caused by video processing and multiple internet hops the IP packets are routed through and the impact this has on packet latencies and the variations therein before they arrive at the 5G BS. This way we obtain more realistic traffic characteristics to incorporate into the social XR conference simulator.

Using simulations in an extensive sensitivity analysis we intend to obtain insights into the impact of various system and scenario parameters, such as the video application bit rate or the E2E latency budget, on the attainable performance and capacity gains from our cross-layer optimisation

approach. We assess the results from a network perspective with the Packet Drop Rate (PDR) as a QoS measurement. Using a concrete reference video, we can also evaluate the impact of the presented cross-layer scheduling solutions on the user-perceived video quality (QoE) using the Peak Signal-to-Noise Ratio (PSNR) [21] and Structural Similarity Index Measure (SSIM) [22]. This way, we can determine the differences between assessing a system based purely on QoS or QoE and recommend cross-layer solutions for future implementations in practice.

1.4 Outline

The remainder of this thesis is organised as follows. In Chapter 2 we give a literature review for research in cross-layer optimisations of video and VR streaming that utilise the E2E latency budget as well as the video frame type. In addition we list our contributions beyond the state of the art research for social XR and cross-layer optimisations in particular.

Chapter 3 first presents a high-level description of the addressed system as a whole and the underlying components for the VR delivery pipeline. Afterwards, we address each of the modelling aspects and assumptions in more detail, such as the generation of the traffic stream, the user behaviour model, the social XR application model, the physical configurations of the RAN, as well as the (non-)cross-layer packet schedulers.

Next, we present the results of our simulations in Chapter 4. We start off by defining the different scenarios we will take into account. We discuss the results with respect to the benefits of utilising various levels of cross-layer information in the packet scheduling. We assess the simulation outcomes both from a network QoS perspective and from a QoE perspective utilising known video quality metrics to get a better understanding of the video quality as perceived by the user.

Lastly, in Chapter 5 we formulate our conclusions by summarising the most important findings and give pointers for potential research in the future.

Chapter 2

Literature Review

2.1 Review of Cross-Layer Scheduling

In this section, we discuss and provide an overview of some studies addressing the subject of cross-layer scheduling for video transmission in wireless networks. We first review studies that utilise the E2E latency budget of packets as cross-layer information to optimise the packet scheduling. Then, we show approaches that exploit information about the video frame type to prioritise certain packets and enhance the user experience.

E2E Latency-Based Cross-Layer Scheduling

First, we discuss work related to exploiting information about E2E delay requirements in video streaming applications to optimise the QoE. In [23] and [24], the E2E latency budget of video packets is utilised as cross-layer information in the MAC-layer. However, the approaches here are linked to optimising the packet dropping mechanism, whereas our focus lies in the packet scheduling. In [25], Xie et al. propose an approach to optimise the routing of video packets by utilising knowledge about the packets' E2E latency requirements. Since the context here deviates too much from our thesis, it will not be discussed in more detail.

In [26], the resource allocation problem for video transmission over mesh networks with multiple users is addressed. In the proposed solution both the packet scheduling and buffer management mechanism consider the E2E latency deadlines of video packets to optimise the bandwidth allocation among all end users. The results show that this way video distortions can be minimised and the video qualities among all users balanced. It is important to note that the considered setting of video streaming over multihop networks differs from our social XR use case. Further, in [26] all users have different E2E latency requirements varying between 0.1 and 1 s, unlike in this thesis.

A packet scheduling algorithm based on E2E delay requirements is presented in [13]. The authors consider delay sensitive video streaming services over mobile ad hoc networks. As such, the proposed algorithm is applied at every network node and takes into account the accumulated delay per hop and the maximum tolerable E2E delay to optimise the scheduling of packets across the network. It is shown that by considering the total E2E delay, the total E2E delay of each packet is kept in a tolerable range as much as possible, which greatly improves the video quality. Note, that the presented scenario of a mobile ad hoc networks is different than cellular mobile networks considered in this thesis.

Frametype-Based Cross-Layer Scheduling

There exists a number of studies about leveraging application level information like the frame type to optimise the QoE for video streaming over wireless networks. While in [27,28], packet scheduling optimisations using cross-layer information about the video is used, these studies are not recent. In [29,30], cross-layer optimisations using video frame type information for improving the video QoE are proposed. However, in both cases it involves the packet fragmentation process rather than the packet scheduling, which is the focus in this thesis. As such, these studies will not be further discussed in detail.

In [31], cross-layer solution for enhancing the QoS for medical video transmission in wireless networks is presented. In particular, the authors propose a packet differentiation and prioritisation method based on characteristics like the type of medical video transmitted in the flow, the various types of layers in layered encoded streams, as well as the video frame types mentioned in Section 1.2.1. In the proposed method packets are marked based on the different priority levels, which are then considered in the packet scheduling and dropping mechanisms. Results show that this cross-layer approach is effective in enhancing the QoS and QoE. However, it needs to be noted that compared to social XR medical video have lower resolutions and thus require less throughput. In addition, the scenarios in [31] allowed up to 300 ms E2E latency, which is far lower than E2E latency requirements in social XR.

A cross-layer scheduling scheme for VR services in 5G is proposed in [11]. There, Zhang and Huang propose a solution which comprises the use of video frame type differentiation in latency-aware packet schedulers. To our knowledge, this is the only paper that implements the usage of video frame type information for packet schedulers in the context of multi-user VR streaming over 5G networks. The packet scheduler takes the frame type and the experienced RAN latency of the packet t_{RAN} to obtain the scheduling priority w as follows

$$w = \begin{cases} a_I + b_I t_{RAN} & \text{for an I-frame} \\ a_P + b_P t_{RAN} & \text{for a P-frame} \end{cases} \quad (2.1)$$

with t_{RAN} denoting the experienced RAN latency of a packet, as defined in [11]. Results of the proposed system show a decrease in the overall packet transfer latency especially in high-load scenarios and an increase in the user capacity in the considered scenarios.

The coefficients a_I, b_I, a_P and b_P determine the exact priority I-frame packets receive over P-frame packets. In Chapter 3.2.4, we discuss the exact modelling of the frame type prioritisation parameter for our cross-layer schedulers.

2.2 Contributions

With this thesis we aim to contribute to the state of the art research in supporting social XR over mobile networks. More specifically, our main objective is to assess the gain from using cross-layer information such as the E2E frame latency budget and video frame type in the packet scheduling at the RAN BS in a multi-user social XR scenario.

In more detail, the contributions are the following:

- Proposal of several cross-layer schedulers using various levels of cross-layer information.
- Development of tools for the performance assessment of the (non-)cross-layer schedulers by modifying existing 5G RAN simulator and building a network hop simulator to model packet dispersion due to transmission over the internet.

- Extensive performance assessment of the proposed schedulers using the PDR as network QoS metric.
- Evaluation of cross-layer schedulers in terms of QoE metrics and assessing the correlation between resulting QoS and QoE.

Chapter 3

System Description and Modelling

As we want to objectively measure and get an understanding of the gains from cross-layer scheduling, our system model needs to cover all relevant aspects of delivering social XR to multiple users in a 5G network. This chapter presents the modelling for the delivery of a social XR video stream through the internet and 5G RAN to the end users. A high-level overview of all system components and their interactions is given first. Then, each component's modelling parameters and the assumptions behind them are outlined in detail.

3.1 Overview of a Multi-User Wireless VR Application Scenario

This section gives a high-level overview for the components of the modelled social XR system as illustrated in Figure 3.1. Our focus is only on the downlink to keep the complexity reasonable since there would be additional modelling assumptions required for the video processing in the upstream direction. In our model the source of the downlink stream is the media processing cloud which comprises the Multi-point Control Unit (MCU) that receives and combines video streams from all participants in the social XR session and outputs the encoded VR streams for the UEs. We model the VR stream at the source in the media processing cloud, which combines the feed of the virtual and physical participants.

The packet stream of every generated frame exits the MCU not as a single chunk of packets, but rather in bursts with a certain dispersion due to the encoding process [32] (see Section 3.3.1), before it then enters the IP network. Depending on the routing between the MCU and the RAN, the VR packets go through various network hops, which cause further packet dispersion. Section 3.3 describes our models for the VR traffic and the packet dispersion in detail. Then, the VR packet stream arrives at the 5G BS that serves the physical participants in the social XR conference, as modelled in the next section. (Non-)cross-layer packet schedulers (Section 3.2.4) then deliver the VR packets to the end users.

3.2 Social XR Conference Model

This section contains the models and methods used to simulate a social XR session. The presented model is mostly based on the thoroughly assessed and optimised model by Morais et al. [14]. Parts such as the physical setting, user behaviour and components of the radio resource management are adopted with modifications made regarding the antennas, traffic model and packet schedulers used, as the latter is the focal point of this work.

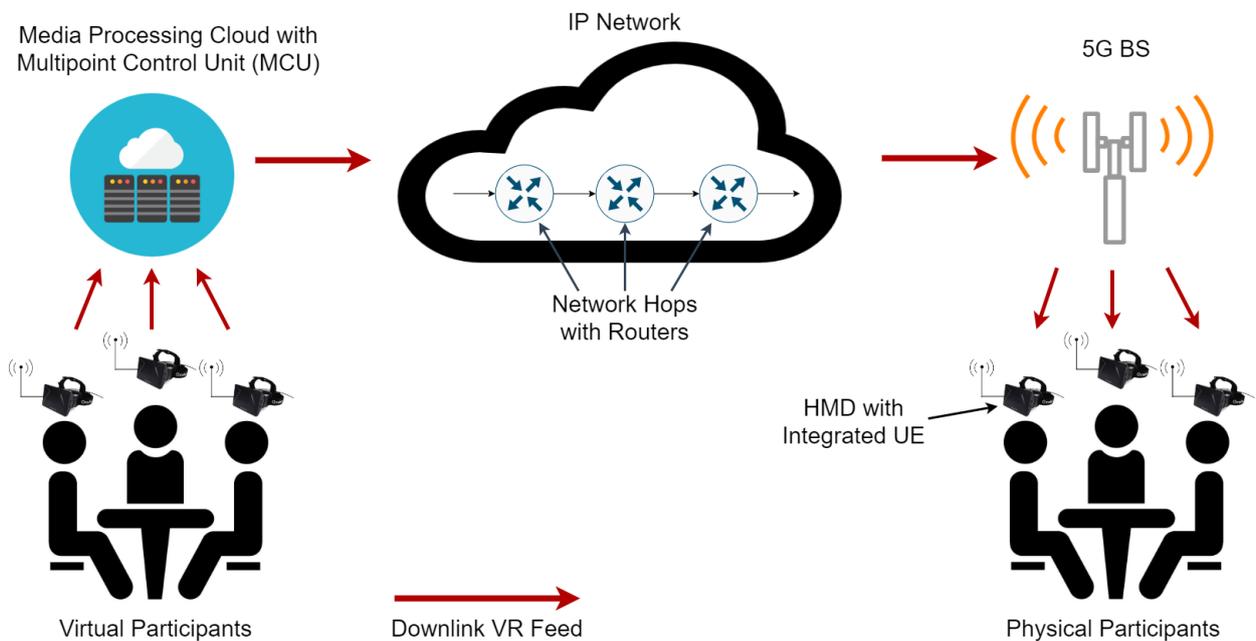


Figure 3.1: Overview of modelled social XR system.

3.2.1 Social XR Setting

We simulate the downlink portion of users participating in a social VR conference in an indoor office. Half the users participating are joining virtually and half are physically present. We consider a round table in the center of the room with all users uniformly distributed around it in an alternating seating fashion with the serving BS located on the ceiling at the center of the table as shown in Figure 3.2.

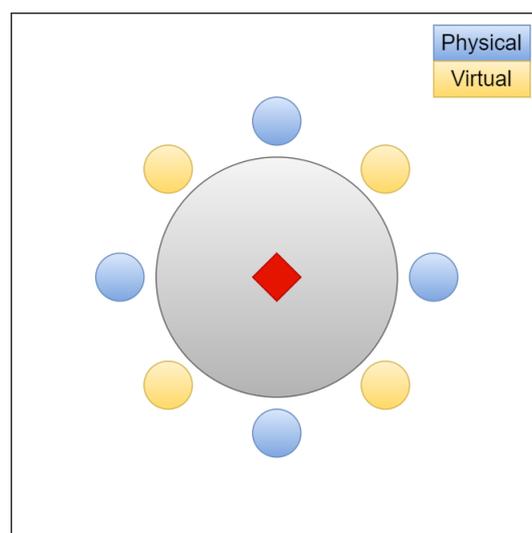


Figure 3.2: Example social XR setting with four physical and four virtual participants, with RAN BS position marked in red.

Every physical user is wearing an HMD with integrated User Equipment (UE) with the processed video stream coming from the MCU, which combines the feed from virtual and physical participants,

across the IP network to the BS and then transmitted to the UEs. A new modelling approach compared to [14] for the traffic streams based on real video data is used, which is elaborated on in Section 3.3.

The intent to model user behaviour in form of head motion is due to changes in the position and orientation of a user’s head also directly affecting the position and orientation of the UE’s receiver antennas which affects the downlink channel quality due to multipath fading. To model the head motion, the head’s position and orientation are represented as a set of Cartesian coordinates and rotational angles respectively. The target trajectory of a user’s head is then specified by a predetermined order of speakers, with a set time interval for every speaker and with all users turning their head to the speaker at every iteration. To introduce realistic head wobbling, the sampling of the offset coordinates and angles to reach the target head position and orientation are randomised according to a normal distribution.

3.2.2 Propagation Environment

To simulate our indoor office scenario that includes aspects like path loss, shadowing or multi-path fading, we opt for the QUasi Deterministic RadIo channel GenerAtor (QuaDRiGa) [33] to generate channel traces according to the indoor Line-of-Sight (LoS) office as described by 3rd Generation Partnership Project (3GPP) [5]. Each trace contains the complex impulse response between the receiver and transmitter antenna pair, taking into account path loss, our scenario specific fading and the antenna orientations. As QuaDRiGa does not take noise into account, we add that afterwards using the equation for thermal noise power $P_N = k_B T B$, with the Boltzmann constant $k_B = 1.380649 \times 10^{-23}$ J/K, temperature $T = 290$ K and frequency bandwidth B . Figure 3.3 illustrates the multi-path fading effects in the channel gain of four UEs for the considered propagation environment.

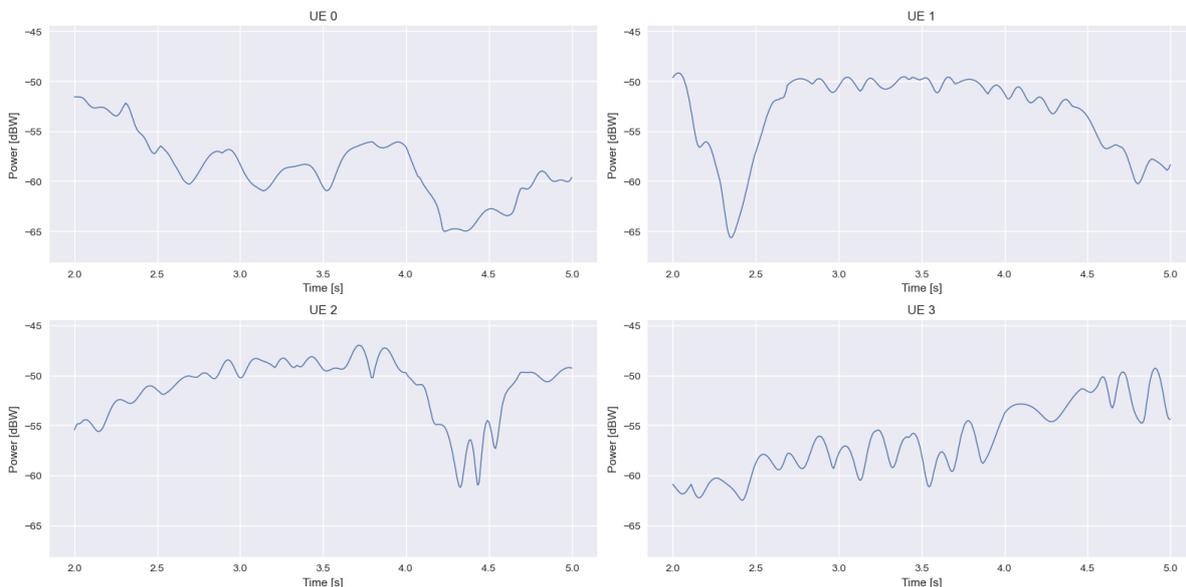


Figure 3.3: Channel gain variations over time for four users, at 3.5 GHz carrier frequency.

3.2.3 Radio Access Network

Considering that the main purpose of this study is to get insights from cross-layer scheduling, we disregard the use of beamforming as well as massive Multiple-Input Multiple-Output (MIMO) an-

tenna arrays to avoid unnecessary complexity for our assessment of cross-layer schedulers. Instead, we assume a single omni-directional antenna for the BS as well as for the UEs. The BS uses a Time Division Duplex (TDD) carrier in the 3.5 GHz Band with a ratio of 4:1 between Downlink (DL) and Uplink (UL) slots. Given 5G’s F-OFDM-based radio interface featuring a flexible numerology, we use *numerology 2*, meaning our subcarrier spacing is 60 kHz and the Transmission Time Interval (TTI) duration $T_{TTI} = 0.25 \text{ ms}$, which is the smallest time unit at which the BS schedules transmissions. To cut down the computation time and memory usage required to generate the channels and considering the assumed wideband (not frequency-selective) scheduler, we create the impulse responses for fewer Physical Resource Blocks (PRBs) than there would be in reality. As the channels do not change too significantly from PRB to PRB, it is sufficient to only generate the responses for ten PRBs across a 50 MHz frequency bandwidth, and scale them accordingly to the different frequency bandwidths used for simulating the social XR scenarios.

Simulation Procedure

As previously stated, we focus in this study on the BS scheduling and transmitting to UEs in the downlink. This section outlines the functionalities needed for every TTI, as shown in the flowchart in Figure 3.4.

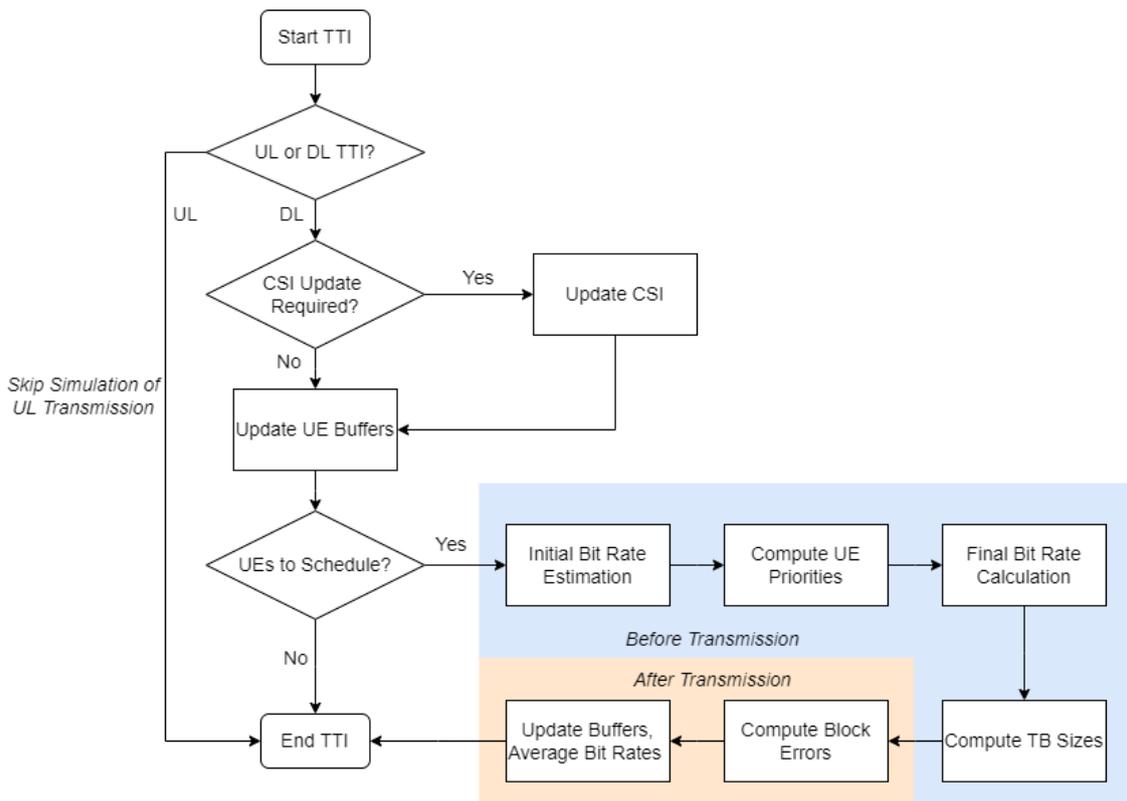


Figure 3.4: Simulation steps in every TTI

We configure the TDD carrier with four DL TTIs followed by one UL TTI, which we do not explicitly model here. To estimate the channel quality, which is then further used to, e.g. determine the attainable bit rate, a Channel State Information (CSI) update is performed every $N_{TTI}^{CSI} = 5$ TTIs. Similar to how it is done in practice, the CSI updates happen only every few TTIs as channels do not change enough to update too frequently. In our model, the CSI is acquired through the generated channel impulse responses.

If there is no CSI update necessary, a TTI begins with updating the UE buffers. The packet schedulers used are all latency-aware, hence all packets in the buffer are checked whether they need to be dropped due to exceeding their latency budget based on the configured scheduler. We go in-depth into the modelling aspects for the (non-)cross-layer packet schedulers in the following Section 3.2.4. As not all UEs will have packets in their buffer all the time, only UEs with packets to be sent are listed for scheduling.

To estimate the instantaneous bit rates for the UEs, an SINR estimation is done based on the last reported CSI. The SINR is used to determine the Channel Quality Indicator (CQI) from the Block Error Rate (BLER) curves (Appendix A). The highest CQI corresponding to the SINR is chosen, which still achieves a lower percentage of block errors than the configured BLER target of 10%, this determines the selection of the Modulation and Coding Scheme (MCS) for the transmission. The MCS defines the code rate R_c and hence the the number of bits in a symbol N_{bits}^{Symbol} . We first assume that all 168 resources elements in a PRB are used for data, i.e. $N_{Symbol}^{PRB} = 168$. The attainable bit rate R in bit/s is computed by dividing the number of bits transmitted in a PRB over the TTI duration as shown in Equation (3.1). The bit rate is used for computing UE priorities, in case a channel-aware packet scheduler is configured.

$$R = \frac{R_c N_{bits}^{Symbol} N_{Symbol}^{PRB}}{T_{TTI}} \quad (3.1)$$

The data transmitted in a given TTI is split into multiple Transport Blocks (TBs). We consider the same number of TBs $N_{TB} = 5$ for every TTI. To decide whether a transmission of a TB was successful, a BLER-based coin is flipped based on the applicable BLER curves with the realised SINR and the selected MCS. To determine the realised SINR, Mutual Information Effective SINR Mapping (MI-ESM) is used to aggregate the SINRs over all PRBs into one effective SINR (see [34] for details). The buffers are updated and the bits removed if the transmission was successful. If all bits of a packet have been successfully sent, the timestamp of the last transmission is recorded as the final arrival time for that packet.

3.2.4 Packet Scheduling

This section presents the modelling aspects of the baseline non-cross-layer schedulers first, and then explains the two types of cross-layer information used and how they affect the scheduling decision. The baseline non-cross-layer schedulers are the Maximum-Largest Weighted Delay First (M-LWDF) and Earliest-Due-Date (EDD) schedulers. We did not include the widely used Proportional Fair (PF) scheduler, as the PF scheduler is purely throughput based and thus not suited for scheduling latency-sensitive packets. Another latency-aware alternative also used in practice is the Exponential/Proportional Fair (EXP/PF) scheduler. It is not included here, since it was shown that the M-LWDF scheduler, which also includes a PF component, outperforms it as a whole [12].

