UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Samira Afzal

CAPA: Content-Aware and Path-Aware Scheduling Strategy for Wireless Video Streaming

CAPA: Estratégia de escalonamento sensível a conteúdo e rotas para transmissão de vídeo em redes sem fio

Campinas

2019

Samira Afzal

CAPA: Content-Aware and Path-Aware Scheduling Strategy for Wireless Video Streaming

CAPA: Estratégia de escalonamento sensível a conteúdo e rotas para transmissão de vídeo em redes sem fio

Thesis presented to the Faculty of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Computer Engineering.

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Engenharia Elétrica, na Área de Engenharia de Computação.

Supervisor: Prof. Dr. Christian Rodolfo Esteve Rothenberg

Co-supervisor: Dr. Vanessa Testoni

Este exemplar corresponde à versão final da tese defendida pelo aluna Samira Afzal, e orientada pelo Prof. Dr. Christian Rodolfo Esteve Rothenberg

Campinas

2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

Af98c

Afzal, Samira, 1986-
CAPA : content-aware and path-aware scheduling strategy for wireless video streaming / Samira Afzal. – Campinas, SP : [s.n.], 2019.

Orientador: Christian Rodolfo Esteve Rothenberg.
Coorientador: Vanessa Testoni.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de sinal de vídeo. 2. Redes heterogêneas. 3. Qualidade de serviço (Redes de computadores). 4. Sistemas de comunicação sem fio. I. Esteve Rothenberg, Christian Rodolfo, 1982-. II. Testoni, Vanessa. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: CAPA : estratégia de escalonamento sensível a conteúdo e rotas para transmissão de vídeo em redes sem fio
Palavras-chave em inglês:
Video streaming
Heterogeneous networks
Quality of service
Wireless networks
Área de concentração: Engenharia de Computação
Titulação: Doutora em Engenharia Elétrica
Banca examinadora:
Christian Rodolfo Esteve Rothenberg [Orientador]
Max Henrique Machado Costa
Juliana Freitag Borin
Michele Nogueira Lima
Simone Ferlin
Data de defesa: 29-07-2019
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-4779-3936
- Currículo Lattes do autor: http://lattes.cnpq.br/0776072341126486

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Samira Afzal RA: 162647

Data da Defesa: 29 de Julho de 2019

Título da Tese: "CAPA: Content-Aware and Path-Aware Scheduling Strategy for Wireless Video Streaming"

Prof. Dr. Christian Rodolfo Esteve Rothenberg (FEEC/UNICAMP)(Presidente)

Prof. Dr. Max Henrique Machado Costa (FEEC/UNICAMP)

Profa. Dra. Juliana Freitag Borin (IC/UNICAMP)

Profa. Dra. Michele Nogueira Lima (Universidade Federal do Parana)

Dra. Simone Ferlin (ERICSSON)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de PósGraduação da Faculdade de Engenharia Elétrica e de Computação.

To my beloved mother, Fatemeh, my sweetheart husband, Mohammad,
and my adorable niece, Saina.
Thank you all for your endless love

Acknowledgements

I have had an amazing journey during my PhD full of personal changes and intellectual development. During this journey, many around me had a large part in shaping this experience. Now, here I am. With this dissertation being the result of interacting with many that have made this journey possible and worthwhile.

I would like to express my deep gratitude to my advisor Prof. Christian Esteve Rothenberg, for his valuable support, always guiding and providing me constructive suggestions. I would also like to offer my special thanks to my co-advisor, Dra. Vanessa Testoni, who taught me valuable lessons, both in research and with life. Vanessa, you will always be a role model for me who I look up to. Besides my advisers, I am particularly grateful to Dr. Jean Oliveira, team leader of Samsung research project. This thesis would not have been possible without the help, support and your guidance.

I would also like to acknowledge everyone who supported me throughout the development of this work. Dr. Prakash Kolan and Dr. Imed Bouazizi from Samsung Research America (SRA), for their support and their guidance through each stage of the process. Dr. Pedro Garcia from Samsung Research Brazil (SRBR), for his insightful comments. My friends from INTRIG team were significant to the development of this work. I would also like to mention Dr. Jiyan Wu, who has guided me to understand different aspects of multipath video streaming better.

Besides my advisor, I would like to sincerely thank the rest of my thesis committee, Prof. Dr. Max Henrique Machado Costa (FEEC/UNICAMP), Profa. Dra. Juliana Freitag Borin (IC/UNICAMP), Profa. Dra. Michele Nogueira Lima (Universidade Federal do Parana), Dra. Simone Ferlin (ERICSSON), for their feedback and valuable comments.

This work is part of the results obtained through the project "Hyper Realistic Media", sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8,248/91. I also thank CNPq (Brazilian National Council for Scientific and Technological Development) for grants #141778/2015-6 and #310930/2016-2.

I would like to sincerely thanks my best friend, Golriz Saranjam, who always her friendship and unconditional supports have been greatest my spiritual wealth during my life like a sister. I must express my very deep gratitude to my parents, Fatemeh Marghoosi and Javad Afzal for their love, pray and sacrifice. The most important one, I would like to give special thanks to my beloved husband, Mohammad Katebalhorouf. Thank you for your love and support. Especially, I can't thank you enough for encouraging me throughout this experience.

Abstract

Bandwidth constraint is a real challenge for achieving and sustaining high quality mobile video streaming services. Diverse multipath transmission techniques are being investigated as possible solutions, since recent developments have enabled mobile devices users to receive video data simultaneously over multiple interfaces (e.g., LTE and WiFi). While some multipath protocols have been recently standardized for this purpose (e.g., MPTCP, SCTP), being network layer protocols they cannot properly handle challenging transmission scenarios subject to packet losses and congestion, such as lossy wireless channels. It is important to note that, usually, users just want to have easy access to high quality video content without being necessarily aware of the multipath delivery channels.

In this thesis, we adopt the MPEG Media Transport (MMT) protocol to propose an improvement for multipath wireless video streaming solutions. MMT is an application layer protocol with inherent hybrid media delivery properties. We propose a novel Content-Aware and Path-Aware (CAPA) scheduling strategy for MMT, using full cooperation between network metrics and video content features. Our strategy aims to select the best channel for transmitting each video packet and provides better models to adaptively cope with unstable communication channel conditions and to improve the final user quality of experience (QoE).

For the experimental evaluation, we used ns-3 DCE to simulate different realistic multipath network scenarios, which include channel error models and background traffic. We evaluate CAPA performance over heterogeneous wireless networks under congested network and wireless lossy network conditions, which are common network situations with big adverse effect on video quality. Our approach yields significant video quality improvement in both scenarios compared to Path-Aware strategy (PA) and a simple scheduling strategy for the traditional multipath MMT (ES). For congested network scenario, CAPA could increase PSNR, respectively, by up to 4.25 dB (12.97%), and 7.22 dB (20.58%) compared to PA and ES. It could also improve SSIM, respectively, by up to 0.033 (3.78%), and 0.102 (12.54%) compared to PA and ES. For wireless lossy network scenario, CAPA could increase PSNR, respectively, by up to 6.84 dB (20.30%), and 9.43 dB (30.32%) compared to PA and ES. Our proposed strategy could also reach improvement of SSIM, respectively, by up to 0.100 (12.72%), and 0.113 (14.23%) compared to PA and ES.

We also evaluate videos with different bit rates in our environment under congested network condition to check the behaviour of our proposed strategy, and how it handles streaming of videos with different bit rates to provide sufficient perceived video quality. Furthermore, we have a basic validation of fairness for CAPA to confirm fair access to the available resources in all paths.

Keywords: Video streaming; MMT protocol; Multipath streaming; Content-Aware and Path-Aware scheduling strategy.

Resumo

A restrição de largura de banda é um desafio real para alcançar e sustentar serviços de streaming de vídeo móvel de alta qualidade. Diversas técnicas de transmissão de múltiplos caminhos estão sendo investigadas como possíveis soluções, uma vez que desenvolvimentos recentes permitiram que usuários de dispositivos móveis recebessem dados de vídeo simultaneamente em várias interfaces (ex.: LTE e WiFi). Embora alguns protocolos com suporte a múltiplos caminhos tenham sido padronizados recentemente para esse propósito (ex.: MPTCP, SCTP), sendo protocolos de camada de rede, eles não podem lidar adequadamente com cenários de transmissão desafiadores sujeitos a perdas de pacotes e congestionamentos, como canais sem fio com perdas. É importante notar que, geralmente, os usuários só querem ter acesso fácil a conteúdo de vídeo de alta qualidade sem necessariamente estarem conscientes dos canais de distribuição de múltiplos caminhos.

Nesta tese, adotamos o protocolo MPEG Media Transport (MMT) para propor um aprimoramento para soluções de streaming de vídeo por múltiplos caminhos sem fio. O MMT é um protocolo da camada de aplicação com propriedades inerentes de entrega de mídia híbrida. Propomos uma nova estratégia de programação Content-Aware and Path-Aware (CAPA) para o MMT, usando cooperação entre métricas de rede e recursos de conteúdo de vídeo. Nossa estratégia tem como objetivo selecionar o melhor canal para transmitir cada pacote de vídeo e fornecer modelos melhores para lidar de forma adaptável com as condições do canal de comunicação instável e melhorar a qualidade da experiência do usuário final (Quality of Experience – QoE).

Para a avaliação experimental, usamos o ns-3 DCE para simular diferentes cenários realísticos de rede de múltiplos caminhos, que incluem modelos de erro de canal e tráfego de fundo. Avaliamos o desempenho do CAPA em redes sem fio heterogêneas congestionadas e condições de rede sem fio com perdas, que são situações de rede comuns com grande efeito adverso na qualidade do vídeo. Nossa abordagem gera melhoria significativa da qualidade de vídeo em ambos os cenários em comparação com a estratégia Path-Aware (PA) e uma estratégia de escalonamento simples para o tradicional multipath MMT (ES). Para o cenário de rede congestionada, o CAPA pode aumentar a PSNR, respectivamente, em até 4,25 dB (12,97%) e 7,22 dB (20,58%) em comparação com PA e ES. Também pode melhorar o SSIM, respectivamente, em até 0,033 (3,78%) e 0,102 (12,54%) em comparação com PA e ES. Para o cenário de rede com perdas sem fio, o CAPA pode aumentar a PSNR, respectivamente, em até 6,84 dB (20,30%) e 9,43 dB (30,32%) em comparação com PA e ES. Nossa estratégia proposta também pode alcançar melhorias de SSIM, respectivamente, em até 0,100 (12,72%) e 0,113 (14,23%) em comparação com PA e ES.

Também avaliamos vídeos com diferentes taxas de bits em nosso ambiente em condição de rede congestionada para verificar o comportamento de nossa estratégia proposta e como ela lida com streaming de vídeos com diferentes taxas de bits para fornecer qualidade de vídeo percebida suficiente. Além disso, temos uma validação básica de justiça para o CAPA para confirmar o acesso justo aos recursos disponíveis em todos os caminhos.

Palavras-chaves: Transmissão de video; Protocolo MMT; Transmissão por múltiplos caminhos; Estratégia de escalonamento consciente de caminho e conteúdo.

List of Figures

List of Tables

Acronyms

5G Fifth Generation

ABR Adaptive Bit Rate

ADC Asset Delivery Characteristics

AIMD Additive-Increase/Multiplicative-Decrease

AL-FEC Application Layer Forward Error Correction

API Application Programming Interface

AR Augmented Reality

ARQ Automatic Repeat reQuest

ATSC Advanced Television Systems Committee

CAPA Content-Aware and Path-Aware

CCID Congestion Control IDentifier

CDF Cumulative Distribution Function

CDN Content Delivery Networks

CI Composition Information

CMT Concurrent Multipath Transfer

CRC Cyclic Redundancy Check

DASH Dynamic Adaptive Streaming over HTTP

DCCP Datagram Congestion Control Protocol

DDE Distributed Decision Engine

DDS Data Distribution Scheduler

DRM Digital Rights Management

DSN Data Sequence Number

DUP DUPlication

DVB-H Digital Video Broadcasting Handheld

ECN Explicit Congestion Notification

EDPF Earliest Delivery Path First

ES Evenly Split

ESNR Effective Signal-to-Noise Ratio

FEC Forward Error Correction

FHD Full HD

FIFO First-In First-Out

GFD Generic File Delivery

GoP Group of Picture

HARQ Hybrid-ARQ

HAS HTTP Adaptive Streaming

HEVC High Efficiency Video Coding

HOL Head-of-Line

HRBM Hypothetical Receiver Buffer Model

HTTP Hypertext Transfer Protocol

IPTV Internet Protocol TeleVision

ISO/IEC International Organization for Standardization/International Electrotechnical Commission

ISOBMFF ISO Base Media File Format

ITS Intelligent Transport Systems

JSCC Joint Source and Channel Coding

LDGM Low-Density Generator Matrix

LDPC Low-Density Parity-Check

MDC Multiple Description Coding

MDP Markov Decision Process

MMT MPEG Media Transport

MMTP MPEG Media Transport Protocol

MOS Mean Opinion Score

MPD Media Presentation Description

MPEG-2 TS MPEG-2 Transport System

MPTCP Multipath TCP

MPU Media Processing Unit

MR Mixed Reality

MS-SSIM MultiScale Structural SIMilarity

MSE Mean Squared Error

MVV Multi-View Video

MXD MMT eXtension Document

NAMF Network Abstraction for Media Feedback

NAT Network Address Translator

NTP Network Time Protocol

ORP Optimal Retransmission Policy

OVS Open vSwitch

P2P peer to peer

PA Path-Aware

PFEC Proactive FEC

PI Presentation Information

PLR Packet Loss Rate

POC Picture Order Counts

PQEM Path Quality Estimation Model

PSNR Peak Signal-to-Noise Ratio

QoE Quality of Experience

QoS Quality of Service

QUIC Quick UDP Internet Connection

RAT Radio Access Technologies

RFEC Reactive FEC

RLM Receiver-driven Layered Multicast

RQF Reception Quality Feedback

RS Reed-Solomon

RTC Real-Time Communications

RTMP Real Time Messaging Protocol

RTO Retransmission Time-Out

RTP Real-time Transport Protocol

RTSP Real-time Streaming Protocol

RTT Round-Trip Time

rwnd receiver window

SACK Selective Acknowledgements

SAMVIQ Subjective Assessment Methodology for Video Quality

SAND Server and Network-assisted DASH

SDN Software-Defined Networking

SNR Signal-to-Noise Ratio

SPTCP Single Path TCP

sRTT smooth Round Trip Time

SSIM Structural SIMilarity

SSL Secure Sockets Layer

SSN Subflow Sequence Number

SVC Scalable Video Coding

SWF Small Web Format

T-DMB Terrestrial Digital Multimedia Broadcasting

TCP Transmission Control Protocol

TFRC TCP-Friendly Rate Control

TLS Transport Layer Security

TSN Transmission Sequence Number

TTFB Time To First Byte

UDP User Datagram Protocol

UHD Ultra HD

V2X Vehicle-to-everything

VMAF Video Multimethod Assessment Fusion

VoD Video on Demand

VQM Video Quality Metric

VR Virtual Reality

WRR Weighted Round Robin

Contents

1 Introduction

Multimedia services (e.g., Skype, FaceTime) and on-demand mobile video content (e.g., Hulu, YouTube, Netflix) have become part of daily Internet content exchange. Likewise, online cloud gaming is a very popular entertainment (JARSCHEL *et al.*, 2011). End users always expect the highest quality video experience, regardless of the network situation. To this end, delivering high throughput and low delay are challenging requirements especially on lossy wireless channels.

Several solutions have been proposed to provide Quality of Experience (QoE) for video streaming. One type of approach are packet loss resilient methods (KAZEMI *et al.*, 2014; Huo *et al.*, 2015) such as automatic repeat request (ARQ), Forward Error Correction (FEC), and Error Resilient Coding (ERC), which cope with noisy networks by reducing the effect of data loss. Adaptive streaming mechanisms (TRESTIAN *et al.*, 2018; Kim *et al.*, 2016; ZAHRAN *et al.*, 2018) are also remarkable options to dynamically adjust the video delivery data rate to the underlying network conditions. Another type of solution is to exploit multipath strategies (TRESTIAN *et al.*, 2018; AFZAL *et al.*, 2018; WU *et al.*, 2016a) by taking advantage of the multiple network interfaces that are currently available in most wireless devices (e.g., laptops, tablets, smartphones). A user connected to more than one network opens the opportunity to leverage different video delivery strategies to provide better coverage and overall more stable network connectivity by circumventing congested network paths and aggregating the bandwidth available over multiple paths.

This is specific to the context where this thesis aims at delivering contributions by exploring the advantages of multipath video streaming using the MPEG Media Transport (MMT) protocol (KOLAN; BOUAZIZI, 2016; MPEG, 2017). MMT is an appropriate protocol for exploiting multipath streaming because it is a multimedia application layer protocol with the inherent ability to transmit video in heterogeneous network environments, and it also has the capability of hybrid media delivery.

In a nutshell, we propose investigating novel multipath scheduling strategies which consider both video features (content-aware) and path characteristics (path-aware). Considering video content features in the scheduling strategy helps to define the priority of each packet, and subsequently, unequal importance packets can be sent through different network paths based on network-level quality estimators. The proposed Content-Aware and Path-Aware (CAPA) scheduling strategy improves video streaming QoE by increasing goodput, decreasing packet losses and end-to-end delay.

In this introductory chapter, we provide further motivation behind the topic and describe the benefits and challenges of multipath video transmission over heterogeneous wireless networks. We also introduce the main goals of this thesis towards a solution that advances the state of the art along the methodologies used to validate and evaluate our proposed methods. The achievements and contributions are also briefly presented in this chapter in addition to the short description of the thesis outline.

1.1 Motivation and Scope

The fast development of media technologies, applications such as on-demand video, video conferences and online cloud gaming have been responsible for explosive growing amount of network traffic. According to the annual Cisco's report (CISCO, 2017), IP video traffic would take 82 percent of all consumer Internet traffic by 2021. Delivering high-quality video streaming services makes the task of providing real-time wireless transmission of multimedia while ensuring perceived quality quite challenging due to bandwidth and time constrains (SANI *et al.*, 2017).

Technologies such as 4K Ultra HD (UHD) and 8K UHD, commonly known as Super Hi-Vision in Japan, have gotten a lot of attention over the past few years. Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR) as well as Multi-View Video (MVV) also become true in the close future. However, the real problem behind these technologies is the huge bit rate they produce (VIEIRA, 2014). For instance, an uncompressed 4K UHD sequence generates a maximum bit rate around 36 Gbps. The Super Hi-Vision format, which has four times the spatial resolution of 4K UHD mode, generates a 144 Gbps bit rate sequence. Even compressing with H.264/MPEG-4 AVC or High Efficiency Video Coding (HEVC) generates extremely high output bit rate. NHK Science & Technology Research Laboratories encoded an 8K signal in real-time using HEVC and achieved an output bit rate of 85 Mbps. Therefore, such services demand high bandwidth transmission while it is insufficient to transmit it through the current available IP based networks.

Wireless communication has grown very fast with various developments and inventions. Nowadays, it has become a ubiquitous part of modern life and it is a crucial part of mobile communications. Considering that mobile means wireless communications which may or not include mobility scenarios (e.g., a wireless laptop at home or a wireless laptop on a train). According to (CISCO, 2017; CISCO, 2015), since 2012, mobile video has represented more than half of global mobile data traffic and will keep being responsible for the largest traffic growth upfront. Mainly, technological innovations such as portable devices (e.g., smartphones, tablets and laptops) cause to increase the number of mobile users. These mobile devices have remarkable functionalities such as built-in camera, high-resolution display, high computing power and multiple network interfaces. Therefore, users expect high-quality video streaming services. However, current networks, even with the last development of novel network infrastructures, cannot guarantee to sup-

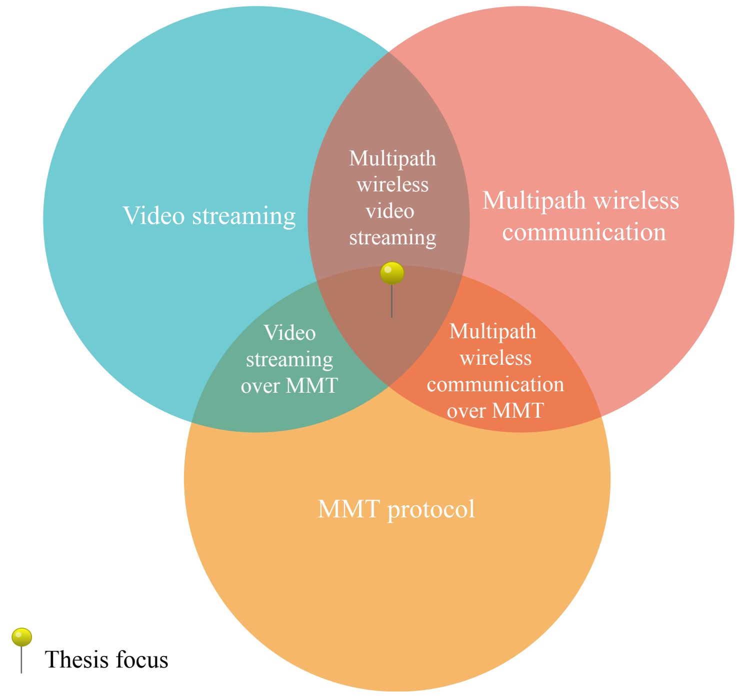


Figure 1 – The focus of this thesis is the confluence of MMT protocol for video streaming applications over heterogeneous wireless networks.

port this requirement. Multipathing (QADIR *et al.*, 2015; POLIAKOV *et al.*, 2018) is an emerged technology which plays a major role in the wireless networks since mobile devices, with multipath technology, can exploit their interfaces to simultaneous transmission over multiple heterogeneous wireless paths. Therefore, multipathing overcomes network limitations by increasing bandwidth and balancing the loads over paths, and consequently, improves throughput performance, reliability and fault tolerance.

5G Next generation mobile networks target to support new services that require extreme bandwidth and ultra-low latency. One concept presented in 5G to reach this goal is the by exploiting multihoming capabilities to reach the required QoS by effectively combining all available wireless access networks.

Multiple efforts have been done regarding multipath data transmission, since the traditional TCP and UDP protocols do not support multiple paths. Different multipath protocols have been proposed (CHAKARESKI *et al.*, 2005; CHAKARESKI; GIROD, 2003). However, most of the current transport protocols do not match the requirements of video streaming applications or are not designed to address relevant issues, such as delay constraints, networks heterogeneity, and head-of-line blocking issues.

At the crossroads, we observe MPEG Media Transport (MMT) (MPEG, 2014), as a popular multimedia protocol developed standardized in 2014 as a part of the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 23008 High Efficiency Coding and Media Delivery in Heterogeneous Environments (MPEG-H) standard suite (LIM *et al.*, 2014). Several other standards have already

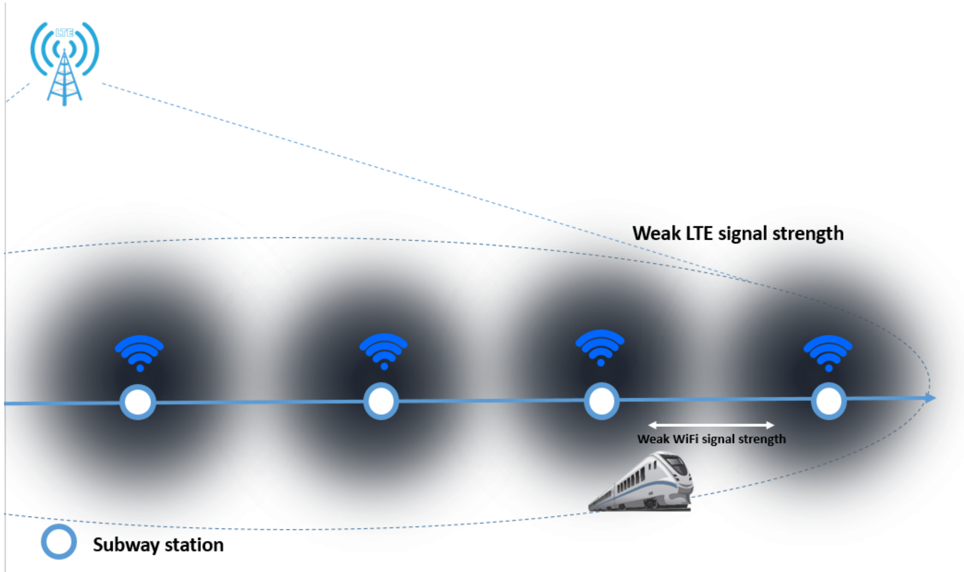


Figure 2 – Multipath mobile video streaming example where a user is watching a real-time video on his/her smartphone at subway train with existing LTE and WiFi signal in variable strengths.

adopted MMT for major advances in televisual technology worldwide (BAE, 2013; AOKI, 2017). MMT supports UHD, on-demand, live video streaming, and MMT has also been widely used for VR, AR and MVV technologies. We regard MMT as an appropriate protocol for exploiting multipath streaming because it inherits the ability to transmit video in heterogeneous network environments. Actually, the capability of hybrid media delivery is one of the main MMT properties. Hybrid media delivery refers to the combination of delivered media components over different types of network. For example, one could be a broadcast channel and another broadband, or two simultaneous broadband channels can be combined. More technical details on MMT are provided in Sections 2.2 and 2.3.

Altogether, as shown in Figure 1, the scope and motivation of this thesis is based on improving the MMT protocol with multipath strategies to deliver improved video quality of experience for real-time wireless video streaming.

Motivational examples. A motivation example for the benefit of multipath video streaming strategy is watching a real-time video on a mobile device (e.g., smartphone, tablet, notebook, laptop) at a public place where could be crowed in some period of times (e.g., library, campus, airport). For example, suppose a user is watching a video on his/her smartphone at a campus equipped with WiFi access. The smartphone is connected to both LTE and WiFi. When the campus is crowded, the WiFi becomes congested, and consequently, it has adverse effects on the perceived video quality. But in a short time, the multipath strategy can recover the video quality by rerouting the video traffic through the LTE path. Then, since the campus becomes even more crowded, it may happen that LTE also becomes congested and causes low video quality for the user. However, utilizing an efficient multipath strategy leads to optimal perceived video quality.

Another motivation example for the benefit of multipath video streaming strategy is watching a real-time video on a mobile device at a moving vehicular (e.g., subway train, train, public transition bus, personal car). For example, suppose a user is watching a video on his/her smartphone at a subway where stations are equipped with public WiFi access points but the subway itself does not provide WiFi access. Besides, the LTE signal strength in the subway area is not high, see Figure 2. The smartphone is connected to both LTE and WiFi. When the subway leaves the station, there is not high signal strength for neither LTE nor WiFi. However, utilizing an efficient multipath strategy leads to optimal perceived video quality. Then, since the subway gets farther away from the station, the signal strength of the WiFi becomes even weaker and causes the low video quality for the user. But in a short time, the multipath strategy can recover the video quality by rerouting the video traffic through the LTE path. Therefore, multipath strategies adaptively monitor the paths' conditions and cope with unstable paths to keep optimal QoE.

1.2 Benefits and Challenges of Multipath wireless Video

As previously stated, providing high/optimal QoE for the final user in the wireless video streaming scenario requires high bandwidth and low transmission delay. This is a challenging task, considering the several aspects involved in wireless transmission, such as bandwidth constraints, lossy wireless channels, delay, lack of coverage and congested networks. Adding multipath transmission ability can help with this challenge and its benefits can be summarized as:

- **Reliability and seamless connectivity:** using multipath allows the user to simultaneously utilize multiple available network connections. A better coverage is achieved and the probability of keeping an End-to-End connection alive is increased. In the case of failure or congestion in one network path, multipathing provides a resilient alternative, resulting in improved user video experience.
- **Throughput increase:** by aggregating bandwidth and distributing video traffic over multiple network paths, faster transmission can be achieved, which is essential for real-time video streaming applications (YAP *et al.*, 2012).
- **Load balancing:** refers to efficiently distribute video traffic through the available network paths in order to relieve congestion (QADIR *et al.*, 2015). Load balancing improves stability by achieving lower variability and inter-packet delay (jitter).
- **Reduction of burst loss length:** burst loss length refers to the continuous packet losses which is harmful to perceived video streaming quality (APOSTOLOPOULOS,

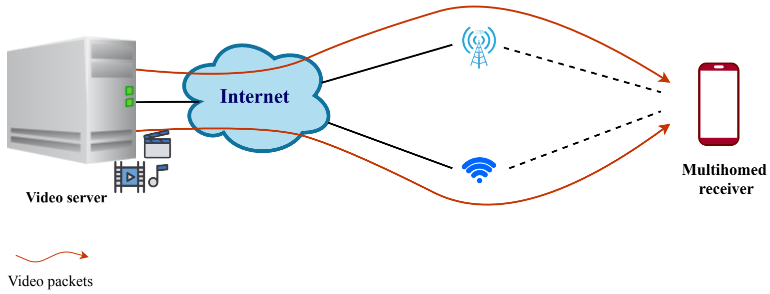


Figure 3 – Multipath wireless video streaming over LTE and WiFi networks.

2000). This is because decoder can recover the loss of a small number of video packets by exploiting correlations in the previously received video sequence to conceal the lost information. However, the effectiveness of this recovery decreases dramatically in case of losing large number of continuous video packets. Using multipath streaming benefits to convert burst losses to isolated losses, and consequently, probability of recovering lost packets would be increased.

- **Delay decrease:** Using multiple paths contributes to having video data ready at the receiver faster, thus, decreasing the effective delay especially from an application perspective. Probing multiples paths enables to get the data from the lowest delay path, reducing the Time To First Byte (TTFB), i.e., the time between the video request being sent and the first packet received after the request (GUTTERMAN *et al.*, 2019).

- **Security:** video split over multiple paths improves protection to some security threats (SINGH *et al.*, 2015) once that each network path only carries parts of the whole video stream.

Fortunately, with the recent development, many of current devices (MUSHROOM, 2017; GALAXYS5, 2017) are already equipped with both cellular and WiFi interfaces. Multiple interface devices, which could be equipped with two or more than two interfaces, having the ability to connect simultaneously to multiple network paths are known as multihomed devices, as illustrated in Figure 3. Multihomed devices can utilize multipath communication by aggregating the available bandwidth from multiple Radio Access Technologies (RATs). With multiple interfaces, users can receive data through parallel paths with multiple IP addresses.

Despite all of its benefits, attempting to deploy a multipath solution for video delivery can bring the following potential roadblocks from a practical perspective:

Compatibility. Implementation of a general multipath solution usually requires changing one or both of server and client sides, modifying standardized protocols, improving operating systems kernel and/or changing third-party network equipment.

Networks heterogeneity. Heterogeneous wireless networks vary based on different bandwidth constraints, delays, jitters and packet loss rates. These different physical properties cause asymmetric communication for video transmission, and consequently, may decrease the overall streaming quality. For instance, a large difference between LTE and WiFi bandwidth decreases the bandwidth aggregation performance (BUI *et al.*, 2013). Second and third generation (2G and 3G) of cellular networks have not enough bandwidth to support live video streaming due to high data rate (HAN *et al.*, 2011; LUO, 2011; CHEN *et al.*, 2013). In addition, 3G cellular networks local retransmission mechanism (CHEN *et al.*, 2013) may increase the Round-Trip Time (RTT) and the rate variability. Nowadays, new 3G services evolved and offer ten to twelve times faster services (BONDERUD, 2017) while 4G LTE supports higher data transmission rate and signal coverage than 2G/3G (YOON *et al.*, 2012; CHEN *et al.*, 2013). On the other hand, WiFi networks have shorter RTT compared to LTE but suffer from lossy channels due to the constraints on the signal coverage and mobility support (XU *et al.*, 2013; MITCHELL, 2017).

For example, observations in (CHEN *et al.*, 2013) show that while the loss rate over 3G/4G networks is generally lower than 0.1%, for WiFi networks, this value varies from 1% to 3%. The average RTT for WiFi networks is around 30 ms, while the base RTT of 4G networks is 60 ms, which can even increase by four to ten times in a single 4G connection depending on the carrier and the flow sizes, and up to twenty times in 3G networks. The authors also pointed that although 4G networks have larger RTTs in general, in many of their experiment measurements, WiFi did not perform better than 4G LTE.

In addition, wireless losses are recognized as congestion by some protocols (e.g., TCP), resulting in decreased network throughput (CHAN; RAMJEE, 2005). Therefore, multipath transmission in heterogeneous wireless networks becomes a truly challenging task.

Out-of-order packets. Spreading data over heterogeneous paths with different RTTs, throughput fluctuations, and jitter existence introduces the out-of-order packets problem. This phenomenon causes unnecessary packets retransmissions, wasted bandwidth, and consequently, network congestion. In addition, more time is required to recover the ordered data. A robust multipath transmission solution is required to cope with packet reordering in heterogeneous wireless networks (LI *et al.*, 2016b) to avoid video quality degradation.

Head-of-Line (HOL) blocking. When many packets are stored in the destination

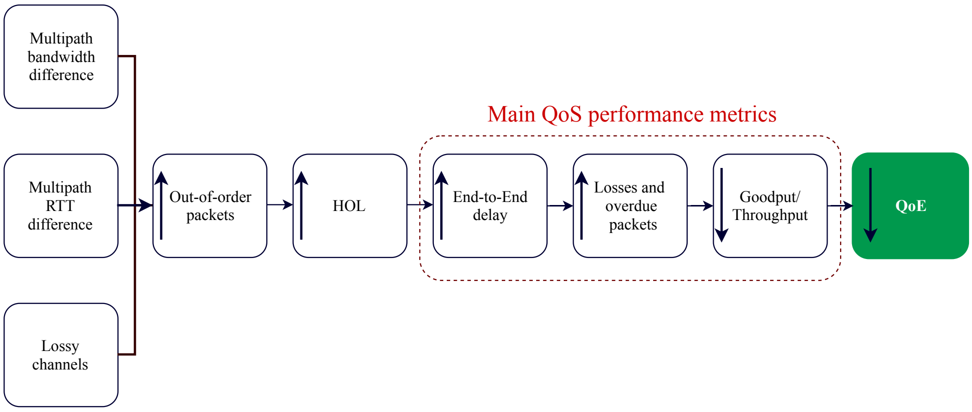


Figure 4 – Challenges of employing multipath transmission in wireless video streaming applications and possible adverse effects to be avoided.

buffer waiting for delayed packets, the buffer may become full and blocked. This issue is referred as Head-of-Line blocking (CORBILLON *et al.*, 2016; FERLIN-OLIVEIRA *et al.*, 2014). Generally, buffer blocking occurs with reliable protocols that guarantee in-order packet delivery, such as TCP, and it may become worse in case of multipath delivery. The Bufferbloat phenomenon is the main reason of HOL blocking, contributing to high latency, especially in 3G/4G cellular networks (CHEN; TOWSLEY, 2014; FERLIN-OLIVEIRA *et al.*, 2014). Bufferbloat occurs because of significantly large network buffers (e.g., large router queues) that avoid packet loss at the cost of adding high latencies under congestion. The problem can become worse in the case of multipath delivery because if bufferbloat occurs in one of the paths, those packets arrive at the destination with high delay and out-of-order, resulting in HOL blocking. Consequently, HOL blocking not only increases End-to-End delay and jitter but successfully arrived packets may become obsolete (i.e. discarded) due to the long waiting time in the destination buffer.

End-to-End delay. Real-time video streaming requires a bounded End-to-End delay (SUN *et al.*, 2016), which refers to the measured delay from the generation of a video frame to the moment when it can be decoded. End-to-end delay includes holding time of a video frame at both sender and receiver sides, and the transmission delay. It could also include the queuing delay, propagation delay, access delay, and reordering delay. The queuing delay refers to packet buffering in the sender, receiver and other nodes in the network during packet transmission. The transmission delay and radio access delay occur in the physical transmitter to map the data from packets to bits on physical radio interface's hardware. The distance between entities causes the propagation delay. The access points introduce transfer and propagation delay. In the case of video streaming on multipath networks, reordering delay can be increased (BROSH *et al.*, 2010).

Overdue packets. Video data packets arriving at the destination after decoding deadlines are expired and known as overdue packets. While overdue packets for UDP-like transmissions may cause video distortions (i.e., degradation of the visual video fidelity (SANI *et al.*, 2017)) similar to lost packets, in reliable transport protocols like TCP the effects surface as stalling (i.e., video freezes) or rebuffering. Avoiding stalls becomes most critical in live streaming scenarios. Thus, this kind of real-time applications, even when based on TCP-like solutions, consider the overdue packets as lost packets since they are discarded. This concept is called liveness (SANI *et al.*, 2017). Therefore, suitable multipath streaming strategies need to consider potential decoding deadlines of the receivers.

Wrapping up the Challenges. Figure 4 aims at putting together all key issues and possible adverse effects of multipath wireless video streaming. The design of any multipath wireless video streaming solution, it is important to avoid or at least minimize such effects. In other words, optimization of QoS-related parameters leads to improved QoE (BARAKOVIĆ; SKORIN-KAPOV, 2013). QoS measurements may differ based on the type of video streaming service (SCHAAR; CHOU, 2011; SANI *et al.*, 2017) such as VoD, live or real-time. VoD is a video streaming service which encoded media is pre-stored at the server, and the user can select and watch it at any time (e.g., Netflix movies). In contrast, in live and real-time video streaming services (e.g., live sport streaming, real-time video including interactive video call, gaming, etc.) the video content is not pre-stored or available when the streaming starts. In live streaming, the buffer is smaller compared to VoD streaming to avoid long delays and it also has stricter deadlines. Real-time video streaming has even shorter delay constraint. For example, according to the ITU-T recommendation G.1010 (ITU-T, 2001) and (WU *et al.*, 2016a), a large delay of 5 seconds may be acceptable for VoD and around 1 second delay is acceptable for live streaming, but in order to achieve excellent real-time streaming quality, the solution should provide the End-to-End delay not exceed 150 ms. Besides, packet loss rates higher than 1% are not acceptable for live video streaming solutions (AUSTERBERRY, 2005; CHOW *et al.*, 2009). In some applications with high scenes variability, such as football, it was reported (MWELA, 2010) that subjects already become uncomfortable for packet loss rate slightly above 0.3%. Finally, meeting all QoS requirements does not necessarily guarantee high(est) user QoE. Devices' operating system, hardware, battery, operator pricing, light, people around the user and emotion are some examples of factors that impact the users' experience (TRESTIAN *et al.*, 2018; BARAKOVIĆ; SKORIN-KAPOV, 2013).

Our proposed approach to overcome above-mentioned challenges is applying path-aware strategies, protection mechanisms, and content-aware methods to the video streaming process. Path-aware scheduling approaches employ network environment knowledge to deliver better performing traffic distribution and overall content transmission quality, whereas protection mechanisms decrease data loss rate, and content-awareness leverage video features to prioritize specific frames.

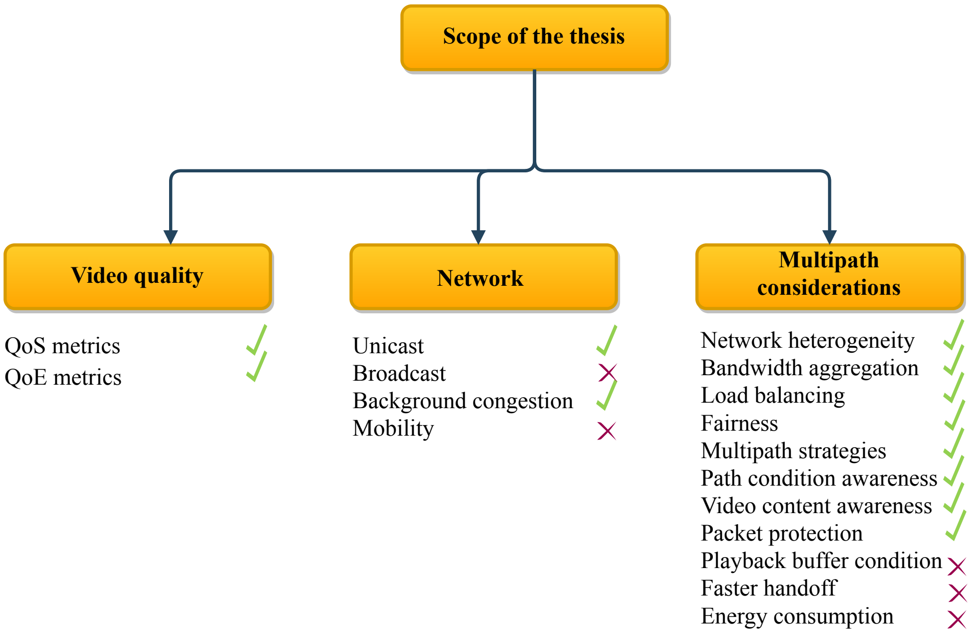


Figure 5 – Investigated scope considered in this thesis.

1.3 Objectives and Methodology

The objective of this thesis is developing novel multipath strategies to improve MMT protocol for video streaming applications over heterogeneous wireless networks in order to provide high quality of experience. In particular, from the challenges explained in Section 1.2 and illustrated in Figure 4, we focus on improving goodput, reducing the amount of packet losses and decreasing delay, which are the main QoS performance metrics impacting the user perceived quality of experience. In this way, the network path heterogeneity is among the main challenges of this work when attempting to properly leverage multipath delivery.

Figure 5 provides further details of the scope of this thesis. For example, the investigation of mobility is beyond the scope of this thesis. Certain topics related to multipathing are also out of the scope of this thesis. For example, faster handoff between multiple networks and energy consumption are not part of our analysis. Besides, the analysis of client-side playback buffering for video streaming such as buffer starvation or bursty packet arrivals is also outside the intended scope of this thesis.

This thesis concerns the implementation and evaluation of the proposed multipath strategies for unicast video streaming over MMT protocol. The protocol stack in scope is illustrated in Figure 6. The proposed approach is implemented at the application layer and, therefore, it does not require any change in the protocol itself since the scheduler is implemented as part of the client/server applications. Besides, the proposed approach uses the native MMT signaling messages to provide the required information for the scheduling strategy. To this end, the main research and development activities can be presented as

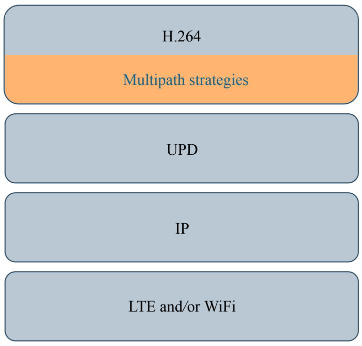


Figure 6 – Proposed protocol stack.

follows:

- **Design of advanced MMT multipath strategies**
  - **Content-Aware and Path-Aware scheduling strategy (CAPA):** understanding how to collect paths' characteristics, estimate path conditions and distribute packets adaptively over different paths based on per-path estimated conditions, in addition, understanding how to consider video content features to distribute packets over different paths, altogether resulting in the proposal of a novel Content-Aware and Path-Aware scheduling strategy for MMT.
  - **Discard strategy:** investigating how discard strategies to drops specific packets (for example, the packets which would overdue considering timed media) in order to better use of the available bandwidth and improve QoE. Consequently, proposing a discard strategy for MMT.
  - **Adaptive bit rate variation method:** understanding how adaptive bit rate variation methods divide video bit rate in a way that better paths receive higher bit rate and paths under bad condition receive lower bit rate workloads. As a result, proposing an adaptive bit rate variation method for MMT.

- **Prototype Implementation:** software development of the multipath scheduler which is the main module of this proposed approach. The scheduler should monitor the paths' conditions and access the video frame type of each video packet. The prototype implementation should support the proposed content-aware and path-aware scheduling strategy as well as the discard strategy and the adaptive bit rate variation method. In this thesis, we opt for an implementation based on DCE (Direct Code Execution), (NS3, 2016) a framework to execute applications within NS-3 without changing their source code.

- **Experimental Evaluation:** through a rich set of use case scenarios we experimentally evaluate the impact of the proposed method. Use case experiments are simulated using NS-3 to investigate the impact of the proposed approach on the quality of delivered video under varying realistic scenarios.

- **Analysis of the results:** analyzing our proposal results for different combinations and factors based on important performance metrics such as goodput, delay, losses, and QoE objective video quality metrics such as PSNR and SSIM.

1.4 Contributions

The main intellectual contributions of this thesis are the proposed multipath wireless methods for video streaming over MMT to address the open challenges and deliver a series of benefits (discussed in Sec. 1.2 by exploiting (*i*) multipath connectivity, (*ii*) path condition estimation, and (*iii*) video content characteristics).

From the core of this work, we have shown that the results of the proposed methods include (AFZAL *et al.*, 2018):

- **Goodput increase:** employing more than one path for data transmission leads to an aggregate of the available bandwidth over multiple network paths resulting in an effective increase of application goodput. Data becomes ready at the receiver faster and higher video resolution can be reached.
- **Packet loss reduction:** under congested path situations, traffic load balancing allows redistributing part of data to alternate path with better condition. As a result, total packet loss rate due to congested path decreases, leading to more stable video streaming.
- **QoE improvement:** the perceived QoE can be significantly increased due to packet loss reduction. In addition, the proposed content-aware approach also leads to better packet protection for video frames that have higher impact on video quality.
- **Easy implementation and adoption:** since the scheduler is implemented as a compatible module and the standardized MMT signaling messages are re-used to provide feedback information, the proposed approach can be easily implemented without requiring changes to the protocol itself, altogether facilitating the adoption of the proposed solution.

The work within this thesis on literature research, design, implementation, and evaluation have resulted in further notable contributions:

- **Comprehensive survey of the state-of-the-art:** to the best of our knowledge, our survey entitled in 'A Holistic Survey of Multipath Wireless Video Streaming' is the first survey focusing on the multipath transmission of wireless video. The contributions of this survey include new insights, protocol technique discussions, and a taxonomy of existing solutions, contributing with a valuable source of information to researchers and developers in this field.

- **Contribution to the MMT standard:** as the first attempt to improve the MMT standard by adding multipath scheduling support, invention proposed as the collaboration document m44902r1 for the MMT Implementation Guideline (IG) standardization activity in 2018 was adopted for the 4th edition of the ISO/IEC 23008-13 standard. This contribution is valuable because MMT has been already adopted by other standards such as ATSC 3.0 and is expected to be adopted by digital television broadcasting/broadband transmission systems as well as AR, VR, MR devices, smartphones, and tablets.

- **Intellectual property rights:** the invention at the core of the proposed CAPA method has been protected through patent applications (INPI (OLIVEIRA J. F.; AFZAL; TESTONI, 2018) and US patent (OLIVEIRA J. F.; AFZAL; TESTONI, 2019)) and could impact products of different companies implementing the MMT protocol.

1.5 Outline

The remainder of this thesis is organized as follows. Chapter 2, “Literature Review", surveys different multipath video streaming approaches over heterogeneous wireless networks, shedding light on the different alternatives from an End-to-End layered stack perspective, unveiling trade-offs of each approach and presenting a suitable taxonomy to classify the state-of-the-art. Finally, open issues and avenues for future work are discussed. Chapter 3, “Advancing MMT for Multipath Wireless Video Streaming", details our developed solution, the multipath improved MMT system, with related components and strategies. Chapter 4, “Performance Evaluation", presents the evaluation of the proposed solution over heterogeneous wireless networks, and also provides our contribution to tackle the challenge of scheduling. Beside, it presents video sequences and testbed descriptions, evaluation scenarios, results and analysis. Finally, Chapter 5 “Conclusions and Future Work", concludes the thesis by pointing out the main findings, contributions, research issues, and directions.

2 Literature Review

Most of today's mobile devices are equipped with multiple network interfaces and one of the main bandwidth-hungry applications that would benefit from multipath communications is wireless video streaming. Table 1 presents published results on the potential performance gains wireless video streaming when exploiting multiple network paths.

In the literature, several surveys have covered different aspects of multipath data communications in general, such as (QADIR *et al.*, 2015; SINGH *et al.*, 2015; LI *et al.*, 2016b; DOMZAL *et al.*, 2015; ADDEPALLI *et al.*, 2013; HABAK *et al.*, 2015). However, this chapter, to the best of our knowledge, is the first one surveys related state-of-art focusing especially on the multipath transmission of wireless video. The contributions of this chapter include new insights, protocol technique discussions, and a taxonomy of existing solutions, altogether serving as a valuable source of information to researchers and developers in this field. More specifically, we focus on the data plane problem of how to schedule data on multiple paths and intentionally leave out works on multipath routing, i.e. the control plane aspects of how to compute routes such as multipath proposals (BARAKABITZE *et al.*, 2018a; HERGUNER *et al.*, 2017) based on Software-Defined Networking (SDN) (KREUTZ *et al.*, 2015). Readers interested in those aspects are referred to recent surveys (QADIR *et al.*, 2015; SINGH *et al.*, 2015) covering control plane approaches. We also leave out of scope wireless sensor networks and refer to surveys on multipath video streaming in this type of networks (HASAN *et al.*, 2017; SHA *et al.*, 2013). Peer-to-peer (P2P) video streaming applications, which have been surveyed in-depth (ZHANG; HASSANEIN, 2012; LIU *et al.*, 2008), are also not in the scope of this chapter.

In the following, we provide a brief overview of the most related and recently published surveys on multipath data communications (QADIR *et al.*, 2015; SINGH *et al.*, 2015; LI *et al.*, 2016b; TRESTIAN *et al.*, 2018), summarized in Table 2.

Qadir et al. (QADIR *et al.*, 2015) investigated multipathing for data in general, mainly on the network layer. Besides that, they have also investigated multipath transmission on the transport layer. Their investigation is organized by discussing key aspects of network-layer multipathing: 1) route computation (source routing, hop-by-hop routing, overlay routing, and SDN-based routing); 2) routing metrics (e.g., delay, bandwidth); 3) load balancing techniques (static or dynamic); 4) number of paths to use; 5) how to use multiple paths together.

Singh et al. (SINGH *et al.*, 2015) covered multipathing for data communications in general, studying the fundamentals of multipath routing, multipath computation algo-

Table 1 – Published results from selected works on multipath video streaming.

Publication	**Network environment**	**Protocol/ feature**	**Performance improvements compared to single path**
MRTP (MAO *et al.*, 2006)	Mesh ad hoc network with high burst loss	RTP	PSNR gains of 1.26 dB more in multipath than single path, 64.14% loss rate reduction together with making packet losses more random.
MPRTP (SINGH *et al.*, 2013)	Two 3G links with bandwidth variations	RTP	In the quick bandwidth scenario, PSNR is better than the single path with 0.5% and 1.0% loss rate. In the slow bandwidth change scenario, it is comparable to the single path with 1.0% loss rate.
RTRA (XING; CAI, 2014)	WiFi and Bluetooth networks with bandwidth variations	DASH	RTRA shows better results for both slow and rapid changing bandwidth scenarios in terms of startup delay (reduced up to half), playback fluency average (no segment missing in multipath but high misses in some single path scenarios), playback quality (PSNR improved 1 to 3 dB), quality switch (up to 4 times reduction), and bandwidth utilization.
MPLOT (SHARMA *et al.*, 2008)	Wireless mesh network with burst loss rate of 50%	TCP	MPLOT archives 75%, with a mean of 50%, more bandwidth aggregation compared to the single path.
Apostolopoulos et al. (APOSTOLOPOULOS, 2000)	Burst lossy wireless network	IP source routing/relay	While the proposed approach results in drops of only 1.5 to 7 dB, but single path drops of 12 to 15 dB.

Table 2 – Related surveys on multipath.

Reference	**Year**	**Scope**	**Comments**
Qadir et al. (QADIR *et al.*, 2015)	2015	Control and data plane	Multipath for data in general, Focus on network-layer multipath solutions
Singh et al. (SINGH *et al.*, 2015)	2015	Control and data plane	Multipath for data in general, Limited research on video streaming services
Li et al. (LI *et al.*, 2016b)	2016	Data plane	Multipath for data in general. Regarding video streaming, relevant aspects not covered
Trestian et al. (TRESTIANet al., 2018)	2018	Data plane	Multimedia delivery solutions following three key directions: adaptation, energy efficiency and multipath limited to MPTCP and SCTP/CMT
Current Survey	2018	Data plane	Multipath investigation mainly for video streaming

rithms, multipath forwarding algorithms, and traffic splitting algorithms. The work also reviews various multipath protocols following a layer-based structure, from the application layer to the physical layer.

Li et al. (LI *et al.*, 2016b) investigated multipath solutions for data in general and presented research problems at various protocol layers including cross layer approaches. Although some video streaming multipath solutions are discussed in the survey, many of the video streaming specific issues are not considered (e.g., importance and influence of video content). In addition, the work does not cover multipath attempts on key video streaming protocols, e.g., Dynamic Adaptive Streaming over HTTP (DASH), and MPEG Media Transport (MMT).

As a related survey, we should also consider Trestian et al. (TRESTIAN *et al.*, 2018), which is a survey on seamless multimedia delivery within a heterogeneous wireless networks environment. The authors evaluated three key aspects of multimedia delivery: adaptation, energy efficiency, and multipath delivery. Regarding the latter, only proposals based on the Multipath TCP (MPTCP) and Stream Control Transmission Proto-

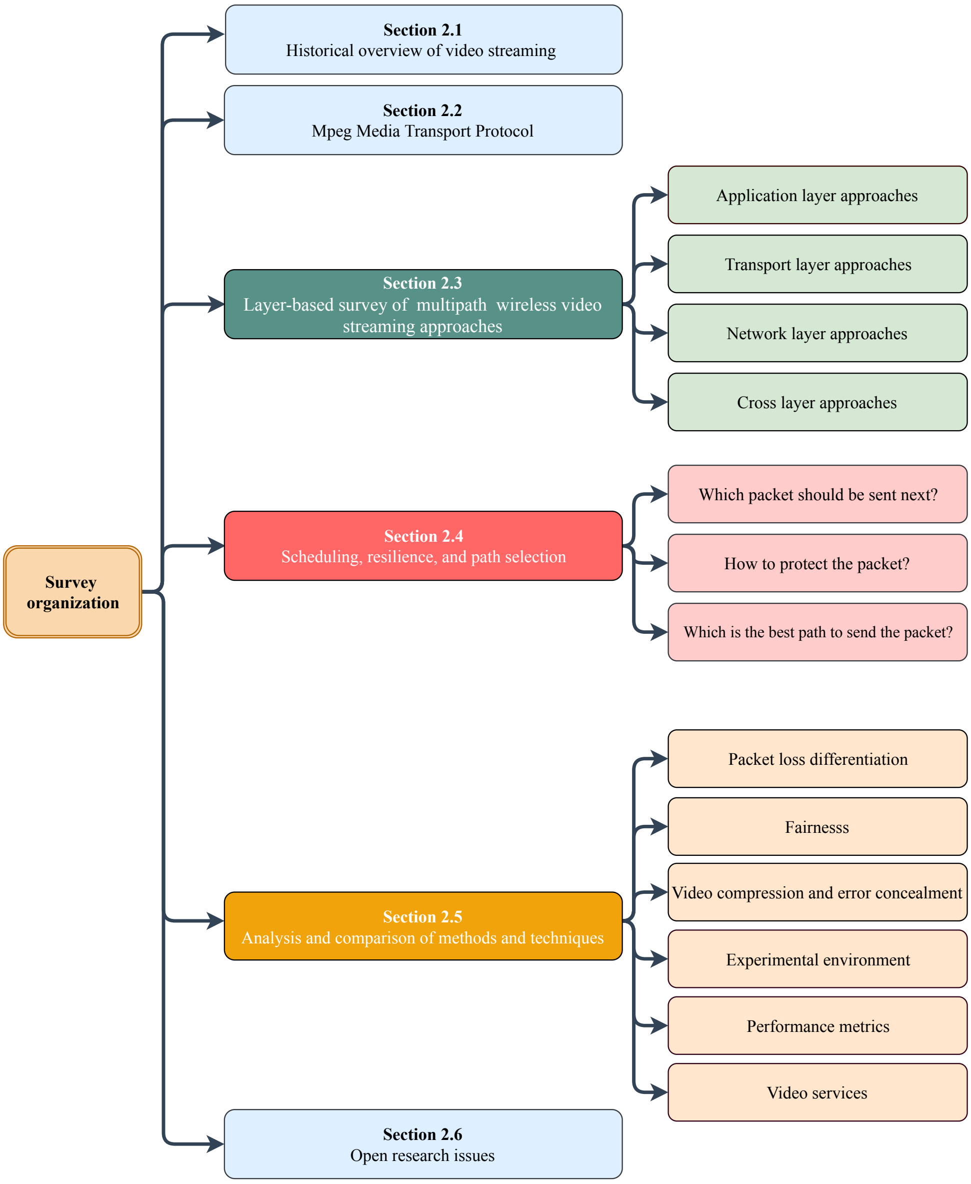


Figure 7 – Visual representation of the organization of the chapter.

col (SCTP)/Concurrent Multipath Transfer (CMT) are studied.

Differently from existing surveys, the survey presented in this chapter is centered around multipath methods for wireless video streaming to mobile devices. In this chapter, mobile means wireless communications which may or not include mobility scenarios (e.g., a wireless laptop at home or a wireless laptop on a train). In this chapter, we cover in-depth relevant approaches and new techniques in the field. We majorly categorize existing works considering two main aspects. The first aspect relates to the protocol layer perspective of each work: application layer, transport layer and/or network layer. We sub-group each layer approach based on which standard protocol/feature is used in the schemes proposed by the authors. Such classification is beneficial to understand the advantages, drawbacks, and trade-offs of each layer and protocol/feature. We also indicate which part of the network (server and/or client) requires adjustment in order to become compatible with the multipath transmission approach. In the second aspect, we analyze the approaches based on the specific scheduling functions to transmit video data over wireless link technologies. The works are classified according to the following scheduling functions: packet selection, packet protection, and path selection. In addition to these two aspects, we also discuss primary research problems related to multipath video transmission, such as network heterogeneity, out-of-order packets, Head-of-Line (HOL) blocking, End-to-End delay, overdue packets, implementation aspects, and pros and cons of each approach.

The high-level organization of this chapter is illustrated in Figure 7. An overview of video streaming protocols is provided in Section 2.1. The benefits and challenges of adding multipath transmission in the video streaming scenario are presented in Section 1.2. Research works are then initially introduced in Section 2.3 and classified based on the protocol layer stack position and on the used protocol/feature. In Section 2.4, the works are then investigated based on the scheduling functions: choice of the next packet to be transmitted (packet selection), data packet protection method (packet protection), and selection of the proper network channel (path selection). Section 2.5 provides additional information about the research works that may also be of interest for the reader, such as packet loss differentiation, fairness consideration, video codecs, the employed network simulator, performance metrics, and video services. Section 2.6 presents research issues and directions. Finally, Section 2.7 provides concluding remarks. There is also a list of abbreviations in the appendix to help readers track them easily.

2.1 Historical Overview of Video Streaming

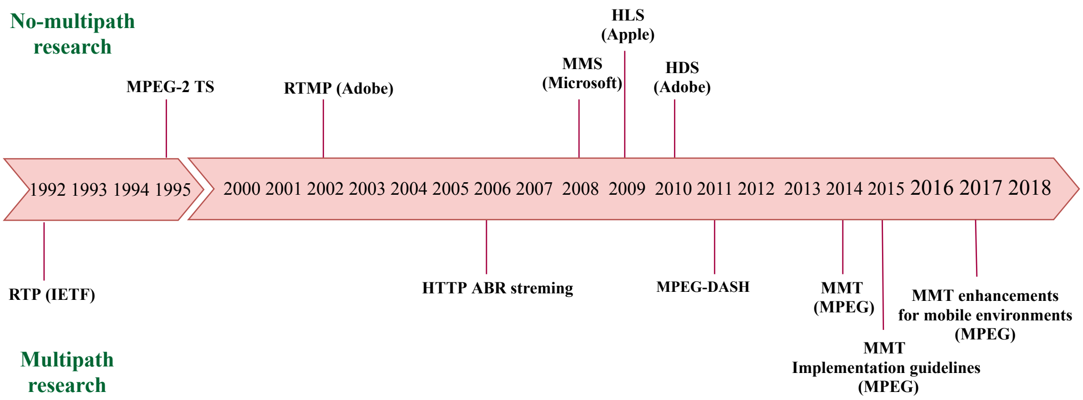


Figure 8 – Historical overview of video streaming protocols.

This section provides a general picture of video streaming development as presented by the timeline and milestones in Figure 8. Interested readers are referred to (YUSTE *et al.*, 2015) for a deeper review of different MPEG standards.

The first widely used video streaming protocol is the Real-time Transport Protocol (RTP) (SCHULZRINNE, 1992), which was initially released in 1992 by IETF. It is a UDP-based protocol used for unidirectional real-time video streaming. The advantage of this protocol is that it has very low overhead and works well in managing IP networks. However, RTP has also some disadvantages. For example, it requires a payload format for each media type or codec (LIM *et al.*, 2014), suffers from lack of multiplexing and has limited support for non-real-time video. Another disadvantage of RTP is that many CDNs do not support it because the server must manage a separate streaming session for each client, turning large-scale deployment more resource intensive. Moreover, RTP cannot traverse firewalls and is connectionless. Therefore, RTP is generally employed for private managed networks where the number of packet losses is small, such as pay-TV cable networks. More technical details on RTP will be provided in Section 2.3.1.

The next widely known and adopted video streaming protocol shown in Figure 8 is the MPEG-2 Transport System (MPEG-2 TS) (ISO/IEC, 2007). It has been widely used since 1995 in digital broadcasting, mobile broadcasting systems and streaming over the Internet. Several standards have also adopted this protocol, such as the Terrestrial Digital Multimedia Broadcasting (T-DMB), the Digital Video Broadcasting Handheld (DVB-H), the Advanced Television Systems Committee (ATSC) and the Internet Protocol TeleVision (IPTV) (YIE; LEE, 2016). MPEG-2 TS is not only a format for fast and reliable packetized streaming delivery but also a format for storage. In addition, MPEG-2 TS is fully codec agnostic.

Since the requirements for on-demand and personalized video delivery over the Internet have dramatically increased, it became challenging for MPEG-2 TS to achieve the high requirements of broadcasting over IP (LIM *et al.*, 2014; LIM *et al.*, 2013). For example, MPEG-2 TS is not appropriate for UHD delivery over packet networks due to the pre-multiplexing mechanism, not flexible packetization and small-fixed packet size (188 bytes).

The next protocol in the timeline of Figure 8 is the Real-Time Messaging Protocol (RTMP) (PARMAR; THORNBURGH, 2017). It is an Adobe proprietary protocol standardized in 2002 that was initially developed by Macromedia. RTMP is a TCP-based protocol used for bidirectional video streaming. This protocol provides the advantage of multiplexing capability but requires flash player plugin. All video and audio files must be sent in a Small Web Format (SWF) (ADOBE, 2012) file to make it able to play with flash player. Another disadvantage is that RTMP suffers from not being codec agnostic and for not supporting some newer video codecs, such as High Efficiency Video Coding (HEVC) (ITU-T, 2013). Yet another disadvantage is that it is blocked by firewalls and not supported by all Content Delivery Networks (CDNs).

Multiple types of RTMP protocols were designed: RTMPS, RTMPE, RTMPT and

RTMFP. RTMPS is an encrypted RTMP over a Transport Layer Security (TLS)/Secure Sockets Layer (SSL) connection. RTMPE uses Adobe's own encryption security mechanism. RTMPT is used to encapsulate RTMP, RTMPS, or RTMPE packets within Hypertext Transfer Protocol (HTTP) (FIELDING *et al.*, 1999) requests in order to traverse firewalls. Finally, RTMFP is used to replace RTMP over User Datagram Protocol (UDP) (POSTEL, 1980) instead of TCP (POSTEL, 1981).

The next highlighted point in the timeline of Figure 8 is not a protocol, but a video streaming technique introduced in 2006 that become highly adopted in the subsequent streaming protocols. Adaptive Bit Rate (ABR) streaming was introduced by Move Networks (MOVENETWORKS, 2017; SEUFERT *et al.*, 2015) and is over HTTP (FIELDING *et al.*, 1999) with some adaptations sequentially described. At the server side, the video sequences are stored in various resolutions (bit rates) and are fragmented into small segments. The streaming logic is located on the client side and it is responsible for selecting the suitable segment by considering different parameters (e.g., bandwidth availability, media playout situations (SEUFERT *et al.*, 2015)). Without such a flexible service, if only one bit rate video is available, a smaller video bit rate than the network bandwidth would lead to a smooth video but waste available resources. On the other hand, a higher video bit rate than the bandwidth network would impose delay.

One advantage of HTTP-based video streaming solutions is that they are easy to deploy in the current Internet architecture. In addition, HTTP flow can traverse middleboxes, such as firewalls, Network Address Translators (NATs) and network proxies. Moreover, the client can manage the streaming without the need to maintain a session state on the server, thus it improves scalability and servers can supply a large number of clients at no additional cost (SODAGAR, 2011). Therefore, HTTP is supported by most of CDNs (JAMES *et al.*, 2016) and the interest in using HTTP for video streaming has been significantly increased in recent years. In 2015, a new version of HTTP, namely HTTP/2, was standardized (BELSHE; PEON ROBERTO THOMSON, 2015) and received the attention of researchers in the multimedia community (YAHIA *et al.*, 2017; HUYSEGEMS *et al.*, 2015; XIAO *et al.*, 2016). The results show that advancing video streaming services, which are built on top of HTTP/1.1, to HTTP/2 could improve the video quality and performance. More technical details on HTTP/2 will be provided in Section 2.3.1.

The initial HTTP Adaptive Streaming (HAS) (SEUFERT *et al.*, 2015) commercial successful protocols were the Microsoft Silver light Smooth Streaming (MSS) (BOCHAROV, 2017) developed by Microsoft in 2008, the HTTP Live Streaming (HLS) (PANTOS; MAY, 2017) developed by Apple in 2009 and the Adobe HTTP Dynamic Streaming (HDS) (ADOBE, 2017) developed by Adobe in 2010. Since all these protocols were proprietary and incompatible, in 2011, the Dynamic Adaptive Streaming over HTTP (MPEG-

DASH) (MPEG, January 2011) protocol was developed to become a unified codec agnostic standard (MUELLER, 2017). MPEG-DASH flexible delivery and codec agnostic properties have turned it into a successful protocol widely adopted by content providers (BITMOVIN, 2017). For example, Netflix and YouTube are currently using MPEG-DASH with Hypertext Markup Language (HTML) as their core streaming technology (BITMOVIN, 2017). Another advantage is that DASH supports both multiplexed and unmultiplexed encoded content. However, the protocol has some disadvantages. For instance, DASH cannot support low latency delivery because the server requires waiting until the completion of movie fragments or of the whole file before transmission (LIM *et al.*, 2014). Besides, the inaccurate bandwidth estimation, especially in mobile networks, causes several switches, freezes, and poor QoE for DASH (YAHIA *et al.*, 2017). More technical details on DASH are provided in Section 2.3.1.

All ABR-based protocols employ similar versions of the previously explained technology and support live and video on demand (VoD) delivery. The differences rely on technical parameters (SEUFERT *et al.*, 2015). For instance, MPEG-2 TS is used by HLS/DASH while ISO Base Media File Format (ISOBMFF) is used by DASH and fragmented MP4 is used by HDS/MSS. Regarding video codecs support, H.264 and H.265/HEVC are supported by all HAS protocols. In addition, MSS also supports VC-I (KALVA; LEE, 2007), HDS also supports VP6 (RAO *et al.*, 2014) and DASH supports all video codecs. Finally, segment lengths are specified as 2 seconds for MSS, 10 seconds for HLS, 2-5 seconds for HDS and are not specified for DASH. More details on HAS protocols are discussed in (SEUFERT *et al.*, 2015).

The next protocol in the timeline of Figure 8 is the MPEG Media Transport (MMT) (MPEG, 2014). It was standardized by MPEG in 2014 considering recent changes in multimedia delivery and requirements for Internet technologies, such as IP and HTML for Internet-based video streaming solutions (LIM *et al.*, 2014). This protocol also supports UHD resolution and HEVC video codec. MMT was designed to inherit some MPEG-2 TS features, such as content agnostic media delivery, easy conversion between storage and delivery format and support of multiplexing. In addition, MMT was developed due to a need for an international standard to support hybrid delivery in various heterogeneous network environments. Then, in 2015, implementation guidelines standardized to provide technical guidelines for implementing and deploying MMT systems.

The last highlight point in the timeline is MMT enhancement for mobile environment specifying multipath support which has already been added to the protocol and standardized (MPEG, 2017). MMT was adopted by some recent standards, such as the ATSC 3.0 (YE *et al.*, 2016), which is a recent standard with a hybrid delivery model which includes MMT and DASH. Especially, in ATSC 3.0, MMT protocol (MMTP) is proposed for broadcasting, and DASH over HTTP is proposed for broadband service.

One important difference between DASH and MMT is that, typically, DASH supports a client driven Quality of Service (QoS) control standard, while MMT supports a server driven QoS control services (YE *et al.*, 2016). More technical details on MMT are provided in Section 2.3.1.

To that end, multipath has been investigated in some of the above mentioned video streaming protocols but not in all of them, especially it has not investigated for the proprietary protocols due to their closed and incompatible design (Figure 8).

Commercial services. We already mentioned some companies using their own developed proprietary protocols such as Move Networks, Microsoft (MMS), Apple (HLS) and Adobe (RTMP, HDS). Besides them, there are other company services adopting or in the process of developing streaming solutions. For example, Skype and WhatsApp are mobile application platforms providing video calls or video conferences for their users. These services use RTP for video streaming (KARYA *et al.*, 2018). Hulu (HULU, 2019) is an online video service providing on-demand shows, movies, documentaries, and more. Hulu requires flash player for video streaming through the RTMP protocol (TRESTIAN *et al.*, 2018).

A number of service providers use DASH. Among the most famous ones are YouTube (YOUTUBE, 2019), Netflix (NETFLIX, 2019a), Twitch (TWITCH, 2019) and Vimeo (VIMEO, 2019). YouTube (YOUTUBE, 2019) is a video-sharing website providing live and on-demand video streaming. Netflix (NETFLIX, 2019a) allows watching on-demand movies and Twitch is the world's leading live streaming platform for gamers. Vimeo (VIMEO, 2019) also provides free video viewing services. Another commercial streaming service is Bitmovin, which provides adaptive streaming supporting MPEG-DASH and HLS (BITMOVIN, 2019). Generally, DASH has gotten broad support from commercial companies – see DASH Industry Forum member list available in (DASHIF, 2019). In addition, browsers such as Chrome and Firefox also support DASH (BITMOVIN, 2017).

NHK, Nippon Hoso Kyokai, is a Japan's telecommunication company (public service broadcaster) uses MMT as the protocol of choice for 4K/8K Super Hi-Vision.

2.2 MPEG Media Transport Protocol

MPEG Media Transport (MMT) for heterogeneous environments is developed as a part of the ISO/IEC MPEG-H standard suite (LIM *et al.*, 2014). An overview of the MPEG-H standard is shown in Figure 9. The MPEG-H functional areas are HEVC, 3D audio, and MMT. The MMT standard consists of parts 1, 4, 7, 10, 11 and 12. This application layer transport protocol supports VoD and live video streaming. MMT has been widely used for Virtual Reality (VR) and Augmented Reality (AR) technologies, three-dimensional (3-D) scene communication, Multi-View Video (MVV), and for major advances in televisual technology worldwide (BAE, 2013; AOKI, 2017). We previously explained some of the properties and behaviors of MMT in Section 2.1. Here, firstly, we indicate more properties of the protocol. Then, we explain the related technologies, MMT functions, and data transmission details. Multipath support of MMT is discussed later in Section 2.3.1.3.

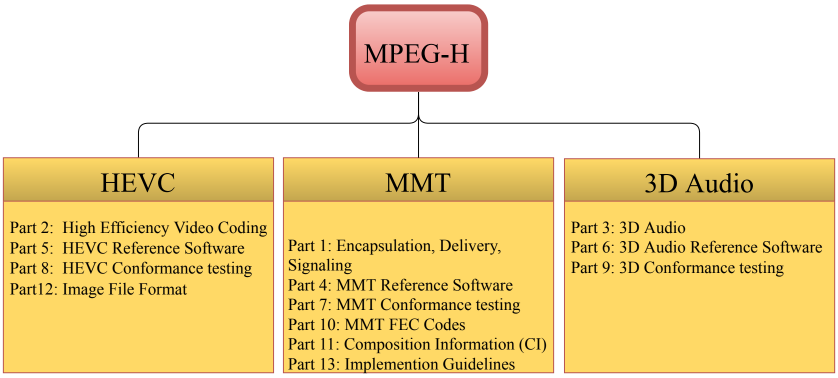


Figure 9 – MPEG-H Overview (source: adapted from (JUNG *et al.*, 2015)).

MMT properties. MMT could be used for all unidirectional, bidirectional, unicast, multicast, multisource and, even, multipath media delivery. Besides, MMT supports both broadcast and broadband video streaming. It also provides traditional IPTV broadcasting service and all-Internet Protocol (All-IP) networks.

