
UNIVERSITÀ DI PADOVA FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

SCUOLA DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE

INDIRIZZO IN SCIENZA E TECNOLOGIA DELL'INFORMAZIONE

XXVI Ciclo

# Video transport optimization techniques design and evaluation for next generation cellular networks

Ph.D. candidate

DANIELE MUNARETTO

**Supervisor:**

Chiar.mo Prof. Michele Zorzi

**Ph.D. School Director:**

Chiar.mo Prof. Matteo Bertocco

**Course Director:**

Chiar.mo Prof. Carlo Ferrari

Academic Year 2013/2014

*I want to dedicate my achievements to my beloved one and to my family, who believed in me and in my capabilities. A special thought goes to my grandfather, who passed away after my bachelor degree. He hoped to see me graduated as his last wish and strongly pushed me towards the hard study and to be hungry of knowledge. My career could not happen without the support of my mother, my grandmother and my sister, who always made sure that I could continue my studies at my fullest and with all the support that one would wish to get. A special dedication goes to Sarah, who supported me in the bad and good moments of my private and work lives. She always believes in me and takes care of my person, trying to make me feel beloved by and being part of this adverse reality.*

# Contents

# List of Figures

# List of Tables

# Abstract

Video is foreseen to be the dominant type of data traffic in the Internet. This vision is supported by a number of studies which forecast that video traffic will drastically increase in the following years, surpassing Peer-to-Peer traffic in volume already in the current year. Current infrastructures are not prepared to deal with this traffic increase. The current Internet, and in particular the mobile Internet, was not designed with video requirements in mind and, as a consequence, its architecture is very inefficient for handling this volume of video traffic. When a large part of traffic is associated to multimedia entertainment, most of the mobile infrastructure is used in a very inefficient way to provide such a simple service, thereby saturating the whole cellular network, and leading to perceived quality levels that are not adequate to support widespread end user acceptance. The main goal of the research activity in this thesis is to evolve the mobile Internet architecture for efficient video traffic support. As video is expected to represent the majority of the traffic, the future architecture should efficiently support the requirements of this data type, and specific enhancements for video should be introduced at all layers of the protocol stack where needed. These enhancements need to cater for improved quality of experience, improved reliability in a mobile world (anywhere, anytime), lower exploitation cost, and increased flexibility. In this thesis a set of video delivery mechanisms are designed to optimize the video transmission at different layers of the protocol stack and at different levels of the cellular network. Upon the architectural choices, resource allocation schemes are implemented to support a range of video applications, which cover video broadcast/multicast streaming, video on demand, real-time streaming, video progressive download and video upstreaming. By means of simulation, the benefits of the designed mechanisms in terms of perceived video quality and network resource saving are shown and compared to existing solutions. Furthermore, selected modules are implemented in a real testbed and some experimental results are pro-

vided to support the development of such transport mechanisms in practice.

# Sommario

Il traffico video sarà il tipo di applicazione dominante in Internet nei prossimi anni. Già in questi anni assistiamo al sorpasso del traffico video mobile rispetto al Peer-to-Peer. Le infrastrutture attuali non sono preparate ad affrontare questo aumento di traffico video. Internet, e in particolare Internet mobile, non è stata progettata sulla base di requisiti video e, di conseguenza, la sua architettura è inefficiente nel gestire questo tipo di traffico. Quando il traffico è associato all'intrattenimento multimediale, la maggior parte dell'infrastruttura mobile è utilizzata in un modo inefficiente pur fornendo un servizio semplice, saturando in tal modo l'intera rete cellulare e portando il servizio a livelli di qualità non adeguati a sostenere quella che gli utenti si aspettano di ricevere. L'obiettivo principale dell'attività di ricerca in questa tesi è quello di evolvere l'architettura di Internet mobile per un efficiente supporto del traffico video. Poiché il video è previsto rappresentare la maggior parte del traffico, l'architettura di rete deve supportare in modo efficiente le esigenze di questo tipo di traffico e miglioramenti specifici dovrebbero essere introdotti a tutti i livelli dello stack protocollare. Questi miglioramenti hanno lo scopo di incrementare la qualità percepita del servizio, di dare una maggiore affidabilità in un mondo mobile, di abbassare i costi di servizio e di aumentare la flessibilità della rete. In questa tesi una serie di meccanismi di trasmissione video sono progettati per ottimizzare la consegna di applicazioni video su reti cellulari di nuova generazione a diversi livelli dello stack protocollare ed a differenti livelli della rete cellulare. Sulla base di queste scelte architetturali, sistemi di allocazione delle risorse sono implementati per supportare una gamma di applicazioni video che copre il video broadcast/multicast in streaming, video on demand, streaming in tempo reale, il video download progressivo e il video upstreaming. Tramite campagne di simulazioni, i benefici sotto forma di qualità percepita e di risorse di rete risparmiate sono riportati attraverso il confronto con soluzioni pre-esistenti. Inoltre moduli selezionati sono implemen-

tati in un vero e proprio banco di prova e alcuni dei risultati sperimentali conseguiti sono usati per sostenere lo sviluppo di nuovi meccanismi di trasporto video nelle reti mobili future.

# List of Acronyms

| | |
|---|---|
| **AL-FEC** | Application Layer-Forward Error Correction |
| **AN** | Access Network |
| **APM** | Application Performance Metric |
| **AVC** | Advance Video Coding (H.264) |
| **BER** | Bit Error Rate |
| **BS** | Base Station (Node B) |
| **CAGR** | Compound Annual Growth Rate |
| **CDN** | Content Delivery Network |
| **CIF** | Common Intermediate Format (Video resolution) |
| **CN** | Core Network |
| **DASH** | Dynamic Adaptive Streaming over HTTP |
| **DL** | Downlink |
| **DM** | Decision Manager |
| **DMM** | Distributed Mobility Management |
| **E2E** | End-to-end |
| **E-UTRAN** | Evolved UTRAN |
| **eNodeB** | Evolved Node B |

| | |
|---|---|
| **EPC** | Evolved Packet Core |
| **EPS** | Evolved Packet System |
| **FEC** | Forward Error Correction |
| **GoP** | Group of Pictures |
| **GSM** | Global System for Mobile communications |
| **HD** | High Definition |
| **HSS** | Home Subscriber Server |
| **HSPA** | High Speed Packet Access |
| **HTTP** | HyperText Transfer Protocol |
| **IETF** | Internet Engineering Task Force |
| **IMS** | IP Multimedia Subsystem |
| **IP** | Internet Protocol |
| **JSVM** | Joint Scalable Video Model |
| **LTE** | Long Term Evolution (3.9 G) |
| **LTE-A** | Long Term Evolution-Advanced (4 G) |
| **MAC** | Medium Access Control |
| **MAR** | Mobile Access Router |
| **MCDN** | Mobile Content Delivery Network |
| **MCS** | Modulation Coding Scheme |
| **MOS** | Mean Opinion Score |
| **MN** | Mobile Node |
| **MPD** | Media Presentation Description |