3.2.4.1 Non-Cross-Layer Schedulers

Traditionally, latency-based schedulers only take into account a packet's RAN latency [11] which does not require the BS to learn additional information about a packet other than its time of arrival at the BS. The decision whether a packet is dropped from the buffer is only made based on whether it has exceeded the configured RAN latency budget, usually configured to a value in the order of few to tens of milliseconds.

M-LWDF-Scheduler

The M-LWDF packet scheduler [12], a mix of latency-based and PF scheduling, is one of the baseline non-cross-layer schedulers in this study. It calculates the UE priorities $P_{i,m-lwdf}(t)$ for UE i at TTI t as follows:

$$P_{i,m-lwdf}(t) = -\frac{\log_{10} \delta}{\Delta_{LAT}} t_{i,LAT} \frac{R_i(t)}{\hat{R}_i(t-1)} \quad (3.2)$$

The latter part of the equation is a PF component, with $R_i(t)$ denoting the instantaneously attainable bit rate and $\hat{R}_i(t-1)$ the exponentially smoothed bit rate experienced by UE i thus far. δ denotes the maximum allowed PDR of the flow, that would only influence the scheduling if different UEs have different maximum allowed PDRs, which is not the case here. $t_{i,LAT}$ is the experienced latency of the packet at the Head Of Line (HOL) in the buffer of UE i and Δ_{LAT} the latency budget. As we will have different flavours of the schedulers, experienced latency and latency budget here can be either the above-mentioned experienced RAN latency t_{RAN} and RAN latency budget Δ_{RAN} , or the experienced E2E latency t_{E2E} and E2E latency budget Δ_{E2E} explained below.

EDD-Scheduler

The EDD scheduler [20] represents a purely latency-based approach of scheduling, giving priority to the UE with the most urgent latency deadline, or due date, with $T_{i,Deadline,i} - t$ representing the remaining time from the current time t until the deadline $T_{i,Deadline}$ of the packet at the HOL in the buffer of UE i .

$$P_{i,edd}(t) = \frac{1}{T_{i,Deadline} - t} \quad (3.3)$$

As we want to schedule the packet with the most urgent deadline, the UE priority is its inverse. $T_{Deadline}$ will be determined differently depending on whether the scheduler follows a fixed RAN latency budget or the E2E latency budget explained in the next section. It is important to note that unlike the M-LWDF scheduler, the EDD scheduler is channel-oblivious, which impacts the resource efficiency.

3.2.4.2 Cross-Layer Schedulers

With the M-LWDF and the EDD as the non-cross-layer schedulers, the main idea of cross-layer schedulers is extending the M-LWDF and EDD schedulers by using (1) E2E latency budget instead of RAN-latency budget, (2) I-frame prioritisation and (3) both E2E latency budget and I-frame prioritisation.

E2E Latency Budget

As opposed to the RAN latency budget, which applies at the IP packet level, and is set by the RAN operator, an E2E latency budget set by the application is the time in which a frame should successfully be conveyed from the source to the Head Mounted Display (HMD), including the time required for encoding, transmission and then decoding and rendering.

Since we do not consider parts of the video processing in our model, we solely focus on the part of the E2E latency budget for transmission in the IP network and 5G RAN for the cross-layer scheduling. As the latency from video processing differs depending on e.g. whether it is done on the device or edge, we take this into account by assessing different values for the E2E networking latency budget.

To calculate the E2E latency of a packet, we first look at which frame it belongs to. The frame index f and frame rate R_F determine the time T_f at which the frame has been generated at the

source, assuming time is initiated at zero at the start of the session

$$T_f = fT_{frame}, \quad \text{with} \quad T_{frame} = 1/R_F \quad (3.4)$$

Having an E2E networking latency budget Δ_{E2E} means the E2E deadline $T_{Deadline,f} = t_f + \Delta_{E2E}$ is identical for all packets n belonging to the same frame f . Similarly to above, packets will be dropped if the target E2E latency budget has been exceeded. With different arrival times of packets of the same frame at the BS, packets of the same frame will have different remaining RAN latency budgets $\Delta_{RAN-E2E,n}$, which are calculated as follows:

$$\Delta_{RAN-E2E,n} = t_f + \Delta_{E2E} - t_{network,n}. \quad (3.5)$$

$t_{network,n}$ denotes the latency the packet n may have already experienced due to dispersion in the network at the source encoder (see Chapters 3.3 and 3.3.2). Correspondingly, the experienced total E2E latency of a packet $t_{E2E,n}$ is calculated as

$$t_{E2E,n} = t_{network,n} + t_{RAN,n}. \quad (3.6)$$

with $t_{RAN,n}$ denoting the experiencing RAN latency of packet n .

I-Frame Prioritisation

Similarly to the scheduler with I-frame prioritisation from [11] introduced in Section 2.1, we use frame type as a form of cross-layer information in the scheduling. This is effectuated by including an additional parameter $\phi_{frame} \geq 1$ to give a higher weight for I-frame packets for the EDD and M-LWDF schedulers as follows

$$P_{i,edd-frame\text{type}}(t) = \begin{cases} \phi_{frame} P_{i,edd}(t) & \text{for I-frame} \\ P_{i,edd}(t) & \text{for P-frame} \end{cases} \quad (3.7)$$

$$P_{i,mldwf-frame\text{type}}(t) = \begin{cases} \phi_{frame} P_{i,m-lwdf}(t) & \text{for I-frame} \\ P_{i,m-lwdf}(t) & \text{for P-frame} \end{cases} \quad (3.8)$$

We consider different options for ϕ_{frame} and assess the performance impact in Section 4.1.

All in all, we have eight different schedulers in total. Two non-cross-layer schedulers that are RAN-latency based (*RAN-EDD* and *RAN-M-LWDF*), the variations using E2E instead of RAN latency budgets (*E2E-EDD* and *E2E-M-LWDF*), the variations using additional frame type information, (*RAN-Frametype-EDD* and *RAN-Frametype-M-LWDF*) and lastly the two schedulers using both types of cross-layer information (*E2E-Frametype-EDD* and *E2E-Frametype-M-LWDF*).

3.3 Traffic Model

This section describes the motivation and model behind creating packet traces from real video data and modelling the packet dispersion caused by both the video encoder and series of internet routers the packets travel through towards the RAN.

3.3.1 Creating Packet Traces from Recorded Video

As an extension to [14], we use a more realistic video traffic model in this thesis. In particular, instead of choosing constant values throughout the simulation for traffic parameters such as the number of packets per I- and P-frame or the size of each packet, we generate a packet trace from a real video and model the burstiness of the video encoding process [32].

Generating Video Traces

We use a recorded video sequence with characteristics similar to one in a real social XR conference, with a neutral office setting and limited user movement. These points about the video content are insofar crucial, as it has been shown that when evaluating QoE metrics, such video characteristics play an important role in determining the perceived quality, as repeatedly demonstrated [16,21,35]. Figure 3.5 shows two snapshots of the recorded video with an indoor office background and different user positions due to head movement.



(a) Snapshot of user looking at the camera

(b) Snapshot of user looking away from camera

Figure 3.5: Snapshots of recorded video with indoor office background and different user positions.

Using GStreamer [36], we can stream the video packet-wise through the network and capture the resulting packet trace with a packet analyser tool like Wireshark [37] or tcpdump [38]. When streaming the video, we set the application parameters such as the bit rate, the frame rate and the GoP size. The default video bit rate is chosen as 100 Mbps based on the minimum requirement for 'entry-level VR' [6] and will be varied later, but throughout the simulations, we use a constant frame rate R_F of 30 frames per second [6] and a GoP size of ten frames (one I-frame and nine P-frames per GoP) [17].

The models in [11, 14] are not considering B-frames and for simplicity, we do the same for our modelling approach. From the trace we read out the necessary parameters of each packet, which are its timestamp, size, number of the frame the packet belongs to, and the frame type of said frame. Table 3.1 shows a partial output of a trace containing five packets, parsed as comma-separated values (CSV). From left to right, the values shown in each row are the packet timestamp in seconds, the packet size in bytes, the RTP timestamp, the frame number, and the frame type (either I- or P-frame). The packet timestamps shown here are the recorded network timestamps by Wireshark due to the experimental setup. These are removed as we will model and enforce the packet dispersion of the video encoder.

Modelling Burstiness of Video Encoder

We include a model for the burstiness of the output from a video encoder, as in practical implementations, the encoder releases the packets of a video frame in a burst, which is then passed through a smoother before they are sent into the network [32]. In our case, the parsed timestamps in the packet trace are network timestamps recorded by Wireshark. Since the start of the timestamps have to match with the start time in the simulator, they have to be adjusted to represent the burstiness pattern of the encoder.

Table 3.1: Snippet of a CSV packet trace.

packet timestamp	packet size	RTP timestamp	frame number	frame type
1629202824.60665	1428	4119239446	10	I
1629202824.60666	1428	4119239446	10	I
1629202824.60667	1142	4119239446	10	I
1629202824.63926	42	4119242446	11	P
1629202824.63933	183	4119242446	11	P

First, for all N_{frames} frames in the video, we give each of the n packets of frame f an identical timestamp τ_f by taking the frame index f (for $f = 0, 1, \dots, N_{frames} - 1$) and multiplying it by the inter-frame time T_{frame} , the inverse of the video’s frame rate R_F .

$$\tau_f = fT_{frame}, \quad \text{with} \quad T_{frame} = 1/R_F \quad (3.9)$$

for all n packets in the frame.

We then model the packet dispersion at the encoder output by defining a burstiness parameter $b \in [0, 1]$ that determines the maximum percentage of the inter-frame time T_{frame} , wherein all packets of a frame can be dispersed. This is denoted as ΔF and calculated as follows

$$\Delta F = (1 - b)T_{frame}. \quad (3.10)$$

Within ΔF , all packets are dispersed with the same inter-packet time, denoted δp .

To determine δp for a specific video stream and burstiness b , we:

- i Determine the largest number of packets $N_{f_{max}}$ in a single frame in the video stream, with $N_{f_{max}} = \max_f N_f$, where N_f denotes the number of packets in the frame f .
- ii Obtain the inter-packet time $\delta p = \Delta F / N_{f_{max}}$, based on the largest frame in terms of number of packets
- iii Calculate the new packet timestamps $\tau_{f,n}$ of the n -th packet in frame f as:

$$\tau_{f,n} = fT_{frame} + n\delta p, \quad \text{for } n = 0 \text{ to } N_f - 1. \quad (3.11)$$

This approach ensures that we have the same inter-packet time δp throughout the video stream while keeping the departure of the first packet of each frame synced with the frame generation times without an overlap of packets belonging to different frames.

Synchronisation of Traffic Streams

In practice, users will join a social XR session not necessarily at the same time. As such, the start of their VR streams, which is always an I-frame, will not be synchronised. In the worst-case scenario it can happen that the I-frames of all users overlap and create a spike in traffic, which we define as the perfectly synchronised case. In the maximally asynchronised case, we consider the I-frames of all users to be maximally temporally separated. We calculate this I-frame offset per user as the time of a GoP over the number of total users. In an example of four users and GoP size ten, which is depicted in Figure 3.6, this would result in a maximal offset of 2.5 frames (0.083 s) of the I-frames.

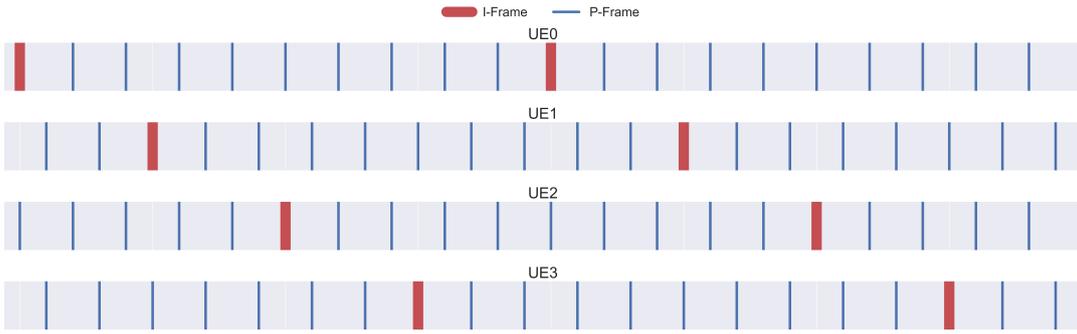


Figure 3.6: Frame type sequence for four maximally asynchronised video traffic streams.

3.3.2 Modelling Packet Dispersion in IP Networks

The VR packet stream will not only experience dispersion at the video encoder, but also by travelling through the IP network to the BS transmission buffer. As our cross-layer approach also exploits knowledge of the total E2E latency budget as compared to previous models [11], we implement a simulator to create realistic random packet dispersions caused by packets going through multiple network hops that affect the experienced latency and consequently the remainder of the E2E latency budget of packets arriving at the BS buffer. Further, we assume that all packets of a given session would follow the same route through the IP network, we ignore the possibility that packets of a later frame could overtake packets of an earlier frame and hence arrive earlier at the BS buffer. In practice, this would mean that only an E2E latency-based scheduler could rearrange packets in the BS buffer and deal with this occurrence properly.

Figure 3.7 illustrates a series of network hops between the source at the MCU and the RAN BS, with two types of packet streams as inputs. We model routers at every network hop as a First-In-First-Out (FIFO) queue and track the timestamps of the VR packets after every queue. As shown in the figure, the two types of inputs for the queues are the foreground VR packet stream, and the background traffic generated according to independent but identically configured Poisson-based packet generation processes at each router. By adding a certain amount of background traffic, the VR packets experience random queuing delays before entering the next hop. This way we get a certain somewhat randomised dispersion pattern for the VR packets after going through a specific number of queues.

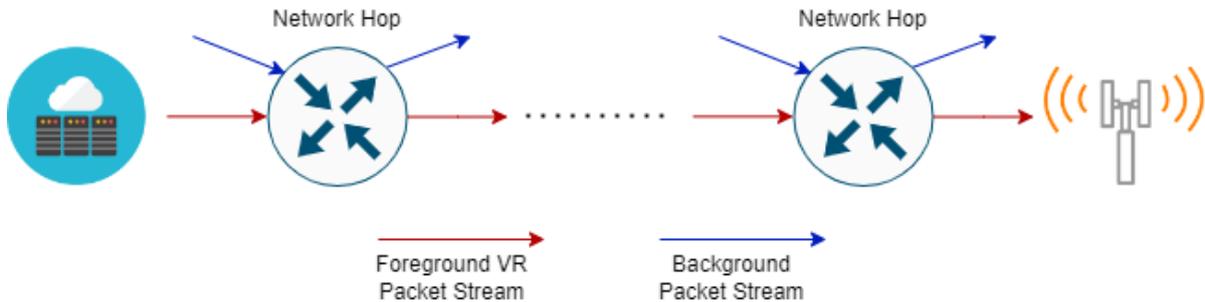


Figure 3.7: Series of network hops between video source and BS with fore- and background packet streams.

Modelling Scenario Parameters

Here we define the main modelling aspects for the network hop simulator and the assumed scenarios and the rationale behind those choices.

We define three scenarios to give an idea of the impact of different numbers of hops representing different distances between source and destination. The first is between two nearby cities within a country (e.g Amsterdam - Delft). The second is between two cities further away from different countries but not yet intercontinental (e.g Amsterdam - Berlin). Lastly, we consider an intercontinental scenario (e.g Amsterdam - New York). Using traceroute [39], we chose five, ten, and fifteen as the number of hops between the two locations in the three scenarios (see also Appendix A.2). Additionally, a propagation delay t_{prop} is considered for every scenario in a straightforward manner based on the straight-line distance d between the two locations and assuming signals are travelling at the speed of light in glass fiber.

In practice, multiple routers with different capacities (1/10/100 Gbps) are deployed [40]. As we will be simulating multiple queues with realistic packet sizes in the tens or hundreds of bits compared to the capacity of a switch in the order of Gbps, there will be hundreds of thousands, if not millions of iterations of events that need to be simulated. To keep the computational complexity manageable, we choose a traffic handling bit rate in each router of $S_{router} = 1$ Gbps.

Now, we determine the modelling aspects of the background traffic stream and define the considered scenarios. Studies have concluded that a majority of IP traffic consists of TCP flows and observed a large part of very small packet sizes just above 40 bytes, which are TCP acknowledgements and large packets around 1500 bytes, which is the maximum Ethernet packet size. Based on studies of internet traffic characteristics [41, 42], for our background packet size distribution P_{BG} we choose the packet sizes as 44, 1300, and 1500 Byte, and their respective probabilities as 44%, 19%, and 37% for our model. This results in an average packet size of $s_{avg} = 821$ Byte.

We define the background load L_{router} as a percentage of S_{router} . Test runs revealed that for background loads lower than 50% for all three numbers of hops, there is no huge difference in the final packet dispersion. To choose a scenario with a very high background load, the additional load to the system from the VR stream has to be considered. With 90% background load and the VR stream having 100 Mbps default bit rate, the system will be overloaded, which we want to avoid. Therefore we choose 50% as the lowest amount of background load, 70% as a still manageable amount, and 85% as the highest amount of background load which does not yet lead to an overloaded system. Using the average packet size s_{avg} and the different background loads, we obtain the arrival rate λ_{BG} in average number of background packet arrivals per second for each of the three scenarios as $\lambda_{BG} = \{76103, 106610, 129366\}$ packets per s.

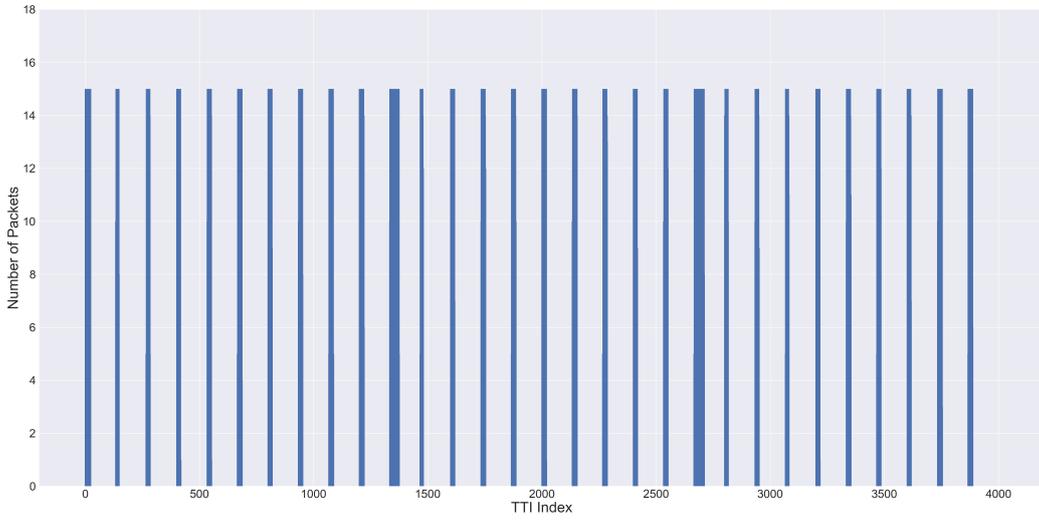
In summary, the parameters for the network hop simulator are as follows:

1. Number of queues N_{router} and propagation delays t_{prop}
 - I Amsterdam - Delft: $d \approx 60$ km, $N_{router} = 5$, $t_{prop} = 0.3$ ms
 - II Amsterdam - Berlin: $d \approx 600$ km, $N_{router} = 10$, $t_{prop} = 3$ ms
 - III Amsterdam - New York: $d \approx 6000$ km, $N_{router} = 15$, $t_{prop} = 30$ ms
2. Traffic handling bit rate of each router $S_{router} = 1$ Gbps
3. Packet size distribution of background traffic
 $P_{BG} = \{44 \text{ Byte (44\%)}, 1300 \text{ Byte (19\%)}, 1500 \text{ Byte (37\%)}\}$
with average packet size $s_{avg} = 821$ Byte

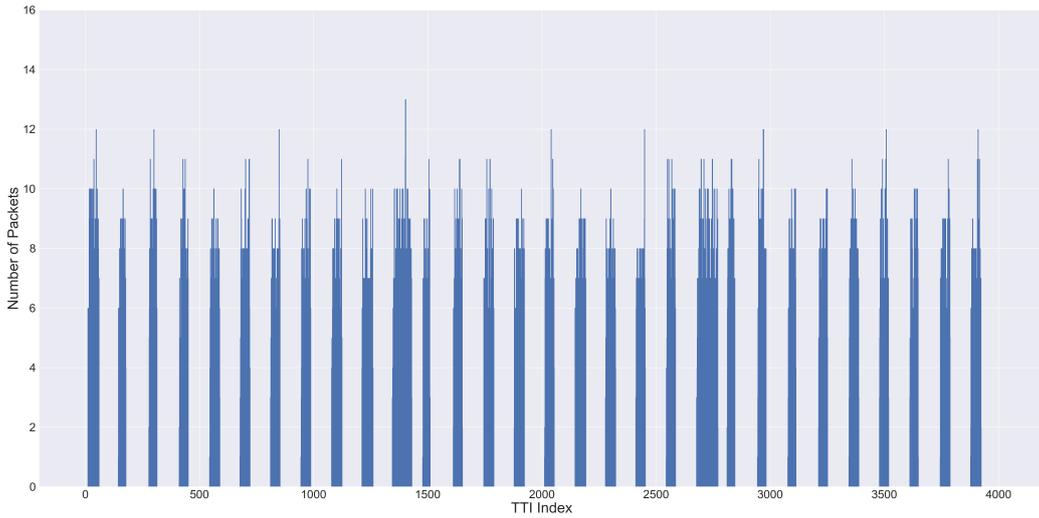
4. Background loads $L_{router} = \{0.5, 0.7, 0.85\}$,
with arrival rates $\lambda_{BG} = \{76103, 106610, 129366\}$ packets per s

Figure 3.8 shows an illustrative trace-based comparison for the packet dispersion in terms of packet arrivals per TTI at the BS for the input and output of the network hop simulator with 10 queues and 70% background load. The packet trace has an initial encoder burstiness $b = 0.6$.

We see in Figure 3.8a that even before entering the network hops the packet dispersion due to the encoder is already slightly larger for the I-frames, which comprise more packets, than for P-frames. As can be seen from Figure 3.8b, the additional dispersion causes the number of packet arrivals per TTI to go down from fifteen per TTI and also to be randomised instead of constant. Further, the dispersion of I-frame packets is now even larger, with the last packets of the second and third I-frame experiencing an E2E delay of more than half of the inter-frame time due to encoding and queuing delays.



(a) Trace with 100 Mbps bit rate and burstiness $b = 0.6$ before network hop simulator.



(b) Same trace after network hop simulator with $N_{router} = 10$, and $L_{router} = 70$.

Figure 3.8: Comparison of packet arrivals per TTI for input video packet trace (a) and output trace (b) of network hop simulator.

Chapter 4

Simulation Scenarios and Results

This chapter assesses the schedulers presented in the previous chapter, particularly concerning the benefits of exploiting cross-layer information. We first lay out the range of simulation scenarios and parameters for the sensitivity analysis and the motivation behind those choices in Section 4.1. Then in Section 4.2, we present and analyse the sensitivity analysis results, particularly the performance impact of different types of cross-layer schedulers. Next, in Section 4.3 we determine for a subset of schedulers what combinations of the number of UEs, the application bit rate, and the E2E latency budget are achievable given an imposed 5% PDR requirement for a subset of schedulers. This section intends to give an overview of the social XR scenarios that can be supported from a network operator’s point of view. Lastly, we use the objective video quality metrics PSNR and SSIM and analyse their correlation with the previously obtained PDR results focusing on the context of the benefits from cross-layer scheduling in Section 4.4.