Capability of hybrid media delivery is one of the most important properties of MMT. Hybrid media delivery (JUNG *et al.*, 2015) refers to the combination of delivered media components over different types of network. For example, it could be one broadcast channel and one broadband, or it could be two broadband channels. MMT has different hybrid service scenarios that are classified into three groups by ISO/IEC 23008-13 (MPEG, 2013): live and non-live, presentation and decoding, and same transport schemes and different transport schemes. The first one, live and non-live, refers to the combination of live streaming components (Figure 10(a)) or combination of live with prestored components (Figure 10(b)). The second group, presentation and decoding, is the combination of the live stream components for synchronized presentation (Figure 10(a)) or synchronized decoding (Figure 10(c)). The third group, same transport schemes and different transport schemes, supports the combination of just MMT components or MMT components with another format components (e.g., MPEG-2 TS). An instance of hybrid model comprises of MMT (as a broadcast channel) and DASH (as a broadband channel) over heterogeneous networks is also presented in ISO/IEC 23008-13 (MPEG, 2013). The work in (LI *et al.*, 2016a) compared two MMT broadband systems: a combination of MMT with HTTP versus a combination of MMT with Quick UDP Internet Connection (QUIC) (HAMILTON *et al.*, 2016). QUIC is a transport protocol atop UDP for broad-

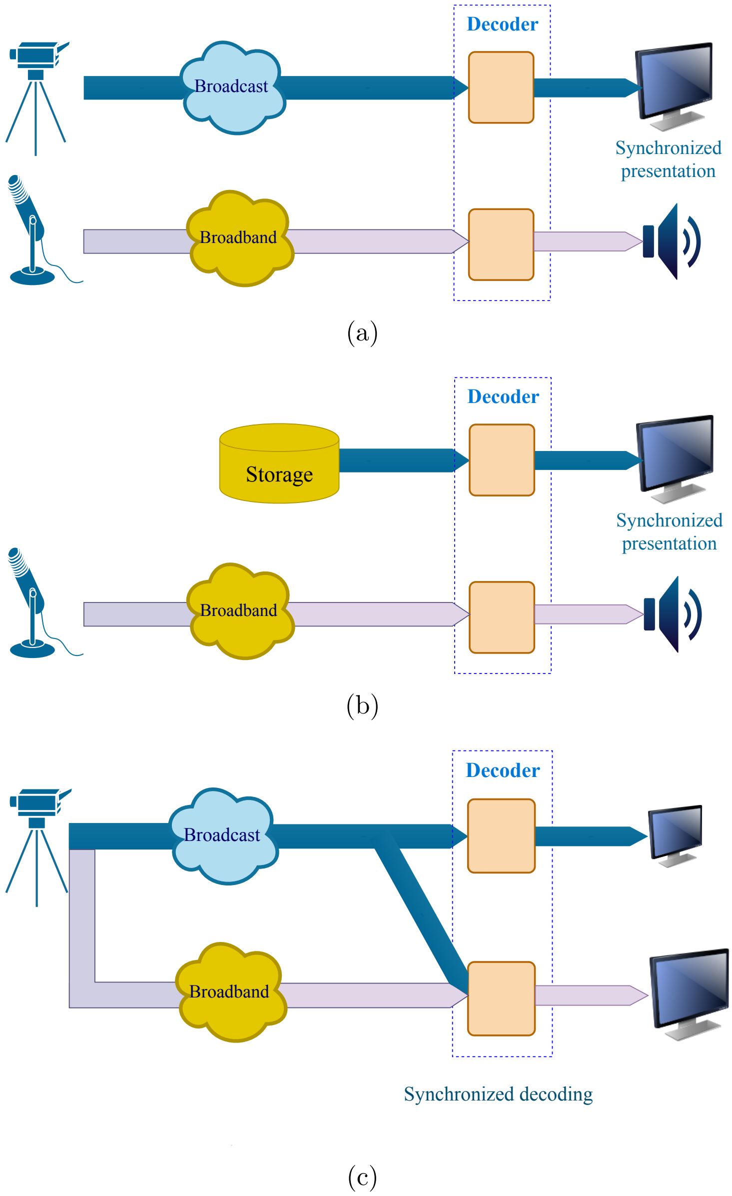


Figure 10 – Hybrid delivery examples (source: adapted from (JUNG *et al.*, 2015)): (a) Combination of streaming components for presentation (b) Combination of streaming component with pre-stored component for presentation (c) Combination of components for decoding.

band systems, which was developed by Google. QUIC aims to reduce latency because it has zero round trip connection establishment in many cases. For example, when the client talked to the server before and there is some cached context (repeated connections). In addition, QUIC has multiplexed transport with no HOL blocking. Other features of QUIC are utilizing congestion control, FEC protection, and its own retransmission mechanism. The results of (LI *et al.*, 2016a) show that using QUIC is more appropriate than HTTP in the networks with high delay and lossy networks. This experiment is just for a single path and there is room to evaluate it in multipath transmission defined for QUIC (CHAN *et al.*, 2017).

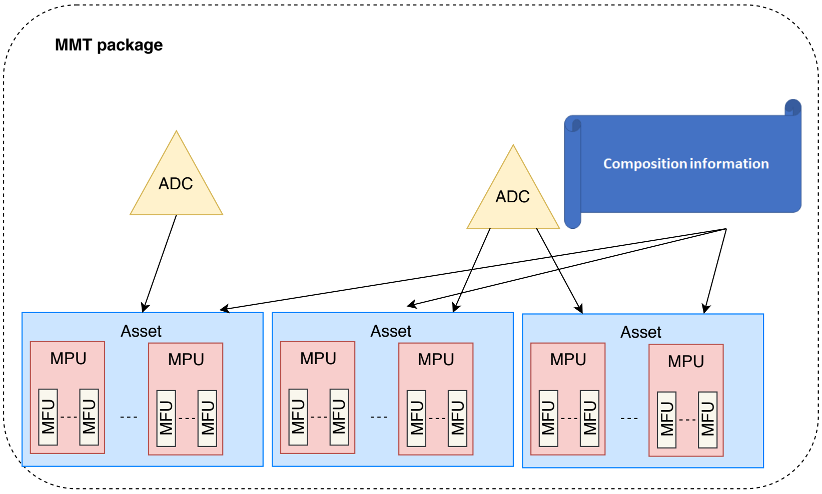


Figure 11 – Logical MMT package (source: adapted from (LIM *et al.*, 2013)).

Related technologies. ISO/IEC 23008-1 (MPEG, 2014) defined some related MMT technologies. For example, Application Layer Forward Error Correction (AL-FEC) to repair data, ARQ to retransmit lost data, MMT data model and built-in hypothetical buffer model.

Regarding the MMT data model (MPEG, 2014; PARK *et al.*, 2017), MMT package is a logical entity, illustrated in Figure 11, that comprises of one or more assets and required information for video delivery, such as Composition Information (CI), Presentation Information (PI) and Asset Delivery Characteristics (ADC). Asset refers to a logical data entity containing a number of Media Processing Units (MPUs). Video, audio, picture, text are some examples of assets. CI provides information on temporal relationships among MPUs written in XML. HTML5 file is referred to PI, which provides initial information on spatial relationships among media elements, and ADC contains QoS information for multiplexing.

Built-in hypothetical buffer model (MPEG, 2014) aims to compensate for jitter and multipath delay delivery. In this model, the sending entity runs the hypothetical receiver buffer model (HRBM) to emulate the receiving entity behavior. In this way, the sending entity determines the required buffering delay and buffer size. Then, sending entity signals this information to the receiving entity. Since at the receiver entity, several buffers exist to reconstruct of MPU from the MMT packets, the received signal is used to define operations of the buffers to ensure that at any time the buffer occupancy is within the buffer size requirement. These buffers are FEC decoding buffer to perform FEC decoding. De-jitter buffer to provide the fixed transmission delay, and MMTP de-capsulation buffer to perform MMT packet processing (e.g., de-encapsulation, de-fragmentation/de-aggregation).

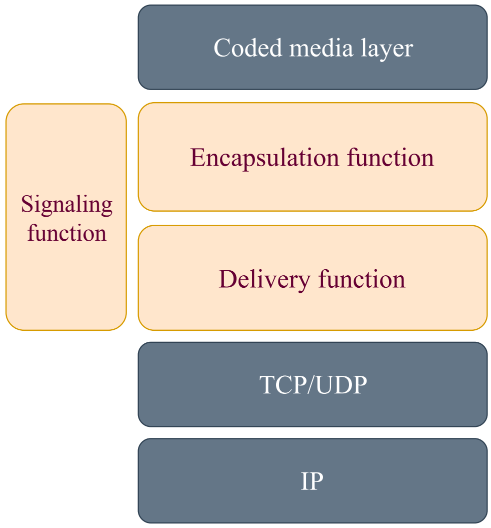


Figure 12 – Major functional areas of MMT (source: adapted from (LIM *et al.*, 2013)).

MMT functions. MMT has three major functional layers, shown in Figure 12, independent of video codecs (JUNG *et al.*, 2015): encapsulation, signaling, and delivery. Encapsulation functional layer is responsible for encapsulating MPUs, which are complied with ISOBMFF (ISO/IEC, 2012). Thereby, it enables easy conversion between storage and delivery format (LIM *et al.*, 2014). MMT is beneficial to the broadcasting system because MPUs are self-contained, which means that they can completely decode at the terminal without requiring any further information. Signaling functional layer is responsible for signaling messages and delivery management (e.g., CI, PI and ADC). Delivery functional area defines the application layer protocol that supports packetized streaming including the payload format through a heterogeneous network environment. Delivery functional area also provides Multiplexing, flow control and cross layer. Cross layer ability provides exchanging QoS between application layer and transport layer.

MMT data transmission. Regarding MMT data transmission, each MMTP session consists of one MMTP flow (JUNG *et al.*, 2015). MMTP flow is defined as all packet flows that are delivered to the same IP and port destination. A MMT flow may carry multiple assets, which are identified with a unique packet_id. MMTP packet uses two types of sequence number as different purposes: packet_counter and packet_sequence_number. packet_counter represents sequence of packets in a delivery session and it is regardless of the value of packet_id. packet_counter enables packet loss detection. However, packet_sequence_number is the sequence number specific to each packet_id (each asset).

Initially, MMT was designed for broadcast networks (over UDP/IP) with reserved channel capacity. Therefore, congestion control was left to the implementation of the senders. However, MMT inherently supports receiver and sender feedback for stream thinning and bitstream switching. It also may support any Receiver-driven Layered Multicast (RLM)-based congestion control algorithms (e.g., WEBRC, TFMCC).

MMT has four modes for payload format; MPU mode to transport MPU packetized streaming, Signaling mode to transfer signaling information, Repair symbol mode to carry FEC repair data, and also Generic File Delivery (GFD) mode, which transports all types of files.

Further, a discussion about multipath support of MMT and description of the related research works attempting to enhance this protocol with multipath capable delivery are provided in Section 2.3.1.3.

2.3 Layer-based Survey of Multipath Wireless Video Streaming Approaches

In this section, the surveyed multipath wireless video streaming works are initially introduced and classified in Table 3 according to protocol stack layers and protocol/features. The table also indicates which parts of the network equipment (whether client, server, network or a combination of them) need to be adjusted in order to become compatible with multipath transmission schemes. Most flexible solutions require only client side modification because they are compatible with the current network infrastructure and does not need any change neither on the server nor in the network infrastructure. On the other hand, there are some other approaches that require server side modification or even both server and client together. Most difficulties are with solutions that they need to adjust network infrastructure.

2.3.1 Application Layer Approaches

Video streaming approaches focused on the application layer have the advantage of accessing player buffer status and relevant video content information, such as frames priorities and coding dependencies. Application-specific information provides the multipath approach with rich inputs to define the video streaming scheduling strategies. One key advantage is that there is no need to change lower layer protocols. However, a big drawback of these solutions is that they commonly require modifications of the video software. In application layer approaches, generally, an application level sequence number is used for loss detection, which often increases the overall protocol overhead. In addition, in order to perform knowledgeable packet scheduling decisions, the application requires a mechanism to estimate the network paths' performance, e.g., through application-specific probes or from TCP congestion control information (LI *et al.*, 2016b).

In this subsection, we discuss relevant works that are based on RTP, DASH, MMT

Table 3 – Classification of research works according to protocol layers.

Protocol layer	Applied protocols/features	Works	Compatibility
Application Layer	RTP	MRTP (MAO *et al.*, 2006)	Server and Client
		MPRTP (SINGH *et al.*, 2013)	Server and Client
		MRTP-AR (LEI *et al.*, 2017)	Server and Client
	DASH	Xing et al. (XING *et al.*, 2012)	Client
		Chowrikoppalu et al. (CHOWRIKOPPALU; GOWDA, 2013)	Client
		RTRA (XING; CAI, 2014)	Client
		Houzé et al. (HOUZÉ *et al.*, 2016)	Client
	MMT	Sohn et al. (SOHN *et al.*, 2015)	Server and Client
		Kolan et al. (KOLAN; BOUAZIZI, 2016)	Server and Client
		Afzal et al. (AFZAL *et al.*, 2018)	Server and Client
	Other Adaptive Streaming Approaches	Evensen et al. (EVENSEN *et al.*, 2010)	Client
		Evensen et al. (EVENSEN *et al.*, 2011)	Client
		Evensen et al. (EVENSEN *et al.*, 2012)	Client
		GreenBag (BUI *et al.*, 2013)	Client
Transport Layer	Multipath UDP	BEMA* (WU *et al.*, 2016a)	Server and Client
		Freris at al. (FRERIS *et al.*, 2013)	Server and Client
		Correia at al. (CORREIA *et al.*, 2012)	Server and Client
	Multipath TCP	MPLOT (SHARMA *et al.*, 2008)	Server and Client
	Multipath DCCP	MP-DCCP (HUANG *et al.*, 2012)	Server
	MPTCP	ADMIT (WU *et al.*, 2016b)	Server and Client
		MPTCP-SD (DIOP *et al.*, 2012)	Server
		MPTCP-PR (DIOP *et al.*, 2012)	Client
		Xu et al. (XU *et al.*, 2016)	Server and Client
		PR-MPTCP$^+$ (CAO *et al.*, 2016)	Server
	SCTP and CMT (extension of SCTP)	Kelly et al. (KELLY *et al.*, 2004)	Not defined
		Okamoto et al. (OKAMOTO *et al.*, 2014)	Server
		SRMT (SILVA *et al.*, 2016)	Server
		PR-SCTP (SANSON *et al.*, 2010)	Server
		CMT-QA (XU *et al.*, 2013)	Server and Client
		CMT-DA (WU *et al.*, 2015)	Server and Client
		CMT-CA (WU *et al.*, 2016d)	Server and Client
Network Layer	SDN	Yap at al. (YAP *et al.*, 2012)	Server (depends on the application), Client and Network
		MARS (SUN *et al.*, 2016)	Network
	Proxy	BAG (CHEBROLU; RAO, 2006)	Client and Network
Cross Layer	Application Layer Decision	Corbillon et al. (CORBILLON *et al.*, 2016)	Server
		Ojanperä et al. (OJANPERÄ; VEHKAPERÄ, 2016)	Server, Client and Network
		GALTON (WU *et al.*, 2015)	Server and Client
		FRA-JSCC (WU *et al.*, 2013)	Server and Client
	Transport Layer Decision	MP-DASH (HAN *et al.*, 2016)	Server and Client
		Nam et al. (NAM *et al.*, 2016)	Server (depends on the application), Client and Network
		CMT-CL/FD (XU *et al.*, 2015)	Server

*BEMA: UDP (for video data transmission) and TCP (for connection establishment and feedback information).

and other adaptive streaming approaches. All of these protocols were previously introduced in Section 2.1 and will be further detailed here. Figure 13 illustrates the protocol stack position of these protocols and Table 3 presents each category.

2.3.1.1 RTP

The Real-time Transport Protocol (RTP) used for unidirectional real-time video streaming was first published in 1992 (SCHULZRINNE, 1992) and then updated in RFC 1889 (SCHULZRINNE *et al.*, 1996), later obsoleted by RFC 3550 (SCHULZRINNE *et al.*, 2003). The newest protocol specification is RFC 8108 (LENNOX *et al.*, 2017) published in 2017. RTP is an application layer transport protocol that provides End-to-End network transport functions to support live, on-demand, and interactive multimedia applications.

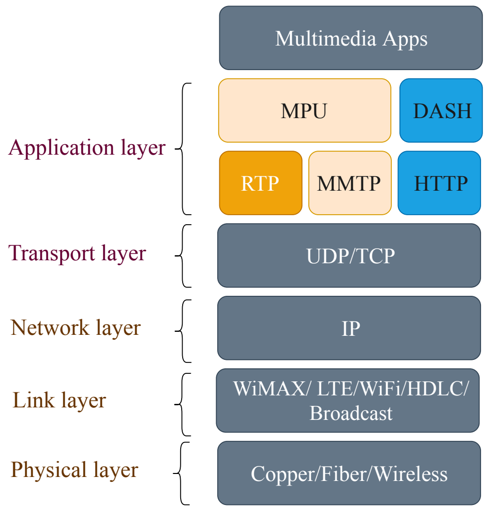


Figure 13 – Application layer protocols position in a network protocol stack.

Next, we highlight more properties of the protocol, and then we survey multipath works based on RTP.

RTP Properties. Although RTP is designed to run over UDP, it could also carry data over other transport protocols such as TCP or SCTP. Another property of RTP is that it can be used in conjunction with the RTP Control Protocol (RTCP) to send monitored information and QoS parameters periodically. RTP also can be used in conjunction with other protocols, such as Real-time Streaming Protocol (RTSP) (SCHULZRINNE, 1998), which is used to control multimedia playback. A big problem of RTP, running over UDP, is that it lacks congestion control and it is unfair to give room to other flows. There is also no guarantee of reliable delivery and it needs a method to protect high priority frames (I frames). Furthermore, a challenge to improve RTP to support multipath streaming is that RTP establishes at the media session level and receiver reports per media (video or audio) flow (SINGH *et al.*, 2013).

Multipath support. Multiflow Real-Time Transport Protocol (MRTP) (MAO *et al.*, 2006), Multipath RTP (MPRTP) (SINGH *et al.*, 2013) and Multipath Real-Time Transport Protocol Based on Application-Level Relay (MPRTP-AR) (LEI *et al.*, 2017) improved RTP to support multipath video streaming.

The works MRTP and MPRTP are Constant Bit Rate (CBR) approaches. Since RTP lacks congestion control, a considerable receiver buffer is required to compensate the different path latencies of RTP streams when playing a CBR video (SINGH *et al.*, 2015). Both MRTP and MPRTP use QoS reports (e.g., sender report and receiver report), similar to RTCP reporting in RTP, to carry periodic per flow and session statistics. The time interval between reports is set by the application in MRTP, and can be adapted based on network conditions by the receiver.

The goal of Multiflow Real-time Transport Protocol (MRTP) (MAO *et al.*, 2006) is to remedy the failure and congestion in mobile wireless ad hoc networks. It is claimed by its authors that the approach is also applicable to the Internet. MRTP is used in conjunction with the Multi-flow Realtime Transport Control Protocol (MRTCP). Inherently, MRTP/MRTCP is an extension of the RTP/RTCP to support media delivery over multiple wireless networks. Unlike RTP, MRTP is a session-oriented protocol. Therefore, MRTCP establishes the session in a three handshake to exchange information (e.g., available paths). Data transmission could be over UDP/TCP/SCTP and during transmission, it is possible to add or remove paths based on the QoS reports. In particular, media divides into flows, and each flow is for one path (in MRTP, the concept of flow is used for series of video packets which are transmitted through an individual path). MRTCP manages flows by utilizing ADD/DELETE acknowledgments (ACKs) for flows.

QoS reports are transferred through the best path or multiple paths to guarantee reliable delivery. These reports are useful for the sender to adapt to transmission errors. For example, by adding redundancy to increase error resilience and by assigning data to more proper paths. There is a reassembly buffer at the receiver side to compensate jitter, reorder and reassemble packets by utilizing session ID, flow ID and flow sequence number.

MRTP uses a retransmission mechanism to retransmit packets to cope with unreliable UDP/IP. The timeout value for retransmission is set by RTT and the maximum number of retransmissions is set by the application. Different error control schemes, including Forward Error Correction (FEC), Multiple Description Coding (MDC) or Automatic Repeat reQuest (ARQ) could incorporate with MRTP. Finally, the results of the research work show that MRTP outperforms single path RTP on received video quality.

In MRTP, it is possible to choose the data distribution method. For example, it could be just a simple Round Robin, striping (over multiple servers), layered coding, multiple description coding or object-oriented coding (video or audio objects encode individually).

Multipath RTP (MPRTP) protocol (SINGH *et al.*, 2013) is a RTP extension with multipath transmission for real-time media. The target of MPRTP is minimizing latency. Initially, the scheduler distributes equal traffic rate to each path and then after gathering information about the path characteristics, it recalculates the data distribution for each path. It uses RTCP to monitor and control information (e.g., jitter and packet loss). As a result, paths are categorized as congested, mildly congested, and non-congested conditions based on the packet loss information. The scheduler, which is responsible for packet distribution over different paths, assigns more media data on the non-congested path and fewer media data on congested ones. I frames have the highest priority and are transferred over the path with the highest bandwidth, the least delay and packet losses. The sender is informed to retransmit packets by NACK and also retransmits packets on the path with the highest bandwidth, least delay and packet losses.

The approach is not integrated with congestion control but tries to keep the load balancing by using network characteristics. The authors developed a de-jitter algorithm at the receiver side to overcome the variation of RTT and packet reordering with an adaptive playback buffer. An MPRTP sender assigns a subflow ID to each path (in MPRTP, the concept of subflow is used for series of video packets transmitted over an single path) and subflow-specific sequence numbers to determine subflow-related packet jitter, packet loss, and packet discards at the receiver side. The approach is less unfair than RTP with the aim of system balancing and spreading data over paths.

Recently, Multipath Real-Time Transport Protocol Based on Application-Level Relay (MPRTP-AR) (LEI *et al.*, 2017) was defined by IETF. As shown in Figure 14, the proposed MPRTP-AR protocol stack has two sub-layers: RTP sub-layer and multipath transport control (MPTC) sub-layer. The RTP sub-layer helps this protocol to be fully compatible with existing RTP applications. Therefore, there is no need to change the Application Programming Interface (API). The MPTC sub-layer is responsible for functions such as flow partitioning, subflow packaging and recombination, and also subflow reporting.

At the sender side, data from the application layer are formatted in RTP packets which are sent to the MPTC sub-layer. Then, MPTC formats them into MPRTP-AR data packets. At the receiver side, MPTC extracts the fixed header of MPRTP-AR data packets and sends them to the RTP sub-layer. RTCP packets could also be generated by the RTP sub-layer for generating media transport statistics. RTCP data could be packaged in MPRTP-AR data packets which would be distributed over multiple paths by MPTC sub-layer.

In addition to MPRTP-AR data packets, MPRTP-AR control packets are defined for providing keep-alive packets and MPRTP-AR reports. MPRTP-AR reports (MPRTP-AR Subflow Receiver Report (SRR) and MPRTP-AR Flow Recombination Report (FRR)) contain transport qualities of active paths (e.g., packet loss rate and jitter) and effects on scheduling and flow partitioning. Flow partitioning methods are categorized into two groups that are named coding-aware methods and coding-unaware methods. Coding-aware methods are used for layering coding, multiple description coding or object-oriented coding, and are on RTP sub-layer. In this method, each coding flow is assigned to a subflow, or several coding flows are multiplexed into one subflow. Coding-unaware methods are on MPTC sub-layer, and the RTP/RTCP that are passed from upper layer would evenly spread based on the quality of the associated active paths. Flow reporting is also optionally available for the whole recombined flows.

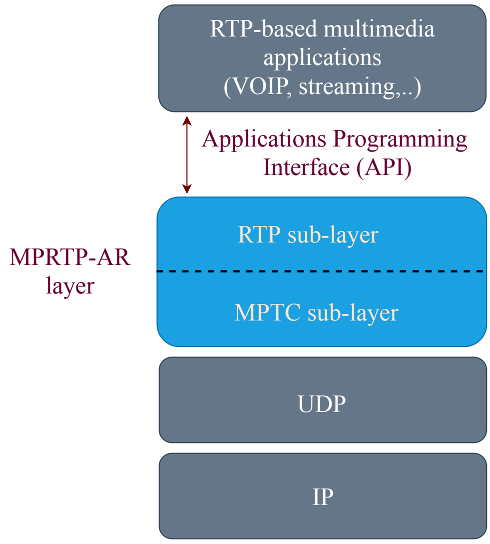


Figure 14 – MPRTP-AR protocol stack (source: adapted from (LEI *et al.*, 2017)).

2.3.1.2 DASH

Dynamic Adaptive Streaming over HTTP (MPEG-DASH) (MPEG, January 2011) was standardized in 2011 by MPEG. DASH supports both VoD and live video delivery. We first detail the DASH system and its main performance limitations. Then, we explain rate adaptation methods. Finally, we discuss the works that are based on this protocol.

DASH system. As explained in Section 2.1, DASH has the same background technology of HTTP adaptive streaming and its system is shown in Figure 15. In DASH system, representations are fragmented into small segments at the server side. DASH component characteristics (text, video, audio, etc.) are described in a XML document named Media Presentation Description (MPD). Typically, DASH client is responsible for choosing the next media segments and requesting the related HTTP URL. Therefore, a rate adaptation method, named adaptation engine in Figure 15, is required to select the proper segments' bit rate by considering the segment availability indicated by the MPD, the network conditions and the media playout situation (e.g., playout buffer level) (SEUFERT *et al.*, 2015).

Performance limitations. The rate adaptation method is responsible for key issues that influence QoE, namely, startup delay, stall, and video quality switches. Startup delay refers to the time since the client request a video until it starts to play, namely pre-buffering. This delay occurs because, generally, one or more segments have to be downloaded completely before the video starts to play. Although this delay helps to prevent stalls under poor network conditions, studies show that it often results in users stopping from watching the video (KRISHNAN; SITARAMAN, 2013). It is important to note that while VoD streaming applications can pre-buffer few seconds of video, live and interactive video streaming can only pre-buffer few hundreds of ms of video (SINGH *et al.*, 2013).

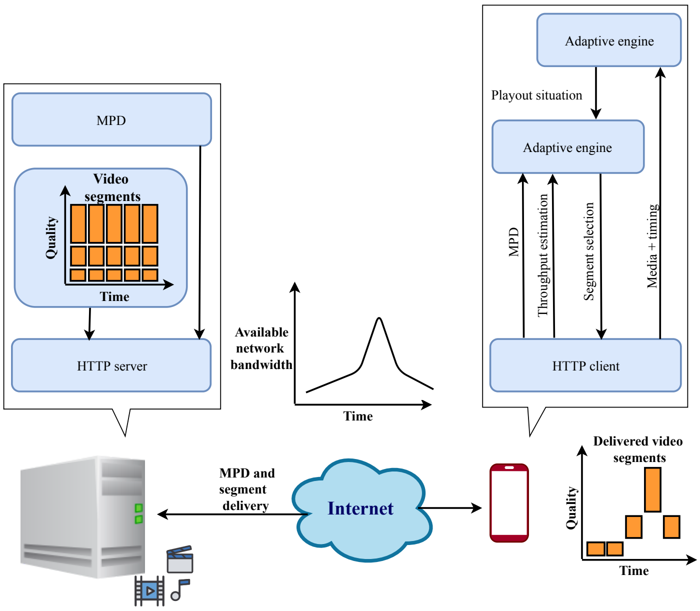


Figure 15 – DASH system (source: adapted from (SEUFERT *et al.*, 2015)).

Stall or interruption refers to the pauses during the video playback due to the playback buffer is emptied, and it needs to wait to buffer video (also called re-buffering) (SINGH *et al.*, 2012). Studies show that stalling happens 40 - 70% of all sessions (JAMES *et al.*, 2016). Generally, this issue occurs because of insufficient bandwidth. In DASH, each segment is available to transmit after completing the process of coding. In addition, there is a dead time between receiving the last packet of one segment and requesting for the next segment. For example, this process time together with transmission time over TCP takes at least 3s (when segments have 1 second) (SWAMINATHAN; WEI, 2011).

One approach to mitigate the stall problem is using a dynamic method to find reasonable segment size (segment duration). Studies in (XIAO *et al.*, 2016) and (WEI; SWAMINATHAN, 2014) show that segment size has a high effect on live latency. While with shorter size segmentation, latency significantly decreases, but the number of HTTP requests increases and consequently, the available bandwidth decreases (XING *et al.*, 2012).

Another approach to decrease latency, and consequently solve the stall issue, is applying subsegmentation transmission. In this approach, each segment divides into several subsegments, and receiver fetches subsegments before the whole segment coding terminates (FEUVRE *et al.*, 2015). This approach is improved by sending subsegments over more than one link simultaneously, which means adding multipath transmission capability, to increase the fetching segment speed. Multipath transmission approach is used in a few works (HOUZÉ *et al.*, 2016; XING *et al.*, 2012; XING; CAI, 2014) to achieve this target. However, the subsegmentation transmission technique also increases HTTP request overhead (XING *et al.*, 2012). In particular, the overhead problem is caused by subsegmentation transmission because at the client side after each request, an average of RTT is required to receive the response from the server. In the case of a large file with small segments/subsegments, this overhead causes a high latency. The HTTP pipelining (KASPAR *et al.*, 2010) is a technique to decrease both number and length of each idle time. In this technique, the client sends the next subsegment request before completing the download of the current subsegment. However, in pipelining, the responses of the requests at the server have to return back in the same order that the requests arrived at the server. Therefore, if it takes a long time to process one request, the other requests would be blocked. For this reason, pipelining is not widely adopted.

To mitigate the overhead problem, the new version of HTTP (HTTP/2) could be used due to its ability to push content in advance, and consequently reduce live latency and network traffic (XIAO *et al.*, 2016).

Switching between different video quality representations is also a problem that impacts the video quality on the user side and causes annoying of viewers. Video switching happens because of the network bandwidth changing or buffer occupancy status. Therefore, it is important to adapt a suitable rate adaptation method, which could identify the network resources and congestion on time in order to have an optimal user experience (OJANPERÄ; VEHKAPERÄ, 2016).

Rate adaptation methods. Typically, rate adaptation methods use throughput monitoring, receiver buffer status, or power level in the process of video segment bit rate decision (SANI *et al.*, 2017).

Throughput-based methods monitor throughput to estimate the available bandwidth. These type of methods try to avoid re-buffering while providing the highest possible video quality. In case of using a throughput-based method, the video quality is unstable (LE *et al.*, 2016) due to throughput variations, which could be caused by TCP behavior (THANG *et al.*, 2014; OJANPERÄ; VEHKAPERÄ, 2016). For example, TCP underestimates bandwidth when segmentation sizes are small (because the corresponding congestion window does not increase) or when bandwidth prediction is weak in networks with fast throughput variation such as cellular networks (SAMBA *et al.*, 2017).

Buffer-based methods choose the video segment bit rate based on the buffer characteristics and usage. The proposed algorithms try to provide a smooth video streaming, but often result in sudden changes in video quality, or freezing when the buffer level (number of unplayed segments in the queue) drops to zero (LE *et al.*, 2016; ZHOU *et al.*, 2014).

Power-based methods select the video segment bit rate using the battery level. Regarding (TRESTIAN *et al.*, 2012), video streaming consumes twice the energy of playing

Table 4 – Multipath support for DASH.

Works	Year	Separate TCP	MPTCP
Xing et al. (XING *et al.*, 2012)	2012	Y	N
Chowrikoppalu et al. (CHOWRIKOPPALU; GOWDA, 2013)	2013	N	Y
RTRA (XING; CAI, 2014)	2014	Y	N
Houzé et al. (HOUZÉ *et al.*, 2016)	2016	Y	N
Corbillon et al. (CORBILLON *et al.*, 2016)	2016	N	Y
Ojanperä et al. (OJANPERÄ; VEHKAPERÄ, 2016)	2016	N	Y
MP-DASH (HAN *et al.*, 2016)	2016	N	Y
Nam et al. (NAM *et al.*, 2016)	2016	N	Y

the same content offline. Therefore, power-based methods try to maximize the battery life time during a video streaming session.

Due to the bit rate selection being more accurate, the work in (SHUAI *et al.*, 2014) shifted the adaptation logic to the server side by deploying a mirrored client buffer on the server. Wilk et al. (WILK *et al.*, 2015) use a proxy server while Mao et al. (MAO *et al.*, 2017a) leverage Pensieve's neural network model on an ABR server to enforce or assist the mobile client adaptation. Such server side approaches have better network utilization compared to the client side approaches (SANI *et al.*, 2017). However, server side approaches are not considered scalable. Rate adaptation methods also perform more efficiently if they can access the network information (OJANPERÄ; KOKKONIEMI-TARKKANEN, 2016). For example, SDN is a technology to implement such a mechanism (KLEINROUWELER *et al.*, 2016; COFANO *et al.*, 2017). Another example is Server and Network-assisted DASH (SAND) (ISO/IEC, 2014; THOMAS *et al.*, 2016) which is a system standardized recently by MPEG to collect and propagate the network information for DASH bit rate adaptation decision. The proposed architecture in (OJANPERÄ; VEHKAPERÄ, 2016) is built upon the Distributed Decision Engine (DDE) (LUOTO *et al.*, 2015) framework to provide more network information (e.g., available capacity, load, QoS) for better rate adaptation decision in multipath scenario.

Multipath support. Current DASH version lacks multipath support, but it is being promoted as its future. Table 4 presents some efforts to integrate DASH with Multipath separate TCP (e.g., (HOUZÉ *et al.*, 2016; XING *et al.*, 2012; XING; CAI, 2014)) and MPTCP (e.g., (CORBILLON *et al.*, 2016), (OJANPERÄ; VEHKAPERÄ, 2016), MP-DASH (HAN *et al.*, 2016), (NAM *et al.*, 2016) and (CHOWRIKOPPALU; GOWDA, 2013)). The table shows attention for combining DASH with MPTCP has increased recently. This is due to MPTCP aggregates bandwidth and supports mobility. MPTCP is also middlebox friendly, and it is supported by the Linux kernel. Besides, MPTCP has got high attention in industry (LI *et al.*, 2016b), (HABIB *et al.*, 2016). More technical details about MPTCP will be provided in Section 2.3.2.4.

James et al. (JAMES *et al.*, 2016) explored “Whether MPTCP would always benefit mobile video streaming?". This research analyzed the performance of different scenarios for DASH over MPTCP. The results show that having two paths with stable bandwidths is beneficial even with small bandwidth capacity on the secondary path. Another positive impact of an additional link is when the primary path has high bandwidth variability. However, there are some harmful cases too. For example, adding an unstable secondary path could harm the stable primary path or when the bandwidth of the secondary path is not enough to transmit higher video bit rates. Therefore, MPTCP is significantly sensitive to bandwidth fluctuation. The results also show that unnecessary multipath causes more energy consumption, resource wasting or increase cost of the quality switch.

One note regarding provide multipath delivery for DASH is about which one of the client or the server is responsible for packet scheduling decisions. In all the research works that spread data over separate TCP connections in Table 4, the client is responsible for choosing the proper path and fetching the suitable segments/subsegments over that path due to the fact that DASH logic is on the client side. But, integration of DASH with MPTCP is challenging when DASH logic resides on the client side, and MPTCP scheduler is on the server side. Besides, MPTCP is transparent to the application layer. Therefore, in the research works of Table 4, which MPTCP is used as transport protocol, rate adaptation logic is kept at the client side. But, scheduling decisions related to packet selection and distributing them through the paths are placed at the server side or both client and server side. The research works (CORBILLON *et al.*, 2016), (OJANPERÄ; VEHKAPERÄ, 2016), MP-DASH (HAN *et al.*, 2016) and (NAM *et al.*, 2016) are more related to the cross layer protocol stack. So, we will discuss them with details about scheduling strategies in Section 2.3.4. The other works are explained in more details below.

Xing et al. used Markov Decision Process (MDP) (BOKANI *et al.*, 2013) to formulate video streaming process as a reinforcement learning task in their works (XING *et al.*, 2012) and (XING; CAI, 2014) for non-scalable and Scalable Video Coding (SVC), respectively. The works' goals are decreasing startup delay, improving video quality and achieving better smoothness. In each of these works, the implemented rate adaptation method selects the next segment based on the current queue length and estimated available bandwidth. To estimate an accurate available bandwidth, Markov channel model is used. This way, adaptation logic finds the transit probability of each link in real-time and determines the best action (e.g., using both links, using only WiFi link, client wait or smoothing). There is also a reward function implemented to reward each action with concern of video QoS requirements (by measuring interruption rate, video quality, video smoothness and search time cost). However, the major problem of using MDP is the high computational cost of solving the complex optimization problem, especially in online and high mobile speed users (BOKANI *et al.*, 2013). In addition, the approach is not a content-aware solution.

Chowrikoppalu et al. (CHOWRIKOPPALU; GOWDA, 2013) modified DASH protocol in order to utilize multipath capability. In this work, the adaptation logic is fed with a proposed bandwidth estimation algorithm and some proposed parameters, including path stability, total path stability and buffer level. The bandwidth estimation algorithm is based on sniffing packets on the interface level. Path stability and total path stability are defined to show the variation of bandwidth on each path and on MPTCP connection, respectively. However, the main problem of this approach is that it does not access the video content information.

Houzé et al. (HOUZÉ *et al.*, 2016) implemented a video player utilizing multipath capability over multiple TCP connections. The goal of this scheme is achieving low-latency in DASH video delivery (below 100 ms). In this approach, server encodes frames of each representation and put them in the related segment every x ms (x depends on the frame rate, for example, x is 40 ms for 25 fps). The client has to fetch each whole frame before the deadline (play time of the frame) and in x ms before a new frame becomes available to fetch. For this target, the authors utilized video delivery over multiple paths as a way to reduce latency. Each frame divides to byte ranges to transfer over different paths and the approach has a mechanism to find the best byte range size in order to receive them with a small inter-arrival time. The larger byte ranges are transferred over faster paths, this way, the variation of transfer delay decreases, and consequently, HOL problem mitigates. Besides, another adapted mechanism is proposed to select the proper representation. In this mechanism, when a segment starts, the biggest frame of each representation is considered in making the decision. The biggest frame is commonly the first frame of each representation (I frame). Therefore, a representation would be selected that the biggest frame has high probability of reaching the destination on time. The problem, however, is that the approach does not consider the video content information. In addition, while the work uses RTT to estimate each path speed, it needs to improve the scheduling strategy to manage the paths.

2.3.1.3 MMT

We previously, in Section 2.2, explained MPEG Media Transport (MMT) (MPEG, 2014) properties, related technologies, MMT functions, and data transmission details. Here, we discuss about multipath support of MMT and a research work that is based on the MMT protocol.

Multipath support.

Regarding multipath delivery, Kolan et al. (KOLAN; BOUAZIZI, 2016) defined a method to establish multipath delivery over MMT, Afzal et al. (AFZAL *et al.*, 2018) proposed a path-and-content-aware scheduling strategy for packet distribution, and Sohn et al. (SOHN *et al.*, 2015) proposed a synchronization scheme for hierarchical video streams over heterogeneous networks. Next we explain each work in more detail.

kolan et al. (KOLAN; BOUAZIZI, 2016) defined a method to establish multipath delivery over MMT. In this method, MMT protocol utilizes signaling protocols such as RTSP or HTTP to establish and control multipath sessions between sender and receiver (transport connection could be either TCP or UDP). For example, in RTSP, the client and the server could be aware of the multipath capability by sending OPTIONS request to each other. This new option tag, called "multipath", could be implemented in the header of the OPTIONS request. The same way, while HTTP is used to set up multipath sessions, the client includes "Multipthid" header to tell the server about its multipath capability. It is also possible to add or drop a network path during the connection. While media is delivering, MMT periodically sends feedbacks to the sender to inform about the path quality information (e.g., loss, delay and jitter). Therefore, the sender could have a view of different paths' situations and dynamically select better performing paths for packets.

Afzal et al. (AFZAL *et al.*, 2018) proposed a novel path-and-content-aware scheduling strategy for MMT to stream real-time video over heterogeneous wireless networks. The authors claim to be the first work attempting to improve the MMT standard by adding multipath scheduling strategies. The path-and-content-aware scheduling strategy, implemented at the server side, applies some methods to improve the perceived video quality based on adaptive video traffic split schemes, Markovian-based techniques, in addition to a discard and a content-aware strategy. Adaptive video traffic split scheme allocates a proper bit rate for each transmission path considering heterogeneous network context with the aim of executing load balancing, relieve congestion, and proper utilizing of each path capacity. The Markovian-based method estimates path conditions and transition probabilities. Discard strategy reduces congestion by avoiding sending packets that would probably be lost. Content-aware strategy protects packets with high priority (I frames and the closest n P frames, named as near-I (NI) frames in this work) by duplicating or assigning them to the best path. The client constantly monitors the path condition, calculates the path metrics which are sent as feedback information packets to the server through the best path. For this purpose, the feedback signaling mechanisms defined in the MMT standard are leveraged. Finally, the proposed path-and-content-aware scheduling strategy lead to QoE improvements around 12 dB for PSNR and 0.15 for SSIM by significant packet loss rate reductions ($\sim 90\%$). It is important to note that the approach does not require any change in the protocol itself since the scheduler can be implemented as part of the client/server applications.

Sohn et al. (SOHN *et al.*, 2015) proposed a synchronization scheme for hierarchical video streams over heterogeneous networks. This scheme is a combination of MMT (for broadcasting) and HTTP (for broadband) video streaming. The work utilizes scalable video streaming. Each layer is segmented in time (in seconds), and duration value can vary according to the user's definition. SHVC-encoded stream is used in the experiment with 3-layers: base layer (HD), first enhanced layer (Full HD (FHD): 2K) and second enhanced layer (UHD: 4K). Base layer and first enhanced layer of video are transferred over the broadcast network (MMT supports multiplexing on packet level), and the second enhanced layer is transferred over broadband network. If the receiver's display has HD-resolution, it will drop the data of the first enhanced layer among data delivered over the broadcasting channel, and it does not need to have a connection with the server for the second enhanced layer, even if it can connect the networks. PI contains essential information, such as the content resolution, location of content, and MMT eXtension Document (MXD), and can also be transferred on broadcast paths. MXD is inserted in PI and mimics the MPD of DASH-SVC. MXD synchronizes the contents over heterogeneous networks, and organizes content synchronization information. The synchronization scheme is implemented at the receiver side. Receiver requests the segments that can deliver on time. For this target, the expected time to download each segment is computed based on bandwidth calculation and segment size information from MXD. This approach is not aware of video content and there is no scheduling strategy to manage the paths.

2.3.1.4 Other Adaptive Streaming Approaches

here we discuss other adaptive streaming approaches that also use HTTP to retrieve data. For example, DAAVI (JOHANSEN *et al.*, 2009) has the same core functionality than DASH by making different bit rate segments on the server, providing MPD for the client, being client logic-based and transferring data over HTTP. However, the MPD structure of DAAVI is different from DASH's MPD. In our research works, the proposed approaches in (EVENSEN *et al.*, 2010; EVENSEN *et al.*, 2011) and (EVENSEN *et al.*, 2012) are all based on DAAVI. These DAAVI-based approaches are for on-demand and live streaming, and the authors claimed that the solutions could also be implemented in a DASH approach.

All adaptive video streaming approaches have the same challenges explained for DASH-based protocols in Section 2.3.1.2. One of these challenges is stalling during video playback. The works, (EVENSEN *et al.*, 2010; EVENSEN *et al.*, 2011; EVENSEN *et al.*, 2012) and GreenBag (BUI *et al.*, 2013) utilized multipath transmission of subsegments to decrease latency, and consequently mitigate the stall issue. As previously explained in Section 2.3.1.2, fetching subsegments over multiple paths can cause the overhead problem. These works used pipelining techniques (KASPAR *et al.*, 2010) to mitigate the overhead issue.

The works (EVENSEN *et al.*, 2011; EVENSEN *et al.*, 2012) and GreenBag (BUI *et al.*, 2013) also proposed dynamic size subsegment methods to determine the size of each subsegment based on the throughput of each interface. As previously explained in Section 2.3.1.2, large sized segments increase the out-of-order packet delivery. Instead, small size segments provide smoother video, but impose higher overhead time (LEDERER *et al.*, 2012). Another problem with using a fixed size subsegment method, instead of a dynamic one, is that a high buffer size is required to compensate for path heterogeneity, which is not desirable. This problem exists in the approach proposed in (EVENSEN *et al.*, 2010).

A feature of GreenBag (BUI *et al.*, 2013) is that it is a middle-ware approach for video streaming over HTTP. Middle-ware approaches are designed to enable multipath interfaces to the current applications without application modifications. Therefore, middle-ware approaches are easy to deploy, but complex to implement (HABAK *et al.*, 2015). This middle-ware approach, GreenBag, locates between a local video player and a remote server. The client requests a video file URL normally over HTTP. GreenBag extracts the URL, determines how to download portions of the video (segments/subsegments), and requests for portions over the decided links. RTT is used to determine when to send the requests for the next segments. Therefore, GreenBag is conventional without requiring any modification in Internet infrastructure or server side.

GreenBag is also an energy-aware bandwidth aggregation approach. Therefore, when single path can provide the required QoS, GreenBag stops using multipath and switches to the single path to improve energy efficiency. Besides, the approach has a medium load balancing and a recovery mechanism. Recovery occurs when a subsegment is lagging and it may pass the deadline. Therefore, the rest of the subsegment will be downloaded through both links. Finally, GreenBag leads to mitigate packet reordering problem and decrease latency.

Noteworthy, none of the adaptive streaming surveyed approaches considers video content features.

2.3.2 Transport Layer Approaches

Video streaming approaches focusing on transport layer protocols have direct access to the network information. Therefore, they can estimate End-to-End characteristics of each path, such as capacity and congestion (RAICIU *et al.*, 2012), that are useful in multipath scenarios. However, the biggest challenge of these solutions is that they generally require modifications in the standardized multipath transport protocols, which may require changes even in the kernel of operating systems.

There are several works exploiting multipath transmission in transport layer, but MPTCP and SCTP are the two main employed transport protocols with multihomed support. In this subsection, we will discuss research works that are implemented based on UDP, DCCP, TCP, MPTCP and SCTP/CMT. Table 3 presents each category.

2.3.2.1 UDP

The User Datagram Protocol (UDP) (POSTEL, 1980), standardized by IETF in 1980, is widely used for unidirectional, broadcast, unicast, multicast, and anycast communications. Next, we provide a brief recap of UDP basics and discuss relevant multipath efforts.

UDP overview. UDP was designed to use a single path for data transmission. It is a connectionless protocol, it does not use sequence numbers for data transmission (HABIB *et al.*, 2016), and there is no guarantee for in-ordered and reliable delivery. UDP also has no congestion control for bandwidth adaptation. These properties make UDP a fast transmission protocol (FAIRHURST *et al.*, 2017) upon which video streaming solutions can be easily implemented. However, the lake of bandwidth adaptation causes UDP to transmit the data with the same bit rate as sent by the application. Therefore, when the network is congested, unless the application holds back, packets get discarded leading to video distortion and reduced QoE (HOSSFELD *et al.*, 2014). Moreover, without congestion control, UDP may occupy a high fraction of the available bandwidth, and consequently, acting unfair to other congestion-avoiding network flows (HUANG *et al.*, 2012).

Multipath support. There are several efforts to add multipath transmission and bandwidth aggregation to UDP for video streaming (WU *et al.*, 2016a; FRERIS *et al.*, 2013; CORREIA *et al.*, 2012). Note that the approaches proposed in BEMA (WU *et al.*, 2016a) and (FRERIS *et al.*, 2013) introduced rate balancing methods to avoid network congestion.

Wu et al. (WU *et al.*, 2016a) designed a Bandwidth-Efficient Multipath streAming (BEMA) protocol and claimed that it was the first work that employed Raptor coding and priority-aware scheduling to stream HD real-time video over heterogeneous wireless networks. This content-aware model sends packets with higher priority on the better-qualified paths and I frame packets through all available paths. The approach also utilizes Forward Error Correction (FEC) to protect transmission data. BEMA also provides a TCP-Friendly Rate Control (TFRC) in order to guarantee fairness toward TCP flows. TFRC (FLOYD *et al.*, 2000) is an equation-based congestion control algorithm, which is designed for unicast multimedia traffic. TFRC estimates the loss event rate at receiver and informs it to the sender, which adapts its transmission rate based on the congestion estimation and on the equation that models TCP congestion control behavior. TFRC responds to the congestion with less fluctuation than standard TCP congestion control and over longer periods of time (CEN *et al.*, 2003). However, TFRC may cause unnecessary reduction of transmission rate during wireless losses. BEMA then adds a ZigZag scheme (CEN *et al.*, 2003) in order to distinguish congestion losses from wireless losses. Only if ZigZag classifies a packet loss as a congestion loss, TFRC will consider it as a lost packet (CEN *et al.*, 2003). Considering the relevance of the feedback information for the proper scheduling process and its high effect on the performance, it is sent periodically from the client to the server over a reliable TCP connection.

Freris et al. (FRERIS *et al.*, 2013) proposed a distortion-aware scalable video streaming to multiple multihomed clients. The authors claimed that their work is the first that simultaneously considered End-to-End rate control and scalable stream adaptation for multipath over heterogeneous access networks. In this approach, the requested video stream is divided into substreams on the server side. The authors developed an algorithm to determine the rate of each substream and the packets to be included in each substream considering network information (e.g., available bandwidth and RTT) and video content features in order to minimize video distortion. Besides that, different cost functions are proposed to provide service differentiation and fairness among users.

The authors also developed heuristic algorithms for deterministic packet scheduling. Once it is a scalable streaming approach, each packet is transmitted only if all other related packets in lower layers have been sent before. Substreams integrate into a single scalable video stream at the client. The authors also studied the trade-off between performance and computational complexity and concluded that it works better for a small number of clients because of overhead.

Correia et al. (CORREIA *et al.*, 2012) proposed a video streaming approach for networks with path diversity using MDC as an error resilience technique. The authors proposed a priority classification. A limited number of packets were classified as high priority because they minimize the distortion of the decoded video affected by packet loss. These packets are delivered without losses. Remaining low priority packets are prone to transmission losses.

2.3.2.2 TCP

Transmission Control Protocol (TCP) (POSTEL, 1981) is a transport protocol standardized by IETF in 1981. This protocol has been widely adopted for video streaming in Real-Time Communications (RTC) (WEBRTC, 2017) and in HTTP-based applications. We previously discussed TCP lack of throughput stability (WU *et al.*, 2016a) with its negative effect on adaptive bit streaming in Section 2.3.1.2. Here, we provide more details about TCP and discuss one research work that is based on this protocol.

TCP overview. TCP is designed to use a single path for data transmission. Regarding data transmission process, TCP uses sequence numbers to detect losses, guarantee in-order packet delivery, and reconstruct the received data (HABIB *et al.*, 2016). The receiver sends ACKs for the correctly received packets. These ACKs are used to provide reliable communication. Retransmission occurs in two cases. First, when there is no ACK from the receiver, which is detected by using a retransmission timer referred to as Retransmission Time-Out (RTO). Second, when the sender receives three duplicate ACKs, which means loss occurred. As previously also discussed in Section 1.2, retransmission wastes bandwidth and adds significant delays. Several protocol improvements have been proposed. For example, Selective Acknowledgements (SACK) (FLOYD *et al.*, 2000), where the receiver informs the sender all successfully arrived packets, so the sender retransmits only the segments that have actually been lost, and Cumulative ACK, which acknowledge the last successfully received packet to the sender. In addition, Explicit Congestion Notification (ECN) (BLACK, 2018) has been proposed as an optional capability to collect congestion information hop by hop and inform the sender about the congestion levels.

TCP Utilizes congestion control by monitoring packet losses and/or delay variations (HABIB *et al.*, 2016) in order to adapt the data rate to network congestion and leads to minimize packet loss (HOSSFELD *et al.*, 2014). In case of not enough network bandwidth available, TCP sends video data with a lower bit rate than the required video bit rate. Thus, video transmission takes longer than the video playback, and consequently may cause the playback to stall. While stall has a severe effect on the perceived video quality, in case of VoD delivery, typically, stall is preferred over video distortions (HOSSFELD *et al.*, 2014). Previously in Section 1.2, we explained about HOL issues and liveness strategies used in TCP-based applications for live or interactive video streaming to cope with stall and delay constraints requirements. Besides all the explained properties, TCP has also the advantage of traversing through firewalls and NATs, a common issue in UDP, altogether turning TCP into a dominant transport protocol for video services (SANI *et al.*, 2017).

Multipath support. Sharma et al. (SHARMA *et al.*, 2008) proposed MultiPath LOss-Tolerant (MPLOT) protocol based on SACK-based TCP and cumulative ACK. A framework, named Hybrid-ARQ (HARQ)/FEC, is defined for MPLOT. Based on HARQ/FEC, MPLOT is using adaptive FEC proactively and reactively instead of high retransmissions to recover losses. Proactive FEC (PFEC) packets are used to recover losses and when PFEC packets in a block are not enough to recover lost data, then Reactive FEC (RFEC) packets need to transmit. This method leads to goodput improvement and decreased recovery latency in high lossy channels (CHOW *et al.*, 2009). Regarding packet scheduling, paths in MPLOT are categorized into good and bad paths. The channels with ranks higher than a threshold (median rank) are categorized as good paths. Ranks are calculated based on network parameters, such as congestion window, PLR and RTT. MPLOT provides an uncoupled congestion control which means each path has its own congestion control. ECN is used to find congestion losses (from faulty/lost channels) and to change the congestion window size. However, MPLOT is deployed for wireless mesh networks and it is not easily expendable on the Internet due to scalability and compatibility issues. The authors assume that a buffer is enough to compensate out of order delivered packets, which are important in video quality (CHOW *et al.*, 2009; LI *et al.*, 2014). Moreover, the approach is using a CBR coding scheme, which decreases the performance when the path quality decreases sharply (CHOW *et al.*, 2009).

2.3.2.3 DCCP

Datagram Congestion Control Protocol (DCCP) (KOHLER *et al.*, 2006) is a transport protocol standardized by IETF in 2006. Here, firstly, we provide an overview of DCCP, such as data transmission process, and its properties. Then, we discuss one research work that is based on this protocol.

DCCP overview. DCCP is designed to use a single path for data transmission providing bidirectional and unicast data delivery. Regarding data transmission process, DCCP uses sequence numbers. Therefore, the client can detect losses and inform them to the sender by ACKs. There is no retransmission method and in-order data delivery. In addition, there is an ability for feature negotiation before or during transmission, such as ECN capability, ACK ratio, and congestion control mechanism.

DCCP has different congestion control mechanisms that are represented by Congestion Control IDentifier (CCID), for example, CCID2 and CCID3. CCID2 has a TCP-like Congestion Control. Thus, the sender has a congestion window and sends data until making the window full. Both dropped packets and ECN trigger the congestion algorithm and halve the congestion window. Acknowledgments contain a list of received packets within some window, like Selective Acknowledgements (SACK)-based TCP. Therefore, CCID2 (FLOYD; KOHLER, 2006) provides quick access to available bandwidth and deals with quick bit rate changing (KOHLER *et al.*, 2006; HUANG *et al.*, 2012). CCID3 (FLOYD *et al.*, 2006) provides TFRC. CCID3 responds to congestion smoothly and maintains steady bit rate (KOHLER *et al.*, 2006; HUANG *et al.*, 2012).

A comparison among UDP, TCP and DCCP variants (CCID2 and CCID3) for transferring MPEG4 video, shows that DCCP provides higher throughput and less packet loss compared to UDP while UDP supplies much less delay and jitter. Finally, DCCP comes up with the best QoS compared with TCP and UDP transport protocols over congested network (AZAD *et al.*, 2009). However, since subjective results in the work (HOSSFELD *et al.*, 2014) shows stalling caused by TCP is preferred over distortion caused by UDP for VoD streaming, DCCP without retransmission may also suffer from video distortion and may not outperform TCP and UDP for VoD in terms of QoE.

Multipath support. In our research works, Huang et al. (HUANG *et al.*, 2012) proposed a Multipath Datagram Congestion Control Protocol (MP-DCCP) for supplying a multipath transmission to DCCP. In MP-DCCP, each link has its own DCCP connection, which means that each link can maintain its own congestion control window, sending rate adjustment and CCID. The proposed schedule scheme in MP-DCCP is called QoS-aware Order Prediction Scheduling (QOPS). QOPS assigns important frames, such as I frames into paths with less Packet Loss Rate (PLR). Besides, QOPS predicts the order of packets at the receiver side by estimating the path latency to deal with the out-of-order problem. Based on the final results, among the congestion control algorithms defined in DCCP standard, conjunction of CCID3 to MP-DCCP is recommended due to its steady transmission.

2.3.2.4 MPTCP

Multipath TCP (MPTCP) (FORD *et al.*, 2009; FORD *et al.*, 2011) is a prominent protocol for multipath transmission developed at IETF since 2009. MPTCP has been implemented in the Linux kernel (ICTEAM, 2017), and also as an experimental kernel patch for FreeBSD-10.x (CAIA, 2019). Industry has also adopted MPTCP on smartphones (BONAVENTURE; SEO, 2016; ICTEAM, 2019). Two major deployments are voice recognition (SIRI) application (APPLE, 2017) since 2013, and for any application on iOS11 (CHESHIRE, 2019). Another major MPTCP deployment in high-end Android smartphones (e.g., Samsung Galaxy S6 and Galaxy S6 Edge smartphones) relies on network-operated SOCKS proxies, reaching bandwidth of 1 Gbps by KT Corporation, in Korea 2015 (SEO, 2015). In the following, we first provide an overview of MPTCP. Then, we discuss performance problems. Finally, we survey relevant works based on this protocol.

MPTCP overview. MPTCP was designed to use multiple paths for data transmission. In particular, MPTCP establishes multiple subflows for a single MPTCP session.

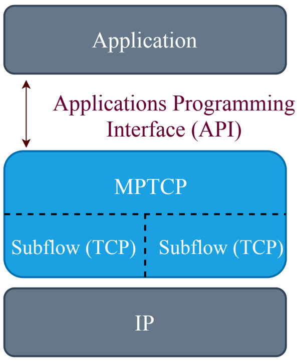


Figure 16 – MPTCP protocol stack (source: adapted from (FORD *et al.*, 2009)).

A subflow is a TCP flow over an individual path and looks similar to a regular TCP connection. Besides, there is a MP_CAPABLE option to identify that the connection is MPTCP rather than TCP. Further, a token is associated to the MPTCP session. This token is used for subflows to add to this particular session. In MPTCP, application layer sees MPTCP connections as unique, as shown in Figure 16. Therefore, sender's transport layer packetizes data to TCP packets and receiver's transport layer reorders and recreates the byte stream without application layer knowing about it. As a result, application layer stays unmodified and a standard socket API is used.

Regarding data transmission process, each packet contains two sequence numbers: the Subflow Sequence Number (SSN) to loss detection and an additional Data Sequence Number (DSN) to reconstruct the original data at the receiver. MPTCP also utilizes ACKs for subflow and connection level. SACK/Cumulative ACKs are used at subflow level and DSN-ACKs are used at connection level (HABIB *et al.*, 2016). For data transmission protection, MPTCP uses retransmission mechanism as in regular TCP. Besides, in the case of packet loss over a subflow, retransmission could be over another subflow.

Default MPTCP uses coupled congestion control (each MPTCP connection has its own congestion control) to avoid an unfair TCP connection. This algorithm provides better congestion balancing than just using TCP congestion control over each subflow (uncoupled) (WISCHIK *et al.*, 2011; RAICIU *et al.*, 2011) because MPTCP over regular TCP connections could behave unfairly.

A shared MPTCP receiving buffer is used at the receiver side to receive and reorder packets of different paths (CORBILLON *et al.*, 2016). In other words, there is a single window shared by all subflows at the receiver side.

Because in multipath approaches, packet scheduling strategy has an important role, there are different strategies introduced for MPTCP. Performance comparison of scheduling methods for multipath transfer is analyzed in (SINGH *et al.*, 2012) and different schedulers are implemented and evaluated in (PAASCH *et al.*, 2014) for MPTCP. Default MPTCP packet scheduling strategy selects the packets in First-In First-Out (FIFO) order and maps them to the different paths according to RTT-based policy.

MPTCP supports middleboxes and is compatible with the current network infrastructure (HABIB *et al.*, 2016). This is due to this fact that SSN contains a consecutive sequence number for each subflow packet. Therefore, it can pass through middleboxes (BARRÉ *et al.*, 2011). However, in case of conflict, MPTCP handles middleboxes by fallback to the regular TCP (HESMANS *et al.*, 2013). Moreover, MPTCP provides resilience, mobility and load balancing (FAIRHURST *et al.*, 2017).

Performance challenges. Studies in (DENG *et al.*, 2014) and (SINGH *et al.*, 2012) show that MPTCP presents performance issues most critically in the case of heterogeneous paths. The reasons of MPTCP performance limitations are discussed below:

- **Out-of-order packets:** MPTCP suffers from out-of-order packet problem. A comparison between Single Path TCP (SPTCP) and MPTCP in (NAM *et al.*, 2016) shows that SPTCP outperforms MPTCP when paths are heavily imbalanced in terms of throughput. MPTCP operates poorly in this case due to a large number of out-of-order delivered packets. Such imbalance throughput could also happen frequently in the case of using 5G network simultaneously with other wireless networks. In our research works, the approach proposed in (NAM *et al.*, 2016) introduced a dynamic MPTCP path control to remedy out-of-order problem.

- **HOL blocking due to ARQ mechanism:** Using ARQ mechanism by MPTCP causes frequently HOL blocking problem, even more than a single TCP connection (CORBILLON *et al.*, 2016). As previously explained in Section 1.2, HOL incurs large End-to-End delay and low performance. In our research works, the proposed approaches in ADMIT (WU *et al.*, 2016b), (DIOP *et al.*, 2012), (XU *et al.*, 2016) and (CAO *et al.*, 2016) attempted to solve the retransmission problem in order to decrease End-to-End delay.

- **Frequent throughput fluctuation and unnecessary fast retransmission:** MPTCP uses Additive-Increase/Multiplicative-Decrease (AIMD) congestion control algorithm to set congestion window sizes. The problem is that AIMD causes frequent throughput fluctuation and significant End-to-End delay (WU *et al.*, 2016b; LIM *et al.*, 2014). For example, out-of-order packet delivery, which is common in multipath transmission, and losses, which could be wireless loss and not congestion loss, could trigger unnecessary fast retransmission, which impacts undesirable reduction in the size of congestion window and waste useful bandwidth (HABIB *et al.*, 2016). In our research works, ADMIT (WU *et al.*, 2016b) considered the packet loss differentiation to mitigate this problem.

- **Content-agnostic traffic scheduling:** In MPTCP, availability of multipath connections is unknown to the application. Therefore, MPTCP is unaware of application information and video content features. The approaches proposed in (CORBILLON *et al.*, 2016) and (HAN *et al.*, 2016) introduced cross layer solutions to access the video content and deadlines, respectively.

- **Fully reliable and ordered service:** MPTCP is an extension of TCP protocol with inherited fully reliable and ordered services, which are not required by video streaming. In our research works, there are some efforts (DIOP *et al.*, 2012; XU *et al.*, 2016), PR-MPTCP^{+}(CAO *et al.*, 2016) applying the concept of partial reliability in MPTCP for real-time video delivery. This concept avoids retransmission for acceptable loss rates and provides partial reliable video data transmission to the upper layers (DIOP *et al.*, 2012; XU *et al.*, 2016; CAO *et al.*, 2016).

  Partial reliability leads to improved network performance parameters (e.g., delay, bandwidth), and consequently, better QoE (DIOP *et al.*, 2012).

Improved scheduling mechanisms. There are several proposals to improve MPTCP regarding the above mentioned problems through scheduling functions that define the multipath decision. Next, we briefly review them and provide more details. Cross layer works to adapt application/network layer protocols with MPTCP (e.g., (HAN *et al.*, 2016), (CORBILLON *et al.*, 2016) and (NAM *et al.*, 2016)) will be presented later in Section 2.3.4.

Wu J. et al. proposed quAlity-Driven MultIpath TCP (ADMIT) protocol (WU *et al.*, 2016b) for streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks. ADMIT is an extension of MPTCP with inheriting basic mechanisms from it, including coupled congestion control, the same connection, subflow level acknowledgments, and retransmission mechanism. The authors claimed that ADMIT is the first MPTCP scheme that incorporates the quality-driven FEC coding and rate allocation to mitigate End-to-End video streaming distortion. The proposed FEC Coding in ADMIT, adaptively chooses FEC redundancy and FEC packet sizes according to the network situations (e.g., RTT, bandwidth and, packet loss rate) and delay constraint. This adaptive FEC coding leads to remedy the shortcomings of packet retransmission (e.g., serious delay and performance degradation (WU *et al.*, 2016a)) by protecting video data. Besides that, the proposed rate allocator algorithm is responsible for load balancing. ZigZag scheme (CEN *et al.*, 2003) is also used in ADMIT. ZigZag has high effect on the FEC coding and rate allocator results due to distinguishing congestion losses from wireless losses. Finally, packet scheduling strategy maps FEC packets to the different paths according to the rate allocation vector. However, there is no mechanism to ACK for reconstructed lost packets in FEC unit. Therefore, the ADMIT protocol keeps sending retransmissions of the lost packets until receiving the ACK. Besides, the packet scheduling strategy is not aware of the frames different priorities. Another problem is that all packets of the Group of Pictures (GoP) and redundant packets must be received before the GoP frames are processed. Each video unit may consist of several packets and it may also depend on other units.

The works (DIOP *et al.*, 2012; XU *et al.*, 2016), and PR-MPTCP$^+$(CAO *et al.*, 2016) apply the concept of partial reliability in MPTCP. These works demonstrate that capability of partial reliability for MPTCP outperforms the default MPTCP for real-time video streaming. As a comparison among these works, one can note that the approach in PR-MPTCP$^+$ (CAO *et al.*, 2016) defines that switching between MPTCP and partial reliable capability occurs dynamically based on the network situation. However, in (XU *et al.*, 2016), partial reliability is only activated in the initial handshake, and there is no explanation about how switching occurs in (DIOP *et al.*, 2012). Besides, the works in (DIOP *et al.*, 2012) and PR-MPTCP$^+$ (CAO *et al.*, 2016) used old versions of MPTCP. Finally, these works defined different methods for applying partial reliability, which are explained in more details below.

Diop et al. (DIOP *et al.*, 2012) introduced QoS-ORIENTED MPTCP in order to improve QoS in terms of End-to-End delay. In this work, two QoS-aware mechanisms are implemented with the concept of "partial reliability" in MPTCP for interactive video applications. The first one, MPTCP-SD (selective discarding), eliminates the least important packets (B frames) at the sender side. This could decrease the network traffic and avoid latency and loss of I and P frames. The capability of gathering priority information for MPTCP is implemented by using Implicit Packet Meta Header (IPMH) interface (EXPOSITO *et al.*, 2009).

In the second mechanism, a time-aware policy is used. In MPTCP-PR (time constrained partial reliability), delay of each queued packet on the receiver side is calculated and whenever it gets close to a time limit (400 ms), packets are sent to the application, and acknowledge would be sent for the missed packets. In addition, delivered packets after a specific time limit are considered as losses, but acknowledgments are sent for them to the sender. The results show that MPTCP-SD provides better video QoS than MPTCP-PR and MPTCP.

Another MPTCP Partial Reliability extension is introduced in (XU *et al.*, 2016) to provide different required reliability level and recommended for video streaming. There is a threshold for the maximum number of retransmission attempts, or maximum delay of transmission for each packet. In this approach, the sender and receiver negotiate about partial reliability function in the initializing phase. During data transmission whenever a packet exceeds the defined threshold, the sender informs it to the receiver. Therefore, the receiver will not wait anymore to receive that packet. Consequently, the receiver will send a forced acknowledgment and sender eliminates that packet from its buffer similar to the time the packet delivered successfully. The forced acknowledgment also shows losses and congestion in the network and triggers the congestion control algorithm.

Cao et al. (CAO *et al.*, 2016) proposed Context-aware QoE-oriented MPTCP Partial Reliability extension (PR-MPTCP$^+$). In this work, sender monitors network congestion and receiver buffer blocking to determine when it should enable partial reliability. In order to detect network congestion, a function of RTT for each path is proposed and to detect the buffer blocking, advertised receiver window (rwnd) is used. In the case of a congested network, only the packets with enough deadline to play would be sent and the packets with the highest priority could be retransmitted. In particular, in this work, the concept of context is used to refer to the video content where I frames have the highest priority. Whenever buffer blocking is detected, a subset of paths are adaptively selected based on their quality (e.g., bandwidth). The approach switches to the full MPTCP mode (standard MPTCP) when there is no buffer blocking. Authors of PR-MPTCP$^+$ demonstrate that this method outperforms the proposed approach in (DIOP *et al.*, 2012) in terms of video performance metric.

2.3.2.5 SCTP and CMT (Extension of SCTP)

The first SCTP specification was published in the now obsolete RFC 2960 (STEWART *et al.*, 2000) in 2000 and then it was updated in RFC 3309 (STONE *et al.*, 2002) and RFC 4460 (STEWART *et al.*, 2006). The current protocol specification is in RFC 4960 (STEWART, 2007) containing updates and standardized by IETF in 2007. SCTP provides multihoming, multistreaming, and there is support for SCTP by different operating systems and platforms (e.g., Linux and Android). Here, firstly, we have an overview of SCTP, such as data transmission process, and SCTP properties. Then, we indicate performance limitations. Finally, we discuss the research works that are based on this protocol.

SCTP overview. SCTP is a message-oriented protocol like UDP and supports reliability by using congestion control and retransmission like TCP (STEWART, 2007). Default SCTP uses one path as a primary path for transferring data packets, and other paths are used for redundancy transferring (retransmission and backup packets). Redundant paths are used to have more resilience and reliable data transferring than using only a single path. In particular, SCTP sets up an association with different IP addresses for each end host (FU; ATIQUZZAMAN, 2004). Association, in SCTP, refers to the connection between SCTP end hosts.

SCTP provides multistreaming capabilities that reduce the HOL blocking problem. In SCTP, each stream is a subflow within the overall data flow, where multistreaming refers to the simultaneous transmission of several independent streams of data in an SCTP association. SCTP multistreaming works by adding stream sequence numbers to the chunks of each stream. Sequence numbering guarantees the in-order packet delivery inside a stream while unordered delivery can happen across streams. Therefore, arrived data of a stream can be delivered to the application layer even if other streams are blocked because of losses. Default SCTP also uses another sequence space called Transmission Sequence Number (TSN) for each chunk – the unit of information within an SCTP packet (STEWART, 2007). TSN is global for all streams with the goal of loss detection and reconstructing the original data at the receiver. Besides, SACK/Cumulative TSN ACK are lveraged as acknowledgment methods. Cumulative TSN ACK is a field of SACK to acknowledge the TSN of the last successfully received DATA chunk to the sender. For data transmission protection, SCTP uses a retransmission mechanism upon two types or events. First, whenever RTO expires. Second, after four SACK chunks have reported gaps with the same data chunk missing. Besides, SCTP uses uncoupled congestion control, and a shared buffer is used for all paths on the receiver side.

SCTP performance limitations. SCTP presents performance limitations in heterogeneous paths and it is challenging to adopt it for video streaming:

- **Applications modification requirement:** SCTP requires distinct socket API and applications modifications (BARRÉ *et al.*, 2011).
- **Lack of support in middleboxes:** SCTP suffers from lack of support in middleboxes (BARRÉ *et al.*, 2011).
- **Frequent primary path exchange:** SCTP is slow due to frequent primary path exchanges in case of failure. In SCTP, the process of path primary exchange takes a long time (SILVA *et al.*, 2016) by, for example, detecting 6 lost packets. In SCTP, a packet is recognized as lost if the sender does not receive ACK at a specific time of RTO. RTO is set to 1 second at the start and after each loss detection, it doubles. Finally, the minimum time to change the path is 63 seconds. Therefore, the process of path primary exchange takes a long time and causes a high delay. This issue is considered in the works, (KELLY *et al.*, 2004; OKAMOTO *et al.*, 2014), and SRMT (SILVA *et al.*, 2016).
- **Lack of load balancing support:** Default SCTP is not load balancing over multiple paths. Load balancing is an important factor in multipath transmission. Several efforts have been done to add capability of bandwidth aggregation to SCTP, and also adapting this protocol for video streaming. This issue is considered in the research

works, CMT-DA (WU *et al.*, 2015), CMT-CA (WU *et al.*, 2016d) and CMT-QA (XU *et al.*, 2013).