| | |
|---|---|
| **MSE** | Mean-Square Error |
| **P2P** | Peer-to-Peer |
| **P-GW** | Packet Data Network Gateway |
| **PCRF** | Policy and Charging Rules Function |
| **PDT** | Packet Dropping with Threshold (mechanism) |
| **PHY** | Physical |
| **PS** | Path Selection (Algorithm) |
| **PSNR** | Peak Signal-to-Noise Ratio |
| **QoE** | Quality of Experience |
| **QoS** | Quality of Service |
| **RAN** | Radio Access Network |
| **RB** | Resource Block |
| **RM** | Resource Management (Algorithm) |
| **RSF** | Rate Scaling Factor |
| **S-GW** | Serving Gateway |
| **SDN** | Software-Defined Networking |
| **SNR** | Signal-to-Noise Ratio |
| **SSIM** | Structural Similarity (index) |
| **SVC** | Scalable Video Codec (H.264) |
| **TCP** | Transmission Control Protocol |
| **UE** | User Equipment |
| **UEP** | Unequal Error Protection |

| | |
|---|---|
| **UL** | Uplink |
| **UMTS** | Universal Mobile Telecommunications System |
| **UTRAN** | Universal Terrestrial Radio Access Network |
| **VAC** | Video Access Control (Algorithm) |
| **VoIP** | Voice over IP |
| **VSSIM** | Video Structural Similarity (index) |
| **VQM** | Video Quality Metric |
| **WiFi** | Wireless Fidelity |
| **WLAN** | Wireless Local Area Network |
| **WSN** | Wireless Sensor Network |

Chapter 1

# Introduction

The goal of this thesis is to enhance the cellular network architecture for efficient video traffic support. The need to cater for improved Quality of Experience (QoE) (see Appendix A) at the user side goes along with improved reliability in a mobile world (anywhere, anytime), lower exploitation costs for mobile operators, and increased flexibility of the network.

A set of delivery frameworks are designed to optimize the transmission of video applications over next generation cellular networks. We first present the topics investigated in this work per architectural area of the cellular network, followed by a detailed discussion about the different layers of the protocol stack that are taken into account in the study. Upon these architectural choices, resource allocation schemes are implemented to support a range of video applications, which include video broadcast/multicast streaming, video on demand, real-time streaming, video progressive download and video up-streaming.

The work presented in this thesis appeared in some articles reported in the list of publications at the end of the document and was partially supported by the EU MEDIEVAL (MultimEDia transport for mobIlE Video AppLications) research project [1], a medium-scale focused research project of the 7th Framework Programme of the European Commission, addressing the core of the strategic objective "The Network of the Future".

Starting from the access network area, cross-layer video packet scheduling mechanisms are proposed in [2–6] to best make use of the knowledge about the application layer, i.e., how video packets are encoded with the H.264-Scalable Video Coding (SVC) encoder [7] (see Appendix B) and how they can be protected from erasures by means of unequal error protection (UEP), i.e., Forward Error Correction (FEC) at the Application Layer (AL-FEC).

Jointly with the application layer information the scheme takes into account the physical layer information, i.e., how the modulation and coding schemes (MCS) are opportunistically selected to best transmit the video into the wireless medium to achieve a target level of perceived quality, while making a smart use of the available network resources. In this thesis we limit the focus of this research avenue to a robust opportunistic scheduling for broadcast video patented in [6] and published in [4, 5]. The architectural design choices are presented in Sec. 3.1, the implementation details are given in Sec. 4.1 and the simulation results are reported in Sec. 5.1.

Moving to the core network side, here the concept of making the Content Delivery Networks (CDN) mobile, i.e., adapted to the cellular network and thus to mobile users, is integrated with the need for novel mobility protocols which make it possible to keep the video session continuity when changing point of access, i.e., the so called Distributed Mobility Management (DMM) [8–10]. Moreover, the concept of video popularity is as well taken into account to design the policy for best caching the video contents in the network in terms of network latency and network operators costs. The architectural design choices for this topic are presented in Sec. 3.2, the implementation details are given in Sec. 4.2 and the simulation results are reported in Sec. 5.2.

Furthermore, from an end-to-end point of view, a set of network metrics are considered to identify the available video paths in the network in order to find the video path which delivers the best QoE to the user [11–15]. This topic crosses the previous activities and provides insights for network operators into the use of Software-Defined Networking (SDN) [16] tools to manage the network in a novel and smart way, e.g., as proposed by Open Flow [17]. The architectural design choices for this topic are presented in Sec. 3.3, the implementation details are given in Sec. 4.3 and the simulation results are reported in Sec. 5.3.

The concept of QoE can be used to differentiate videos based on their sensitivity to the rate changes, i.e., some videos, when the rate is reduced by half the original rate, will have a negligible impact on the QoE of the user, opposite to other videos which will be showing severe impairments. Thus, video call admission control schemes are designed in [18], while in [19] we make use of the concept of popularity of the videos to differentiate the resource allocation among videos. Our solution is provided in Sec. 4.4 while the simulation results

are given in Sec. 5.4. No original architectural contributions are provided in this case since our solution fits the existing cellular network architectures and scales well with the network size.

Last but not least, we mention coding schemes for underwater communications and wireless sensor networks designed to make such communications feasible in practice. These works are reported in Appendixes C, D since they are marginal with respect to the main scope of this thesis. We refer the interested reader to [20, 21].

The thesis is structured is as follows. In Chapter 2 we give an overview of the next generation cellular architecture and the motivation for our work, in Chapter 3 we present our architectural design choices. In Chapter 4 we describe how the envisioned solutions are implemented, and the simulation and experimental results are presented in Chapter 5. We conclude the thesis in Chapter 6.

Chapter 2

# Background

## 2.1 Next Generation Mobile Networks

In this chapter we introduce our reference scenario, i.e., the next generation cellular networks. In these years the market demand has grown considerably for mobile broadband with consumers and businesses enjoying the full benefits of email and web browsing on the move. However, limitations in speed and high latency (compared to fixed line broadband) has made 3G services unsuitable for high quality demanding or time-sensitive applications such as Voice over IP (VoIP), video streaming and on-line game play.