4.1 Simulation Scenarios and Approach

In this section we will define the simulation scenarios through which the cross-layer schedulers are assessed and how the choice of default configurations was made. The performance indicator with which the schedulers and scenario parameters are assessed is the PDR. We chose **5%** as a threshold, which was shown to be an acceptable network QoS without too much QoE degradation [43]. Contrary to previous work [14], we not only count the packets dropped at the BS resulting in a RAN PDR but also those that have been transmitted successfully to the UE but have not arrived within the given E2E latency budget, giving us an E2E PDR. Note that with the considered schedulers in this study, dropping additional packets after transmission may only occur with the RAN latency-based schedulers.

We simulate every set of parameters for 20 different random generations of radio channels. The resulting PDR in each plot is the average over all seeds and the number of users in the scenario. Each bar plot includes a black bar indicating the confidence interval at a 95% confidence level.

First, we conduct a sensitivity analysis for different scenario parameters by varying them in a *ceteris paribus* manner. Table 4.1 shows the parameters and the range of values considered with the *default* highlighted.

To determine the default carrier bandwidth, we set all parameters to their default value and chose the minimum bandwidth needed to achieve <5% PDR with the best non-cross-layer scheduler. 125 MHz was shown to be insufficient to support the default scenario with <5% PDR, hence, we chose 150 MHz as the default carrier bandwidth (see Appendix C.1).

Table 4.1: Parameters and (default) values considered in the sensitivity analysis.

Parameter	Values (<i>Default in italics</i>)
Application bit rate	50, 100, 150, 200 Mbps
E2E latency budget	25, 50, 100 ms
RAN latency budget	1, 5, 10, 20, 30, 40, 50 ms
Carrier bandwidth	100, 150, 200 MHz
Number of UEs	1, 2, 4, 6, 8
Background traffic load	50%, 70%, 85%
Number of hops	5, 10, 15
Encoder burstiness	0.6, 1.0
Synchronisation of video traffic streams	<i>max. asynchronous</i> , perfectly synchronised

Next, the default RAN latency budget is selected such that the PDR can be minimised for the non-cross-layer packet scheduler. Figure 4.1 shows the PDR versus the RAN latency budget for the non-cross-layer RAN-EDD (left chart) and RAN-M-LWDF (right chart) schedulers for all nine configurations of the background traffic load and number of hops. All curves show the same general pattern, in the sense that a decrease in RAN latency budget leads to a higher PDR, which is caused by increases packet dropping due to insufficient radio resource. After a certain minimum PDR, the PDR rises again for higher RAN latency budgets, since packets are increasingly dropped at the UE. We select 20 ms as the default RAN latency budget since it gives the lowest PDR in most scenarios for both schedulers. Since we consider a default E2E latency budget of 50 ms [5], a setting with fifteen network hops would exceed the 5% target PDR for all RAN latency budgets and does not influence the choice of default RAN latency budget for this sensitivity analysis.

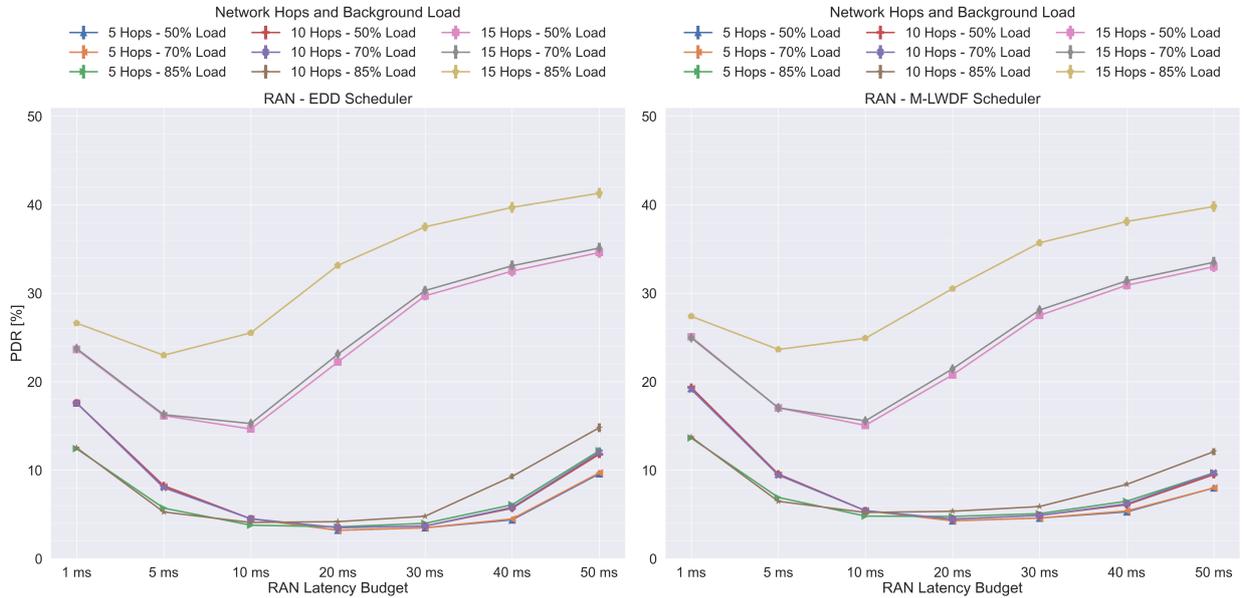


Figure 4.1: PDR for non-cross-layer schedulers for all combinations of background network load, number of hops and range of RAN latency budgets.

In Section 3.2.4.2, we have defined the I-frame prioritisation parameter ϕ_{frame} . Here, we determine the value of ϕ_{frame} , such that the performance of the frametype-cross-layer schedulers is optimal. In Figure 4.2, we show the PDR for all frames, I-frames, and P-frames versus different values of ϕ_{frame}

for the four cross-layer schedulers using frame type information. For all schedulers, we observe that the higher the value of ϕ_{frame} is, the lower the PDR for I-frames becomes, with gradually diminishing returns until a minimum is reached. On the other hand, the PDR for P-frames will naturally increase with ϕ_{frame} . This is likewise the case for the PDR of all frames, because although the number of packets per I-frame is higher than per P-frame, the substantially higher number of P- than I-frames in a GoP results in a net increase in the overall PDR. We see for the E2E latency-based schedulers, that the green and blue curves increase less dramatically for very high values of ϕ_{frame} , as compared to the RAN latency-based schedulers. In E2E latency-based schedulers, packets of the same frame have the same E2E deadline, and as such smaller values for ϕ_{frame} can already result in the scheduler purely prioritising I-frames over P-frames, with packet latencies in the order of milliseconds. This is the most noticeable for the E2E-Frametype-EDD scheduler, where $\phi_{frame} = 2$ already leads to the I-frame packets having constant priority, as it does not contain a PF component and purely considers E2E latency deadlines. Since RAN latency-based schedulers consider different RAN latencies and hence priorities for different packets of the same frame, a convergence is only observed for much higher values of ϕ_{frame} . We configure ϕ_{frame} as 5, meaning I-frame packets will receive a five times higher scheduling priority than P-frame packets, since for $\phi_{frame} = 5$ most of the gains have been attained for the I-frame PDR for all schedulers, while the increase of the PDR of P-frames and all frames is still reasonably low. Note, that a setting of $\phi_{frame} = 1$ effectively implies turning off the frame type-based scheduling. Comparing with that, the charts in Figure 4.2 clearly show attainable gains from the I-frame PDR perspective.

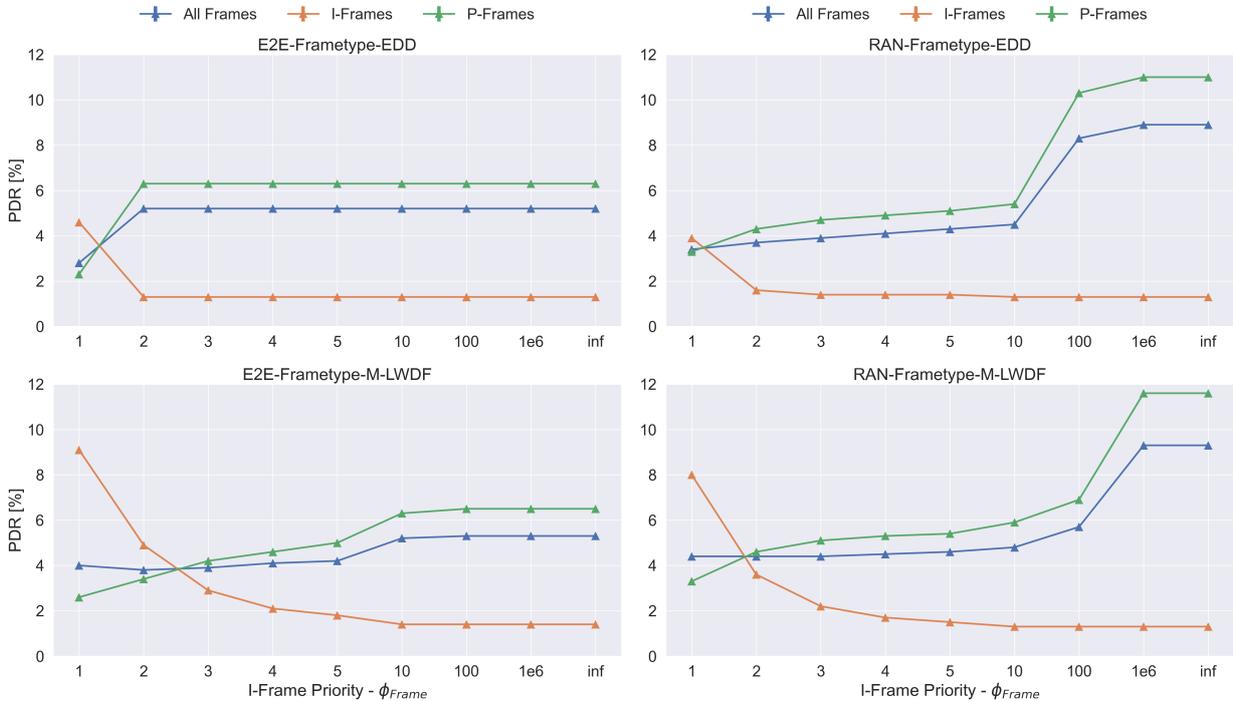


Figure 4.2: PDR for increasing values of ϕ_{frame} for all four cross-layer schedulers using frame type information.

4.2 Results of Sensitivity Analysis

In this section, we present the results from the sensitivity analysis and assess the performance of the (non-)cross-layer schedulers. Every figure below contains three bar plots depicting the PDR for all frames on the top, I-frames in the middle, and P-frames on the bottom. As mentioned previously, we set all parameters except the one named in the figure titles to their respective default values according to Table 4.1. For better clarity, the maximum value for the PDR on the vertical axis is capped at 30%.

4.2.1 Comparison of Schedulers for Default Scenario

To assess how the eight schedulers perform compared to each other, we first look at the results for the overall PDR, the I-frame PDR, and the P-frame PDR for the default scenario, as shown in Figure 4.3.

EDD vs M-LWDF Schedulers

First, we observe that the EDD schedulers (light colours) generally achieve a lower PDR than the M-LWDF counterpart (dark colours). This result is due to the PF component in the M-LWDF scheduler (see Equation (3.2)), which ensures resource fairness among UEs. Consequently, a UE containing I-frame packets in its buffer actually requires more resources than other UEs with only P-frames, which comprise fewer packets. However, due to the nature of the PF component, the UE with I-frame packets will not receive the additional resources needed to transmit all I-frame packets, leading to increased I-frame PDRs. This situation is also clearly noticeable in Figure 4.3. In the middle plot, we see that the PDRs of the I-frames for the RAN- and E2E-M-LWDF scheduler without frame type differentiation are much higher than for the RAN- and E2E-EDD schedulers.

RAN vs E2E Latency-Based Schedulers

The gain in overall PDR of E2E latency-based schedulers compared to RAN latency-based schedulers is minor in the default scenario. As a matter of fact, we observe a minor increase in I-frame PDR when using E2E latency-based schedulers. This shows that at the cost of requiring to tune the RAN latency budget, RAN latency-based schedulers can come very close in performance to E2E latency-based schedulers. We see that the PDR of E2E schedulers is lower for the P-frames but slightly higher for I-frames. As I-frames contain more packets and are thus more affected by packet dispersion, tuning the RAN latency budget optimises the PDR of I-frames. Tuning is complex in practice when the RAN operator does not know the paths of the video packets across the internet on their way to the BS and hence has no knowledge about the packet dispersion. Hence in practice, the operator would likely conservatively choose a RAN latency budget based on a range of possible scenarios in terms of pre-RAN latency and packet dispersion.

Frametype vs Non-Frametype Schedulers

Next, we assess the results from utilising frame type information in the schedulers. We see, as expected, that the I-frames' PDRs of the frametype schedulers are significantly lower than those of non-frametype schedulers. The downside is that the P-frames will have many more dropped packets, leading to a higher overall PDR for cross-layer schedulers using frame type information. Section 4.4 will discuss how the different distribution of dropped packets affects the experienced video quality by end users, using the PSNR and SSIM as objective video quality metrics, which effectively combine the effects of I- and P-frame PDR in a single meaningful metric.

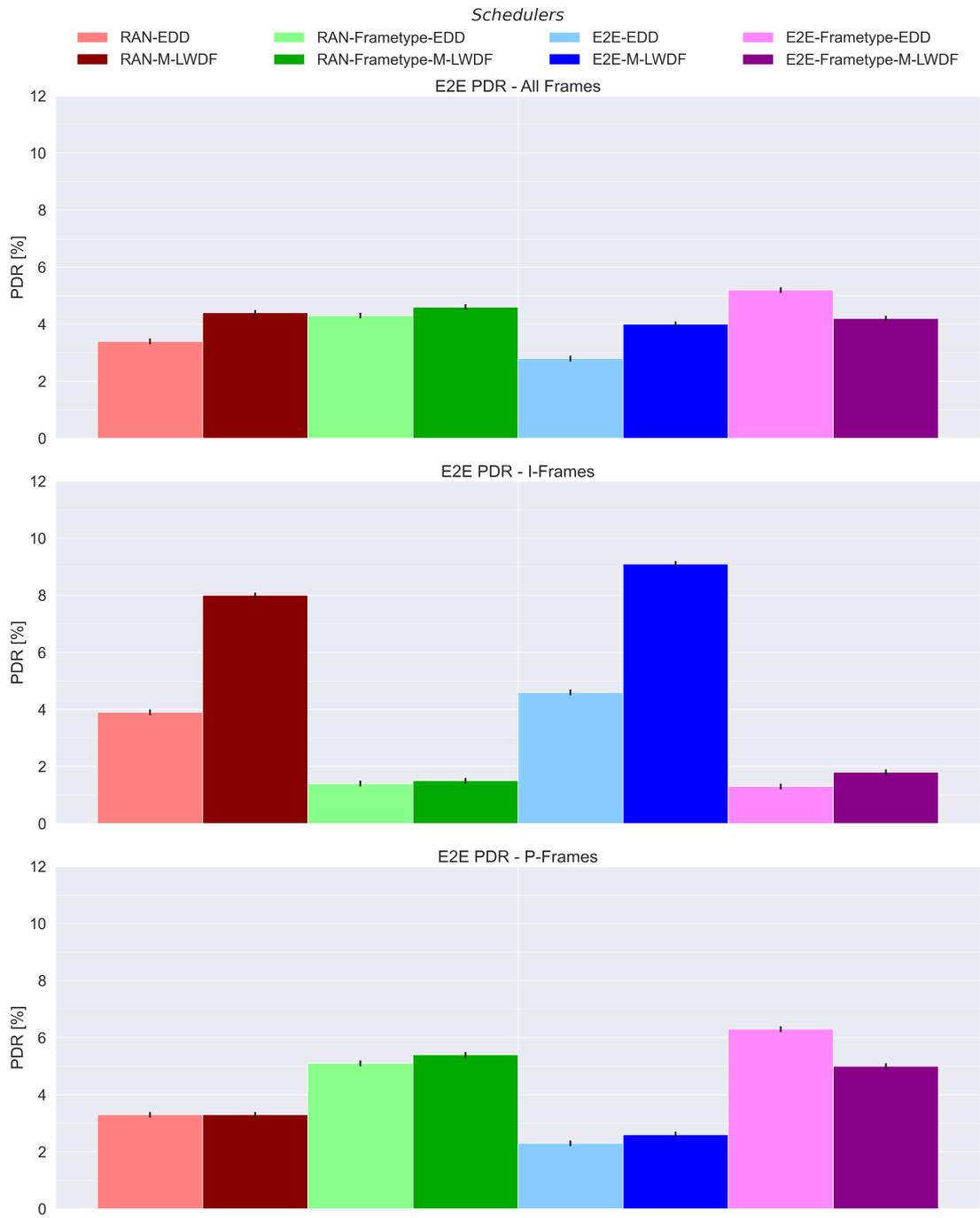


Figure 4.3: PDRs for all schedulers with parameters set to default according to Table 4.1.

One exception to the points mentioned above is the performance of the E2E-Frametype-EDD scheduler. Contrary to the observations above, it gives a higher PDR than the E2E-Frametype-M-LWDF or RAN-Frametype-EDD schedulers and shows the highest overall PDR. The explanation for this outlier in the results lies in how the E2E-Frametype-EDD scheduler prioritises I-frame packets based on the E2E latency budget.

To understand these results, recall that for both the E2E-EDD and the E2E-Frametype-EDD schedulers, the E2E deadline for packets belonging to the same frame will be the same. With the asynchronous scenario considered here, the I-frame packets of all UEs are maximally spread out within a GoP (see Figure 3.6). When the E2E-Frametype-EDD scheduler prioritises I-frame packets, all packets of the I-frame will have the same deadline and thus the same level of priority over P-frame packets in the buffer of other users. For example, assuming UE A has I-frame packets in its buffer, the E2E-Frametype-EDD scheduler will only schedule the other UEs after UE A’s buffer has no I-frame packets left at its HOL. With I-frames consisting of more packets than P-frames, they also experience larger packet dispersions when travelling through network hops before arriving at the BS (see Figure 3.7b). UE A, in this example, will be scheduled for many TTIs in a row, resulting in low throughput fairness and leading to high PDRs for the other UEs’ P-frames. These effects are reflected in a higher average P-frame PDR and a higher net increase in overall PDR for the E2E-Frametype-EDD scheduler. Since packet deadlines for the packets of a given I-frame are different in the case of, e.g. the RAN-Frametype-EDD schedulers, prioritising one UE for such an extended period of TTIs will not happen, resulting in a lower P-frame PDR.

4.2.2 Impact of Scenario Parameters

This section assesses the impact of various system parameters on the schedulers’ performance. We divide the parameters based on how they affect the overall system. We first examine the video application bit rate, channel bandwidth, and number of UEs served by the BS. These all affect the total effective load offered to the BS. Then, we examine the outcome of varying two types of latency budgets; the packet-level RAN latency budget set by the network operator and the frame-level E2E latency budget required by the application. Next, we assess the different causes of packet dispersion, including the number of network hops, the amount of background load in the network hop simulator, and the encoder burstiness b . Finally, we show the impact of having perfectly synchronised versus maximally asynchronous video traffic streams.

4.2.2.1 Varying System Load

The video application bit rate (Figure 4.4), the available carrier bandwidth (Figure 4.5) and the number of UEs (Figure 4.6) are all parameters that directly influence the total load on the RAN BS. As expected, we see the trend for all three parameters, that increasing the load leads to higher PDRs for all schedulers.

In all three figures, the observed increase in PDR is not linear but exponential. For example, increasing the load by only 50%, such as increasing the video bit rate from 100 to 150 Mbps (Figure 4.4) or the UEs from four to six (Figure 4.6), increases the overall PDR roughly eightfold. Figure 4.5 confirms this observation as well. It shows that increasing the carrier bandwidth by a third barely nets any gains in terms of PDR, while decreasing it by a third shows an increase in PDR by roughly three times. These results suggest that when deploying a system in practice, these configurations have to be well-tuned for a given network QoS target, as overloading the system can considerably deteriorate the performance and potentially significantly affect the end user experience. Notably, the relative performance overall of the different schedulers does not fundamentally change. But the distinctions between the frametype and non-frametype schedulers are more accentuated under

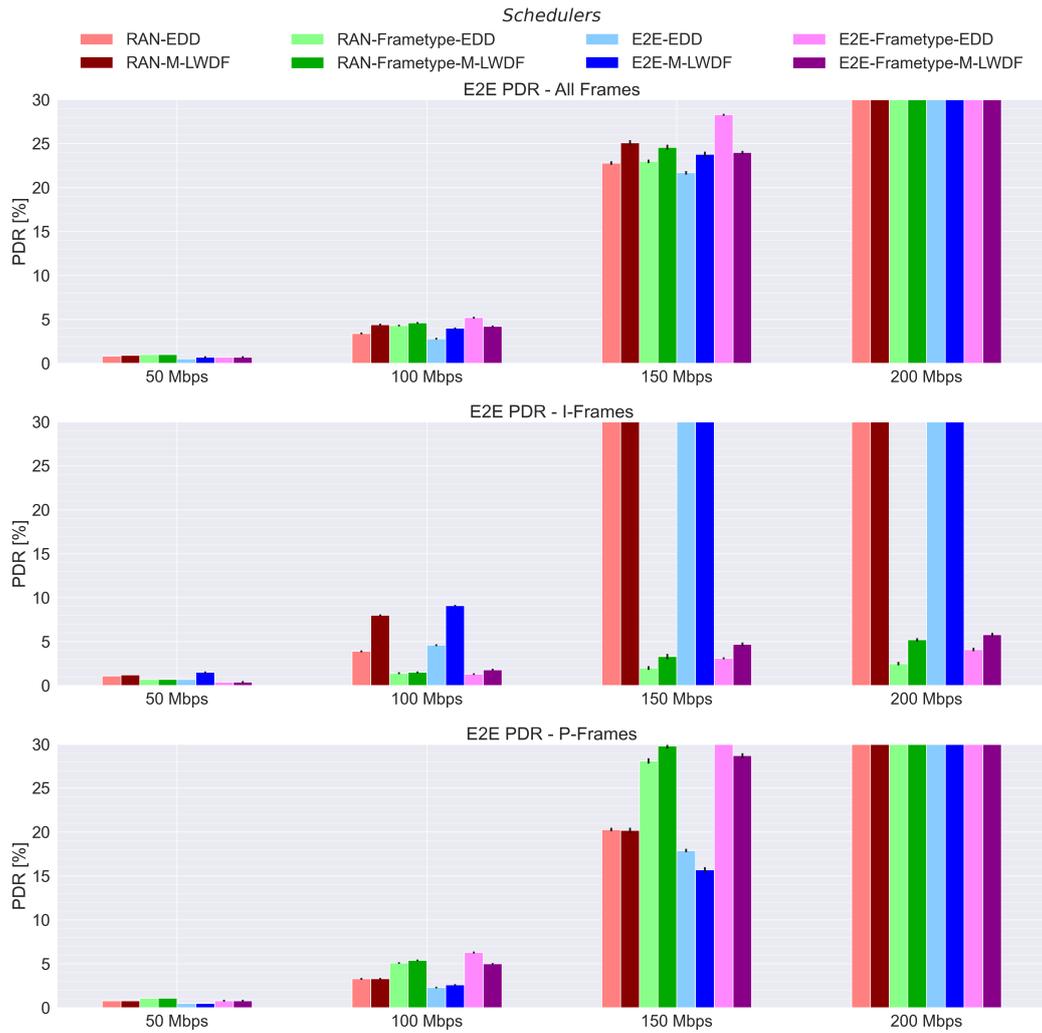


Figure 4.4: PDRs of all schedulers for different video application bit rates.

higher cell load scenarios. We observe that in these scenarios, where the overall PDR is drastically increased, the I-frame PDR stays low for all frametype schedulers. For example in Figure 4.6, we see that the overall PDR for all schedulers in a setting with eight UEs is way above 30%, whereas the I-frame PDR is still in many cases within the 5% target.