- **Unnecessary fast retransmission:** Out-of-order packet delivery and wireless losses could trigger unnecessary fast retransmissions, decrease goodput sharply, and consequently mitigate transmission efficiency (XU *et al.*, 2013). This issue is considered in the research works, CMT-DA (WU *et al.*, 2015), CMT-CA (WU *et al.*, 2016d) and CMT-QA (XU *et al.*, 2013).

- **Content-agnostic traffic scheduling:** While considering video content features in scheduling strategy could improve the QoE and network utilization, default SCTP scheduling treats in a content-agnostic fashion. This issue is considered in the research work, CMT-CA (WU *et al.*, 2016d).

- **Fully reliable and ordered service:** SCTP is a fully reliable and in order protocol, which is not required by video streaming. In our research works, PR-SCTP (SANSON *et al.*, 2010) applied the concept of partial reliability in SCTP for real-time video delivery.

Improved scheduling mechanisms. There are several approaches to improve SCTP to solve the above mentioned problems and provide video streaming over this transport protocol. We briefly mentioned them and next, we provide more details.

To reduce the explained problem of longtime primary path exchanging in SCTP, Kelly et al. (KELLY *et al.*, 2004) proposed a delay-centric strategy to set the primary path based on the lowest End-to-End delay and RTT. The solution improves quality, but using this adaptive primary path selection in the lossy wireless environment makes the SCTP slow due to frequent path exchanges. This approach does not use the full ability of all paths and uses the primary path for data transmission and secondary paths as backup.

A more stable solution based on SCTP is in (OKAMOTO *et al.*, 2014). The authors defeated with packet loss by proposing a selective bicasting method. Therefore, instead of sending the same data through two different paths (bicasting), which would lead to significant congestion and reduce the throughput, the selective bicasting method duplicates only important packets. These important packets are retransmissions. However, this approach has not defined sensitive data, like I frames, as important packets.

Da Silva et al. (SILVA *et al.*, 2016) proposed a Selective-Redundancy Multipath Transfer (SRMT) scheme. In this approach, the primary path is used to transfer data and secondary paths are used to send redundant packets, which have more priority and stronger delay limitation. These redundancies mitigate degradation QoE. There are two key factors for packet selection over secondary paths. The first one is the amount of redundant packets to be transferred, which is calculated based on smooth Round Trip

Time (sRTT) of the primary path and the maximum delay tolerated by the application. The second one is the selection of packets, which have to be sent redundantly based on the importance of packets for reconstructing the video (a content-aware approach). For example, I frames have the highest priority and among the I frame packets, the initially ordered ones have more priority than others. P frames are the next and the lowest priority is for B frames. Duplicated packets on the receiver side would be discarded. SRMT uses the default SCTP handover scheme to avoid HOL problem.

In order to make reliable SCTP protocol flexible for video streaming, the Partially Reliable SCTP (PR-SCTP) extension was firstly defined in (STEWART *et al.*, 2004), and later additional policies were specified in (TUEXEN *et al.*, 2015). Similar to the explained concept of partial reliability for MPTCP in Section 2.3.2.4, PR-SCTP introduced some policies for choosing reliability level. PR-SCTP supports choosing the retransmission policy by using either a maximum number or a time for retransmissions, and after that, the packet will not be retransmitted anymore. PR-SCTP shows benefits for time-sensitive applications involving video and audio streaming (WANG *et al.*, 2003). In our research works, the proposed approach in (SANSON *et al.*, 2010) utilized the partial reliability services of PR-SCTP for real-time H.264/AVC video streaming. H.264/AVC has a Network Adaptation Layer (NAL) feature, which is a layer of abstraction over the actual encoded data. NAL header contains decoding parameters and its level of importance for decoding. This information is used by PR-SCTP to decide the number of retransmissions for each I, P and B frames. A probabilistic model is developed to find optimum values for the maximum number of retransmissions for different types of frames in order to provide a trade-off between reliability and delay. Retransmissions are over the secondary paths. The result shows that the proposed solution outperforms UDP and TCP.

Another extension solution of SCTP is Concurrent Multipath Transfer (CMT) (IYENGAR *et al.*, 2006). Most CMT solutions use all the available paths simultaneously for data transferring to increase the throughput and network resiliency. There are many schemes developed based on CMT, such as CMT-DA (WU *et al.*, 2015), CMT-CA (WU *et al.*, 2016d) and CMT-QA (XU *et al.*, 2013). Among these works, CMT does not use any path selection method and uses Round Robin for data distribution. Using Round Robin for CMT not only increases out-of-order delivery, and HOL blocking at receiver, but also increases SACK overhead and additional unnecessary retransmission. CMT evolved to perform better estimation of the network situation and choosing qualified paths for data transmission in CMT-QA (XU *et al.*, 2013), CMT-DA (WU *et al.*, 2015) and CMT-CA (WU *et al.*, 2016d). CMT-CA (WU *et al.*, 2016d) is also fed with video content properties besides the network situation. These works are also different in designing of congestion control and retransmission mechanism. More details will be presented in Section 2.5.1.

Xu et al. (XU *et al.*, 2013) proposed a path and quality-aware adaptive concurrent multipath transfer (CMT-QA) approach for packet scheduling over network channels. The goal of this scheme is decreasing out-of-order problem by reducing the unnecessary fast retransmissions and reordering delay. To achieve this target, a path quality estimation model (PQEM), an Optimal Retransmission Policy (ORP) and Data Distribution Scheduler (DDS) are introduced. PQEM calculates each path quality by estimating the rate of the distributed data, which is a function of sending buffer size and transmission delay. In PQEM, the shared sender buffer is divided into subbuffers. Each path has its own subbuffer and management independently and the allocation of buffer space size is dynamical. ORP handles packet loss differentiation and retransmits the lost packets over faster paths. DDS predicts the arrival time of data distributed over each path, and determines the amount of data to be transferred based on the congestion control parameters including cwnd, rwnd and sender buffer size. Therefore, DDS distributes data per path in the way that they arrive to the receiver in order. SACK is used for acknowledgment method. However, the approach does not concern TCP fairness toward other traffic flows (XU *et al.*, 2015) and it is not appropriate for video due to the lack of use of video content parameters.

Wu et al. (WU *et al.*, 2015) proposed a distortion-aware concurrent multipath transfer (CMT-DA) scheme and claimed that this approach was the first work to introduce the video distortion into SCTP for enhancing HD video quality in heterogeneous wireless environments. The goal of this approach is decreasing video distortion by mitigating the effective loss rate for variable bit rate video streaming. To achieve this goal, three main methods are proposed: path status estimation and congestion control, flow rate allocation, and data retransmission control. CMT-DA estimates path situations (e.g., RTT and available bandwidth) by processing ACK feedbacks, and applies a distortion-aware model at the flow level to schedule the packets. Aggregated feedback packets are sent after each packet delivery. The used SACK/Cumulative ACK feedback packets return to the sender through the most reliable paths to avoid losing or dropping during the network transmission. In addition, the congestion control is designed per path and defined parameters are RTT, cwnd and RTO. ECN detects path congestion and changes the congestion window size. The rate controller is proposed to choose a subset of paths dynamically and assign data transmission rates. The data retransmission control is defined to retransmit the packets which are estimated to arrive at the destination within the deadline. However, only flow level distortion consideration without analyzing frame priority and decoding dependency of frames is not adequate for video streaming.

In another research work, Wu et al. (WU *et al.*, 2016d) proposed a content-aware CMT (CMT-CA) scheme and claimed this approach was the first SCTP to incorporate the video content analysis into the scheduling for enhancing HD video quality in heterogeneous wireless environments. The goal of CMT-CA is to accurately estimate the video content parameters and appropriately schedule the video frames to achieve the optimal quality. To achieve this goal, three main methods are proposed: quality evaluation based decision making, congestion control, and data distribution. Quality evaluation based decision making estimates network situation and frame level distortion. Further, these pieces of information are used for packet scheduling. Similar to what explained for CMT-DA, SACK/Cumulative ACK feedbacks are used for path situation estimation and they are sent after each packet delivery through the most reliable paths. The congestion control for CMT-CA is designed per path, Markov model-based (MDP), and is TCP-Friendliness. Congestion control parameters are RTT, cwnd, RTO and ssthresh. ZigZag scheme (CEN *et al.*, 2003) detects path congestion and MDP changes the congestion window size. Data distribution is responsible for packet scheduling and different transmission is applied for I and P frames. Therefore, high priority frames can be transmitted first, which helps to decrease video distortion. Besides that, the proposed algorithm drops the video frame if its parent frame cannot be delivered due to bandwidth restriction. Therefore, this algorithm conserves network resources. Besides the proposed methods, CMT-CA also utilizes similar data retransmissions methods designed in CMT-DA. For example, Selective Acknowledgements (SACK) (FLOYD *et al.*, 2000), which provides a list of correctly/incorrectly received packets to the sender, and cumulative ACK, which informs the last successfully received packet to the sender.

2.3.3 Network Layer Approaches

Video streaming approaches focusing on the network layer have access to the IP level and to useful information in multipath scenarios, such as network, routing and data forwarding information. In addition, network layer multipath approaches take care of data spread over different interfaces without the application awareness about this process. The biggest challenge of these solutions is that they generally require network changes, new infrastructure or modifications in the kernel of operating systems. Our research works are categorized into two groups based on the required network technologies: SDN/OpenFlow-based and Proxy-based approaches. These research works will be discussed in this subsection. Table 3 presents each category.

2.3.3.1 SDN/OpenFlow

Software-Defined Networking (SDN) is a network architecture based on a logically centralized control plane (KREUTZ *et al.*, 2015) and programmatic abstractions (e.g., OpenFlow) to define the behaviour of the forwarding devices (e.g., routers, switches). SDN controllers gather network information including capacity and packet loss rate of links in real-time and dynamically change routing paths based on the network situations and policy definitions. In this chapter, we leave out of scope the topic of how paths are computed. We only cover relevant works on refactoring and modifying the networking stack on Android and Linux devices to be able to use multiple network interfaces simultaneously in (YAP *et al.*, 2012), and we also discuss SDN feedback approach for path decision actions, as proposed by MARS (SUN *et al.*, 2016).

Yap et al. (YAP *et al.*, 2012) explored how to make use of all the available networks around us. The approach provides a seamless HTTP connectivity on heterogeneous networks. In this approach, to transfer data from one application over multiple interfaces, the application uses one IP source address. Then, the networking stack spreads data over multiple interfaces and assigns an IP address for each one. This was implemented by using a virtual Ethernet interface to connect the application, with its local IP address, to a special gateway inside the Linux kernel. This gateway combines multiple interfaces together without the application knowledge. To implement the solution, the authors re-factored the networking stack connectivity service of the Android kernel and added a controller Open vSwitch (OVS) in the kernel of the mobile devices. OVS has an OpenFlow interface and can utilize flow table entries. Therefore, controller and OVS helped to route and re-route the flows and packet controlling.

The goal of Multiple Access Radio Scheduling (MARS) (SUN *et al.*, 2016) is solving out-of-order problem and reducing the End-to-End delay. MARS is implemented on separate TCP connections. The authors used SDN for flow aggregation and flow splitting, and also designed a scheduling scheme, named MARS, which is based on relative RTT measurement (which will be explained in Section 2.4.3). The relative RTT is calculated each fixed period of time to make sure it is always valid. Accordingly, the low-latency paths are chosen for data transmission. In MARS, the controller calculates bandwidth and RTT of each path, and notifies them to the sender. The sender can also inquiry such information from the controller. This information would be used in scheduler to split video blocks into several paths. These flows combine on edge router close to the client for one-interface receiver, but it can also work for the receiver with two interfaces. However, the approach considers neither packet loss for path quality calculation nor priority of video data units.

2.3.3.2 Proxy Solutions

It is possible to use proxy at one side (client/server) or at both sides. Using proxy at one side hides multipath transmission from the other side. In the case of using proxy on both sides, each endpoint communicates with the proxy via a normal connection without awareness of the multipath communication. In proxy-based applications, a tunneling IP-in-IP mechanism (to encapsulate one IP packet as a payload in a new IP packet) is used to redirect data to different paths over routing level. Consequently, proxy-based approaches are transparent to both transport and application layers and do not require any changes in them (LI *et al.*, 2016b).

Chebrolu et al. designed a network layer architecture, Bandwidth Aggregation (BAG) (CHEBROLU; RAO, 2006), to utilize bandwidth aggregation for real-time applications. In BAG, server streams video data to the client by using a UDP socket. In particular, there is a proxy at the client side, which is aware of client interfaces and splits flow over these network interfaces by using IP-in-IP tunneling (see Figure 17). The proposed scheduling algorithm, Earliest Delivery Path First (EDPF), estimates the delivery time of each packet over each path and spreads packets over the fastest path in order to avoid packets from missing their deadlines and minimizing packet reordering. Delay and wireless bandwidth between the proxy and the client are used for delivery time estimation. As a result, EDPF is more efficient than Round Robin in avoiding HOL (LI *et al.*, 2016b). The advantage of using proxy at the client side is that no change is required at the server side (LI *et al.*, 2016b).

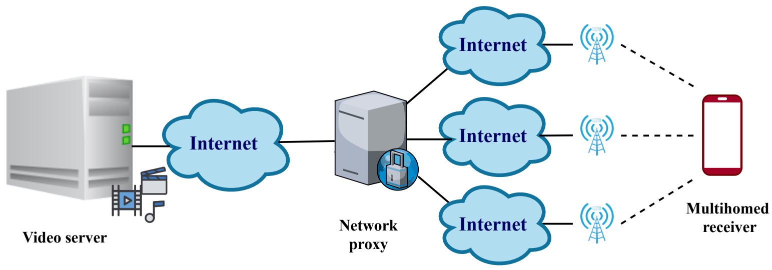


Figure 17 – BAG (CHEBROLU; RAO, 2006) system architecture featuring the use of a proxy and IP-in-IP tunneling between a client and the proxy (source: adapted from (CHEBROLU; RAO, 2006)).

2.3.4 Cross Layer Approaches

Although it is possible to estimate throughput or bandwidth and other network parameters at the application layer, they are not as accurate as the transport or network layer measurements. Different layers have different knowledge levels. For instance, the application layer is aware of video features, player buffer and deadlines. The transport layer is able to calculate the bandwidth and RTT, and it also has a congestion control mechanism. The network layer accesses IP level and routing paths, and the link layer has wireless parameter access.

Therefore, the interaction between different layers has the benefit of utilizing the advantages of different layers by signaling messages among them. This interaction is known as cross layer and was epitomized in the Transport Services (TAPS) working group by IETF (FAIRHURST *et al.*, 2017). Mostly, lower layers gather network information and feed them to higher layers (LI *et al.*, 2016b).

In cross layer approaches, usually application layer or transport layer becomes the main layer. The main layer could decide a path for data transferring and manage load balancing or apply a method to save energy. The main layer could even change other layers behaviors. For example, application layer could change the TCP window size in order to control throughput, modifies routing tables, disconnect and reconnect the interfaces to manage failure or energy saving (LI *et al.*, 2016b).

Therefore, we categorize our research works into two groups: decision by application layer, and decision by transport layer, depending on which layer can be considered the main one, as discussed further in this subsection and summarized in Table 3.

2.3.4.1 Application Layer Decision

Corbillon et al. (CORBILLON *et al.*, 2016) proposed a cross layer approach with interaction between application and transport layer. In this approach, an adaptive mechanism is used to select the segments on application layer and MPTCP is used as transport protocol. The main goal of this approach is to maximize the amount of data that is received on time to destination. Therefore, it utilizes the benefit of being application aware to estimate playback deadline and it only sends the video units that have chance to arrive in time. As there is no cross layer feedback available in MPTCP, it is assumed that such a feedback exists and can be used. The feedback should indicate which path should be selected by MPTCP to send the next packet and only after that the cross layer scheduler would give MPTCP the data to send on this selected path (only one packet at a time). Therefore, the scheduler, which is content-aware, can decide if and when a video unit is given to the transport layer.

Ojanpera et al. (OJANPERÄ; VEHKAPERÄ, 2016) proposed a cross layer approach with interaction between application and network layer. The goal of this approach is to improve quality and availability of video streaming. The approach utilizes DASH to provide transparently bit rate adaptation support and MPTCP with default settings (coupled congestion control and default scheduling strategy) to provide multipath transmission capability. As explained in Section 2.3.1.2, rate adaptation method available in DASH system could perform more efficiently if it could access accurate network information. Therefore, in this work, a network management system, built upon the Distributed Decision Engine (DDE) framework, is proposed. DDE provides network information, including QoS, load, and capacity. Consequently, the client is adjusted to support DDE in order to incorporate the gathered network information into the bit rate adaptation decision in order to cope with changes in the network available bandwidth. Then, the MPTCP scheduler on the server side is responsible for mapping data on the different paths. For achieving network load balancing, the operator network management (of DDE) can dynamically disable the access network for the client by DDE signaling. MPTCP reacts to the event by stopping the usage of the corresponding path and mapping the traffic to other available paths. Finally, the results of the work show that using more network information for client bit rate adaptation decision outperforms standalone throughput-based by improving the stability of the video.

Wu et al. (WU *et al.*, 2015) developed a model, Goodput-Aware Load distribuTiON (GALTON), in application-network layer. GALTON optimizes the goodput performance of video streaming over multipath networks. Goodput is an application level throughput, a key parameter for video QoS and refers to the successfully received data at the receiver within the deadline. In GALTON, the receiver monitors network status (e.g., available bandwidth, RTT, PLR) and informs this information to the sender via feedback. The sender estimates the path quality based on the reported network information and detects congested paths by ZigZag scheme. There is also a proposed flow rate allocator which is responsible for partitioning flows to several subflows and assigning them to the available paths to optimize the aggregated goodput. It is also responsible for performing load balancing. Then, packets scheduled to the same path would be spread out within imposed deadline through the UDP connections. Besides that, scheduler adjusts probe rate and probing packet sizes dynamically over the congested paths.

Wu et al. (WU *et al.*, 2013) proposed a flow rate allocation-based Joint Source and Channel Coding (FRA-JSCC) approach in an application-physical layer. Joint Source and Channel Coding (JSCC) is an efficient solution for improving error-resilient in wireless video transmission. Therefore, in this work, JSCC is optimized to a FRA-JSCC for mobile video broadcasting in multipath networks. In FRA-JSCC approach, three main methods are proposed. First, FEC redundancy estimation to protect video data against channel losses. Second, source rate adaptation based on the calculated encoding rate. The encoding rate is concerned because high encoding rate makes more channel distortion and imposes high delay due to heavier load and network congestion. On the other hand, low encoding rate cannot provide the video delay requirements. Third, flow rate allocation is responsible to dynamically select the appropriate paths out of all available access networks and assign the transmission rates to them based on Weighted Round Robin (WRR) scheduling strategy.

2.3.4.2 Transport Layer Decision

Han et al. (HAN *et al.*, 2016) proposed MP-DASH framework, with overall goal of enhancing MPTCP to support adaptive video streaming (DASH) under user-specified interface preferences. For this goal, MP-DASH is designed as a cross layer approach with interaction between application and transport layer. In order to implement MP-DASH two components are designed: MP-DASH scheduler, and MP-DASH video adapter, as

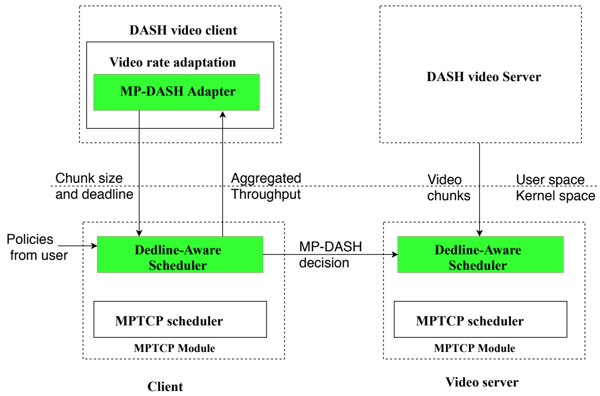


Figure 18 – MP-DASH system architecture (source: adapted from (HAN *et al.*, 2016)).

shown in Figure 18.

MP-DASH scheduler is implemented with MPTCP scheduler with knowledge of network interface preferences from the user and aggregated throughput. MP-DASH video adapter component, which is a lightweight add-on, is implemented to integrate the MP-DASH scheduler with DASH rate adaptation. Video adapter exchanges information between video player and MP-DASH scheduler (segment sizes and deadlines from video player to MP-DASH scheduler, and throughput from MP-DASH scheduler to the video player). This way, DASH algorithms becomes multipath friendly and MP-DASH scheduler becomes aware of delivery deadline. Besides that, MP-DASH splits the MP-DASH scheduling functions into two parts: decision function on the client, and enforcement function on the server. Decision function determines how to manage paths based on information from video player (e.g., segment sizes and deadlines), and enforcement function operates the decisions. The knowledge of network interface preferences is used to reduce cellular data usage while maintaining video QoE. Therefore, the approach starts data transferring with WiFi link and checks WiFi throughput dynamically to see if it is sufficient. If WiFi cannot deliver data before deadline time, the cellular network should be enabled. The results of the work show cellular usage reduced up to 99%, and radio energy consumption reduced up to 85% compared with the default MPTCP.

The work in (NAM *et al.*, 2016) proposed a dynamic MPTCP path control using Software-Defined Networking (SDN) (which makes cross layer approach of transport and network layer). The goal of the approach is to cope with out-of-order delivered packets to speed up download rate and improve video QoE in ABR streaming. In this work, the authors show the feasibility of using SDN platform regarding MPTCP. The SDN controller monitors information and estimates path capacity. Then, the SDN controller communicates periodically with the SDN clients to inform which paths are the best. The SDN platform on the client side removes poor and low capacity links because poor links increase the MPTCP reordering queue size. The removed paths attach again when they return to the proper capacity. Throughput measurement is used to find the available path capacity. It also may consider other multiple factors, such as RTT and delay to compute the best paths depending on the applications (e.g., video, VoIP or web surfing). Therefore, SDN application dynamically selects the proper paths and adjusts the number of paths in real-time. The evaluation shows that dynamically switch between MPTCP and SPTCP increases download time. In addition, the results of DASH implementation over the proposed dynamic MPTCP path control shows less bit rate change and rebuffering than without dynamic MPTCP path control.

Cross-layer fairness-driven SCTP-based CMT solution (CMT-CL/FD) approach (XU *et al.*, 2015) is a path quality-aware approach over CMT. In CMT-CL/FD, cross layer evaluates path quality by using loss rate information in Effective Signal-to-Noise Ratio (ESNR) (which is calculated at the link layer), and bandwidth or transmission rate information (which are estimated at the transport layer). ESNR is an upgrade calculation for signal-to-noise ratio/noise ratio (SNR) to evaluate wireless communication quality because the default SNR method has some shortcomings. For example, SNR is not accurate in real-time communication, and is not able to capture co-channel interference, frequency-selective fading and signal multipath effects (WU *et al.*, 2013). Then, CMT-CL/FD distributes data intelligently over different paths depending on their estimated quality. A loss-cause dependent retransmission (RTX) policy is also introduced to distinguish wireless loss from congestion loss. Consequently, in case of congested network, cwnd is changed and retransmission occurs (as explained in Section 2.3.2.5). Finally, this proposed approach mitigates reordering, losses, and consequently decreases HOL problem. However, none of these works use video content features for the scheduling strategy.

2.4 Scheduling, Resilience, and Path Selection

A key characteristic of video data is that, based on the en/decoding technology, packets may have unequal importance (e.g. I frames vs P frames). Considering the importance of each packet, different error protection levels can be applied. In addition, packets can be sent over different network paths based on paths quality to meet real-time deadlines, increase reliability, minimize out-of-order packet delivery, circumventing path heterogeneity issues (LI *et al.*, 2016b), as discussed in Section 4. Therefore, wirelesss multi-path video scheduling strategies need to consider, at least, three main functional aspects; packet selection, packet protection and path selection.

We now revisit the works surveyed in Section 2.3 through the new classification presented in Tables 5, 6, 7, and 8 based on the following questions:

- Which packet should be sent next?
- How to protect the packet?

Table 5 – Classification of research works according to scheduling functions - application layer.

Works	Which packet?		How to protect the packet?					Which path?				
	Content Awareness	Video Distortion (Frame Level)	JSCC/ Channel Level ARQ	DUP	FEC	Error Resilience/ Source Level Scalability	MDC	RTT/ Delay	PLR	Bandwidth/ Throughput/ Goodput	Delay Constraint	Video Distortion (Flow Level)
MRTP (MAO *et al.*, 2006)	N	N	Y	Y	Y	Y	Y	Y	Y	Y	N	N
MPRTP (SINGH *et al.*, 2013)	Y	N	Y	N	N	N	N	Y	Y	Y	N	N
Xing et al. (XING *et al.*, 2012)	N	N	Y	N	N	N	N	N	N	Y	N	N
RTRA (XING; CAI, 2014)	N	N	Y	N	N	Y	N	N	N	Y	N	N
Houzé et al. (HOUZÉ et al. 2016)	N	N	Y	N	N	N	N	Y	N	N	Y	N
Sohn et al. (SOHN *et al.*, 2015)	N	N	Y	N	N	Y	N	Paths pre-selected				
Afzal et al. (AFZAL et al., 2018)	Y	N	N	Y	N	N	N	Y	Y	Y	N	N
Evensen et al. (EVENSEN et al., 2010)	N	N	Y	N	N	N	N	Y	N	Y	N	N
Evensen et al. (EVENSEN et al., 2011)	N	N	Y	N	N	N	N	Y	N	Y	Y	N
Evensen et al. (EVENSEN et al., 2012)	N	N	Y	N	N	N	N	Y	N	Y	Y	N
Greenbag (BUI *et al.*, 2013)	N	N	Y	N	N	N	N	Y	N	Y	Y	N

Table 6 – Classification of research works according to scheduling functions - transport layer.

Works	Which packet?		How to protect the packet?					Which path?				
	Content Awareness	Video Distortion (Frame Level)	JSCC/ Channel Level ARQ	DUP	FEC	Error Resilience/ Source Level Scalability	MDC	RTT/ Delay	PLR	Bandwidth/ Throughput/ Goodput	Delay Constraint	Video Distortion (Flow Level)
BEMA (WU *et al.*, 2016a)	Y	Y	N	Y	Y	N	N	Y	Y	Y	Y	N
Freris at al. (FRERIS et al., 2013)	Y	Y	N	N	N	Y	N	Y	Y	Y	N	N
Correia at al. (CORREIA et al., 2012)	Y	N	N	?	N	N	Y	Paths pre-selected				
MPLOT (SHARMA et al., 2008)	N	N	Y	N	Y	N	N	Y	Y	N	N	N
MP-DCCP (HUANG et al., 2012)	Y	N	N	N	N	N	N	Y	Y	N	N	N
ADMIT (WU *et al.*, 2016b)	N	N	Y	N	Y	N	N	Y	Y	Y	Y	Y
MPTCP-SD (DIOP *et al.*, 2012)	Y	N	Y	N	N	N	N	Y	N	N	N	N
MPTCP-PR (DIOP *et al.*, 2012)	N	N	Y	N	N	N	N	Y	N	N	N	N
PR-MPTCP+ (CAO *et al.*, 2016)	Y	N	Y	N	N	N	N	Y	N	Y	Y	N
SRMT (SILVA *et al.*, 2016)	Y	N	Y	Y	N	N	N	Paths pre-selected				
CMT-QA (XU *et al.*, 2013)	N	N	Y	N	N	N	N	Y	N	N	N	N
CMT-DA (WU *et al.*, 2015)	N	N	Y	N	N	Y	N	Y	Y	Y	N*	Y
CMT-CA (WU *et al.*, 2016d)	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	N

- Which is the best path to send the packet?

2.4.1 Which Packet Should Be Sent Next?

One important scheduling task is selecting the next packet to be sent. Content awareness and video distortion at frame level are key features to select the proper packets. These features will be discussed in this subsection. Tables 5, 6, 7, and 8 present each category related to the protocol layer.

Note that, generally, ABR approaches rely on HTTP and separate TCP connections do not consider each one packet for data transmission and proper path for a DASH

Table 7 – Classification of research works according to scheduling functions - network layer.

Works	Which packet?		How to protect the packet?					Which path?				
	Content Awareness	Video Distortion (Frame Level)	JSCC/ Channel Level ARQ	DUP	FEC	Error Resilience/ Source Level Scalability	MDC	RTT/ Delay	PLR	Bandwidth/ Throughput/ Goodput	Delay Constraint	Video Distortion (Flow Level)
Yap at al. (YAP *et al.*, 2012)	N	N	Y	N	N	N	N	Paths pre-selected				
MARS (SUN *et al.*, 2016)	N	N	Y	N	N	N	N	Y	N	Y	N	N
BAG (Chebrolu et al., 2006)	N	N	N	N	N	N	N	Y	N	Y	N	N

Table 8 – Classification of research works according to scheduling functions - cross layer.

Works	Which packet?		How to protect the packet?					Which path?				
	Content Awareness	Video Distortion (Frame Level)	JSCC/ Channel Level ARQ	DUP	FEC	Error Resilience/ Source Level Scalability	MDC	RTT/ Delay	PLR	Bandwidth/ Throughput/ Goodput	Delay Constraint	Video Distortion (Flow Level)
Corbillon et al. (Corbillon et al., 2016)	Y	N	Y	N	N	N	N	Y	Y	Y	Y	N
Ojanperä et al. (OJANPERÄ et al. 2016)	N	N	Y	N	N	N	N	Y	N	N	N	N
GALTON (WU *et al.*, 2015)	N	N	N	N	Y	Y	N	Y	Y	Y	Y	N
FRA-JSCC (WU *et al.*, 2013)	N	N	N	N	Y	Y	N	Y	Y	Y	Y	Y
MP-DASH (HAN *et al.*, 2016)	N	N	Y	N	N	N	N	N	N	Y	Y	N
Nam et al. (NAM *et al.*, 2016)	N	N	Y	N	N	N	N	N	N	Y	N	N
CMT-CL/FD (XU *et al.*, 2015)	N	N	Y	N	N	N	N	Y	Y	Y	N	N

segment/subsegment need to be determined instead of packet (e.g., (XING *et al.*, 2012; XING; CAI, 2014; HOUZÉ *et al.*, 2016; EVENSEN *et al.*, 2010; EVENSEN *et al.*, 2011; EVENSEN *et al.*, 2012; BUI *et al.*, 2013)). However, when using MPTCP for HTPP-based ABR video, the MPTCP scheduler performs its own transport-level scheduling for the received DASH data stream.

2.4.1.1 Content Awareness

Considering video content features in the scheduling strategy helps to define the priority of each packet, and subsequently choose the frame packets with higher priority to send it first or via more qualified paths. In video streaming, some frames have higher effect on video quality, and large frame inter-dependency. For example, I frames have highest priority among other frames. These strategies are generally referred to as content-aware scheduling strategies. In addition, a content-aware scheduling strategy could use stronger packet protection for higher priority packets than the less priority packets, for example, by applying adaptive FEC, which will be explained in next subsection. On the other hand, if the scheduler is unaware of the video content features, the sending buffer would transmit data packets in the same order as they arrived in the buffer (FIFO) without considering the priority of packets (e.g., MPTCP scheduler).

Video content features are considered as inputs to the scheduling strategy in the following works: MPRTP (SINGH *et al.*, 2013), (AFZAL *et al.*, 2018), BEMA (WU *et al.*, 2016a), (FRERIS *et al.*, 2013), (CORREIA *et al.*, 2012), MP-DCCP (HUANG *et al.*, 2012), MPTCP-SD (DIOP *et al.*, 2012), PR-MPTCP$^+$ (CAO *et al.*, 2016), CMA-CA (WU *et al.*, 2016d), (CORBILLON *et al.*, 2016). In SRMT (SILVA *et al.*, 2016), the primary path is used for all data while the secondary paths are used to send redundant packets, which are, in turn, chosen based on their priority (e.g. I frame packets have highest priority).

2.4.1.2 Video Distortion (Frame Level)

Video distortion impacts perceived video quality. Generally, video distortion is considered at both frame level and flow level. In this section, we study the frame level video distortion because it assesses inter-frame dependencies and analyzes each specific video frame, including the frame priority and decoding dependency (FRERIS *et al.*, 2013). We will discuss flow level video distortion in Section 2.4.3.5. In particular, frame level distortion refers to the quality degradation of each frame of GoP after data transmission and video decoding process (WU *et al.*, 2016a). This way, the frame level distortion is calculated as a total of truncation and drifting distortion. The truncation distortion refers to the video quality degradation caused by packet drops during transferring data, and the drifting distortion refers to the video quality distortion occurred by imperfect reconstruction of parent frames which are used for inter-frame prediction. In the research works, frame level distortion is used by BEMA (WU *et al.*, 2016a) for calculating FEC coding parameters (e.g., code rate and symbol size), and also it is used by (FRERIS *et al.*, 2013) to assign higher priority values to the pictures which minimize the distortion of the decoded video affected by packet loss. Such information could also be used for path selection in CMT-CA (WU *et al.*, 2016d).

2.4.2 How to Protect the Packet?

Providing packet protection techniques to the scheduler leads to data loss rate decreases, and consequently, better video streaming throughput and QoE. In fact, interdependency among video frames causes a compressed video to be very sensitive to data loss. By this idea (GALL, 1991), individual frames of pictures are grouped together, which is called GoP. Each GoP consists of one initial Intra (I)-frame, several Predicted (P)-frames and possibly Bidirectional (B)-frames (FANG; CHAU, 2005). While an I frame is encoded without reference to any other video frames, but a P frame is encoded with reference to previous I or P frames, and a B frame is encoded with reference to both immediate previous and forward I or P frames. Therefore, in the decoding process, loss of some frames may preclude a proper decoding, especially in the miss of I frames. Thus, it is important to protect frames (especially I frames) in lossy wireless channels. For this purpose, some JSCC/Channel Level and Error Resilience/Source Level techniques have been implemented. These techniques will be discussed in this subsection. Tables 5, 6, 7, and 8 present each category of such techniques divided by the protocol layer.

2.4.2.1 Joint Source and Channel Coding (JSCC)/Channel Level Techniques

The channel level techniques for JSCC are Automatic Repeat reQuest (ARQ), Duplication (DUP) and Forward Error Correction (FEC).

Automatic Repeat reQuest (ARQ) retransmits requests to provide reliable data transmission. The retransmission occurs in case of packets lost or received with bit error. Inherently, all protocols atop or extensions of TCP (e.g., HTTP, DASH, MPTCP) use ARQ. However, the retransmission wastes bandwidth, causing network congestion, and consequently, increasing End-to-End delay. For example, in efforts to mitigate these problems: CMT-QA (XU *et al.*, 2013) retransmits packets over the path with minimum transfer delay; CMT-DA (WU *et al.*, 2015) and CMT-CA (WU *et al.*, 2016d) retransmit only the estimated packets to arrive at the destination within the deadline; and CMT-CL/FD (XU *et al.*, 2015) selects the path with the largest cwnd for the retransmission, which sends the lost packet before all the other packets that exist in the path buffer. In addition, considering the existence of many clients in multicast communications, responding to the retransmission requests of all clients might be difficult for the server.

Other research works, which utilize ARQ as JSCC technique are MRTP (MAO *et al.*, 2006), MPRTP (SINGH *et al.*, 2013), (HOUZÉ *et al.*, 2016), (XING *et al.*, 2012), RTRA (XING; CAI, 2014), (SOHN *et al.*, 2015), (EVENSEN *et al.*, 2010), (EVENSEN *et al.*, 2011), (EVENSEN *et al.*, 2012), Greenbag (BUI *et al.*, 2013), MPLOT (SHARMA *et al.*, 2008), ADMIT (WU *et al.*, 2016b), MPTCP-SD (DIOP *et al.*, 2012), MPTCP-PR (DIOP *et al.*, 2012), PR-MPTCP$^+$ (CAO *et al.*, 2016), SRMT (SILVA *et al.*, 2016), (YAP *et al.*, 2012), MARS (SUN *et al.*, 2016), (CORBILLON *et al.*, 2016), (OJANPERÄ; VEHKAPERÄ, 2016), MP-DASH (HAN *et al.*, 2016), and (NAM *et al.*, 2016).

Packet duplication (DUP) refers to concurrently transmission of packet duplicates through all available paths. While this could protect packets and improve the QoE, sending all the video packets through different paths leads to significant congestion and reduce the throughput. Instead, it is possible to duplicate only selected packets according to their importance for image reconstruction. For example, in our research works, BEMA (WU *et al.*, 2016a) concurrently transmits only I frame packets through all paths. The approach proposed in (AFZAL *et al.*, 2018) not only considers the video content but also network quality conditions. This way, the approach duplicates I frame packets if only all available paths are highly congested. Accordingly, MRTP (MAO *et al.*, 2006) and SRMT (SILVA *et al.*, 2016) utilize duplication to protect packets.

Forward Error Correction (FEC) appeared to remedy the shortcoming of packet retransmission and delay constraints, especially for live video streaming. Such a technique is also applied in multicast communication and whenever retransmission is costly or impossible, for instance, in one-way communication links (HUO *et al.*, 2015).

FEC can be applied to circumvent packet erasures/loss by cross-packets FEC in the application or transport layer (inter-packet FEC), and/or to handle bit errors in the physical layer (ZHAI *et al.*, 2004) (intra-packet FEC). In wired networks, it can have packet loss and packet truncation due to congestion. Therefore, either the packets are dropped by the network routers or the receiver due to excessive delay. In wireless networks, besides packet loss and packet truncation, there exists also bit errors due to noisy channels. Next, more details about inter- and intra-packet FEC techniques are provided.

In inter-packet FEC, redundant/parity packets are commonly generated in addition to source packets to perform cross-packet FEC, which is usually achieved by erasure codes. These allow the receiver to detect error packets and correct data without retransmission. The capability of FEC to recover the lost data depends on the added redundant symbols. Among the many existent erasure codes, the most commonly studied ones are Reed-Solomon (RS) (FROSSARD, 2001), Low-Density Generator Matrix (LDGM) (NAKACHI *et al.*, 2013) and Raptor codes (SHOKROLLAHI, 2006). In our research works, ADMIT (WU *et al.*, 2016b), GALTON (WU *et al.*, 2015) and FRA-JSCC (WU *et al.*, 2013) utilize RS due to stringent delay constraint. MPEG-H part 10 defines several MMT AL-FEC algorithms, including RS codes and LDGM. Raptor coding is used in BEMA (WU *et al.*, 2016a) due to low processing time and high error correction capability. Such erasure codes could be applied at frame level, GoP level, or subGoP level for video protection (WU *et al.*, 2013).

In frame level (KUO *et al.*, 2014), the frames in each GoP are classified in terms of their type and their distance from the leading I frame. Then, FEC is applied on the frames according to their priority. Besides, low priority frames can be dropped based on network conditions. In GoP level (see Figure 19), each GoP packetizes in k source packets. Then, FEC encoding maps source packets to some encoded packets. A FEC block of n data packets contains of k source packets, and $(n-k)$ redundant packets. Redundancy in FEC is calculated as $(n-k)/k$, and the code rate is equal to k/n. In SubGoP level (WU *et al.*, 2016a), each GoP consists of several subgroups, each mapped to a source block. In our research works, GoP level is used in ADMIT (WU *et al.*, 2016b), GALTON (WU *et al.*, 2015), FRA-JSCC (WU *et al.*, 2013), and SubGoP level is used in BEMA (WU *et al.*, 2016a).

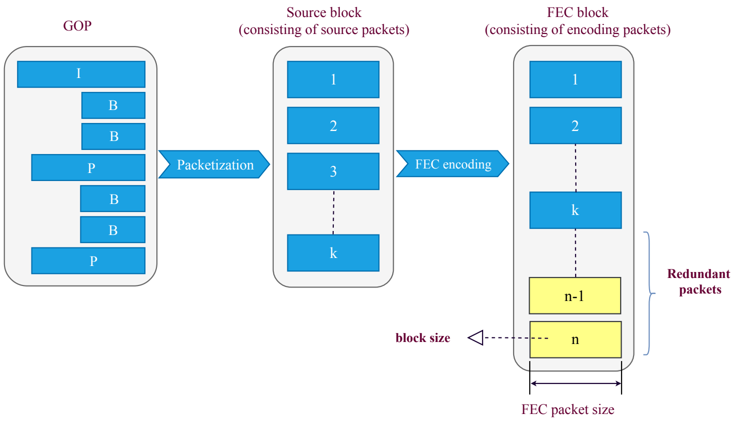


Figure 19 – GoP level FEC technique.

A trade-off between bandwidth/End-to-End delay and FEC redundancy is required. In particular, a smaller FEC packet size indicates a larger FEC block size due to the larger number of redundant packets (WU *et al.*, 2016b). While higher redundancy leads to better recoverability, it also increases overhead rate and bandwidth consumption. Consequently, congestion, packet reordering, FEC decoding delay and End-to-End delay have their probability increased, especially in the presence of burst losses. Therefore, an adaptive FEC is required to minimize these problems (e.g., bandwidth consumption and End-to-End delay), and maximize the recoverability by adaptively changing FEC parameters (e.g., adequate FEC packet size and FEC redundancy) according to the network channel status, application delay characteristics, or based on the importance of content data. For example, a stronger FEC would be used in a more lossy channel while not required in a more stable channel with less loss rate percentage, or more robust FEC could also be used only for I frames rather than B or P frames.

In our research works, adaptive FEC is used in several works, like FRA-JSCC (WU *et al.*, 2013) and GALTON (WU *et al.*, 2015) to find FEC redundancy, ADMIT (WU *et al.*, 2016b) to adjust FEC redundancy and code rate, and BEMA (WU *et al.*, 2016a) to set code rate and symbol size. Moreover, MPLOT (SHARMA *et al.*, 2008) also adaptively chooses block sizes, considering the usage of large block sizes in order to reduce bursty loss for delay-tolerant applications. We also identified FEC usage in MRTP (MAO *et al.*, 2006).

Besides using FEC method, an adequate technique is also requested to distinguish losses due to traffic congestion with the ones caused by wireless channel disturbances and impairments. It is based on the fact that FEC redundancy in wireless lossy networks leads to better packet recovery; however, adding more FEC redundancy in a congested network worsens network situation since it pushes higher congestion and more losses (WALLACE; SHAMI, 2012) due to bit stuffing operations. More technical details on packet loss differentiation are provided in Section 2.5.1.

In intra-packet FEC, channel coding is applied to correct bit errors in the physical layer. Turbo Codes (parallel Concatenated Constitutional coding) and Low-Density Parity-Check (LDPC) codes are generally used. Error detection is performed at the link layer, based on Cyclic Redundancy Check (CRC). Due to this approach, only packets passing CRC stage are visible on the network/Internet layer.

Therefore, FEC provides reliable access network and End-to-End video distortion minimization. Moreover, a joint-ARQ and FEC usage approach can enhance efficiency, depending on the adopted strategy to couple both techniques. For example, in our research works, ADMIT (WU *et al.*, 2016b) utilizes FEC for reconstructing data, and consequently, it leads to delay reduction, making video data ready for fast video playback. However, there is no additional help to mitigate the number of retransmissions and bandwidth consumption increases drawbacks, since there is no ACK message sending to inform the server that the data is successfully reconstructed. Therefore, the MPTCP protocol on the server keeps sending retransmission of each lost packet until it receives the ACK from the receiver. This scenario outlines a motivation for a proper ARQ-and-FEC joint approach, using FEC for data protection, while retransmitting events only occur when there is no way to perform data reconstruction.

2.4.2.2 Error Resilience/Source Level

Besides employing JSCC techniques to recover from packet loss and bit errors, increasing the error resilience of the video sequence itself is also an important task. To provide this functionality, error resilience techniques embrace, among others, the usage of Scalable Video Coding (SVC) and Multiple Description Coding (MDC) methods.

In SVC (KAZEMI *et al.*, 2014), source video is encoded in one base layer and several enhancement layers. These layers are hierarchically dependent to each other. This means that, at the receiver, each layer can be decoded only when its lower layers have been correctly received. Therefore, video quality is improved based on the number of received enhancement layers. In order to improve the efficiency of SVC, base layer is often protected by FEC or it is transmitted through more reliable paths, due to its importance. In the proposed approach (FRERIS *et al.*, 2013), each packet is transmitted to the network only if all other related packets in lower layers have been sent before. Other research works, which utilize SVC as Error Resilience are MRTP (MAO *et al.*, 2006), RTRA (XING; CAI, 2014), (SOHN *et al.*, 2015), CMT-DA (WU *et al.*, 2015), GALTON (WU *et al.*, 2015) and FRA-JSCC (WU *et al.*, 2013).

In MDC (KAZEMI *et al.*, 2014), source video is encoded into several independent compressed streams which are called descriptions. Each description can be decoded independently and shall provide acceptable quality. When one or more descriptions arrive at the receiver, a video with a certain quality level would be made by the decoder. MDC is a good alternative to retransmission in order to remedy the delay constraint in real-time video streaming.

According to a reviewed work about MDC techniques for video streaming (KAZEMI *et al.*, 2014), MDC is more useful than FEC in the case of high lossy networks, since FEC uses long code block sizes, increasing bandwidth consumption as well. MDC also outperforms SVC in high lossy networks, but SVC is more proper than MDC in low loss rate networks, due to overhead reduction. MDC is also recommended for multicast with heterogeneous receivers (KOBAYASHI *et al.*, 2009). Accordingly, works like (CORREIA *et al.*, 2012) and MRTP (MAO *et al.*, 2006) utilize MDC as error resilience technique.

2.4.3 Which Is the Best Path to Send the Packet?

Before discussing how could select the proper path to transfer the packet, it is worthwhile to mention that using many paths for data transmission does not always lead to better QoE, since many paths for video delivery make large overheads due to parallel connections (HABIB *et al.*, 2016). According to (MITZENMACHER, 2001), it is possible to achieve maximum multipath benefits with just using two paths by using a proper scheduling strategy.

The simplest scheduling strategy is Round Robin (LI *et al.*, 2016b). This strategy sorts paths and sends data to the next available path in circular order without taking into account the heterogeneous paths' characteristics. In Round Robin strategy, slow channels would be overloaded while fast channels remain underutilized (e.g., CMT (IYENGAR *et al.*, 2006)).

Obviously, scheduling strategies that are aware of path characteristics (e.g., RTT, packet loss rate) generate wiser scheduling decisions. These strategies generally referred to path-aware scheduling strategies. For example, Weighted Round Robin (WRR) is a scheduling strategy which assigns weight to each path. Weight shows path capability regarding available bandwidth/delay/packet loss rate. This way, data distribution is proportional to the path transmission capability (e.g., MPTCP and FRA-JSCC (WU *et al.*, 2013)). Earliest Delivery Path First (EDPF) is another scheduling strategy that estimates the delivery time of each packet over each path. Then, the packets are transmitted over the fastest path in order to prevent from missing their deadlines and minimizing packet-reordering (e.g., BAG (CHEBROLU; RAO, 2006) and MPLOT (SHARMA *et al.*, 2008)).

Finding End-to-End path capability of real-time video traffic communication leads to estimate path quality or path reliability (AFZAL *et al.*, 2018), (WU *et al.*, 2016b), (WU *et al.*, 2016a), (XU *et al.*, 2013). Therefore, scheduling strategy could map higher priority packets to the more reliable or qualified paths (assume that it is a combination of content-aware and path-aware scheduling strategy).

It is important to note here that mapping many packets to most qualified or reliable paths pushes congestion over that path, and consequently decreases video quality, which is called load imbalance problem (WU *et al.*, 2015). Therefore, using a method to balance the data over the available paths is required. In our research works, BEMA (WU *et al.*, 2016a), (FRERIS *et al.*, 2013), ADMIT (WU *et al.*, 2016b), CMT-CA (WU *et al.*, 2016d), CMT-DA (WU *et al.*, 2015) and GALTON (WU *et al.*, 2015) use load balancing mechanism to avoid imbalance problem.

Most network characteristics used to find the quality or reliability probability of network channels are RTT/Delay, PLR, Available bandwidth/Throughput/Goodput. There are also some other metrics that lead to better path selection and scheduling decision, such as delay constraint and video distortion at flow level. These network characteristics and metrics will be discussed in this subsection. Tables 5, 6, 7, and 8 present each category per protocol layer.

2.4.3.1 RTT/Delay

Round Trip Time (RTT) is the time required for a packet to be sent plus the time it takes to receive an ACK of that packet (WU *et al.*, 2015; WU *et al.*, 2016b). Therefore, RTT consists of the packet transmission time and path propagation delay (WU *et al.*, 2016b). In order to avoid sudden variations of RTT, some approaches (e.g., MPTCP and SCTP) apply a smoothing factor to the RTT which is called smooth Round Trip Time (sRTT). In approaches without ACK method, for example, UDP-based approaches, one-way delay could be considered instead of RTT.

Considering RTT/Delay for path scheduling decreases the probability of expired arrival packets, stall or out-of-order packet delivery. In our research works, MARS (SUN *et al.*, 2016), which is implemented over separate TCP connections, utilized a relative RTT measurement method based on OpenFlow protocol. In this approach, duplicated packets (probes) are sent through different interfaces. The probes would return to the sender through the common reverse path from the edge switch close to the client side. The transfer process can be implemented with the tables of OpenFlow at the edge switch. The approach measures the relative delay of forward paths instead of their absolute delay because, in case of absolute forward path delays, the tight clock synchronization between sender and receiver is required. More information and comparison details between relative and absolute delay can be found at (RIBEIRO; LEUNG, 2006).

In SCTP protocol, the acknowledgment of the sent packet (SACK) can be transmitted over different paths. Mostly the acknowledgment packet returns through the most reliable path to mitigate the probability of dropped or overdue feedback packets. Since paths have different delay characteristics, the estimated RTT is incorrect and using this estimated RTT to find the path quality leads to the wrong result. For this reason, CMT-QA (XU *et al.*, 2013) does not use RTT directly. Instead, it uses transmission delay. Transmission delay refers to the time difference between the time of the first chunk entering each path sender buffer from a group of distributed data chunks and the time of the last chunk leaving the path sender buffer. CMT-CL/FD (XU *et al.*, 2015) utilizes the SCTP heartbeat mechanism to calculate RTT. In this mechanism, the HEARTBEAT-ACKs have to return through the same path used to send the HEARTBEAT messages.

Since in RTCP protocol, which is generally used by RTP, is possible to calculate RTT by using sender and receiver reports, the multipath transmission approaches over RTP, such as MRTP (MAO *et al.*, 2006) and MPRTP (SINGH *et al.*, 2013) extended RTCP in order to calculate RTT in multipath transmission solutions.

FRA-JSCC (WU *et al.*, 2013) and BAG (CHEBROLU; RAO, 2006), which are the approaches that use UDP as transport protocol, utilize propagation delay. FRA-JSCC (WU *et al.*, 2013) calculates propagation delay network characteristic by using the existing time stamp in each packet header.

RTT/Delay is also used for packet loss differentiation decision in CMT-QA (XU *et al.*, 2013), CMT-CL/FD (XU *et al.*, 2015), BEMA (WU *et al.*, 2016a), ADMIT (WU *et al.*, 2016b), GALTON (WU *et al.*, 2015), and CMT-CA (WU *et al.*, 2016d). More technical details on packet loss differentiation are provided in Section 2.5.1. Besides, RTT/Delay can also be used for other tasks. For example, MRTP (MAO *et al.*, 2006) sets retransmission timeout value by RTT, and Greenbag (BUI *et al.*, 2013) utilizes RTT to determine when to send requests for the next segments.

Other research works, which consider RTT/Delay network characteristic for their scheduling decision are (HOUZÉ *et al.*, 2016), (AFZAL *et al.*, 2018), (EVENSEN *et al.*, 2010), (EVENSEN *et al.*, 2011), (EVENSEN *et al.*, 2012), (FRERIS *et al.*, 2013), MPLOT (SHARMA *et al.*, 2008), MP-DCCP (HUANG *et al.*, 2012), MPTCP-SD (DIOP *et al.*, 2012), MPTCP-PR (DIOP *et al.*, 2012), PR-MPTCP$^+$ (CAO *et al.*, 2016), CMT-DA (WU *et al.*, 2015), (CORBILLON *et al.*, 2016), and (OJANPERÄ; VEHKAPERÄ, 2016).

2.4.3.2 PLR

Packet Loss Rate (PLR) comprises of network transmission lost packets, which are lost/error arrived packets during the communication paths, and the expired arrival packets (overdue) (WU *et al.*, 2015). Three basic reasons cause packet losses (XU *et al.*, 2013); 1) congestion due to limited bandwidth or buffer size, 2) noise or interference in the wireless networks, 3) path failure or handover. Therefore, sending highest priority frame packets on the paths with less PLR leads to better QoE. Besides, PLR network characteristic and distinguishing packet loss differentiation are key factors for adaptively FEC protection (Section 2.4.2.1), avoiding unnecessary fast retransmission (Section 2.3.2), and video distortion estimation (Section 2.4.1.2 and Section 2.4.3.5).

PLR is considered for scheduling decision in the following works: MRTP (MAO *et al.*, 2006), MPRTP (SINGH *et al.*, 2013), (AFZAL *et al.*, 2018), BEMA (WU *et al.*, 2016a), (FRERIS *et al.*, 2013), MPLOT (SHARMA *et al.*, 2008), MP-DCCP (HUANG *et al.*, 2012), ADMIT (WU *et al.*, 2016b), CMT-DA (WU *et al.*, 2015), CMT-CA (WU *et al.*, 2016d), (CORBILLON *et al.*, 2016), GALTON (WU *et al.*, 2015), FRA-JSCC (WU *et al.*, 2013), and CMT-CL/FD (XU *et al.*, 2015).

2.4.3.3 Available Bandwidth/Throughput/Goodput

Available bandwidth is defined as the maximum video rate that can be transmitted over End-to-End path (WU *et al.*, 2016b). Different methods are introduced to estimate available bandwidth in the literature (PAUL *et al.*, 2016; JAIN; DOVROLIS, 2002; ZHOU *et al.*, 2008). Some approaches utilize throughput or goodput for this purpose. The amount of data that could traverse through a path is known as throughput. Throughput refers to all useful and not useful data, including data retransmissions, and overhead data (e.g., headers). If the scheduler considers only throughput among all network characteristics, it may distribute packets over high loss rate channels, and consequently, serious degrade of goodput performance and video quality occurs (WU *et al.*, 2016c). Goodput refers to the amount of useful data (exclusive protocol overhead or retransmission) delivered successfully to the destination within the imposed specific deadline(WU *et al.*, 2015). Goodput is also known as application level throughput. Regarding (WU *et al.*, 2016b), the approaches over HTTP/TCP could estimate the available bandwidth by using the observed TCP throughput. In our research works, (FRERIS *et al.*, 2013) measures bandwidth by using Abing (SUEHRING, 2017). GALTON (WU *et al.*, 2015) and FRA-JSCC (WU *et al.*, 2013) implement pathChirp algorithm (RIBEIRO *et al.*, 2003) for this purpose. CMT-CL/FD (XU *et al.*, 2015) computes available bandwidth as the ratio between the average packet length and average inter-packet sending time. CMT-CA (WU *et al.*, 2016d) and CMT-DA (WU *et al.*, 2015) believe that cwnd has effect on bandwidth, therefore, these works calculate it as ($cwnd/RTT$). In RTRA (XING; CAI, 2014), once a segment has been successfully downloaded, the transmission bandwidth would be calculated as division of the total size of transmitted data over the transmission time, and then, a Markov channel model is used to estimate future available bandwidth.

Other research works, which consider Available bandwidth/Throughput/Goodput network characteristic for their scheduling decision are MRTP (MAO *et al.*, 2006), MPRTP (SINGH *et al.*, 2013), (XING *et al.*, 2012), (AFZAL *et al.*, 2018), (EVENSEN *et al.*, 2010), (EVENSEN *et al.*, 2011), (EVENSEN *et al.*, 2012), Greenbag (BUI *et al.*, 2013), BEMA (WU *et al.*, 2016a), ADMIT (WU *et al.*, 2016b), PR-MPTCP$^+$ (CAO *et al.*, 2016), MARS (SUN *et al.*, 2016), BAG (CHEBROLU; RAO, 2006), (CORBILLON *et al.*, 2016), MP-DASH (HAN *et al.*, 2016), (NAM *et al.*, 2016).

2.4.3.4 Delay Constraint

A real-time video application imposes a decoding deadline. In this manner, the overdue packets cannot handle at the decoder, even if they arrive successfully. Therefore, the End-to-End delay has to be less than delay constraint (WU *et al.*, 2016b). Besides that, considering delay constraint in scheduling strategy could also avoid playback buffer starvation (WU *et al.*, 2016a).

In our research works, the delay constraint of GALTON (WU *et al.*, 2015), ADMIT (WU *et al.*, 2016b), FRA-JSCC (WU *et al.*, 2013) and CMT-CA (WU *et al.*, 2016d) are set with values 300, 500, 250 and 100 ms for each video frame respectively. This value in BEMA (WU *et al.*, 2016a) is set equal to its playback duration, so the delay constraint should be 40 ms if the video is encoded at 25 frames per second. GALTON (WU *et al.*, 2015) uses delay constraint to compute transmission intervals in order to mitigate consecutive losses. ADMIT (WU *et al.*, 2016b) calculates the rate allocation vector and FEC coding parameters respect to delay constraint. FRA-JSCC (WU *et al.*, 2013) finds source rate adaption under delay constraint. CMT-CA (WU *et al.*, 2016d) finds the optimal congestion window sizes and frame scheduling vector to mitigate video distortion. While CMT-DA (WU *et al.*, 2015) is not appropriate for the video streaming with stringent delay constraint but the retransmission method is based on the delay constraint. In the work (FRERIS *et al.*, 2013), the same deadline time is assumed for all users, which is determined as a system parameter by the service provider. Then, this is used to find packet loss probability.

Proposed approaches in (HOUZÉ *et al.*, 2016), (EVENSEN *et al.*, 2011; EVENSEN *et al.*, 2012) and GreenBag (BUI *et al.*, 2013) are application-aware, therefore, they are aware of buffer level at the receiver in order to calculate the delay constraint. These approaches utilize adaptive streaming over multiple separate TCP connections, and mostly path selection is integrated with the adaptation logic. In works (EVENSEN *et al.*, 2011) and (EVENSEN *et al.*, 2012), the delay constraint is calculated by the client to select the suited bit rate. The client calculates the amount of already received content to playout in the buffer (transfer-deadline) and estimates how long it takes to receive the already requested data (pipeline-deadline). The difference between pipeline-deadline and transfer-deadline shows the amount of time that the client can wait to receive the next segment without interruption. Then, this estimation is compared with the estimation of the times it takes to receive the desired segment in the different bit rates, and the most proper bit rate is selected. After that, the segment is divided into subsegments. The size of each subsegment is decided based on the measured throughput of each interface that it will be requested through. The approach in (HOUZÉ *et al.*, 2016) finds suited segment bit rate with checking the size of the first frame in each segment representation. It chooses the representation with the highest bit rate and high probability to get the frame on time. Then, it finds the best size of byte range per path dynamically based on paths' RTT. GreenBag (BUI *et al.*, 2013) utilizes paths' delay and available bandwidth to determine per path subsegment size. If one path received its subsegment within a segment, but the other path is significantly lagging, so, the former path takes over some portion of the problematic path to recover. The above mentioned approaches could achieve zero or close to zero interruption during playback time.

Two more other application-aware approaches with concerning delay constraint are (CORBILLON *et al.*, 2016) and MP-DASH (HAN *et al.*, 2016). These approaches utilize adaptive streaming over MPTCP paths. MP-DASH (HAN *et al.*, 2016) feed the modified MPTCP with the deadline of each video data unit in order to further use and path selection. The approach in (CORBILLON *et al.*, 2016) understands the display time of each video unit with access to the Picture Order Counts (POC) and the coding identifier of each frame (because it is content awareness). Therefore, the approach estimates the deadlines and ignores transmission of packets which will miss their playback deadline and instead, assigns more priority to the packets which their deadline time is close. The high priority packets can be spread through less RTT paths. This helps to use bandwidth more efficiently and experience less video distortion. In PR-MPTCP$^+$ (CAO *et al.*, 2016), when the network is detected as congested, only the packets with enough deadline time to play would be sent.

2.4.3.5 Video Distortion (Flow Level)

We previously discussed frame level video distortion in Section 2.4.1.2. Here, we study flow level video distortion. End-to-End video distortion at flow level (intra-coding) is calculated as total of source and channel distortion (WU *et al.*, 2015). Source distortion is determined by the video source rate and video sequence parameters because of their impact on the efficiency of video codec. For example, in case of the same video encoding rate, a more complex video sequence has higher distortion. As another example, increasing the video encoding rate causes decreasing distortion. Channel distortion refers to the packet losses during the network transmission and expired arrivals. Some other features including the frame structure and GoP size also have an impact on both the source and the channel distortion. Flow level video distortion is considered for scheduling strategy in the following research works: ADMIT (WU *et al.*, 2016b), CMT-DA (WU *et al.*, 2015) and FRA-JSCC (WU *et al.*, 2013).

Although most important network characteristics and metrics for path selection were discussed, but there are some other parameters that are used directly or indirectly (to calculate RTT, PLR or other metrics) by different approaches. For example, cwnd is used in MPLOT (SHARMA *et al.*, 2008), MP-DCCP (HUANG *et al.*, 2012) (CCID2), CMT-CA (WU *et al.*, 2016d) and CMT-DA (WU *et al.*, 2015), sending rate is used in MP-DCCP (HUANG *et al.*, 2012) (CCID3) and CMT-CL/FD (XU *et al.*, 2015), cost function is utilized in MP-DASH (HAN *et al.*, 2016) and GreenBag (BUI *et al.*, 2013). In MP-DASH, cost can be data usage, energy consumption or both, and in GreenBag (BUI *et al.*, 2013), cost refers to energy consumption. Other useful factors can be buffer size, packet size, packet count and etc.

2.5 Analysis and Comparison of Methods and Techniques

In the previous two sections, we analyzed different multipath wireless video streaming works based on layer dependency and scheduling functions. In this section, we study other effected features and related methods that are used in these works. Table 9 reclassified the candidate previously explained research works based on the features or methods the authors used.

2.5.1 Packet Loss Differentiation

A packet loss differentiation method can distinguish congestion losses from wireless losses. In the heterogeneous wireless networks, packet losses due to lossy channels, handover, noise or interface in the wireless network occur more than losses due to congestion (XU *et al.*, 2013). Identifying reason for losses is essential. For example, if losses occur because of congestion in the network, then retransmission or adding more FEC redundancy pushes worse congestion and more losses (WALLACE; SHAMI, 2012) (Section 2.4.2). But, decreasing cwnd mitigates congestion. On the other hand, if losses occur because of wireless lossy network, then decreasing cwnd drops goodput sharply (Section 2.3.2). But, adding more FEC redundancy leads to better recovery. Therefore, with an accurate loss differentiation method could react properly to the network situation.

In our research works, MPRTP (SINGH *et al.*, 2013) categorizes a path as a lossy one if feedback reports show only transmission losses and no discards (overdue packets)

Table 9 – Comparison of approaches.

Works	**Packet loss Differentiation Method**	**Fairness**	**Video Compression**	**Error Concealment**	**Experimental Environment**	**Performance Metrics**	**Video Services**
MRTP (MAO *et al.*, 2006)	Not used	N	Not defined	Not defined	OPNET	PSNR, Bandwidth utilization, Buffer overflow probability, Playout buffer size	Real-time
MPRTP (SINGH *et al.*, 2013)	Used	Y	H.264/AVC	x264	Realistic testbed, NetEm, Disjoiont paths, Client interfaces: WiFi and 3G or multiple 3G	PSNR, Loss rate, Bandwidth utilization, Connection setup time	Live, Real-time
Xing et al. (XING *et al.*, 2012)	Not used	N	H.264/AVC	x264 encoder, FFmpeg decoder	Realistic testbed, Android framework, Disjoint paths, Client interfaces: WiFi and 3G	Playback fluency average, Playback quality, Quality switch, Average 3G traffic, Playback traces, Buffer occupancy	Not defined
RTRA (XING; CAI, 2014)	Not used	N	H.264/SVC	JSVM	Realistic testbed, Android framework, Client interfaces: WiFi and Bluetooth	PSNR, Startup delay, Playback fluency average, Playback quality, Quality switch, Bandwidth utilization, Playback traces, Buffer occupancy	Real-time
Houzé et al. (HOUZÉ *et al.*, 2016)	Not used	N	HEVC	HM	NS-3, Client interfaces: five homogeneous xDSL links	Cumulative Distribution Function (CDF) of frame sizes, QoE (SAMVIQ method)	Live
Afzal et al. (AFZAL *et al.*, 2018)	Not used	N	H.264	FFMPEG	NS3-DCE, Client interfaces: LTE, WiFi (802.11n)	PSNR, SSIM, Goodput, Loss rate, I and NI frame packet loss rate, Delay	Real-time
Sohn et al. (SOHN *et al.*, 2015)	Not used	N	SHVC	JSVM	Own visual studio implementation, Client interfaces: WiFi and Ethernet	Throughput, Play time for base layer, Quality switch	Live, VoD
Evensen et al. (EVENSEN *et al.*, 2010)	Not used	N	Not defined	Not defined	Realistic testbed, Ubuntu framework, NetEm, Client interfaces: WiFi (IEEE 802.11b) and Cellular (HSDPA)	Quality distribution, Missed deadlines, Throughput	Live
Evensen et al. (EVENSEN *et al.*, 2011)	Not used	N	Not defined	Not defined	Realistic testbed, Ubuntu framework, NetEm, Client interfaces: WiFi (IEEE 802.11b) and Cellular (HSDPA)	Quality distribution, Missed deadlines, Throughput	Live
Evensen et al. (EVENSEN *et al.*, 2012)	Not used	N	Not defined	Not defined	Realistic testbed, Ubuntu framework, NetEm, Client interfaces: WiFi (IEEE 802.11b) and Cellular (HSDPA)	Quality distribution, Missed deadlines, Throughput	Live, VoD
GreenBag (BUI *et al.*, 2013)	Not used	N	Not defined	Not defined	Realistic testbed, Own C and JAVA implementation, Android framework, NetEm, Client interfaces: WiFi and LTE	Playback time, Interruption time, Energy consumption, Buffer size, In-order data	Real-time
BEMA (WU *et al.*, 2016a)	ZigZag	Y	H.264/AVC	JM	Exata, Client interfaces: Cellular, WiFi (802.11a/g) and WiMAX (802.16)	PSNR, End-to-End delay, Goodput, Streaming rate, Number of frames lost, Inter-packet delay, Bandwidth utilization, Loss rate	Live
Freris at al. (FRERIS *et al.*, 2013)	Not used	Users' fairness	H.264/SVC	Not defined	NS-2, Matlab for subroutins, Client interfaces: Ethernet, WiFi (802.11b) and WiFi (802.11g)	PSNR, Streaming rate, Packet delivery delay, Delivery ratio, Run time, Cost functions evaluation (service differentiation)	VoD
Correia at al. (CORREIA *et al.*, 2012)	Not used	N	H.264 /AVC	Picture-Copy Method	Not defined	PSNR	Not defined
MPLOT (SHARMA *et al.*, 2008)	ECN	Y	Not defined	Not defined	NS-2	Bandwidth utilization, Congestion window size (fairness test), Goodput, Effect of loss correlations	Not defined
MP-DCCP (HUANG *et al.*, 2012)	ECN	N	H.264 /AVC	Not defined	NS-2, Disjoint Paths, Client interfaces: WiFi, 3G and Ethernet	Decodable ratio of transmitted frames	Live
ADMIT (WU *et al.*, 2016b)	ZigZag	Y	H.264/AVC	JM	EXata, Client Interfaces: WiFi, Cellular and WiMAX	PSNR, End-to-End delay, Goodput, Congestion window size (fairness test), Inter-packet delay, FEC redundancy, Out-of-order packets	Live

Works	Packet loss Differentiation Method	Fairness	Video Compression	Error Concealment	Experimental Environment	Performance Metrics	Video Services
MPTCP-SD (DIOP *et al.*, 2012)	Not used	Y	H.264/AVC	Not defined	NS-2, Disjoint paths, Client Interfaces: 3G and 3G	PSNR	Real-time (interactive)
MPTCP-PR (DIOP *et al.*, 2012)	Not used	Y	H.264/AVC	Not defined	NS-2, Disjoint paths, Client Interfaces: 3G and 3G	PSNR	Real-time (interactive)
PR-MPTCP$^+$ (CAO *et al.*, 2016)	Not used	Y	Not defined	Not defined	NS-3, Disjoint paths, Client interfaces: WiFi and LTE	PSNR, VQM, SSIM, Number of frames received or dropped	Real-time
SRMT (SILVA *et al.*, 2016)	Not used	N	H.264/AVC	Not defined	Simulator not defined, Client interfaces: WiFi (802.11g), 3G Or WiFi, ADSL	PSNR, SSIM, Goodput, Delay distribution	Live, VoD
PR-SCTP (SANSON *et al.*, 2010)	Not used	N	H.264/AVC	Not defined	Realistic testbed, FreeBSD framework, Netem	Successful frame transmission ratio, Frame late index	Real-time
CMT-QA (XU *et al.*, 2013)	ORP	N	H.264/AVC	Not defined	NS-2, Disjoint paths, Client interfaces: 3G, WiMAX (802.16) and WiFi (802.11)	PSNR, VQM, SSIM, Number of frames lost, Out-of-order packets, Average retransmission, Average throughput	Real-time
CMT-DA (WU *et al.*, 2015)	ECN	N	H.264/SVC	JSVM	EXata, Client interfaces: Cellular, WiFi and WiMAX	PSNR, Inter-packet delay, Goodput, Loss rate, Out-of-order packets	Real-time
CMT-CA (WU *et al.*, 2016d)	ZigZag	Y	H.264/AVC	FFmpeg	EXata, Client interfaces: Cellular, WiFi and WiMAX	PSNR, End-to-End delay, CDF of inter-packet delay, Out-of-order packets, Goodput, Number of frames (I,P) lost	Real-time, live
Yap at al. (YAP *et al.*, 2012)	Not used	N	Not defined	Not defined	Realistic testbed, Android and Ubuntu framework, Real access networks, Up to 10 client interfaces composed of: 3G (HSPA, CDMA), WiMAX and WiFi (802.11a/g)	Throughput, Goodput, CPU load, Power consumption, RTT	Not defined
MARS (SUN *et al.*, 2016)	Not used	N	Not defined	Not defined	Own JAVA socket implementation, Four client interfaces composed of: WiFi and LTE	Out-of-order packets, Reordering delay, End-to-End delay, Throughput	Real-time
BAG (CHEBROLU et al., 2006)	Not used	N	H.263	Not defined	Realistic testbed, Up to five client interfaces composed of 3G	Delay distribution, Lost frame ratio, Required Bandwidth, Video disruption (glitch statistics)	Real-time (interactive)
Corbillon et al. (CORBILLON et al., 2016)	Not used	Y	HEVC	FFmpeg	Own C++ implementation, Disjoint paths, Client interfaces: 3G and WiFi	PSNR, MS-SSIM, Received frame ratio, Received tile ratio	Live, VoD
Ojanperä et al. (OJANPERÄ et al., 2016)	Not used	Y	H.264/AVC	FFmpeg	Realistic testbed, Ubuntu framework, Client interfaces: WiFi (802.11g) and WiFi (802.11a)	Throughput, Quality switch,	Not defined
GALTON (WU *et al.*, 2015)	ZigZag	N	H.264/SVC	JSVM	EXata, Client interfaces: WiFi, WiMAX, Cellular (HSDPA) or multiple wired interfaces	PSNR, Goodput, End-to-End delay, Loss rate	Real-time
FRA-JSCC (WU *et al.*, 2013)	Not used	N	H.264/SVC	JSVM	EXata, Client interfaces: WiFi (802.11b), WiMAX and Cellular	PSNR, End-to-end delay, Loss rate, Available bandwidth	Real-time
MP-DASH (HAN *et al.*, 2016)	Not used	N	H.264/AVC	Not defined	Realistic testbed, Ubuntu framework, Real access networks, Client interfaces: WiFi and Cellular	Throughput, Energy consumption, Download time, Average 3G traffic	Not defined
Nam et al. (NAM *et al.*, 2016)	Not used	Y	H.264/AVC	Not defined	Realistic testbed, Ubuntu framework, Real MPEG-DASH platform, Mininet over WiFi for SDN, Real access networks, Client interfaces: WiFi (802.11g) and WiFi (802.11a)	Played bit rate, Rebuffering, Out-of-order packets	Real-time
CMT-CL/FD (XU *et al.*, 2015)	RTX	Y	Not defined	Not defined	NS-2, Disjoint paths, Server and client interfaces: 3G (WCDMA), WiMAX (802.16) and WiFi (802.11)	PSNR, Video buffer underflow, Throughput, Fairness test	Real-time

over that path. A path is categorized as a mildly congested one if feedback reports show both transmission losses and discards either in a single or consecutive reports. If this behavior occurs in more than three consecutive reports, it means that the path is congested. CMT-QA (XU *et al.*, 2013) handles the packet loss differentiation by proposing optimal retransmission policy (ORP). In ORP, when a loss occurs, $(RTT/cwnd)$ is calculated, and the result would be compared with a threshold. This threshold is defined as path quality. Therefore, if $(RTT/cwnd)$ is more than the threshold, the loss is due to wireless loss. Otherwise, it is a congestion loss. If losses occur more than once and consecutively, then congestion is the reason. CMT-CL/FD (XU *et al.*, 2015) proposed loss-cause dependent retransmission (RTX) policy. In RTX, two cases are considered; 1) When the loss is detected by fast retransmission. Thus, the residual capacity of the path is calculated. If it is a positive value, it means that the path is underused and wireless loss is occurred. Otherwise, if the residual path value is negative, congestion is the reason. 2) When the loss is detected by expiring RTO. In this case, the path is failed or severe congestion is occurred. CMT-DA (WU *et al.*, 2015), MPLOT (SHARMA *et al.*, 2008) and (CORREIA *et al.*, 2012) utilize Explicit Congestion Notification (ECN) to distinguish loss differentiation. ECN is defined by IETF (RAMAKRISHNAN *et al.*, 2001) in 2001. ECN-aware routers informs congestion by setting a mark in the IP header, without dropping any packet. BEMA (WU *et al.*, 2016a), ADMIT (WU *et al.*, 2016b), GALTON (WU *et al.*, 2015) and CMT-CA (WU *et al.*, 2016d) use ZigZag scheme, which is introduced in (CEN *et al.*, 2003). ZigZag classifies losses as wireless based on the number of losses and on the difference between relative one-way trip times and the mean of relative one-way trip times. For further information about the effect of different types of losses like random loss or bursty loss on video streaming quality refer to (APOSTOLOPOULOS, 2000).