The Third Generation Partnership Program (3GPP) [22] has defined Long Term Evolution (LTE) [23] as part of the 3GPP Release 8 specifications. LTE may also be referred more formally as Evolved Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access (E-UTRA) and Evolved UMTS Terrestrial Radio Access Network (E-UTRAN). Even though 3GPP created standards for the Global System for Mobile communications (GSM) and UMTS families, the LTE standards are completely new. In the following list the main objectives of the LTE standardization group are listed [24]:

- Increased downlink and uplink peak data rates: 100 Mbps for downlink (DL) with 20 MHz bandwidth (2 receiving antenna at the User Equipment (UE)), 50 Mbps for uplink (UL) with 20 MHz;
- Scalable bandwidth;
- Improved spectral efficiency: 5 bps/Hz for DL and 2.5 bps/Hz for UL;

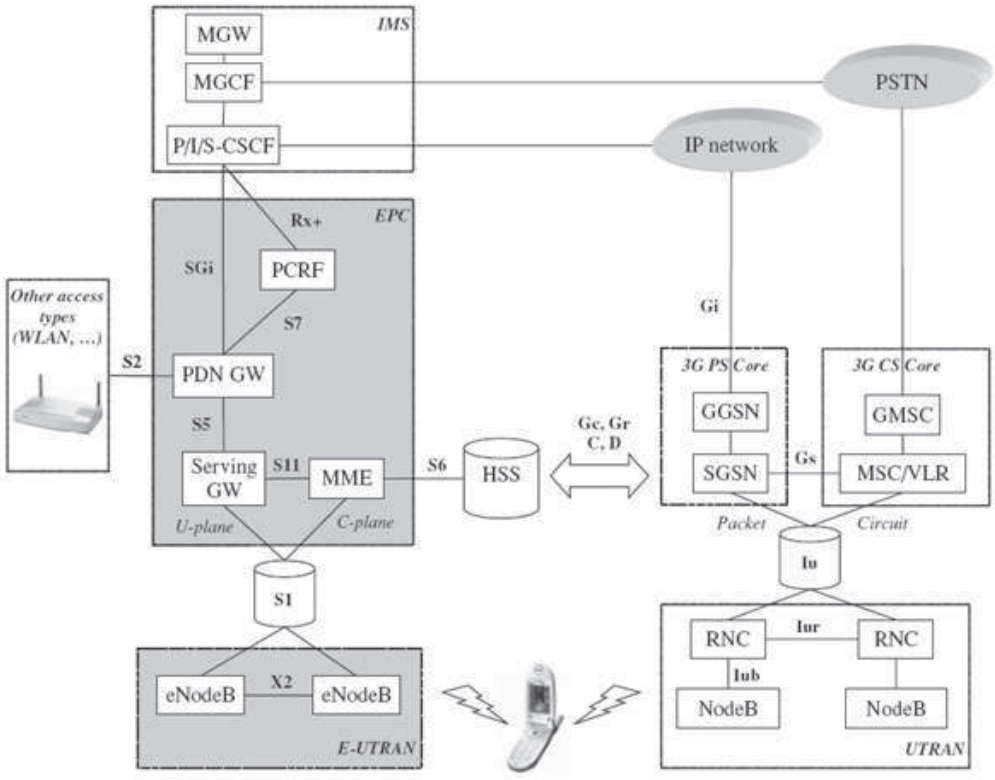



**Figure 2.1.** *LTE core and access networks architecture and 3G network compatibility.*

- All IP network;
- Standard's based interface that can support a multitude of user types;
- Reduced cost for the operator;
- Improved system capacity and coverage, reduced latency.

LTE networks are intended to bridge the functional data exchange gap between very high data rate fixed wireless Local Area Networks (LAN) and very high mobility cellular networks. LTE introduces the possibility of complementing High-Speed Packet Access (HSPA) networks with higher peak data rates, greater flexibility for heterogeneous networks and flatter network architecture.

In Fig. 2.1 we show the main entities and interfaces of the LTE networks and how they cooperate with 3G networks, i.e., LTE is backward compatible. The new blocks specific to Evolved UMTS evolution, also known as the Evolved Packet System (EPS), are the Evolved

Packet Core (EPC) and the E-UTRAN. Other blocks from the classical UMTS architecture are also displayed, such as the UTRAN, the Packet Switch and the Circuit Switch Core Networks, respectively 3G-PS and 3G-CS cores, connected to the public (or any private) IP and Telephone Networks. The IMS (IP Multimedia Subsystem) is located on top of the Packet Core blocks and provides access to both public or private IP networks, and the public telephone network via Media Gateway network entities. The Home Subscriber Server (HSS), managing user subscription information, is shown as a central node, providing services to all Core Network blocks of the 3G and evolved 3G architectures.

Compared with UTRAN, the E-UTRAN OFDM-based structure is quite simple. It is only composed of one network element: the eNodeB (evolved Node B). The 3G RNC (Radio Network Controller) inherited from the 2G BSC (Base Station Controller) has disappeared from E-UTRAN and the eNodeB is directly connected to the Core Network using the S1 interface. As a consequence, the features supported by the RNC have been distributed among the eNodeB, the Core Network Mobility Management Entity (MME), and the Serving Gateway (S-GW).

From a functional perspective, the eNodeB supports a set of legacy features, all related to physical layer procedures for transmission and reception over the radio interface, such as modulation and de-modulation and channel coding and de-coding.

Besides, the eNodeB includes additional features, coming from the fact that there are no more Base Station controllers in the E-UTRAN architecture. Those features include the following:

- Radio resource control: this relates to the allocation, modification and release of resources for the transmission over the radio interface between the user terminal and the eNodeB;

- Radio mobility management: this refers to the handover procedures;

- Radio interface full Layer 2 protocol: the layer 2 purpose is to ensure the transfer of data between network entities. This implies the detection and possibly correction of errors that may occur in the physical layer.

The EPC (Evolved Packet Core) is composed of several functional entities [25]:

- The MME, that is in charge of all the control plane functions related to subscriber and session management;

- The HSS, that is in charge of storing and updating when necessary the database containing all the user subscription information;

- The Serving Gateway, that serves as a local mobility anchor, meaning that packets are routed through this point for intra E-UTRAN mobility and mobility with other 3GPP technologies, such as 2G/GSM and 3G/UMTS;

- The PDN (Packet Data Network) Gateway (P-GW), that is the termination point of the packet data interface towards the Packet Data Network; it supports operator-defined rules for resource allocation and usage as well as packet filtering, like Deep Packet Inspection (DPI) for virus signature detection, and evolved charging support (like per URL charging);

- The PCRF (Policy and Charging Rules Function) Server, that manages the service policy and sends Quality of Service (QoS) setting information for each user session and accounting rule information.