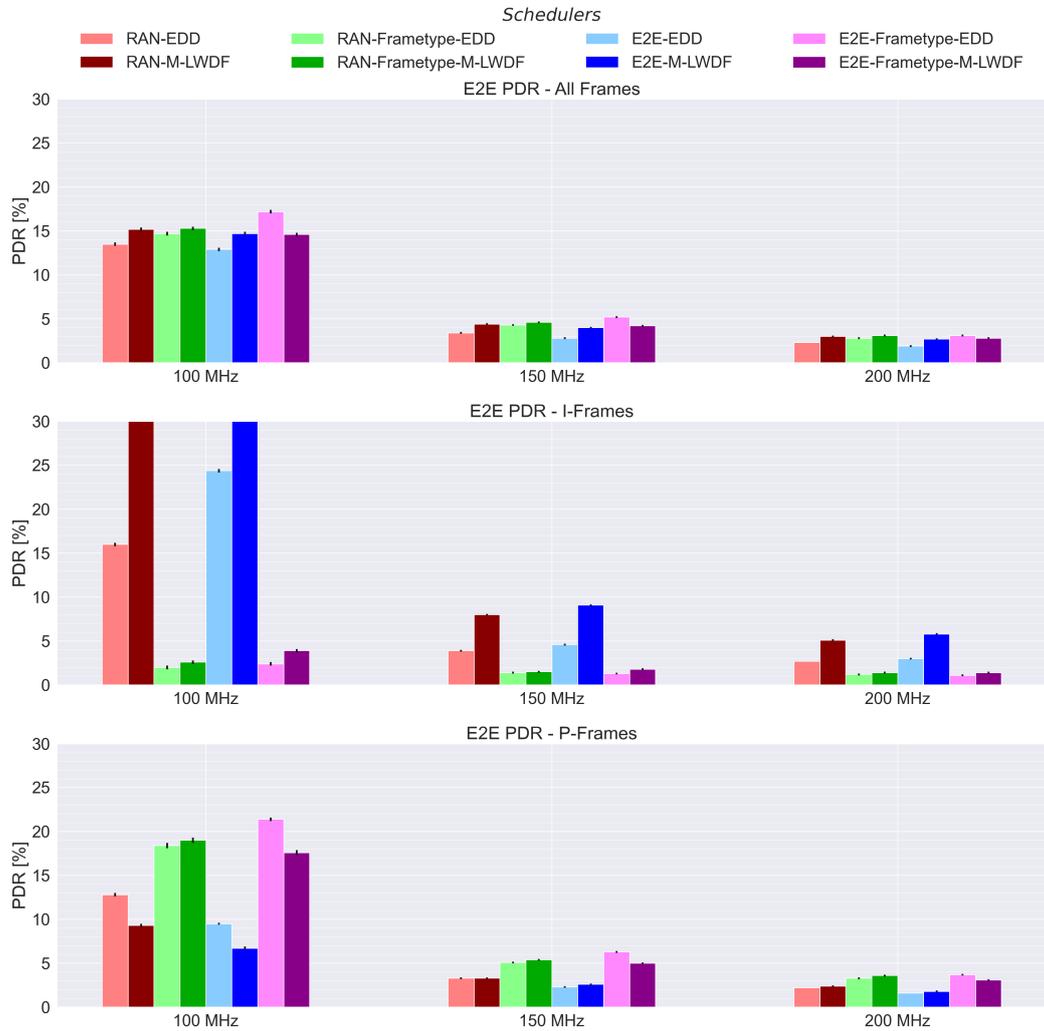


Figure 4.5: PDRs of all schedulers for different carrier bandwidths of the RAN BS.

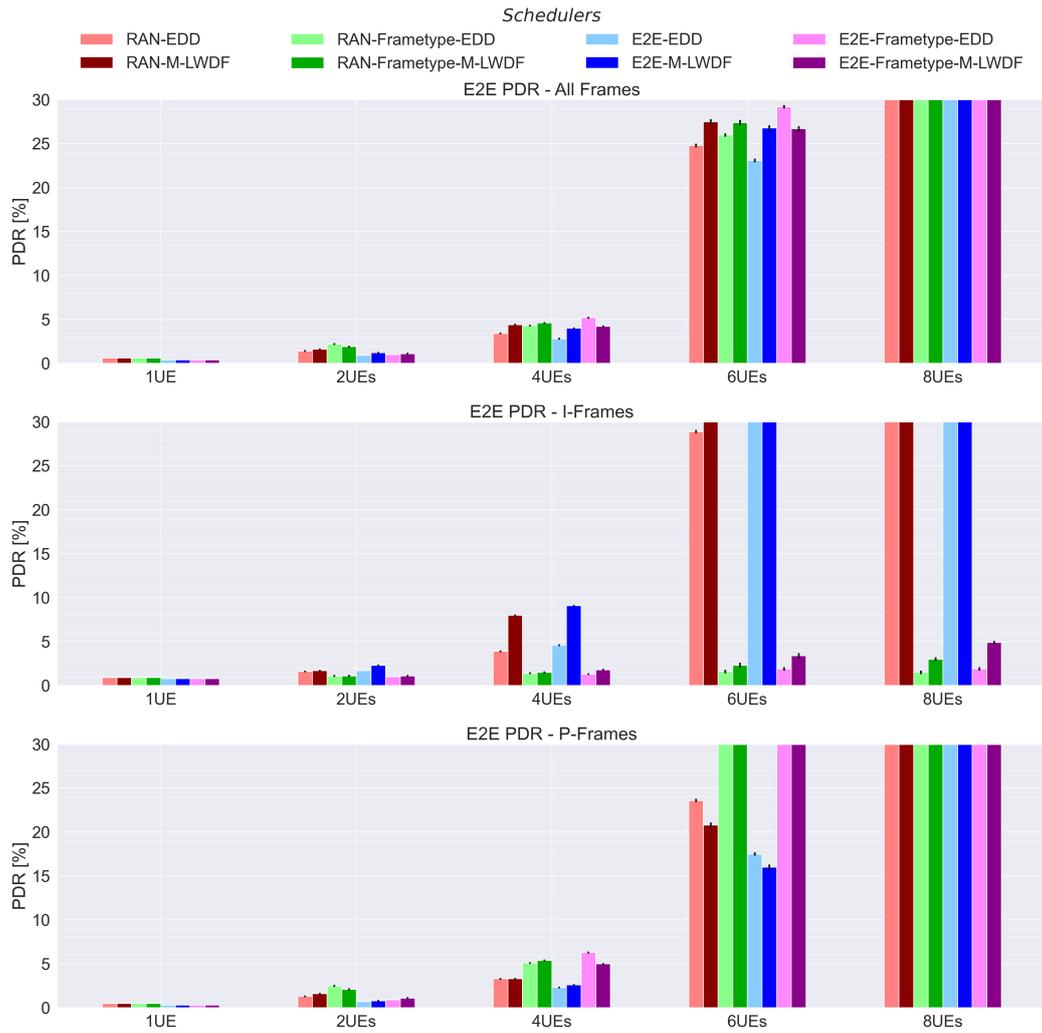


Figure 4.6: PDRs of all schedulers for different number of UEs in the RAN.

4.2.2.2 Varying RAN and E2E Latency Budgets

Here, we look at the implications of the mobile network operator configuring different packet-level RAN latency budgets and the application imposing different frame-level E2E latency budget requirements. First, we see the results for different RAN latency budgets in Figure 4.7, assuming a fixed default value of the E2E latency budget. Since the RAN latency budget does not exist in the four E2E latency-based schedulers, their PDRs stay constant in all scenarios.

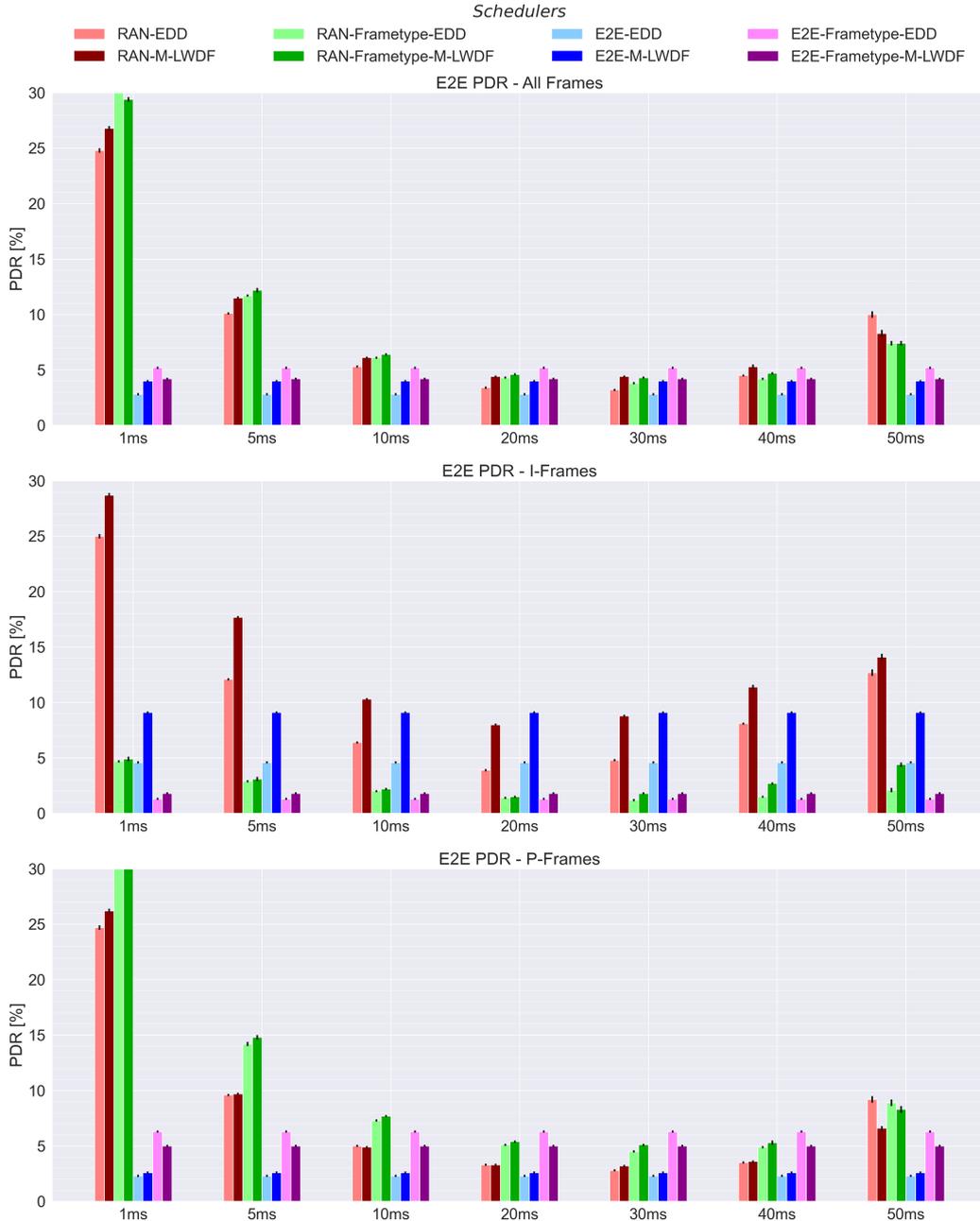


Figure 4.7: PDRs of all schedulers for different packet level RAN latency budgets.

Quite the opposite is the case for the RAN latency-based schedulers. We saw in the previous section that choosing the optimal RAN latency budget for a given E2E latency budget significantly improves their performance. With a larger RAN latency budget, only few packets will be dropped at the BS,

but more packets will in turn violate the E2E latency budget when arriving at the UE, leading to more packet dropping at the UE and a higher over PDR. Another key downside of a too high RAN latency budget is the waste of scarce radio resources corresponding with the packet transmissions that will later on be dropped at the UE. Configuring lower RAN latency budgets leads to many more unnecessarily dropped packets at the BS, causing a high PDR. Hence, the main tradeoff is reducing the unnecessary droppings at the BS and improving the PDR, while not increasing the RAN latency budget beyond the optimal value such that transmission resources are wasted on packets that will eventually be dropped at the UE. This resource waste leads to unnecessary congestion for radio resources and consequently more scheduling delays and dropping of packets.

In contrast, having different E2E latency budget requirements will affect all schedulers, as shown in Figure 4.8, which depicts the PDR for 25, 50, and 100 ms of E2E latency budget. We observe that RAN latency based schedulers see a higher increase in PDR than E2E latency-based schedulers if the E2E latency budget is reduced.

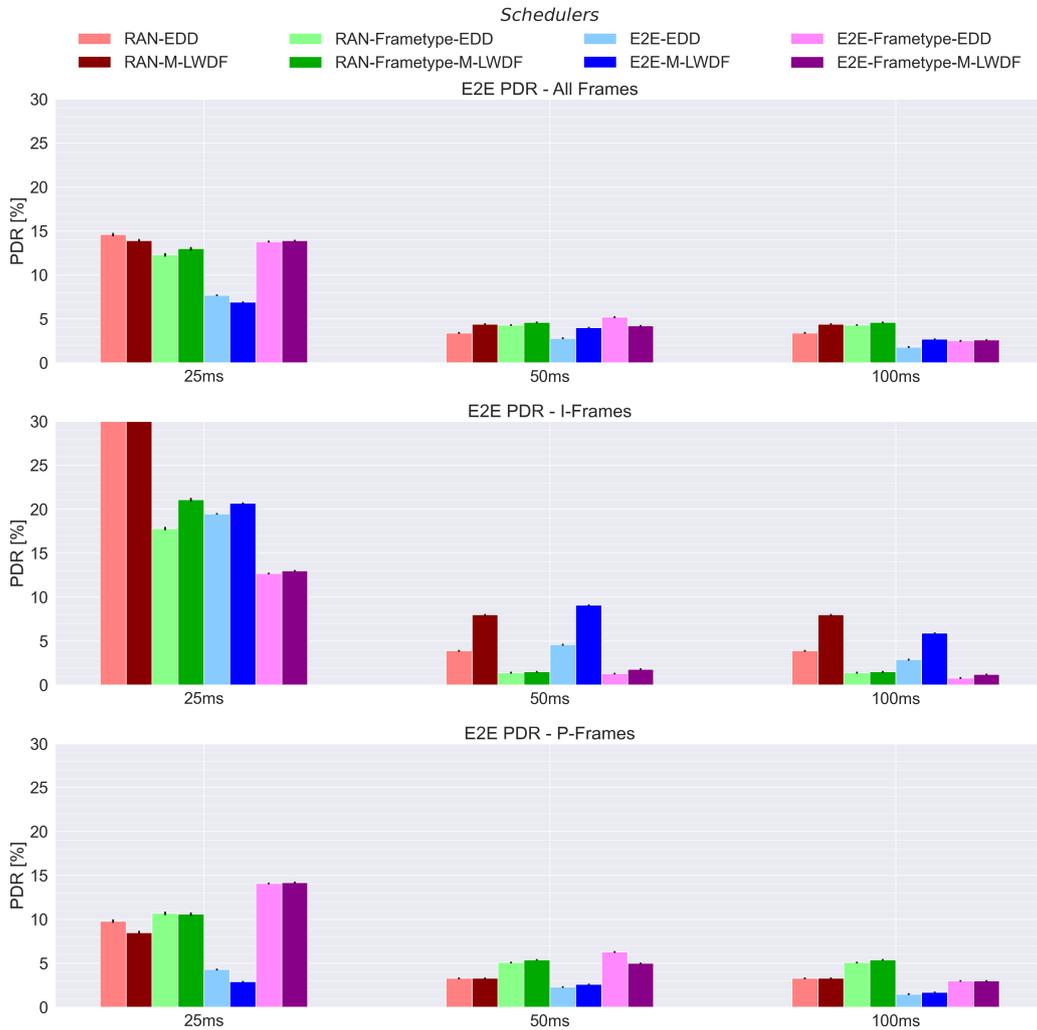


Figure 4.8: PDRs of all schedulers for different frame-level E2E latency budgets.

Ideally, the RAN latency budget would need to be retuned to the different E2E latency budgets. Since this is not done here, the PDR suffers more for the RAN-latency based schedulers for lower E2E latency budgets. Similarly, changing the E2E latency budget from 50 ms to 100 ms primarily benefits

the E2E latency-based schedulers. With a well-tuned RAN latency budget, the RAN latency-based schedulers see only very few or no packet droppings at the UE in the first place. Hence, the gain from raising the E2E latency budget is negligible for RAN latency-based schedulers.

4.2.2.3 Varying Packet Dispersion

Now we investigate the impact of three different causes of packet dispersion on PDR. Figure 4.9 shows the PDR for the two considered settings of encoder burstiness, with $b = 0.6$ (left charts) indicating the implementation of a smoother in the video encoder as mentioned in Section 3.3.1, and $b = 1.0$ indicating the absence of it, with all packets of a frame sent out in one burst. We see that there are no visible differences in the results, implying that the type of video encoder has no effect on the resulting packet dispersion and PDR. There would likely only be a noticeable difference with burstiness much lower than 0.6. As mentioned in Section 3.3.1, this could lead to packets from a later frame could overtaking a packet of an earlier frame and hence arriving earlier at the BS buffer, which we do not consider here.

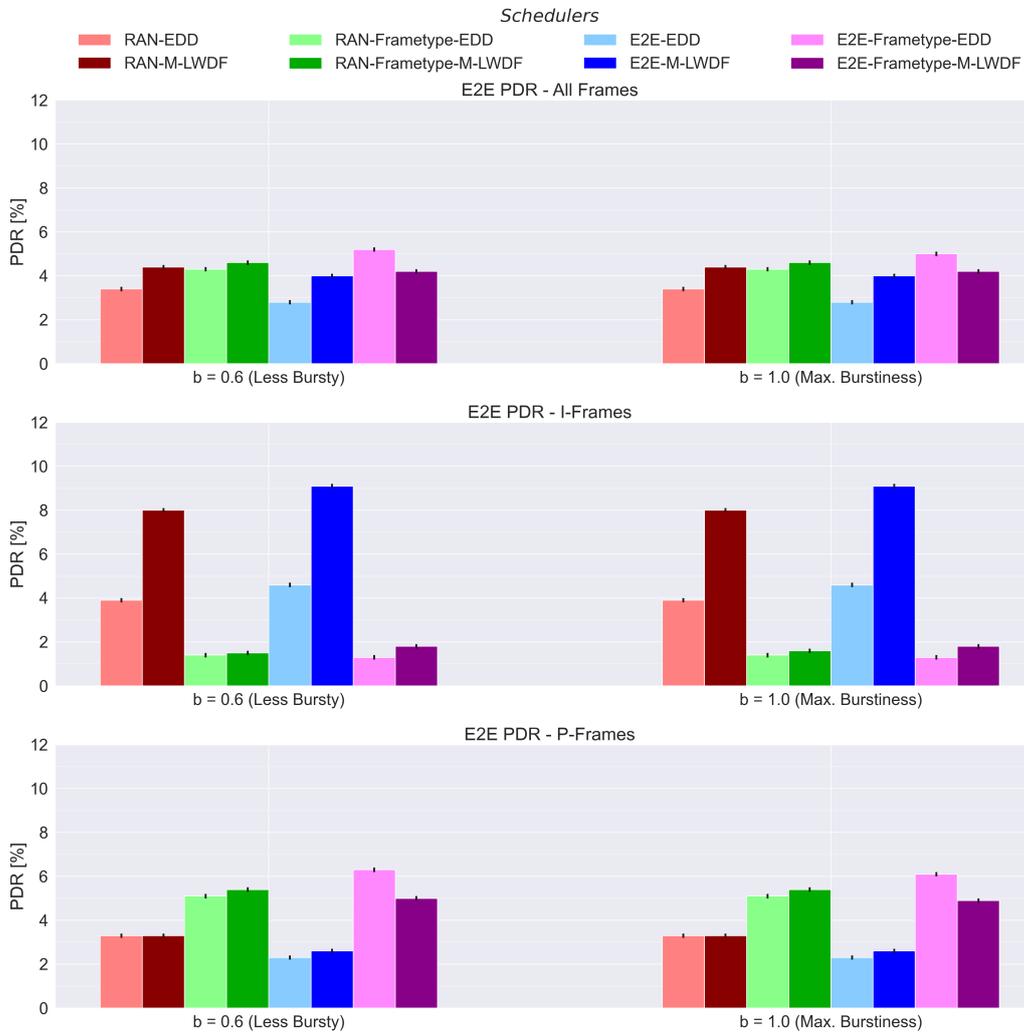


Figure 4.9: PDRs of all schedulers for different encoder burstiness b .

Figures 4.10 and 4.11 show the PDR for the various settings of the number of network hops and background traffic load respectively.

We see from Figure 4.10 that there is not a noticeable difference between going through five or ten network hops, but a massive increase in PDR for the case of fifteen. The cause of this is the propagation delay of 30 ms in the scenario with fifteen hops. It takes up already a significant part of the 50 ms of E2E latency budget, which leads to the observed increase in PDR of up to four times. This implies that the packet dispersion is not affected as much by the total number of hops as such, but rather by more considerably higher propagation delays.

In Figure 4.11, we notice that the background load at each router only influences the packet dispersion and PDR if the background traffic load is at 85%. These results indicate, that only very high levels of background traffic load severely increases the queuing delays for VR packets that it negatively affects the PDRs. With higher background loads, the earlier mentioned caveat of the E2E-Frametype-EDD scheduler is visible again. Since packets of an individual frame are in general even more dispersed, the arrival of I-frame packets is delayed by so much, that more TTIs are used only to transmit the I-frame packets of one UE, at the cost of dropping even more P-frame packets. We observe similar with the E2E-Frametype-M-LWDF scheduler, although to a smaller degree due to its PF component.

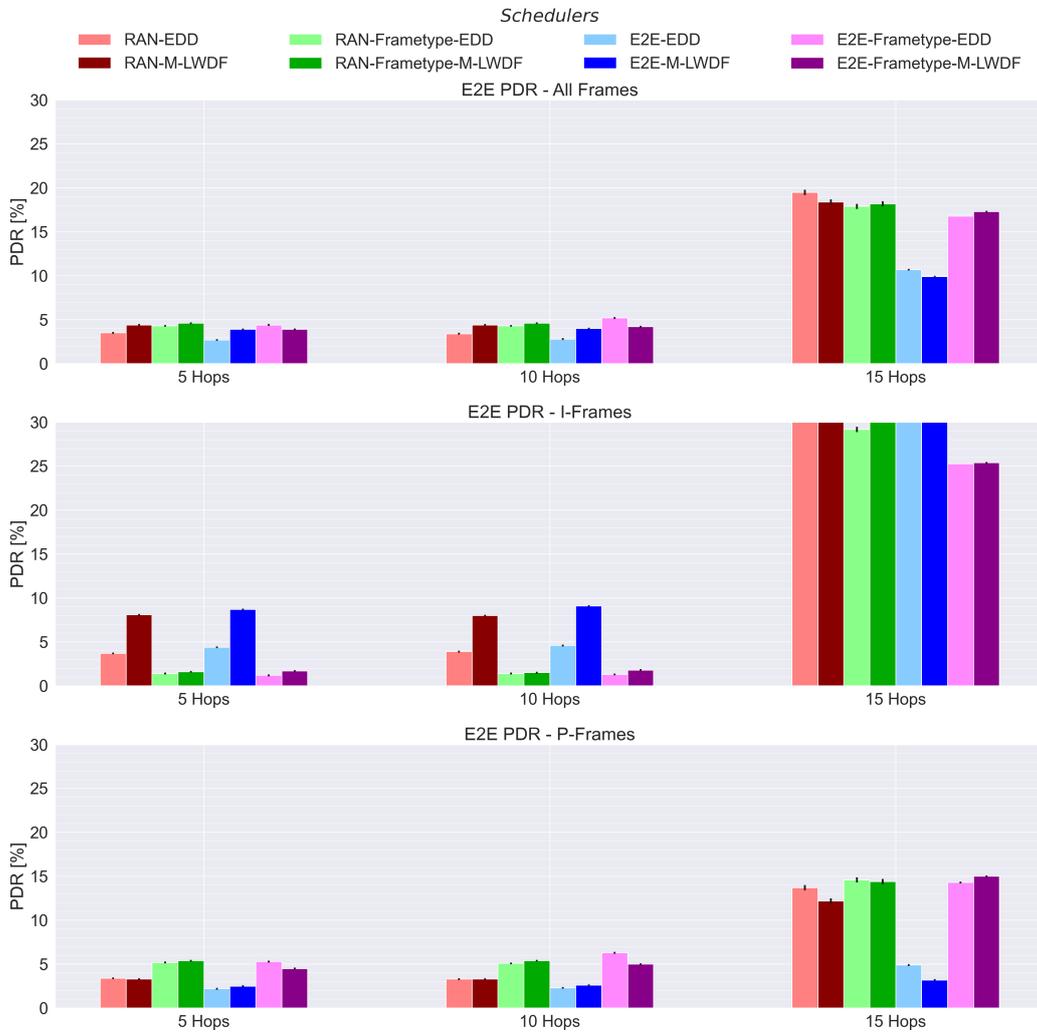


Figure 4.10: PDRs of all schedulers for different numbers of network hops.