2.5.2 Fairness

Table 9 summarizes the research works that consider fairness, which was previously introduced in Section 2.3. Works address fairness in terms of consumed resources by the proposed congestion algorithms (e.g., MPRTP (SINGH *et al.*, 2013), MPLOT (SHARMA *et al.*, 2008), CMT-CL/FD (XU *et al.*, 2015)), or as adopted by TFRC (e.g., BEMA (WU *et al.*, 2016a), CMT-CA (WU *et al.*, 2016d)) or in terms of MPTCP coupled congestion control (e.g., ADMIT (WU *et al.*, 2016b), MPTCP-SD/PR (DIOP *et al.*, 2012), PR-MPTCP^{+} (CAO *et al.*, 2016), Corbillon et al. (CORBILLON *et al.*, 2016), Ojanperä et al. (OJANPERÄ; VEHKAPERÄ, 2016)). Besides, in our research works, Freris at al. (FRERIS *et al.*, 2013) consider user fairness of network resources.

2.5.3 Video Compression and Error Concealment

Several video codecs were used in the research works cited in Table 9, such as H.263 (RIJKSE, 1996), H.264/AVC (ITU-T, 2003), H.264/SVC (SCHWARZ *et al.*, 2007), HEVC (ITU-T, 2013), and SHVC (BOYCE *et al.*, 2014). After video transmission, if protection methods are not able to recover the lost packets, the decoder itself can employ error concealment. This way, decoder exploits correlations in the previously received video sequence to conceal the lost information. JM, for instance, performs frame copy while FFmpeg performs temporal interpolation. According to (CHANG *et al.*, 2012), in case of whole-frame losses, when isolated B frames were lost and concealed by either JM or FFmpeg, about 40% of the losses were not even noticed by observers. Our research works used JM ((JVT), 2017a), x264 (X264, 2017), JSVM ((JVT), 2017b), FFmpeg (FFMPEG, 2017), HM ((JCT-VC), 2017) for error concealment.

2.5.4 Experimental Environment

Table 9 shows that experimental evaluation is mostly dominated by network simulators, such as OPNET (OPNET, 2017), NS-2 (NS2, 2017), NS-3 (NS3, 2017), EXata (EXATA, 2017), NetEm (NETEM, 2017; HEMMINGER, 2005). Only few works, mainly due to costs, scale, and scope, carried their evaluation on real testbeds. Wireless-enabled network emulators like Mininet-WiFi (FONTES *et al.*, 2015) are also another category of experimental environments. We also cover some additional implementation details. For example, which type of network interfaces are used in experiments, or if the simulation uses disjoint paths (no common link or node). Using disjoint paths improves bandwidth aggregation and has the benefit of additional fault-tolerance compared with non-disjoint paths (SINGH *et al.*, 2015), altogether contributing to the users video experience.

2.5.5 Performance Metrics

Several performance metrics were used in the research works cited in Table 9. Most of them are explained in Section 2.3.2. We have added some additional video quality metrics, such as Peak Signal-to-Noise Ratio (PSNR), Video Quality Metric (VQM), Structural SIMilarity (SSIM) (WANG *et al.*, 2004), MultiScale Structural SIMilarity (MS-SSIM) (WANG *et al.*, 2003), and Subjective Assessment Methodology for VIdeo Quality (SAMVIQ) (BLIN, 2006).

2.5.6 Video Services

The last column of Table 9 presents for each of research works which type of video service was considered by the authors, such as VoD, live, and real-time as an upper set including interactive video streaming applications. As discussed in Sections 1.2 and 2.3, each type of video services has different QoS requirements such as delay-sensitivity.

2.6 Open Research Issues

Many research avenues around multipath wireless video streaming are open. In the following, we overview some relevant evolving aspects and present future work opportunities.

Standardization developments. MMT is a recent standard protocol with potential abilities discussed in the survey. Future work could evaluate the performance of MMT over MPTCP or QUIC utilizing multipath scheduling methods defined in these protocols for video streaming over heterogeneous networks. HTTP/2 provides noticeable features such as the ability to push content in advance, and frame multiplexing. Therefore, further attention on multipath delivery over HTTP/2 shall be pursuit (FRÖMMGEN *et al.*, 2017). Another standards related topic would be the use of HEVC, especially SHVC, which do not seem to be widespread in the networking literature despite being widespread in the video coding community.

Network Softwarization. Attempts to integrate SDN with multipath video streaming (Section 2.3.3) promise effectiveness for path-aware strategies due to its ability to programmatically define the End-to-End network behaviour. While OpenFlow is considered the mostly accepted interface between control and data planes (KREUTZ *et al.*, 2015), alternative means for southbound interaction of controllers and datapath devices (e.g. P4 programmable data planes), including SDN protocol extensions relevant for wireless communications (e.g., (Yan *et al.*, 2015; LEE *et al.*, 2014)) deserve further research efforts. SDN and NFV as enabling technologies of multi-domain network service orchestration (SOUSA *et al.*, 2019) will certainly keep attracting research attention and will play a critical role in the realization of multipath strategies for video streaming and other types of services.

5G. Fifth generation (5G) cellular wireless roll-outs will become reality over the next years. 5G aims to introduce new services that require extreme bandwidth and ultra-low latency (DANDACHI, 2017). One concept presented in 5G to reach this goal is the presence of multihoming capability. There is some research on 5G multihoming open challenges and multihoming services (IBNALFAKIH *et al.*, 2016; KARIMI *et al.*, 2017). Furthermore, there are several study and efforts for MPTCP operation in 5G (PURKAYASTHA *et al.*, 2017; KARIMI *et al.*, 2017). In addition, studies show that emerging technologies such as SDN to MPTCP in 5G networks could improve the transmission performance due to SDN capability to control the subflows by monitoring network condition (LEI *et al.*, 2018; BARAKABITZE *et al.*, 2018b). Therefore, video streaming over 5G networks is an important emerging research area where innovative solutions will be required considering multihoming solutions along SDN/NFV-based technologies.

WiFi Evolution. There are also big advancement in wireless technologies from WiFi communication to increase the wireless networking performance such as 802.11ad and 802.11ay for 60 GHz, or 802.11ax for 2.4 Ghz and 5 Ghz concurrently, in the near future (GHASEMPOUR *et al.*, 2017; HUAWEI, 2019) aim to achieve great throughput and ultra-low latency. Therefore, there is room to evaluate whether these new WiFi technologies can support video streaming alone or not? Another open question here is to explore the impact of these new WiFi technologies on multipath video streaming.

Energy considerations. Power efficiency is an essential requirement. The work (LI *et al.*, 2012) shows high power consumption by LTE when video streams over HTTP. Energy consumption even increases more by using multiple network interfaces. Therefore, optimizing power consumption needs further attention in the proposed approaches.

Security. Multipath delivery could mitigate some security threats inherently through the use of alternative paths throughout the network. There is little work in the scope of multipath multimedia streaming security. At the same time, Digital Rights Management (DRM) and the license issues are also security related issues critical for some video services.

Mobility and Internet of Vehicles (IoV). Although terminal mobility, velocity, motion degree and related mobile aspects are factors affecting video quality, they are rarely discussed in the literature. This type of considerations are be key in the delivery of wireless video in mobile environments in scope of Intelligent Transport Systems (ITS) (Kaul *et al.*, 2017) and Vehicle-to-everything (V2X) communications (FONTES *et al.*, 2017).

Machine Learning and Artificial Intelligence. Leveraging artificial intelligence and machine learning methods are increasingly becoming key tools for network and service optimization (LATAH; TOKER, 2018) and can be used for advanced scheduling and adaptive coding decisions (MAO *et al.*, 2017b). The importance of machine learning approaches to improve video quality has been recognized by Netflix which proposed a new video quality assessment method named Video Multimethod Assessment Fusion (VMAF). VMAF is a machine learning-based model that is trained and tested using the results of a subjective experiment in order to deliver the best video quality to the user (TRESTIAN *et al.*, 2018). Besides, there are also several machine learning-based efforts to learn QoS measurements (AL-JAWAD *et al.*, 2018) or QoE from user reactions (VASILEV *et al.*, 2018; ALZAHRANI *et al.*, 2018) to solve various optimization and control problems for a single path video streaming. In our state-of-art, there are some approaches utilizing machine learning systems learning QoS from user device and using it for multipath scheduling decisions (XING *et al.*, 2012; XING; CAI, 2014). Thus, similarly, an interesting solution could be utilizing machine learning systems learning QoE from user reactions and using it for multipath scheduling decisions.

2.7 Conclusion

Demand for live and on-demand video delivery have dramatically increased along always higher user expectation on QoE. Mobile environments increase the challenges when a client is in movement and requires a seamless connection while throughput varies, and unpredictable latencies and failure exist.

One promising approach in order to improve QoE for wireless video streaming is multipath delivery, which increases available bandwidth, resilience and load balancing. From the industry perspective, several companies have implemented their own multipath approaches, such as AVAYA (AVAYA, 2019) and Cisco (EtherChannels, 2019). Apple and SAMSUNG have also started to support multipath on smartphones (CHESHIRE, 2019; GALAXYS5, 2017) for different services like voice recognition, or to increase the download speed of specific software packages. Therefore, we expect a growth in multipath video streaming in the near future. However, there are still many issues to be solved, especially for solutions which are not compatible witch each other or that require changes in servers and/or clients, or network equipment to support it (CHESHIRE, 2019).

In this work, we have provided an in-depth survey of multipath wireless video streaming proposals, covering over forty relevant pieces of work. We have categorized and explained the research works based on the layer in the protocol stack and the original protocol/feature dominating in each work. Network equipment compatibility has be also discussed. In addition, scheduling, resilience and path selection techniques are presented. Finally, we have studied different key related methods, such as packet loss differentiation, video compression, error concealment, etc.

To conclude with, we highlight some points and observations resulting from the literature survey. Several challenges exist when designing a multipath video streaming approach, which are explained in Section 1.2. We observe that in order to overcome these challenges, packet scheduling strategy should consider several factors. The first one is the layer dependency that is discussed in Section 2.3, and the research works are summarized and categorized based on it in Table 3. Research shows that the scheduler has a better decision when it has complete and accurate information about video contents, packet delivery deadlines, playback buffer, RTT, available bandwidth, and other network information. Implementing scheduling functions on a specific layer could access only a part of this information. Therefore, cross layer approaches get high attention due to their ability of gathering information of different layers for better scheduling decision.

Another important factor to design a scheduler is client and network equipment compatibility. This topic is also discussed in Section 2.3 and summarized in Table 3. While the most flexible case to implement is when only client modification is required, some approaches require changing the server, or both server and client, or also the network infrastructure. It is also important to note the ability to traverse middleboxes.

Fundamental aspects to be considered to improve the performance of scheduling functions discussed in Section 2.4 and summarized in Tables 5, 6, 7, 8 include which packet should be sent next, through which path, and with which type of error protection. An adequate scheduling strategy should be content-aware, and path-aware, as well as it should utilize a proper channel or source level packet protection method. Such a scheduling approach improves QoE, bandwidth aggregation, load balancing, and mitigates HOL blocking and out-of-order video packet deliveries.

Section 2.5 and Table 9 show some related methods used in the research works and the key performance indicators to evaluate the approaches. One observation is that while calculating video quality metrics is very useful to understand the performance of each approach, many of the works only consider network QoS metrics without assessing video performance in terms of QoE as the key performance indicators from an end-user perspective.

The path ahead towards the broad realization of multipath wireless of video streaming solutions is not without issues. In Section 2.6, we overview a series of open challenges and point to some research opportunities.

3 Advancing MMT for Multipath Wireless Video Streaming

This chapter introduces the multipath improved MMT to reduce the negative effects of network congestion and wireless lossy network condition on the perceived video quality for wireless video streaming. In this regard, the multipath MMT system is presented regarding video parsing, statistic computation, feedback mechanism, periodic path probing and mainly explanation of how the proposed CAPA scheduling strategy works in Section 3.1. Finally, Section 3.2 concludes remarks in this work.

3.1 Multipath MMT System

A diagram overview of the proposed multipath MMT system considered in this thesis is presented in Figure 20. The goal of this system is to achieve high quality video streaming solution for the MMT protocol by adopting scheduling strategies considering path conditions and video content features. This system is completely defined at application layer. We define a unicast video transmission system considering the multipath data transmission of a single video flow over multiple access networks. Each network is modeled as an independent End-to-End communication path and UDP is employed to transmit video data. The modules proposed for this MMT system are implemented at both sender and receiver sides.

At the sender side, there are different modules, namely parser, feedback RX, scheduler, and packet TX. Firstly, each video packet is handled by the parser module to extract the frame type of video content (described in Subsection 3.1.1). This information would be informed to the scheduler module and later would be used for path decision by it. Additionally, feedback RX module receives feedback packets periodically from the receiver. This module discards the overdue received feedback packets and informs the updated paths' information to the scheduler. In sequence, the scheduler module evaluates the path metrics and uses them together with the frame type information to assign a proper path for each video packet based on its own strategy (described in Subsection 3.1.5). Finally, packet TX module transmits the video packet through the decided path.

At the receiver side, existing modules include packet RX, statistic, feedback TX, reassembler, decoder, renderer. After receiving packets by the packet RX module, statistics (such as goodput, average delay, number of lost packets, and jitter) should be generated via the statistics module periodically (after each predefined time interval) in order to monitor the quality of the transmission paths (described in Subsection 3.1.2). This

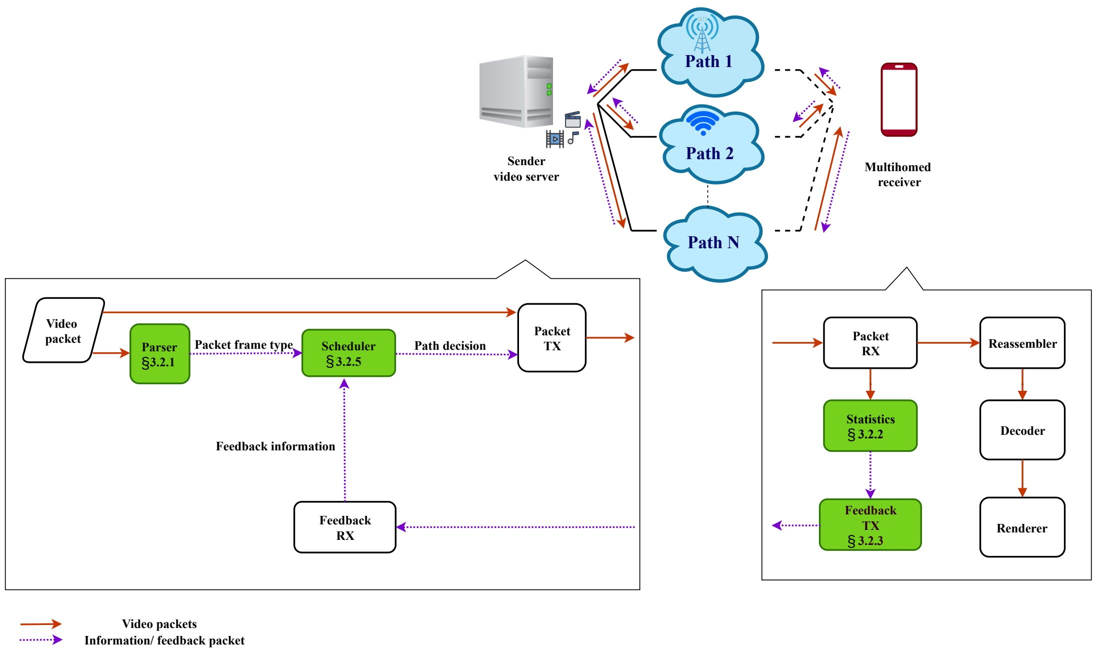


Figure 20 – Diagram overview of the proposed multipath MMT system.

information should be sent to the sender in a feedback packet by assembling the following two signaling messages defined in MMT standard (MPEG, 2014): Reception Quality Feedback (RQF) and Network Abstraction for Media Feedback (NAMF) (described in Section 3.1.3). Since feedback information is very important and has a high effect on the sender scheduler decision, it is necessary to send it over the best path (instead of using the same receiving path). Thus, there is a scheduler implementation at the receiver side as well. Therefore, feedback packets could be delivered to the sender fast and with a low probability of losing.

If the received packets by the packet RX module are not overdue, then they go to the reassembler module to reassemble the encoded video bitstream, and consequently, to the decoder module for error concealment and decoding. Finally, the decoded video is ready to display by the video renderer module.

3.1.1 Video Parsing

As previously explained in Section 2.4.1.1, understanding the frame type of each video packet content and considering this information in scheduling strategy helps to choose the frame packets with higher priority to transmit through more qualified paths or to protect them during transmission. In our multipath MMT system, the parser module is responsible for extracting the frame type information of each MMT video packet content by accessing the ISOBMFF structure of the fragmented video and parsing the first bits of each fragment. The ISOBMFF structure of a fragmented video sequence is shown in Figure 21. As it is shown in this figure, the MPU metadata consists of ‘ftyp’, ‘mmpu’, and 'moov' boxes. It can also contain any other boxes that are applied to the whole MPU. Fragment metadata consists of 'moof' box and 'mdat' box header. The video data is then split into multiple data units of MFU in 'mdat' box. Therefore, the parser module, implemented in this work, looks for the 'moof' box and with this information, it reaches to the content of fragment, which is stored in the 'mdat' box, and then it can access to the frame type information available there. This information is sent to the scheduler to be used for path decision.

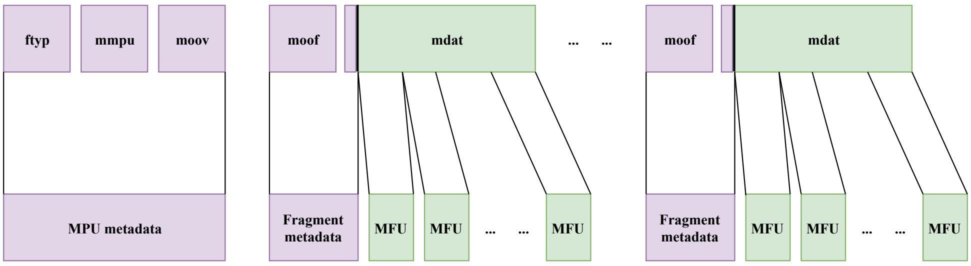


Figure 21 – Structure of ISOBMFF file for a fragmented video sequence (source: adapted from (MPEG, 2014)).

3.1.2 Statistic Collection

The statistics are computed periodically for each path, and they are sent to the sender via feedback packets to make the sender aware of all paths' conditions. Our multipath MMT system employs delay, jitter, packet loss, and goodput statistics.

- **Packet delay:** upon receiving the packet, the receiver computes the difference between the received timestamp and the sent timestamp as marked in the packet to compute the delay of the packet, called one-way delay. Note that, sender and receiver have access to a common global clock reference (e.g., via Network Time Protocol (NTP) or other clock synchronization technologies) (YUSTE *et al.*, 2015). The delay of the received packets in each feedback interval time is then used to compute the average packet delay in that interval time.
- **Packet jitter:** our MMT system computes the jitter as the difference between the current average packet delay and the moving average packet delay. The moving average packet delay uses the average packet delays of the last jitter window size number of values. The value configured for this parameter is 16.
- **Packet loss:** in this work, each MMT packet contains two sequence numbers: the subflow sequence number, which is used to detect packet losses over each path, and flow sequence number to detect total losses and reconstruct the original data sequence at the receiver. Even if we didn't check middleboxes in this work, but based on state-of-the-art, the consecutive subflow sequence number could assist packets

to traverse middleboxes, such as firewalls, because using regular sequence number could lead to problems such as packets dropping due to gaps in sequence numbers or due to out of order packets (BARRÉ *et al.*, 2011).

To detect packet losses occurred over each path, the receiver keeps track of the received packets by monitoring the subflow sequence number field in the packets. When a packet with expected subflow sequence number does not arrive at the receiver, the receiver adds an entry in the delayed or missing packet list that it maintains for each subflow. To categorize a packet as a lost one, the receiver waits for a certain time and starts a timer. If the packet with the expected subflow sequence number does not arrive at the receiver by the time and the timer expires, the receiver categorizes that packet as a lost one. Then, the receiver updates the packet loss statistics for the feedback time interval and deletes the entry from the missing packet list. We also identify overdue packets and included them as lost packets.

To compute total packet losses, the flow sequence number is considered. Therefore, if a packet is sent through both paths (duplicated) and it would be lost over both of them, then, it is only counted once as total losses.

- **Goodput:** in this work, we compute goodput in Mbps (Mega bits per second) for each path. The formula below defines the goodput computation of each path (gp_p) as the division of the number of delivered bits by Δt.

$$gp_p = \frac{\#received\ bits}{\Delta t} \tag{3.1}$$

Figure 22 illustrates how Δt is computed. For instance, in the WiFi path (shown in blue in Figure 22), Δt is the time difference between the time the first packet in the feedback time interval was sent (packet 11) and the time the last packet in the feedback time interval was delivered (packet 15). Similarly, in the LTE path (shown in orange in Figure 22), Δt is the time difference between the time the first packet in the feedback time interval was sent (packet 12) and the time the last packet in the feedback time interval was delivered (packet 16).

3.1.3 Feedback Mechanism

Upon computing the statistics described in Section 3.1.2, the receiver sends statistic information as feedback to the sender. The periodicity of the feedback is defined by the feedback interval time. Setting interval time, which defines feedback frequency, has a trade-off between accuracy and overhead. For example, with higher feedback frequency, the accuracy can be improved but there is more overhead since it needs to send more feedback packets. Conversely, less feedback frequency indicates lower accuracy, despite of

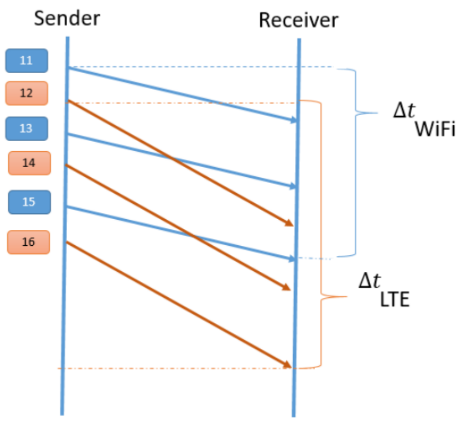


Figure 22 – Time differences for goodput computation.

lower overhead. Moreover, the feedback frequency should be according to the video transmission bit rate. While a video transmission has has higher bit rate, feedback frequency should be higher than those with low video transmission bit rate. For our feedback mechanism we opted to send feedback messages to the sender every 0.5 seconds (WU *et al.*, 2016a) due to it is small enough to react the network changes but long enough to not to cause too much overhead.

Other possibility to define feedback time interval, rather than constant interval time employed in this work, is to configure an adaptive time interval for it which could be according to the network conditions. For example, if the receiver notices that the transmission path is in congestion, by monitoring path condition, it can increase feedback frequency (SINGH *et al.*, 2013). Another adaptive solution is sending feedback after each frame (MAO *et al.*, 2006). In this case, there is a very high frequency of feedback. Therefore, it is useful for very high dynamic networks, such as ad hoc networks where the sender can quickly adapt to transmission errors. For example, the encoder changes its encoding mode or parameters, producing variable redundancy for error resilience, or assigning variable traffic paths based on the network condition.

The standard MMT protocol defines a signaling mechanism to deliver feedback information to the sender. While several signaling messages are defined in the MMT protocol, two of them were specially identified by us to fulfill the need of delivering back computed statistics to the sender: RQF and NAM, which are described in Figure 23. The feedback message is composed of two sets of NAM and RQF data for each path and is sent as a MMT signal. The fields that met our needs for the current feedback mechanism are listed below:

- RQF
  - packet loss ratio (to carry the packet loss)
  - inter arrival jitter (to carry the jitter)

- propagation delay (to carry the one-way delay)

- NAM
  - available bit rate (to carry the goodput)
  - current delay (to carry the highest one-way packet delivered delay in each feedback time interval)

RQF message fields

Field	Number of Bits
message ID	16
version	8
length	16
measurement duration	16
packet loss ratio	8
inter arrival jitter	32
propagation delay	32
feedback timestamp	32

NAM message fields

Field	Number of Bits
message ID	16
version	8
length	16
CL ID	8
available bit rate	32
buffer fullness	32
peak bit rate	32
current delay	32
SDU size	32
SDU loss ratio	32
generation time	32
BER	32

Figure 23 – RQF and NAM message fields.

3.1.4 Periodic Path Probing

In order to check previously congested paths and identify changes in stagnant path conditions, we have implemented periodic path probing by sending a video packet - called probe packet - every predefined time interval (0.2 second in our work). After receiving feedback information containing metrics computed by the receiver due to the probe packet, the sender is aware of the current condition of the previously congested path. If the condition improved, the path can be considered again by the sender as a possible choice for sending subsequent packets.

3.1.5 Content-Aware and Path-Aware Scheduling Proposal

For selecting a path to send packets, the sender follows the CAPA strategy defined by its scheduler module to allocate the data through the multiple transmission paths. In this regard, the scheduler monitors the current conditions of each path and verifies the content of each MMT video packet to select the best path for each packet to be transmitted. Besides, the scheduler also considers heterogeneous network context to properly allocate bit rate to each path in order to efficiently utilize path capacities, cope with network conditions such as congestion or lossy channels, and execute load balancing.

An overview of the proposed scheduler module architecture is presented in Figure 24. This proposed scheduler module is composed of four sub-modules: traffic splitter, path estimator, discarder and content-aware path assignment arbiter.

Traffic splitter considers heterogeneous network context to properly split the video traffic, copes with each path capacity and its current conditions in order to avoid either congestion or underutilization of network resources and to perform load balancing. Therefore, for each feedback information received as input, traffic splitter applies the adaptive traffic split scheme proposed in Subsection 3.1.5.1 to assign a packet rate to each path.

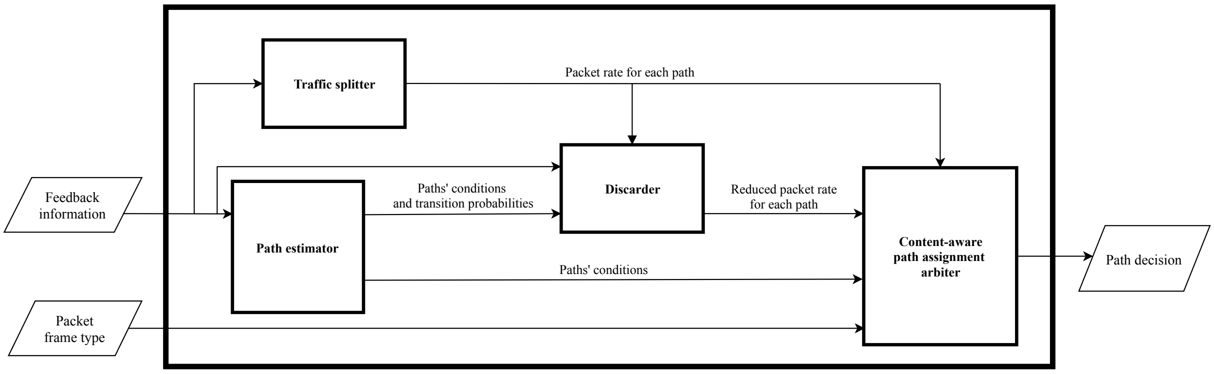


Figure 24 – Overview of the proposed scheduler module architecture.

Besides, path estimator also uses the feedback information to adaptively categorize each path condition as good, mild/lossy or bad by following the method detailed in Subsection 3.1.5.2. Each path could be categorized as good, mild or bad considering path's congestion situation. The path condition is estimated as lossy when the channel is a lossy channel with burst wireless losses due to wireless errors (e.g., noise or interface). Therefore, if the path condition is estimated as mild (congestion)/(burst wireless) lossy or bad, the possibility of losing video packets is high, either because they will be dropped by the network itself or because they will be considered as overdue packets when arriving at the receiver side. In case of congestion network condition, discarder applies a discard strategy, described in Subsection 3.1.5.3 on the input data, using the estimated packet rate, feedback information, paths' conditions and transition probabilities to find the packet discard rate for each path. This strategy helps reducing congestion by avoiding sending packets that would probably be lost.

The content-aware path assignment arbiter receives all the calculated information by aforementioned modules, the packet rate, the packet discard rate, and paths' conditions, then it applies the defined content-aware strategy to decide a path for each packet transmission. This way, it considers the content of each video packet to better protect I frames and the closest n P frames, named as near-I (NI) frames in this work. The reason for protecting NI frames is based on the fact that errors on these first P frames of the GoP have a higher impact on the perceived quality of experience (HUSZÁK; IMRE, 2010). Our proposed content-aware strategy follows the rules described in Subsection 3.1.5.4. The mathematical notations used throughout this thesis work are summarized in Table 10.

Table 10 – Definition of parameters.

Symbol	**Definition**
N	total number of transmission paths
bw_p	maximum bandwidth of path p [kbps]
$delay_p$	minimum delay of path p [ms]
gp_p	goodput of path p [Kbps]
d_p	average one-way delay of path p [ms]
GDD_p	GDD (goodput-division-delay) of path p
λ_p	the packet rate split factor for path p
P	a matrix of transition probabilities among the states
p_{ij}	transition probability from each state (i) to state (j)
C	a weighted matrix to store c_{ij}
c_{ij}	the number of transitions from each state (i) to state (j)
$d_{p,wma,cur}$	current weighted moving average of one-way delay of path p
$d_{p,wma,pre}$	previous weighted moving average of one-way delay of path p
$\sigma_{d_{p,cur}}$	current standard deviation of one-way delay of path p
$\sigma_{d_{p,pre}}$	previous standard deviation of one-way delay of path p
T_l	packet loss rate threshold [%]
T_d	one-way delay threshold [ms]
D_p	highest one-way delay of path p [ms]
L_p	packet loss rate of path p

3.1.5.1 Adaptive Video Traffic Split

The flowchart in Figure 25 depicts the process of the proposed adaptive video traffic split strategy. We propose to split the video traffic based on a goodput-division-delay (GDD) metric dynamically computed by the scheduler after receiving a feedback packet. GDD for each path is calculated as $GDD_p = \frac{gp_p}{d_p}$, where gp_p is the goodput of path p measured in [kbps] and d_p is the average one-way delay of path p measured in [s]. The packet rate split factor for each path (λ_p) is then computed as

$$\lambda_p = \frac{GDD_p}{\sum_{i=1}^{N} GDD_p}, \tag{3.2}$$

where N is the total number of transmission paths. As an initial estimation, when feedback packets were not yet received by the scheduler, λ_p is computed as

$$\lambda_p = \frac{\frac{bw_p}{delay_p}}{\sum_{i=1}^{N} \frac{bw_p}{delay_p}}, \tag{3.3}$$

where bw_p is the inherent maximum path bandwidth and $delay_p$ is the minimum path delay. Therefore, higher bit rate is assigned to the path with higher goodput and lower one-way delay. Another solution considering the resource availability is defined in (WU *et al.*, 2016b).

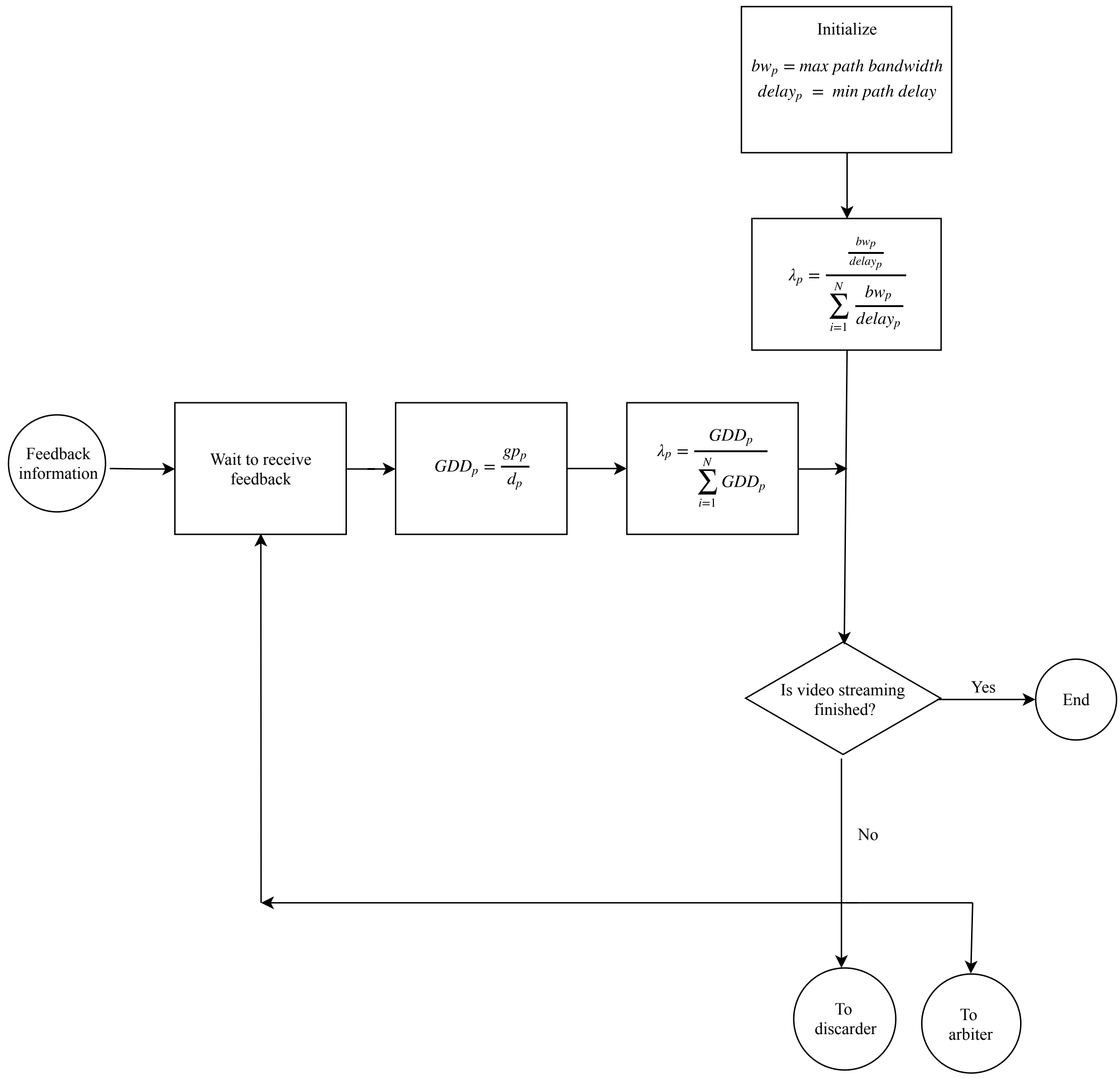


Figure 25 – Flowchart of adaptive video traffic split strategy.

3.1.5.2 Estimation of Path Condition

The flowchart in Figure 27 depicts the process of the proposed strategy to estimate paths' conditions. We propose to model the path condition estimation problem as a three-state Markov model where each state represents one path condition: Good Condition (**GC**), Mild/Lossy Condition (**MC/Lossy**) and Bad Condition (**BC**) - Figure 26. A matrix P of transition probabilities among the three states is computed and periodically updated by the scheduler. A matrix C is also kept to store the number of transitions from each state i to state j (c_{ij}). Following (XING; CAI, 2014), the elements of matrix P are computed by the following equation:

$$p_{ij} = \frac{c_{ij} + 1}{\sum_{j=1}^{N} c_{ij} + N}, \tag{3.4}$$

where p_{ij} is the transition probability from i to j.

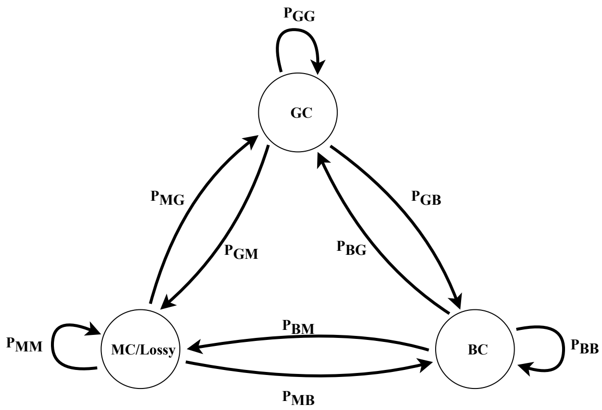


Figure 26 – Three-state Markov model used for estimation of path condition.

In order to define the path condition state, two predefined thresholds are used: T_d for one-way delay and T_l for packet loss rate. T_d was set as 50 milliseconds following recommendations in (WU *et al.*, 2016c), where this is the maximum limit delay for achieving high quality multipath HD video transmission in heterogeneous wireless networks. For the definition of T_l, we were inspired by the work in (CHOW *et al.*, 2009), which specifies a multipath streaming scheme (EMS) with FEC (Forward Error Correction). The work states that, with H.264 encoding, the packets loss rate should be less than 1% in order to ensure high quality real-time live streaming. Since FEC is not applied in this work to MMT packets, this limit was slightly extended and T_l was set as 2%.

The following two metrics specified in (CEN *et al.*, 2003) were also computed and used in this work:

$$d_{p,wma,cur} = \frac{31}{32} \cdot d_{p,wma,pre} + \frac{1}{32} \cdot d_p \tag{3.5}$$

$$\sigma_{d_{p,cur}} = \frac{15}{16} \cdot \sigma_{d_{p,pre}} + \frac{1}{16} \cdot |d_p - d_{p,wma,cur}| \tag{3.6}$$

where $d_{p,wma,cur}$ is the current weighted moving average of one-way delay of path p, $d_{p,wma,pre}$ is the previous weighted moving average of one-way delay of path p, $\sigma_{d_{p,cur}}$ is the current standard deviation of one-way delay of path p and $\sigma_{d_{p,prev}}$ is the previous standard deviation of one-way delay of path p.

The two thresholds (T_d and T_l) and the two computed metrics ($d_{p,wma,cur}$ and $\sigma_{d_{p,cur}}$) are then combined in the following way:

- Path is in **GC** state if $D_p \leq T_d$ && $L_p \leq T_l$, where D_p and L_p are the highest one-way delay of path p and packet loss rate of path p, respectively, for a single feedback interval time;

- Path is in **MC/Lossy** state if $D_p \leq T_d$ && $L_p > T_l$;
- Path is in **BC** state if $D_p > T_d \parallel d_p > d_{p,wma,cur} + \frac{\sigma_{d_{p,cur}}}{2}$.

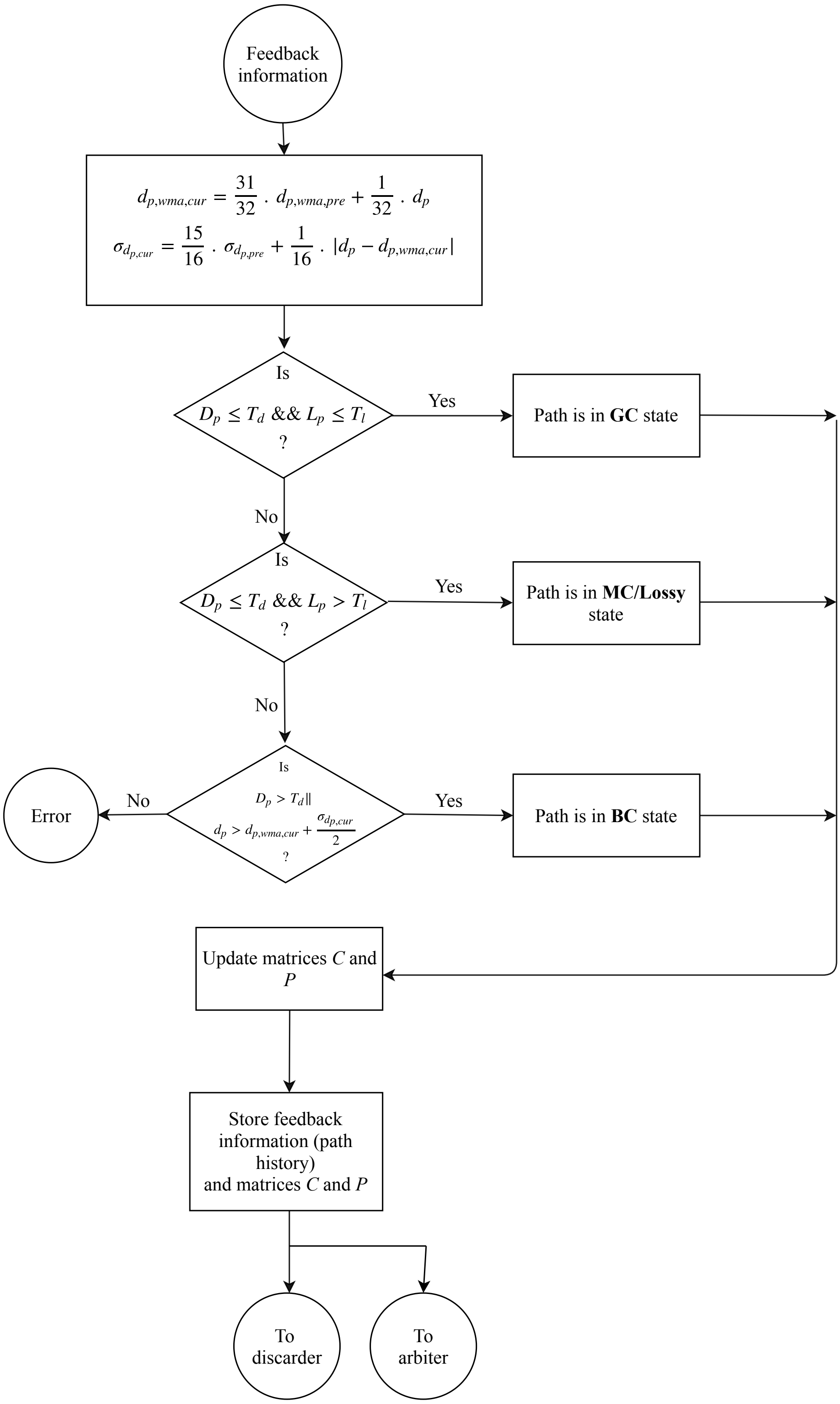


Figure 27 – Flowchart of paths' conditions estimation strategy.

3.1.5.3 Discard Strategy

The flowchart in Figure 28 depicts the process of the discard strategy. Based on this proposed strategy, if the path is congested, a discard strategy is applied by the scheduler to avoid increasing network congestion by not even sending packets that will be probably either overdue or dropped. As part of the strategy, the transition probabilities computed according to Eq. (3.4) are used to reduce the path bit rate split factor λ_p in the following way:

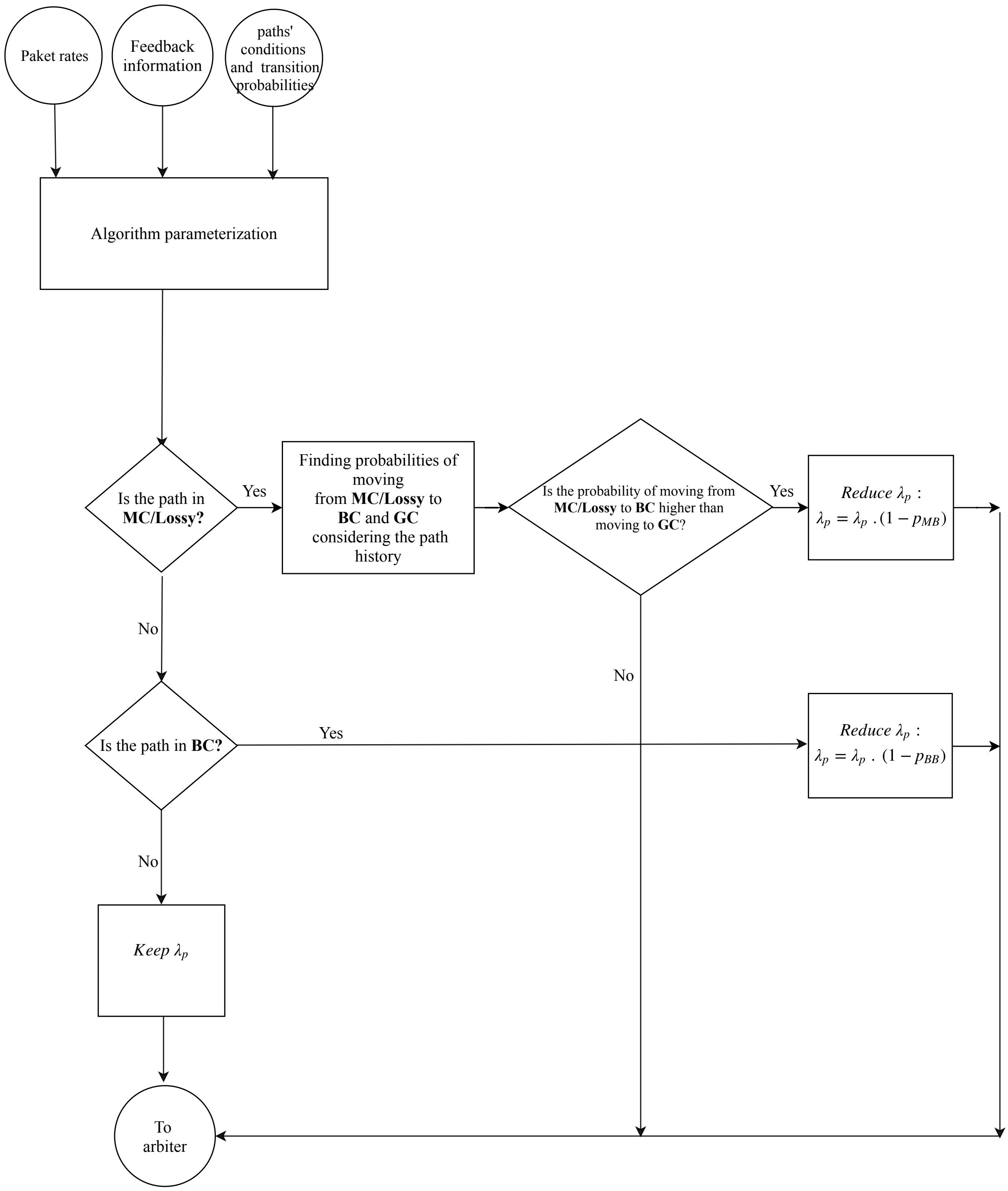


Figure 28 – Flowchart of discard strategy.

- if the path is in **MC/Lossy** it is important to consider the path history in order to verify if there is a higher probability of moving to **GC** or to **BC**. The last computed objective metrics are compared with the metrics received in the previous feedback message. If the number of lost packets, jitter and delay have increased, then the probability of moving to **BC** is higher and λ_p is updated as $\lambda_p = \lambda_p .(1 - p_{MB})$, where p_{MB} is the probability of transition from **MC** to **BC**. Otherwise, it means that the path congestion condition is improving or there is only a lossy channel. Therefore, no packet will be discarded;

- if the path is in **BC**, then $\lambda_p = \lambda_p \, . \, (1 - p_{BB})$, where p_{BB} is the probability of being in **BC** and staying in **BC** state.

Another solution defined in (CORBILLON *et al.*, 2016) is estimating of packet display deadline and discarding ones which have no chance to arrive in time.

3.1.5.4 Adding Content-Aware Protection

The flowchart in Figure 29 depicts the process of content-aware protection strategy. In addition to the path-aware scheduling strategy detailed in Subsections 3.1.5.1, 3.1.5.2 and 3.1.5.3, we also define a content-aware scheme to better protect I and near-I (NI) frame packets in this work. Our content-aware scheme does not discard any I or NI packet and employs packets duplication and/or rerouting according to the following rules specified for the scenario with two transmission paths ($N = 2$):

- if both paths are in **GC**, then no packet will be duplicated and the bit rate split factors computed according to Eq. (3.2) for each path will be used.

- if one path is in **GC** and the other path is in **MC/Lossy** or **BC**, then all I packets will be sent only through the path in **GC**;

- if one path is in **MC/Lossy** and the other path is in **BC** or both paths are either in **BC** or **MC/Lossy**, then all I packets will be duplicated and sent through both paths;

The same rules are applied for NI frame packets, except they are duplicated only when both paths are in **MC/Lossy**. Note that the protection scheme is not applied to P frame packets. Therefore, they are transmitted according to all the rules specified in previous subsections, including the discard strategy.

It must be highlighted that our scheduling strategy, CAPA, does not discard any packet containing I or NI frame data. In other words, the proposed discard strategy applies only to P frame packets.

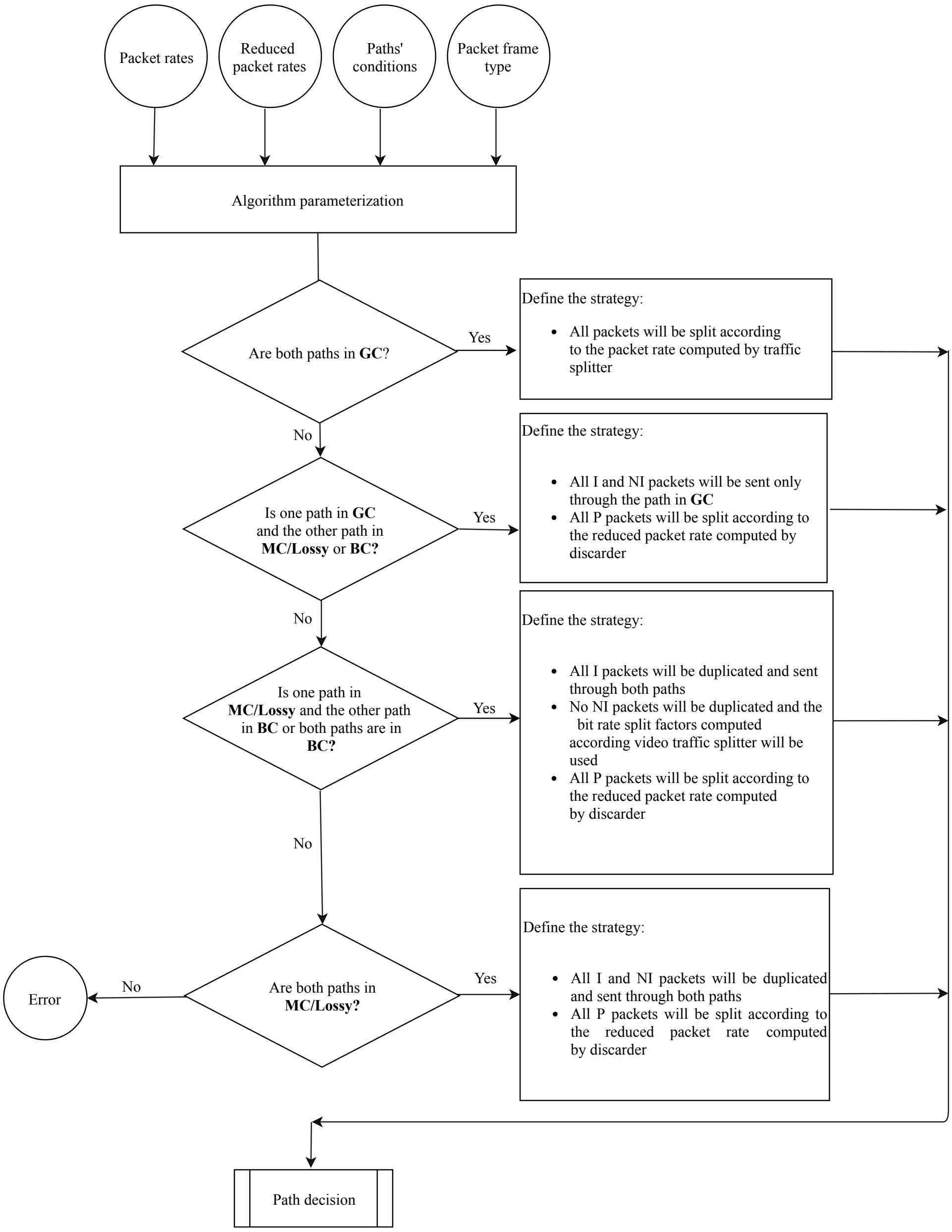


Figure 29 – Flowchart of content-aware strategy.

3.2 Conclusion

In this chapter, we have introduced a novel scheduling strategy as a multipath MMT solution. Our proposed scheduling strategy, CAPA, applies some methods leading to achieve the main goal of improving perceived video quality; 1) adaptive video traffic split scheme to allocate a proper bit rate for each transmission path considering heterogeneous network context with aim of executing load balancing, relieve congestion, and proper utilizing of each path capacity; 2) markovian based method to estimate path conditions and transition probabilities; 3) discard strategy to reduce congestion by avoiding sending packets that would probably be lost; and 4) content-aware strategy to select the best path for transmitting each video packet, and protecting packets with high priority.

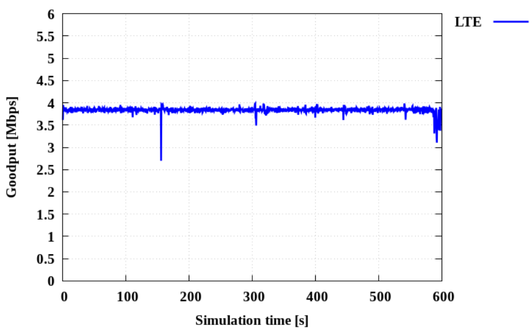
It is important to highlight that our approach does not require any change in the protocol itself since the scheduler can be implemented as part of the client/server applications. Besides, we leveraged the feedback signaling mechanisms defined in the MMT standard.

Even if CAPA can successfully adapt to varying network path conditions taking into account the characteristics of the video content, however, there are some candidate improvements to be done. For example, one possible improvement for content-aware strategy is adding a Forward Error Correction (FEC) stage exploiting what is already specified as the Application Layer FEC of the MMT standard. Besides, it is possible to improve adaptive video traffic split scheme by considering more network metrics. One other candidate future work is that the fix parameters defined in the path estimator could be adaptive either at configuration or per application or per profile. The current discard strategy could also improve considering the packet deadline, estimating the arrival time of each packet, and consequently, discarding the ones who cannot reach the receiver in time. Estimating arrival time for packets could also help the scheduler to send the packets with closer deadlines on the path with the shortest delay. Other candidate improvements are employing Scalable Video Coding (SVC) and Variable Bit Rate (VBR).

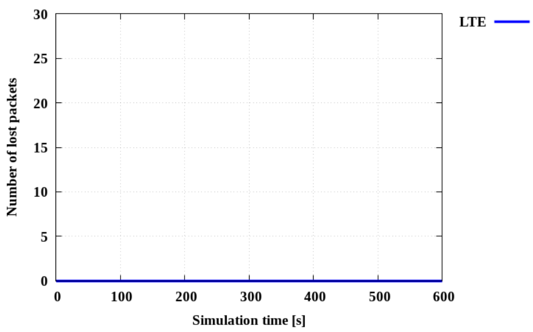
4 Performance Evaluation

In this chapter, we provide simulation results to evaluate the efficacy of the proposed CAPA. In this way, firstly, we explain our implemented environment by ns-3 simulation in Section 4.1. Then, video sequences, selected and encoded to test in our environment, are described in Section 4.2. We also define key performance indicators to evaluate the user-perceived streaming video quality in Section 4.3.

In the defined environment (Section 4.1), a sender and a receiver are connected via WiFi and LTE paths. Each of these paths can support the video streaming alone without congestion when they have unchanged normal background traffic (70% of full link capacity for downlink and 10% of full link capacity for uplink) with no wireless loss rate. For instance, the achieved goodput and number of lost packets for LTE path when all the *Elephants Dream* packets are sent over LTE with a rate of 4 Mbps are illustrated, respectively, in Figures 30(a) and 30(b). These figures attest that LTE can handle its background traffic plus the video traffic without congestion and loss. The achieved goodput and number of lost packets for WiFi path when all the *Elephants Dream* packets are sent over WiFi with a rate of 4 Mbps are illustrated, respectively, in Figures 31(a) and 31(b). Similarly, these figures attest that WiFi can handle its background traffic plus the video traffic without congestion and loss. However, even if each path can support the video streaming but having multipath video delivery could increase the speed and achieve higher goodput.



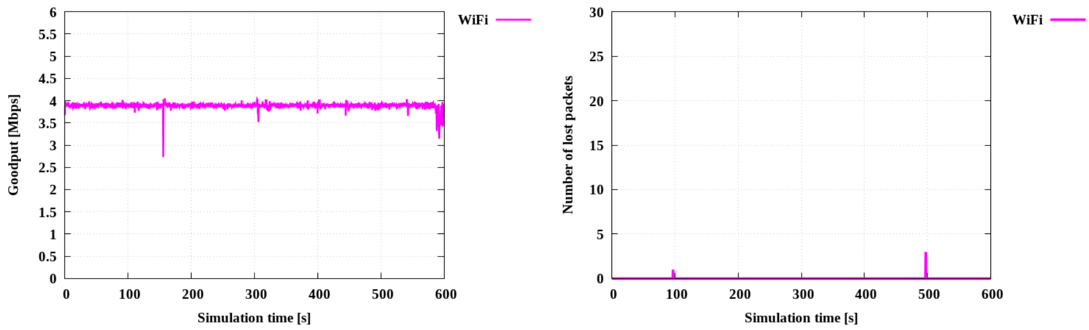
(a) LTE goodput



(b) LTE packet losses

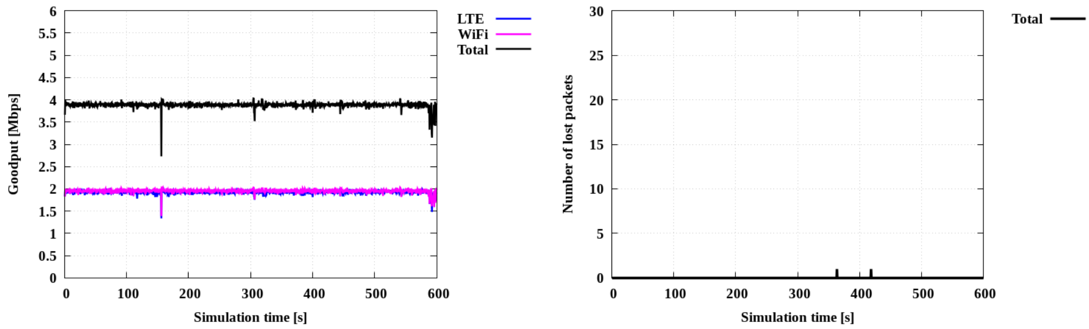
Figure 30 – LTE path condition when all the *Elephants Dream* packets are sent over LTE with unchanged normal background traffic and no wireless loss rate.

There are many ways to do multipath. The simplest one is evenly splitting (ES), where packets are evenly split in both transmission channels. Even though ES is a simple scheduling strategy, it can take advantage of the network multipath capabilities and it can increase the total achieved goodput. The achieved goodput and number of lost packets for



(a) WiFi goodput (b) WiFi packet losses

Figure 31 – WiFi path condition when all the *Elephants Dream* packets are sent over WiFi with unchanged normal background traffic and no wireless loss rate.



(a) LTE, WiFi and total (joint) goodput (b) Total packet losses

Figure 32 – The achieved goodput and total losses results when the *Elephants Dream* packets are distributed according to ES with unchanged normal background traffic and no wireless loss rate.

ES with a rate of 4 Mbps when similarly the paths have unchanged normal background traffic (70% of full link capacity for downlink and 10% of full link capacity for uplink) with no wireless loss rate are illustrated, respectively, in Figures 32(a) and 32(b). Results show the same behaviour for ES too, which is ES ability to support the video traffic without congestion and loss. However, ES cannot achieve the available capacity of each path properly. Besides, the network situation is not always with constant background traffic. In real network environments, background traffic varies and congestion can occur in the network. Therefore, it is required to have a multipath method which can achieve load balancing in order to relieve the congestion. Another challenging network condition is burst wireless losses, which is also a common network condition. A proper multipath strategy could protect the video content for this network condition.

Therefore, we came up with another scheduling strategy; path-aware (PA). PA considers paths' condition for packet transmission. This approach assigns a proper bit rate for each path in order to achieve load balancing. In addition, it also has a discard strategy to reduce congestion. However, although considering video content features in scheduling strategy could improve the QoE and network utilization, it is not considered in PA. PA scheduling strategy previously detailed in Subsections 3.1.5.1, 3.1.5.2 and 3.1.5.3.

Finally, we proposed a Content-Aware and Path-Aware (CAPA) scheduling strategy. CAPA is not only aware of paths' conditions to balance the load and apply discard strategy but also it is aware of video contents. Therefore, CAPA is capable of selecting the best path for transmitting each video packet, and it can also protect the ones with high priority in order to face all the previously mentioned challenges and improve the perceived video quality. CAPA scheduling strategy previously detailed in Subsections 3.1.5.1, 3.1.5.2, 3.1.5.3, and 3.1.5.4. In other words, PA is the CAPA strategy while content-aware is disabled. Comparing CAPA with PA makes clear how much content-aware protection contributes to the overall performance.

In the next step, we define different network conditions in order to evaluate our proposed strategy, CAPA. Firstly, we consider congestion network scenario, which could significantly decrease the perceived video quality. We also study our scheduler behaviour with different video qualities under our congested network environment. Then, we consider wireless channel lossy networks, where channels have high burst losses. Regarding (XU *et al.*, 2013), this condition could even occur more than congestion condition in the network, which has a high adverse effect on the perceived video quality. These defined simulation scenarios are described together with results discussion in Section 4.4. In Section 4.5, we evaluate our proposed strategy when it competes with another MMT flow for fairness support. Finally, we conclude remarks in Section 4.6.

4.1 Simulation Setup

A ns-3 DCE (NS3, 2016) model is implemented to simulate the proposed strategy in a realistic network scenario. DCE is a module for ns-3 that allows to execute existing implementations of network protocols or applications without source code changes, and it has the capability of using Linux kernel instead of using some open source implementation, which may be unstable and incomplete.

In the evaluation environment, as shown in Figure 33, our multipath simulation setup comprises of two wireless networks: LTE and WiFi. These networks are implemented by the LTE and WiFi modules available in the ns-3 simulation library. We chose LTE and WiFi connections because this is the most common setup for today's mobile devices. The different specifications and heterogeneity between LTE and WiFi are among the big challenges of this work for properly splitting video traffic and achieving load balancing. As summarized in Table 11, for the LTE path, based on (OPENSIGNAL, 2017) and (USMAN; PRIHATMOKO, 2011), bw_p and $delay_p$ are defined, respectively, as 18.3 Mbps and 15 milliseconds (ms). The 802.11n/5GHz model is chosen for the WiFi path. Therefore, bw_p and $delay_p$ are defined for the WiFi path, respectively, as 54 Mbps and 10 ms (KIM *et al.*, 2004; OH; LEE, 2015). The packet loss rate values are set later based on different

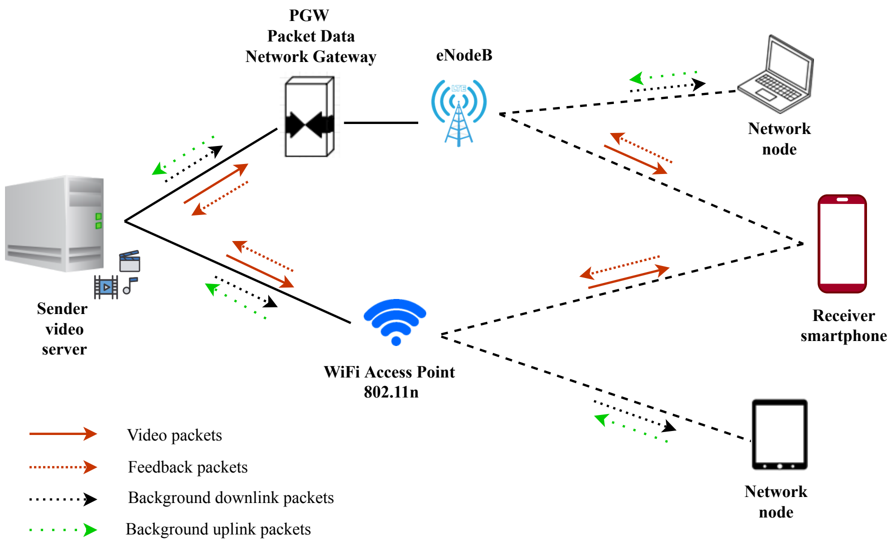


Figure 33 – Evaluation environment where a sender video server uses multiple paths to stream video to a receiver smartphone.

network scenarios.

Table 11 – Main network parameters.

Path	Maximum bandwidth of path (bw_p)	Minimum delay of path ($delay_p$)
LTE	18.3 Mbps	15 ms
WiFi	54 Mbps	10 ms

Besides, as depicted in Figure 33, uplink and downlink background traffics are added in the ns-3 simulation. The downlink background traffic is generated by the server for both paths. The uplink background traffic is generated by the network nodes. Background traffic condition detail is explained later based on the different network scenarios.

4.2 Video Sequences

We have selected a broad range of sequences including cartoon and natural scene which are the movies with motion, details and very tough colors. The cartoon sequences are named *Elephants Dream*, *Big Buck Bunny* and *Sintel*, and the natural video sequences are named *Meridian*, and *LIVE*. The *LIVE* video sequence is a concatenated of nine short videos available in Image & Video Engineering (LIVE) Laboratory[1]; *AirShow*, *AsianFusion*, *Chimera1102347*, *Chimera1102353*, *ElFuenteDance*, *ElFuenteMask*, *Skateboarding*, *Soccer*, and *Sparks*. We show a sample frame of each sequence in Figure 34. We also summarize a short video description and some of the video content characteristics in Table 12.

The video sequences have 1920×1080 resolution and 15,000 total number of frames. The GoP size is of 16 frames, and the employed GoP structure is IPPPP...P without B frames. The H.264/AVC JM Reference Software (JM, 2015) is used as the encoding tool

[1] http://live.ece.utexas.edu/research/Quality/index.htm

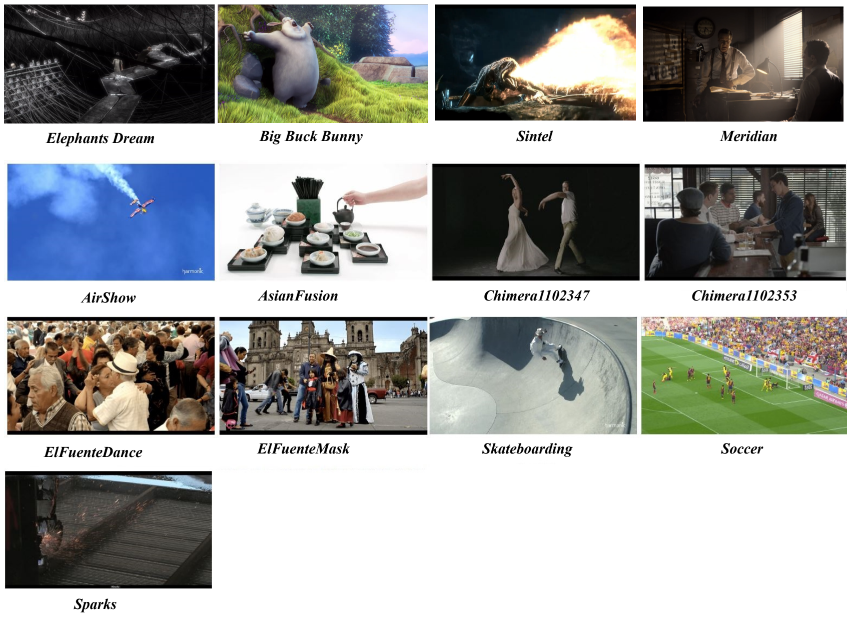


Figure 34 – Video frame examples of each video sequence and each short video content of *LIVE* sequence.

and the MP4 fragmentation procedures are done by the GPAC MP4BOX (GPAC, 2015) tool. Decoding and error concealment are performed with FFmpeg (FFMPEG, 2017). More details about encoding and decoding process is provided in the next subsection.

All the sequences are encoded with the same source bit rate of 4 Mbps. The video original PSNR values (PSNR value without losing any packet) are shown in Table 13, ordered from the highest to the lowest number. As expected, we can see that videos have different original qualities due to the different content of each video. However, all of them have enough quality for video transmission above 40 dB.

We also encoded *Meridian* and *Big Buck Bunny* with source bit rates of 3 and 5 Mbps for a specific scenario explained in Subsection 4.4.1.3. We selected these two video sequences because they have respectively the highest and the lowest original PSNR when encoded with source bit rate of 4 Mbps. Table 14 shows the video original PSNR values related to each of these encoding rates.

Note that video encoded with higher rate has larger PSNR. This is clearly shown in Table 13 together with Table 14. For example, *Meridian* with bit rates of 3, 4, and 5 has 51.1, 52.35, and 53.34 dB respectively. Similarly, *Big Buck Bunny* with bit rates of 3, 4, and 5 has 41.72, 42.95, and 44.02 dB respectively. We provide more details in Section 4.4.1.3.

Parameter choice. To select the number of P frames as NI frames in this work, we compare the PSNR results of CAPA considering different number of P frames, from 0 to 4, under the congested network scenario, which will be explained in Subsection 4.4.1. In this scenario, all the video sequences are encoded with 4 Mbps. PSNR results are summarized in Table 15. The second column of the table shows that PSNR results in

Table 12 – Description and content characteristics of video sequences.

Type of video	Video sequence	Description and content characteristics
Cartoon sequences	*Elephants Dream* (BLENDER, 2005)	Developed by the Blender Institute, About two characters exploring the surrounding world composed by machines, Long sequence, spatial detail complexity (PINSON *et al.*, 2013), amount of camera motion, fast motion (KYUNG *et al.*, 2016)
	Big Buck Bunny (BLENDER, 2007)	Developed by the Blender Institute, Showing animals of the forest, Long sequence, outdoor environments, large grassy fields and trees with leaves, Featuring animals' hair and fur, high detail, fast motion (KYUNG *et al.*, 2016)
	Sintel (BLENDER, 2009)	Developed by the Blender Institute, About a female protagonist and a dragon, Long sequence, fire/smoke/volumetrics and explosions, fast motion (PINSON *et al.*, 2013)
Natural video sequences	*Meridian* (NETFLIX, 2016)	Produced by Netflix, About detectives investigating, Long sequence, smoke, rain, harsh lighting, human face, low motion
	LIVE (UTEXAS, 2018)	*LIVE* is concatenated of following short video sequences produced by Netflix (BAMPIS *et al.*, 2018): **AirShow:** showing aircraft exhibition, Camera tracks object of interest, blue sky background **AsianFusion:** showing fusion cuisine, Static camera, zoom-in, uniform background **Chimera1102347:** showing bar scene, Multiple human faces, zoom-in, low motion **Chimera1102353:** showing couple dancers, Multiple human faces, zoom-in, low motion **ElFuenteDance:** showing Aztec ritual dance, drums, close-up, couples dancing tango, wide angle view, Rich spatial activity, multiple human faces **ElFuenteMask:** showing "Día de los muertos" sculpture closeup, costumes group, costume closeup, Medium spatial activity, saliency, very tough colors, rich texture **Skateboarding:** showing skateboard riding, Fast motion, complex camera motion, saliency, rich texture **Soccer:** showing epic goals in soccer and football fans in stadium, Fast moving camera, rich spatial and temporal activity **Sparks:** showing sparks, Slow camera motion, human face, fire sparks, water

Table 13 – Original PSNRs for video sequences encoded with 4 Mbps.