The trend of the fourth generation (4G, LTE-Advanced) mobile communications protocols theoretically leads to speeds of 100 Mbps with minimal latency, thus potentially becoming an access technology capable of handling all applications from basic email to bandwidth-demanding High Definition (HD) video. As affordable higher-speed 3G+ (HSPA) services have been the driver behind the increase in smartphones, 4G is expected to drive a new wave of devices and applications that can benefit from high-speed, high-quality mobile broadband. The easier expansion of services geographically is also a driver behind 4G. Wireless communications has long been envisaged as a solution for providing rural communities and developing nations with high-speed broadband service. With its accessibility via mobile phones and USB modems has the potential to become the preferred “last mile” link for high-speed broadband access in the very next future.

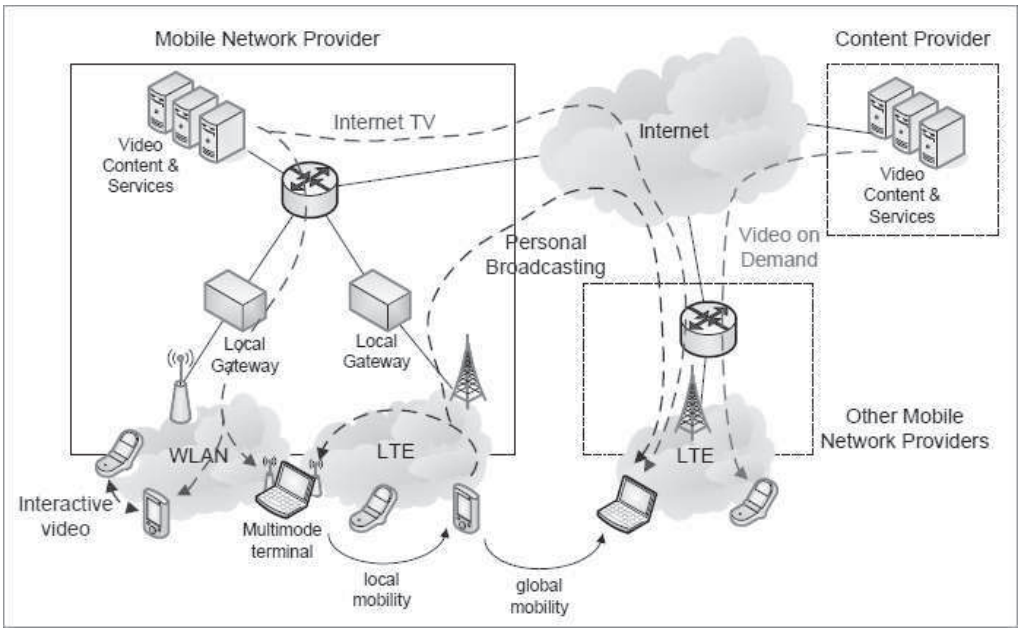



**Figure 2.2.** *Reference cellular architecture and use cases.*

## 2.2 Motivation

Nowadays sophisticated smartphones and increasingly optimized mobile networks provide a high-quality always-connected experience to mobile customers. Mobile devices can connect to the Internet via different broadband wireless access technologies as in Fig. 2.2, allowing the users to access multimedia contents from anywhere and at any time. This enables a set of real-time consuming video services such as personal broadcast, video on demand, video streaming and so on.

The deployment of multiple video sources in different regions, which is already a reality in fixed networks, is a hot topic for mobile service providers which have to deal as well with suboptimal video transport mechanisms on the way to the users. For instance, the integration of CDNs in the mobile service provider network gives the operator an additional degree of freedom when selecting, based on pre-defined procedures, the path to deliver the content from the video source (CDN cache) to the radio access.

The tremendous increase of mobile traffic generated by the wide range of multimedia applications for portable devices will force mobile network operators to face new transport optimization challenges. Video traffic is expected to be the major driver of such steep growth, as reported in the latest global mobile data traffic forecast released by [26], which

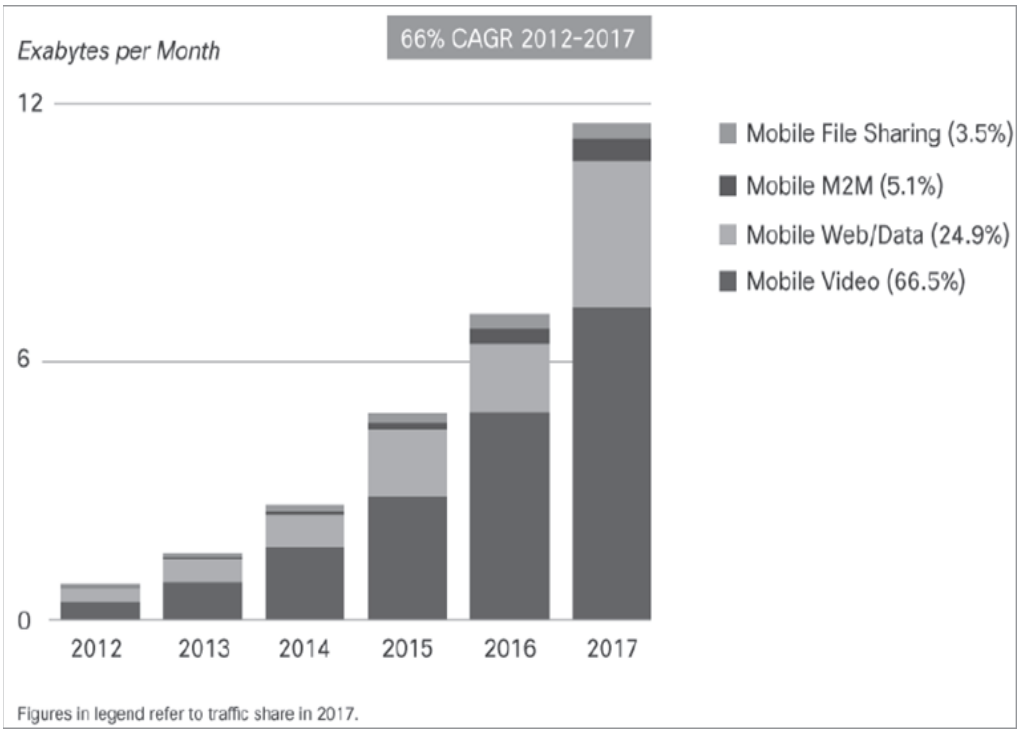



**Figure 2.3.** *Traffic increase in the period 2012-2017.*

predicts that video traffic will grow by 75% between 2012 and 2017, and will account for over 66% of total mobile data traffic by the end of the forecast period, as shown in Fig. 2.3.