Overall, we can conclude that realistic levels of encoder burstiness do not affect the PDR and the main factors for high degrees of packet dispersion which affect the PDR are propagation delays for larger distances between the source and BS and very high traffic loads at network hops. Using cross-layer schedulers with knowledge about the E2E latency budget improves performance, especially in cases where the propagation delay is close to the configured E2E latency budget. This means that although the application might demand a certain latency, in case of a long-distance use case, the application should trade off a bit in terms of perceived end user experience and allow a high enough E2E latency budget such that the application is usable at all.

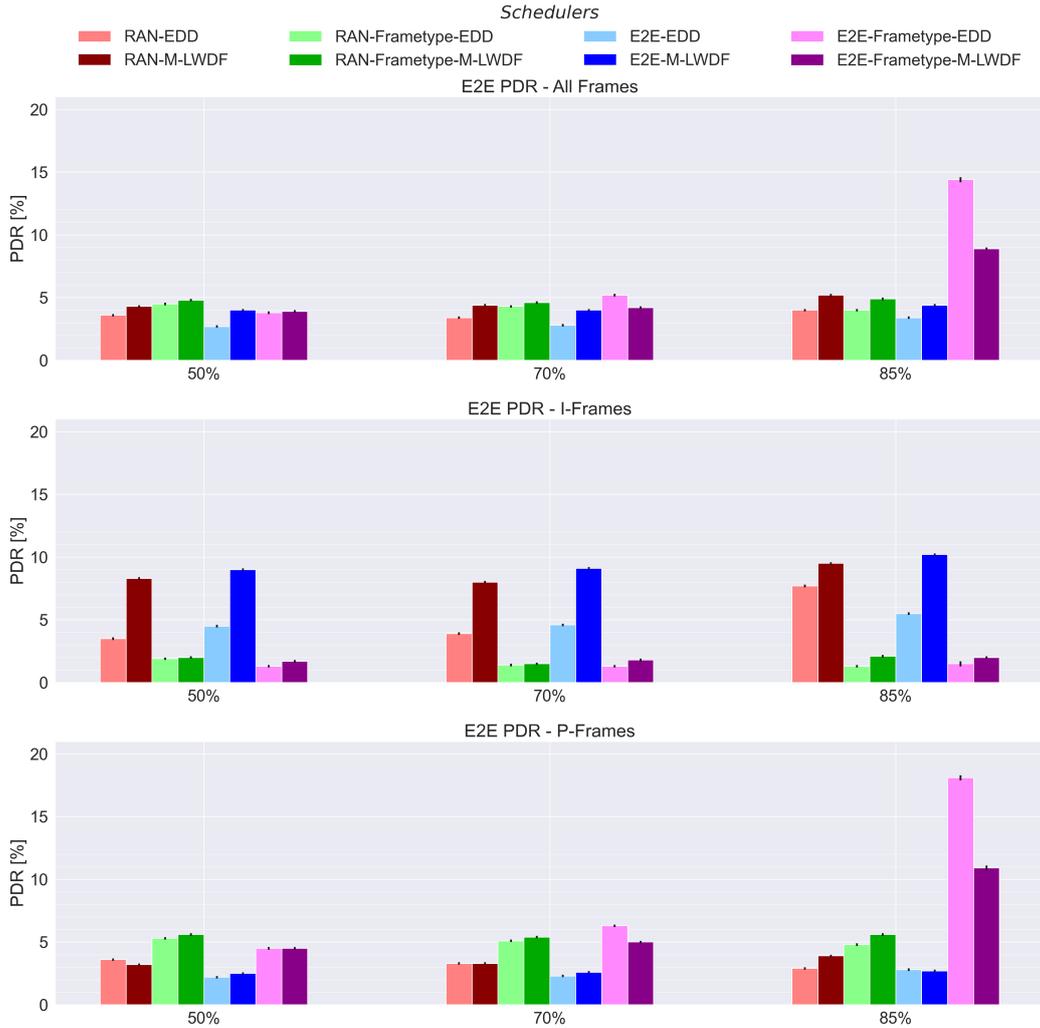


Figure 4.11: PDRs of all schedulers for different background traffic loads at network hops.

4.2.2.4 Synchronisation of Video Traffic Streams

Figure 4.12 shows the difference between having maximally asynchronous and perfectly synchronised video traffic streams among the different involved users. In the latter case, we observe an increase in PDR for all schedulers. The cause for this is the stark increase in the aggregate number of packet arrivals whenever there is an I-frame. With the I-frames now also arriving simultaneously for all UEs, there is no difference in performance from using additional frame type information in the scheduling. The E2E-schedulers are also noticeably outperforming their RAN counterparts in both I- and P-frame PDR. Because we previously tuned the RAN latency budget for the asynchronous scenario, it is now more beneficial to have additional cross-layer information in the form of the E2E latency budget. This exemplifies once again that RAN latency-based scheduling in itself may work fine, but only if the RAN latency budget is well-tuned to the scenario at hand, which in practice is virtually impossible to do.

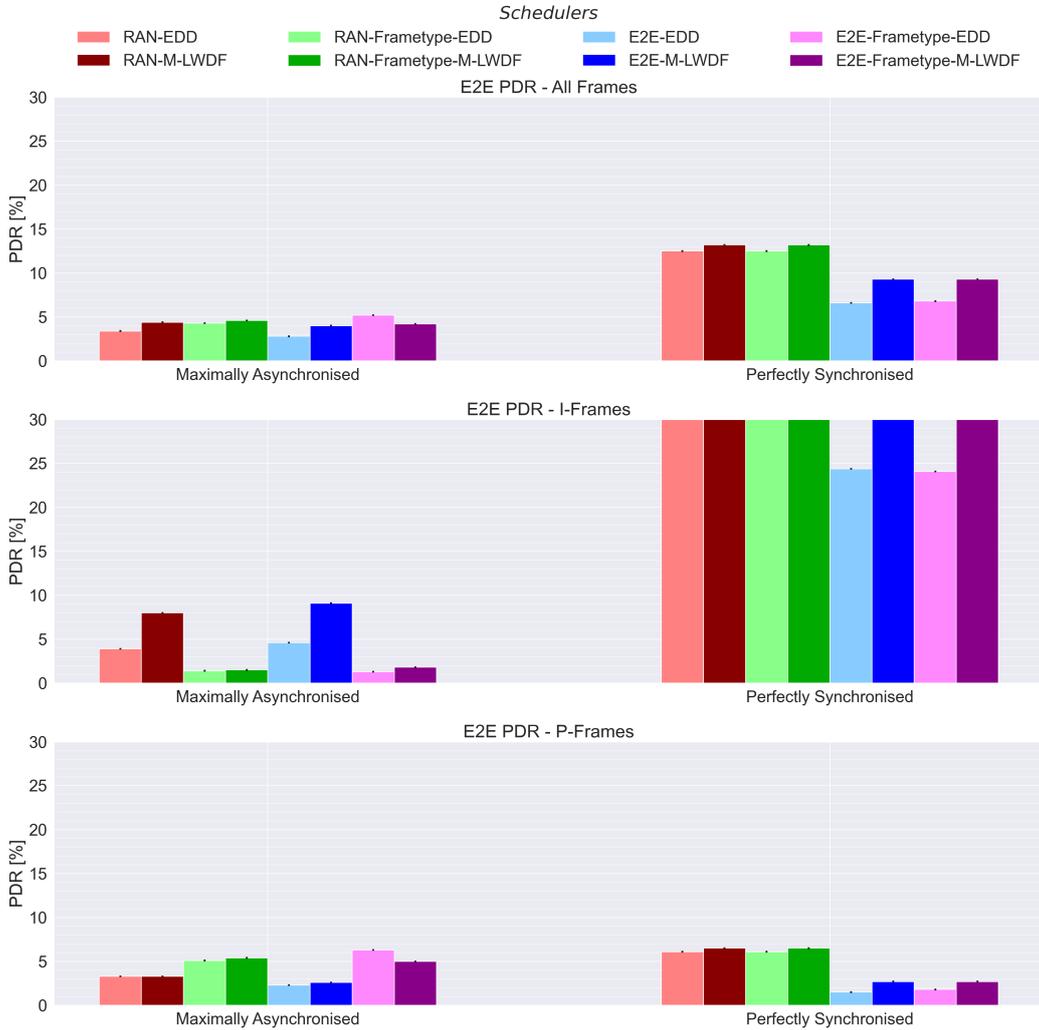


Figure 4.12: PDRs of all schedulers for different video traffic stream synchronisations.

Ultimately, we can conclude from Section 4.2 that in most cases, using E2E latency-based cross-layer schedulers gives a better performance than a well-tuned RAN latency-based schedulers, especially considering that tuning the RAN latency budget is not feasible in practice. The frametype cross-layer schedulers show in almost all cases a significant decrease in I-frame PDR at the cost of higher

P-frame PDR and slightly higher overall PDR. The latter may seem worse from a network QoS perspective, but how this affects the arguably more crucial user-perceived video quality is discussed in Section 4.4.

Further, we notice drawbacks for both the EDD and M-LWDF schedulers in some special cases. The former lacks channel-awareness that can lead to inefficient usage of resources if I-frames are over-prioritised as we saw in the case of the E2E-Frametype-EDD scheduler. On the other hand, without additional I-frame prioritisation, the PF component of the M-LWDF schedulers lead to an undesired under-prioritisation of UEs with many I-frame packets. Therefore the RAN/E2E-M-LWDF scheduler without frame type information show the highest I-frame PDRs. This shows the problem of having to balance the prioritisation of the I-frame packets of one user while at the same time not overly neglecting P-frame packets of other users. In that sense, there is still room for improvement for both the EDD and M-LWDF scheduler. We saw that the RAN load could influence the PDR the most, meaning the video bit rate, channel bandwidth, and number of UEs should be geared to each other as good as possible to reach a specific QoS target. Moreover, the packet dispersion mainly influences the PDR in a scenario where, e.g. there is a high load in network routers, or the E2E latency budget was configured too small for a long distance between source and destination with a large propagation delay.

4.3 Feasible Load and Application Configurations

This section aims to identify what combinations of the number of UEs and application bit rate are supportable under the three different E2E latency budgets, given a 5% upper limit for the PDR.

We consider a subset of the schedulers assessed in Section 4.2. Since the EDD schedulers showed generally lower PDRs than M-LWDF schedulers in the sensitivity analysis, we focus on the EDD scheduler using different types of cross-layer information and compare the RAN-EDD, the RAN-Frametype-EDD, the E2E-EDD, and the E2E-Frametype-EDD scheduler in this section.

4.3.1 Scenarios and Settings

Unlike the previous section’s sensitivity analysis, we assume that the RAN operator has no prior knowledge about underlying networking and propagation delays but only knows the E2E latency budget. Consequently, we tune the RAN latency budget based on worst-case scenarios for the packet dispersion. We saw in the previous section, that the number of network hops is the biggest cause for packet dispersion. Hence, we tune the RAN latency budgets for the E2E latency budgets of 50 and 100 ms based on the scenarios with fifteen network hops. Because satisfying an E2E latency requirement of 25 ms in scenarios with fifteen network hops is physically not possible due to the 30 ms of propagation delay, the RAN latency budget for 25 ms E2E latency budget is tuned based on ten hops instead. The chosen RAN latency budgets are 10 ms for 25 and 50 ms of E2E latency budget and 50 ms for 100 ms E2e latency budget (see Appendix A.3). We simulate and compare results for all possible E2E latency budgets for ten (Figure 4.13) and 15 network hops (Figure 4.14). As the default for the number of UEs was four and for the application bit rate 100 Mbps, we now consider one to eight UEs in single-UE increments and 50 to 150 Mbps in 10 Mbps increments. We set all other system parameters to default as given in Table 4.1.

4.3.2 Results and Observations

In Figures 4.13 and 4.14, the coloured area indicates the feasible combinations of application bit rate and number of UEs for the given scheduler and E2E latency budget.

In Figure 4.13, the areas covered in blue (50 ms E2E latency budget) and green (100 ms E2E latency budget) are very similar, indicating that there are diminishing returns in terms of the capability to support more demanding scenarios by increasing the E2E latency budget. However, much fewer combinations of application bit rate and number of UEs are supported if the E2E latency budget is only 25 ms, as we see with the red areas. We saw from Figure 4.8 that a lower E2E latency affects the PDR much more heavily. Notably, only a 120 Mbps application bit rate is possible even for only one UE. Because higher application bit rates mean higher loads on network routers, we see an increase in PDR due to higher levels of packet dispersion, similar to our observations for increasing the background traffic load in Figure 4.11.

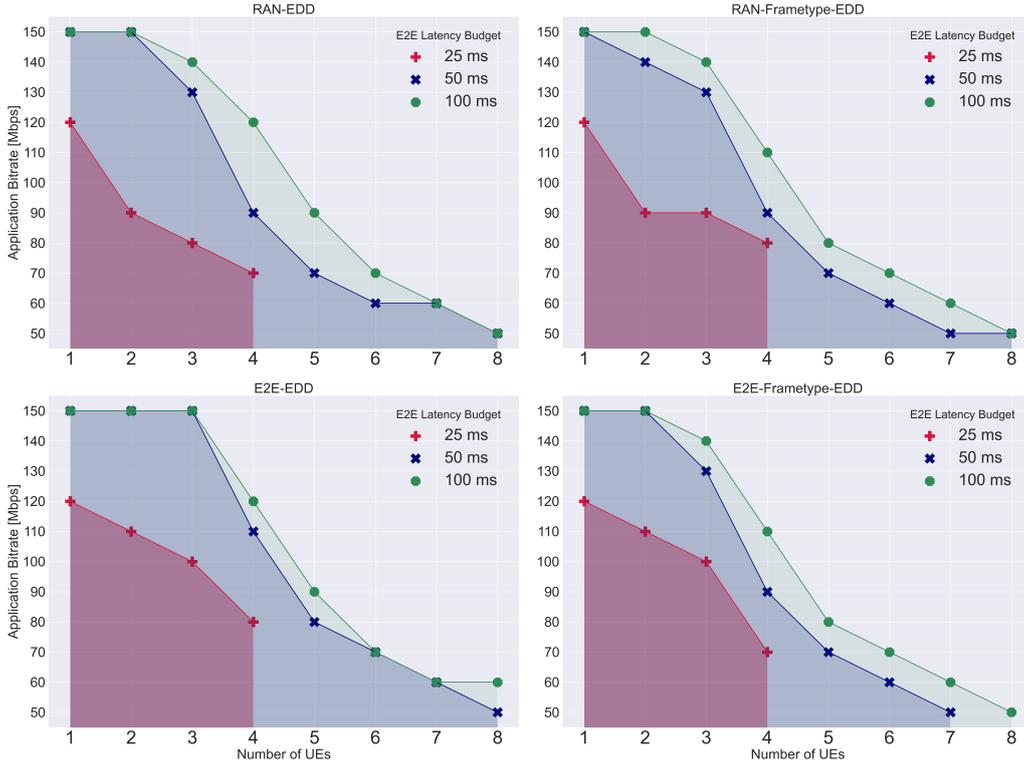


Figure 4.13: Feasible combinations of application bitrate, number of UEs and E2E latency budget for ten network hops for RAN-EDD schedulers (top) and E2E-EDD schedulers (bottom).

Recall that the E2E latency requirement of 25 ms is physically impossible for the scenario with fifteen network hops since the propagation latency alone exceeds the E2E latency budget. Therefore, Figure 4.14 contains no areas shaded in red. The areas in blue for 50 ms E2E latency budget show again the effect of packet dispersion on the performance. Before arriving at the BS, the networking delay is over 30 ms with 15 network hops, which takes up at least two thirds of the total E2E latency budget if the budget is 50 ms. Similar to Figure 4.13 and the case with 25 ms E2E latency budget, bitrates higher than 90 Mbps are not possible to support with only 50 ms E2E latency budget because of the more significant packet dispersion for packet streams with higher bit rate.

Overall, the E2E-EDD scheduler shows the best results in every scenario among the four schedulers. The E2E-Frametype-EDD scheduler has the worst performance for higher numbers of UEs but better performance than the RAN latency-based schedulers when fewer UEs are present. As we saw in Section 4.2.1, more UEs lead to a higher increase in P-frame PDR due to the E2E-Frametype-EDD scheduler overly prioritising I-frame packets. Similar can also be said about the RAN-Frametype-EDD scheduler in Figure 4.13, where it cannot support seven UEs with 60 Mbps application bit

rate and 50 ms E2E latency budget, while the RAN-EDD and E2E-EDD schedulers both can.

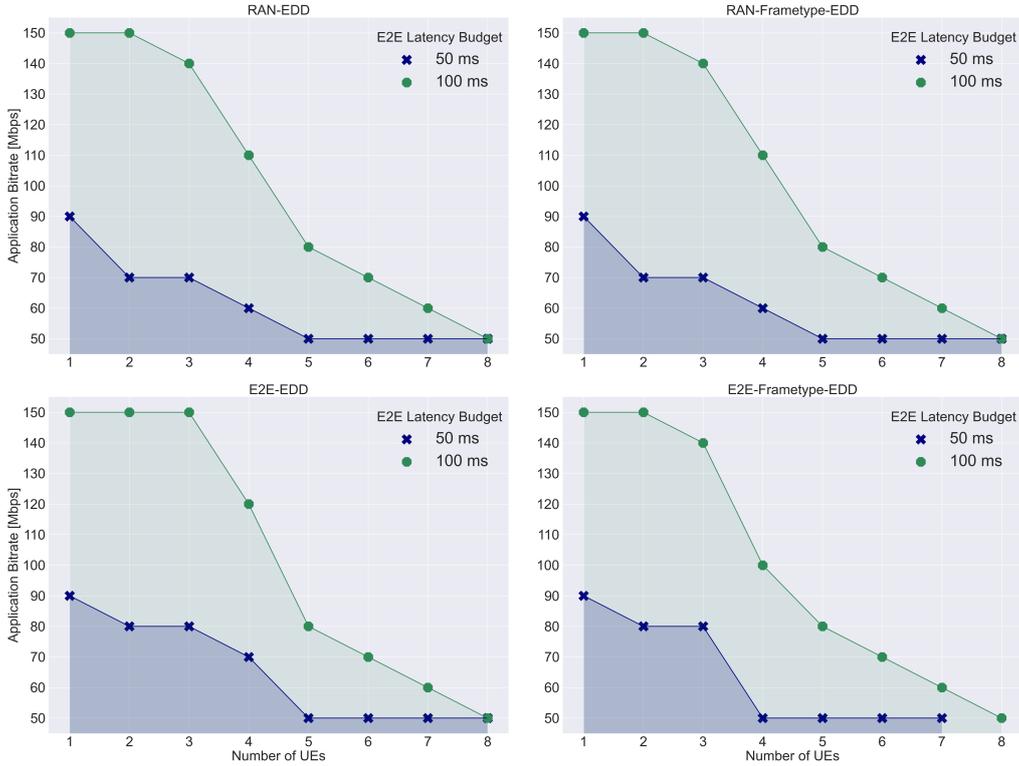


Figure 4.14: Feasible combinations of application bitrate, number of UEs and E2E latency budget for fifteen network hops for RAN-EDD schedulers (top) and E2E-EDD schedulers (bottom).

We can generally increase the set of feasible scenarios by using cross-layer schedulers. Both the RAN-Frametype-EDD and E2E-EDD show a better performance than the non-cross-layer RAN-EDD scheduler in almost all cases. The E2E-Frametype-EDD scheduler is also better if there are fewer UEs present. It is important to mention that the 5% PDR threshold is chosen from a network QoS perspective. Potential scenarios where frametype cross-layer schedulers exhibit PDRs slightly higher than 5% due to the overall net increase caused by higher PDRs for P-frames are not present. How low PDRs for I-frames with overall PDR higher than 5% affect the video quality experienced by users in practice, can again not be expressed in these figures. For this reason, we will assess the simulation results with video quality metrics in the next section to better quantify the effects of cross-layer schedulers.

4.4 Analysing Quality of Experience

Since the PDR as a network QoS metric can only represent the performance from a network efficiency standpoint, we need another way to illustrate the perceived user experience. We will use the PSNR [21] and SSIM [22] as video quality metrics. Both PSNR and SSIM are full-reference metrics, meaning the output video with any quality impairment due to packet losses is directly compared with the original reference video. We discuss the two metrics in more detail below.

4.4.1 Interpreting PSNR and SSIM Values

The PSNR is given in dB, with a higher value indicating a closer resemblance between the original and reconstructed video. Each pixel in a colour image is represented as a combination of three different colours in a certain color space, e.g. with red, green, and blue in the RGB colour model. The PSNR is calculated based on the average Mean Squared Error (MSE), representing the cumulative squared error in all three colour spaces between the received and original image [21]. The colour value of each pixel in an image will change when the image quality degrades due to packet loss. A maximum PSNR (infinity) is only reached if the MSE is zero, meaning the compared images are identical. The overall PSNR value for a video is determined as the average PSNR over all frames. In [35], anything above 30 dB is perceived as very good video quality, with other studies giving lower thresholds until 25 dB to be still acceptable for wireless transmissions [21].

Unlike the PSNR, which calculates absolute errors in an image, the SSIM considers changes in structural information to estimate the image degradation [22] with a value ranging between zero and one, with the latter only achieved if the compared images are identical. The thought behind structural information is that spatially close pixels contain information about the structure of objects in an image due to a strong correlation. An SSIM value of 0.95 is deemed a threshold for good quality with just noticeable differences [22]. The SSIM and PSNR can be unreliable if used to compare videos with different contents or video codecs [21, 44]. We avoid this as we use the same original video as a reference throughout this study.

4.4.2 Analysing PSNR and SSIM Results

Figure 4.15 shows all schedulers' PSNR and SSIM values with the default configuration. When comparing the video quality, higher values are desired, as opposed to lower values when comparing the PDR. Contrary to the PDR results in Figure 4.3, the M-LWDF schedulers perform as good or better than the EDD schedulers. This observation alludes to the fact that EDD schedulers distribute the resources less fairly than M-LWDF schedulers which inherently provide some degree of throughput fairness among users. Better video quality due to enhanced fairness suggests, that the improvements in video quality see diminishing returns for lower PDR, and on the contrary slightly higher PDRs with less variance among users leads to on average better video qualities. This also hints towards linear increases in PDR leading to exponential decreases in PSNR/SSIM.

In Figure 4.16, we consider the same scenario as in Figure 4.15, but depict the PSNR and SSIM per user instead of the average. Here it clearly shows the difference and higher variance in PSNR/SSIM between users for the E2E-EDD schedulers compared to the E2E-M-LWDF scheduler. Due to our implementation, UE 1 receives more resources than the other UEs in case the frame generation times are almost identical and the E2E-EDD scheduler is used as it only considers the E2E deadline of packets that is dictated by the frame generation times. In practice, this is of course an unwanted scenario which could realistically still happen if the frame generation time for different UEs at the source is almost identical, leading to the over-prioritisation of one UE with the E2E-EDD scheduler. This again showcases the drawbacks of the EDD scheduler's lack of channel-awareness.