Video sequence	Original PSNR [dB]
Meridian	52.35
Sintel	50.86
Elephants Dream	47.07
LIVE	43.60
Big Buck Bunny	42.95

case of considering none of P frames as NI frame (NI=0), the third column shows PSNR results considering the first initial P frame as NI frame (NI=1) and so on. One can see in this table is that most of the video sequences have the best results in case of considering 3 initial P frames as NI frames. Therefore, in this work, only the initial 3 P frames in the GoP were considered as NI frames and the remaining 12 frames in each GoP are regular P frames. Table 15 also clearly shows the efficiency of protecting NI frame packets to improve the PSNR.

Table 14 – Original PSNRs for video sequences encoded with 3 and 5 Mbps.

Video sequence	Encoding rate [Mbps]	Original PSNR [dB]
Meridian	3	51.1
	5	53.34
Big Buck Bunny	3	41.72
	5	44.02

Table 15 – PSNR results from the different video sequences encoded with 4 Mbps considering different number of P frames as NI frames. These results are according to CAPA under congested network scenario.

Video sequence	PSNR [dB]				
	NI=0 (No P frame)	NI=1 (The initial 1 P frame)	NI=2 (The initial 2 P frame)	NI=3 (The initial 3 P frame)	NI=4 (The initial 4 P frame)
Meridian	41.18	41.88	42.41	42.30	42.28
Sintel	38.54	39.30	39.53	39.80	39.62
Elephants Dream	36.14	36.42	36.65	37	36.31
LIVE	33.52	33.72	34.29	34.43	34.23
Big Buck Bunny	33.53	33.98	34.50	34.25	34.49

Since the PSNR results always depend on the content and original sequences, we show the distribution of I and NI frame packets when the videos are encoded with the source bit rate of 4 Mbps in Table 16. The second column of Table 16 shows the distribution of I frame packets. The third column to the last column of Table 16 show the distribution of NI packets in case of considering different numbers of initial P frames in the GoP as NI frames. One can observe from this table is that the sequences have different packet distributions due to the different amount of texture, details, and actions. As shown in this table, among our video sequences, *Big Buck Bunny* has the most I packets (41.20%). Considering 3 initial P frames as NI frames, *Big Buck Bunny* has 10.98% NI frame packets, and the rest (47.82%) is P frame packets. Note that, since CAPA protects I and NI packets, therefore in this example, it protects 52.18% video packets of this video. Considering three initial P frames as NI frames, *Big Buck Bunny* has the highest I and NI frame packets (52.18%) and after that, *Meridian* has 48.97%. This values for *LIVE* and *Elephants Dream* are, respectively, 42.74% and 42.53%, and finally, *Sintel* has the lowest I and NI frame packets of 37.58%.

Table 16 – Packets distribution according to frame type for video sequences encoded with 4 Mbps.

Video sequence	I packets [%]	NI packets [%]			
		NI=1	NI=2	NI=3	NI=4
Meridian	32.04	6.04	12.03	16.93	21.65
Sintel	18.88	7.00	12.70	18.70	24.25
Elephants Dream	25.38	6.25	11.98	17.15	22.13
LIVE	23.99	7.90	13.54	18.75	23.90
Big Buck Bunny	41.20	4.55	10.34	10.98	19.78

Selecting 3 initial P frames as NI frames in the GoP, Table 17 shows the distribution of I and NI frames for *Meridian* and *Big Buck Bunny* encoded with source bit rates of 3 and 5 Mbps.

Table 17 – Packets distribution according to frame type for video sequences encoded with 3 and 5 Mbp when NI = 3.

Video sequence	Encoding rate [Mbps]	I packets [%]	NI packets [%]
Meridian	3	33.51	16.66
	5	30.84	17.09
Big Buck Bunny	3	42.20	14.54
	5	41.17	15.14

4.3 Key Performance Indicators

We evaluate our experimental results in terms of QoS metrics including delay, packet loss rate, and goodput in order to evaluate performance from the network perspective. Moreover, we also evaluate our experimental results in terms of QoE metrics including PSNR, and SSIM to evaluate video quality. We previously explained how goodput, packet loss rate, and delay are computed in Section 3.1.2. Here, we explain PSNR and SSIM metrics.

The QoE can be performed by subjective or objective methods. In subjective methods, a number of observers (i.e., "subjects") judge the quality of video and the results for each video sequence are expressed in Mean Opinion Score (MOS) given as an average for all observers. Objective methods are computational models to estimate the video quality (VRANJEŠ *et al.*, 2013; SHAHID, 2014). However, subjective quality experiments are expensive to process. Besides that, in this work, it is important to measure quantification of the improvement or deterioration in quality. Therefore, we used objective metrics of PSNR, and SSIM to evaluate video quality cause these two are popular full reference metrics. In full reference metrics, the original video is used as a reference and distorted video is compared with the original one. More information about objective video quality metrics and performance comparison can be found at (VRANJEŠ *et al.*, 2013).

- **PSNR:** PSNR is a basic, yet, important objective video quality metric. This metric is based on the pixel-by-pixel comparison expressed as a function of Mean Squared Error (MSE) between the original and the distorted video frames represented in dB. Frames with more similarity give higher PSNR values. The correspondence between the values of PSNR with the MOS is shown in Table 18.

- **SSIM:** SSIM is one of the most common objective video quality metrics, which shows the similarity of the original frames with distorted video frames. This metric compares not the pixel values but the frame elements perceived by the human. Therefore, it describes the image quality differences better than PSNR. It uses structural distortions to evaluate the perceptual distortion based on the three factors: luminance, contrast, and texture. The final SSIM index takes together all of these distortions. Result of SSIM is a number between -1 and 1 that represents the video quality, where 1 can only be achieved if both frames are identical. The correspondence between the values of SSIM with the MOS is shown in Table 18.

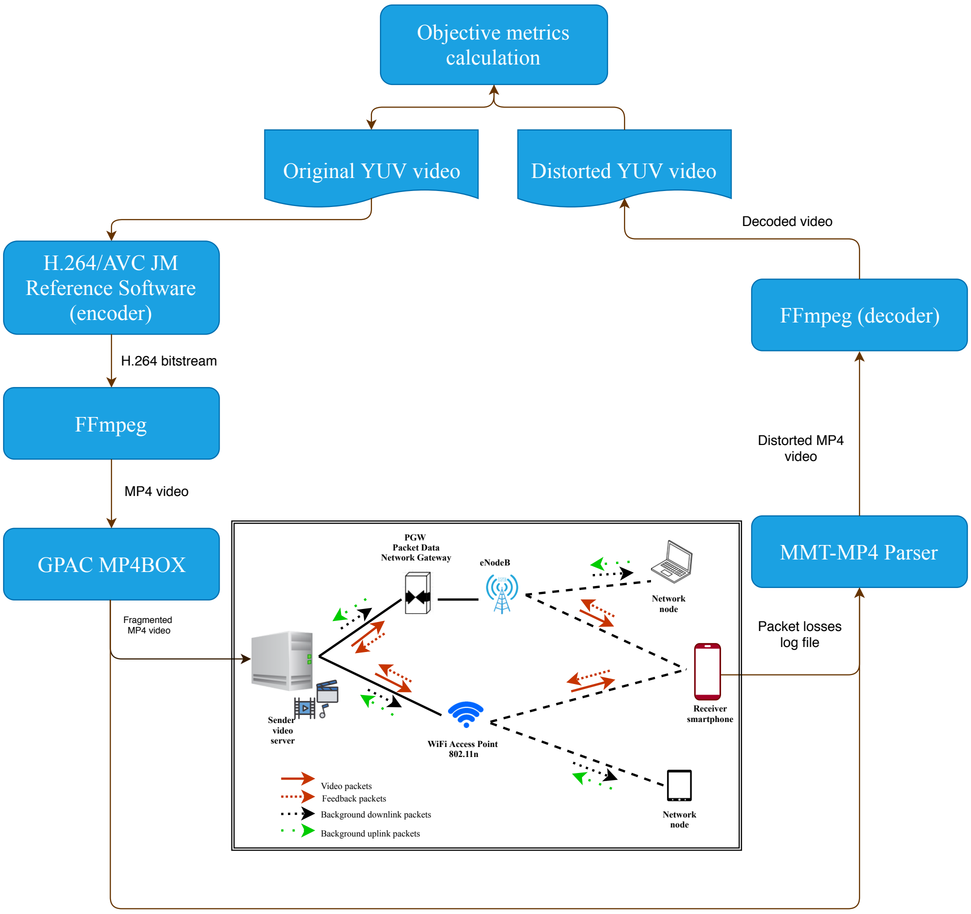


Figure 35 – System architecture for performance evaluation in this thesis - objective video quality metrics.

The system architecture for performance evaluation considering objective metrics is shown in Figure 35. In this figure, the original YUV video is encoded by the H.264/AVC JM Reference Software (JM, 2015). Then, FFmpeg is used to encapsulate the video in MP4 file. After that, the MP4 fragmentation procedures are done by the GPAC MP4BOX (GPAC, 2015) tool. After data transmission through the ns-3 simulation, in order to provide the distorted video to compute PSNR and SSIM, we developed a C++ application, namely MMT-MP4 parser. MMT-MP4 parser fills the lost packets' positions with zeroes in the original MP4 file. In particular, MMT-MP4 parser is capable of understanding the MP4 atom structure and then capable of applying zeroes in the right positions according to the information of packet losses calculated on the receiver. This way, from the original MP4 video, we can generate the received MP4 sequence by means of adding the transmission errors (distorted MP4 video). After this, the FFmpeg error concealment method is responsible for trying to repair the corrupted missy bitstream and therefore reconstruct the decoded video (distorted YUV video). Then, PSNR and SSIM are computed between the original YUV video and the distorted YUV video.

Different objective metrics (JUNG, 2017; NETFLIX, 2019b; BT.1683, 2004; J.144, 2004) have been proposed in literature to predict perceived video quality. One candidate future work could be evaluating our video quality with more metrics such as MS-SSIM which shows close aligned with the results that would have been obtained with subjective testing (CLARITY, 2019).

Table 18 – Mapping PSNR and SSIM to MOS scale based on VQAMap (MOLDOVAN *et al.*, 2016) rules.

MOS Quality	PSNR	SSIM
5 (Excellent)	≥ 36	≥ 0.93
4 (Good)	≥ 29 & < 36	≥ 0.85 & < 0.93
3 (Fair)	≥ 22 & < 29	≥ 0.76 & < 0.85
2 (Poor)	≥ 20 & < 24	≥ 0.62 & < 0.76
1 (Bad)	< 20	< 0.62

4.4 Different Network Scenarios

In this section, we discuss the performance of our proposed CAPA in different network situations; congested network in Subsection 4.4.1 and wireless lossy network in Subsection 4.4.2. These common network situations cause burst packet losses in the network, and consequently, the receiver may not receive a sufficient number of packets to properly decode the video. Therefore, it could have a severe adverse effect on the perceived video quality. We remind, in this thesis, the packet losses caused due to network congestion are called congestion losses and the packet losses caused due to wireless errors (e.g., lossy channels, noise or interface) are called wireless losses.

4.4.1 Congested Network Scenario

In this subsection, we discuss the performance of our proposed CAPA in a congested network situation. Therefore, in this scenario, we vary background traffic to make congestion in the network in order to observe the response of the scheduling strategy. This background traffic condition will be detailed in Subsection 4.4.1.1. Besides, in order to have a more realistic simulation setup, the ns-3 channel random error is employed to capture the effects of noisy wireless channels. Related work (CHEN *et al.*, 2013) assumes loss rate values of 1% for the WiFi path and 0.1% for the LTE path. However, due to the results of our simulations where sequence losses for WiFi become extremely high (cf. results in Annex A), we opted to reduce the maximum random loss rate to 0.5%, which is still a realistic assumption.

We discuss the performance of our proposed scheduler under the defined congested network scenario (where all sequences are encoded with the same bit rate) in Subsection 4.4.1.2. We also study the behaviour of our proposed strategy, CAPA, for different video bit rates under the same congested network scenario in Subsection 4.4.1.3.

4.4.1.1 Background Traffic Condition

Evaluation considering different background traffic models has been one of the main challenges to be addressed. For a broad and sound evaluation, the background traffic conditions to be explored should cover most of the potential real cases while being able to evaluate the performance of the scheduling strategy.

In this scenario, constant uplink and variable downlink background traffics are added in the ns-3 simulation to stress out the proposed scheduling strategy in different network congestion situations. The downlink background traffic is generated by the server and initially set as 70% of full link capacity for both paths. On the other hand, the uplink background traffic is generated by the network nodes and set as 10% of each full link capacity, in accordance to real network scenarios where the uplink traffic is smaller than the downlink traffic.

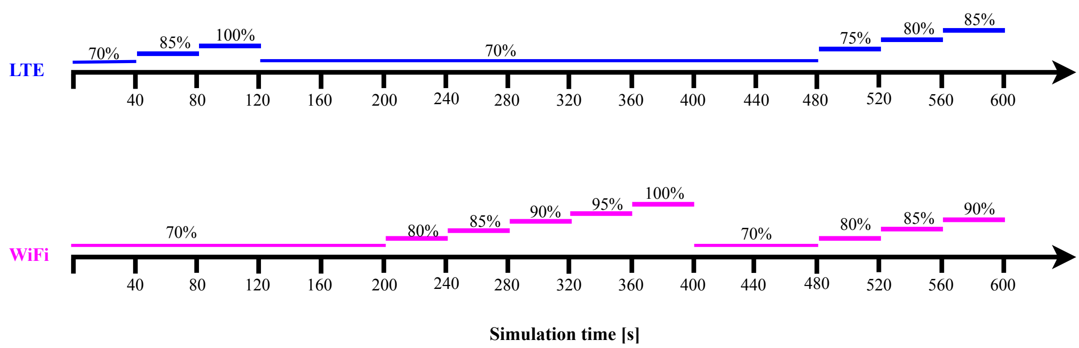


Figure 36 – Variable downlink background traffic setup to first simulate congestion separately in each path and then simultaneous congestion in both paths.

Variable downlink background traffic is illustrated in Figure 36. As illustrated in this figure, we have 3 network congestion parts in our simulation; LTE congestion, WiFi congestion, LTE and WiFi congestion. The first part of the simulation, LTE congestion, is from 40 to 120 seconds of simulation time. During this period, the background traffic of only the LTE path is increased up to completely (100%) saturating the LTE channel while the background traffic of the WiFi is kept constant. For this purpose, after 40 seconds of simulation time, the LTE background traffic is increased to 85% of its full link capacity, then from 80 to 120 seconds, it's background traffic is increased to 100%.

In the second part, WiFi congestion, the opposite behaviour is simulated and, from 200 to 400 seconds of simulation time, the background traffic of the WiFi path is increased up to completely (100%) saturating the WiFi channel while the background traffic of the LTE is kept constant. For this purpose, after 200 seconds of simulation time, the WiFi background traffic is increased to 80% of its full link capacity, then from 240 to 280 seconds, it's background traffic is increased to 85%, to 90% from 280 to 320 seconds, to 95% from 320 to 360 seconds, and to 100% from 360 to 400 seconds.

Finally, in the last part, from 480 to 600 seconds of simulation time, background traffic is increased in both paths to simulate simultaneous congestion in LTE and WiFi. For this purpose, the LTE background traffic is increased to 75% from 480 t0 520 seconds, to 80% from 520 to 560 seconds, to 85% from 560 to 600 seconds. Simultaneously, the WiFi background traffic is increased to 80% from 480 t0 520 seconds, to 85% from 520 to 560 seconds, to 90% from 560 to 600 seconds of simulation time.

This explained background traffic is also depicted as a plot in Figure 37. This plot would be used in the rest of this thesis together with goodput and PSNR graphs in order to facilitate readers to track the network condition and understand graphs explanation easily. Besides, in this figure, three background traffic level is defined; normal, high and very high background traffic. We defined that the path has normal background traffic when its amount is around 65% to 75% of full link capacity. We defined it as high background traffic when its amount is around 75% to 90% of full link capacity, and we defined it as very high background traffic when its amount is more than 90% of full link capacity. Therefore, one can note regarding this proposed background traffic is that we considered extreme cases. For example, having very high background traffic is a rare network situation and not a condition that happens all the time.

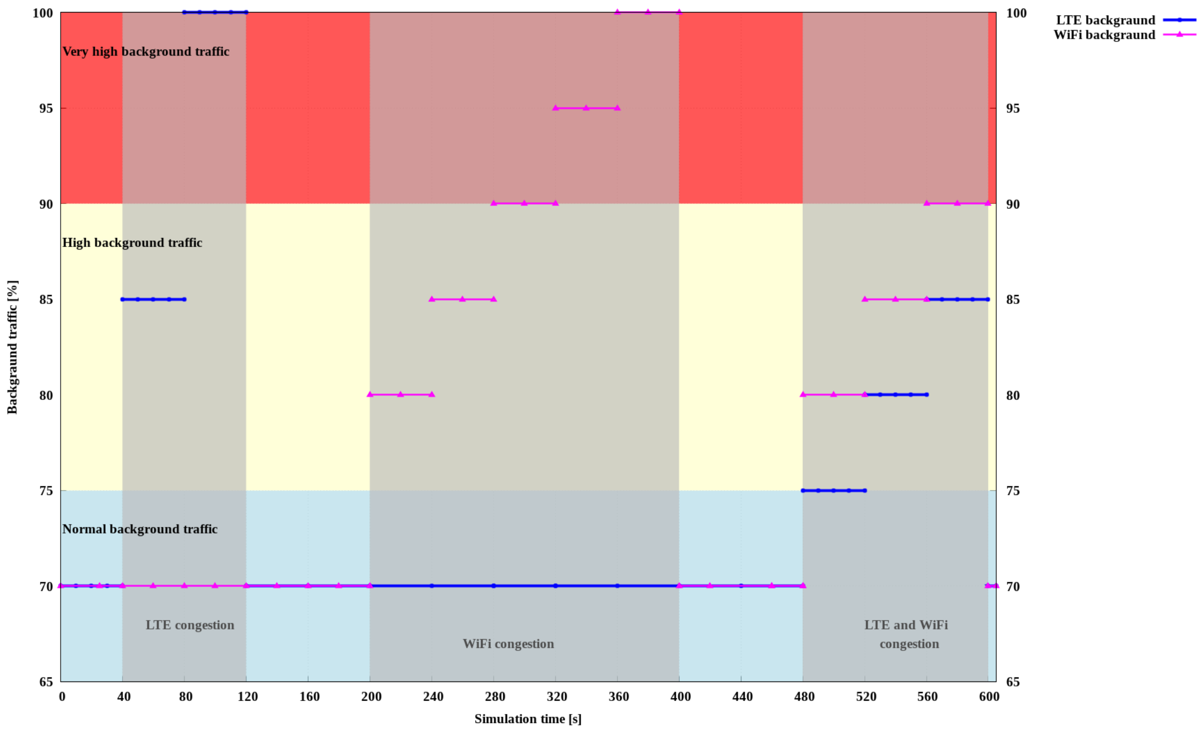


Figure 37 – Variable downlink background traffic setup to first simulate congestion separately in each path and then simultaneous congestion in both paths showing in a plot form.

4.4.1.2 Experimental Evaluation for Constant Transmission Bit Rate

Here, we discuss the performance of our proposed scheduler under the defined congested network scenario. In order to compare results in the same network simulation scenario, all sequences are encoded with the same source bit rate (4 Mbps) - previously explained in Section 4.2. We also consider the constant transmission bit rate stream in this experiment (4 Mbps for this scenario).

Figure 38 depicts the absolute number of video packets (VP) for each video, the number of packets sent, the total number of packets received unique (no duplicated packet), and the total number of packets received (including duplicated packets), measured from the various video sequences according to the different scheduling strategies. One can see in this figure is that more packets received unique by CAPA strategy compared to both PA and ES strategies for all sequences. The reason is that CAPA can effectively allocate a proper bit rate for each path based on their available bandwidth. Besides, some packets with low priority are discarded to reduce congestion, and the packets with high priority would be rerouted or even duplicated to ensure reliable data delivery during congestion network. However, duplication for protecting packets has cost of sending more packets through the network. It is important to highlight that high priority frames have higher number of packets than low priority ones. Therefore, it is possible to see this fact in the figure that the total packets sent according to CAPA strategy is higher than the number of video packets for all sequences. Note that content of video always effects on the performance and the amount of duplication.

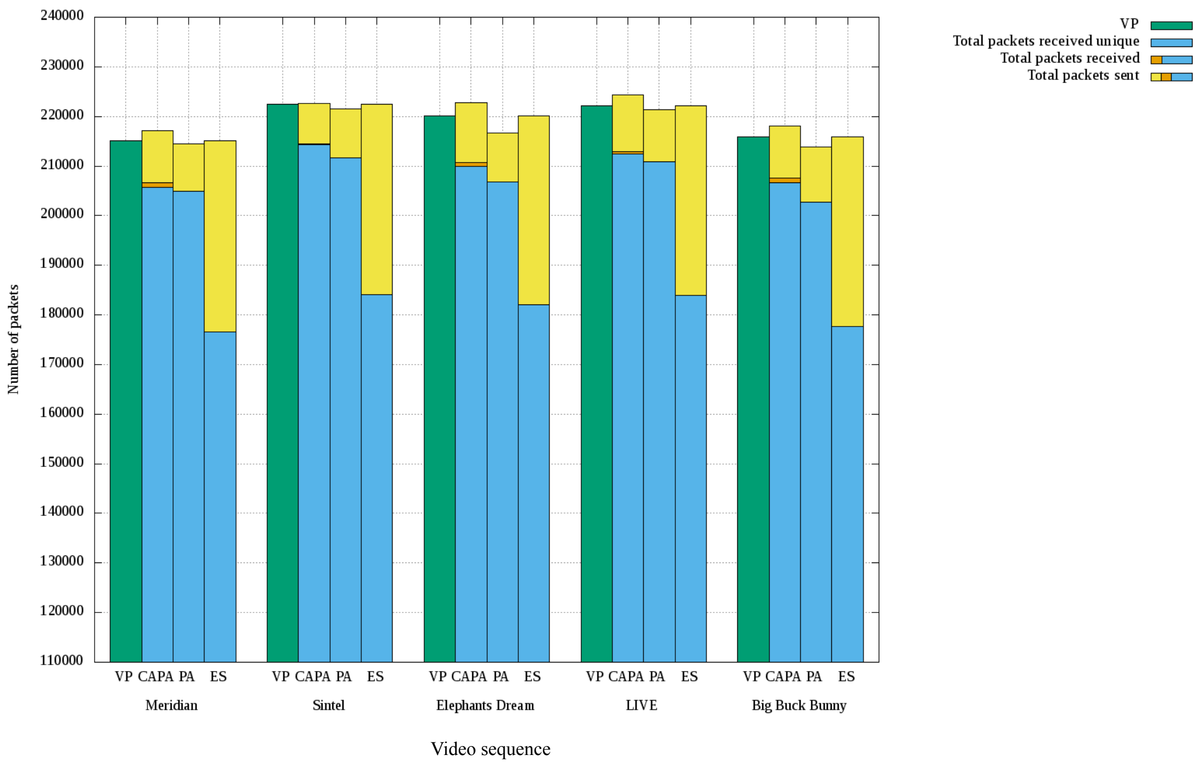


Figure 38 – Performance comparison of the various video sequences according to the different scheduling strategies under congested network scenario.

In contrast to the CAPA, as shown in Figure 38, the total packets sent according to PA strategy is less than the number of video packets. This is due to this fact that PA discards some packets at the sender to reduce congestion (and there is no duplication). Even if PA does not outperform CAPA due to lack of content awareness, but yet, PA has received more packets compared to ES because it has a proper load balancing to cope with network congestion. Regarding ES strategy, the total number of packets sent is the same as the number of video packets due to there is not either any discard or any duplication. ES has received the lowest number of packets compared to the other approaches for all sequences, which means it has the highest losses. This is because ES suffers from lack of any proper strategy to cope with network condition and load balancing.

Yet another note regarding Figure 38 is that the difference of total received packets and total received packets unique (orange color in bars), in CAPA strategy, shows the number of packets received duplicated. It can be clearly noted in this figure that these duplicated received packets are very small for *Sintel* sequence compared to the others. The reason is that *Sintel* has the lowest I and NI packets and, therefore, the lowest number of duplication among our sequences.

Network quality of service results.

Goodput: Figure 39 shows the LTE, WiFi and total (joint) goodput for *Elephants Dream* sequence under congested network scenario according to the different scheduling strategies; (b) CAPA, (c) PA, and (d) ES. In order to facilitate readers tracking the network condition, the defined background traffic, previously explained in Subsection 4.4.1.1, is also shown in Figure 39(a) top of the goodput subfigures.

Figure 39(b) illustrates that the total achieved goodput is higher and more stable when our proposed scheduling strategy is applied compared to PA and ES distribution. One can note is that load balancing is clearly achieved between both paths and the higher inherent capacity of the WiFi path is exploited. In addition, congestion (first in LTE and then in WiFi) is properly handled by the scheduler by switching traffic among paths and keeping a stable total goodput of 4 Mbps through all simulation, except for the last part where both paths get congested. Goodput decrease in this last part is not only due to congestion but also to packets lost due to the discard strategy applied by the scheduler to avoid further congestion. One important note is that discard strategy is only applied on the low priority packets and not on high priority packets. Moreover, the content-aware packet protection method protects the high priority packets by duplicating or rerouting.

Figure 39(c) illustrates the total achieved goodput according to PA scheduling strategy. Similar to the CAPA strategy, PA can provide load balancing, achieve higher inherent capacity of the WiFi path and keep a stable total goodput of 4 Mbps through all simulation, except for the last part where both paths get congested. One can note is that the goodput reduction in the last part of the network according to PA strategy is more than goodput reduction at the same time according to CAPA strategy. This is due to this fact that the discard strategy is applied on all types of packets (blindly and without

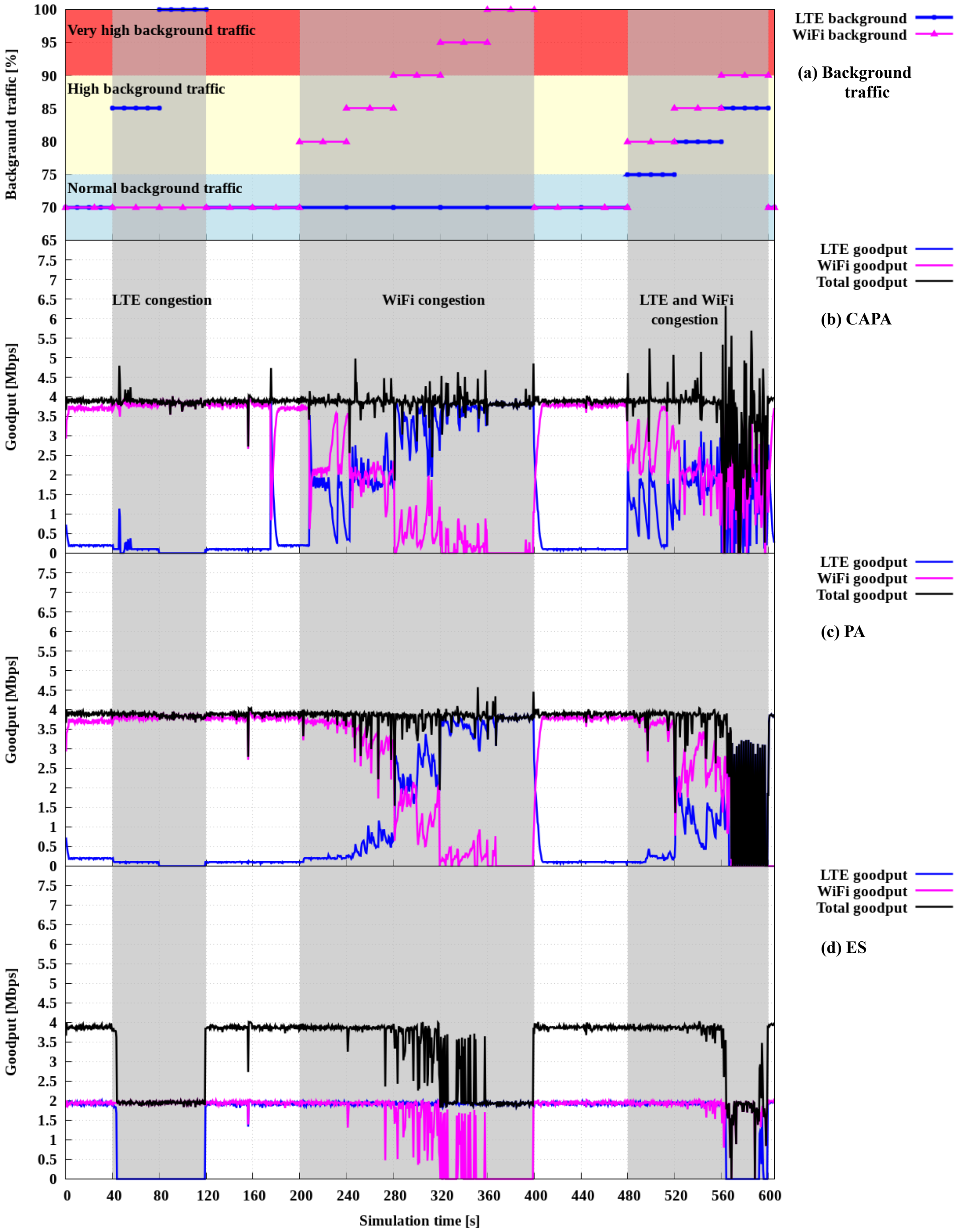


Figure 39 – LTE, WiFi and total (joint) goodput for *Elephants Dream* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under congested network scenario.

considering the video packet content priority) and there is no content-aware strategy to protect high priority packets.

One can observe from Figure 39(d) is that, at the initial part of simulation, where there is normal background, video traffic is equally split between LTE and WiFi. Therefore, each channel has a goodput of 2 Mbps and the total goodput is of 4 Mbps. Then, when LTE gets heavily congested, its goodput sharply decreases due to packet losses, and the total goodput (2 Mbps) achieved in this period is only due to packets transmitted over WiFi. In the second part of the simulation, LTE recovers from congestion and then WiFi gets congested. However, since WiFi inherent capacity is higher than LTE, congestion is better handled for a while but since the congestion is increasing then the the goodput decreases to almost zero for WiFi. Finally, in the last part where both paths get congested, the goodput reduction is clearly noticeable. This behaviour of ES is due to lack of any path-aware or content-aware strategy.

The goodput performance according to different strategies for other tested video sequences are provided in Annex C due to they have the same behaviour of what already explained for *Elephants Dream*.

Packet loss rate: Figure 40 shows the effectiveness of the proposed scheduling strategy to decrease loss rate in CAPA compared to PA and ES scheduling strategies. The results show that CAPA decreases the total loss rate, respectively, by up to **31.82%**, and **78.96%** compared to PA and ES. There is also better protect of I and NI frame packets according to CAPA compared to PA and ES which is shown in Figure 41. The results show that CAPA decreases the I and NI frame packet loss rate, respectively, by up to **53.37%**, and **86.39%** compared to PA and ES. In addition, as it is shown in Figure 41, due to differences of the video sequences previously mentioned and shown in Table 16, *Big Buck Bunny*, the one with the highest percentage of I and NI frame packets, has also the most I and NI packet losses in all compared conditions.

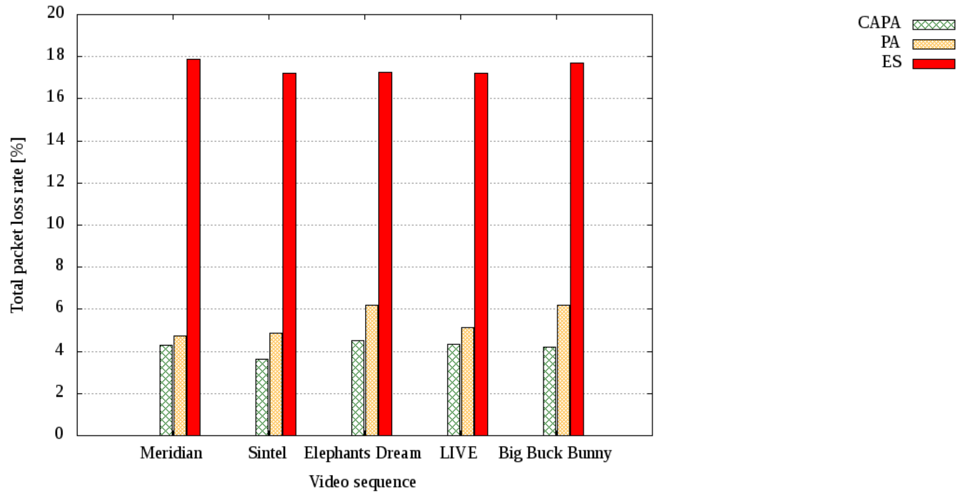


Figure 40 – Comparison of loss results from the various video sequences according to the different scheduling strategies under congested network scenario.

Having Figure 40 together with Figure 41 shows the proposed scheduling strategy gains better protect of I and NI frame packets. For all sequences, the I and NI frame packet loss rate over the total packet loss rate decreases. For instance, according to our proposed

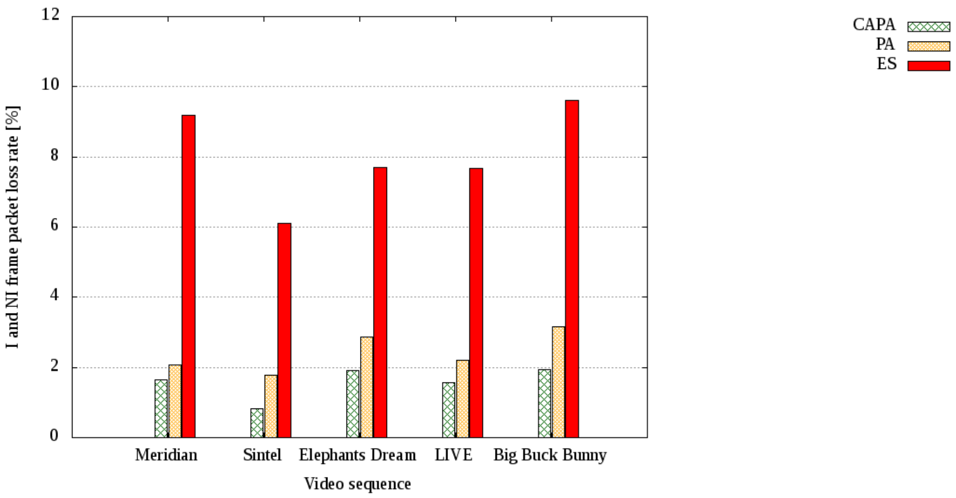


Figure 41 – Comparison of I and NI loss results from the various video sequences according to the different scheduling strategies under congested network scenario.

scheduling strategy, total lost packets for *Meridian* are 4.30% (of total sent packets) and 1.64% of these losses are I and NI losses (of total sent packets), which corresponds to 38.13% of the total losses. When PA scheduling strategy is applied, total lost packets are 4.72% and 2.06% of these losses are I and NI losses, which corresponds to 43.64% of the total losses. When ES scheduling strategy is applied, total lost packets are 17.87% and 9.18% of these losses are I and NI losses, which corresponds to 51.37% of the total losses. Therefore, the rate is reduced from 43.64% to 38.13% compared to PA, and from 51.37% to 38.13% compared to ES.

Similarly for *Sintel*, according to our proposed scheduling strategy, total lost packets are 3.62% and 0.83% of these losses are I and NI losses, which corresponds to 22.92% of the total losses. When PA scheduling strategy is applied, total lost packets are 4.86% and 1.78% of these losses are I and NI losses, which corresponds to 36.62% of the total losses. When ES scheduling strategy is applied, total lost packets are 17.21% and 6.10% of these losses are I and NI losses, which corresponds to 35.44% of the total losses. Therefore, the rate is reduced from 36.62% to 22.92% compared to PA, and from 35.44% to 22.92% compared to ES.

Also for *Elephants Dream*, according to our proposed scheduling strategy, total lost packets are 4.52% and 1.92% of these losses are I and NI losses, which corresponds to 42.47% of the total losses. When PA scheduling strategy is applied, total lost packets are 6.19% and 2.88% of these losses are I and NI losses, which corresponds to 46.52% of the total losses. When ES scheduling strategy is applied, total lost packets are 17.25% and 7.70% of these losses are I and NI losses, which corresponds to 44.63% of the total losses. Therefore, the rate is reduced from 46.52% to 42.47% compared to PA, and from 44.63% to 42.47% compared to ES.

Likewise for *LIVE*, according to our proposed scheduling strategy, total lost packets are 4.33% and 1.57% of these losses are I and NI losses, which corresponds to 36.25% of the total losses. When PA scheduling strategy is applied, total lost packets are 5.12% and 2.20% of these losses are I and NI losses, which corresponds to 42.96% of the total losses. When ES scheduling strategy is applied, total lost packets are 17.23% and 7.66% of these losses are I and NI losses, which corresponds to 44.45% of the total losses. Therefore, the rate is reduced from 42.96% to 36.25% compared to PA, and from 44.45% to 36.25% compared to ES.

Finally for *Big Buck Bunny*, according to our proposed scheduling strategy, total lost packets are 4.22% and 1.95% of these losses are I and NI losses, which corresponds to 46.20% of the total losses. When PA scheduling strategy is applied, total lost packets are 6.19% and 3.17% of these losses are I and NI losses, which corresponds to 51.21% of the total losses. When ES scheduling strategy is applied, total lost packets are 17.68% and 9.62% of these losses are I and NI losses, which corresponds to 54.41% of the total losses. Therefore, the rate is reduced from 51.21% to 46.20% compared to PA, and from 54.41% to 46.20% compared to ES.

Delay: Table 19 shows the CAPA average one-way delay reduction of non-overdue packets compared to PA and ES scheduling strategies under congested network scenario. In this table, the larger the negative values, the performance is better. The results clearly indicate that our proposed CAPA strategy efficiently enables a better adjustment compared to ES by balancing the bit rate distribution and the discard strategy. However, CAPA does not have always delay reduction compared to PA because actually the packets are not lost, meaning that we have a priority. The priority is to increase the quality of the perceived quality (measured by PSNR and SSIM). This way, CAPA reduced the number of packets, which would be lost in PA.

Table 19 – Results for delay reduction of CAPA compared with PA and ES scheduling strategies from the different video sequences under congested network scenario.

Video sequence	Scheduling strategy	Delay
Meridian	PA	-2.7%
	ES	-10%
Sintel	PA	0%
	ES	-7.6%
Elephants Dream	PA	0%
	ES	-7.5%
LIVE	PA	0%
	ES	-7.7%
Big Buck Bunny	PA	0%
	ES	-5.12%

Objective video quality results.

PSNR: the PSNR values, shown in Figures 42, attest to our objective of improving the QoE of end users by employing our scheduling strategy. The results show that CAPA improves the average video PSNR, respectively, by up to **4.25 dB (12.97%)**, and **7.22 dB (20.58%)** compared to PA and ES. This is due to proper load balancing and considering path conditions together with video packet contents to transmit packets and perform packet protection. The differences in measured quality results are caused by the different amount of texture, details, action, etc.

Figure 42 also depicts the original video qualities (PSNRs) together with the average PSNR values measured from the various video sequences according to the different scheduling strategies in order to show how much quality reduction occurs for each scheduling strategy. It can be clearly observed that CAPA has lowest quality reduction compared to PA and ES for all sequences. For instance, for *Meridian* sequence, CAPA is lost 10.07 dB of original quality. This value is increased to 14.32 dB and 17.07 dB for PA and ES, respectively.

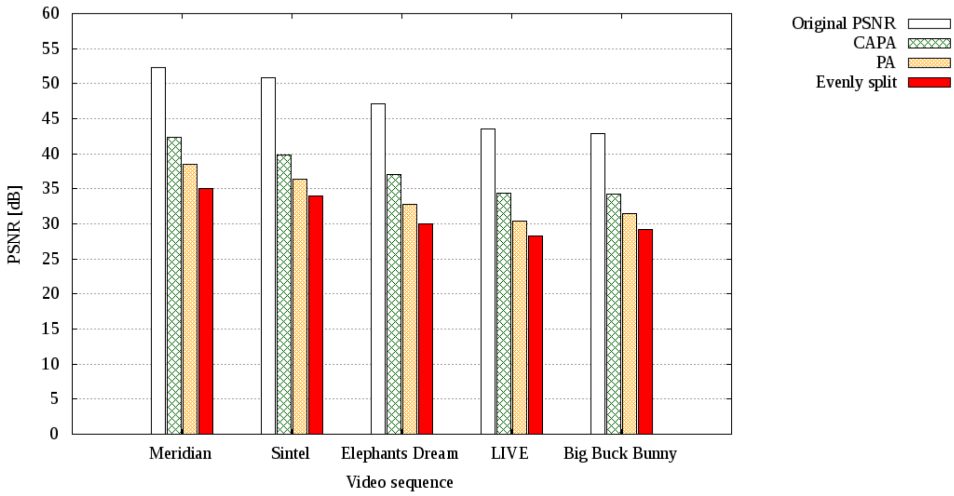


Figure 42 – Comparison of average PSNR results from the various video sequences according to the different scheduling strategies under congested network scenario.

To have a more detail view of the PSNR results, Figure 43 shows the PSNR values from *Elephants Dream* sequence for all video frames. In this figure, the three defined network congestion parts are shown in gray colour in terms of frame number. Therefore, the LTE congestion part, which is from 40 to 120 seconds of simulation time, corresponds to frame numbers from 960 to 2889 of *Elephants Dream* sequence. The WiFi congestion part, from 200 to 400 seconds of simulation time, corresponds to frame numbers from 4814 to 9643, and LTE and WiFi congestion part, from 480 to 600 seconds of simulation time, corresponds to frame numbers from 11561 to 14836. Note that, compared to the original PSNR shown in Figure 43(a), there are PSNR variations for all the three strategies even when there is no network congestion. This is due to having random loss through all the simulation time while FEC is not applied in this work to MMT packets.

One can observe from this figure is that CAPA (Figure 39(b)) achieves apparently higher PSNR values compared to PA (Figure 39(c)) and ES (Figure 39(d)) for all the three network congestion parts. This is due to the higher achieved goodput previously discussed and showed about Figure 39. Another observation from this figure is that while for the first and second part of the simulation, LTE congestion, and WiFi congestion, ES has the lowest values, but in the last part, during the period time of LTE and WiFi congestion, the PSNR values for PA are the lowest, even less than ES. Mainly because of blindly discard strategy applied on the packets by PA, which may discard even high priority packets (I and NI packets).

The PSNR values for all video frames according to different strategies for other tested video sequences are provided in Annex D due to they have the same behaviour of what already explained for *Elephants Dream*.

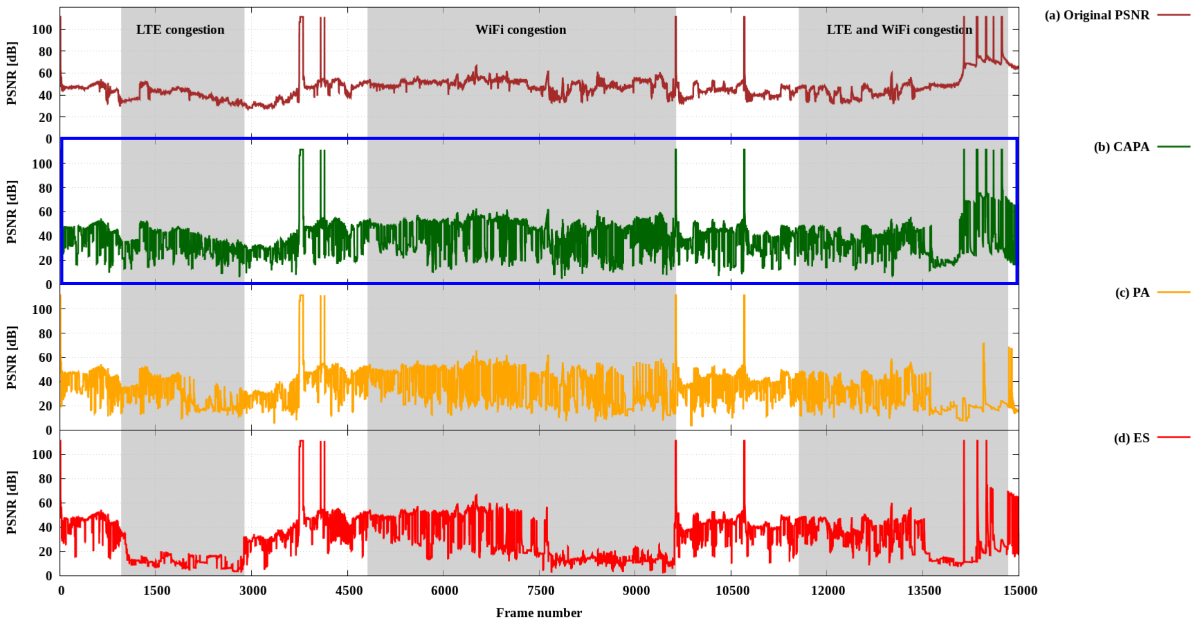


Figure 43 – PSNR values from *Elephants Dream* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

SSIM: besides PSNR, the SSIM values, shown in Figures 44, also attest to our objective of improving the QoE of end users by employing our scheduling strategy. CAPA substantially outperforms other strategies in improving the video SSIM and increases the video SSIM, respectively, by up to **0.033 (3.78%)**, and **0.102 (12.54%)** compared to PA and ES.

In order to reveal the differences in subjective video quality, results from the *Elephants Dream* video sequence according to the different scheduling strategies are shown in Figures 45, 46, and 47, respectively, for LTE congestion, WiFi congestion, LTE and WiFi congestion network situations. Note that it is just one frame to illustrate the type of artifacts after losing packets and decoding the sequence.

In conclusion, the experimental evaluation of this subsection for CAPA strategy shows that our proposed strategy efficiently outperforms PA and ES in improving QoE

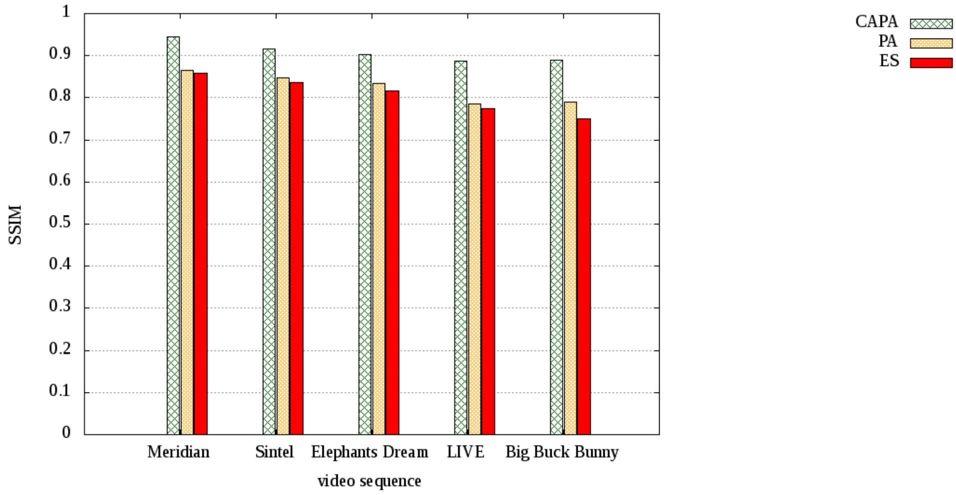


Figure 44 – Comparison of SSIM results from the various video sequences according to the different scheduling strategies under congested network scenario.

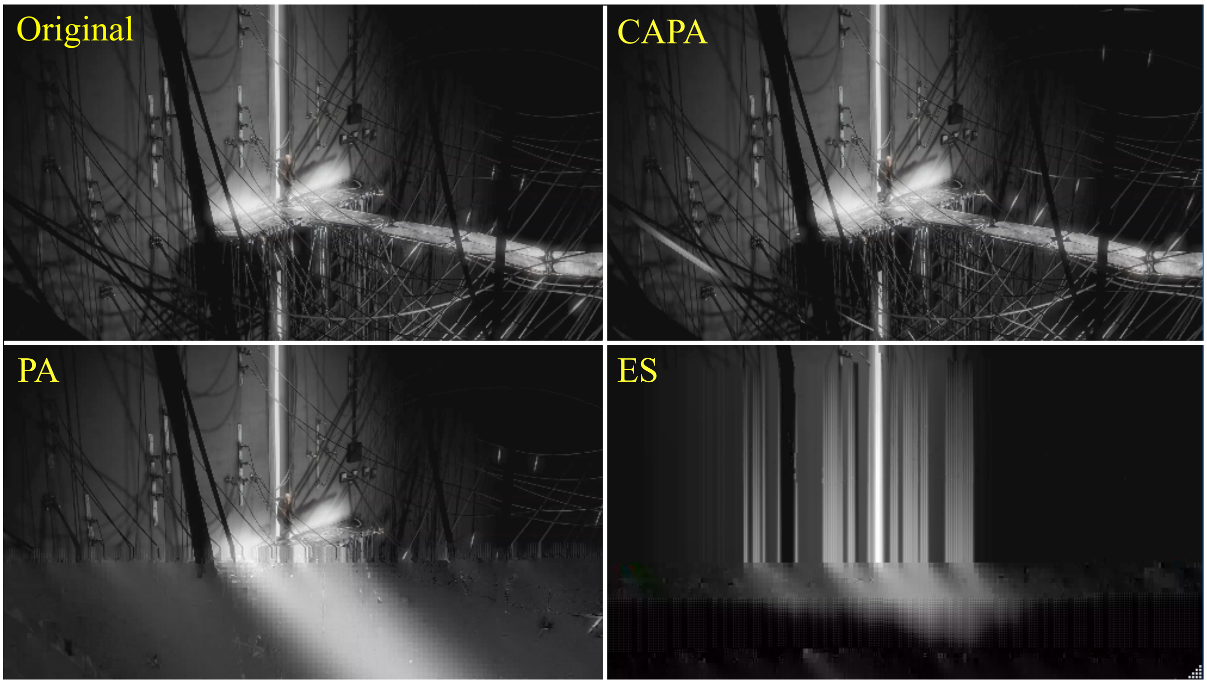


Figure 45 – Comparison of subjective quality measured from the 2051-th frame of the *Elephants Dream* sequence during period time of LTE congestion.

while optimizing the total network goodput under congested network scenario.

4.4.1.3 Experimental Evaluation for Different Video Bit Rates

Here, we have some experiments to check the behaviour of our proposed strategy, CAPA, for different video bit rates in our defined congested network scenario but it is not the propose of this work to determine for each sequence what is the best. Therefore, for this simulation scenario, sequences are encoded with the source bit rates of 3 and 5 Mbps (previously explained in Section 4.2) and packets are distributed on the network, respectively, with the constant transmission bit rates of 3 and 5 Mbps in order to fit fewer

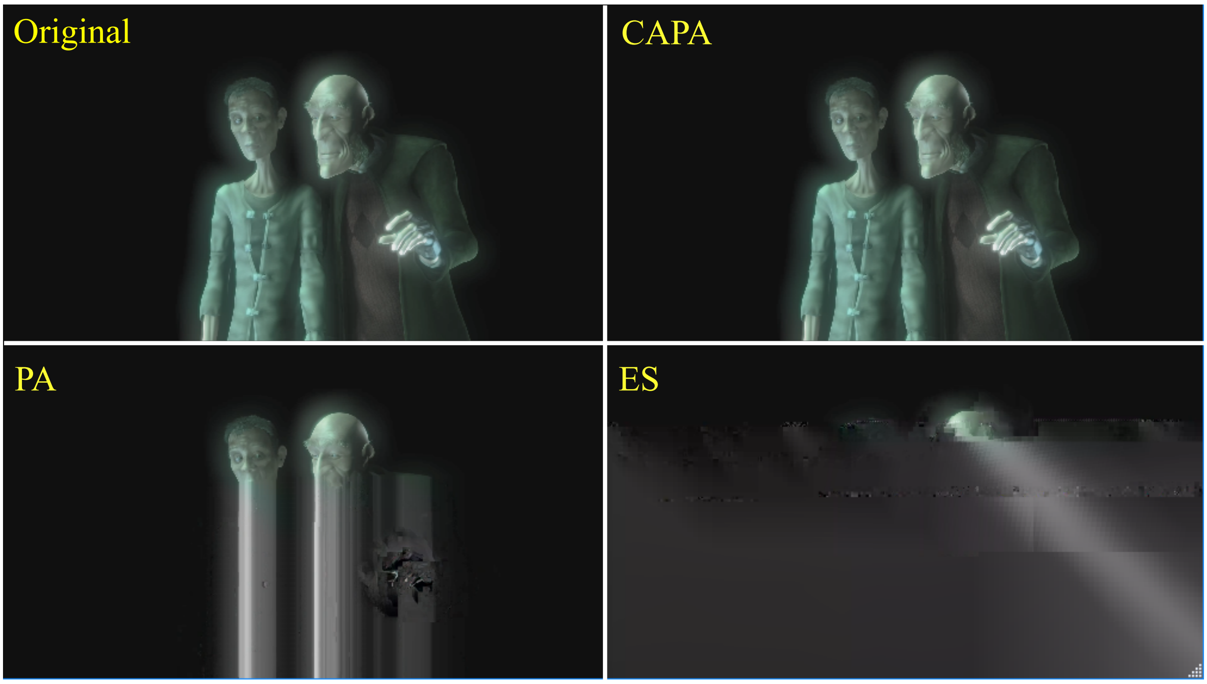


Figure 46 – Comparison of subjective quality measured from the 9138-th frame of the *Elephants Dream* sequence during period time of WiFi congestion.

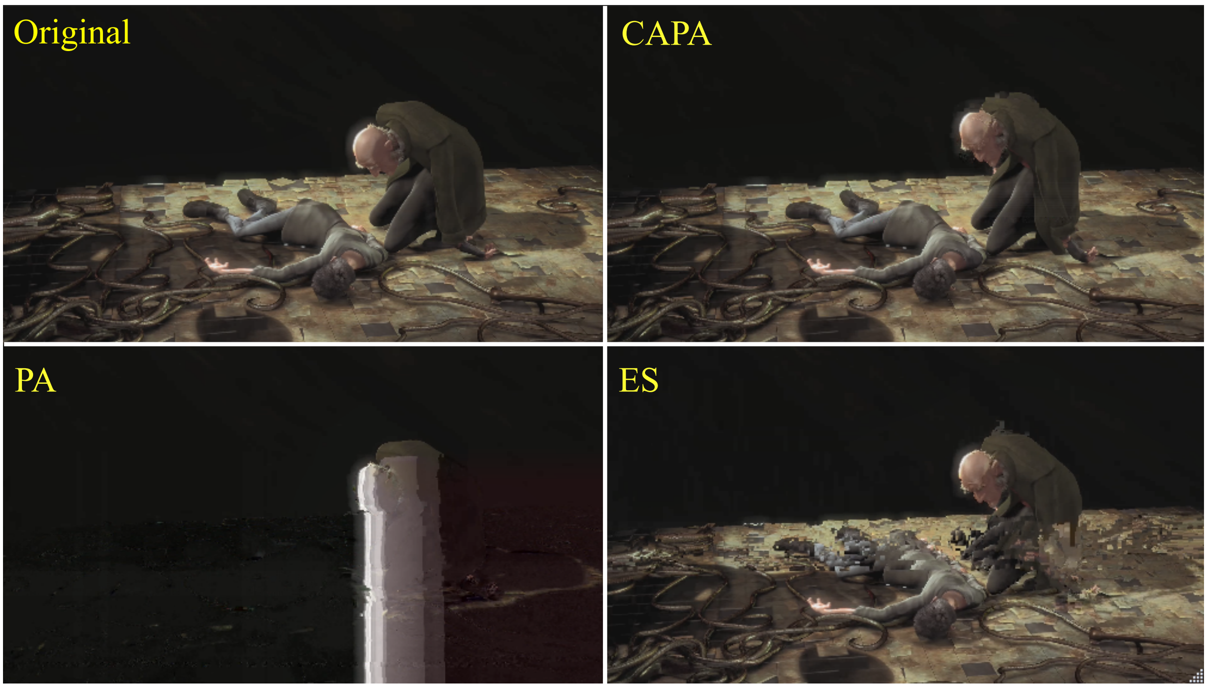


Figure 47 – Comparison of subjective quality measured from the 13518-th frame of the *Elephants Dream* sequence during period time of LTE and WiFi congestion.

packets or more packets in the same simulation time.

Figure 48, similar to what depicted in Figure 38, illustrates performance comparison of different scheduling strategies under the defined congestion network scenario, but differently, it is measured from *Meridian* and *Big Buck Bunny* sequences with different source bit rates (B) of 3 and 5 Mbps.

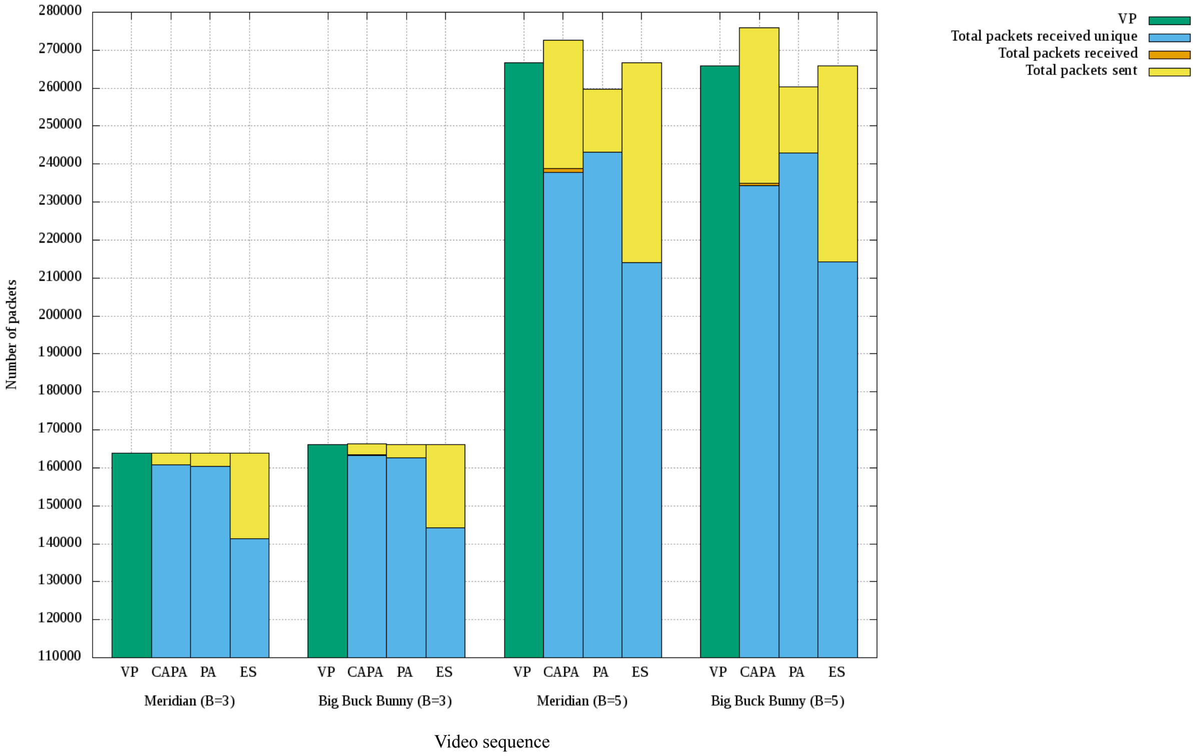


Figure 48 – Performance comparison of *Meridian* and *Big Buck Bunny* video sequences for different bit rates of 3 and 5 Mbps according to the different scheduling strategies under congested network scenario.

One observation from Figure 48 together with Figure 38 is that, as expected, encoded video with higher bit rate has higher number of packets too. For example, while *Meridian* sequence with bit rate of 4 Mbps has 215080 packets (52.35 dB quality), it has 163839 number of packets (51.1 dB quality) with source bit rate of 3 Mbps, corresponds to the 23.82% fewer video packets (and 1.25 dB less quality). Number of packets increases to 266618 (53.34 dB quality) when it has source bit rate of 5 Mbps, which corresponds to the 23.96% more video packets (and 0.99 dB more quality) compared to video with source bit rate of 4 Mbps. Similarly, while *Big Buck Bunny* sequence with source bit rate of 4 Mbps has 215912 packets (42.95 dB quality), it has 166137 number of packets (41.77 dB quality) with source bit rate of 3 Mbps, corresponds to the 23.05% fewer number of video packets (and 1.18 dB less quality). Number of packets increases to 265858 when it has source bit rate of 5 Mbps (44.02 dB quality), which corresponds to the 23.13% more video packets (and 1.07 dB more quality).

The number of packets is what matters because having fewer video packets means sending fewer data through the network, having less congestion, and consequently, fewer losses. In contrast, having more video packets means sending more data through the network, having more congestion, and consequently, more losses which could decrease perceived video quality. Furthermore, since there is less congestion in the network when sequences are distributed with transmission bit rate of 3 Mbps, CAPA duplicates fewer packets for both *Meridian* and *Big Buck Bunny*. Therefore, it is possible to see this fact in the figure that the total packets sent according to CAPA strategy are only slightly higher than the number of video packets. In contrast, due to high congestion for both *Meridian* and *Big Buck Bunny*, when they are distributed with transmission bit rate of 5 Mbps, our proposed scheduler needs to duplicate more packets to protect them. Therefore, the total packets sent according to CAPA strategy, in this case, is notably higher than the number of video packets.

Finally, we observe that, in Figure 48, when sequences are encoded with source bit rate of 3 Mbps, CAPA received more unique packets and outperformed both PA and ES strategies for all sequences, but when sequences are encoded with the source bit rate of 5 Mbps, CAPA could only exceed ES. However, studying QoS and QoE metrics in the rest will show that CAPA could yet outperform PA and achieve our objective of improving QoE.

Network quality of service results.

Goodput: Figure 49 shows goodput performance of different scheduling strategies for *Meridian* sequence with bit rate of 5 Mbps through our network environment under such defined extreme cases of background traffic to emphasizes the scheduler behaviour when transmitting high data. We chose this sequence because it has the highest number of packets (2666618) among our sequences. Generally, the behaviour of strategies obtained for *Meridian* sequence with bit rate of 5 Mbps are very similar to what obtained for *Elephants Dream* with bit rate of 4 Mbps, which we explained previously in Subsection 4.4.1.2 about Figure 39. The difference here is that the network gets more congested due to higher data transmission. However, yet, CAPA outperforms other scheduling strategies and could achieve higher and more stable goodput compared to PA and ES distribution.

The goodput performance according to different strategies for *Meridian* with source bit rate of 3 Mbps and *Big Buck Bunny* with source bit rates of 3 and 5 Mbps are provided in Annex E due to they have the same behaviour of what already explained for *Meridian* with constant transmission bit rate of 5 Mbps.

Packet loss rate: Figure 50, similar to what depicted for Figure 40, shows the comparison of loss results of the different scheduling strategies under congested network scenario, differently, for *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps. Regarding loss results for video with source bit rate of 3, similar to Figure 40, loss results show effectiveness of the proposed scheduling strategy to decrease loss rate in CAPA compared to PA and ES scheduling strategies. The results show that CAPA decreases the total loss rate, respectively, by up to **20.65%**, and **87.22%** compared to PA and ES. There is also better protect of I and NI frame packets according to CAPA compared to PA and ES which is shown in Figure 51. The results show that CAPA decreases the I and NI frame packet loss rate, respectively, by up to **39.60%**, and **91.71%** compared to PA and ES.

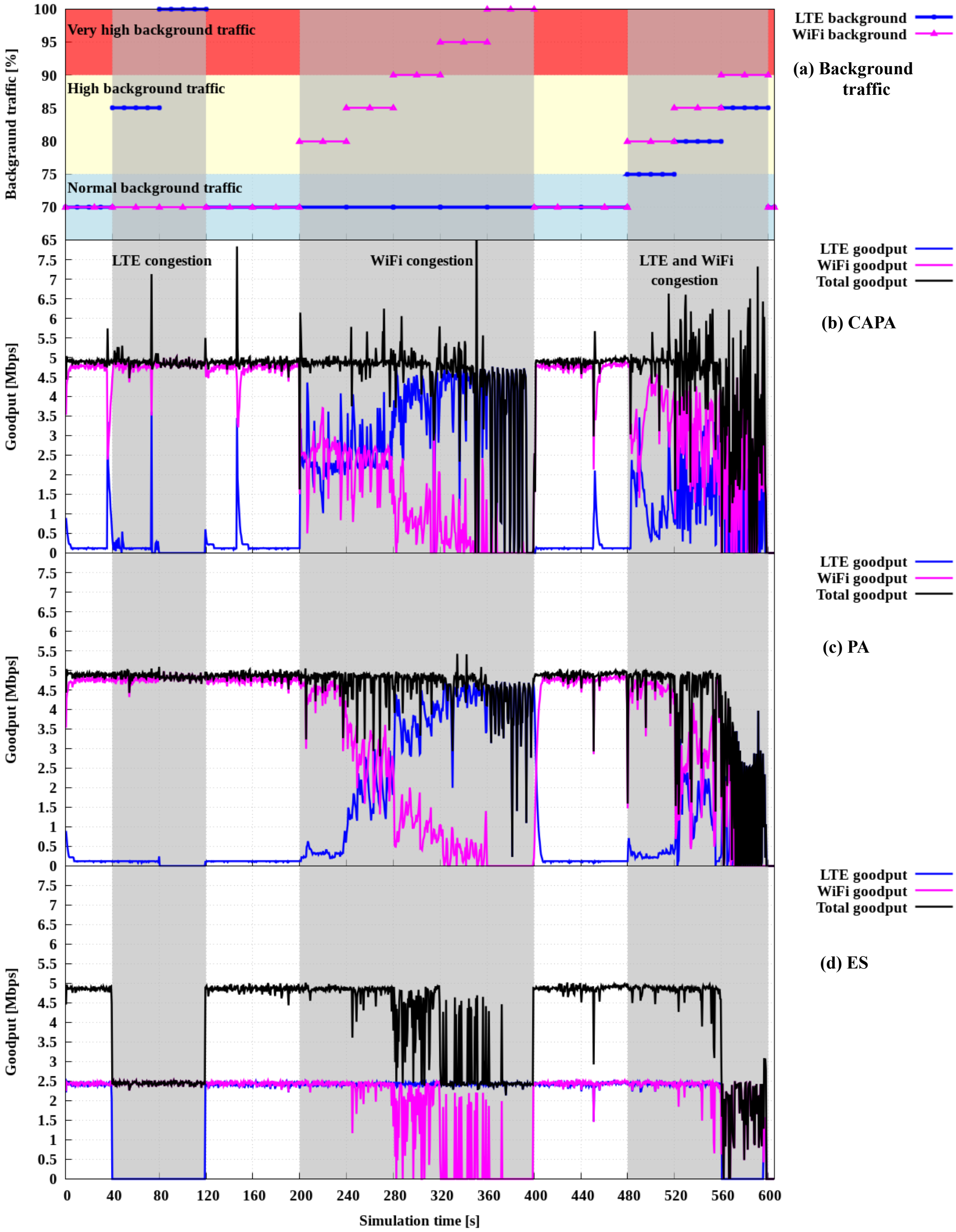


Figure 49 – LTE, WiFi and total (joint) goodput for *Meridian* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under congested network scenario.

Regarding loss results for video transmission with source bit rate of 5, however, while CAPA outperforms ES scheduling strategy decreasing the total loss rate, and I and NI loss rate, respectively, by up to 86.24% and 55.65%, but it is not success to outgo PA strategy, not either in total loss rate (Figure 50) and not in I and NI loss rate (Figure 51). However, it does not mean that CAPA cannot achieve the main objective of our work,

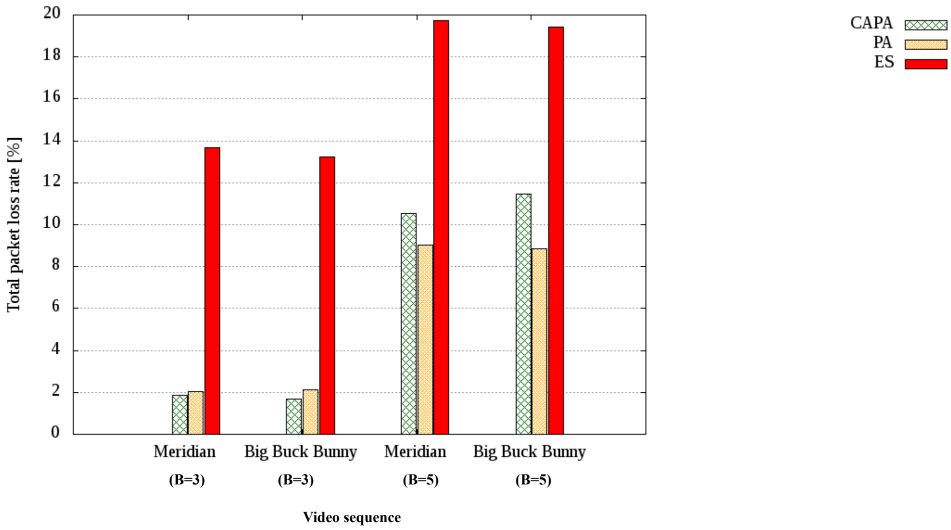


Figure 50 – Comparison of loss results from *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps according to the different scheduling strategies under congested network scenario.

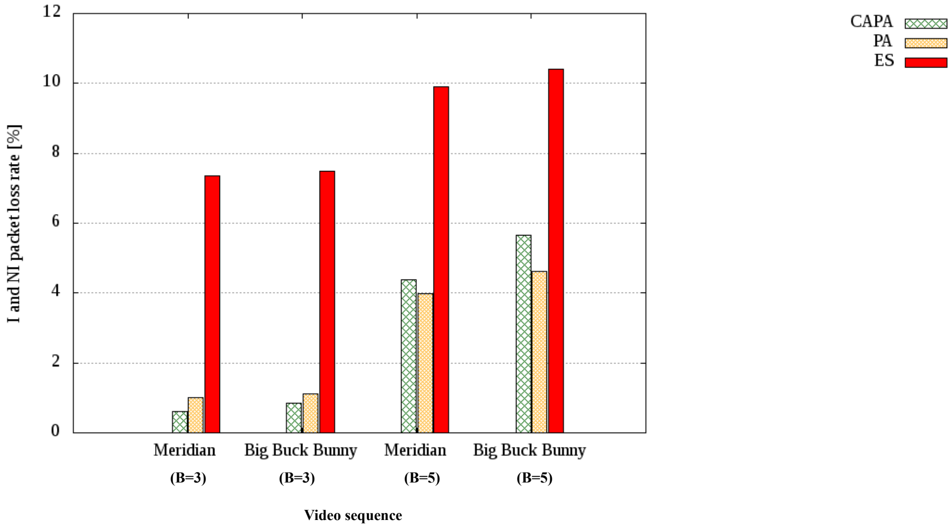


Figure 51 – Comparison of I and NI loss results from from *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps according to the different scheduling strategies under congested network scenario.

which is better perceived video quality than PA. In fact, CAPA has different content-aware strategies to protect high priority packets and based on them, it has the strongest protection case for I frame packets, and after that, for NI frame packets. Therefore, having higher loss rate in this simulation is due to losing more P and NI frame packets. The I frame loss rate for CAPA in this simulation for *Meridian* is 2.2% while this value is 2.5% for PA strategy. Similarly, the I frame loss rate for CAPA in this simulation for *Big Buck Bunny* is 2.22% while this value is 3.13% for PA strategy. Therefore, the content-aware protection method of CAPA could properly protect I frame packets, and consequently, could achieve improvement of video quality, even in such a worse congested network. In the rest, PSNR and SSIM results shown in Figures 52 and 54 attest it.

Yet another note regarding Figures 50 and 51 is that, as expected, sequence with more packets have more losses.

Delay: Table 20, similar to what summarized in Table 19, shows the CAPA average one-way delay reduction of non-overdue packets compared to the alternative scheduling strategies under congested network scenario. Differently, for *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps. Remark that, in this table, the larger the negative values, the performance is better.

Regarding delay results for video with bit rate of 3, the results indicate that our proposed CAPA strategy efficiently enables a better adjustment compared to ES to network conditions by balancing the bit rate distribution and the discard strategy. However, CAPA does not have delay reduction compared to PA.

Regarding delay results for video with bit rate of 5, CAPA is adding some delay over all, but it is delivering I frame packets. For example, for *Meridian*, CAPA reduces I losses 12% and 65.94%, respectively, compared to PA and ES. Similarly, for *Big Buck Bunny*, CAPA reduces I losses 29.07% and 70.43%, respectively, compared to PA and ES.