The foreseen amount of traffic offloaded from smartphones, due to the overloaded networks, is estimated to be as high as 46% in 2017, Fig. 2.4 , and the amount of traffic offloaded from tablets will be 71% in 2017.

The same studies report that the average smartphone usage grew 81 percent in 2012. In particular, the average amount of traffic per smartphone in 2012 was 342 MB per month, up from 189 MB per month in 2011. The monthly global mobile data traffic will surpass 10 exabytes in 2017, whereas the number of mobile-connected devices exceeded the world's population in 2013. The average mobile connection speed will surpass 1 Mbps in 2014 and, due to the increased usage of smartphones, handsets exceeded 50 percent of mobile data traffic in 2013. The monthly mobile tablet traffic will surpass 1 exabyte per month in 2017 and tablets will exceed 10 percent of global mobile data traffic in 2015. For what concerns Italy, we report that the mobile traffic in 2012 was estimated to grow as high as 32% year-over-year.

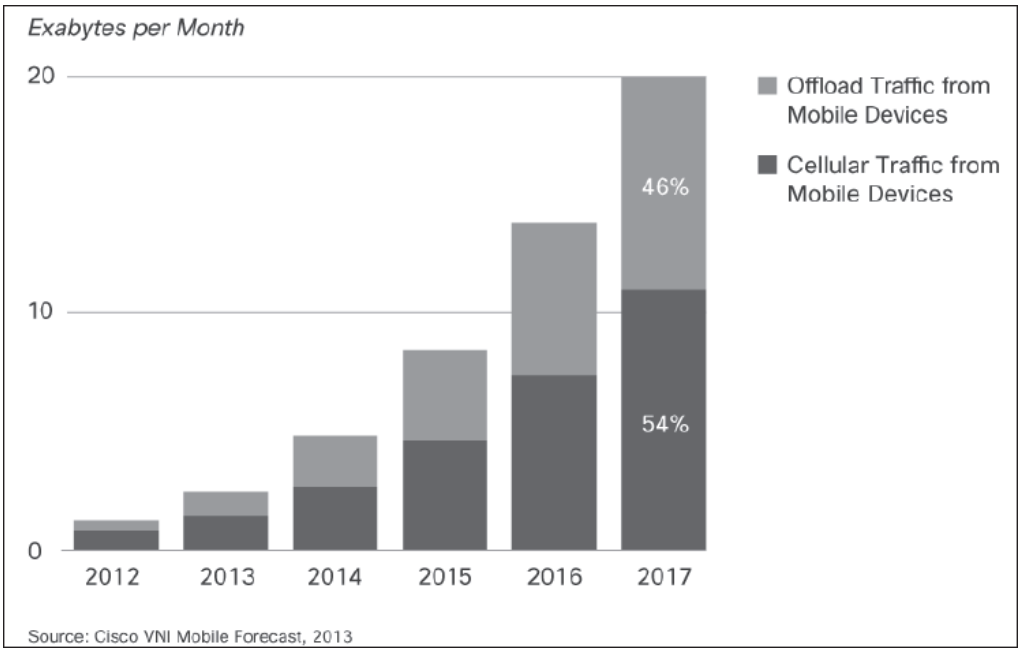



**Figure 2.4.** *Percentage of traffic offloaded in the period 2012-2017.*

Overall, the mobile data traffic volume is expected to grow to 11.2 exabytes per month by 2017, a 13-fold increase over 2012, which means that the mobile data traffic will grow at a Compound Annual Growth Rate (CAGR) of 66 percent from 2012 to 2017, Fig.2.5.

The observation that we draw from these studies is that mobile network operators need to be able to handle this rising data demand by providing high performance computing and enhanced content delivery mechanisms. However, the costs involved in a radical upgrade of the backhaul and core networks make such a brute-force solution very unattractive, and smart techniques of using the current infrastructure, e.g., by managing content differently, are actively being sought. This motivates us to design and implement video transport techniques to efficiently deliver the contents to the users in terms of perceived video quality on the user's side and smart use of the available network resources from the operator's point of view.

The investigation and design of our novel architectural solutions will follow in Chapter 3 and Chapter 4.

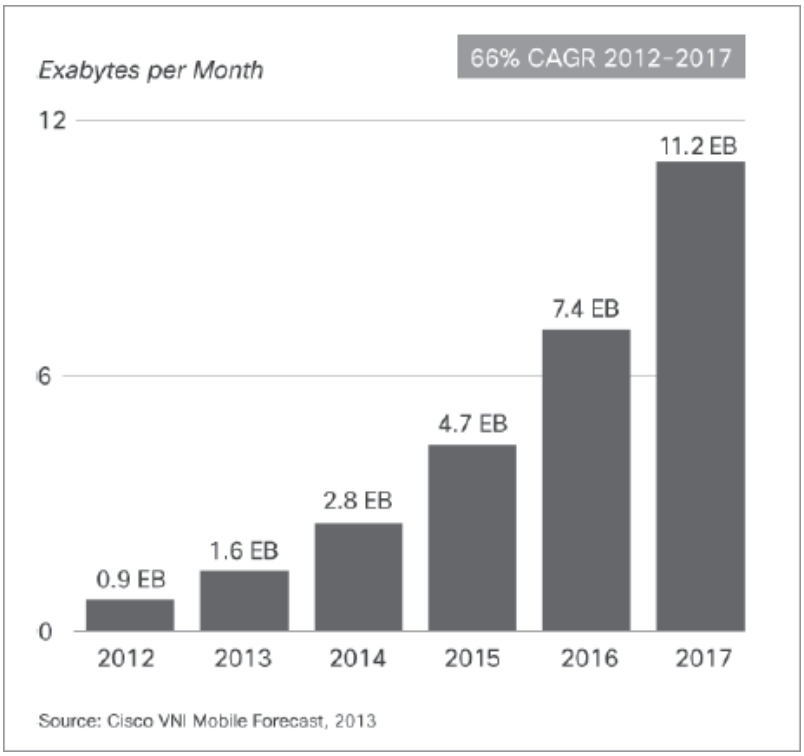



**Figure 2.5.** *Cisco forecasts 11.2 exabytes per month of mobile data traffic by 2017.*

Chapter 3

# Mobile architecture for video transport

In this chapter we introduce and design our proposed video transport architectures. Each of the following sections tackles a specific issue of the video delivery chain at different layers of the ISO/OSI protocol stack and at different levels of the LTE networks. Starting from the access side, in Section 3.1 a robust opportunistic scheduling to be implemented at the base station side is presented with the target of enhancing the video quality perceived at the user side while making efficient use of the available network resources. Moving to the core network side, in Section 3.2 a novel framework integrating the CDN concept with the DMM functionalities is proposed to address the issue of distributing the video contents at the edge of the core network to reduce the consumption of the core network resources and the delivery latencies to the user. Then, from an end-to-end (E2E) perspective, in Section 3.3 a video path selection mechanism is designed to jointly address the issue of allocating resources at the radio access side and the issue of selecting the sources and routing paths at the core network side.