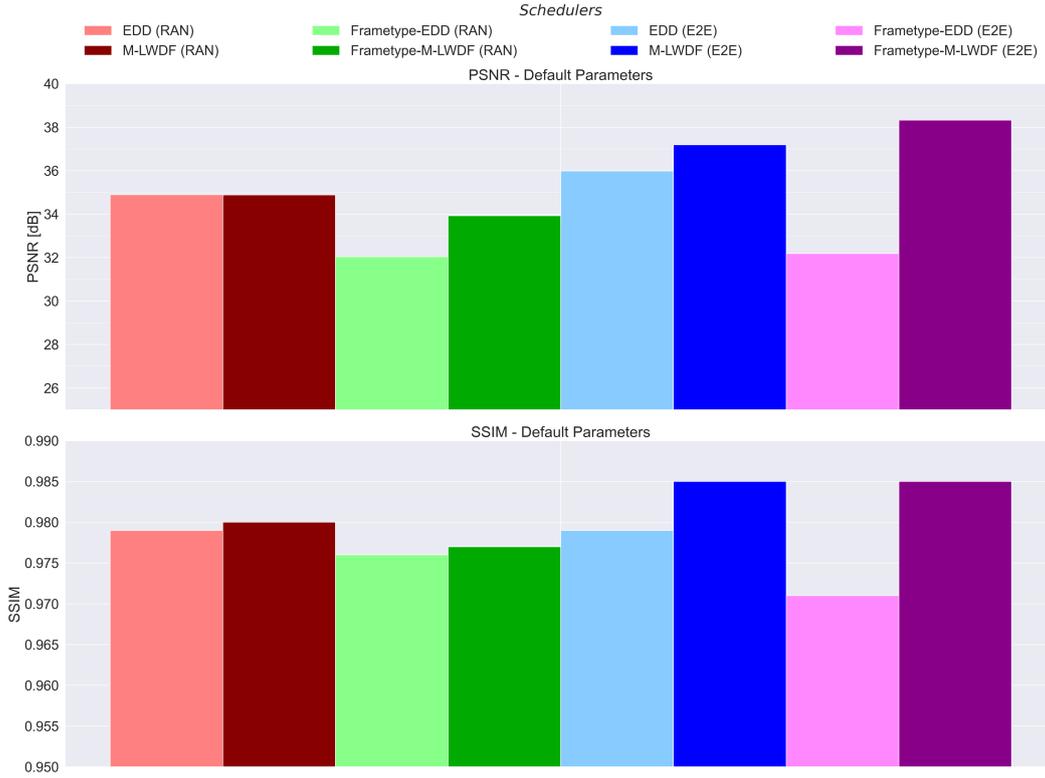


Figure 4.15: PSNR and SSIM for all schedulers with parameters set to default (cf. Figure 4.3).

Next, we observe that most Frametype schedulers perform worse than non-Frametype schedulers, and prioritising I-frame packets over P-frame packets does not convert to better net results for the video quality by default. Similarly to the results in Section 4.2.1, the E2E-Frametype-EDD scheduler shows the worst results out of all schedulers due to its particular way of always prioritising the I-frame of one user, resulting in low resource efficiency. The E2E-Frametype-M-LWDF scheduler performs best although it did not show the lowest PDR in any scenario previously. From this we can conclude that for the best user experience, both types of cross-layer information in combination with a PF component should be used to increase the transmission of important I-frame packets while maintaining good resource efficiency. It is important to note that for all schedulers in the default scenario, the PDR is in a low range below or at worst only slightly above the 5% threshold, which is also reflected in the PSNR/SSIM, with all schedulers achieving high results for PSNR (>32 dB) and SSIM (>0.97).

To better illustrate the gain of using frame type as cross-layer information in the scheduling and the degree of correlation between the perceived (PSNR/SSIM) and the network-level QoS (PDR) metrics, we focus on Figure 4.17. Here, we have plotted the PDR and corresponding PSNR/SSIM of every video file resulting from all simulations in the sensitivity analysis. We represent all UEs from every seed as individual data points for one set of parameters and scheduler and separate them by colour into Frametype (blue) and Non-Frametype (red) schedulers and EDD and M-LWDF schedulers.

As expected, we see the trend that there is a negative correlation between PDR and PSNR, and also PDR and SSIM. Further, the figures show that for the lower regime of PDRs, especially up to 5%, Frametype and Non-Frametype schedulers do not produce very different results in terms of video quality. However, the gain of using Frametype schedulers is noticeable as the PDR increases. We

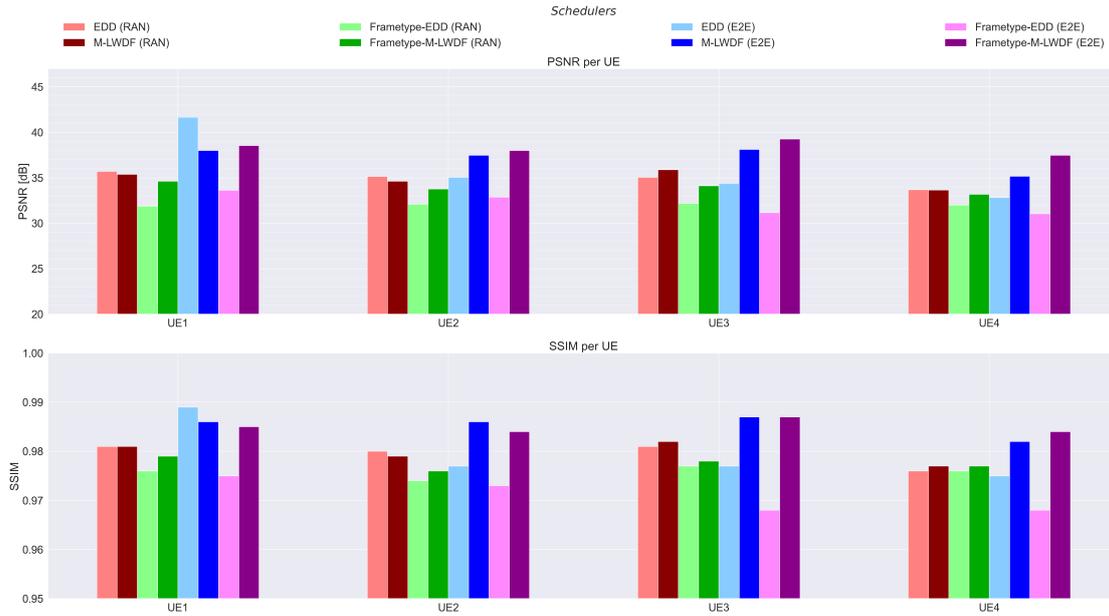


Figure 4.16: PSNR and SSIM for default configuration per UE.

observe in many cases that by using Frametype schedulers, we are still able to achieve high enough PSNR (>25 dB) and SSIM (>0.95) even with PDR as high as 40%. This verdict emphasises the importance of not only looking at network-level QoS metrics, but also at perceived-level metrics such as the PSNR or the SSIM. Thus, we can conclude that using Frametype schedulers will not show any noticeable changes in already good network conditions with low PDR but is more beneficial in scenarios with critical network conditions or too high system load. In these situations, which may occasionally occur even in well-designed networks due to natural variabilities in e.g. the cell load or channel conditions, the impairment of the video quality can be limited successfully by prioritising the radio resources for transmitting I-frame packets.

Overall, from an end user experience perspective, we conclude that the E2E-Frametype-M-LWDF scheduler, which uses both types of cross-layer information and is channel-aware, is the best choice for almost all scenarios.

The full results of the PSNR and SSIM for the scenarios covered in the sensitivity analysis in Section 4.2, corresponding to the PDR results covered in Figures 4.4 to 4.12, can be found in Appendix D.

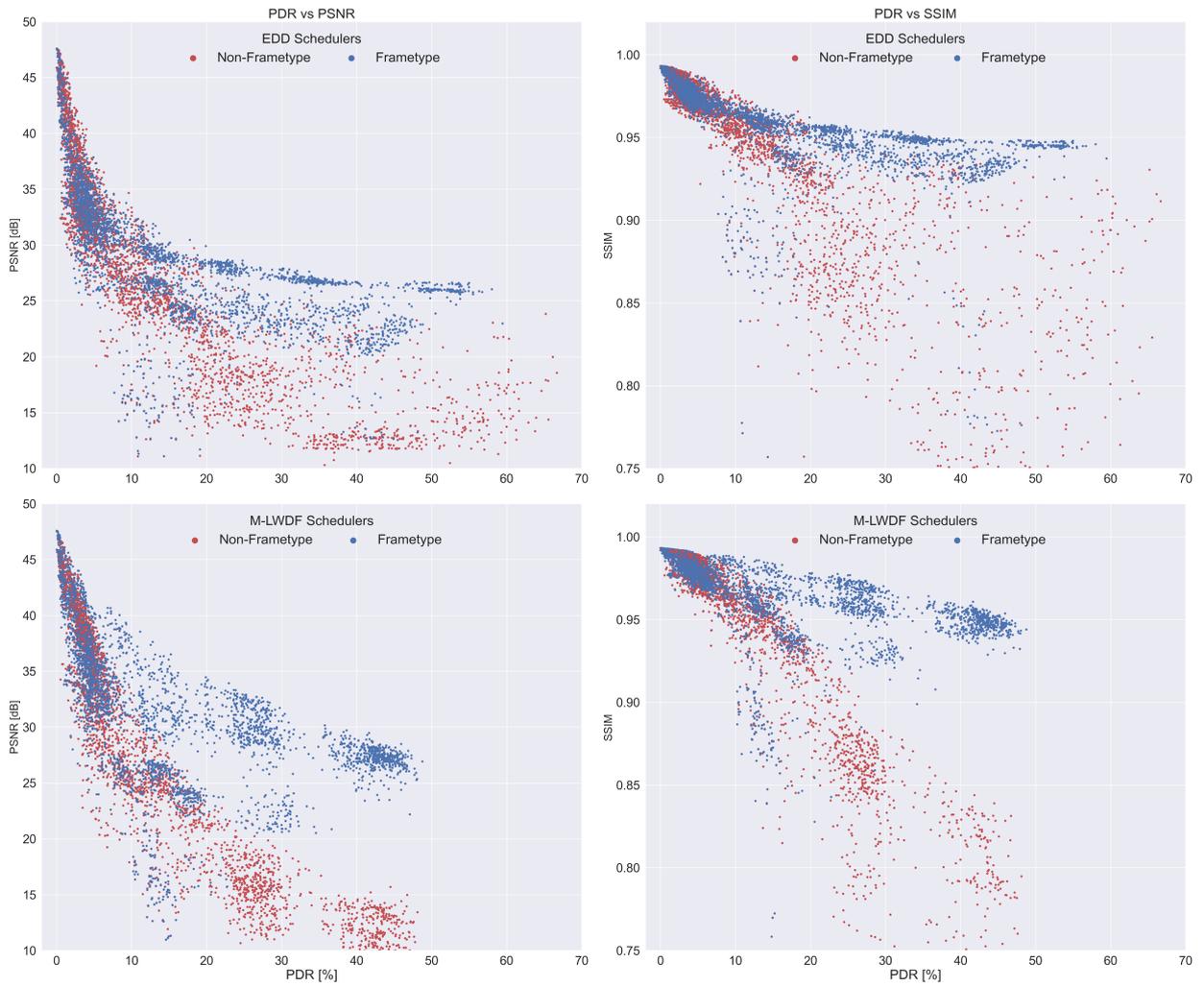


Figure 4.17: PSNR and SSIM for all individual simulations separated for EDD and M-LWDF schedulers as well as Frametype and Non-Frametype schedulers.

Chapter 5

Conclusion

5.1 Conclusion

This section concludes the thesis by summarising the findings and presenting directions for new research in the future.

In this thesis, we modelled and investigated a cross-layer solution for optimising social XR applications over 5G mobile networks. Our focus was on improving the performance of the packet scheduling mechanism in the 5G RAN. In particular, by utilising application-level information in the form of the E2E frame latency budgets and video frame type in the packet scheduling process, our goal was to improve network efficiency and, more importantly, the QoE perceived by end users.

In Chapter 2, we reviewed solutions for video and VR streaming applications that exploit cross-layer information in packet scheduling and identified a lack of studies aimed towards real-time user-interactive VR applications such as social XR. We presented the state of the art research in studying the effects of utilising E2E latency budgets instead of RAN latency budgets for packet schedulers. Further, we showed solutions for video and VR streaming applications that exploit video frame type information to prioritise I-frame packets, which are more important for preserving good QoE. We saw significant improvements in video quality and perceived end user experience in video and VR streaming applications by capitalising on application-level cross-layer information in packet schedulers. We aim to extend the state of the art in cross-layer optimisation for VR delivery by proposing and assessing cross-layer packet schedulers using a realistic model for real-time social XR applications.

Chapter 3 starts with a high-level description of all components of our model for the downlink streams of social XR applications. Then we discussed in detail the modelling of the system starting with the physical setting, the user behaviour and propagation environment as well as the specifications and functionalities of the RAN. For the latter, we exploited an existing simulation framework, which we adapted in order to focus on our objective of optimising the packet scheduling in the RAN. We extended the framework with our proposed cross-layer schedulers and non-cross-layer benchmark schedulers. Then, we used a recorded video in an indoor office setting to create traffic streams with realistic characteristics, including the effects of packet dispersion imposed by video encoders and multi-hop transmission over the internet. We developed a tool for creating that packet dispersion by simulating a series of routers that introduce random queuing delays for the packets travelling through.

In Chapter 4, we assessed through an extensive sensitivity analysis the impact of our system parameters and cross-layer scheduling approach in the context of social XR applications. We compared

the performance of the benchmark EDD and M-LWDF schedulers and the cross-layer schedulers. The results showed significant performance gains from using the E2E latency budget in the scheduling decisions since RAN latency-based schedulers only perform well if the RAN latency budget is well-tuned to the specifics of the scenario at hand, which in practice is virtually impossible to do. Using frametype schedulers, which prioritise packets belonging to an I-frame, decreased the I-frame PDR (as intended) at the cost of increasing the overall PDR, due to many more dropped P-frame packets, which resulted in an unfortunate net PDR increase as a GoP comprises more P-frames than I-frames. We observed one further drawback of using the E2E-Frametype-EDD scheduler. Since the lack of channel-awareness leads to an over prioritisation of I-frame packets and hence in a more significant increase in dropped P-frame packets, it performs worse than the E2E-Frametype-M-LWDF scheduler.

Next, we showed the impact of different scenario parameters on the performance. We noticed significant increases in PDR when raising the effective load to the BS. The relative performance of the different schedulers was not affected but rather key trends we observed before were more prominent, like the frametype-schedulers showing even higher overall PDRs but keeping very low I-frame PDRs in high cell load scenarios. The packet dispersion imposed by the encoder and internet influenced the PDR only in extreme edge cases, for example if total packet arrivals on network hops was nearing 100% of its traffic handling bit rate. In that case the individual traffic streams were Perfectly synchronised traffic streams, where the I-frames of individual UEs arrive at the same time, not only leads to a worse performance due to higher aggregate packet arrivals whenever there is an I-frame, but also means that using frametype schedulers has no effect on the PDR.

Afterwards, we studied the range of feasible combinations of the number of UEs and the application bit rate for different E2E latency budgets and network hops for all variations of the EDD scheduler. We gained insights into the scenarios the (non-)cross-layer EDD schedulers can support for different configurations. In addition, we saw an increase in supportable scenarios by using cross-layer information compared to having a well-tuned non-cross-layer scheduler. So can the E2E-EDD scheduler in the scenario with ten network hops and an E2E latency budget of 50 ms, support three UEs with an application bitrate of 150 Mbps, while the RAN-EDD scheduler is only able to support an application bitrate of 130 Mbps. With 25 ms E2E latency budget, the RAN-EDD scheduler can support two UEs with an application bit rate of 90 Mbps, which is increased by an additional UE, if the RAN-Frametype-EDD scheduler is used instead. Further, having stricter E2E latency budgets implies that high application bit rates cannot be supported due to networking delay caused by more significant levels of packet dispersion taking up too much of the E2E latency budget.

Lastly, we evaluated the scenarios assessed in the sensitivity analysis from the QoE perspective using the PSNR and SSIM as objective video quality metrics. The results showed a negative correlation between the PDR and PSNR, and the PDR and SSIM, since more dropped packets (higher PDR) lead to more decoding errors in the received video (lower PSNR/SSIM). We observed a slight increase in PSNR and SSIM from using E2E latency-based schedulers compared to RAN latency-based schedulers. This is similar to the previous results, where E2E latency-based schedulers also produced lower PDRs. Notably, the benefit of using the frame type as cross-layer information for the video quality was more significant. While under low PDRs, e.g. due to low cell load, there is not much gain to be had from cross-layer schedulers, whereas in more critical scenarios and PDR increasing to undesired levels, e.g. due to high cell load, cross-layer schedulers using the frame type can still deliver satisfactory results. Further, channel-aware schedulers produced better results, so we conclude that the E2E-Frametype-M-LWDF scheduler is the best scheduler from a QoE perspective. Ultimately, the observations about the QoE are in contrast to the results earlier for the PDR, where using frametype schedulers make things worse from a network perspective. Notably, we saw that using frametype schedulers an acceptable PSNR of 25 dB or SSIM of 0.95 could still be reached

with PDRs up to 50%, while using non-frametype schedulers the PSNR and SSIM would drop down to 15 dB and 0.75 respectively. These results imply huge benefits from implementing cross-layer schedulers in practice.

5.2 Future Work

We recommend the following points for future research.

Examine the performance of cross-layer schedulers while including elements of the 5G RAN disregarded in this study, such as massive MIMO, mmWave frequency bands, and beamforming. We showed the benefits for the end user experience by using cross-layer schedulers. The next step should be to conduct studies similar to Section 4.3 while considering all aspects of 5G that would be used in practical deployment. A very stringent requirement of social XR is the low latency requirement for enabling user interactivity. In 5G, flexible slots and mini-slots could facilitate and allow lower transmission latencies especially by using cross-layer schedulers for high priority I-frame packets nearing their latency deadlines. Thus, future research should aim to consider all aspects of 5G to produce guidelines for supported social XR scenarios under different application and network configurations.

Investigate the benefits of using additional information for scheduling decisions. In Chapter 4, we observed disadvantages for both the EDD and M-LWDF schedulers. The former lacks channel-awareness and thus produces high variance in terms of QoE between end users. Whereas with the PF component, the M-LWDF scheduler does not distribute enough resources for UEs with more packets (I-frames) in their buffer. These results suggest developing and assessing a new packet scheduler that utilises cross-layer information in combination with, e.g. knowledge about the buffer status of each UE. We concluded that using the E2E-Frametype-M-LWDF scheduler resulted in most scenarios in the best perceived QoE, even though it did not have the lowest overall PDR or I-frame PDR. Since I-frames comprise more packets than P-frames, by also considering the total amount of packets left in the UEs buffer in the scheduling decision, the I-frame PDR could be further decreased, while keeping the increase in P-frame PDR still within reasonable range due to the PF component. This change could mean a further improvement in resource efficiency and video quality.

Assess the perceived QoE for different application configurations with subjective metrics and train machine learning models for adaptive streaming. We previously showed in Chapter 4 the impact of different configurations on the PDR and video quality. We also mentioned that QoE metrics can only accurately represent the perceived video quality when using the same reference video and these results are only assessing the performance from objective perspectives. It is important to also consider the subjective perception of end users' for different application configurations. Therefore, future studies should look into, for example, how end users perceive higher E2E latencies with higher application bit rates versus lower E2E latencies but with lower application bit rates in social XR applications. Such insights can motivate the development of an adaptation mechanism which e.g. adapts the application requirements based on the physical distance between the two parties of the social XR session and their mobile network conditions. In this context we also recommend to assess and identify new thresholds for video quality metrics like the PSNR and SSIM for different types of video content. By collecting data for various social XR scenarios beyond the indoor conference setting considered in this thesis, one could train machine learning models to predict the perceived end user experience from factors, such as the video content, bit rate, and E2E latency requirements. This can help the system to quickly adapt to, e.g. bad radio channel conditions, and change parameters to maintain adequate experiences for end users.

Bibliography

- [1] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters, and F. De Turck, "Toward truly immersive holographic-type communication: Challenges and solutions," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 93–99, 2020.
- [2] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7573–7586, 2019.
- [3] X. Zhang, G. Cheung, Y. Zhao, P. Le Callet, C. Lin, and J. Z. G. Tan, "Graph Learning Based Head Movement Prediction for Interactive 360 Video Streaming," *IEEE Transactions on Image Processing*, vol. 30, pp. 4622–4636, 2021.
- [4] S. Dijkstra-Soudarissanane, S. Gunkel, A. Gabriel, L. Feroselle, F. t. Haar, and O. Niamut, "XR Carousel: A Visualization Tool For Volumetric Video," in *Proceedings of the 12th ACM Multimedia Systems Conference*, 2021, pp. 339–343.
- [5] 3GPP, *TR 26.928 - Extended Reality (XR) in 5G*, 2020.
- [6] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "Vr is on the edge: How to deliver 360 videos in mobile networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, 2017, pp. 30–35.
- [7] F. Chiariotti, "A survey on 360-degree video: Coding, quality of experience and streaming," *Computer Communications*, vol. 177, pp. 133–155, 2021.
- [8] J. van der Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. De Turck, "Optimizing adaptive tile-based virtual reality video streaming," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 381–387.
- [9] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and VR experiences," *IEEE Transactions on Multimedia*, vol. 23, pp. 716–731, 2020.
- [10] T. T. Le, D. Van Nguyen, and E.-S. Ryu, "Computing offloading over mmWave for mobile VR: Make 360 video streaming alive," *IEEE Access*, vol. 6, pp. 66 576–66 589, 2018.
- [11] M. Huang and X. Zhang, "Mac scheduling for multiuser wireless virtual reality in 5g mimo-ofdm systems," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.
- [12] F. Afroz, K. Sandrasegaran, and P. Ghosal, "Performance analysis of PF, M-LWDF and EXP/PF packet scheduling algorithms in 3GPP LTE downlink," in *2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)*. IEEE, 2014, pp. 87–92.

- [13] W. Kim, H. Joo, K. J. An, I. Lee, and H. Song, "Urgency-based packet scheduling and routing algorithms for delay-sensitive data over MANETs," *Wireless networks*, vol. 19, no. 7, pp. 1595–1609, 2013.
- [14] J. Morais, S. Braam, R. Litjens, S. Kizhakkekundil, and H. van Den Berg, "Performance Modelling and Assessment for Social VR Conference Applications in 5G Radio Networks," in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2021, pp. 225–232.
- [15] H. Kalva, "The H.264 Video Coding Standard," *IEEE MultiMedia*, vol. 13, no. 4, pp. 86–90, 2006.
- [16] P. Orosz, T. Skopkó, and P. Varga, "Towards estimating video QoE based on frame loss statistics of the video streams," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1282–1285.
- [17] H. Yao, R. Ni, and Y. Zhao, "Double compression detection for H.264 videos with adaptive GOP structure," *Multimedia Tools and Applications*, vol. 79, pp. 5789–5806, 2019.
- [18] Bitmovin, *2019 Video Developer Report*, 2019. [Online]. Available: <https://bitmovin.com/bitmovin-2019-video-developer-report-av1-codec-ai-machine-learning-low-latency>
- [19] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [20] M. I. Elhadad, E.-S. M. El-Rabaie, and M. Abd-Elnaby, "Resource allocation for Real-Time services using Earliest Due Date mechanism in LTE networks," in *2016 Fourth International Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC)*, 2016, pp. 9–12.
- [21] Q. Huynh-Thu and M. Ghanbari, "The accuracy of PSNR in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.
- [22] J. R. Flynn, S. Ward, J. Abich, and D. Poole, "Image quality assessment using the SSIM and the just noticeable difference paradigm," in *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 2013, pp. 23–30.
- [23] J. She, F. Hou, and P.-H. Ho, "An application-driven mac-layer buffer management with active dropping for real-time video streaming in 802.16 networks," in *21st International Conference on Advanced Information Networking and Applications (AINA'07)*. IEEE, 2007, pp. 451–458.
- [24] J. She, F. Hou, B. Shihada, and P.-H. Ho, "Mac-layer active dropping for real-time video streaming in 4g access networks," *IEEE systems Journal*, vol. 4, no. 4, pp. 561–572, 2010.
- [25] G. Xie, M. Swamy, and M. O. Ahmad, "Joint optimal multipath routing and rate control for multidescription coded video streaming in ad hoc networks," *IEEE transactions on multimedia*, vol. 10, no. 8, pp. 1687–1697, 2008.
- [26] Y. Zhang, Y. Zhang, S. Qin, B. Li, and Z. He, "Delay-bounded priority-driven resource allocation for video transmission over multihop networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1184–1196, 2014.
- [27] S. H. Kang and A. Zakhori, "Packet scheduling algorithm for wireless video streaming," in *International Packet Video Workshop*, vol. 2002, 2002.