Therefore, the QoE does not come free. There is always a trade off; sending more traffic due to packet duplication adds more delay because of adding more traffic into the buffers and network entities. Hence, it increases delay with still respective the final deadlines. In other words, at the end what matters for the end user is the perceived video quality. Maybe the network does not like the approach because it causes more packets to be transmitted but for the end user it does. Therefore, it needs to pay delay.

Table 20 – Comparison of CAPA delay results with PA and ES scheduling strategies from the different video sequences with different video bit rates under congested network scenario.

Video sequence	Scheduling strategy	Delay
Meridian (B=3)	PA	0%
	ES	-2.70%
Big Buck Bunny (B=3)	PA	0%
	ES	-0.66%
Meridian (B=5)	PA	+8.27%
	ES	+6.13%
Big Buck Bunny (B=5)	PA	+8.18%
	ES	+6.22%

PSNR: similar to what depicted in Figure 42, the PSNR results shown in Figure 52 also attest to our objective of improving QoE by employing our scheduling strategy. The results show that CAPA improves the average video PSNR for video with bit rate of 3 Mbps, respectively, by up to **3.06 dB (7.61%)**, and **5.45 dB (17.87%)** compared to PA and ES. The results also show that CAPA improves the average video PSNR for video with bit rate of 5 Mbps, respectively, by up to **3.33 dB (9.48%)**, and **7.34 dB (29.30%)** compared to PA and ES.

Another observation in Figure 52 is that, as expected, even though *Big Buck Bunny* with higher encoded bit rate has higher original quality but it loses more quality after transmission compared to when it is encoded in bit rate of 3 Mbps. The same behaviour

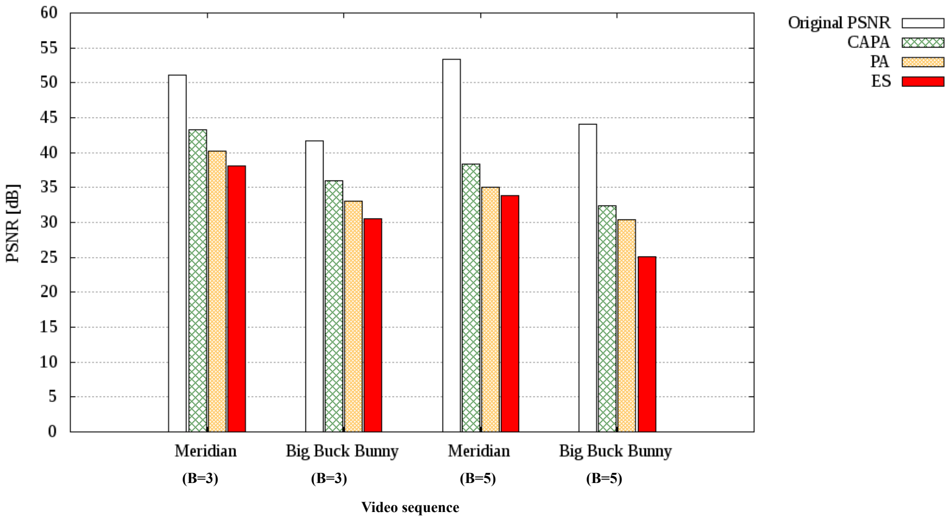


Figure 52 – Comparison of average PSNR results from *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps according to the different scheduling strategies under congested network scenario.

can be noticed for *Meridian*. The reason is that with more packets, more losses occur during data transmission. Remark that the measured quality results always depend on the content and original sequences.

To have a more detail view of the PSNR results, Figure 53 shows the PSNR values from *Meridian* sequence with source bit rate of 5 Mbps for all video frames. In this figure, the LTE congestion part, which is from 40 to 120 seconds of simulation time, corresponds to frame numbers from 971 to 2912 of *Meridian* sequence. The WiFi congestion part, from 200 to 400 seconds of simulation time, corresponds to frame numbers from 4852 to 9722, and LTE and WiFi congestion part, from 480 to 600 seconds of simulation time, corresponds to frame numbers from 11668 to 15000.

The results behaviour obtained for this sequence is very similar to what obtained for the sequence encoded with bit rate of 4 Mbps previously illustrated in Figure 43. This way, CAPA (Figure 53(b)) achieves apparently higher PSNR values compared to the both PA (Figure 53(c)), and ES (Figure 53(d)) for all three network congestion parts. This is due to the higher achieved goodput already discussed and illustrated in Figure 49. Also, while for the LTE congestion part and WiFi congestion part, ES has the lowest values, but in the last part, when both paths are congested, the PSNR values for PA are the lowest (even less than ES). Mainly because of blindly discard strategy applied on the packets by PA which may discard even high priority packets (I and NI packets).

The PSNR values for *Meridian* with source bit rate of 3 Mbps and *Big Buck Bunny* with source bit rates of 3 and 5 Mbps are provided in Annex F due to they have the same behaviour of what already explained for *Meridian* with bit rate of 5 Mbps.

SSIM: besides PSNR, the SSIM values, shown in Figure 54, also attest to our objective

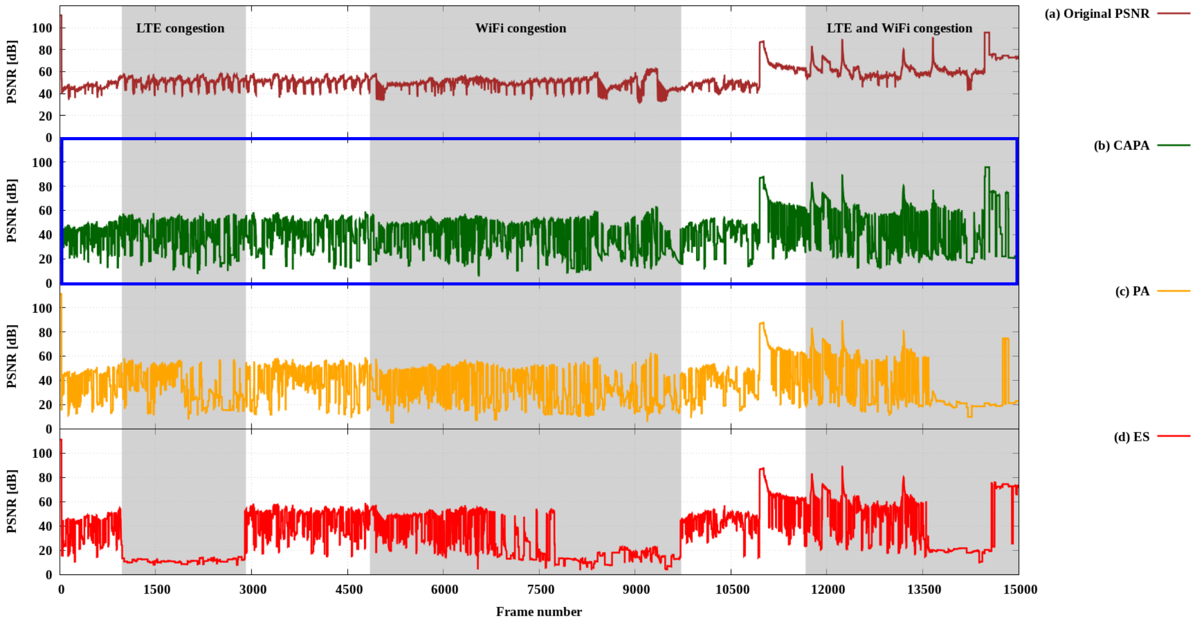


Figure 53 – PSNR values from *Meridian* sequence with bit rate of 5 Mbps according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

of improving the QoE of end users by employing our scheduling strategy. The results show that CAPA improves the video SSIM for video with bit rate of 3, respectively, by up to **0.030 (3.367%)**, and **0.091 (10.96%)** compared to PA and ES. The results also show that CAPA improves the video SSIM for video with bit rate of 5, respectively, by up to **0.019 (2.2%)**, and **0.080 (9.96%)** compared to PA and ES.

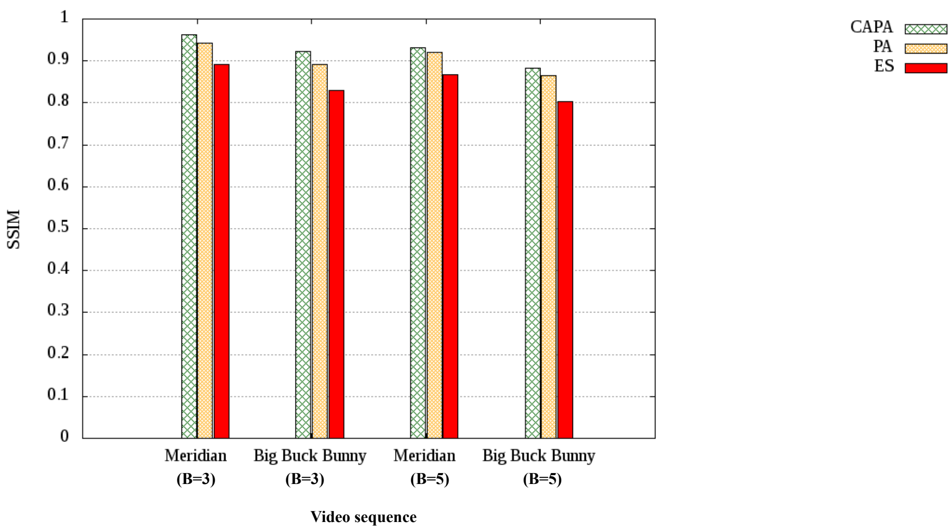


Figure 54 – Comparison of SSIM results from *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps according to the different scheduling strategies under congested network scenario.

In order to reveal the differences in subjective video quality, Figures 55, 56, and 57 show a sample decoded frame with losses impact from the *Meridian* sequence with source bit rate of 5 Mbps according to the different scheduling strategies, respectively, during period time of LTE congestion, WiFi congestion, LTE and WiFi congestion. Remark that it is just one frame to illustrate the type of artifacts after losing packets and decoding the sequence.

Figure 55 – Comparison of subjective quality measured from the 2078-th frame of the *Meridian* sequence during period time of LTE congestion.

In conclusion, experimental results provided in this scenario for different bit rates of video sequences under congested network scenario show that CAPA perfectly outperforms PA and ES to achieve the main objective of PSNR and SSIM for different video bit rates. Furthermore, results in Figures 52 and 54 for video with source bit rate of 5 Mbps (high data) show that not only CAPA significantly outperforms PA and ES scheduling strategies in improving the video PSNR and SSIM, but also, regarding Table 18, it can achieve PSNR and SSIM, respectively, 38.42 dB and 0.93 for *Meridian*, maps to *"excellent"* MOS quality, PSNR and SSIM gain of, respectively, 32.39 dB and 0.883 for *Big Buck Bunny*, maps to *"good"* MOS quality, even in our environment with such defined extreme cases of background traffic. Therefore, CAPA could even handle high data efficiently compared to PA and ES.

Table 21 summarizes the results for *Meridian* and *Big Buck Bunny* sequences with source bit rates of 3, 4 and 5 Mbps. One can see is that increasing the original quality, in our simulation scenario, cannot improve the perceived quality. This is due to the fact that higher quality has more packets, and consequently, increases losses due to higher congestion in our environment with the defined extreme background traffic. Therefore,

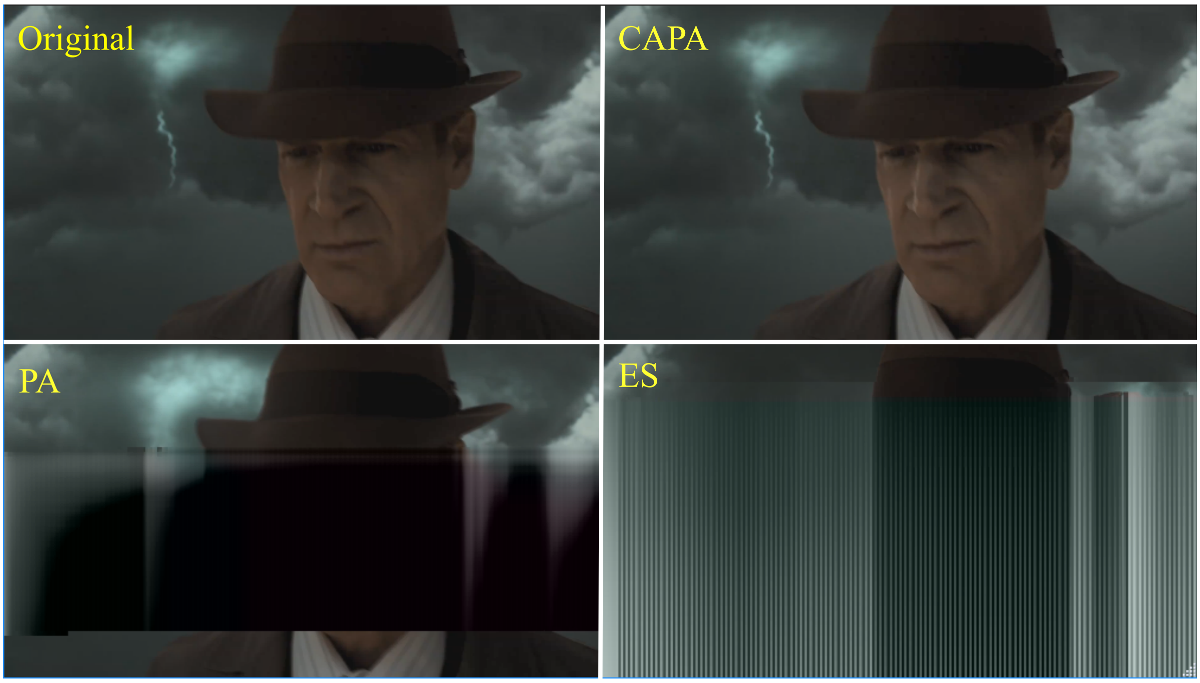


Figure 56 – Comparison of subjective quality measured from the 9152-th frame of the *Meridian* sequence during period time of WiFi congestion.

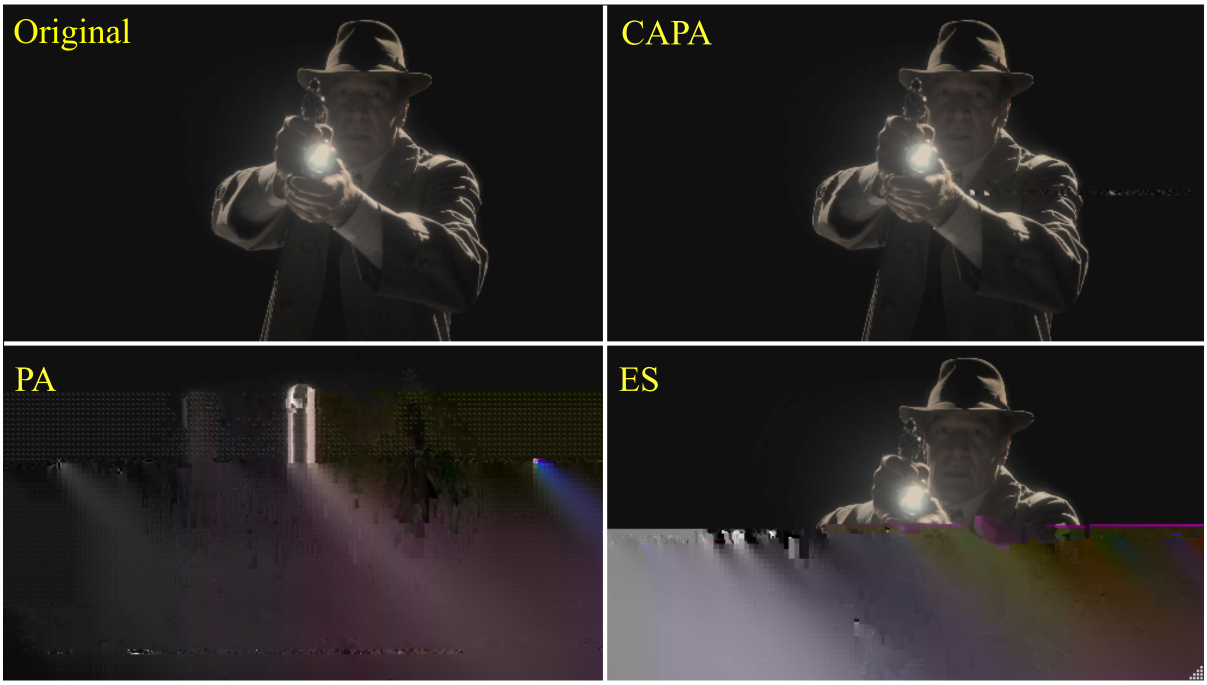


Figure 57 – Comparison of subjective quality measured from the 13492-th frame of the *Meridian* sequence during period time of LTE and WiFi congestion.

even if the original quality is higher but more quality would be lost. Besides that, the original quality here is important because after adding 1 Mbps, the quality does not change too much. Finally, when the original PSNRs (qualities) are good enough, there is no reason to go higher with the initial quality.

Table 21 – Comparison of original PSNR results with perceived video PSNR results from *Meridian* and *Big Buck Bunny* video sequences with bit rates of 3 and 5 Mbps according to CAPA.

Video sequence	Source bit rate [Mbps]	Original PSNR [dB]	PSNR result [dB]
Meridian	3	51.1	43.25
	4	52.35	42.28
	5	53.34	38.42
Big Buck Bunny	3	41.77	35.94
	4	42.95	34.25
	5	44.02	32.39

4.4.2 Wireless Lossy Network Scenario

In this subsection, we discuss the performance of our proposed CAPA in a wireless lossy network situation. Therefore, in this scenario, while the background traffic is kept constant in the whole simulation time, different wireless burst loss conditions are defined for paths to observe the response of the scheduling strategy. Background traffic and wireless loss conditions are detailed in Subsection 4.4.2.1. Then, we discuss the performance of our proposed scheduler under the defined wireless lossy network scenario in Subsection 4.4.2.2 in terms of both QoS and QoE metrics.

4.4.2.1 Background Traffic and Wireless Loss Condition

In this scenario, constant downlink and uplink background traffics are added in the ns-3 simulation. The downlink background traffic is generated by the server and, as illustrated in Figure 58, set as 70% of full link capacity for both paths. On the other hand, the uplink background traffic is generated by the network nodes and set as 10% of each full link capacity, in accordance to real network scenarios where the uplink traffic is smaller than the downlink traffic.

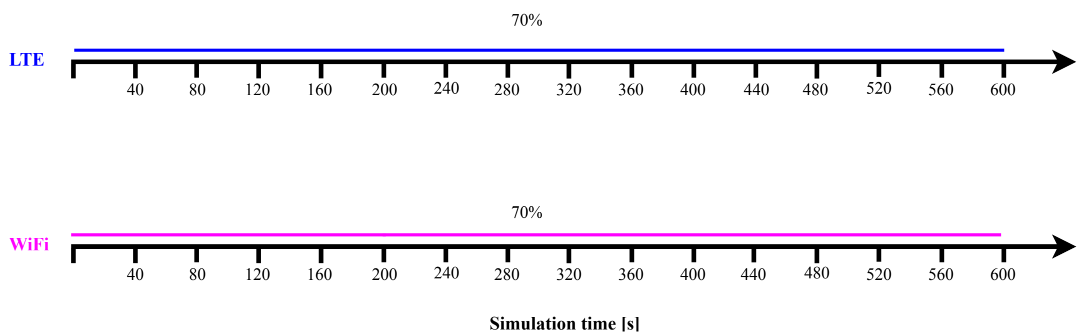


Figure 58 – Constant downlink background traffic setup.

This explained background traffic is also depicted as a plot in Figure 59. This plot would be used in the rest of this wireless lossy network scenario together with goodput and PSNR graphs in order to facilitate readers to track the network condition and understand graphs explanation easily. One can note regarding this proposed background traffic is that both LTE and WiFi background traffics always have normal background traffic in this network scenario.

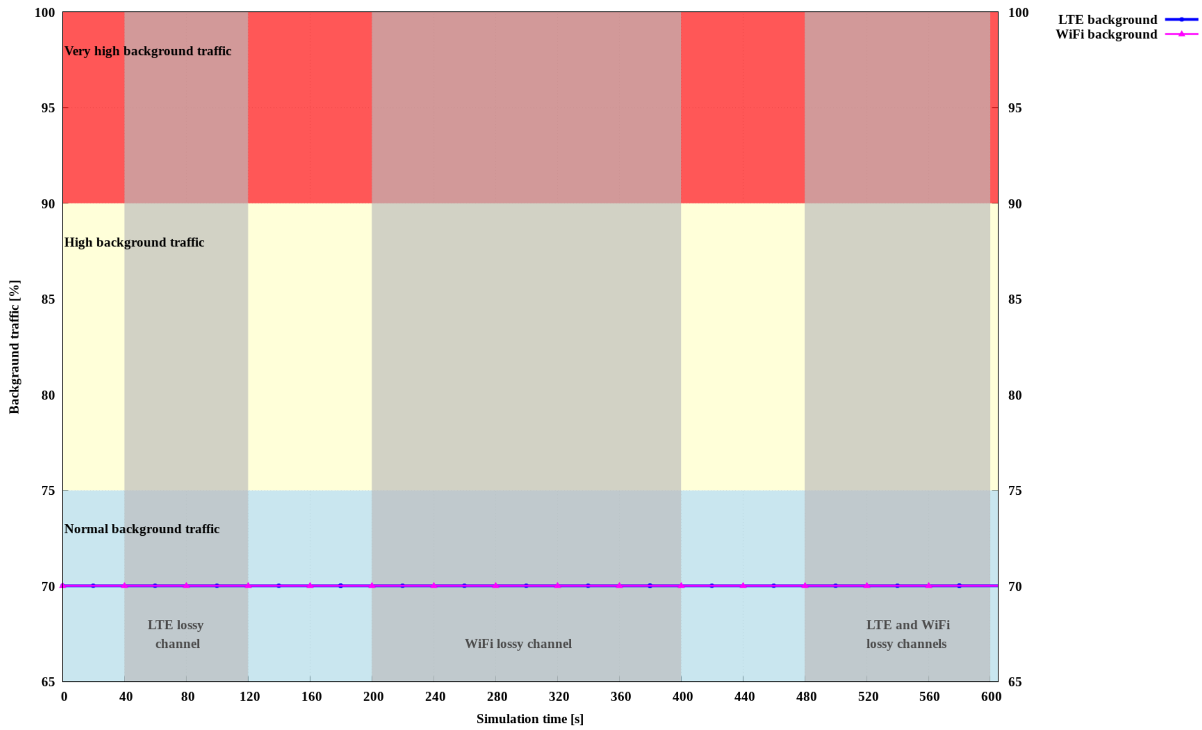


Figure 59 – Constant downlink background traffic setup showing in a plot form.

Wireless loss condition. In this scenario, the ns-3 burst error model was employed to capture the effects of burst packet losses caused due to wireless error channels in the network in order to observe the response of the scheduling strategy. Related work (WU *et al.*, 2013) assumes loss rates of 5% for the WiFi and 2% for the LTE path. However, for most of our simulations and most of our sequences, the losses for WiFi and LTE were too high (cf. Annex B). Therefore, we kept the proportion between the loss rates of WiFi and LTE but reduced to 1% for WiFi, and 0.4% for LTE.

We have defined three wireless loss parts in our simulation; LTE lossy channel, WiFi lossy channel, LTE and WiFi lossy channels. These three parts are shown in gray color in Figure 59 and are defined in the same periods set in Subsection 4.4.1.1 for congested network scenario. Therefore, the first part of the simulation, LTE lossy channel, is from 40 to 120 seconds of simulation time. During this period, the LTE channel has a burst loss rate of 0.4% while the WiFi has no burst loss. In the second part, WiFi lossy channel, the opposite behaviour is simulated and, from 200 to 400 seconds of simulation time, WiFi channel has burst loss rate of 1% while LTE has no burst loss. Finally, in the last part, from 480 to 600 seconds of simulation time, both LTE and WiFi have burst losses, respectively, 0.4% and 1% to simulate simultaneous wireless burst losses in LTE and WiFi.

4.4.2.2 Experimental Evaluation for Constant Transmission Bit Rate

Here, we discuss the performance of our proposed scheduler under the defined wireless loss network scenario. Similar to the congested network scenario explained in Subsection 4.4.1.2, all sequences are encoded with the same source bit rate of 4 Mbps, previously explained in Section 4.2, and are transmitted with constant bit rate of 4 Mbps.

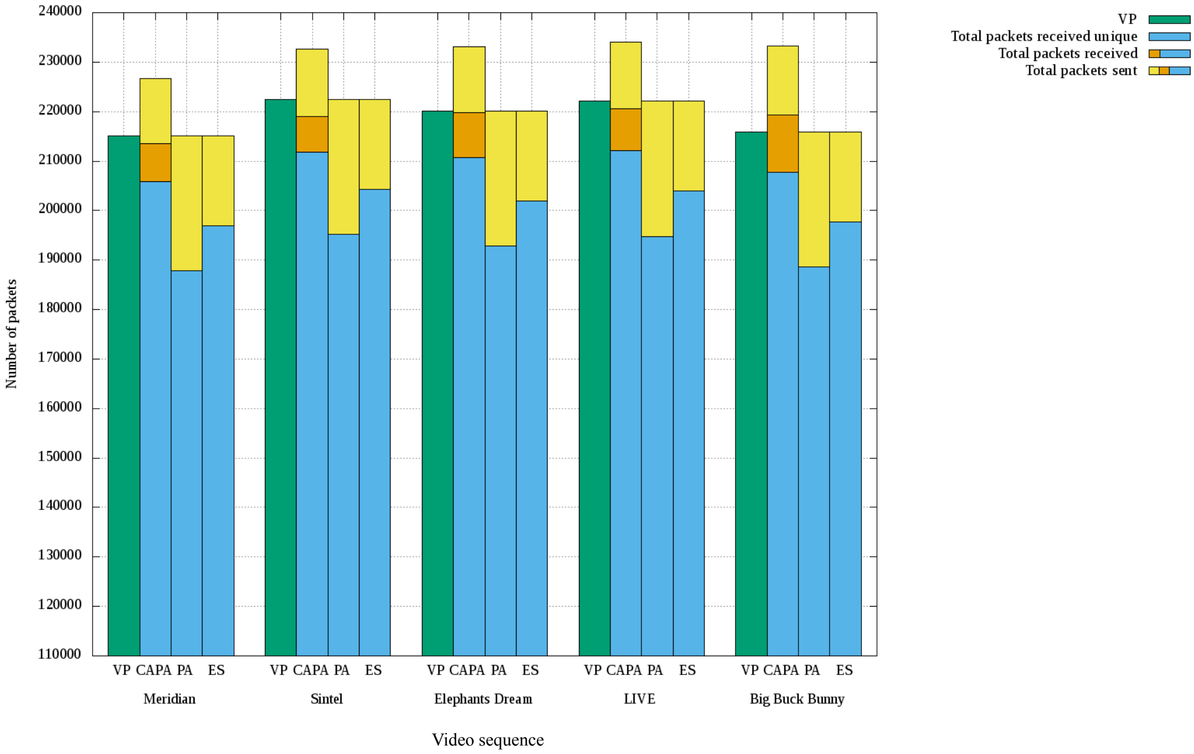


Figure 60 – Performance comparison of the various video sequences according to the different scheduling strategies under wireless lossy network scenario.

Figure 60, similar to what depicted in Figure 38, shows performance comparison of the various video sequences according to the different scheduling strategies, but differently from Figure 38, it is under wireless lossy scenario. Similarly, in this figure, more packets received unique by CAPA strategy compared to both PA and ES strategies for all sequences. Besides, the total number of packets sent according to CAPA strategy is higher than the number of video packets for all sequences due to high amount duplication. However, generally, no packet is discarded under lossy network scenario. Also, PA does not apply discard strategy under wireless lossy networks since it has the same discard strategy as CAPA. Therefore, the number of sent packets are equal to the number of video packets. PA also received less packets compared to ES strategy. This is mainly because PA has higher losses when WiFi is lossy compared to this part in ES (further will be explained in more details in explanation of Figure 61). Pointing out this again that content of video always effects on the performance and the amount of duplication.

Network quality of service results.

Goodput: Figure 61, similar to what depicted in Figure 39, shows the LTE, WiFi and total (joint) goodput for *Elephants Dream* video sequence according to the different scheduling strategies; (b) CAPA, (c) PA, and (b) ES. But, differently from Figure 39, it is under wireless lossy scenario. In order to facilitate readers tracking the network condition, the defined background traffic, previously explained in Subsection 4.4.2.1, is also shown in Figure 61(a) top of the goodput subfigures.

Figure 61(b) illustrates that the total achieved goodput is higher and more stable when our proposed scheduling strategy is applied compared to PA and ES distribution. One can note that load balancing is clearly achieved between both paths and the higher inherent capacity of the WiFi path is exploited. In LTE lossy channel part, video packets are properly handled by the scheduler by switching traffic among paths and keeping a stable total goodput of 4 Mbps. In WiFi lossy part, the situation for WiFi channel is so much worse than the situation for LTE in the first lossy part of simulation due to the much higher burst wireless loss rate. Therefore, in this WiFi lossy part, CAPA duplicates I and NI packets to ensure reliable delivery. Similarly, when both channels are lossy, the content-aware packet protection method protects these high priority packets by duplication.

Figure 61(c) illustrates the total achieved goodput according to PA scheduling strategy. Although PA can keep a stable total goodput of 4 Mbps for the first part of LTE lossy channel, in part of WiFi lossy channel, and in part of LTE and WiFi lossy channels, it is losing too many packets, and consequently, it has big goodput reduction. The reason is that PA does not have content-aware method and packet duplication compared to CAPA.

One can observe from Figure 61(d) is that, at the initial part of simulation, where channels are not lossy, video traffic is equally split between LTE and WiFi. Therefore, each channel has a goodput of 2 Mbps and the total goodput is of 4 Mbps. Then, when LTE becomes lossy, goodput for LTE decreases slightly due to the low burst loss rate. In the second part of the simulation, LTE recovers from burst wireless losses and then WiFi becomes lossy with higher bit rate than LTE at the first lossy part of the simulation. Therefore, there is more goodput reduction due to more losses. In the last part, where both channels are lossy, both paths lose packets and the total goodput is decreased more than two last lossy situation parts of the network. One can observe from this figure compared to 61(c) is that PA is losing more packets in part of WiFi lossy channel simulation due to higher data rate transmission on WiFi in PA strategy.

The goodput performance according to different strategies for other sequences is provided in Annex G due to they have the same behaviour of what already explained for *Elephants Dream*.

Packet loss rate: Figure 62, similar to what depicted in Figure 40, shows the effectiveness

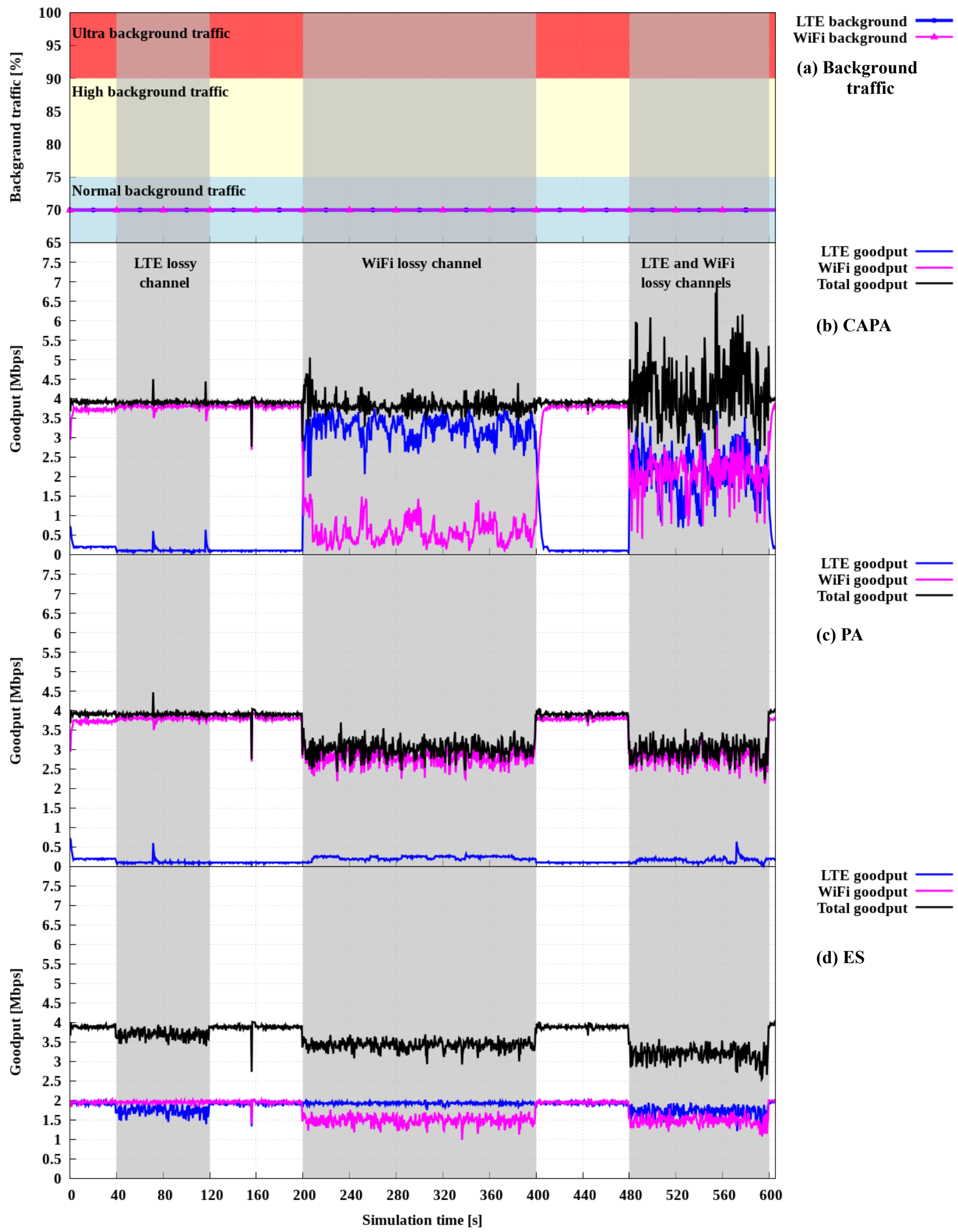


Figure 61 – LTE, WiFi and total (joint) goodput for *Elephants Dream* packets for (a) the constant background traffic with burst wireless loss condition according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under wireless lossy network scenario.

of the proposed scheduling strategy to decrease the loss rate in CAPA compared to PA and ES scheduling strategies. The results show that CAPA reduces the total loss rate, respectively, by up to **72.09%**, and **58.2%** compared to PA and ES. There is also better protect of I and NI frame packets according to CAPA compared to PA and ES, which is shown in Figure 63 similar to what depicted in Figure 41. The results show that CAPA decreases the I and NI frame packet loss rate, respectively, by up to **86.35%**, and **79.65%** compared to PA and ES.

In both Figures 62 and 63, however, PA has higher losses and also higher I and NI losses compared to ES. This is mainly because PA has higher losses when WiFi is lossy compared to this part in ES. Goodput achievement of PA previously illustrated in Figure 61(c) attests these higher losses and lower goodput of PA compared to ES when WiFi is lossy. Yet another point observing from Figure 63 is that due to differences of the video sequences previously mentioned and shown in Table 16, *Big Buck Bunny*, the one with the highest percentage of I ad NI frame packets, also has the most I and NI packet losses in all compared conditions.

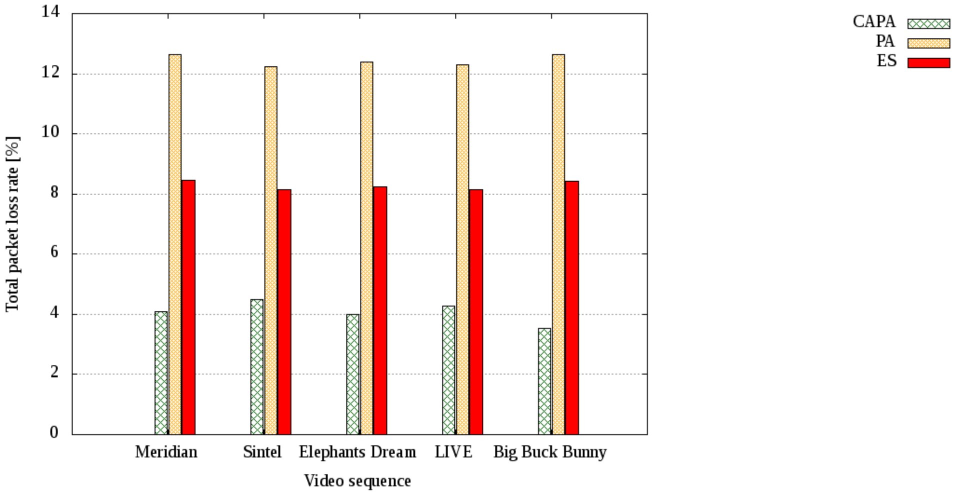


Figure 62 – Comparison of loss results from the various video sequences according to the different scheduling strategies under wireless lossy network scenario.

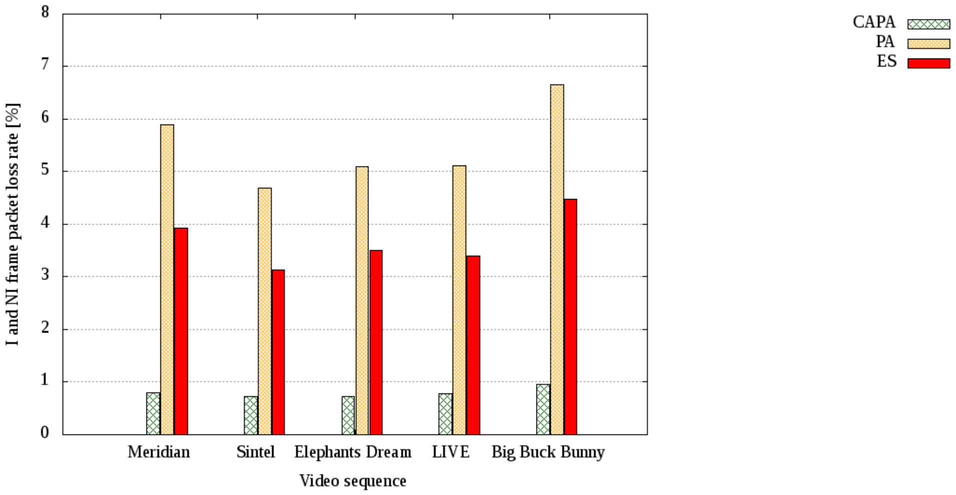


Figure 63 – Comparison of I and NI loss results from the various video sequences according to the different scheduling strategies under wireless lossy network scenario.

Having Figure 62 together with Figure 63 shows the proposed scheduling strategy has better protect of I and NI frame packets. For all sequences, the I and NI frame packet loss rate over the total packet loss rate decreases. For instance, for *Meridian*, according to our proposed scheduling strategy, total lost packets are 4.09%, and 0.80% of these losses are I and NI losses, which corresponds to 19.55% of the total losses. When PA scheduling strategy is applied, total lost packets are 12.65%, and 5.86% of these losses are I and NI losses, which corresponds to 46.32% of the total losses. When ES scheduling strategy is applied, total lost packets are 8.47%, and 3.93% of these losses are I and NI losses, which corresponds to 46.39% of the total losses. Therefore, the rate is reduced from 46.32% to 19.55% compared to PA, and from 46.39% to 19.55% compared to ES.

Similarly for *Sintel*, according to our proposed scheduling strategy, total lost packets are 4.5%, and 0.72% of these losses are I and NI losses, which corresponds to 16% of the total losses. When PA scheduling strategy is applied, total lost packets are 12.23%, and 4.69% of these losses are I and NI losses, which corresponds to 38.34% of the total losses. When ES scheduling strategy is applied, total lost packets are 8.14%, and 3.13% of these losses are I and NI losses, which corresponds to 38.45% of the total losses. Therefore, the rate is reduced from 38.34% to 16% compared to PA, and from 38.45% to 16% compared to ES.

Also for *Elephants Dream*, according to our proposed scheduling strategy, total lost packets are 4%, and 0.72% of these losses are I and NI losses, which corresponds to 18% of the total losses. When PA scheduling strategy is applied, total lost packets are 12.39%, and 5.09% of these losses are I and NI losses, which corresponds to 41.08% of the total losses. When ES scheduling strategy is applied, total lost packets are 8.23%, and 3.50% of these losses are I and NI losses, which corresponds to 42.52% of the total losses. Therefore, the rate is reduced from 41.08% to 18% compared to PA, and from 42.52% to 18% compared to ES.

Likewise for *LIVE*, according to our proposed scheduling strategy, total lost packets are 4.27%, and 0.78% of these losses are I and NI losses, which corresponds to 18.26% of the total losses. When PA scheduling strategy is applied, total lost packets are 12.31%, and 5.11% of these losses are I and NI losses, which corresponds to 41.51% of the total losses. When ES scheduling strategy is applied, total lost packets are 8.16%, and 3.39% of these losses are I and NI losses, which corresponds to 41.54% of the total losses. Therefore, the rate is reduced from 41.51% to 18.26% compared to PA, and from 41.54% to 18.26% compared to ES.

Finally for *Big Buck Bunny*, according to our proposed scheduling strategy, total lost packets are 3.52%, and 0.95% of these losses are I and NI losses, which corresponds to 26.98% of the total losses. When PA scheduling strategy is applied, total lost packets are 12.61%, and 6.66% of these losses are I and NI losses, which corresponds to 52.81% of the total losses. When ES scheduling strategy is applied, total lost packets are 8.42%, and 4.48% of these losses are I and NI losses, which corresponds to 53.20% of the total losses. Therefore, the rate is reduced from 52.81% to 26.98% compared to PA, and from

53.20% to 26.98% compared to ES.

Delay: in this scenario, since the background traffic is constant and in normal level during all simulation time, delay for most of our simulations and most of our sequence is almost the same.

PSNR: similar to what depicted in Figure 42, the PSNR results shown in Figures 64 also attest to our objective of improving the QoE of end users by employing our scheduling strategy, this time under lossy network scenario. The results show that CAPA improves the average video PSNR, respectively, by up to **6.84 dB (20.30%)**, and **9.43 dB (30.32%)** compared to PA and ES. This is mainly because of the proposed content-aware strategy and packet duplication.

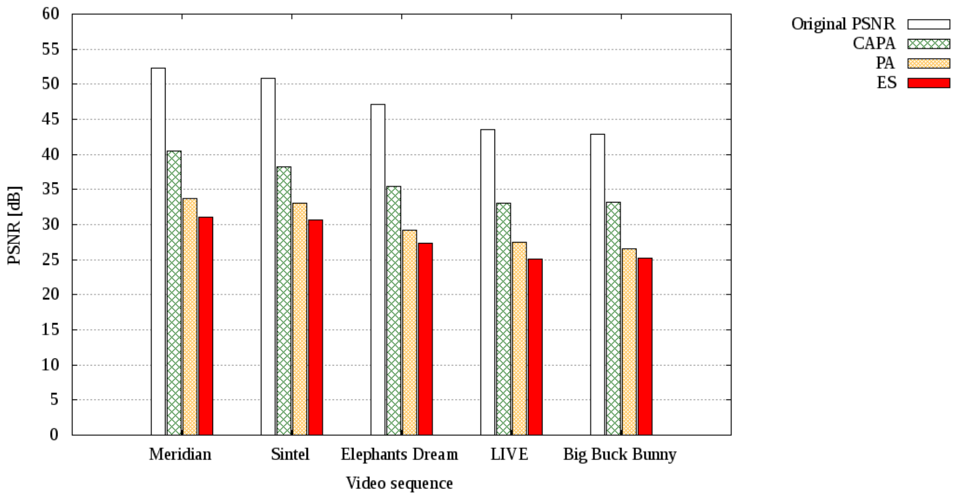


Figure 64 – Comparison of average PSNR results from the various video sequences according to the different scheduling strategies under wireless lossy network scenario.

Similar to what illustrated in Figure 43, Figure 65 shows the PSNR values from *Elephants Dream* sequence for all video frames, but differently, under wireless lossy network scenario. One can observe from this figure, comparing CAPA with PA, is that the PSNR values for CAPA (Figure 65(b)) at the first lossy part, when LTE is lossy, is almost the same as PSNR values of PA, but at the second and third lossy parts, PSNR values for CAPA are significantly higher than those values of PA. This is due to the higher achieved goodput discussed before in goodput graph illustrated in Figure 61. PA could only achieve good PSNR values when LTE is lossy, and it almost completely degrading for the second and third lossy parts. As we previously also discussed, the situation for WiFi when it is lossy is much worse than when LTE is lossy. Therefore, by adding the content-aware strategy, our proposed scheduler could achieve much better quality.

Another observation from this figure, comparing CAPA with ES, is that CAPA (Figure 65(b)) achieves higher PSNR values compared to the ES (Figure 65(d)) for all three wireless lossy network parts. The reason is that ES lacks any sufficient scheduling strategy to handle lossy network situation.

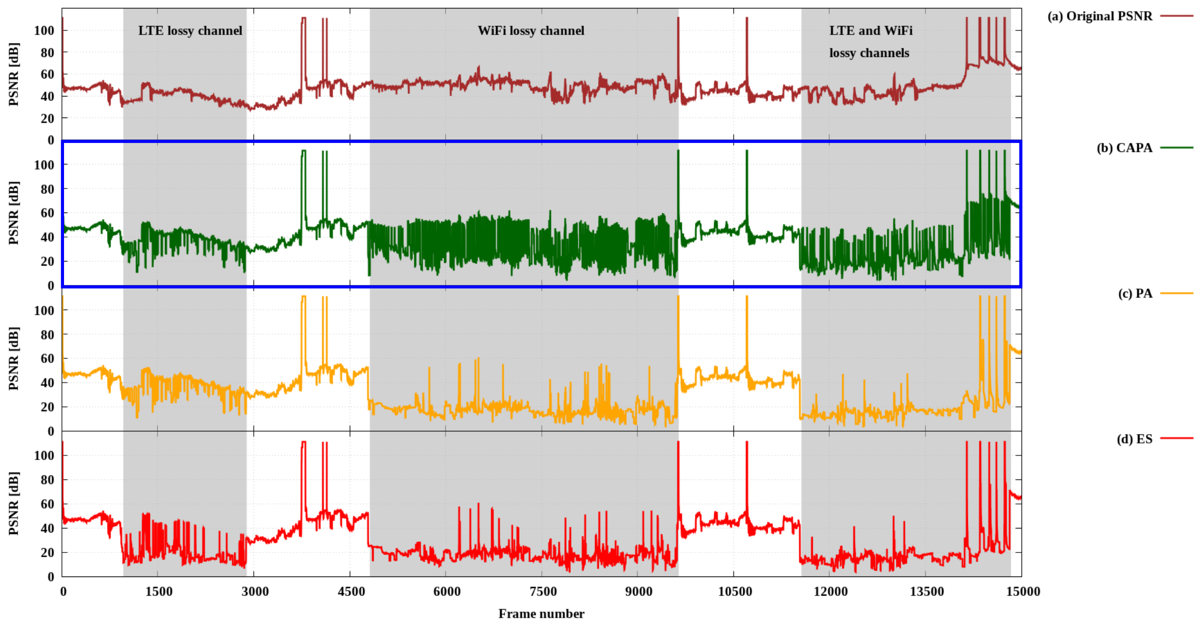


Figure 65 – PSNR values from *Elephants Dream* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under wireless lossy network scenario.

The PSNR values for all video frames according to different strategies for other tested video sequences are provided in Annex H due to they have the same behaviour of what already explained for *Elephants Dream*.

SSIM: besides PSNR, the SSIM values, shown in Figures 66, also attest to our objective of improving the QoE of end users by employing our scheduling strategy. CAPA substantially outperforms other strategies in improving the video SSIM and increases the video SSIM, respectively, by up to **0.100 (12.72%)**, and **0.113 (14.23%)** compared to PA and ES.

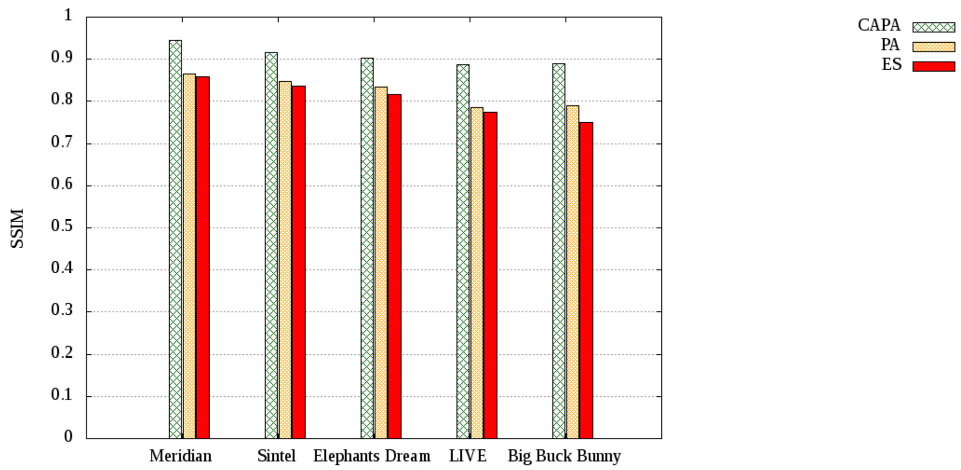


Figure 66 – Comparison of SSIM results from the various video sequences according to the different scheduling strategies under wireless lossy network scenario.

In order to reveal the differences in subjective video quality, typical results measured from the *Elephants Dream* video sequence according to the different scheduling strategies are shown in Figures 67, 68, and 69, respectively, for LTE lossy channel, WiFi lossy channel, LTE and WiFi lossy channels network situations. Note that it is just one frame to illustrate the type of artifacts after losing packets and decoding the sequence.

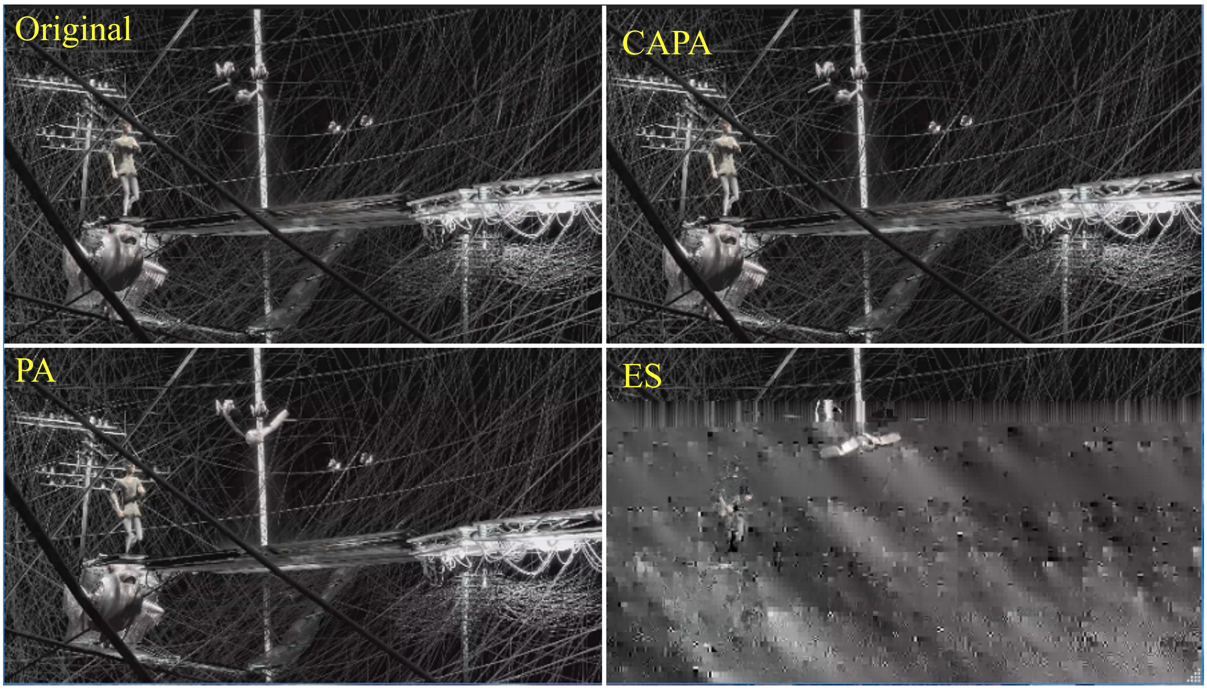


Figure 67 – Comparison of subjective quality measured from the 2750-th frame of the *Elephants Dream* sequence when LTE channel is lossy.

Figure 68 – Comparison of subjective quality measured from the 7509-th frame of the *Elephants Dream* sequence when WiFi channel is lossy.

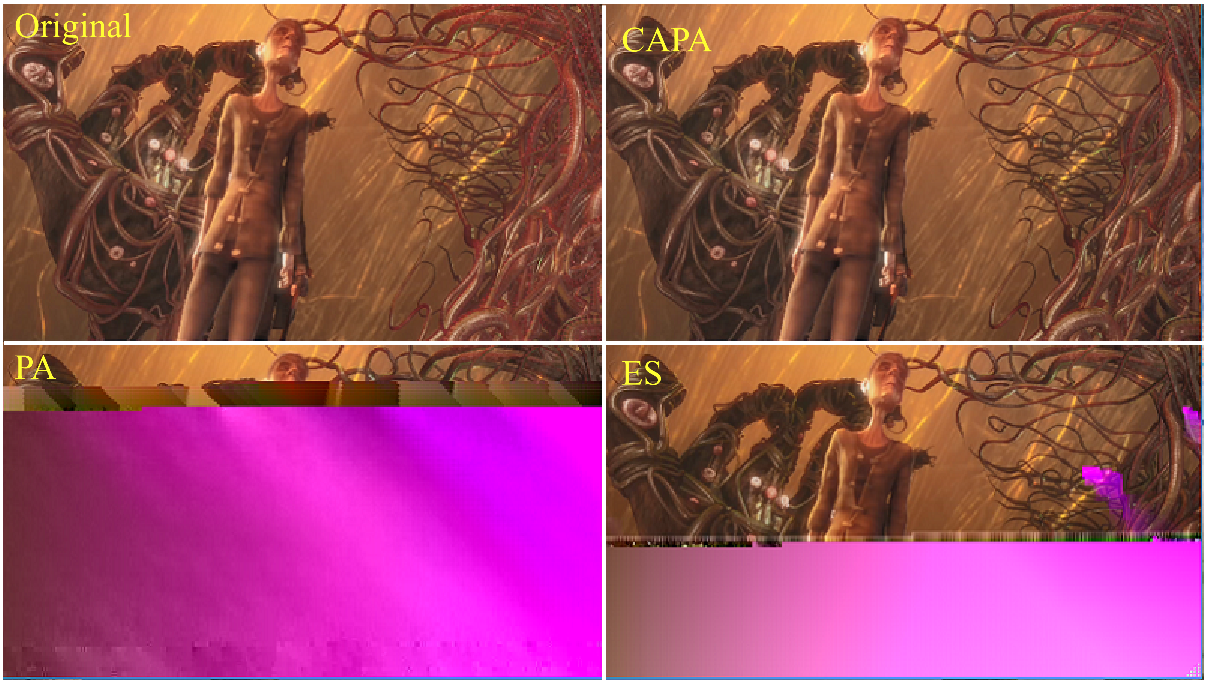


Figure 69 – Comparison of subjective quality measured from the 13107-th frame of the *Elephants Dream* sequence when both of LTE and WiFi channels are lossy.

In conclusion, results illustrated in Figures 64 and 66 show that CAPA efficiently outperforms PA and ES in improving QoE while optimizing the total network goodput.

Remark that the PSNR resulting QoE for PA, as shown in Figure 65 for *Elephants Dream* sequence, would be very low, almost completely degrading the sequence when WiFi channel is lossy and when both LTE and WiFi are lossy. The PSNR resulting QoE for ES would be very low almost completely degrading the sequence in all three defined part of wireless lossy conditions. Even in this challenging scenario, our proposed strategy was able to keep the QoE in higher levels while optimizing the total network goodput.

4.5 Support of Fairness

We, now, evaluate the fairness support of our proposed scheduling strategy when it competes with another MMT flow where all paths between MMT sending and receiving entities share common bottlenecks.

4.5.0.1 Simulation Setup and Background Traffic Condition

For fairness experiment, as shown in Figure 70, we have extended the environment explained in Section 4.1 adding one more server named "Sender video server2" and one more receiver named "Receiver smartphone2". In this environment, each video server uses multiple paths to stream video to its assigned receiver. Therefore, video server1 streams video to receiver1, and video server2 streams video to receiver2. Subsequently, receiver1

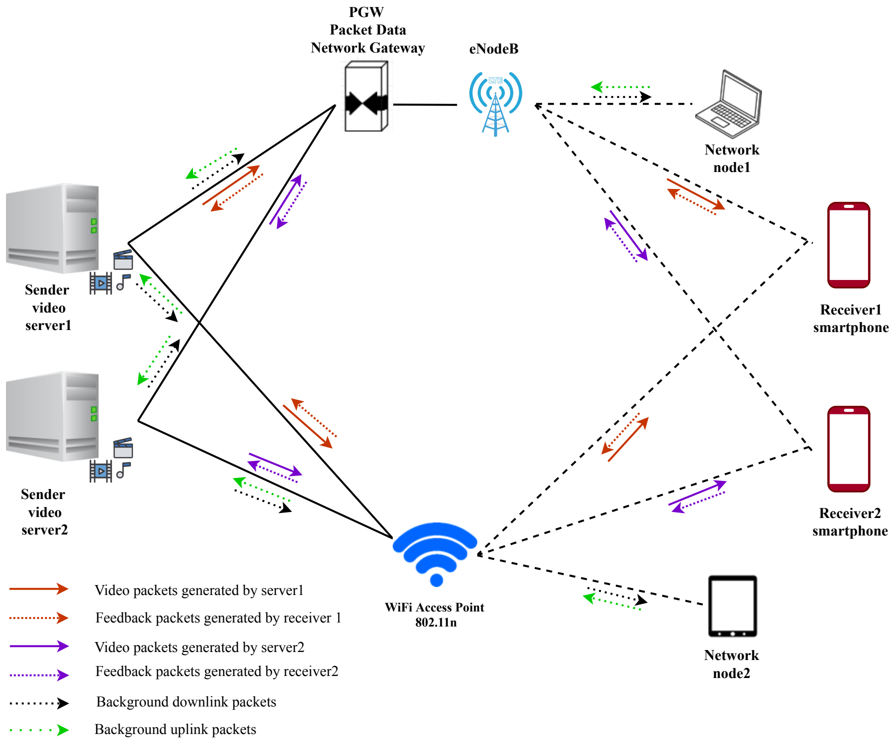


Figure 70 – Evaluation environment where two sender video servers use multiple paths to stream video to receiver smartphones sharing common bottlenecks.

generates feedbacks to server1, and receiver2 generates feedbacks to server2. Therefore, in this environment, there are bottlenecks for both LTE and WiFi paths.

Similar to what explained in Section 4.1, as depicted in Figure 70, uplink and downlink background traffics are added in the ns-3 simulation. The downlink background traffics are generated by the servers for both paths, and the uplink background traffics are generated by the network nodes. However, differently, TCP packets are generated for background traffic instead of UDP packets for this scenario. These TCP packets are generated by iperf in ns-3 DCE. The reason of choosing TCP background traffic instead of UDP is that there is no congestion control and feedback control for UDP. UDP has constant rate, and that's all. However, we need traffic to react to congestion. This is what TCP delivers. Therefore, we can demonstrate how our protocol behaves.

While in this scenario, WiFi and LTE paths' delay and bandwidth are the same as what was defined in Section 4.1, the wireless loss rate is considered as zero in this scenario because we want to focus on fairness in bandwidth sharing and we do not want to have wireless loss impact on the scheduling strategy. The fairness of CAPA, in this environment, is presented in terms of goodput.

4.5.0.2 Experimental Evaluation

Here, we evaluate the initial fairness test of our proposed scheduling strategy in terms of goodput in the already defined environment. In this network simulation scenario, *Elephants Dream* sequence, which is encoded with the source bit rate of 4 Mbps - previously explained in Section 4.2, is used. We also consider the constant transmission bit rate stream in this scenario (4 Mbps for this scenario).

Figure 71 depicts the simulation times from 0 to 120 seconds of simulation time. As it is shown in this figure, at 0 second of simulation time, server1 starts to stream the video to receiver1, then after 50 seconds of simulation time, the server2 starts to stream the video at 50 seconds of simulation time. Therefore, after 50 seconds of simulation time, there are two video streams in parallel and they compete with each other to access the available resource. Figure 71(a) shows the goodput achievement of receiver1 and Figure 70(b) shows the goodput achievement of receiver2. One can see is that both receiver1 and receiver2 have fair access to the available resources in both WiFi and LTE.

4.6 Concluding Remarks

In this chapter, we have evaluated our proposed solution compared to PA and ES scheduling strategies in different types of network condition; congested network and wireless lossy network scenarios, which are common network situations with big adverse effect on the perceived video quality. In the congested network scenario, we considered a variable background traffic and fixed random loss rate. In this scenario, we have mainly simulated and analyzed the results of congestion. But, for wireless lossy network scenario, the background traffic is constant and we have burst wireless losses which is kind of constant without increasing and decreasing. In this wireless lossy network scenario, there is no congestion and we have only wireless channel losses. Therefore, we have different configuration for channels, different effects, and different losses.

One objective of the proposed approach is increasing the quality of the final sequence, PSNR and SSIM. We could achieve this target by reducing the number of packets that are lost. Table 22 summarizes the achievements by CAPA compared to PA and ES provided in Subsections 4.4.1.2 and 4.4.2.2, respectively, under congested network scenario and wireless lossy network scenario where results are provided for various sequences with bit rate of 4 Mbps. We achieved that our proposing approach has good results for both of these network scenarios and could improve the QoE of end users. It does not matter if there is congestion or burst wireless losses. But maybe, the proposed solution is better in handling wireless losses than congestion. While, in wireless lossy network scenario, CAPA could increase the average video PSNR, respectively, by up to **6.84** dB **(20.30%)**, and

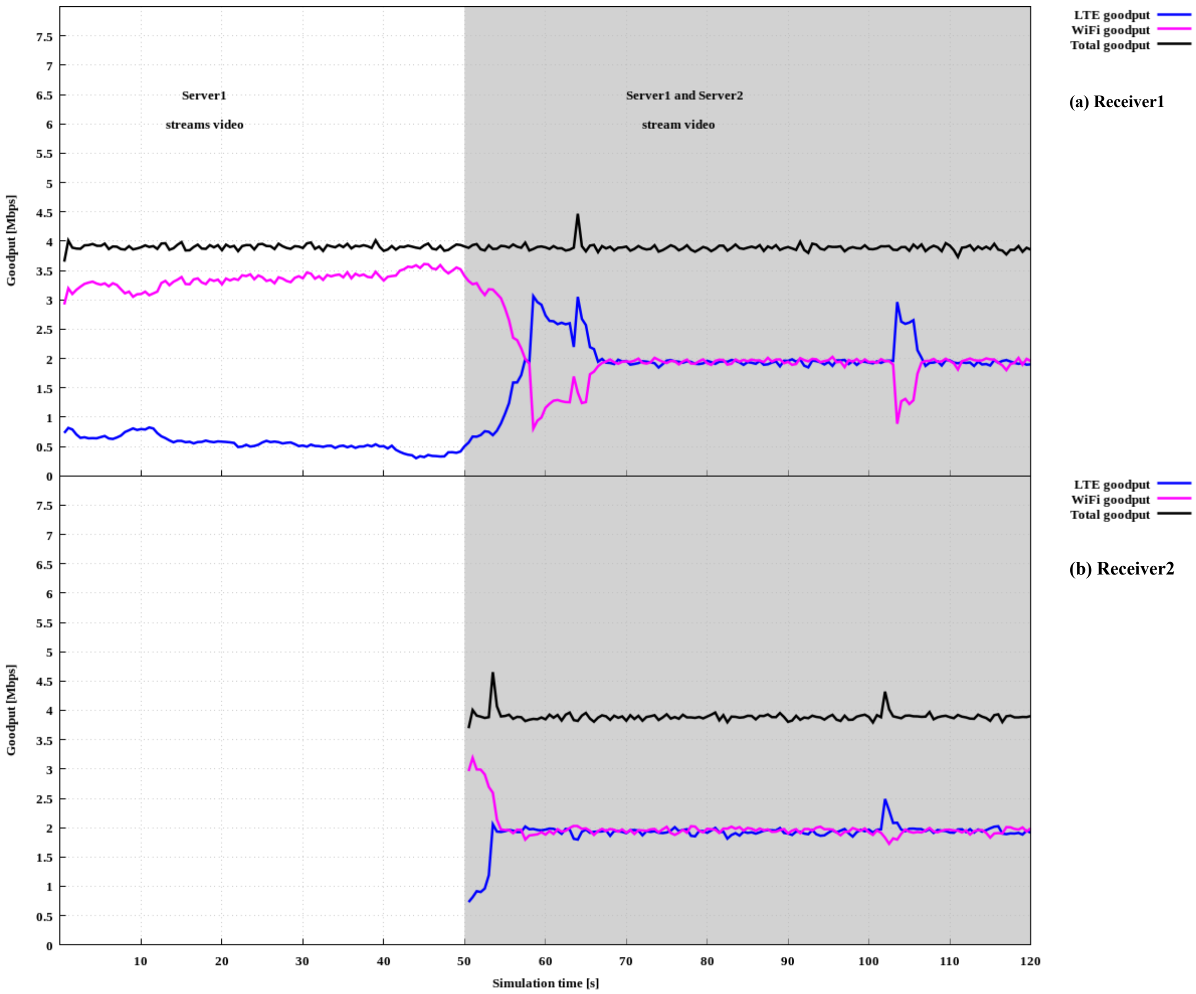


Figure 71 – Fairness results of goodput achievement in WiFi and LTE.

9.43 dB (30.32%) compared to PA and ES, in congested network scenario, CAPA could increase the average video PSNR, respectively, by up to **4.25** dB **(12.97%)**, and **7.22 dB (20.58%)** compared to PA and ES. Similarly, while in wireless lossy network scenario, CAPA could increase the SSIM, respectively, by up to **0.100 (12.72%)**, and **0.113 (14.23%)** compared to PA and ES, in congested network scenario, CAPA could increase the average video PSNR, respectively, by up to **0.033 (3.78%)**, and **0.102 (12.54%)** compared to PA and ES.

We have also checked the behaviour of the proposed strategy with different video bit rates under congested network scenario. The results show that our scheduler could properly handle these videos. Note that video with higher bit rate (original quality) also produces more packets, and consequently, more packets would be lost during transmission due to higher congestion. Therefore, higher original quality cannot necessarily improve the perceived quality.

Table 22 – Brief description of achievements by CAPA compared to PA and ES provided in Subsections 4.4.1.2 and 4.4.2.2, respectively, under congested network scenario and wireless lossy network scenario where results are provided for various sequences with bit rate of 4 Mbps.

Performance metrics	**Congested network scenario**	**Wireless lossy network scenario**
Goodput	Figure 39, CAPA achieves higher and more stable goodput compared to PA and ES.	Figure 61, CAPA achieves higher and more stable goodput compared to PA and ES.
Loss rate	Figure 40, CAPA reduces total loss rate compared to PA and ES, respectively, by up to 31.82% and 78.96%. Since CAPA can protect I packets, or I and NI packets based on different levels of packet protection, it can even achieve less I and/or NI packet loss rate compared to alternatives when there is a very high congested network situation.	Figure 62, CAPA reduces total loss rate compared to PA and ES, respectively, by up to 72.09% and 58.2%.
Delay	Table 19, CAPA outperforms ES but does not have always delay reduction compared to PA. Even if duplication improves the video quality but it costs of increasing congestion and effect on the delay.	Delay for most of our simulations and most of our sequence is almost the same.
PSNR	Figure 42, CAPA outperforms PA and ES, respectively, up to 4.25 dB (12.97%) and 7.22 dB (20.58%).	Figure 64, CAPA outperforms PA and ES, respectively, up to 6.84 dB (20.30%) and 9.43 dB (30.32%).
SSIM	Figure 44, CAPA outperforms PA and ES, respectively, up to 0.033 (3.78%) and 0.102 (12.54%).	Figure 66 CAPA outperforms PA and ES, respectively, up to 100 (12.72%) and 0.113 (14.23%).

We also had basic validation of fairness for CAPA, when two MMT flows compete with each other to access the available resource, where all paths between MMT sending and receiving entities share common bottlenecks. Even if the full understanding of fairness requires much more work and experience, but these initial experiments could properly show that the second MMT flow does not kill the first one and both receivers have fair access to both paths.

5 Conclusions and Future Work

In this thesis, we explore adding multipath support to MMT by means of a novel Content-Aware and Path-Aware (CAPA) scheduling strategy. The proposed strategy improves the scheduling capabilities of MMT when applied to multipath environments to provide a high-quality video streaming service over heterogeneous wireless networks. CAPA distributes packets through the network according to the wireless network conditions as well as the video packet content characteristics. It aims to select the best path for transmission of each video packet. It also provides models to cope with different network conditions and unstable communication channels in order to improve perceived video quality. The proposed CAPA has been fully implemented in the application layer as a compatible module and utilizes the standardized MMT signaling messages to provide feedback information. CAPA is examined over ns-3 DCE wireless network simulation for unicast video streaming. Some of the key properties of CAPA are summarized in Table 23.

Table 23 – Properties of the proposed CAPA adopted for MMT.

<table>
<tr><th colspan="2">Protocol layer</th><th colspan="2">Applied protocol</th><th>Compatibility</th></tr>
<tr><td colspan="2">Application</td><td colspan="2">MMT</td><td>Server and Client</td></tr>
<tr><th colspan="2">Which packet?</th><th colspan="2">How to protect the packet?</th><th>Which path?</th></tr>
<tr><td colspan="2">Content awareness</td><td colspan="2">Duplication</td><td>Path awareness
(Delay, PLR, Goodput)</td></tr>
<tr><th>Fairness</th><th>Video Compression</th><th>Error Concealment</th><th>Experimental Environment</th><th>Performance Metrics</th></tr>
<tr><td>Y</td><td>H.264</td><td>FFMPEG</td><td>NS3-DCE,
Client interfaces:
LTE, WiFi (802.11n)</td><td>Goodput,
Total packet loss rate,
I and NI frame packet loss rate,
Delay PSNR,
SSIM,</td></tr>
</table>

We have evaluated CAPA under different network conditions; congested network with extreme background traffic, and wireless lossy network with high burst wireless loss rates. These common network situations could have a severe adverse effect on the perceived video quality causing burst packet losses in the network. Simulation results have shown that CAPA is suitable for high-quality video streaming service over heterogeneous wireless networks in both network conditions. CAPA also outperforms ES and PA to achieve better QoE. ES is a simple scheduling strategy where packets are evenly split in both transmission channels. It is important to note that, even though evenly splitting packets is a simple scheduling strategy, it can take advantage of the network multipath capabilities and increase the total achieved goodput. PA refers to CAPA when content-aware is disabled. Such a result comparison makes clear how much content-aware protection can contribute to performance gains.

We have also evaluated different original video qualities in our environment under congested network condition. Results show that increasing the original quality does not always increase the perceived video quality, for instance, in our environment with the defined extreme background traffic, higher video quality which has more packets, increased losses due to higher congestion, and consequently, reduced quality.

After analyzing a single server and receiver in an entirely available network, it was time to make the infrastructure more realistic. This was done by expanding the setup with an additional server and receiver, introducing a competing scenario. The initial experiment results have shown that both receivers have fair access to the available resources in both WiFi and LTE.

A major target of this work is improving the MMT standard by adding multipath scheduling strategies. We proposed the collaboration document m44902r1 for the MMT IG standardization activity in 2018 and was adopted for the 4th edition of the ISO/IEC 23008-13 standard. The invention has also been protected through patent applications that has been filled out in INPI (OLIVEIRA J. F.; AFZAL; TESTONI, 2018) and US patent (OLIVEIRA J. F.; AFZAL; TESTONI, 2019). It is important to highlight that, this contribution is valuable because MMT has been already adopted by other standards such as ATSC 3.0 and it is expected to adopt by digital television broadcasting/broadband transmission systems as well as AR, VR, MR devices, smartphones, and tablets. Therefore, there is high potential for our approach to become wildly used.