## 3.1 Robust Opportunistic Broadcast/Multicast Mechanisms

We design a robust opportunistic broadcast scheduler which works based on the average and instantaneous user distributions, i.e., the probability distribution of the maximum channel rate at which users can demodulate video packets and with the path-loss model considered in this work, this corresponds to a spatial distribution of users in MCS serving areas. Based on the average and instantaneous sets of users that can demodulate pack-

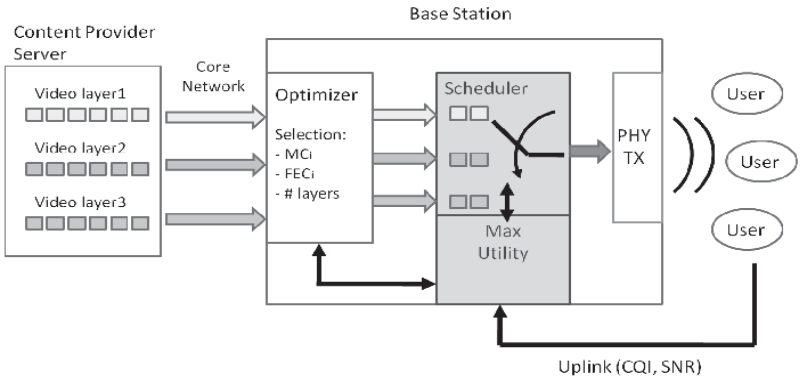



**Figure 3.1.** *Delivery architecture: from the video server to the users.*

ets transmitted with a certain MCS, and the users' channel quality, in each time slot the scheduler makes the decision on the MCS to be used for the current video stream to be transmitted. Thus, our cross-layer approach, which encompasses media characteristics and instantaneous channel conditions of the users, affects the size of the user groups accessing each video substream. Our goal is to minimize the wireless resource usage, i.e., the channel rate required for serving users, while keeping the target QoS in the cell.

From an architectural point of view, we consider a typical media delivery system as shown in Fig. 3.1, where a video server encodes a video into one or multiple video quality substreams for transmission through the network to one or multiple base stations. We consider the video encoder H.264/SVC [7], the scalable extension of H.264/AVC, due to its appealing features for broadcast streaming applications, where providing different video quality levels to a heterogenous set of users is beneficial. According to its position and channel quality, a given user will be able to decode a certain number of layers, starting from the one with the lowest quality. Users experiencing a better channel will have access to more layers, and will therefore enjoy a higher-quality video, whereas the worst-case users will only receive the basic layer. Without loss of generality, we now focus our analysis on a single BS.

We assume that the BS temporarily buffers packets from scalable video substreams in different queues before transmission on the wireless medium. Each substream is dedicated to a possibly different user group (i.e., fewer users will receive the higher-quality layers) and, depending on the aggregated channel conditions of the group, can be modulated with

a possibly different MCS, which is the one that the worst user in the group can still decode. In general, the user groups can be independent (e.g., in the case of different video streams, or different streams that corresponds to different versions of the same video with different quality levels), or inclusive (e.g., in the case of different scalable substreams of the same video, providing different quality levels to different users, according to how many layers (substreams) they can receive). For simplicity, we also assume that the aggregated rates of the scheduled video substreams, given their chosen MCS, do not exceed the channel capacity, e.g., computed per Group of Pictures (GoP). The video rates and MCSs associated to each queue are the result of solving an optimization problem, which is not directly addressed. For an example of such an approach, we refer the reader to the algorithm in [27]. The computation is performed periodically, based on average channel conditions of the target users.

The proposed opportunistic scheduler must transmit at each time instant the available packets in the transmission queues. We assume that the scheduler is informed about the instantaneous channel quality of each user through the feedback channel (via uplink), and is able to take this information into account in its scheduling decision. For each user group associated to one video layer buffered in one queue at the base station, it computes the instantaneous channel conditions. In each time slot, the scheduler opportunistically schedules for transmission the queue associated to the user group which currently experiences good channel conditions, i.e., selecting a higher MCS to serve it, while at the same time taking into account the relative importance of the content of the queued scalable video substreams [7]. In case the associated queue is empty, the scheduler picks the next best user group (with respect to the channel conditions and content). The procedure ensures the efficient use of wireless resources, allowing some channel capacity saving, while keeping the overall expected user perceived video quality.

Furthermore, the scheduler actions should not cause buffer overflow at the base station, nor play-out shortages at the mobile client side.

The implementation details follow in Sec. 4.1.

## 3.2 Mobile Content Delivery Networks and Video Popularity

We design a framework where a CDN system is integrated in a mobile cellular network operated via a mobility protocol that follows the DMM approach [28]. It aims at removing the hierarchical structure intrinsic of current mobile architectures. A flatter mobile network allows to anchor the data sessions at entities at the edge of the network. Hence, by placing the CDN nodes with such entities, we propose a MCDN capable of bringing the contents closer (in terms of network distance) to the mobile users. Thus, our goal is to deliver videos to the users at the expected quality while reducing the requested network resources for providing such services. We propose a delivery framework that retrieves the video content requested by the users from the nearest local CDN cache, when available, according to the mobility of the users. We leverage the Dynamic Adaptive Streaming over HTTP (DASH) protocol [29], as the streaming technique to deliver the media files to the consumers. DASH allows to divide the file into chunks, thus, when a video (or part of it) is highly requested in a given area, i.e., the media content becomes popular, the corresponding chunks are stored in the nearest local cache in that area. In this way, the MCDN system reduces the signaling in the mobile network and the overall video traffic (video chunks) between the video server and the radio access network (RAN).