- [28] P. Buccioli, G. Davini, E. Masala, E. Filippi, and J. C. De Martin, “Cross-layer perceptual arq for h. 264 video streaming over 802.11 wireless networks,” in *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, vol. 5. IEEE, 2004, pp. 3027–3031.
- [29] O. Nawaz, T. N. Minnas, and M. Fiedler, “Optimal mtu for realtime video broadcast with packet loss—a qoe perspective,” in *The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014)*. IEEE, 2014, pp. 396–401.
- [30] S. Paluri, K. K. Kambhatla, M. Medley, J. D. Matyjas, and S. Kumar, “Priority-aware joint packet fragmentation and error protection scheme for H.264 video over wireless channels,” in *2015 IEEE international symposium on multimedia (ISM)*. IEEE, 2015, pp. 95–100.
- [31] T. Ojanperä, M. Uitto, and J. Vehkaperä, “QoE-based management of medical video transmission in wireless networks,” in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–6.
- [32] P. Seeling and M. Reisslein, “Video traffic characteristics of modern encoding standards: H.264 /AVC with SVC and MVC extensions and H.265/HEVC,” *The Scientific World Journal*, vol. 2014, 2014.
- [33] Fraunhofer. (2020) Quadriga - The next generation radio channel model. [Online]. Available: <https://www.quadriga-channel-model.de/docs>
- [34] J. Morais, “Performance Modelling for Social VR Conference Applications in Beyond-5G Radio Access Networks,” Feb 2021. [Online]. Available: <https://fenix.tecnico.ulisboa.pt/cursos/meec/dissertacao/1128253548922237>
- [35] K. Bouraqla, E. Sabir, M. Sadik, and L. Ladid, “Quality of experience for streaming services: measurements, challenges and insights,” *IEEE Access*, vol. 8, pp. 13 341–13 361, 2020.
- [36] (2022) GStreamer Documentation. [Online]. Available: <https://gstreamer.freedesktop.org/documentation/>
- [37] (2022) Wireshark Documentation. [Online]. Available: <https://www.wireshark.org/docs>
- [38] (2022) tcpdump Documentation. [Online]. Available: <https://www.tcpdump.org/index.html#documentation>
- [39] (2022) Windows tracert Documentation. [Online]. Available: [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb491018\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-xp/bb491018(v=technet.10))
- [40] (2022) AMS-IX Topology. [Online]. Available: <https://www.ams-ix.net/ams/documentation/ams-ix-topology>
- [41] W. John and S. Tafvelin, “Analysis of internet backbone traffic and header anomalies observed,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 111–116.
- [42] R. Sinha, C. Papadopoulos, and J. Heidemann, “Internet packet size distributions: Some observations,” *USC/Information Sciences Institute, Tech. Rep. ISI-TR-2007-643*, pp. 1536–1276, 2007.
- [43] A. O. Adeyemi-Ejeye, M. Alreshoodi, L. Al-Jobouri, M. Fleury, and J. Woods, “Packet loss visibility across SD, HD, 3D, and UHD video streams,” *Journal of Visual Communication and Image Representation*, vol. 45, pp. 95–106, 2017.

- [44] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of PSNR in image/video quality assessment,” *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.
- [45] S. Pratschner, B. Tahir, L. Marijanovic, M. Mussbah, K. Kirev, R. Nissel, S. Schwarz, and M. Rupp, “Versatile mobile communications simulation: The Vienna 5G link level simulator,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–17, 2018.

Appendix A

BLER Curves

Figure A.1 shows the BLER curves for the different MCS, with the BLER probability of 10% marked for each curve. The curves are fitted to simulations based on the equations for each MCS curve that can be found in [34].

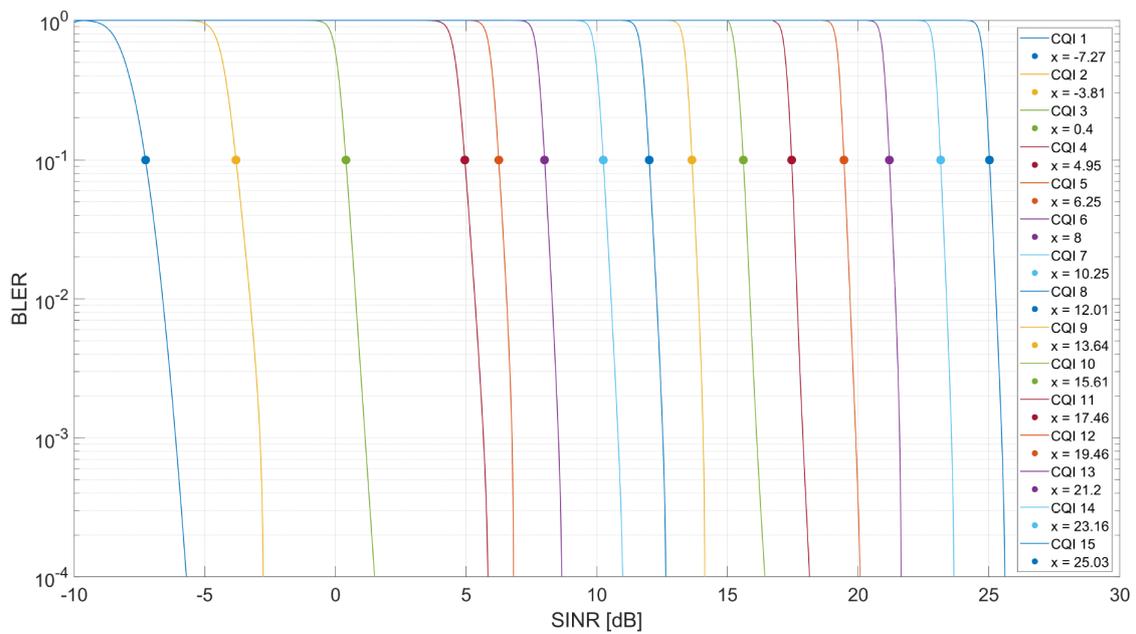


Figure A.1: BLER curves simulated with Vienna Link-Level Simulator [45].

Appendix B

Choosing Number of Hops with Traceroute

In Section 3.3.2, we chose different numbers of network hops for the three considered scenarios. Figures B.1 to B.4 show the traceroutes from the local PC (in Delft) to various destinations for which we looked up random IP addresses. The goal was to have a rough understanding of the different numbers of internet hops for destinations with different geographic distances.

In each traceroute, we can see that the first few hops are always through the same IP addresses. These are the IP addresses of local routers and of the internet service provider. Omitting these, we see that the number for the next few hops up to the destinations does not always correlate with the actual distance. So is New York (Figure B.3) reached in as many hops as Berlin (Figure B.2), which is only a few hops more than to Amsterdam (Figure B.1). This is most likely due to the fact, that New York is very close to the Atlantic ocean, with the whole travel across the ocean cables only consisting of one additional hop. In Figure B.4, we see that for a long-distance connection to Los Angeles, that most likely goes through the network in the United States, there will be many more hops. Hence, for simplicity, we chose the numbers of network hops within realistic ranges, without making too many complicated assumptions. Therefore, the three modelled scenarios in Section 3.3.2 are five, ten, and fifteen network hops, with the important addition that the propagation delay is also considered.

```
C:\WINDOWS\system32>tracert 191.96.168.135

Tracing route to 191.96.168.135 over a maximum of 30 hops

  0  0 ms  0 ms  0 ms  192.168.1.1
  1  112 ms  100 ms  112 ms  1-104-21-31.ftth.glasoperator.nl [31.21.104.1]
  2  14 ms  15 ms  13 ms  10.10.10.253
  3  16 ms  14 ms  14 ms  10.10.12.53
  4  15 ms  14 ms  15 ms  217.161.69.141
  5  14 ms  14 ms  27 ms  ae5-PCR1.adr.cw.net [195.2.20.86]
  6  15 ms  15 ms  15 ms  be1273.rcr22.ams05.atlas.cogentco.com [130.117.14.173]
  7  15 ms  15 ms  15 ms  be3499.ccr41.ams03.atlas.cogentco.com [154.54.60.21]
  8  18 ms  25 ms  15 ms  be3433.rcr21.ams06.atlas.cogentco.com [154.54.58.202]
  9  15 ms  15 ms  15 ms  191.96.168.135

Trace complete.
```

Figure B.1: Traceroute between local PC and IP address in Amsterdam.

```

C:\WINDOWS\system32>tracert 154.13.1.224

Tracing route to 154.13.1.224 over a maximum of 30 hops

  0  0 ms  0 ms  0 ms  192.168.1.1
  1  14 ms  14 ms  13 ms  1-104-21-31.ftth.glasoperator.nl [31.21.104.1]
  2  14 ms  13 ms  14 ms  10.10.10.253
  3  14 ms  14 ms  14 ms  10.10.12.53
  4  14 ms  15 ms  15 ms  217.161.69.141
  5  14 ms  14 ms  14 ms  ae5-pcr1.adr.cw.net [195.2.20.86]
  6  16 ms  16 ms  15 ms  be1273.rcr22.ams05.atlas.cogentco.com [130.117.14.173]
  7  16 ms  15 ms  16 ms  be3500.ccr42.ams03.atlas.cogentco.com [154.54.60.25]
  8  26 ms  27 ms  26 ms  be2816.ccr42.ham01.atlas.cogentco.com [154.54.38.210]
  9  33 ms  31 ms  30 ms  be2174.rcr21.ber01.atlas.cogentco.com [130.117.51.218]
 10  30 ms  30 ms  30 ms  be3724.rcr51.ber02.atlas.cogentco.com [154.54.58.141]
 11  31 ms  30 ms  *      te0-0-2-0.nr12.b021787-0.ber02.atlas.cogentco.com [154.25.8.130]
 12  29 ms  30 ms  30 ms  154.13.1.224

Trace complete.

```

Figure B.2: Traceroute between local PC and IP address in Berlin.

```

Trace complete.

C:\WINDOWS\system32>tracert 191.96.150.167

Tracing route to 191.96.150.167 over a maximum of 30 hops

  0  0 ms  0 ms  0 ms  192.168.1.1
  1  14 ms  15 ms  17 ms  1-104-21-31.ftth.glasoperator.nl [31.21.104.1]
  2  13 ms  13 ms  14 ms  10.10.10.253
  3  13 ms  13 ms  13 ms  10.10.12.53
  4  15 ms  14 ms  16 ms  217.161.69.141
  5  15 ms  17 ms  14 ms  ae5-pcr1.adr.cw.net [195.2.20.86]
  6  15 ms  15 ms  18 ms  be1273.rcr22.ams05.atlas.cogentco.com [130.117.14.173]
  7  *      15 ms  15 ms  be3500.ccr42.ams03.atlas.cogentco.com [154.54.60.25]
  8  90 ms  90 ms  92 ms  be12488.ccr42.lon13.atlas.cogentco.com [130.117.51.41]
  9  87 ms  88 ms  130 ms  be2490.ccr42.jfk02.atlas.cogentco.com [154.54.42.85]
 10  90 ms  91 ms  174 ms  be2897.rcr24.jfk01.atlas.cogentco.com [154.54.84.214]
 11  89 ms  88 ms  88 ms  be2804.rcr21.b001362-2.jfk01.atlas.cogentco.com [154.54.80.6]
 12  89 ms  89 ms  89 ms  191.96.150.167

Trace complete.

```

Figure B.3: Traceroute between local PC and IP address in New York.

```

C:\WINDOWS\system32>tracert 84.17.45.104

Tracing route to unn-84-17-45-104.cdn77.com [84.17.45.104]
over a maximum of 30 hops:

  0  4 ms   2 ms   2 ms  192.168.1.1
  1  14 ms  14 ms  13 ms  1-104-21-31.ftth.glasoperator.nl [31.21.104.1]
  2  17 ms  13 ms  13 ms  10.10.10.253
  3  13 ms  13 ms  13 ms  10.10.12.53
  4  21 ms  15 ms  15 ms  217.161.69.141
  5  89 ms  88 ms  91 ms  ae20-xcr2.nyk.cw.net [195.2.8.89]
  6  89 ms  89 ms  90 ms  be-211-pe02.111eighthave.ny.ibone.comcast.net [50.248.117.121]
  7  90 ms  90 ms  90 ms  be-3102-cs01.newyork.ny.ibone.comcast.net [96.110.38.225]
  8  90 ms  89 ms  96 ms  be-1112-cr12.newyork.ny.ibone.comcast.net [96.110.35.130]
  9  90 ms  90 ms  90 ms  be-301-cr11.newark.nj.ibone.comcast.net [96.110.36.145]
 10  90 ms  91 ms  90 ms  be-1211-cs02.newark.nj.ibone.comcast.net [96.110.35.69]
 11  91 ms  91 ms  91 ms  be-1212-cr12.newark.nj.ibone.comcast.net [96.110.35.86]
 12  95 ms  95 ms  98 ms  be-302-cr13.ashburn.va.ibone.comcast.net [96.110.36.117]
 13  94 ms  94 ms  94 ms  be-1213-cs02.ashburn.va.ibone.comcast.net [96.110.34.149]
 14  95 ms  96 ms  95 ms  be-1211-cr11.ashburn.va.ibone.comcast.net [96.110.32.110]
 15 120 ms 119 ms 119 ms  be-301-cr13.doraville.ga.ibone.comcast.net [96.110.32.1]
 16 120 ms 120 ms 120 ms  be-1313-cs03.doraville.ga.ibone.comcast.net [96.110.34.201]
 17 120 ms 120 ms *
 18 121 ms 120 ms 120 ms  be-1311-cr11.doraville.ga.ibone.comcast.net [96.110.34.170]
 19 121 ms 120 ms 120 ms  be-301-cr13.56marietta.ga.ibone.comcast.net [96.110.39.45]
 20 121 ms 120 ms 120 ms  be-1313-cs03.56marietta.ga.ibone.comcast.net [96.110.34.233]
 21 121 ms 121 ms 120 ms  be-1314-cr14.56marietta.ga.ibone.comcast.net [96.110.34.250]
 22 132 ms 132 ms 132 ms  be-301-cr12.houston.tx.ibone.comcast.net [96.110.32.218]
 23 135 ms 132 ms 132 ms  be-1412-cs04.houston.tx.ibone.comcast.net [96.110.46.149]
 24 133 ms 132 ms 132 ms  be-1413-cr13.houston.tx.ibone.comcast.net [96.110.46.154]
 25 156 ms 156 ms 156 ms  be-302-cr12.losangeles.ca.ibone.comcast.net [96.110.37.190]
 26 156 ms 156 ms 157 ms  be-1112-cs01.losangeles.ca.ibone.comcast.net [96.110.45.165]
 27 156 ms 156 ms 156 ms  be-2102-pe02.losangeles.ca.ibone.comcast.net [96.110.44.114]
 28 155 ms 155 ms 155 ms  66.208.216.74
 29 156 ms 158 ms 155 ms  vl204.lax-cs2-dist-2.cdn77.com [185.229.188.59]
 30 155 ms 155 ms 155 ms  unn-84-17-45-104.cdn77.com [84.17.45.104]

Trace complete.

```

Figure B.4: Traceroute between local PC and IP address in Los Angeles.

Appendix C

Tuning Bandwidth and RAN Budget

C.1 Tuning the Default Carrier Bandwidth

Figure C.1 shows the PDR for the non-cross-layer RAN-EDD (left) and RAN-M-LWDF (right) schedulers with 125 MHz carrier bandwidth under different network hop simulator configurations. We observe that indeed 125 MHz are not enough to support our default scenario with a sub 5% PDR. For simplicity, we chose 25 MHz as the increment for tuning the carrier bandwidth and hence selected 150 MHz as the default carrier bandwidth.

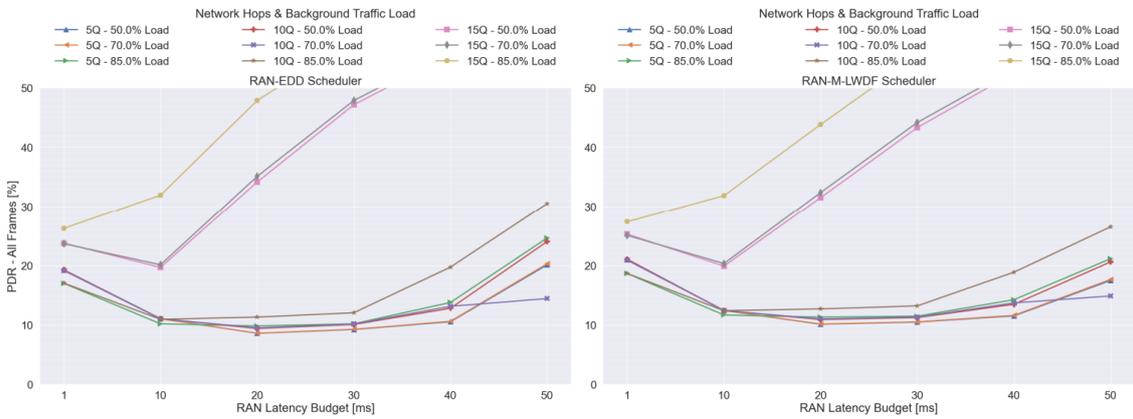


Figure C.1: PDR for non-cross-layer schedulers with 125 MHz carrier bandwidth under different network hop simulator configurations.

C.2 Tuning the Default RAN Latency Budget for Different E2E Requirements

To tune the RAN latency budgets for the different E2E latency requirements as considered in Section 4.3, we use the results shown in Figures C.2 and C.3. Figure C.2 shows that for most IP network scenarios, 10 ms RAN latency budget minimises the PDR. Hence, the RAN latency-based schedulers are configured with 10 ms RAN budget for the E2E latency requirement of 25 ms. In Figure C.3, we see that for many cases, the optimal RAN latency budget would be 60 or 70 ms for an E2E latency budget of 100 ms. However, since in this scenario fifteen network hops can be actually supported as well, the RAN latency budget is chosen as 50 ms, as this configuration allows a sub 5% PDR

for scenarios with fifteen network hops, while minimising the PDR for the other scenarios with five and ten hops.

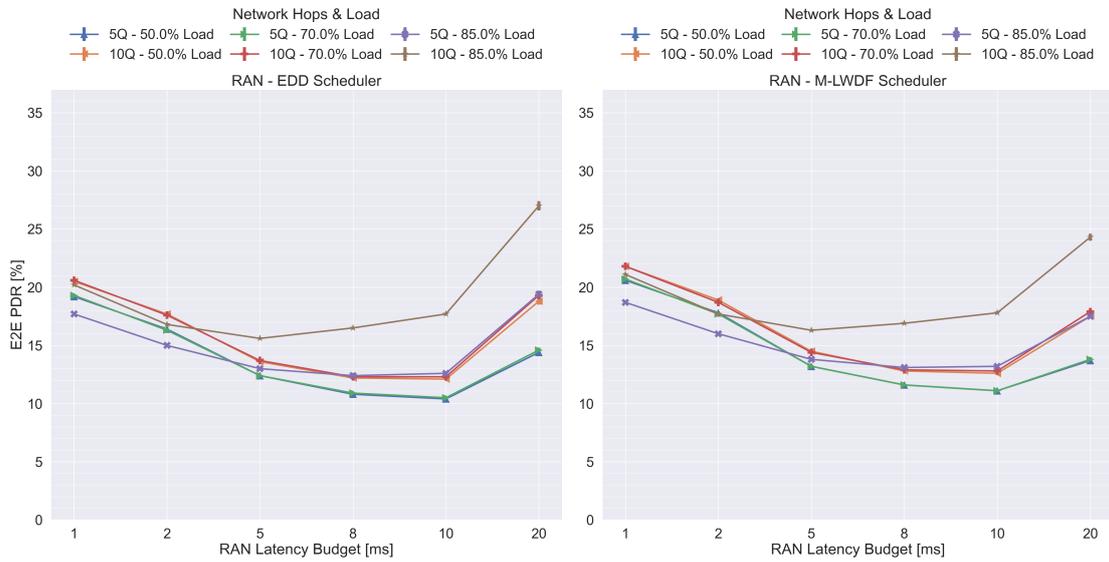


Figure C.2: PDR for non-cross-layer schedulers with 25 ms E2E latency budget under different network hop simulator configurations.

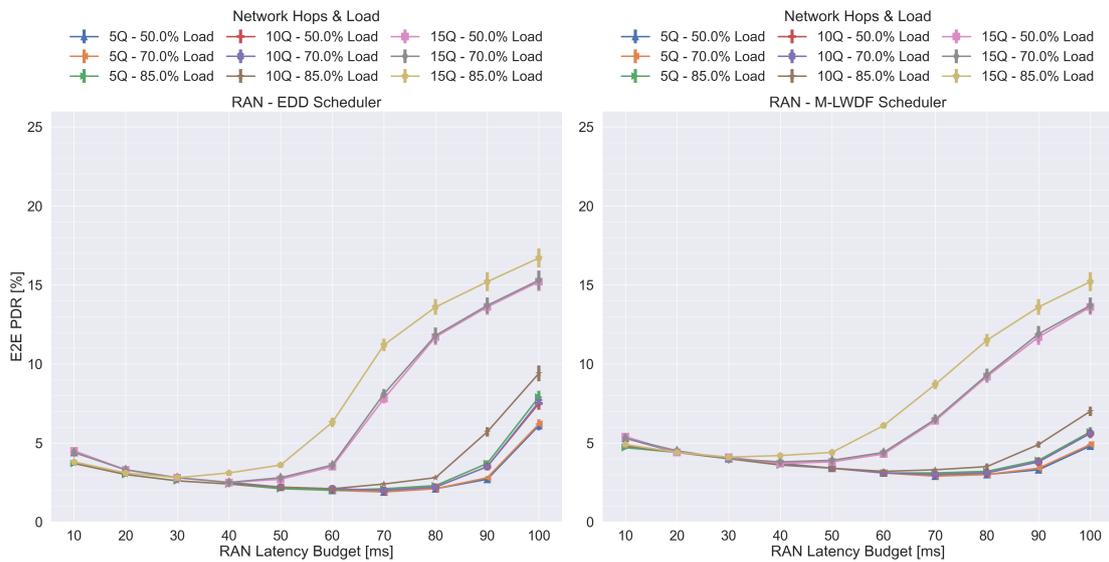


Figure C.3: PDR for non-cross-layer schedulers with 100 ms E2E latency budget under different network hop simulator configurations.

Appendix D

Video Quality Results (PSNR/SSIM) of Sensitivity Analysis

Figures D.1 to D.9 below illustrate the results of the video quality corresponding to the scenarios covered in the sensitivity analysis in Figures 4.4 to 4.12 from Section 4.2.