In future work, we plan to investigate the following methods:

- Adding Forward Error Correction (FEC) stage exploiting what is already specified as the Application Layer FEC of the MMT standard. This way, it is possible to improve packet protection technique, leading to data loss rate reduction and improving QoE.
- Exploitation of variable bit rate for the video sequence instead of only constant bit rate. This approach could improve the QoE by adapting video traffic rate based on the network condition. In this way, video with a higher bit rate would be streamed through the network when the network is in good condition without wasting available resources. On the other hand, a lower video bit rate would be streamed through the network when the network condition is poor to avoid more congestion, losses, and delay.
- Evaluating CAPA while the user has mobility. Testing different speeds, velocity, motion degree and this kind of features which affect video quality. Besides that, we can also test the coverage and handoff between multiple networks.

Bibliography

ADDEPALLI, S.; SCHULZRINNE, H. G.; SINGH, A.; ORMAZABAL, G. *Heterogeneous access: Survey and design considerations.* [S.l.], 2013. Cited on page 41.

ADOBE. *Adobe's Real Time Messaging Protocol.* 2012. <https://www.adobe.com/content/dam/acom/en/devnet/pdf/swf-file-format-spec.pdf>. Cited on page 45.

ADOBE. *Adobe Systems Inc. HTTP Dynamic Streaming.* 2017. <http://www.adobe.com/products/hds-dynamic-streaming.html>. Accessed:15-JUN-2017. Cited on page 46.

AFZAL, S.; TESTONI, V.; de Oliveira, J. F.; ROTHENBERG, C. E.; KOLAN, P.; BOUAZIZI, I. A novel scheduling strategy for MMT-based multipath video streaming. In: *2018 IEEE Global Communications Conference: Communication QoS, Reliability and Modeling (Globecom2018 CQRM).* Abu Dhabi, United Arab Emirates: [s.n.], 2018. Cited 12 times on pages 28, 39, 54, 63, 64, 89, 91, 95, 97, 98, 99, and 102.

AL-JAWAD, A.; SHAH, P.; GEMIKONAKLI, O.; TRESTIAN, R. LearnQoS: a learning approach for optimizing QoS over multimedia-based SDNs. In: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB).* [S.l.: s.n.], 2018. p. 1–6. Cited on page 107.

ALZAHRANI, I. R.; RAMZAN, N.; KATSIGIANNIS, S.; AMIRA, A. Use of machine learning for rate adaptation in mpeg-dash for quality of experience improvement. In: SPRINGER. *5th International Symposium on Data Mining Applications.* [S.l.], 2018. p. 3–11. Cited on page 107.

AOKI, S. Emerging 8k services and their applications towards 2020. *ITU-T 2nd mini-Workshop on Immersive Live Experience*, 2017. Cited 2 times on pages 31 and 49.

APOSTOLOPOULOS, J. G. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. *Photonics West 2001-Electronic Imaging*, p. 392–409, 2000. Cited 2 times on pages 33 and 104.

APPLE. *Improving Network Reliability Using Multipath TCP.* 2017. <https://developer.apple.com/documentation/foundation/nsurlsessionconfiguration/improving_network_reliability_using_multipath_tcp?language=objc>. Cited on page 71.

AUSTERBERRY, D. *The technology of video and audio streaming.* [S.l.]: Taylor & Francis, 2005. Cited on page 36.

AVAYA. 2019. <https://downloads.avaya.com/css/P8/documents/100134063>. Accessed:05-may-2019. Cited on page 108.

AZAD, M. A.; MAHMOOD, R.; MEHMOOD, T. A comparative analysis of dccp variants (ccid2, ccid3), tcp and udp for mpeg4 video applications. *International Conference on Information and Communication Technologies. ICICT*, p. 40–45, 2009. Cited on page 71.

BAE, S. *Method for configuring and transmitting m-unit*. Google Patents, 2013. US Patent App. 13/651,191. Available from Internet: <http://www.google.sr/patents/US20130094594>. Cited 2 times on pages 31 and 49.

BAMPIS, C. G.; LI, Z.; KATSAVOUNIDIS, I.; HUANG, T.-Y.; EKANADHAM, C.; BOVIK, A. C. Towards perceptually optimized end-to-end adaptive video streaming. *arXiv preprint arXiv:1808.03898*, 2018. Cited on page 130.

BARAKABITZE, A. A.; MKWAWA, I.-H.; SUN, L.; IFEACHOR, E. Qualitysdn: Improving video quality using mptcp and segment routing in sdn/nfv. In: IEEE. *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. [S.l.], 2018. p. 182–186. Cited on page 41.

BARAKABITZE, A. A.; SUN, L.; MKWAWA, I.-H.; IFEACHOR, E. A novel qoe-centric sdn-based multipath routing approach for multimedia services over 5g networks. In: IEEE. *2018 IEEE International Conference on Communications (ICC)*. [S.l.], 2018. p. 1–7. Cited on page 107.

BARAKOVIĆ, S.; SKORIN-KAPOV, L. Survey and challenges of qoe management issues in wireless networks. *Journal of Computer Networks and Communications*, Hindawi, 2013. Cited on page 36.

BARRÉ, S.; PAASCH, C.; BONAVENTURE, O. Multipath tcp: from theory to practice. *International Conference on Research in Networking*, p. 444–457, 2011. Cited 3 times on pages 73, 77, and 113.

BELSHE, M.; PEON ROBERTO THOMSON, M. *Hypertext transfer protocol version 2 (http/2)*. 2015. RFC 7540. Cited on page 46.

BITMOVIN. *Why YouTube & Netflix use MPEG-DASH in HTML5*. 2017. <https://bitmovin.com/status-mpeg-dash-today-youtube-netflix-use-html5-beyond/>. Accessed:09-SEP-2017. Cited 2 times on pages 47 and 48.

BITMOVIN. *Bitmovin*. 2019. <https://github.com/bitmovin/bitmovin-player-web-samples>. Accessed:06-may-2019. Cited on page 48.

BLACK, D. *Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation*. 2018. RFC 8311. Cited on page 69.

BLENDER. *Elephants Dream*. 2005. <https://orange.blender.org/>. Cited on page 130.

BLENDER. *Big Buck Bunny*. 2007. <https://peach.blender.org/>. Cited on page 130.

BLENDER. *Sintel*. 2009. <https://durian.blender.org/>. Cited on page 130.

BLIN, J.-L. New quality evaluation method suited to multimedia context: Samviq. *Proceedings of the Second International Workshop on Video Processing and Quality Metrics, VPQM*, v. 6, 2006. Cited on page 105.

BOCHAROV, J. A. *Smooth Streaming Transport Protocol*. 2017. <https://docs.microsoft.com/en-us/iis/media/smooth-streaming/smooth-streaming-transport-protocol>. Accessed:15-AGU-2017. Cited on page 46.

BOKANI, A.; HASSAN, M.; KANHERE, S. Http-based adaptive streaming for mobile clients using markov decision process. *IEEE 20th International Packet Video Workshop (PV)*, p. 1–8, 2013. Cited on page 62.

BONAVENTURE, O.; SEO, S. Multipath tcp deployments. *IETF Journal*, v. 12, n. 2, p. 24–27, 2016. Cited on page 71.

BONDERUD, D. *Internet Speed Test: 3G, 4G, LTE, and Wifi — Who Wins?* 2017. <http://www.bandwidthplace.com/internet-speed-test-3g-4g-lte-and-wifi-who-wins-article/>. Cited on page 34.

BOYCE, J.; CHEN, J.; CHEN, Y.; FLYNN, D.; HANNUKSELA, M.; NACCARI, M.; ROSEWARNE, C.; SHARMAN, K.; SOLE, J.; SULLIVAN, G. *et al.* Edition 2 draft text of high efficiency video coding (hevc), including format range (rext), scalability (shvc), and multi-view (mv-hevc) extensions. *document JCTVC-R1013*, 2014. Cited on page 105.

BROSH, E.; BASET, S. A.; MISRA, V.; RUBENSTEIN, D.; SCHULZRINNE, H. The delay-friendliness of tcp for real-time traffic. *IEEE/ACM Transactions On Networking*, IEEE, v. 18, n. 5, p. 1478–1491, 2010. Cited on page 35.

BT.1683, I.-R. Bt.1683: Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference. 2004. Cited on page 133.

BUI, D. H.; LEE, K.; OH, S.; SHIN, I.; SHIN, H.; WOO, H.; BAN, D. Greenbag: Energy-efficient bandwidth aggregation for real-time streaming in heterogeneous mobile wireless networks. *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, p. 57–67, 2013. Cited 12 times on pages 34, 54, 65, 66, 88, 89, 91, 97, 99, 100, 101, and 102.

CAIA. *MPTCP Implementation for FreeBSD*. 2019. <http://caia.swin.edu.au/urp/newtcp/mptcp/>. Accessed:05-may-2019. Cited on page 71.

CAO, Y.; LIU, Q.; LUO, G.; YI, Y.; HUANG, M. Pr-mptcp+: Context-aware qoe-oriented multipath tcp partial reliability extension for real-time multimedia applications. *Visual Communications and Image Processing (VCIP), 2016*, p. 1–4, 2016. Cited 13 times on pages 54, 73, 74, 75, 76, 88, 90, 91, 97, 99, 100, 103, and 104.

CEN, S.; COSMAN, P. C.; VOELKER, G. M. End-to-end differentiation of congestion and wireless losses. *IEEE/ACM Transactions on networking*, IEEE, v. 11, n. 5, p. 703–717, 2003. Cited 5 times on pages 68, 74, 81, 104, and 119.

CHAKARESKI, J.; GIROD, B. Rate-distortion optimized packet scheduling and routing for media streaming with path diversity. *Data Compression Conference*, p. 203–212, 2003. Cited on page 30.

CHAKARESKI, J.; HAN, S.; GIROD, B. Layered coding vs. multiple descriptions for video streaming over multiple paths. *Multimedia Systems*, Springer, v. 10, n. 4, p. 275–285, 2005. Cited on page 30.

CHAN, H.; WEI, A.; SONG, F.; ZHANG, H. One way latency considerations for multipath in quic. *IETF Internet Draft, subsequent updates*, 2017. Cited on page 50.

CHAN, M. C.; RAMJEE, R. Tcp/ip performance over 3g wireless links with rate and delay variation. *Wireless Networks*, Springer-Verlag New York, Inc., v. 11, n. 1-2, p. 81–97, 2005. Cited on page 34.

CHANG, Y.-L.; LIN, T.-L.; COSMAN, P. C. Network-based h. 264/avc whole-frame loss visibility model and frame dropping methods. *IEEE Transactions on Image Processing*, IEEE, v. 21, n. 8, p. 3353–3363, 2012. Cited on page 105.

CHEBROLU, K.; RAO, R. R. Bandwidth aggregation for real-time applications in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, IEEE, v. 5, n. 4, p. 388–403, 2006. Cited 6 times on pages 11, 54, 83, 95, 97, and 99.

CHEN, Y.-C.; LIM, Y.-s.; GIBBENS, R. J.; NAHUM, E. M.; KHALILI, R.; TOWSLEY, D. A measurement-based study of multipath tcp performance over wireless networks. *Proceedings of the 2013 conference on Internet measurement conference*, p. 455–468, 2013. Cited 3 times on pages 34, 134, and 196.

CHEN, Y.-C.; TOWSLEY, D. On bufferbloat and delay analysis of multipath tcp in wireless networks. In: IEEE. *Networking Conference, IFIP*. [S.l.], 2014. p. 1–9. Cited on page 35.

CHESHIRE, S. *"Advances in networking, part 1"*. 2019. <https://developer.apple.com/videos/play/wwdc2017/707/>. Accessed:05-may-2019. Cited 2 times on pages 71 and 108.

CHOW, A. L.; YANG, H.; XIA, C. H.; KIM, M.; LIU, Z.; LEI, H. Ems: Encoded multipath streaming for real-time live streaming applications. *17th IEEE International Conference on Network Protocols (ICNP)*, p. 233–243, 2009. Cited 3 times on pages 36, 70, and 119.

CHOWRIKOPPALU, Y.; GOWDA, P. *Multipath Adaptive Video Streaming over Multipath TCP*. Tese (Doutorado) — Intel, 2013. Cited 3 times on pages 54, 61, and 63.

CISCO. *Cisco visual networking index: global mobile data traffic forecast update, 2014–2019*. 2015. Cited on page 29.

CISCO. *Cisco Visual Networking Index: Forecast and Methodology, 2016–2021*. 2017. <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>. Cited on page 29.

CLARITY videeo. *Advancing To Multi-Scale SSIM*. 2019. <https://videoclarity.com/wp-content/uploads/2014/02/Advancing-To-Multi-Scale-SSIM-Final.pdf>. Accessed:10-SEP-2019. Cited on page 134.

COFANO, G.; CICCO, L. D.; ZINNER, T.; NGUYEN-NGOC, A.; TRAN-GIA, P.; MASCOLO, S. Design and performance evaluation of network-assisted control strategies for http adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 13, n. 3s, p. 42, 2017. Cited on page 61.

CORBILLON, X.; APARICIO-PARDO, R.; KUHN, N.; TEXIER, G.; SIMON, G. Cross-layer scheduler for video streaming over mptcp. *Proceedings of the 7th International Conference on Multimedia Systems*, p. 7, 2016. Cited 16 times on pages 35, 54, 61, 62, 72, 73, 74, 84, 90, 91, 97, 98, 99, 100, 104, and 122.

CORREIA, P.; FERREIRA, L.; ASSUNCAO, P. A. A.; CRUZ, L.; SILVA, V. Optimal priority mdc video streaming for networks with path diversity. *IEEE International Conference on Telecommunications and Multimedia (TEMU)*, p. 54–59, 2012. Cited 7 times on pages 54, 67, 68, 89, 95, 102, and 104.

DANDACHI, G. *Multihoming in heterogeneous wireless networks*. Tese (Doutorado) — Evry, Institut national des télécommunications, 2017. Cited on page 106.

DASHIF. *DASH industry forum*. 2019. <https://dashif.org/members/>. Accessed:06-may-2019. Cited on page 48.

DENG, S.; NETRAVALI, R.; SIVARAMAN, A.; BALAKRISHNAN, H. Wifi, lte, or both?: Measuring multi-homed wireless internet performance. *Proceedings of the Conference on Internet Measurement Conference*, p. 181–194, 2014. Cited on page 73.

DIOP, C.; DUGUE, G.; CHASSOT, C.; EXPOSITO, E. Qos-oriented mptcp extensions for multimedia multi-homed systems. *IEEE 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, p. 1119–1124, 2012. Cited 11 times on pages 54, 73, 74, 75, 76, 88, 90, 91, 97, 103, and 104.

DOMZAL, J.; DULINSKI, Z.; KANTOR, M.; RZKASA, J.; STANKIEWICZ, R.; WAJDA, K.; WOJCIK, R. A survey on methods to provide multipath transmission in wired packet networks. *Computer Networks*, Elsevier, v. 77, p. 18–41, 2015. Cited on page 41.

EtherChannels. 2019. <https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/31sg/configuration/guide/conf/channel.pdf>. Accessed:05-may-2019. Cited on page 108.

EVENSEN, K.; KASPAR, D.; GRIWODZ, C.; HALVORSEN, P.; HANSEN, A.; ENGELSTAD, P. Improving the performance of quality-adaptive video streaming over multiple heterogeneous access networks. *Proceedings of the second annual ACM conference on Multimedia systems*, p. 57–68, 2011. Cited 8 times on pages 54, 65, 66, 89, 91, 97, 99, and 102.

EVENSEN, K.; KASPAR, D.; GRIWODZ, C.; HALVORSEN, P.; HANSEN, A. F.; ENGELSTAD, P. Using bandwidth aggregation to improve the performance of quality-adaptive streaming. *Signal Processing: Image Communication*, Elsevier, v. 27, n. 4, p. 312–328, 2012. Cited 8 times on pages 54, 65, 66, 89, 91, 97, 99, and 102.

EVENSEN, K.; KUPKA, T.; KASPAR, D.; HALVORSEN, P.; GRIWODZ, C. Quality-adaptive scheduling for live streaming over multiple access networks. *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, p. 21–26, 2010. Cited 8 times on pages 54, 65, 66, 89, 91, 97, 99, and 102.

EXATA. *EXata*. 2017. <http://web.scalable-networks.com/exata>. Cited on page 105.

EXPOSITO, E.; GINESTE, M.; DAIRAINE, L.; CHASSOT, C. Building self-optimized communication systems based on applicative cross-layer information. *Computer Standards & Interfaces*, Elsevier, v. 31, n. 2, p. 354–361, 2009. Cited on page 75.

FAIRHURST, G.; TRAMMELL, B.; KUEHLEWIND, M. *Services provided by IETF transport protocols and congestion control mechanisms*. 2017. RFC 8095. Cited 3 times on pages 67, 73, and 83.

FANG, T.; CHAU, L.-P. Robust group-of-picture architecture for video transmission over error-prone channels. In: IEEE. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. [S.l.], 2005. p. 3447–3450. Cited on page 90.

FERLIN-OLIVEIRA, S.; DREIBHOLZ, T.; ALAY, Ö. Tackling the challenge of bufferbloat in multi-path transport over heterogeneous wireless networks. In: IEEE. *Quality of Service (IWQoS), 22nd International Symposium of*. [S.l.], 2014. p. 123–128. Cited on page 35.

FEUVRE, J. L.; CONCOLATO, C.; BOUZAKARIA, N.; NGUYEN, V.-T.-T. Mpeg-dash for low latency and hybrid streaming services. *Proceedings of the 23rd ACM international conference on Multimedia*, p. 751–752, 2015. Cited on page 59.

FFMPEG. 2017. <https://ffmpeg.org>. Accessed:10-JUL-2018. Cited 2 times on pages 105 and 129.

FIELDING, R.; GETTYS, J.; MOGUL, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P.; BERNERS-LEE, T. *Hypertext transfer protocol–HTTP/1.1*. 1999. Cited on page 46.

FLOYD, S.; HANDLEY, M.; PADHYE, J.; WIDMER, J. Equation-based congestion control for unicast applications. *ACM SIGCOMM Computer Communication Review*, ACM, v. 30, n. 4, p. 43–56, 2000. Cited on page 67.

FLOYD, S.; KOHLER, E. *Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control*. 2006. RFC 4341. Cited on page 70.

FLOYD, S.; KOHLER, E.; PADHYE, J. *Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)*. 2006. RFC 4342. Cited on page 70.

FLOYD, S.; MAHDAVI, J.; MATHIS, M.; PODOLSKY, M. *An extension to the selective acknowledgement (SACK) option for TCP*. [S.l.], 2000. Cited 2 times on pages 69 and 81.

FONTES, R. D. R.; CAMPOLO, C.; ROTHENBERG, C. E.; MOLINARO, A. From theory to experimental evaluation: Resource management in software-defined vehicular networks. *IEEE Access*, v. 5, p. 3069–3076, 2017. Cited on page 107.

FONTES, R. R.; AFZAL, S.; BRITO, S. H. B.; SANTOS, M. A. S.; ROTHENBERG, C. E. Mininet-wifi: Emulating software-defined wireless networks. In: *2015 11th International Conference on Network and Service Management (CNSM)*. [S.l.: s.n.], 2015. p. 384–389. Cited on page 105.

FORD, A.; RAICIU, C.; BARRE, S.; LOUVAIN. *TCP Extensions for Multipath Operation with Multiple Addresses*. 2009. RFC 6824, subsequent updates. Cited 3 times on pages 11, 71, and 72.

FORD, A.; RAICIU, C.; HANDLEY, M.; BARRE, s.; IYENGAR, J. Architectural guidelines for multipath tcp development. *RFC 6182*, 2011. Cited on page 71.

FRERIS, N. M.; HSU, C.-H.; SINGH, J. P.; ZHU, X. Distortion-aware scalable video streaming to multinetwork clients. *IEEE/ACM Transactions on Networking*, IEEE, v. 21, n. 2, p. 469–481, 2013. Cited 12 times on pages 54, 67, 68, 89, 90, 94, 96, 97, 98, 99, 102, and 104.

FRÖMMGEN, A.; RIZK, A.; ERBSHÄUSSER, T.; WELLER, M.; KOLDEHOFE, B.; BUCHMANN, A.; STEINMETZ, R. A programming model for application-defined multipath tcp scheduling. In: ACM. *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference*. [S.l.], 2017. p. 134–146. Cited on page 106.

FROSSARD, P. Fec performance in multimedia streaming. *IEEE Communications Letters*, IEEE, v. 5, n. 3, p. 122–124, 2001. Cited on page 92.

FU, S.; ATIQUZZAMAN, M. Sctp: State of the art in research, products, and technical challenges. *IEEE Communications Magazine*, IEEE, v. 42, n. 4, p. 64–76, 2004. Cited on page 76.

GALAXYS5. 2017. <http://galaxys5guide.com/samsung-galaxy-s5-features-explained/galaxy-s5-download-booster/>. Accessed:15-JUN-2017. Cited 2 times on pages 33 and 108.

GALL, D. L. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, ACM, v. 34, n. 4, p. 46–58, 1991. Cited on page 90.

GHASEMPOUR, Y.; SILVA, C. R. da; CORDEIRO, C.; KNIGHTLY, E. W. Ieee 802.11 ay: Next-generation 60 ghz communication for 100 gb/s wi-fi. *IEEE Communications Magazine*, IEEE, v. 55, n. 12, p. 186–192, 2017. Cited on page 107.

GPAC. *GPAC MP4BOX, Version 0.5.1*. 2015. Cited 2 times on pages 129 and 133.

GUTTERMAN, C.; GUO, K.; ARORA, S.; WANG, X.; WU, L.; KATZ-BASSETT, E.; ZUSSMAN, G. Requet: Real-Time QoE Detection for Encrypted YouTube Traffic. *MMSys, Amherst, MA, US*, 2019. Cited on page 33.

HABAK, K.; HARRAS, K. A.; YOUSSEF, M. Bandwidth aggregation techniques in heterogeneous multi-homed devices: A survey. *Computer Networks*, Elsevier, v. 92, p. 168–188, 2015. Cited 2 times on pages 41 and 66.

HABIB, S.; QADIR, J.; ALI, A.; HABIB, D.; LI, M.; SATHIASEELAN, A. The past, present, and future of transport-layer multipath. *Journal of Network and Computer Applications*, Elsevier, v. 75, p. 236–258, 2016. Cited 6 times on pages 61, 67, 69, 72, 73, and 95.

HAMILTON, R.; IYENGAR, J.; SWETT, I.; WILK, A. Quic: A udp-based secure and reliable transport. *IETF Internet Draft, subsequent updates*, 2016. Cited on page 49.

HAN, B.; QIAN, F.; JI, L.; GOPALAKRISHNAN, V.; BEDMINSTER, N. Mp-dash: Adaptive video streaming over preference-aware multipath. *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, p. 129–143, 2016. Cited 13 times on pages 11, 54, 61, 62, 74, 85, 86, 89, 91, 99, 100, 101, and 103.

HAN, S.; JOO, H.; LEE, D.; SONG, H. An end-to-end virtual path construction system for stable live video streaming over heterogeneous wireless networks. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 29, n. 5, p. 1032–1041, 2011. Cited on page 34.

HASAN, M. Z.; AL-RIZZO, H.; AL-TURJMAN, F. A survey on multipath routing protocols for qos assurances in real-time wireless multimedia sensor networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 3, p. 1424–1456, 2017. Cited on page 41.

HEMMINGER, S. Network emulation with netem. *Linux conf au*, p. 18–23, 2005. Cited on page 105.

HERGUNER, K.; KALAN, R. S.; CETINKAYA, C.; SAYIT, M. Towards qos-aware routing for dash utilizing mptcp over sdn. In: IEEE. *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on.* [S.l.], 2017. p. 1–6. Cited on page 41.

HESMANS, B.; DUCHENE, F.; PAASCH, C.; DETAL, G.; BONAVENTURE, O. Are tcp extensions middlebox-proof? *Proceedings of the workshop on Hot topics in middleboxes and network function virtualization*, p. 37–42, 2013. Cited on page 73.

HOSSFELD, T.; SCHATZ, R.; KRIEGER, U. R. Qoe of youtube video streaming for current internet transport protocols. In: *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance.* [S.l.]: Springer, 2014. p. 136–150. Cited 3 times on pages 67, 69, and 71.

HOUZÉ, P.; MORY, E.; TEXIER, G.; SIMON, G. Applicative-layer multipath for low-latency adaptive live streaming. *IEEE International Conference on Communications (ICC)*, p. 1–7, 2016. Cited 10 times on pages 54, 59, 61, 63, 89, 91, 97, 99, 100, and 102.

HUANG, C.-M.; CHEN, Y.-C.; LIN, S.-Y. The qos-aware order prediction scheduling (qops) scheme for video streaming using the multi-path datagram congestion control protocol (mp-dccp). *IEEE 5th International Conference on Network-Based Information Systems (NBiS)*, p. 276–283, 2012. Cited 9 times on pages 54, 67, 70, 71, 90, 97, 98, 101, and 102.

HUAWEI. *AirEngine: A Leading Wi-Fi 6 Product Among Future 5G Networks.* 2019. <https://e.huawei.com/za/products/enterprise-networking/wlan/wifi-6?utm_medium=cpc&utm_source=google&utm_campaign=EEBGHQ197121L&utm_content=General&source=Adcopy&gclid=CjwKCAjw_MnmBRAoEiwAPRRWWzheigpJFt_8HWSpn_trjC7EpWSUXtdQCkU9bdGC5rRAcDMqElpGyBoCEs4QAvD_BwE>. Accessed:06-may-2019. Cited on page 107.

HULU. *Hulu.* 2019. <https://www.hulu.com/welcome>. Accessed:06-may-2019. Cited on page 48.

Huo, Y.; Hellge, C.; Wiegand, T.; Hanzo, L. A tutorial and review on inter-layer fec coded layered video streaming. *IEEE Communications Surveys Tutorials*, v. 17, n. 2, p. 1166–1207, Secondquarter 2015. ISSN 1553-877X. Cited on page 28.

HUO, Y.; HELLGE, C.; WIEGAND, T.; HANZO, L. A tutorial and review on inter-layer fec coded layered video streaming. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 2, p. 1166–1207, 2015. Cited on page 92.

HUSZÁK, Á.; IMRE, S. Analysing gop structure and packet loss effects on error propagation in mpeg-4 video streams. *Proceedings of the 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, p. 1–5, 2010. Cited on page 116.

HUYSEGEMS, R.; HOOFT, J. van der; BOSTOEN, T.; ALFACE, P. R.; PETRANGELI, S.; WAUTERS, T.; TURCK, F. D. Http/2-based methods to improve the live experience of adaptive streaming. *Proceedings of the 23rd ACM international conference on Multimedia*, p. 541–550, 2015. Cited on page 46.

IBNALFAKIH, S.; SABIR, E.; SADIK, M. Multi-homing as an enabler for 5g networks: Survey and open challenges. In: SPRINGER. *International Symposium on Ubiquitous Networking*. [S.l.], 2016. p. 347–356. Cited on page 106.

ICTEAM. *MultiPath TCP - Linux Kernel implementation*. 2017. <http://www.multipath-tcp.org>. Accessed:15-JUN-2017. Cited on page 71.

ICTEAM. *MultiPath TCP - versions of the MPTCP kernel on smartphones*. 2019. <https://multipath-tcp.org/pmwiki.php/Users/Android>. Accessed:05-may-2019. Cited on page 71.

ISO/IEC. Information technology – generic coding of moving pictures and associated audio information: Part 1 systems. *ISO/IEC 13818-1*, 2007. Cited on page 45.

ISO/IEC. Information technology – coding of audio-visual objects – part 12: Iso base media file format. *ISO/IEC 14496-12*, 2012. Cited on page 52.

ISO/IEC. Information technology – dynamic adaptive streaming over http (dash) – part 5: Server and network assisted dash (sand). *ISO/IEC CD 23009-5*, 2014. Cited on page 61.

ITU-T. G.1010 End-user multimedia qos categories. 2001. Cited on page 36.

ITU-T. Advanced video coding for generic audio-visual services. *ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2003. Cited on page 105.

ITU-T. High efficiency video coding. *ITU-T Rec. H.265 and ISO/IEC 23008-2*, 2013. Cited 2 times on pages 45 and 105.

IYENGAR, J. R.; AMER, P. D.; STEWART, R. Concurrent multipath transfer using sctp multihoming over independent end-to-end paths. *IEEE/ACM Transactions on networking*, IEEE, v. 14, n. 5, p. 951–964, 2006. Cited 2 times on pages 79 and 95.

J.144, I.-T. J.144: Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference. 2004. Cited on page 133.

JAIN, M.; DOVROLIS, C. Pathload: A measurement tool for end-to-end available bandwidth. In: *In Proceedings of Passive and Active Measurements (PAM) Workshop*. [S.l.: s.n.], 2002. Cited on page 98.

JAMES, C.; HALEPOVIC, E.; WANG, M.; JANA, R.; SHANKARANARAYANAN, N. Is multipath tcp (mptcp) beneficial for video streaming over dash? *24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, p. 331–336, 2016. Cited 3 times on pages 46, 59, and 61.

JARSCHEL, M.; SCHLOSSER, D.; SCHEURING, S.; HOSSFELD, T. An evaluation of qoe in cloud gaming based on subjective tests. In: IEEE. *Innovative mobile and internet services in ubiquitous computing (imis), 2011 fifth international conference on*. [S.l.], 2011. p. 330–335. Cited on page 28.

(JCT-VC), J. C. T. O. V. C. *HEVC Model (HM) repository*. 2017. <https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/>. Cited on page 105.

JM. *H.264/AVC JM Reference Software, Version 19.0*. 2015. Cited 2 times on pages 128 and 133.

JOHANSEN, D.; JOHANSEN, H.; AARFLOT, T.; HURLEY, J.; KVALNES, Å.; GURRIN, C.; ZAV, S.; OLSTAD, B.; AABERG, E.; ENDESTAD, T. *et al.* Davvi: A prototype for the next generation multimedia entertainment platform. *Proceedings of the 17th ACM international conference on Multimedia*, p. 989–990, 2009. Cited on page 65.

JUNG, A. *Comparison of Video Quality Assessment Methods*. Dissertação (Mestrado) — Blekinge Institute of Technology, 2017. Cited on page 133.

JUNG, T.-J.; LEE, H.-r.; SEO, K.-d. Overview on mpeg mmt technology and its application to hybrid media delivery over heterogeneous networks. *Pacific Rim Conference on Multimedia*, p. 660–669, 2015. Cited 4 times on pages 11, 49, 50, and 52.

(JVT), J. V. T. *H.264/MPEG4-AVC Model (JM) repository*. 2017. <http://iphome.hhi.de/suehring/tml/download/>. Accessed:5-JUL-2017. Cited on page 105.

(JVT), J. V. T. *Joint Scalable Video Model (JSVM) repository*. 2017. <https://www.hhi.fraunhofer.de/en/departments/vca/research-groups/image-video-coding/research-topics/svc-extension-of-h264avc/jsvm-reference-software.html>. Accessed:10-JUL-2017. Cited on page 105.

KALVA, H.; LEE, J.-B. The vc-1 video coding standard. *IEEE MultiMedia*, IEEE, v. 14, n. 4, 2007. Cited on page 47.

KARIMI, P.; SHERMAN, M.; BRONZINO, F.; SESKAR, I.; RAYCHAUDHURI, D.; GOSAIN, A. Evaluating 5g multihoming services in the mobilityfirst future internet architecture. In: IEEE. *IEEE Vehicular Technology Conference (VTC Spring)*. [S.l.], 2017. p. 1–5. Cited on page 106.

KARYA, O.; SAESARIA, S.; BUDIYANTO, S. RTP analysis for the video transmission process on WhatsApp and Skype against signal strength variations in 802.11 network environments. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2018. v. 453, n. 1, p. 012062. Cited on page 48.

KASPAR, D.; EVENSEN, K.; ENGELSTAD, P.; HANSEN, A. F. Using http pipelining to improve progressive download over multiple heterogeneous interfaces. *IEEE International Conference on Communications (ICC)*, p. 1–5, 2010. Cited 2 times on pages 60 and 65.

Kaul, A.; Obraczka, K.; Santos, M. A. S.; Rothenberg, C. E.; Turletti, T. Dynamically distributed network control for message dissemination in its. In: *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. [S.l.: s.n.], 2017. p. 1–9. Cited on page 107.

KAZEMI, M.; SHIRMOHAMMADI, S.; SADEGHI, K. H. A review of multiple description coding techniques for error-resilient video delivery. *Multimedia Systems*, Springer, v. 20, n. 3, p. 283–309, 2014. Cited 3 times on pages 28, 94, and 95.

KELLY, A.; MUNTEAN, G.; PERRY, P.; MURPHY, J. Delay-centric handover in sctp over wlan. *Transactions on Automatic Control and Computer Science*, v. 49, n. 63, p. 1–6, 2004. Cited 3 times on pages 54, 77, and 78.

Kim, S.; Yun, D.; Chung, K. Video quality adaptation scheme for improving qoe in http adaptive streaming. In: *2016 International Conference on Information Networking (ICOIN)*. [S.l.: s.n.], 2016. p. 201–205. Cited on page 28.

KIM, Y.; CHOI, S.; JANG, K.; HWANG, H. Throughput enhancement of ieee 802.11 wlan via frame aggregation. In: IEEE. *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*. [S.l.], 2004. v. 4, p. 3030–3034. Cited on page 127.

KLEINROUWELER, J. W.; CABRERO, S.; CESAR, P. Delivering stable high-quality video: An sdn architecture with dash assisting network elements. In: ACM. *Proceedings of the 7th International Conference on Multimedia Systems*. [S.l.], 2016. p. 4. Cited on page 61.

KOBAYASHI, M.; NAKAYAMA, H.; ANSARI, N.; KATO, N. Robust and efficient stream delivery for application layer multicasting in heterogeneous networks. *IEEE Transactions on Multimedia*, IEEE, v. 11, n. 1, p. 166–176, 2009. Cited on page 95.

KOHLER, E.; HANDLEY, M.; FLOYD, S. *Datagram Congestion Control Protocol (DCCP)*. [S.l.]: RFC 4340, 2006. Cited on page 70.

KOLAN, P.; BOUAZIZI, I. *Method and apparatus for multipath media delivery*. [S.l.]: Google Patents, 2016. US Patent App. 15/001,018. Cited 4 times on pages 28, 54, 63, and 64.

KREUTZ, D.; RAMOS, F. M.; VERISSIMO, P. E.; ROTHENBERG, C. E.; AZODOLMOLKY, S.; UHLIG, S. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2015. Cited 3 times on pages 41, 81, and 106.

KRISHNAN, S. S.; SITARAMAN, R. K. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Transactions on Networking*, IEEE, v. 21, n. 6, p. 2001–2014, 2013. Cited on page 58.

KUO, C.-I.; SHIH, C.-H.; SHIEH, C.-K.; HWANG, W.-S.; KE, C.-H. Modeling and analysis of frame-level forward error correction for mpeg video over burst-loss channels. *Applied Mathematics & Information Sciences*, Natural Sciences Publishing Corp, v. 8, n. 4, p. 1845, 2014. Cited on page 92.

KYUNG, C.-M. *et al. Theory and applications of smart cameras*. [S.l.]: Springer, 2016. Cited on page 130.

LATAH, M.; TOKER, L. Artificial intelligence enabled software defined networking: A comprehensive overview. *arXiv preprint arXiv:1803.06818*, 2018. Cited on page 107.

LE, H. T.; NGUYEN, H. N.; NGOC, N. P.; PHAM, A. T.; THANG, T. C. A novel adaptation method for http streaming of vbr videos over mobile networks. *Mobile Information Systems*, Hindawi Publishing Corporation, v. 2016, 2016. Cited on page 60.

LEDERER, S.; MÜLLER, C.; TIMMERER, C. Dynamic adaptive streaming over http dataset. *Proceedings of the 3rd Multimedia Systems Conference*, p. 89–94, 2012. Cited on page 66.

LEE, J.; UDDIN, M.; TOURRILHES, J.; SEN, S.; BANERJEE, S.; ARNDT, M.; KIM, K.-H.; NADEEM, T. mesdn: Mobile extension of sdn. In: *Proceedings of the Fifth International Workshop on Mobile Cloud Computing & Services*. New York, NY, USA: ACM, 2014. (MCS '14), p. 7–14. ISBN 978-1-4503-2824-1. Available from Internet: <http://doi.acm.org/10.1145/2609908.2609948>. Cited on page 106.

LEI, K.; ZHONG, S.; ZHU, F.; XU, K.; ZHANG, H. An ndn iot content distribution model with network coding enhanced forwarding strategy for 5g. *IEEE Transactions on Industrial Informatics*, IEEE, v. 14, n. 6, p. 2725–2735, 2018. Cited on page 107.

LEI, W.; ZHANG, W.; LIU, S. Multipath real-time transport protocol based on application-level relay (mprtp-ar). *IETF Internet Draft, subsequent updates*, 2017. Cited 5 times on pages 11, 54, 55, 57, and 58.

LENNOX, J.; WESTERLUND, M.; WU, Q.; PERKINS, C. *Sending Multiple RTP Streams in a Single RTP Session*. 2017. Internet Requests for Comments. Cited on page 54.

LI, B.; WANG, C.; XU, Y.; MA, Z. An mmt based heterogeneous multimedia system using quic. *IEEE 2nd International Conference on Cloud Computing and Internet of Things (CCIOT)*, p. 129–133, 2016. Cited 2 times on pages 49 and 50.

LI, M.; LUKYANENKO, A.; OU, Z.; YLA-JAASKI, A.; TARKOMA, S.; COUDRON, M.; SECCI, S. Multipath transmission for the internet: A survey. *IEEE Communications Surveys Tutorials, vol. PP*, n. 99, p. 1–41, 2016. Cited 9 times on pages 34, 41, 42, 53, 61, 83, 84, 87, and 95.

LI, M.; LUKYANENKO, A.; TARKOMA, S.; CUI, Y.; YLÄ-JÄÄSKI, A. Tolerating path heterogeneity in multipath tcp with bounded receive buffers. *Computer Networks*, Elsevier, v. 64, p. 1–14, 2014. Cited on page 70.

LI, X.; DONG, M.; MA, Z.; FERNANDES, F. C. Greentube: power optimization for mobile videostreaming via dynamic cache management. *Proceedings of the 20th ACM international conference on Multimedia*, p. 279–288, 2012. Cited on page 107.

LIM, Y.; AOKI, S.; BOUAZIZI, I.; SONG, J. New mpeg transport standard for next generation hybrid broadcasting system with ip. *IEEE Transactions on Broadcasting*, IEEE, v. 60, n. 2, p. 160–169, 2014. Cited 5 times on pages 30, 45, 47, 48, and 52.

LIM, Y.; PARK, K.; LEE, J. Y.; AOKI, S.; FERNANDO, G. Mmt: An emerging mpeg standard for multimedia delivery over the internet. *IEEE MultiMedia*, IEEE, v. 20, n. 1, p. 80–85, 2013. Cited 4 times on pages 11, 45, 51, and 52.

LIM, Y.-s.; CHEN, Y.-C.; NAHUM, E. M.; TOWSLEY, D.; LEE, K.-W. Cross-layer path management in multi-path transport protocol for mobile devices. p. 1815–1823, 2014. Cited on page 73.

LIU, Y.; GUO, Y.; LIANG, C. A survey on peer-to-peer video streaming systems. *Peer-to-peer Networking and Applications*, Springer, v. 1, n. 1, p. 18–28, 2008. Cited on page 41.

LUO, F.-L. *Digital Front-End in Wireless Communications and Broadcasting: circuits and signal processing*. [S.l.]: Cambridge University Press, 2011. Cited on page 34.

LUOTO, M.; RAUTIO, T.; OJANPERÄ, T.; MÄKELÄ, J. Distributed decision engine—an information management architecture for autonomie wireless networking. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, p. 713–719, 2015. Cited on page 61.

MAO, H.; NETRAVALI, R.; ALIZADEH, M. Neural adaptive video streaming with pensieve. In: ACM. *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. [S.l.], 2017. p. 197–210. Cited on page 61.

MAO, H.; NETRAVALI, R.; ALIZADEH, M. Neural adaptive video streaming with pensieve. In: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. New York, NY, USA: ACM, 2017. (SIGCOMM '17), p. 197–210. ISBN 978-1-4503-4653-5. Available from Internet: <http://doi.acm.org/10.1145/3098822.3098843>. Cited on page 107.

MAO, S.; BUSHMITCH, D.; NARAYANAN, S.; PANWAR, S. S. Mrtp: a multiflow real-time transport protocol for ad hoc networks. *IEEE Transactions on Multimedia*, IEEE, v. 8, n. 2, p. 356–369, 2006. Cited 15 times on pages 42, 54, 55, 56, 88, 91, 92, 93, 94, 95, 97, 98, 99, 102, and 114.

MITCHELL, B. *About 802.11g Wi-Fi for Wireless Computer Networking*. 2017. <https://www.lifewire.com/how-fast-is-802-11g-wifi-networking-818297>. Accessed:15-FEB-2017. Cited on page 34.

MITZENMACHER, M. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 12, n. 10, p. 1094–1104, 2001. Cited on page 95.

MOLDOVAN, A.-N.; GHERGULESCU, I.; MUNTEAN, C. H. Vqamap: A novel mechanism for mapping objective video quality metrics to subjective mos scale. *IEEE Transactions on Broadcasting*, IEEE, v. 62, n. 3, p. 610–627, 2016. Cited 2 times on pages 18 and 134.

MOVENETWORKS. *Move Networks*. 2017. <https://www.crunchbase.com/organization/movenetworks>. Cited on page 46.

MPEG. Information technology — high efficiency coding and media delivery in heterogeneous environments — part 13: 3rd edition mpeg media transport implementation guidelines. *ISO/IEC 23008-13*, 2013. Cited on page 49.

MPEG. Information technology – high efficiency coding and media delivery in heterogeneous environments – part 1: Mpeg media transport (mmt). *ISO/IEC 23008-1*, 2014. Cited 7 times on pages 11, 30, 47, 51, 63, 111, and 112.

MPEG. Mmt enhancements for mobile environments. *ISO/IEC 23008-1:2017/DAmd 2*, 2017. Cited 2 times on pages 28 and 47.

MPEG. Mpeg systems technologies – part 6: Dynamic adaptive streaming over http (dash). *ISO/IEC FCD 23001-6*, January 2011. Cited 2 times on pages 47 and 58.

MUELLER, C. *MPEG-DASH vs. Apple HLS vs. Microsoft Smooth Streaming vs. Adobe HDS*. 2017. <https://bitmovin.com/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/>. Cited on page 47.

MUSHROOM. *Mushroom networks Inc*. 2017. <http://www.mushroomnetworks.com/>. Accessed:15-JUL-2017. Cited on page 33.

MWELA, J. S. *Impact of packet loss on the quality of video stream transmission*. Tese (Doutorado) — Blekinge Institute of Technology, 2010. Cited on page 36.

NAKACHI, T.; YAMAGUCHI, T.; TONOMURA, Y.; FUJII, T. Next-generation media transport mmt for 4k/8k video transmission. *NTT Technical Review*, v. 12, n. 5, p. 1–7, 2013. Cited on page 92.

NAM, H.; CALIN, D.; SCHULZRINNE, H. Towards dynamic mptcp path control using sdn. *IEEE NetSoft Conference and Workshops (NetSoft)*, p. 286–294, 2016. Cited 10 times on pages 54, 61, 62, 73, 74, 86, 89, 91, 99, and 103.

NETEM. *NetEm*. 2017. <https://wiki.linuxfoundation.org/networking/netem>. Cited on page 105.

NETFLIX. *Meridian*. 2016. <https://www.netflix.com/br-en/title/80141336>. Cited on page 130.

NETFLIX. *Netflix*. 2019. <https://www.netflix.com/br-en/>. Accessed:06-may-2019. Cited on page 48.

NETFLIX. *VMAF - Video Multi-Method Assessment Fusion*. 2019. <https://github.com/Netflix/vmaf>. Accessed:10-SEP-2019. Cited on page 133.

NS2. *Network Simulator - NS-2*. 2017. <http://www.isi.edu/nsnam/ns/>. Cited on page 105.

NS3. *Direct Code Execution (DCE) for ns-3, Version 1.9*. 2016. Cited 2 times on pages 38 and 127.

NS3. *Network Simulator - NS-3*. 2017. <https://www.nsnam.org/>. Cited on page 105.

OH, B.-H.; LEE, J. Constraint-based proactive scheduling for mptcp in wireless networks. *Computer Networks*, Elsevier, v. 91, p. 548–563, 2015. Cited on page 127.

OJANPERÄ, T.; KOKKONIEMI-TARKKANEN, H. Wireless bandwidth management for multiple video clients through network-assisted dash. In: IEEE. *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 IEEE 17th International Symposium on A*. [S.l.], 2016. p. 1–3. Cited on page 61.

OJANPERÄ, T.; VEHKAPERÄ, J. Network-assisted multipath dash using the distributed decision engine. *IEEE International Conference on Computing, Networking and Communications (ICNC)*, p. 1–6, 2016. Cited 8 times on pages 54, 60, 61, 62, 84, 91, 97, and 104.

OKAMOTO, K.; YAMAI, N.; OKAYAMA, K.; KAWANO, K.; NAKAMURA, M.; YOKOHIRA, T. Performance improvement of sctp communication using selective bicasting on lossy multihoming environment. *IEEE 38th Annual Computer Software and Applications Conference (COMPSAC)*, p. 551–557, 2014. Cited 3 times on pages 54, 77, and 78.

OLIVEIRA J. F.; AFZAL, S.; TESTONI, V. *Método para Transmissão de Vídeo Digital Adotando Estratégias de Escalonamento com Reconhecimento de Percurso e Conteúdo em Redes Heterogêneas usando Protocolo MMT*. 2018. INPI BR 10 2018 070605 5. Cited 2 times on pages 40 and 172.

OLIVEIRA J. F.; AFZAL, S.; TESTONI, V. *Method for Digital Video Transmission Adopting Packaging Forwarding Strategies with Path and Content Monitoring in Heterogeneous Networks Using MMT Protocol, Method for Reception and Communication System*. 2019. US Patent 16387249. Cited 2 times on pages 40 and 172.

OPENSIGNAL. 2017. <https://opensignal.com/reports/2017/02/global-state-of-the-mobile-network>. Cited on page 127.

OPNET. *OPNET NETWORK SIMULATOR*. 2017. <http://opnetprojects.com/opnet-network-simulator/>. Cited on page 105.

PAASCH, C.; FERLIN, S.; ALAY, O.; BONAVENTURE, O. Experimental evaluation of multipath tcp schedulers. *Proceedings of the ACM SIGCOMM workshop on Capacity sharing workshop*, p. 27–32, 2014. Cited on page 72.

PANTOS, R.; MAY, W. *HTTP Live Streaming*. [S.l.]: RFC Editor, 2017. RFC 8216. (Request for Comments). Cited on page 46.

PARK, K.; KIM, N.; LEE, B.-D. Performance evaluation of the emerging media-transport technologies for the next-generation digital broadcasting systems. *IEEE Access*, IEEE, v. 5, p. 17597–17606, 2017. Cited on page 51.

PARMAR, H.; THORNBURGH, M. *Real-Time Messaging Protocol (RTMP) specification*. 2017. <http://www.adobe.com/devnet/rtmp.html>. Cited on page 45.

PAUL, A. K.; TACHIBANA, A.; HASEGAWA, T. An enhanced available bandwidth estimation technique for an end-to-end network path. *IEEE Transactions on Network and Service Management*, v. 13, n. 4, p. 768–781, 2016. Cited on page 98.

PINSON, M. H.; BARKOWSKY, M.; CALLET, P. L. Selecting scenes for 2d and 3d subjective video quality tests. *EURASIP Journal on Image and Video Processing*, Nature Publishing Group, v. 2013, n. 1, p. 50, 2013. Cited on page 130.

POLIAKOV, V.; SASSATELLI, L.; SAUCEZ, D. Adaptive video streaming and multipath: can less be more? In: *ICC 2018-IEEE International Conference on Communications*. [S.l.: s.n.], 2018. Cited on page 30.

POSTEL, J. *User Datagram Protocol*. 1980. 1–3 p. RFC 768. <http://www.rfc-editor.org/rfc/rfc768.txt>. Cited 2 times on pages 46 and 67.

POSTEL, J. *Transmission control protocol*. 1981. RFC 793. <http://www.rfc-editor.org/rfc/rfc793.txt>. Cited 2 times on pages 46 and 68.

PURKAYASTHA, D.; PERRAS, M.; RAHMAN, A. *Considerations for MPTCP operation in 5G*. [S.l.], 2017. Available from Internet: <http://www.ietf.org/internet-drafts/draft-purkayastha-mptcp-considerations-for-nextgen-00.txt>. Cited on page 106.

QADIR, J.; ALI, A.; YAU, K.-L. A.; SATHIASEELAN, A.; CROWCROFT, J. Exploiting the power of multiplicity: a holistic survey of network-layer multipath. *IEEE Communications Surveys & Tutorials*, v. 17, n. 4, p. 2176–2213, 2015. Cited 4 times on pages 30, 32, 41, and 42.

RAICIU, C.; HANDLEY, M.; WISCHIK, D. *Coupled Congestion Control for Multipath Transport Protocols*. [S.l.]: Internet Engineering Task Force, 2011. RFC 6356. Cited on page 72.

RAICIU, C.; PAASCH, C.; BARRE, S.; FORD, A.; HONDA, M.; DUCHENE, F.; BONAVENTURE, O.; HANDLEY, M. How hard can it be? designing and implementing a deployable multipath tcp. *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, p. 29–29, 2012. Cited on page 66.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. *The Addition of Explicit Congestion Notification (ECN) to IP*. 2001. RFC 3168. Cited on page 104.

RAO, K.; KIM, D. N.; HWANG, J. J. Vp6 video coding standard. In: *Video coding standards*. [S.l.]: Springer, 2014. p. 159–197. Cited on page 47.

RIBEIRO, E. P.; LEUNG, V. C. Minimum delay path selection in multi-homed systems with path asymmetry. *IEEE Communications Letters*, IEEE, v. 10, n. 3, p. 135–137, 2006. Cited on page 96.

RIBEIRO, V. J.; RIEDI, R. H.; BARANIUK, R. G.; NAVRATIL, J.; COTTRELL, L. pathchirp: Efficient available bandwidth estimation for network paths. In: *Passive and active measurement workshop*. [S.l.: s.n.], 2003. Cited on page 98.

RIJKSE, K. H. 263: video coding for low-bit-rate communication. *IEEE Communications magazine*, v. 34, n. 12, p. 42–45, 1996. Cited on page 105.

SAMBA, A.; BUSNEL, Y.; BLANC, A.; DOOZE, P.; SIMON, G. Instantaneous throughput prediction in cellular networks: Which information is needed? *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2017. Cited on page 60.

SANI, Y.; MAUTHE, A.; EDWARDS, C. Adaptive bitrate selection: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 19, n. 4, p. 2985–3014, 2017. Cited 5 times on pages 29, 36, 60, 61, and 69.

SANSON, H.; NEIRA, A.; LOYOLA, L.; MATSUMOTO, M. Pr-sctp for real time h. 264/avc video streaming. *IEEE 12th International Conference on Advanced Communication Technology (ICACT)*, v. 1, p. 59–63, 2010. Cited 4 times on pages 54, 78, 79, and 103.

SCHAAR, M. Van der; CHOU, P. A. *Multimedia over IP and wireless networks: compression, networking, and systems*. [S.l.]: Elsevier, 2011. Cited on page 36.

SCHULZRINNE and H.; CASNER, S.; FREDERICK, R.; JACOBSON, V. RFC, *RTP: A Transport Protocol for Real-Time Applications*. 1996. Internet Requests for Comments. Cited on page 54.

SCHULZRINNE, H. *A Transport Protocol for Real-Time Applications*. 1992. Cited 2 times on pages 45 and 54.

SCHULZRINNE, H. *Real time streaming protocol (RTSP)*. 1998. RFC 2326. Cited on page 55.

SCHULZRINNE, H.; CASNER, S.; FREDERICK, R.; JACOBSON, V. *RTP: A Transport Protocol for Real-Time Applications*. 2003. Internet Requests for Comments. Cited on page 54.

SCHWARZ, H.; MARPE, D.; WIEGAND, T. Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, IEEE, v. 17, n. 9, p. 1103–1120, 2007. Cited on page 105.

SEO, S. Kt's giga lte. *IETF93*, 2015. Cited on page 71.

SEUFERT, M.; EGGER, S.; SLANINA, M.; ZINNER, T.; HOBFELD, T.; TRAN-GIA, P. A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 1, p. 469–492, 2015. Cited 5 times on pages 11, 46, 47, 58, and 59.

SHA, K.; GEHLOT, J.; GREVE, R. Mutipath routing techniques in wireless sensor networks: A survey. *Wireless personal communications*, Springer, v. 70, n. 2, p. 807–829, 2013. Cited on page 41.

SHAHID, M. *Methods for Objective and Subjective Video Quality Assessment and for Speech Enhancement*. Tese (Doutorado) — Blekinge Institute of Technology, 2014. Cited on page 132.

SHARMA, V.; KALYANARAMAN, S.; KAR, K.; RAMAKRISHNAN, K.; SUBRAMANIAN, V. Mplot: A transport protocol exploiting multipath diversity using erasure codes. *27th Conference on Computer Communications, IEEE INFOCOM*, p. 121–125, 2008. Cited 11 times on pages 42, 54, 69, 91, 93, 95, 97, 98, 101, 102, and 104.

SHOKROLLAHI, A. Raptor codes. *IEEE transactions on information theory*, IEEE, v. 52, n. 6, p. 2551–2567, 2006. Cited on page 92.

SHUAI, Y.; PETROVIC, G.; HERFET, T. Server-driven rate control for adaptive video streaming using virtual client buffers. In: IEEE. *Consumer Electronics–Berlin (ICCE-Berlin), 2014 IEEE Fourth International Conference on*. [S.l.], 2014. p. 45–49. Cited on page 61.

SILVA, C. A. G. da; RIBEIRO, E. P.; PEDROSO, C. M. Preventing quality degradation of video streaming using selective redundancy. *Computer Communications*, Elsevier, v. 91, p. 120–132, 2016. Cited 8 times on pages 54, 77, 78, 88, 90, 91, 92, and 103.

SINGH, A.; GOERG, C.; TIMM-GIEL, A.; SCHARF, M.; BANNIZA, T.-R. Performance comparison of scheduling algorithms for multipath transfer. p. 2653–2658, 2012. Cited 3 times on pages 59, 72, and 73.

SINGH, S. K.; DAS, T.; JUKAN, A. A survey on internet multipath routing and provisioning. *IEEE Communications Surveys & Tutorials*, IEEE, v. 17, n. 4, p. 2157–2175, 2015. Cited 5 times on pages 33, 41, 42, 55, and 105.

SINGH, V.; AHSAN, S.; OTT, J. Mprtp: multipath considerations for real-time media. *Proceedings of the 4th ACM Multimedia Systems Conference*, p. 190–201, 2013. Cited 15 times on pages 42, 54, 55, 56, 58, 88, 89, 91, 97, 98, 99, 101, 102, 104, and 114.

SODAGAR, I. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, IEEE, v. 18, n. 4, p. 62–67, 2011. Cited on page 46.

SOHN, Y.; CHO, M.; SEO, M.; PAIK, J. A synchronization scheme for hierarchical video streams over heterogeneous networks. *KSII Transactions on Internet & Information Systems*, v. 9, n. 8, 2015. Cited 6 times on pages 54, 64, 88, 91, 94, and 102.

SOUSA, N. F. S. de; PEREZ, D. A. L.; ROSA, R. V.; SANTOS, M. A.; ROTHENBERG, C. E. Network service orchestration: A survey. *Computer Communications*, v. 142-143, p. 69 – 94, 2019. ISSN 0140-3664. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S0140366418309502>. Cited on page 106.

STEWART, R. Stream control transmission protocol. *RFC 4960*, September, 2007. Cited 2 times on pages 76 and 77.

STEWART, R.; ARIAS-RODRIGUEZ, I.; POON, K.; CARO, A.; TUEXEN, M. *Stream Control Transmission Protocol (SCTP) Specification Errata and Issues*. 2006. RFC 4460. Cited on page 76.

STEWART, R.; RAMALHO, M.; XIE, Q.; TUEXEN, M.; CONRAD, P. *Stream Control Transmission Protocol (SCTP) Partial Reliability Extension*. [S.l.]: RFC Editor, 2004. RFC 3758. Cited on page 79.

STEWART, R.; XIE, Q.; MORNEAULT, K.; SHARP, C.; SCHWARZBAUER, H.; TAYLOR, T.; RYTINA, I.; KALIA, M.; ZHANG, L.; PAXSON, V. *Stream control transmission protocol*. 2000. RFC 2960. <https://tools.ietf.org/html/rfc2960>. Cited on page 76.

STONE, J.; STEWART, R.; OTIS, D. *Stream Control Transmission Protocol (SCTP) Checksum Change*. [S.l.]: IETF Network Working Group, 2002. RFC 3309. <https://tools.ietf.org/html/rfc3309>. Cited on page 76.

SUEHRING, K. *Abing project page*. 2017. Stanford University, CA. <http://iphome.hhi.de/suehring/tml/download/>. Cited on page 98.

SUN, G.; ENG, K.; YIN, S.; LIU, G.; MIN, G. Mars: Multiple access radio scheduling for a multi-homed mobile device in soft-ran. *KSII Transactions on Internet & Information Systems*, v. 10, n. 1, 2016. Cited 8 times on pages 35, 54, 82, 89, 91, 96, 99, and 103.

SWAMINATHAN, V.; WEI, S. Low latency live video streaming using http chunked encoding. *IEEE 13th International Workshop on Multimedia Signal Processing (MMSP)*, p. 1–6, 2011. Cited on page 59.

THANG, T. C.; LE, H. T.; PHAM, A. T.; RO, Y. M. An evaluation of bitrate adaptation methods for http live streaming. *IEEE Journal on Selected Areas in Communications*, IEEE, v. 32, n. 4, p. 693–705, 2014. Cited on page 60.

THOMAS, E.; DEVENTER, M. van; STOCKHAMMER, T.; BEGEN, A. C.; CHAMPEL, M.-L.; OYMAN, O. Applications and deployments of server and network assisted dash (sand). IET, 2016. Cited on page 61.

TRESTIAN, R.; COMSA, I.-S.; TUYSUZ, M. F. Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet? *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 2, p. 945–977, 2018. Cited 6 times on pages 28, 36, 41, 42, 48, and 107.

TRESTIAN, R.; MOLDOVAN, A.-N.; ORMOND, O.; MUNTEAN, G.-M. Energy consumption analysis of video streaming to android mobile devices. In: IEEE. *Network Operations and Management Symposium (NOMS), 2012 IEEE*. [S.l.], 2012. p. 444–452. Cited on page 60.

TUEXEN, M.; SEGGELMANN, R.; STEWART, R.; LORETO, S. *Additional Policies for the Partially Reliable Stream Control Transmission Protocol Extension*. [S.l.]: Internet Engineering Task Force, 2015. RFC 7496. Cited on page 79.

TWITCH. *Twitch*. 2019. <https://www.twitch.tv/>. Accessed:06-may-2019. Cited on page 48.

USMAN, U. K.; PRIHATMOKO, G. Alternative frequency selection of long term evolution (lte) technology in indonesia. In: *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*. [S.l.: s.n.], 2011. Cited on page 127.

UTEXAS. *LIVE*. 2018. <http://live.ece.utexas.edu/research/Quality/index.htm>. Cited on page 130.

VASILEV, V.; LEGUAY, J.; PARIS, S.; MAGGI, L.; DEBBAH, M. Predicting qoe factors with machine learning. In: *IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2018. p. 1–6. Cited on page 107.

VIEIRA, F. M. d. C. R. *Systematic analysis of super hi-vision systems*. Dissertação (Mestrado) — FEUP - Faculdade de Engenharia da Universidade do Porto, 2014. Cited on page 29.

VIMEO. *Vimeo*. 2019. <https://vimeo.com/>. Accessed:06-may-2019. Cited on page 48.

VRANJEŠ, M.; RIMAC-DRLJE, S.; GRGIĆ, K. Review of objective video quality metrics and performance comparison using different databases. *Signal Processing: Image Communication*, Elsevier, v. 28, n. 1, p. 1–19, 2013. Cited on page 132.

WALLACE, T. D.; SHAMI, A. A review of multihoming issues using the stream control transmission protocol. *IEEE Communications Surveys & Tutorials*, IEEE, v. 14, n. 2, p. 565–578, 2012. Cited 2 times on pages 93 and 101.

WANG, H.; JIN, Y.; WANG, W.; MA, J.; ZHANG, D. The performance comparison of prsctp, tcp and udp for mpeg-4 multimedia traffic in mobile network. *IEEE International Conference on Communication Technology Proceedings(ICCT)*, v. 1, p. 403–406, 2003. Cited on page 79.

WANG, Z.; BOVIK, A. C.; SHEIKH, H. R.; SIMONCELLI, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, IEEE, v. 13, n. 4, p. 600–612, 2004. Cited on page 105.

WANG, Z.; SIMONCELLI, E. P.; BOVIK, A. C. Multiscale structural similarity for image quality assessment. *IEEE Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, v. 2, p. 1398–1402, 2003. Cited on page 105.

WEBRTC. *Web Real-Time Communication*. 2017. <http://www.webrtc.org/>. Accessed:15-JUL-2017. Cited on page 68.

WEI, S.; SWAMINATHAN, V. Low latency live video streaming over http 2.0. *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, p. 37, 2014. Cited on page 59.

WILK, S.; STOHR, D.; EFFELSBERG, W. Vas: a video adaptation service to support mobile video. In: ACM. *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. [S.l.], 2015. p. 37–42. Cited on page 61.

WISCHIK, D.; RAICIU, C.; GREENHALGH, A.; HANDLEY, M. Design, implementation and evaluation of congestion control for multipath tcp. *Networked Systems Design and Implementation (NSDI)*, v. 11, p. 8–8, 2011. Cited on page 72.

WU, J.; CHENG, B.; YUEN, C.; SHANG, Y.; CHEN, J. Distortion-aware concurrent multipath transfer for mobile video streaming in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, IEEE, v. 14, n. 4, p. 688–701, 2015. Cited 15 times on pages 54, 78, 79, 80, 88, 91, 94, 96, 97, 98, 99, 100, 101, 103, and 104.

WU, J.; SHANG, Y.; HUANG, J.; ZHANG, X.; CHENG, B.; CHEN, J. Joint source-channel coding and optimization for mobile video streaming in heterogeneous wireless networks. *European Association for Signal Processing (EURASIP) Journal on Wireless Communications and Networking*, v. 2013, n. 1, p. 283, 2013. Cited 14 times on pages 54, 85, 89, 92, 93, 94, 95, 97, 98, 99, 101, 103, 157, and 197.

WU, J.; YUEN, C.; CHENG, B.; SHANG, Y.; CHEN, J. Goodput-aware load distribution for real-time traffic over multipath networks. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 26, n. 8, p. 2286–2299, 2015. Cited 12 times on pages 54, 85, 89, 92, 93, 94, 96, 97, 98, 99, 103, and 104.

WU, J.; YUEN, C.; CHENG, B.; YANG, Y.; WANG, M.; CHEN, J. Bandwidth-efficient multipath transport protocol for quality-guaranteed real-time video over heterogeneous wireless networks. *IEEE Transactions on Communications*, IEEE, v. 64, n. 6, p. 2477–2493, 2016. Cited 19 times on pages 28, 36, 54, 67, 68, 74, 88, 89, 90, 91, 92, 93, 96, 97, 98, 99, 102, 104, and 114.

WU, J.; YUEN, C.; CHENG, B.; WANG, M.; CHEN, J. Streaming high-quality mobile video with multipath tcp in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, IEEE, v. 15, n. 9, p. 2345–2361, 2016. Cited 17 times on pages 54, 73, 74, 88, 91, 92, 93, 94, 95, 96, 97, 98, 99, 101, 102, 104, and 117.

WU, J.; YUEN, C.; CHEUNG, N.-M.; CHEN, J. Delay-constrained high definition video transmission in heterogeneous wireless networks with multi-homed terminals. *IEEE Transactions on Mobile Computing*, IEEE, v. 15, n. 3, p. 641–655, 2016. Cited 2 times on pages 98 and 119.

WU, J.; YUEN, C.; WANG, M.; CHEN, J. Content-aware concurrent multipath transfer for high-definition video streaming over heterogeneous wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 27, n. 3, p. 710–723, 2016. Cited 14 times on pages 54, 78, 79, 80, 88, 90, 91, 96, 97, 98, 99, 101, 103, and 104.

WU, K.; XIAO, J.; YI, Y.; CHEN, D.; LUO, X.; NI, L. M. Csi-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems*, IEEE, v. 24, n. 7, p. 1300–1309, 2013. Cited on page 87.

X264. 2017. <http://www.videolan.org/developers/x264.html>. Cited on page 105.

XIAO, M.; SWAMINATHAN, V.; WEI, S.; CHEN, S. Evaluating and improving push based video streaming with http/2. *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, p. 3, 2016. Cited 3 times on pages 46, 59, and 60.

XING, M.; XIANG, S.; CAI, L. Rate adaptation strategy for video streaming over multiple wireless access networks. *IEEE Global Communications Conference (GLOBECOM)*, p. 5745–5750, 2012. Cited 10 times on pages 54, 59, 61, 62, 88, 89, 91, 99, 102, and 107.

XING, S. X. M.; CAI, L. A real-time adaptive algorithm for video streaming over multiple wireless access networks. *IEEE Journal on Selected Areas in communications*, IEEE, v. 32, n. 4, p. 795–805, 2014. Cited 13 times on pages 42, 54, 59, 61, 62, 88, 89, 91, 94, 98, 102, 107, and 118.

XU, C.; HUANG, H.; ZHANG, H.; XIONG, C.; ZHU, L. Multipath transmission control protocol (mptcp) partial reliability extension. 2016. Cited 4 times on pages 54, 73, 74, and 75.

XU, C.; LI, Z.; LI, J.; ZHANG, H.; MUNTEAN, G.-M. Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 25, n. 7, p. 1175–1189, 2015. Cited 10 times on pages 54, 80, 87, 89, 91, 97, 98, 101, 103, and 104.

XU, C.; LIU, T.; GUAN, J.; ZHANG, H.; MUNTEAN, G.-M. Cmt-qa: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, IEEE, v. 12, n. 11, p. 2193–2205, 2013. Cited 14 times on pages 34, 54, 78, 79, 80, 88, 91, 96, 97, 98, 101, 103, 104, and 127.

YAHIA, M. B.; LOUEDEC, Y. L.; NUAYMI, L.; SIMON, G. When http/2 rescues dash: Video frame multiplexing. *3rd Communication and Networking Techniques for Contemporary Video Workshop (CNTCV), IEEE INFOCOM*, 2017. Cited 2 times on pages 46 and 47.

Yan, M.; Casey, J.; Shome, P.; Sprintson, A.; Sutton, A. Ætherflow: Principled wireless support in sdn. In: *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*. [S.l.: s.n.], 2015. p. 432–437. ISSN 1092-1648. Cited on page 106.

YAP, K.-K.; HUANG, T.-Y.; KOBAYASHI, M.; YIAKOUMIS, Y.; MCKEOWN, N.; KATTI, S.; PARULKAR, G. Making use of all the networks around us: a case study in android. *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, p. 19–24, 2012. Cited 6 times on pages 32, 54, 82, 89, 91, and 103.

YE, Y.; HE, Y.; WANG, Y.-K. *et al.* Shvc the scalable extensions of hevc and its applications. *ZTE Communications*, v. 14, n. 1, 2016. Cited 2 times on pages 47 and 48.

YIE, C. K.; LEE, Y. J. Method for hybrid delivery of mmt package and content and method for receiving content. *US Patent No. 9,414,123*, 2016. Cited on page 45.

YOON, J.; ZHANG, H.; BANERJEE, S.; RANGARAJAN, S. Muvi: A multicast video delivery scheme for 4g cellular networks. *Proceedings of the 18th annual international conference on Mobile computing and networking*, p. 209–220, 2012. Cited on page 34.

YOUTUBE. *YouTube*. 2019. <https://www.youtube.com/>. Accessed:06-may-2019. Cited on page 48.

YUSTE, L. B.; SEGUI, F. B.; CLIMENT, M. A. M.; MELVIN, H. Understanding timelines within mpeg standards. In: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS (IEEE). *Communications Surveys and Tutorials, IEEE Communications Society.* [S.l.], 2015. v. 18, n. 1, p. 368–400. Cited 2 times on pages 44 and 112.

ZAHRAN, A. H.; QUINLAN, J. J.; RAMAKRISHNAN, K.; SREENAN, C. J. Asap: Adaptive stall-aware pacing for improved dash video experience in cellular networks. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 14, n. 3s, p. 61, 2018. Cited on page 28.

ZHAI, F.; EISENBERG, Y.; PAPPAS, T. N.; BERRY, R.; KATSAGGELOS, A. K. Rate-distortion optimized product code forward error correction for video transmission over ip-based wireless networks. In: *EEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. [S.l.: s.n.], 2004. v. 5, p. V–857. Cited on page 92.

ZHANG, X.; HASSANEIN, H. A survey of peer-to-peer live video streaming schemes - an algorithmic perspective. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 56, n. 15, p. 3548–3579, out. 2012. ISSN 1389-1286. Available from Internet: <http://dx.doi.org/10.1016/j.comnet.2012.06.013>. Cited on page 41.

ZHOU, A.; LIU, M.; SONG, Y.; LI, Z.; DENG, H.; MA, Y. A new method for end-to-end available bandwidth estimation. In: *IEEE Global Communications Conference (GLOBECOM)*. [S.l.: s.n.], 2008. p. 1–5. Cited on page 98.

ZHOU, Y.; DUAN, Y.; SUN, J.; GUO, Z. Towards simple and smooth rate adaption for vbr video in dash. In: IEEE. *Visual Communications and Image Processing Conference, 2014 IEEE*. [S.l.], 2014. p. 9–12. Cited on page 60.

ANNEX A – Random Wireless Loss Rate - Parameter Choice

Although related work (CHEN *et al.*, 2013) assumes loss rate values of 1% for the WiFi path and 0.1% for the LTE path, we opted to reduce the maximum random loss rate of WiFi to 0.5% which is still a realistic assumption. The reason is that the results of our simulations, where packets are lost for WiFi, become extremely high. Here, we show PSNR results for *Elephants Dream* sequence, as an example, under the congested network scenario, explained in Subsection 4.4.1.1, with different loss rates for WiFi. Such results show the effect of different WiFi loss rate values on the QoE. We chose this sequence due to its original PSNR is in the middle of other sequences (47.07 dB). Three network congestion parts are defined for this scenario; LTE congestion, WiFi congestion, LTE and WiFi congestion. Table 24 shows the average PSNR results and correspondence MOS for each part. One note is that when WiFi loss rate is 1%, PSNR results are smaller in all three congested parts compared to when WiFi loss rate is 0.5%. As expected, when the WiFi loss rate is 1%, it has the worst PSNR in the first part, when LTE is congested. The reason is that when LTE is congested, scheduler switches most of the video data through the WiFi and since WiFi has a high loss rate, many packets would be lost. Therefore, in this part of simulation time, the PSNR is 25.04 dB maps to *"Fair"* in MOS quality, even closer to the *poor* than to the *"Good"*, but this value is 31.72 dB maps to *"Good"* in MOS quality when WiFi loss rate is 0.5% which is a significant value of 6.68 dB more.

Table 24 – The average PSNR results from the *Elephants Dream* sequence according to CAPA under congested network scenario with different paths' situations; LTE congestion, WiFi congestion, LTE and WiFi congestion.

Random wireless loss rate	**PSNR [dB] and correspondence MOS**		
	LTE congestion	**WiFi congestion**	**LTE and WiFi congestion**
WiFi = 1% , LTE = 0.1%	25.04 (Fair)	38.22 (Excellent)	32.52 (Good)
WiFi = 0.5% , LTE = 0.1%	31.72 (Good)	39.28 (Excellent)	35.93 (Good)

ANNEX B – Burst Wireless Loss Rate - Parameter Choice

Although related work (WU *et al.*, 2013) assumes loss rates of 5% for the WiFi and 2% for the LTE path. However, for most of our simulations and most of our sequences, the losses for WiFi and LTE were too high. Therefore, we kept the proportion between the loss rates of WiFi and LTE but reduced to 1% for WiFi, and 0.4% for LTE. Here, we show PSNR results for *Elephants Dream* sequence, as an example, under the wireless lossy network scenario explained in Subsection 4.4.2.1 to show the effect of channel loss rates on the results. We chose this sequence due to its original PSNR is in the middle of other sequences (47.07 dB). In the wireless lossy network scenario, three wireless loss parts are defined; LTE lossy channel, WiFi lossy channel, LTE and WiFi lossy channels. Table 25 shows the average PSNR results for each part considering different channel loss rates. One note is that when loss rates are 2% and 0.8% respectively for WiFi and LTE, PSNRs are smaller in all three parts compared to when loss rates are 1% and 0.4%. As expected, when loss rates are 2% and 0.8% respectively for WiFi and LTE, it has the worst PSNR in the third part, when both channels are lossy. Therefore, the losses for WiFi and LTE are too high with sever adverse effect on PSNR. In this part of simulation time, the PSNR is 21.53 dB maps to *"Poor"* in MOS quality, but this value is 26.89 dB maps to *"Fair"* in MOS quality when loss rates are 1% and 0.4% respectively for WiFi and LTE which is a significant difference of 5.36 dB.