The key concept of the DMM solution is that users IP flows are no longer anchored at a single centralized node located in the core network, but there are multiple anchors located at the edge. Indeed, in the current mobility model a single core entity conveys traffic to and from the access and is in charge of redirecting all the IP packets following the Mobile Node (MN) movements from one access network to another. Conversely, in DMM, we introduce a node called Mobility Access Router (MAR), that acts as default gateway for the MNs connected to its access links, but also anchors the flows started in its access network. For instance, when the MN is attached to the access network $A$, all the flows started there are anchored by the MAR located in that network, namely MAR $A$. This means that if the MN moves to the access network $B$, the ongoing connections are redirected by MAR $A$ from the access network $A$ to the access network $B$. Conversely, the new flows started by the MN when attached to the access network $B$ are anchored by MAR $B$. A MAR is linked to the rest of the Internet without the need to traverse other IP gateways, thus, from a topology perspective, the IP flows are bound to a node close to the terminal. Therefore,

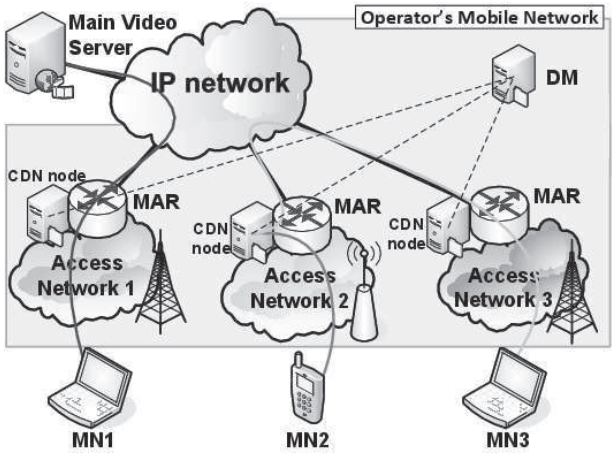



**Figure 3.2.** *Reference architecture with mobility.*

the DMM mobility protocol not only ensures session continuity to mobile users, but further enables a dynamic and flexible distribution of the traffic generated by the users since the data plane is no longer bound to the operators core network[1].

Our content delivery system takes full advantage of the described reference architecture, and is illustrated in Fig. 3.2. A main video server is located in the Internet and several nodes working as video caches (CDN nodes) are co-located with the MARs. Thus, the CDN nodes are installed at the edge of the mobile operators network to provide the popular contents as quickly as possible to the user and to reduce the use of network resources. Indeed, the MARs represent the closest location to the mobile user in terms of network distance. Moreover, we

[1]We assume for simplicity that one P-GW handles a single technology, arguing that it will make no difference if one P-GW manages more access networks

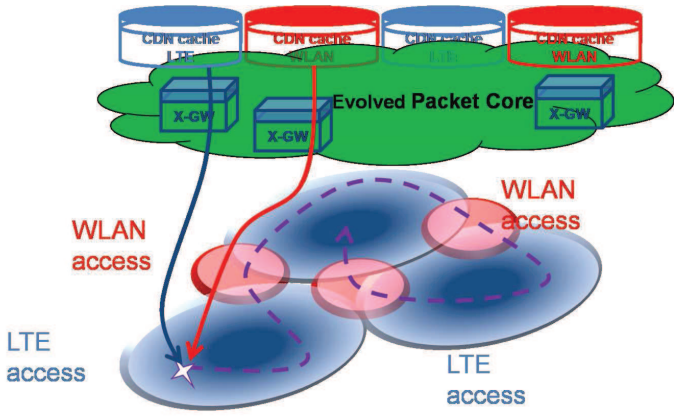



**Figure 3.3.** *Delivery architecture: from the video sources to the users.*

design an additional node called Decision Manager (DM) that handles a central database containing the information about the contents available in the local caches. This module is crucial for retrieving the requested content from the nearest available cache to the user and can be implemented either as a centralized entity or in a distributed manner. In Sec. 4.2, we describe how these three entities interact with each other to provide the closest point of service for the content requests generated by the MN.

## 3.3 Path Selection

We focus on a new range of optimizations taking into account traditional network metrics [30] as well as wireless specific metrics [31] with the goal of finding the end-to-end (E2E) video path to deliver the requested video content at the best available video quality. Our key idea is that, in a mobile network, a video application can be served through different paths, resulting in different video qualities perceived by the end users and possibly avoiding annoying service interruptions. On the other hand, the mobile operator needs tools to make available to the users network metrics such as routing costs and cache load associated to the different possible video caches, and to steer the applications towards an optimized usage of the network resources. Our tool is being defined by the Internet Engineering Task Force (IETF) Working Group ALTO (Application Layer Traffic Optimization) [32], which is designing the ALTO client-server protocol. The ALTO server provides the network operator's view of the network infrastructure, underlying an overlay application, to an ALTO

Client embedded in the mobile device. Through the client-server ALTO exchange, updated information (routing cost, load occupancy, etc...) about the video sources is made available at the mobile. The client, combining the information provided by the ALTO server with the monitored channel quality (WiFi and LTE), can univocally identify each available video path to the source of the content.

The algorithm presented in Sec. 4.3 is designed in a framework to be implemented in the mobile terminal. The whole mobile network architecture interacting with our framework is depicted in Fig. 3.3, which is a simplified vision of the EPC enhanced for optimal CDN integration [33]. CDN caches are integrated on top of the existing P-GWs and take into account the diversity of both cellular and WLAN access. The latest extensions of the 3GPP specifications propose the deployment of several P-GWs with both local and global scope. This way, the same content can be potentially downloaded from several sources, each having different properties with respect to the underlying wireless technology and round trip time delay. It should be further noted that the architectural principles depicted in Fig. 3.3 are an instantiation of the more general guidelines described in [34] where the authors give special attention to wireless heterogeneous access, QoE computation and advanced video coding techniques. All these elements are part of our solution design.

Leveraging on the ubiquitous wireless internet access given by either LTE or WiFi availability makes it possible to reliably stream the video content and to reduce the impact of a lack of sustainable video rate on the streaming process. Moreover, selecting the CDN video cache from which to download the video content taking into account the routing distance and the storage occupancy makes it possible to improve the responsiveness of the network.

The implementation details follow in Sec. 4.3.

Chapter 4

# Video Transport Mechanism Solutions

In this chapter we implement the video transport mechanisms introduced in Chap. 3 from an architectural point of view. Furthermore, a section dedicated to QoE-based resource allocation schemes for videos is given at the end of the chapter in Sec. 4.4. The following sections reflect the same order of topics addressed in Chap. 3. Thus, in Section 4.1 the robust opportunistic scheduling is implemented at the access network side, whereas in Section 4.2 the framework integrating the CDN concept with the DMM functionalities is implemented in the core network and, in Section 4.3, our E2E video path selection mechanism is developed. Each section is concluded with the related work to our solutions, which might be used in Chap. 5 for the sake of comparison.