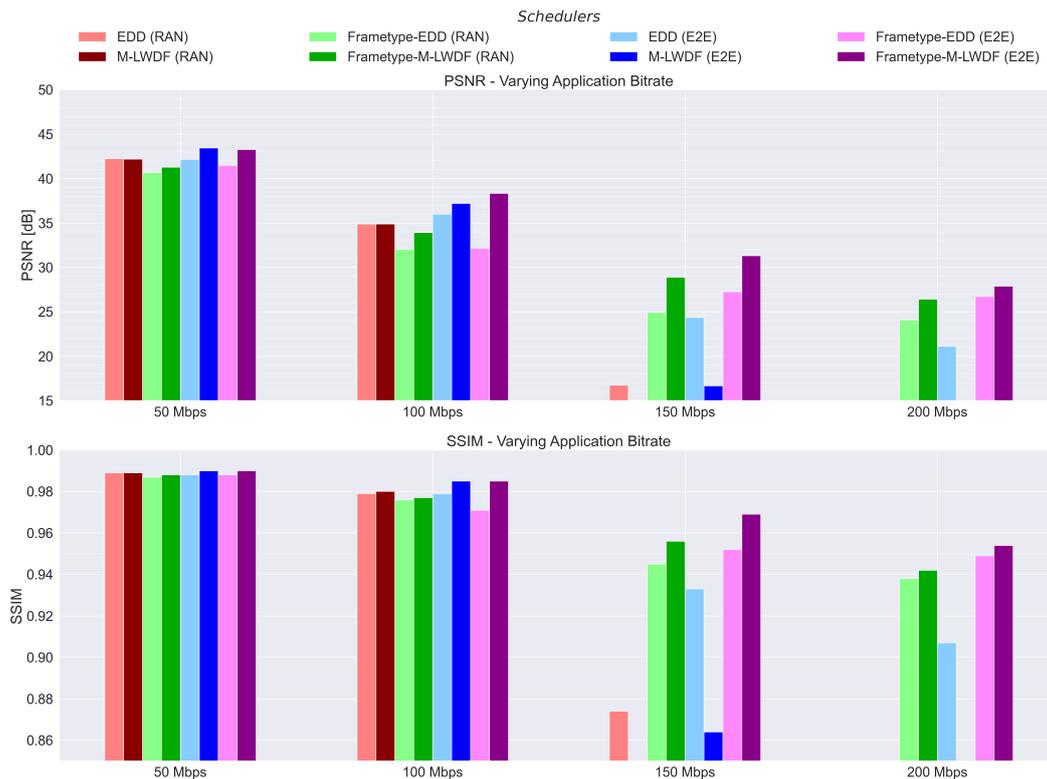


Figure D.1: PSNR and SSIM for varying application bit rates (cf. Figure 4.4).

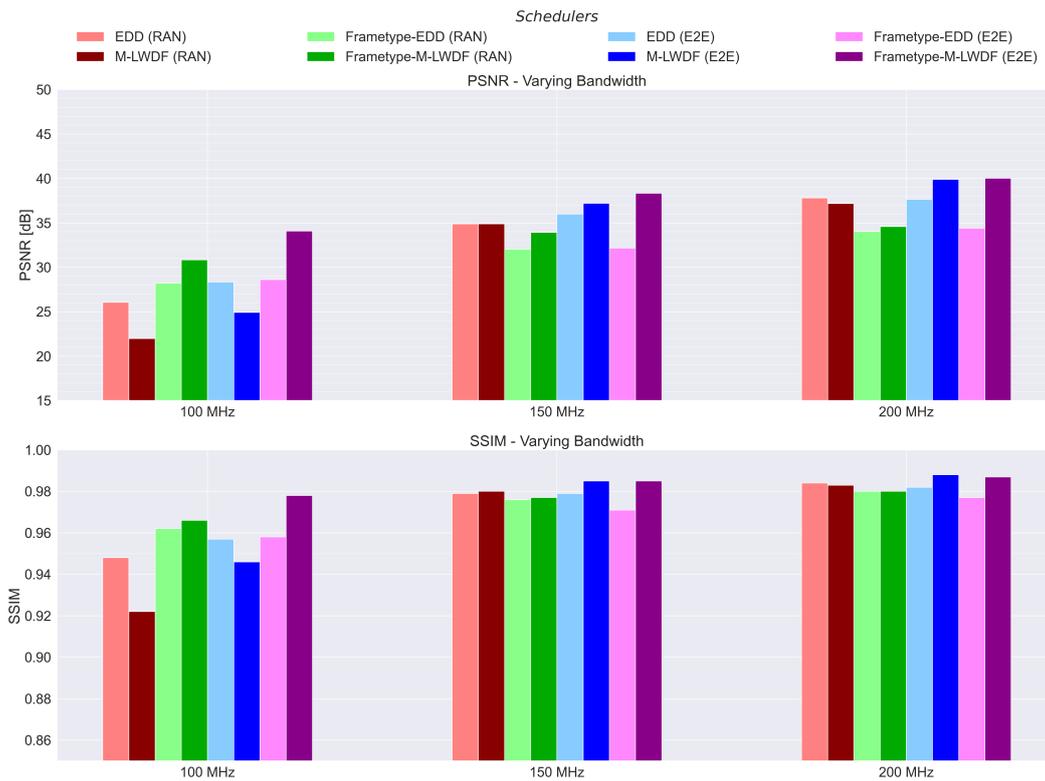


Figure D.2: PSNR and SSIM for varying channel bandwidths (cf. Figure 4.5).

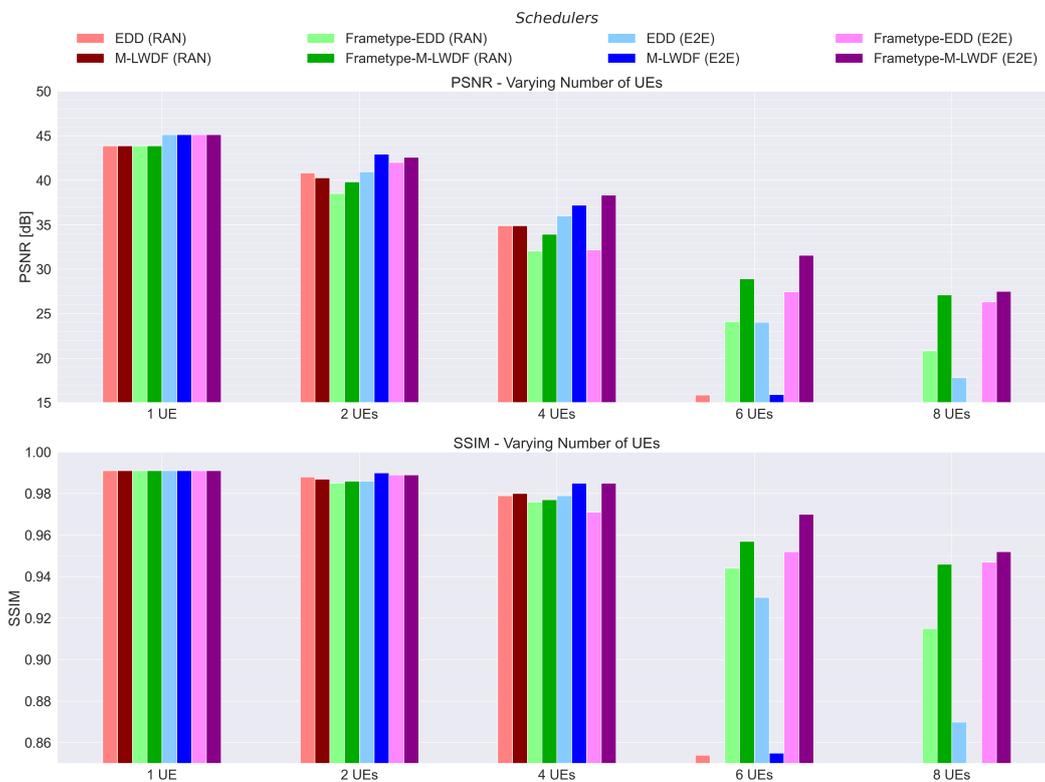


Figure D.3: PSNR and SSIM for varying number of UEs (cf. Figure 4.6).

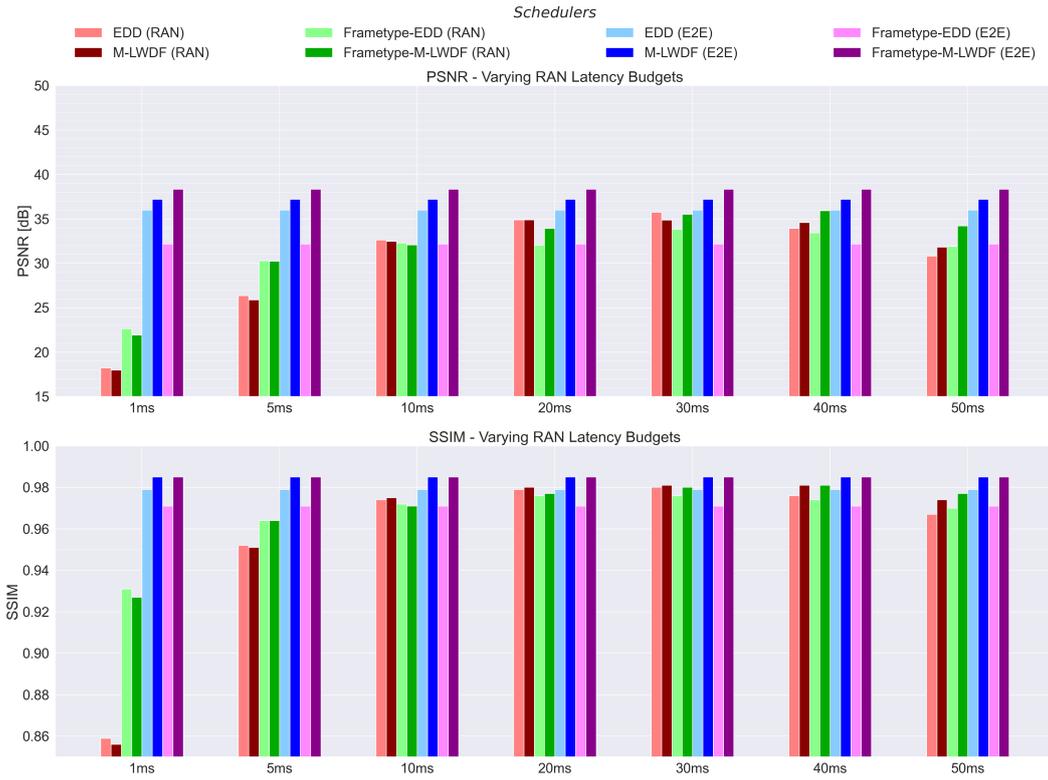


Figure D.4: PSNR and SSIM for varying RAN latency budgets (cf. Figure 4.7).

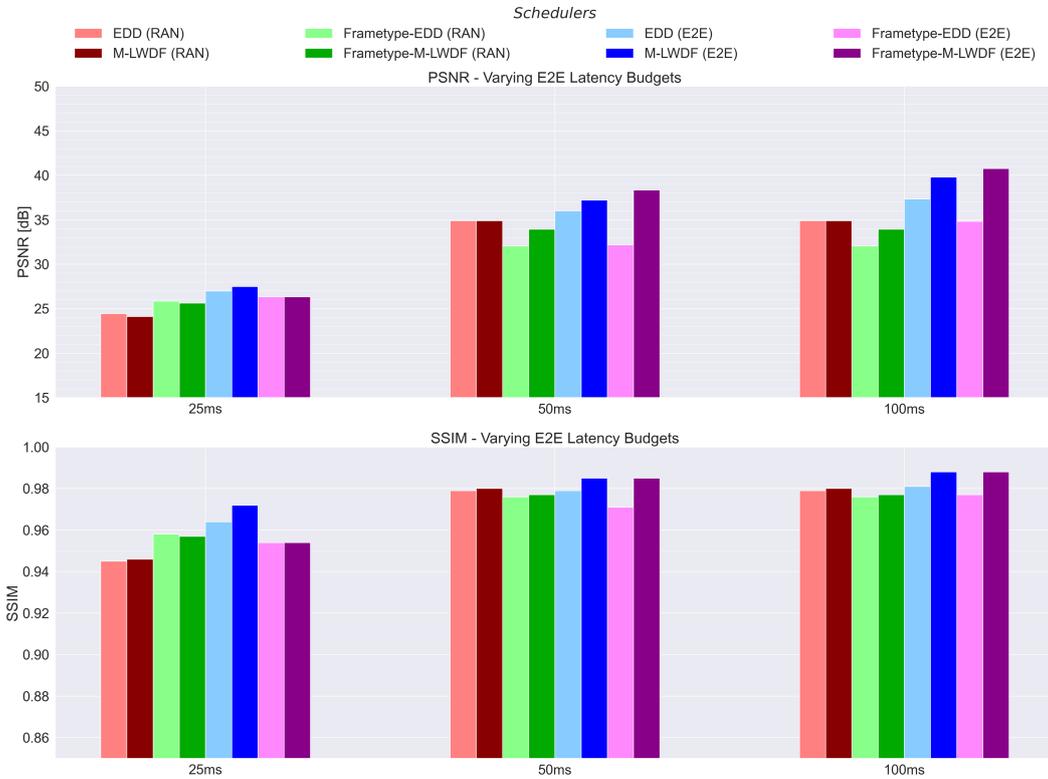


Figure D.5: PSNR and SSIM for varying E2E latency budgets (cf. Figure 4.8).

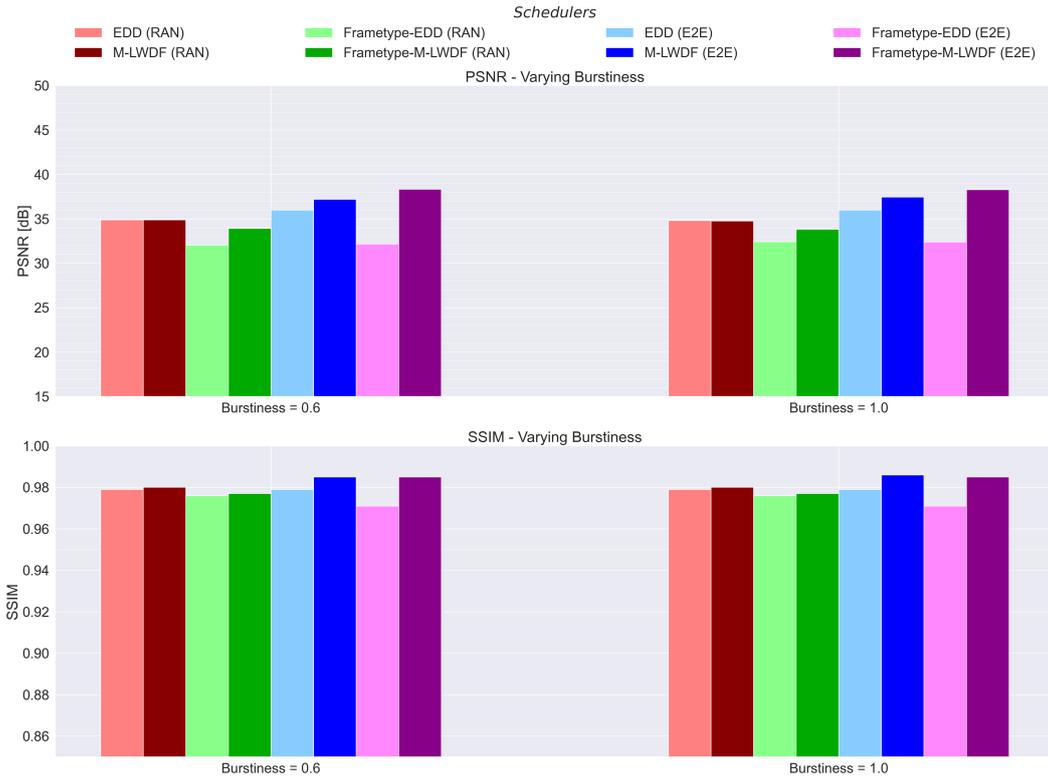


Figure D.6: PSNR and SSIM for varying encoder burstiness (cf. Figure 4.9).

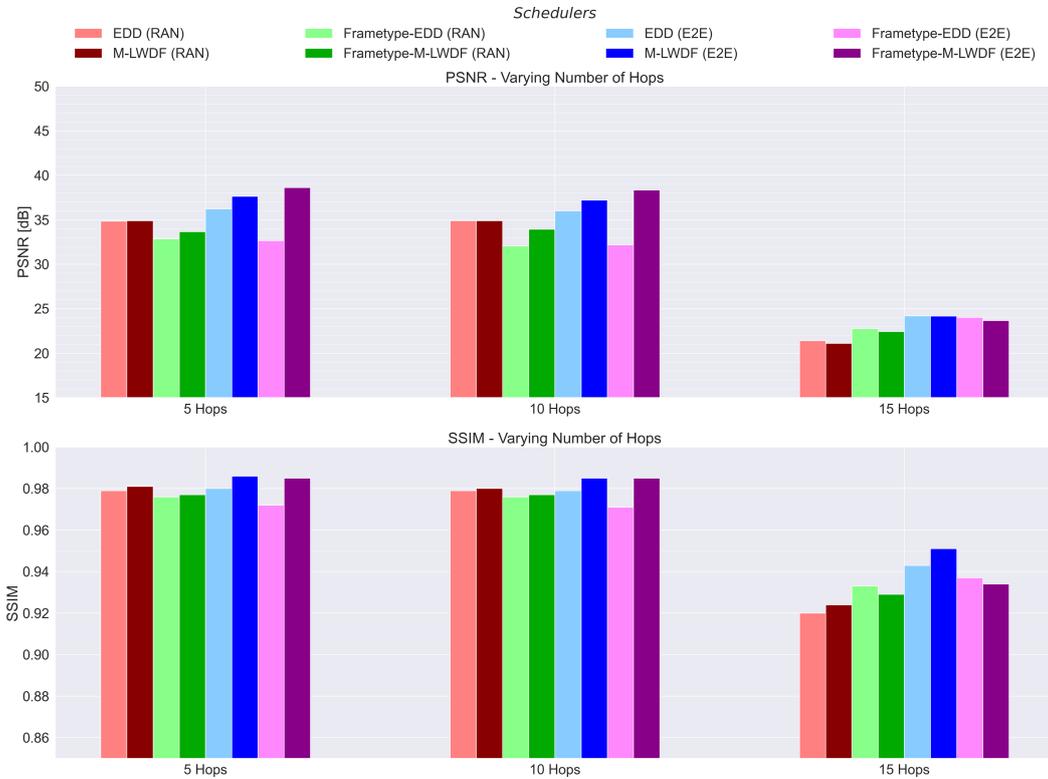


Figure D.7: PSNR and SSIM for varying numbers of network hops (cf. Figure 4.10).

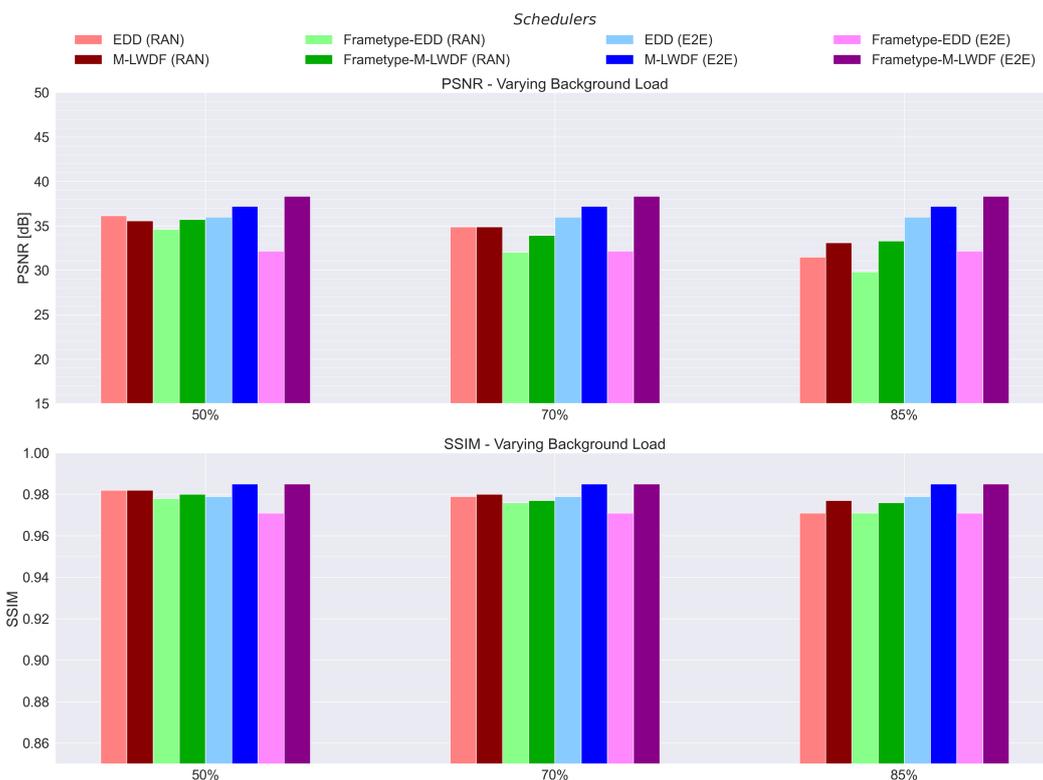


Figure D.8: PSNR and SSIM for varying network background loads (cf. Figure 4.11).

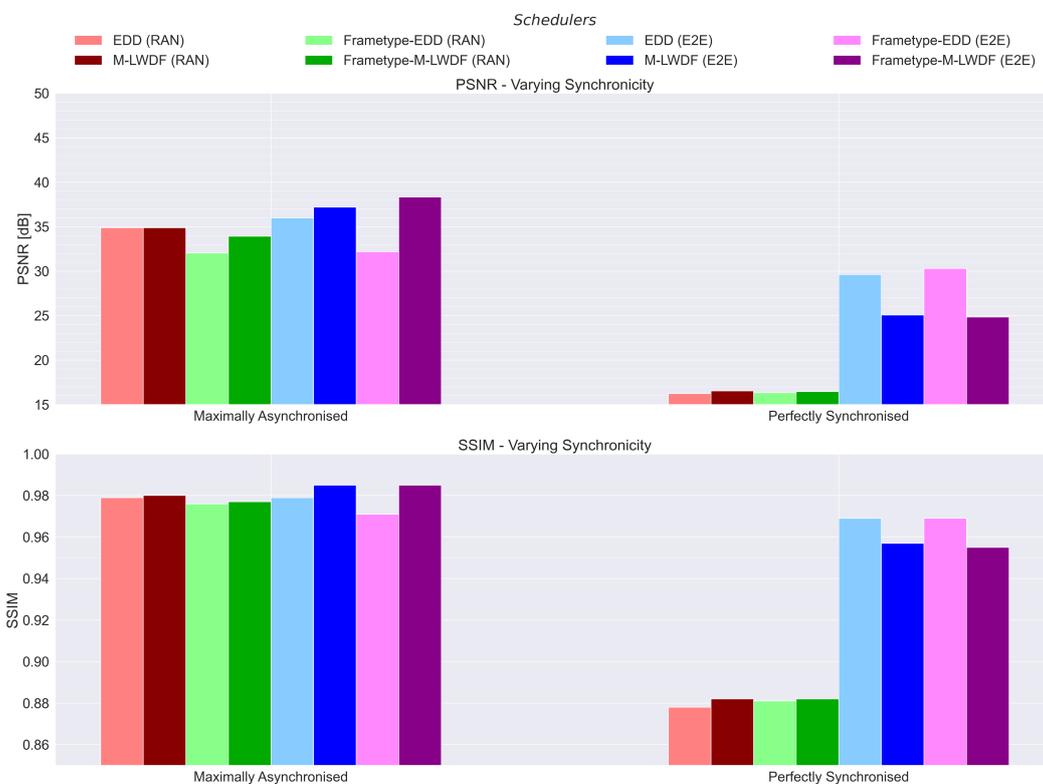


Figure D.9: PSNR and SSIM for varying traffic stream synchronicity (cf. Figure 4.12).