Table 25 – The average PSNR results from the *Elephants Dream* sequence according to CAPA under wireless lossy network scenario with different channels' situations; LTE lossy channel, WiFi lossy channel, LTE and WiFi lossy channels.

Burst wireless loss rate	**PSNR [dB] and correspondence MOS**		
	LTE lossy channel	**WiFi lossy channel**	**LTE and WiFi lossy channels**
WiFi = 2% , LTE = 0.8%	32.29 (Good)	30.33 (Good)	21.53 (Poor)
WiFi = 1% , LTE = 0.4%	36.09 (Good)	30.55 (Good)	26.89 (Fair)

ANNEX C – Goodput Performance for Constant Transmission Bit Rate under Congested Network Scenario

Here, Figures 72, 73, 74, 75 show the LTE, WiFi and total (joint) goodput measured according to the different scheduling strategies, CAPA, PA, and ES, respectively for *Meridian*, *Sintel*, *LIVE*, and *Big Buck Bunny*.

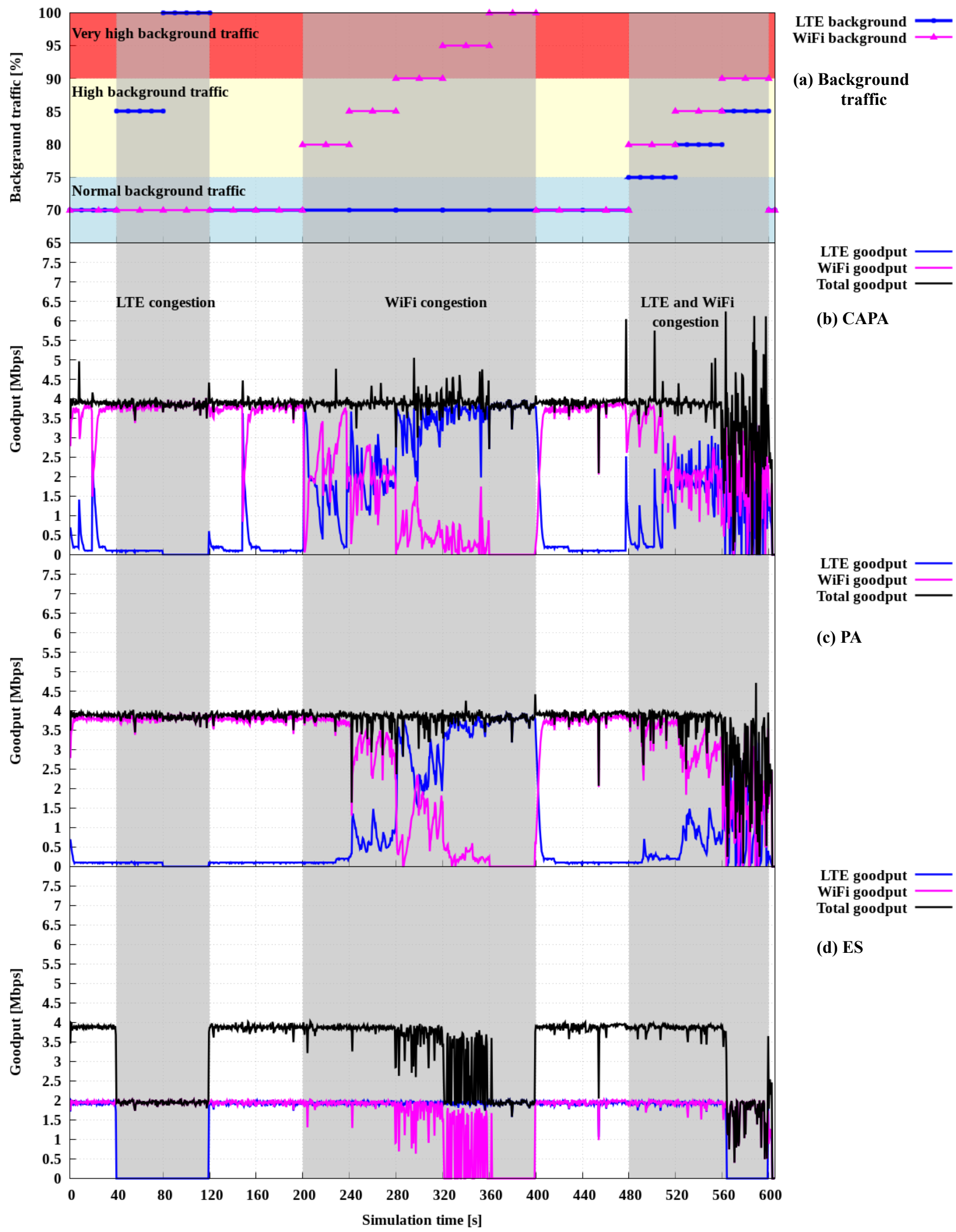


Figure 72 – LTE, WiFi and total (joint) goodput for *Meridian* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

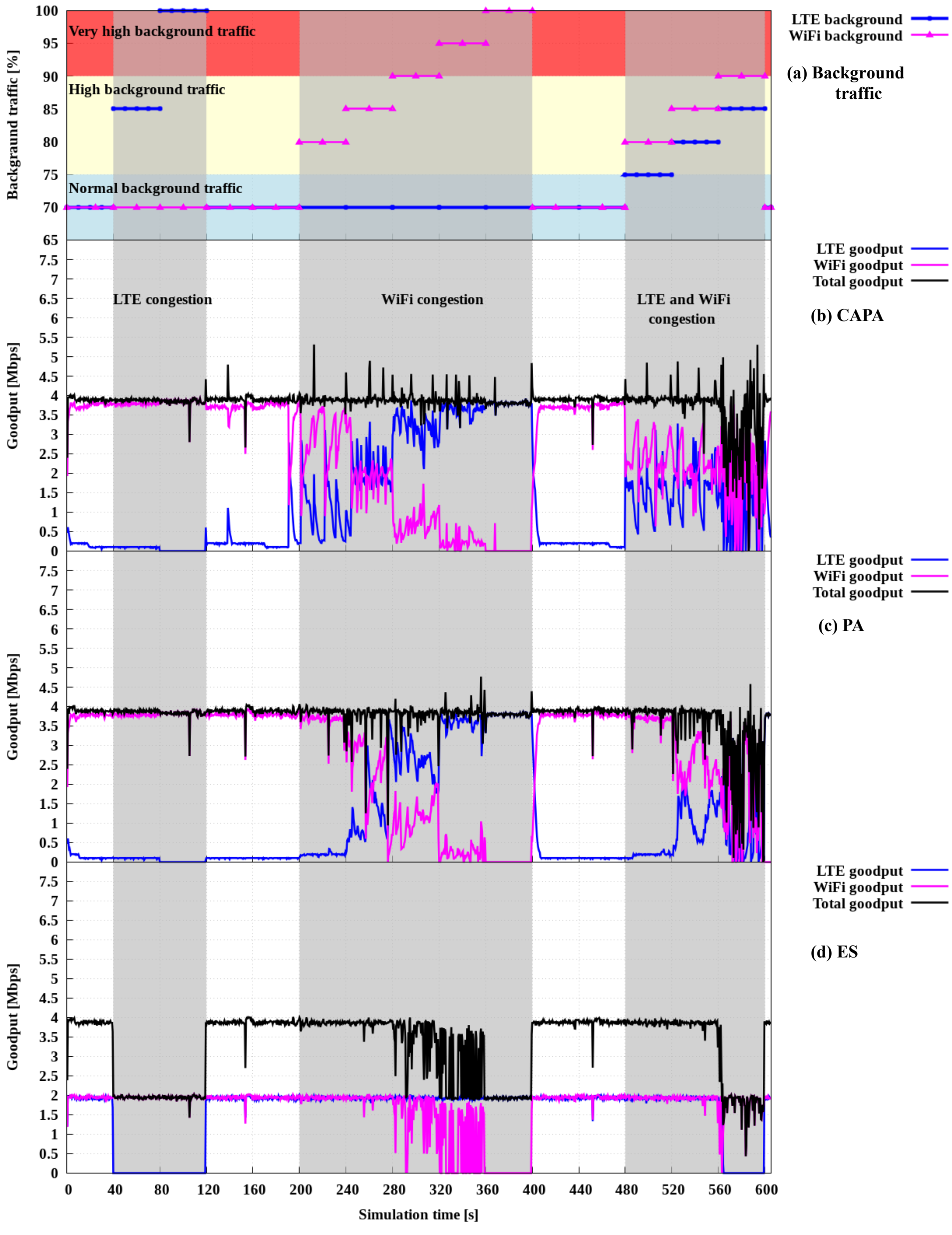


Figure 73 – LTE, WiFi and total (joint) goodput for *Sintel* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

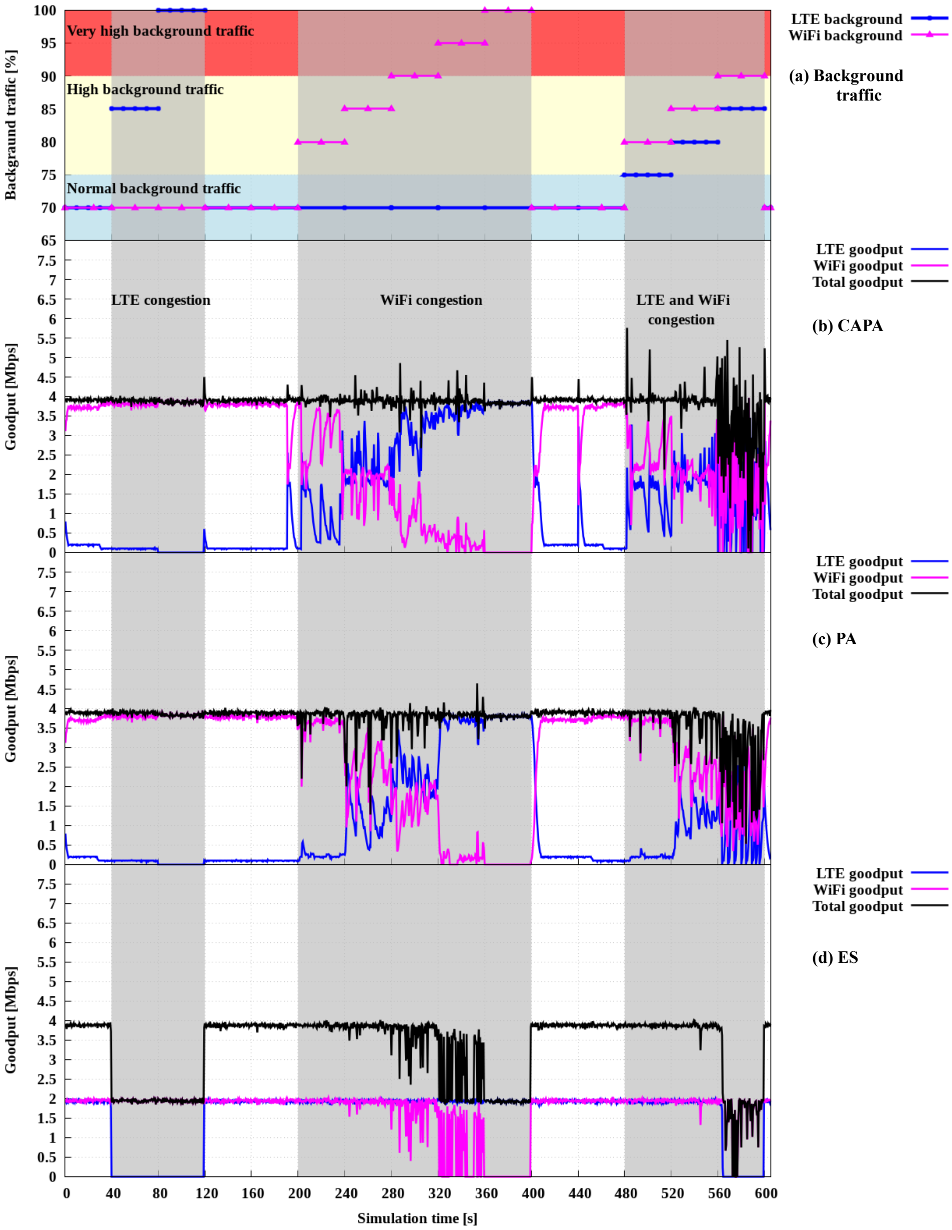


Figure 74 – LTE, WiFi and total (joint) goodput for *LIVE* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

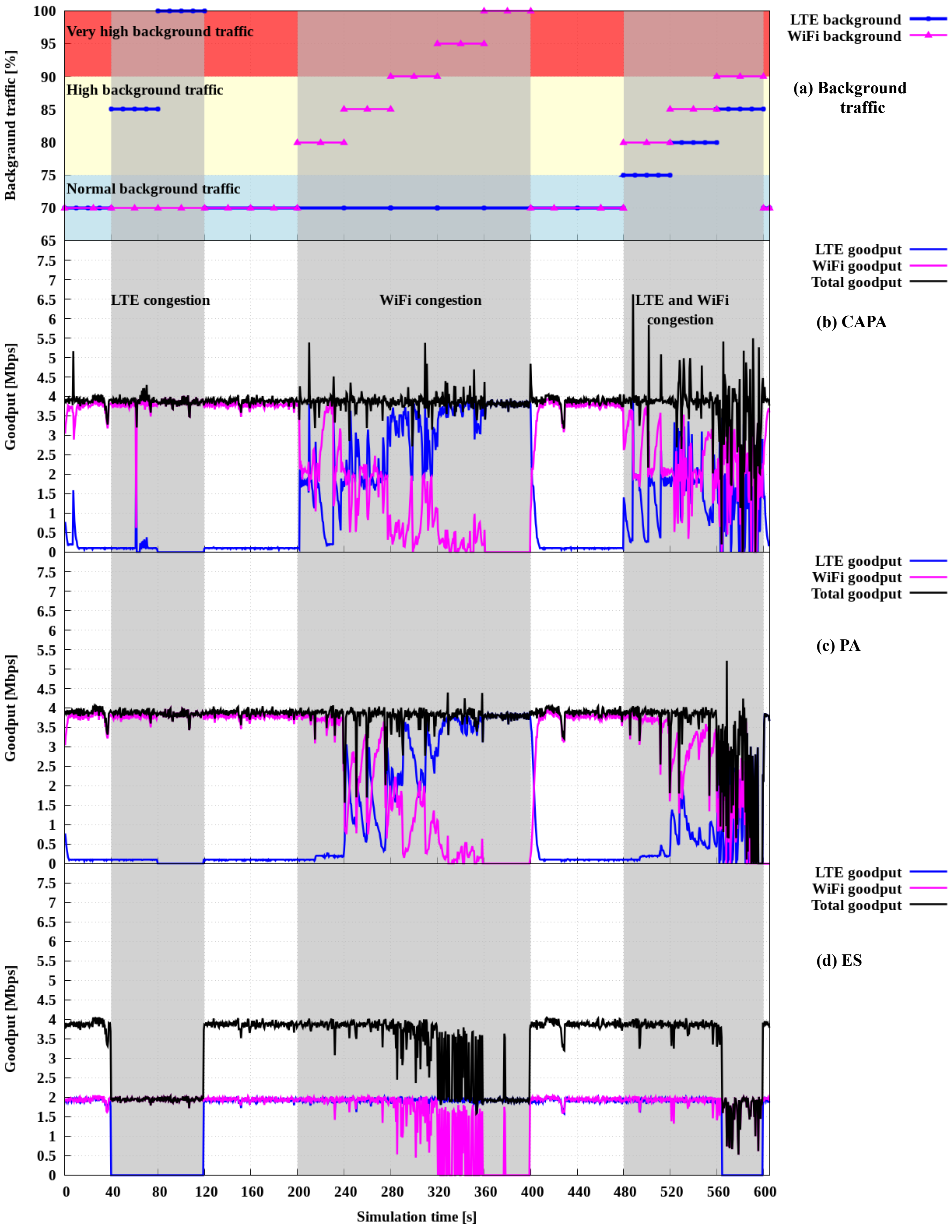


Figure 75 – LTE, WiFi and total (joint) goodput for *Big Buck Bunny* packets for (a) the defined background traffic according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

ANNEX D – PSNR Values for Constant Transmission Bit Rate under Congested Network Scenario

Here, Figures 76, 77, 78, 79 show the PSNR values for all video frames according to the different scheduling strategies, CAPA, PA, and ES, respectively for *Meridian*, *Sintel*, *LIVE*, and *Big Buck Bunny*.

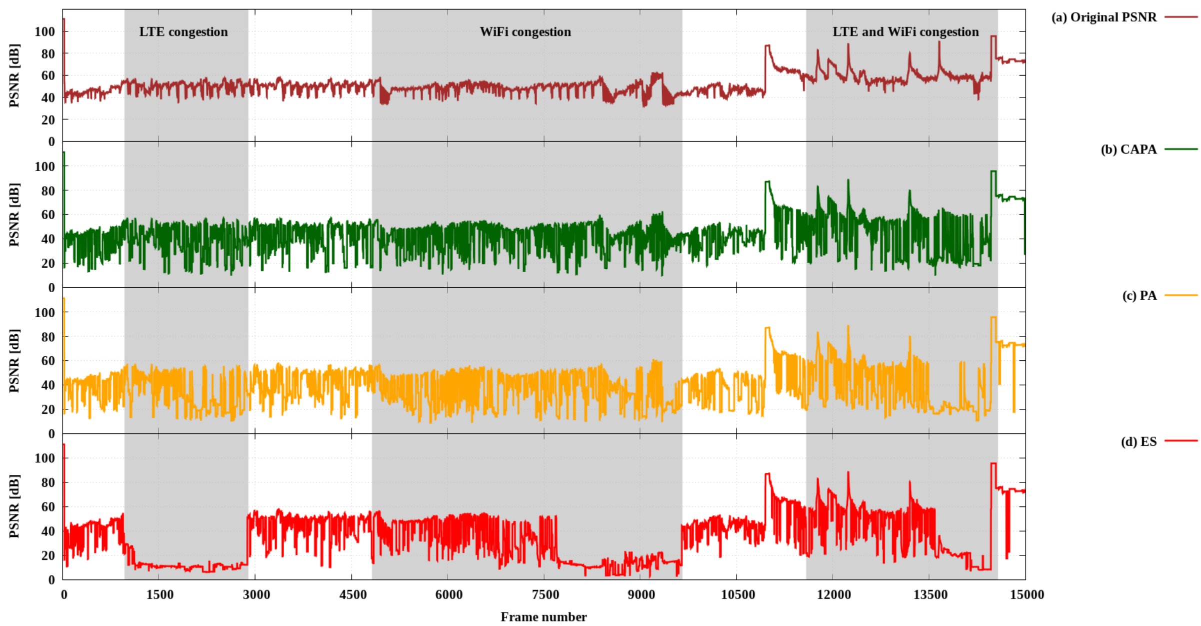


Figure 76 – PSNR values from *Meridian* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

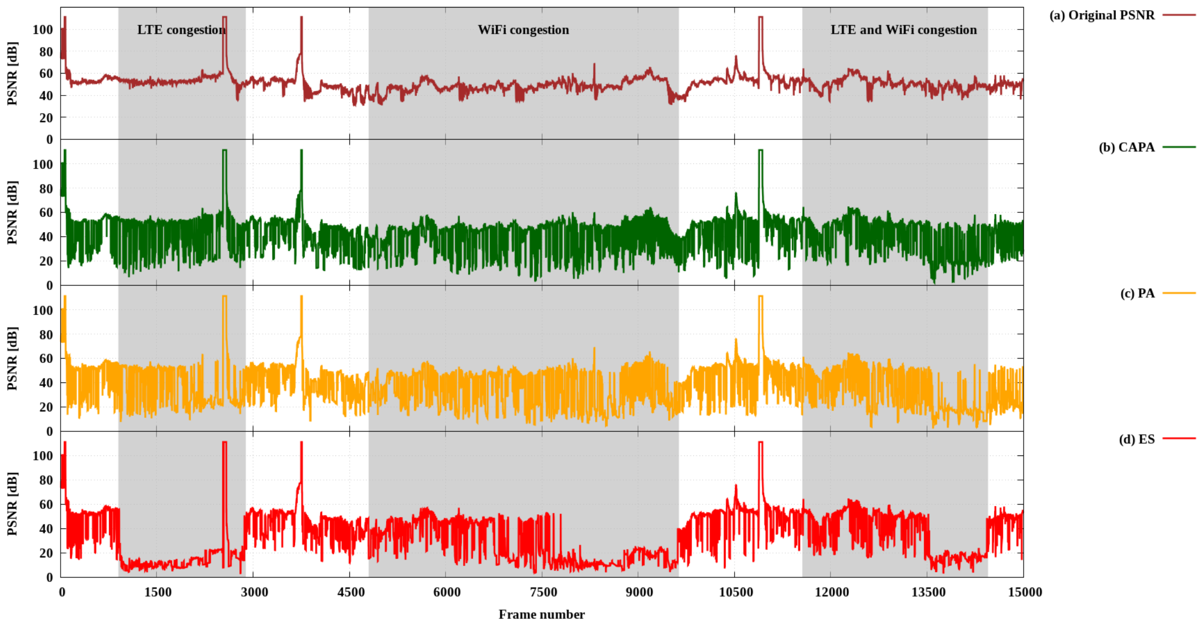


Figure 77 – PSNR values from *Sintel* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

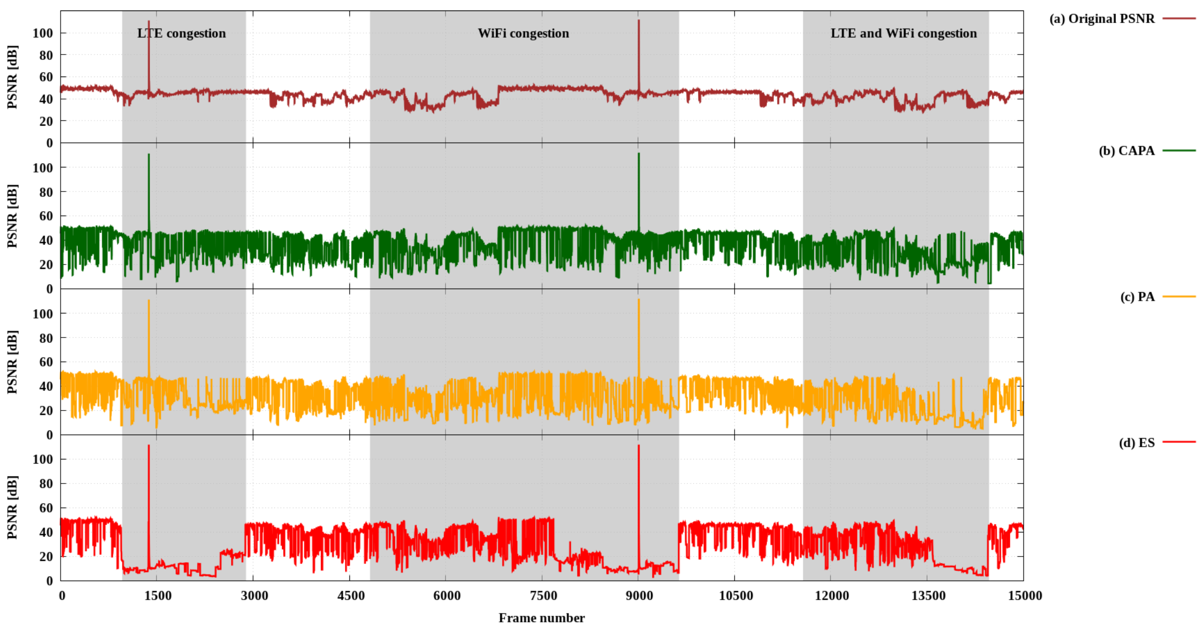


Figure 78 – PSNR values from *LIVE* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

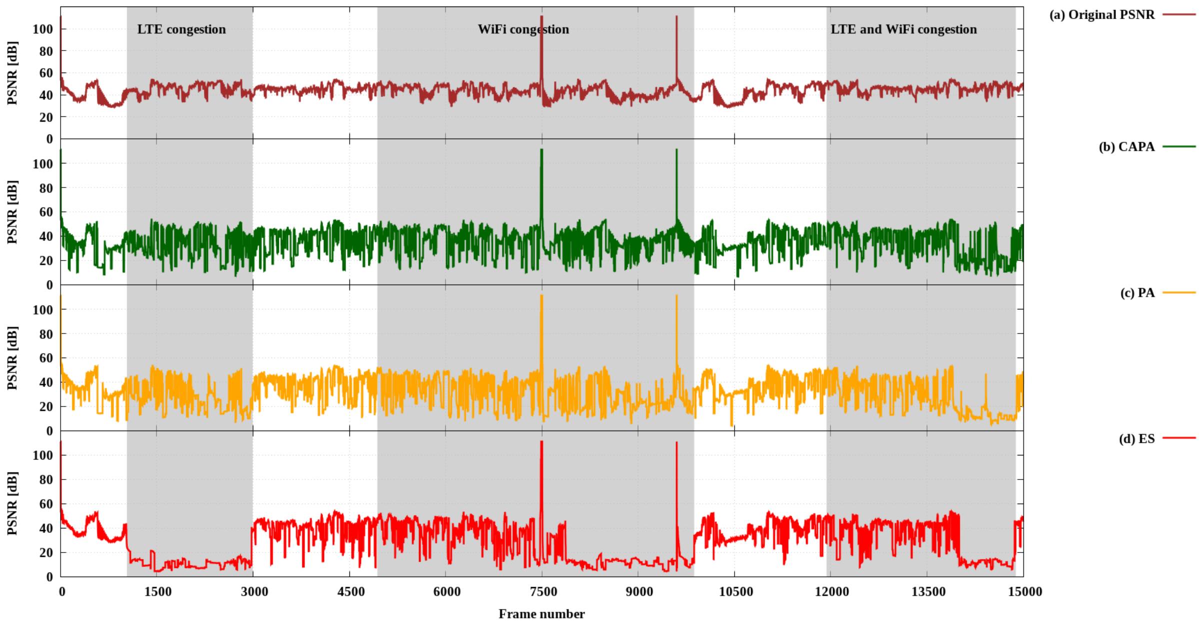


Figure 79 – PSNR values from *Big Buck Bunny* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

ANNEX E – Goodput Performance for Different Video Bit Rates Under Congested Network Scenario

Here, Figures 80, 81 and 82 show the LTE, WiFi and total (joint) goodput for *Meridian* with bit rate of 3 Mbps and *Big Buck Bunny* with bit rates of 3 and 5 Mbps according to the different scheduling strategies; CAPA, PA, and ES, under congested network scenario.

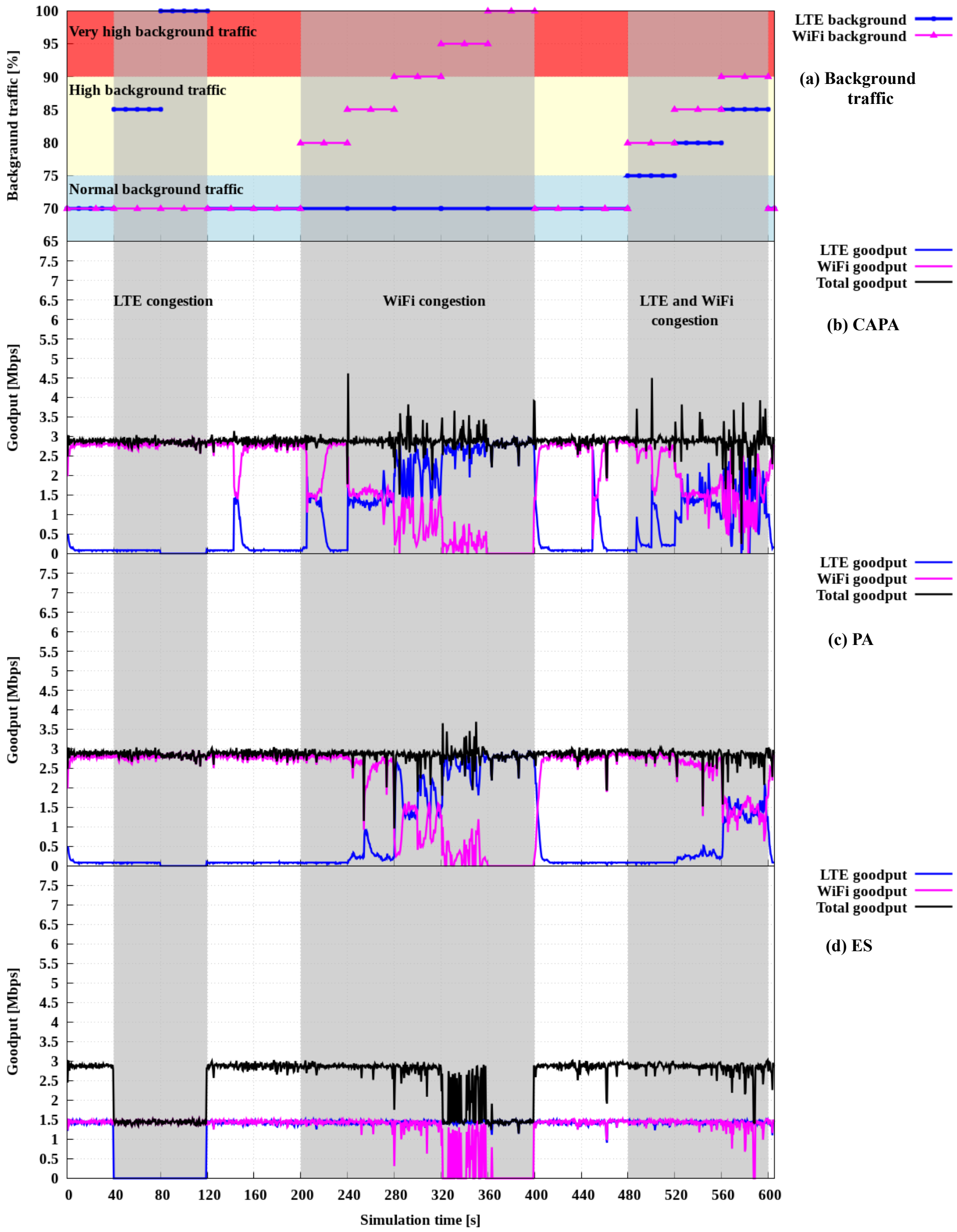


Figure 80 – LTE, WiFi and total (joint) goodput for *Meridian* packets with the constant transmission bit rate of 3 Mbps, for (a) the defined background traffic, and according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

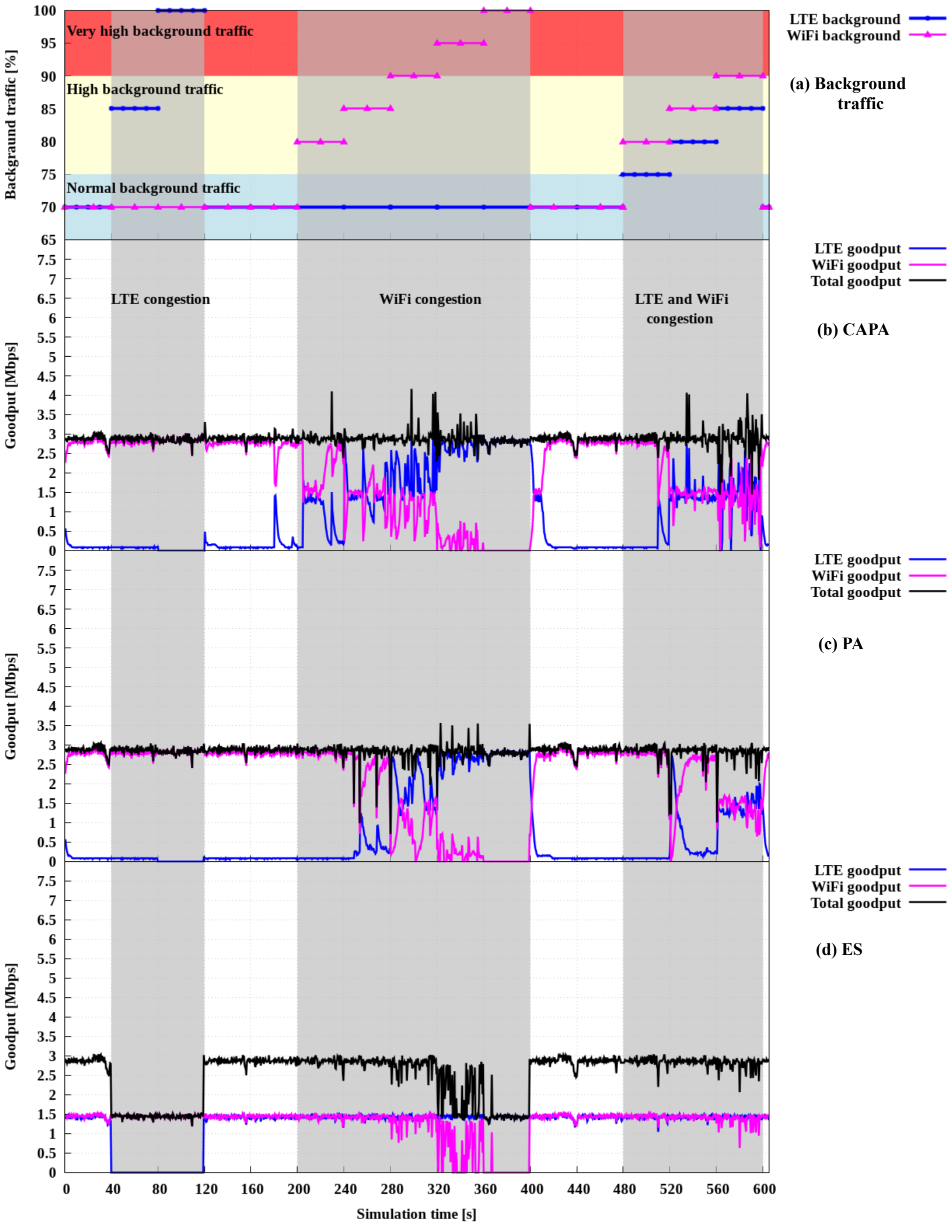


Figure 81 – LTE, WiFi and total (joint) goodput for *Big Buck Bunny* packets with the constant transmission bit rate of 3 Mbps, for (a) the defined background traffic, and according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

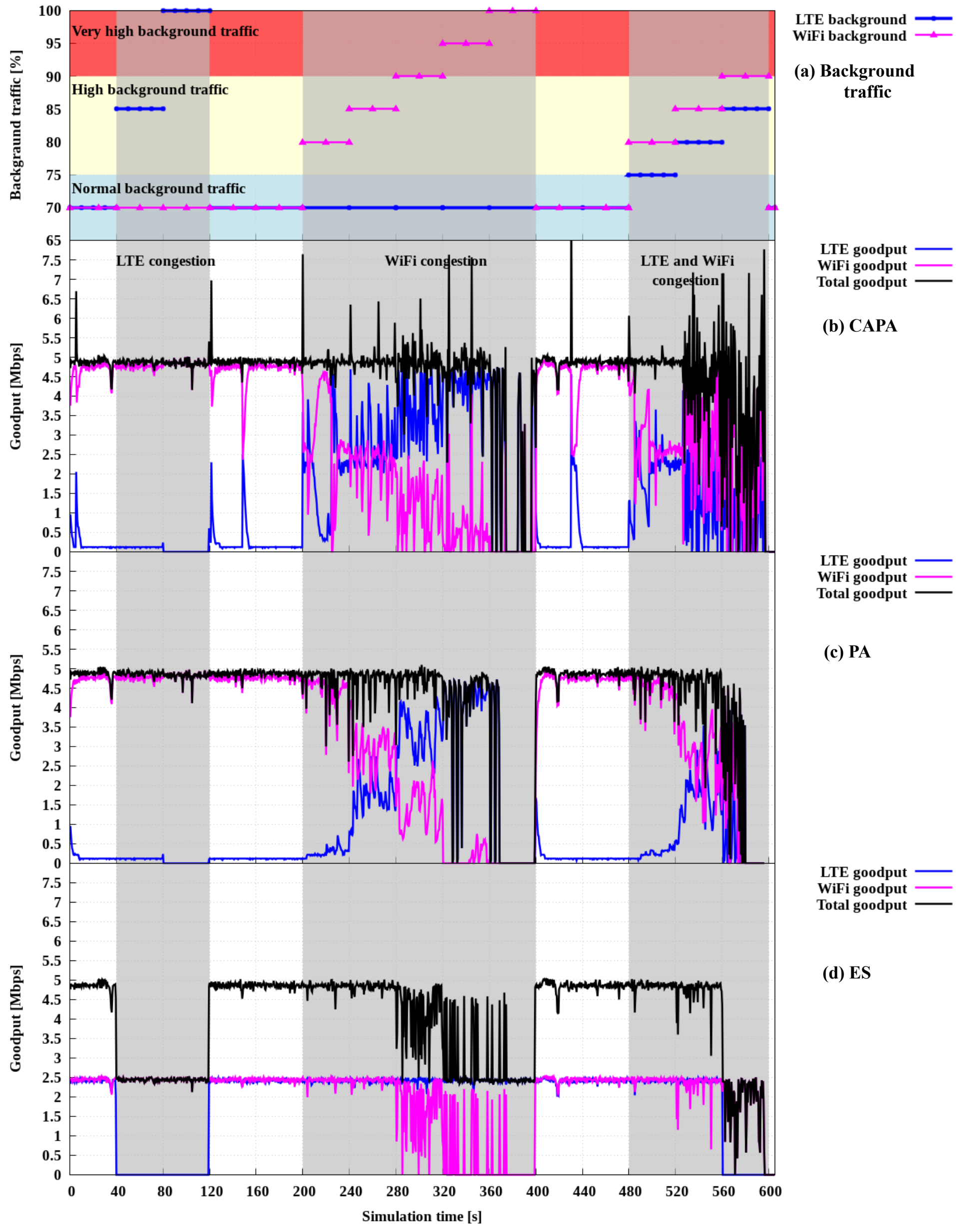


Figure 82 – LTE, WiFi and total (joint) goodput for *Big Buck Bunny* packets with the constant transmission bit rate of 5 Mbps, for (a) the defined background traffic, and according to the different scheduling strategies; (b) the proposed CAPA (c) PA, and (d) ES, under congested network scenario.

ANNEX F – PSNR Values for Different Video Bit Rates under Congested Network Scenario

Here, Figures 83, 84, and 85 show the PSNR values for all video frames according to the different scheduling strategies, CAPA, PA, and ES, respectively for *Meridian* with bit rate of 3 Mbps and *Big Buck Bunny* with bit rates of 3 and 5 Mbps .

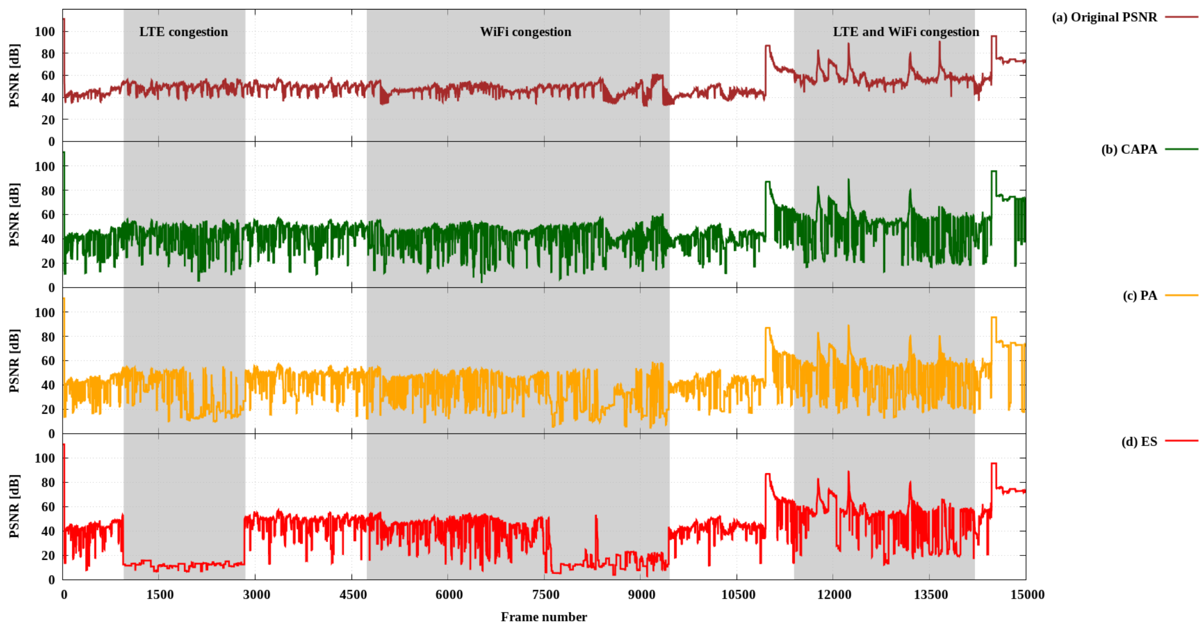


Figure 83 – PSNR values from *Meridian* sequence with bit rate of 3 Mbps according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

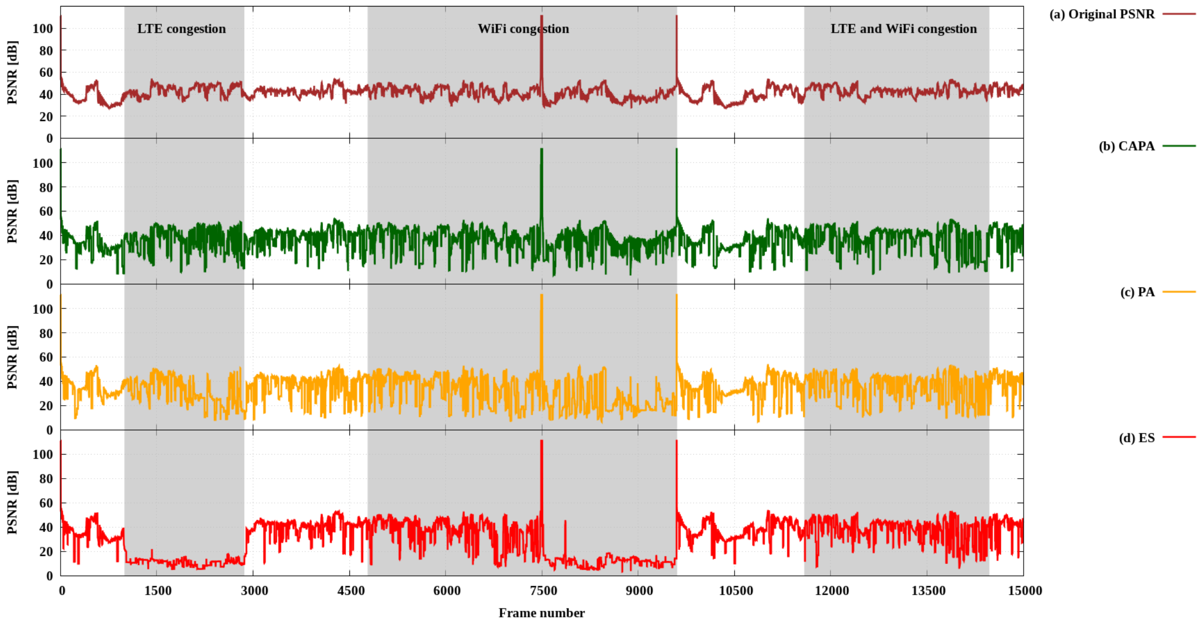


Figure 84 – PSNR values from *Big Buck Bunny* sequence with bit rate of 3 Mbps according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

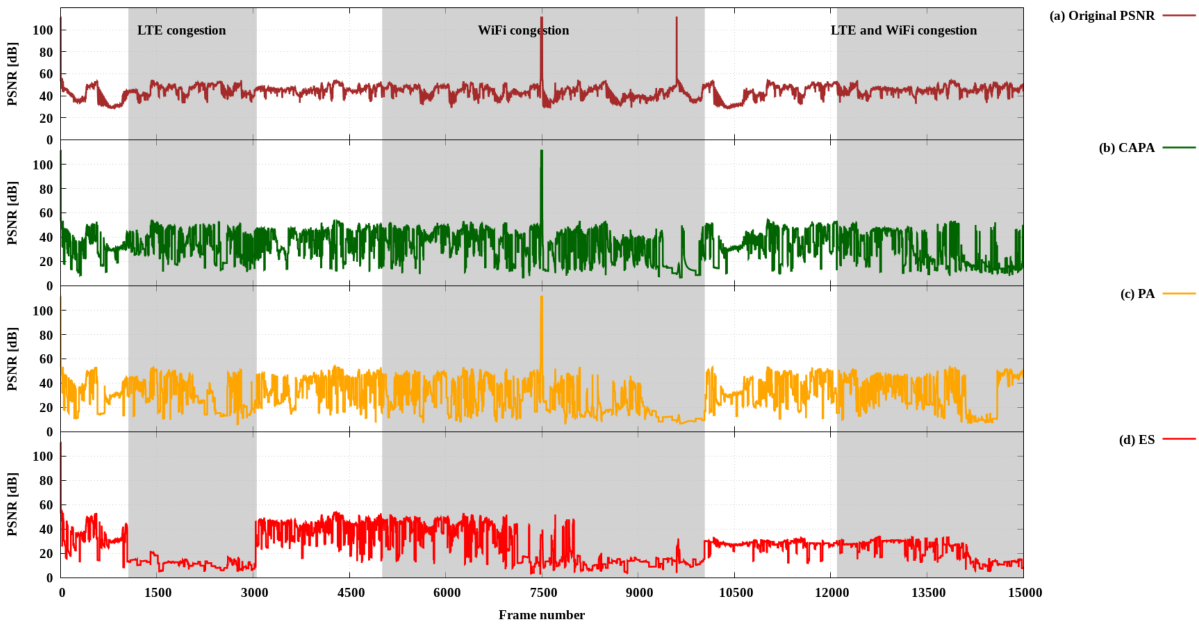


Figure 85 – PSNR values from *Big Buck Bunny* sequence with bit rate of 5 Mbps according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under congested network scenario.

ANNEX G – Goodput Performance for Constant Transmission Bit Rate under Wireless Lossy Network Scenario

Here, Figures 86, 87, 88, 89 show the LTE, WiFi and total (joint) goodput measured according to the different scheduling strategies, CAPA, PA, and ES, respectively for *Meridian*, *Sintel*, *LIVE*, and *Big Buck Bunny* under Lossy Network Scenario.

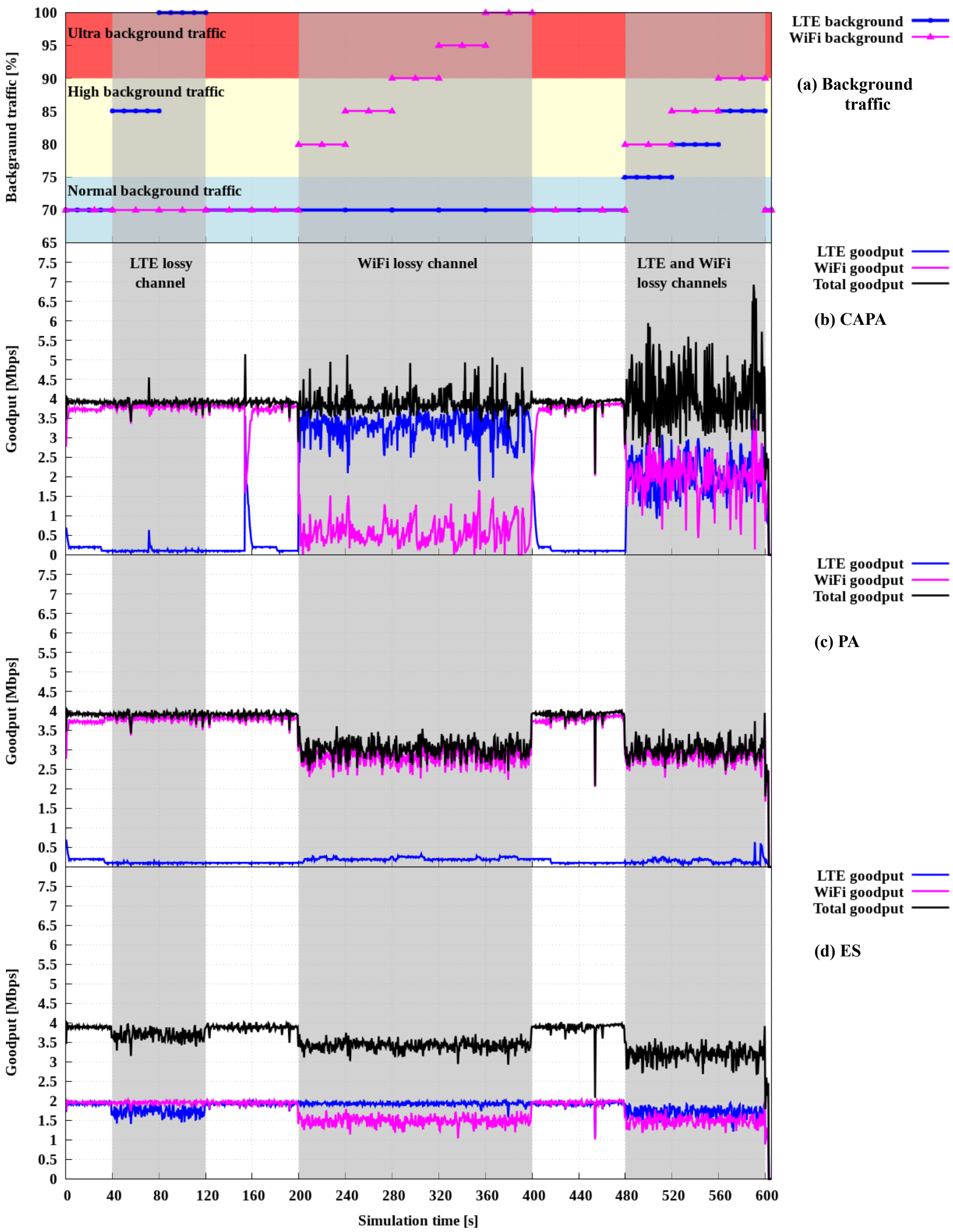


Figure 86 – LTE, WiFi and total (joint) goodput for *Meridian* packets for (a) the constant background traffic with burst wireless loss condition according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under wireless lossy network scenario.

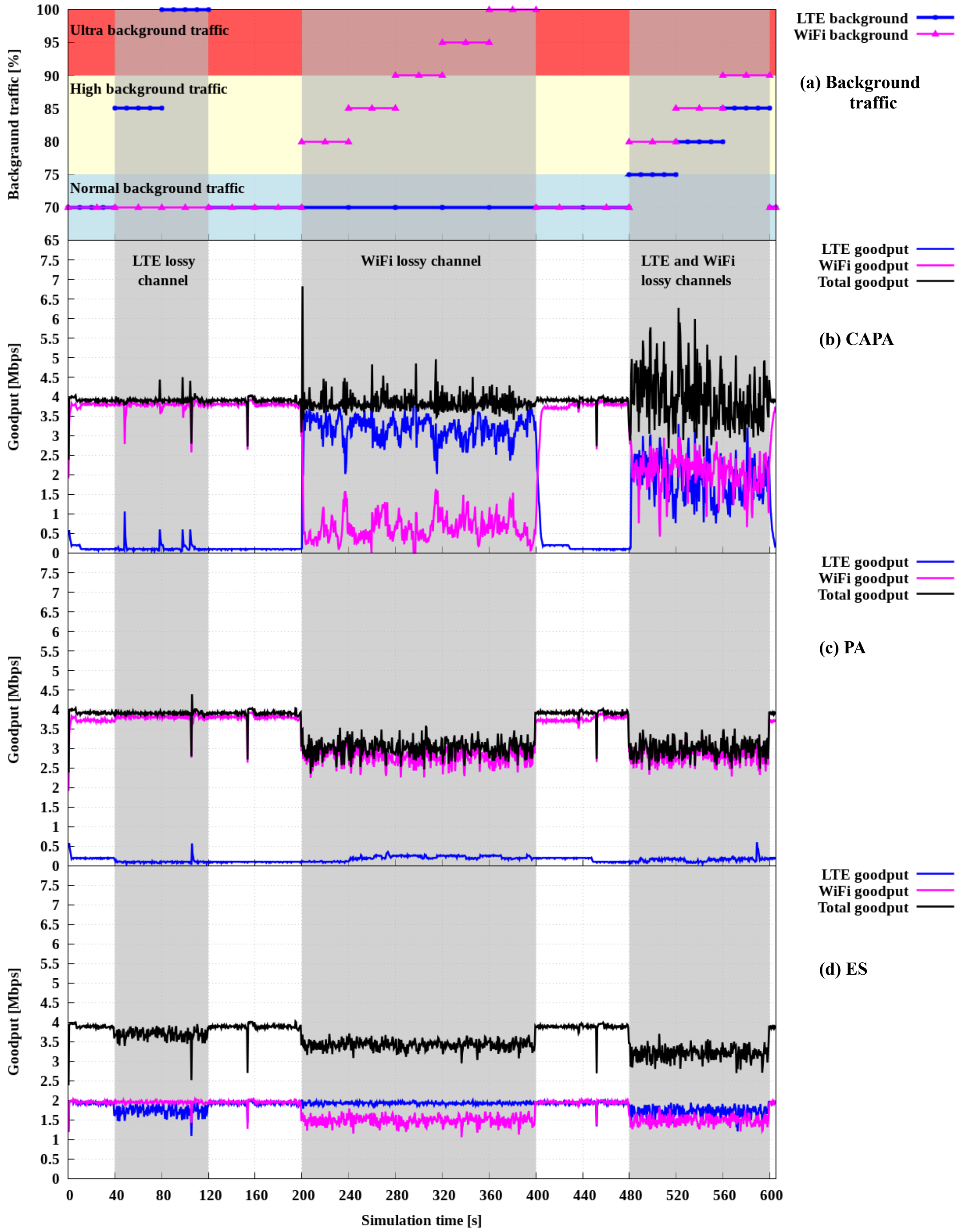


Figure 87 – LTE, WiFi and total (joint) goodput for *Sintel* packets for (a) the constant background traffic with burst wireless loss condition according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under wireless lossy network scenario.

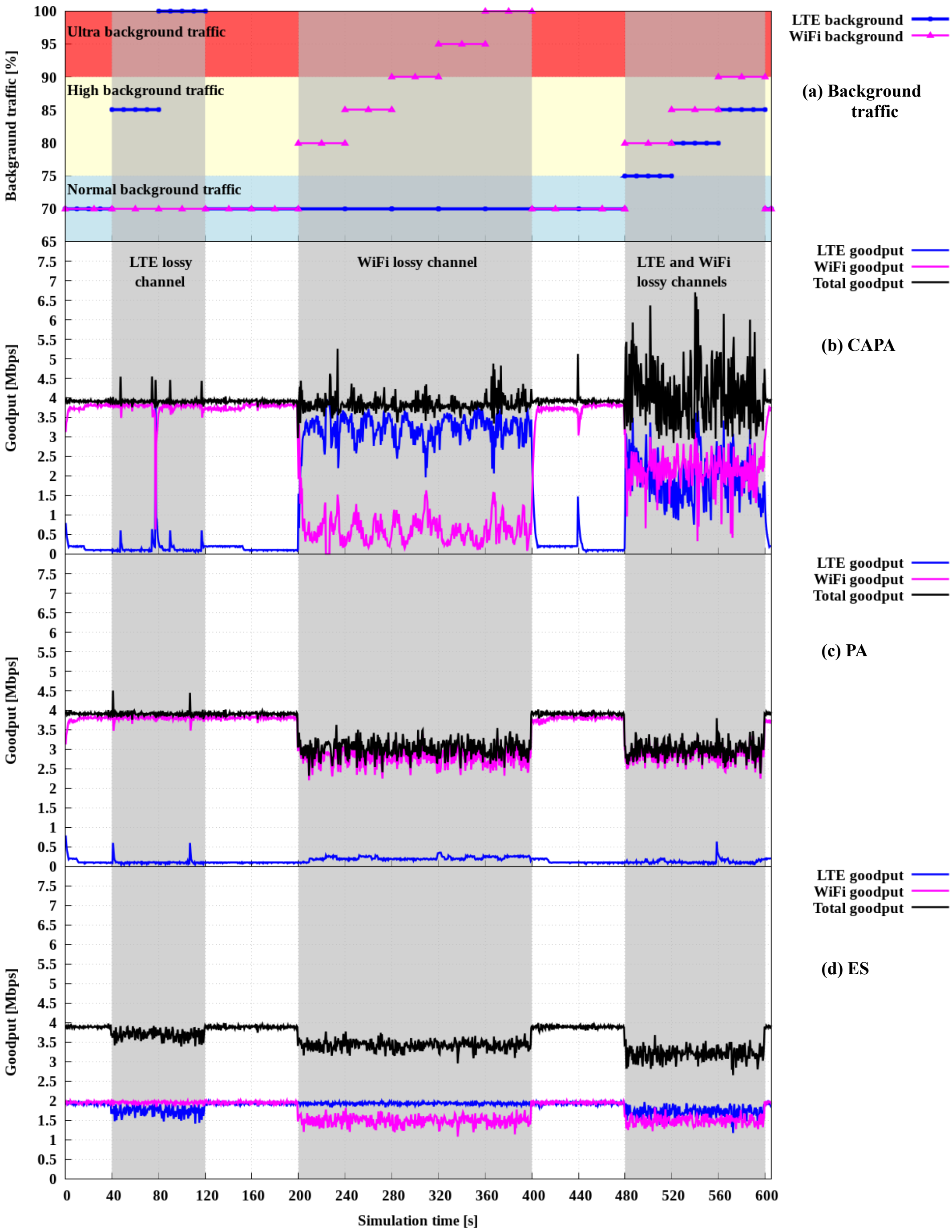


Figure 88 – LTE, WiFi and total (joint) goodput for *LIVE* packets for (a) the constant background traffic with burst wireless loss condition according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under wireless lossy network scenario.

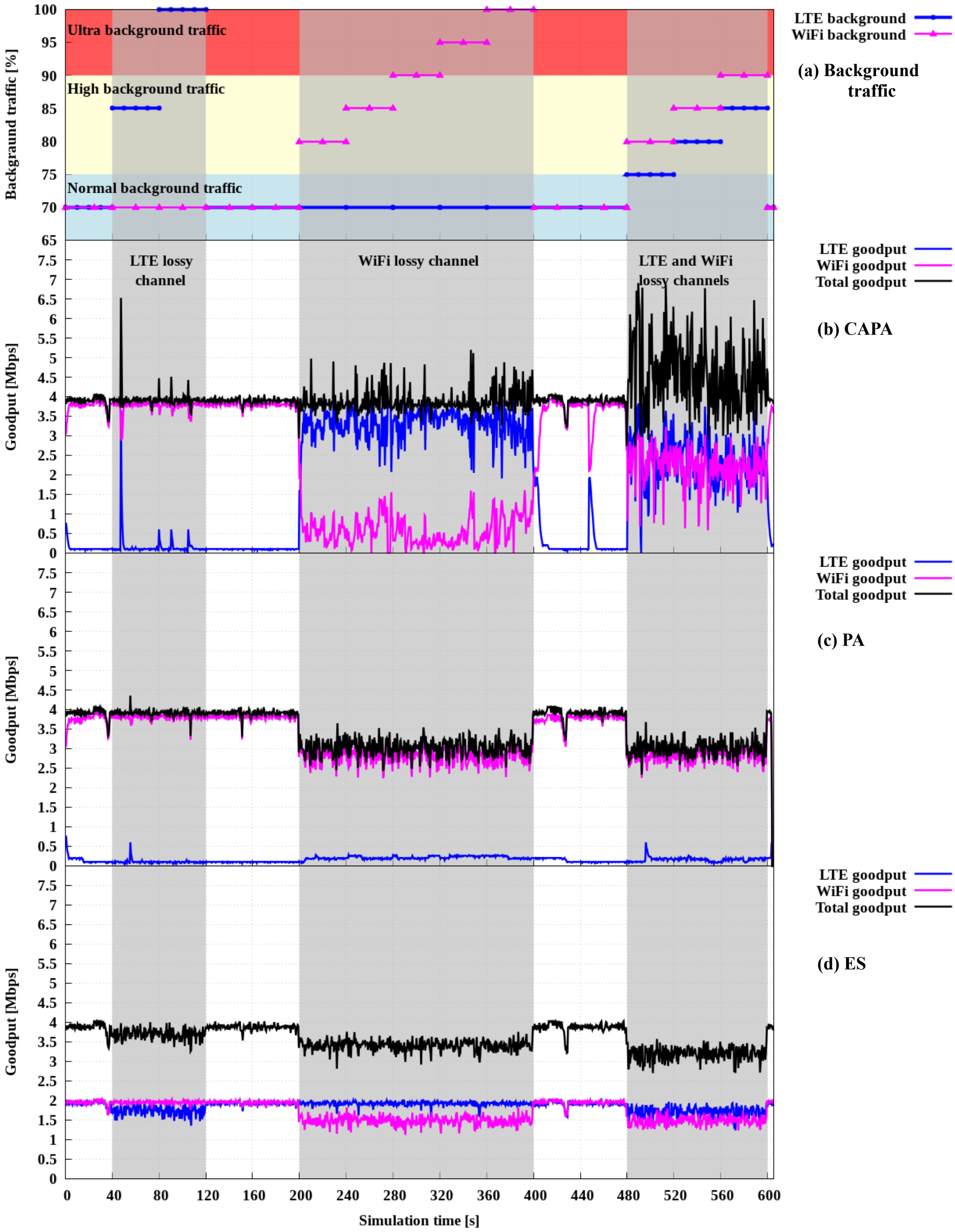


Figure 89 – LTE, WiFi and total (joint) goodput for *Big Buck Bunny* packets for (a) the constant background traffic with burst wireless loss condition according to the different scheduling strategies; (b) the proposed CAPA, (c) PA, and (d) ES, under wireless lossy network scenario.

ANNEX H – PSNR Values for Constant Transmission Bit Rate under Wireless Lossy Network Scenario

Here, Figures 90, 91, 92, 93 show the PSNR values for all video frames according to the different scheduling strategies, CAPA, PA, and ES, respectively for *Meridian*, *Sintel*, *LIVE*, and *Big Buck Bunny* under Wireless Lossy Network Scenario.

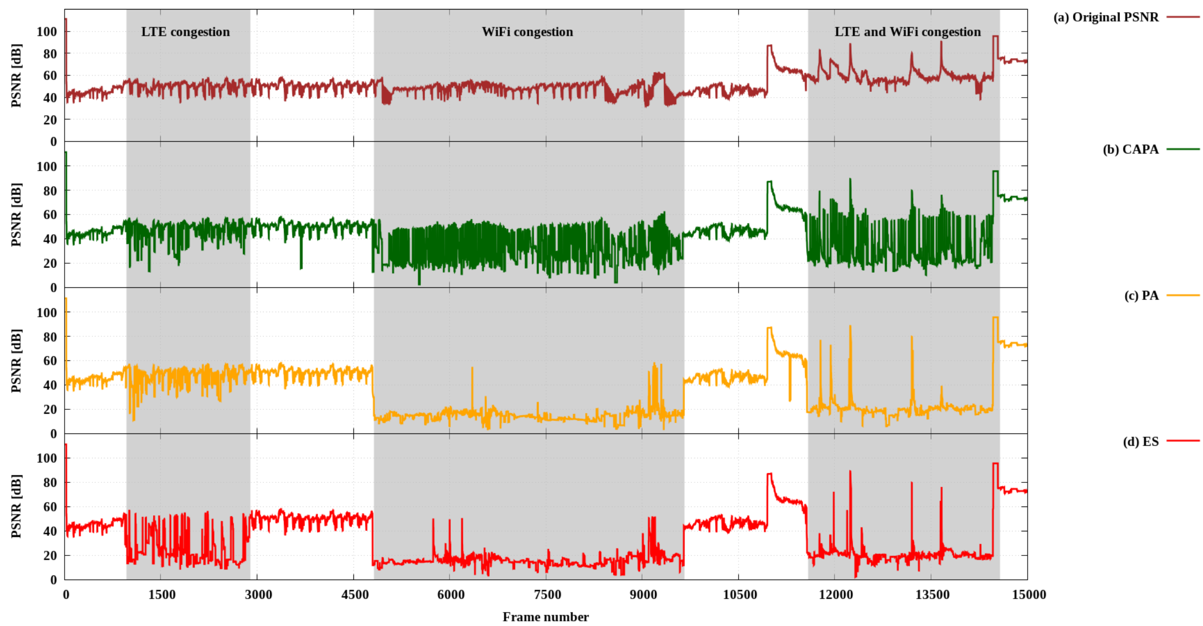


Figure 90 – PSNR values from *Meridian* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under wireless lossy network scenario.

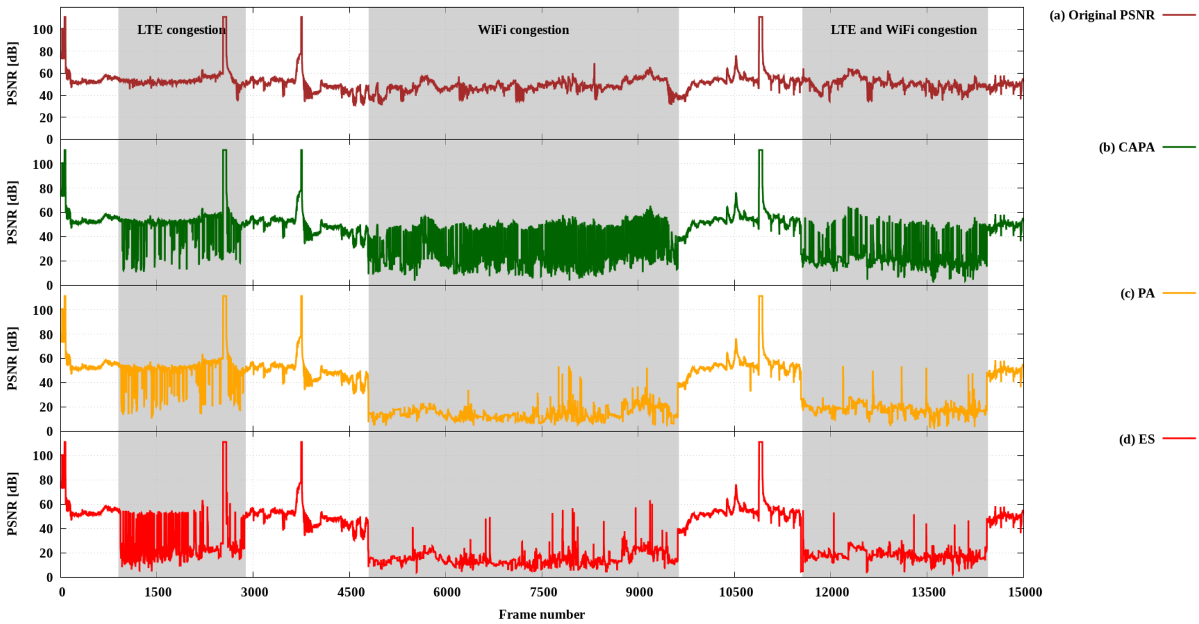


Figure 91 – PSNR values from *Sintel* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under wireless lossy network scenario.

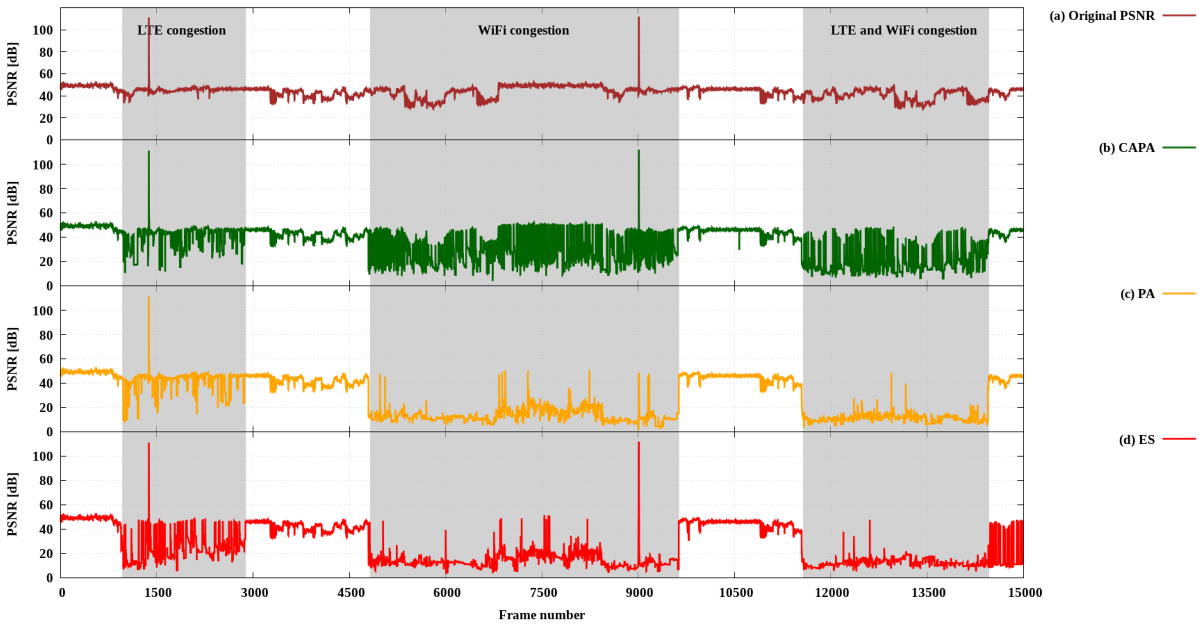


Figure 92 – PSNR values from *LIVE* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under wireless lossy network scenario.

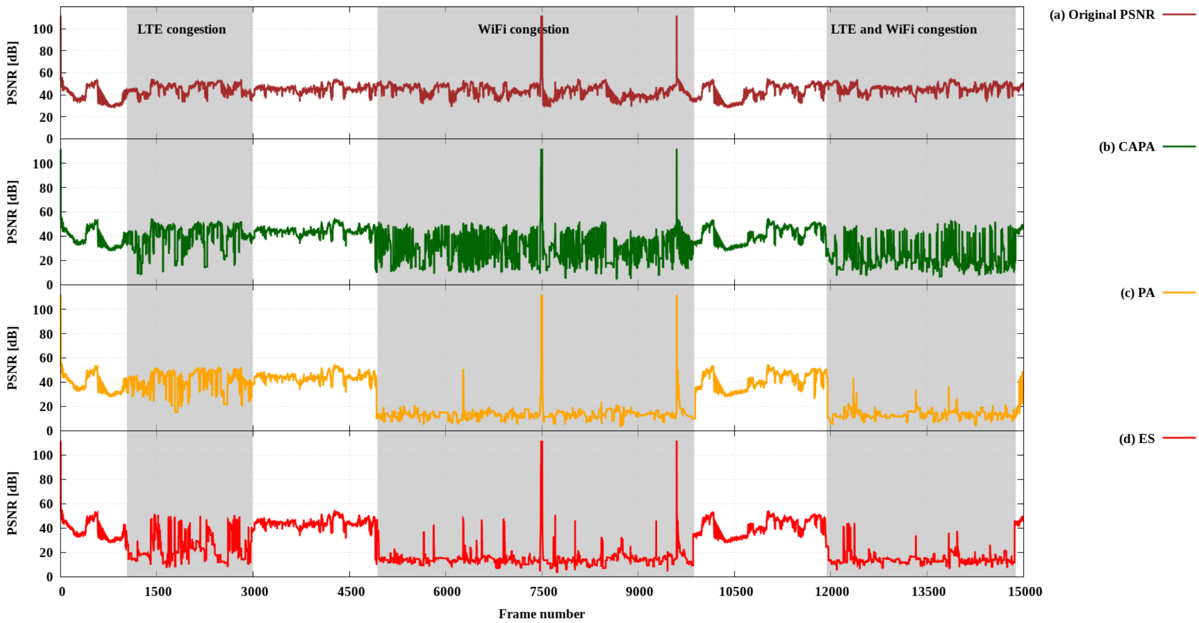


Figure 93 – PSNR values from *Big Buck Bunny* video sequence according to the different scheduling strategies compared to (a) original PSNR; (b) CAPA (c) PA, and (d) ES, under wireless lossy network scenario.