## 4.1 Robust Opportunistic Broadcast/Multicast Mechanisms

### 4.1.1 Baseline solution

We start presenting a baseline solution that will be used as term of comparison. This mechanism simply operates based on the computation provided by the optimizer cited in [27], i.e., at a coarse-grained granularity. It works based on the long-term average user distribution without exploiting any knowledge about the instantaneous user distribution in each time slot. Hence, the scheduler does not operate opportunistically but rather serves a given video layer, in each time slot, to a pre-computed set of receivers which is not necessarily the actual set of users being able to demodulate the transmitted packets. This scheduling

procedure is inefficient since the set of users that can demodulate a certain MCS changes dynamically in the time and space domains.

### 4.1.2 Proposed solution

Opposite to the baseline solution, we propose an efficient use of the available wireless resources, i.e., channel capacity, by letting the BS know how many users can demodulate packets delivered with $MCS_1$, $MCS_2$, and so on, before scheduling the video packets from a video queue. Thanks to the feedback messages received from the users in the uplink, the BS updates the average user distribution which is needed to assign to each selected video layer a certain MCS and AL-FEC. We refer the interested reader to [27, 35].

The MCSs available at the eNodeB and the corresponding mapping of bits at the application layer to symbols in the physical channel is given in Table 4.1. This table shows that the higher the MCS, the shorter the transmission time of an information bit, i.e., the less the required channel resources. The difference between the average and instantaneous user distribution for a certain MCS measures the penalty incurred by a static solution based on average values compared to the opportunistic solution that relies on instantaneous knowledge. Based on the simulations in LTE systems [27, Fig.3], we can say that to each MCS there corresponds an average and an instantaneous user distributions (Fig. 4.1).

**Table 4.1.** *MCSs available at the base station.*

| MCS | Scheme & Coding Rate | Mapping Function $f(.)$ (bits/symbol) |
|---|---|---|
| 1 | QPSK 1/8 | 0.25 |
| 2 | QPSK 1/4 | 0.50 |
| 3 | QPSK 1/2 | 1.00 |
| 4 | 16-QAM 1/2 | 2.00 |
| 5 | 16-QAM 2/3 | 2.66 |
| 6 | 64-QAM 3/5 | 3.60 |

When channel conditions are good, the scheduler chooses a higher spectral efficiency MCS scheme for transmission, resulting in less time (i.e., less wireless resources in terms of channel occupancy) to transmit the video layer to a given number of users. This new MCS scheme is chosen so that the actual number of receivers (given the current channel conditions) meets the QoS requirements set by the operator in terms of number of users to be supported. The highest MCS which satisfies this constraint is chosen for transmission.

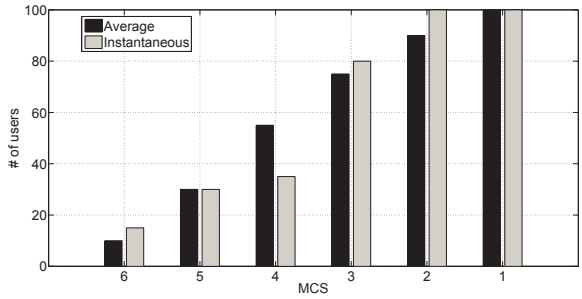



**Figure 4.1.** *Example of average and instantaneous user distributions at a certain time instant.*

Contrary to the baseline scheme, in our proposal the scheduler can change the choice of MCS in each time interval, based on the instantaneous received channel state information. While this scheduler does not increase the overall quality of the received video for the targeted clients, it uses a smaller fraction of the channel capacity than pre-computed by the operator. The potential saving in wireless resources by using a higher MCS can be used, e.g., for:

- adopting lower MCSs later on in case the channel quality decreases,
- other sessions (resource redistribution),
- retransmitting the most important video packets to increase the probability of correct reception, or
- providing additional AL-FEC from the video server.

Furthermore, we also consider the extreme case where the periodic evaluation for the current GoP overestimates the current channel conditions and user distributions, due to the high mobility of users which may quickly move from a MCS region to another. In this case, the scheduler cannot transmit packets with the assigned MCS since the number of users able to demodulate such packets is lower than what was pre-computed. Hence, the scheduler marks the packets so that they are transmitted at a lower MCS which is decoded by a number of users close to the designated one. In the worst case, this translates into having the physical layer send these packets with the lowest order MCS, which may lead to dropping high quality video layers which cannot be scheduled for transmission in the current GoP, due to the limited budget of channel rate reserved for each GoP.

Our opportunistic scheduling algorithm will be compared to a baseline scheduling algorithm in Sec. 5.1.

We now shortly present the pseudo-code for the robust opportunistic scheduler as in Algorithm 1.

### 4.1.3 Algorithm

**Algorithm 1** Opportunistic selection of the $MCS^*$ for video layer $v_i$.

**Input:** set of available $MCS_j$ schemes, $\mathcal{MCS}$, target number of users for each $MCS_j$, $t_{MCS_j}$, application layer bandwidth for layer video $v_i$, $b_{v_i}$, tolerance $\delta$ from the targeted average number of users to be served;

**Procedure: MCS selection for scalable video layer $v_i$ in the current time slot**

· Set of candidate MCS, $\mathcal{U} = \emptyset$;

· $j = 1$;

· Compute the number of users demodulating $MCS_j$, $n_{MCS_j}$;

**while** $n_{MCS_j} > (t_{MCS_j} + \delta)$ **do**

· $j = j + 1$;

· Compute the number of users demodulating $MCS_j$, $n_{MCS_j}$;

**end while**

**while** $n_{MCS_j} > (t_{MCS_j} - \delta)$ **do**

· Compute the required channel rate $C_{v_i,j} = b_{v_i}/f(MCS_j)$;

· $\mathcal{U} = \mathcal{U} \cup \{MCS_j\}$;

· $j = j + 1$;

· Compute the number of users demodulating $MCS_j$, $n_{MCS_j}$;

**end while**

· Compute the operational MCS scheme $MCS^* = \arg\ \min_{MCS_j \in \mathcal{U}} C_{v_i,j}$;

**Output:** MCS scheme for video $v_i$, $MCS^*$.

Let us assume first that a video stream is split into $L$ scalable video layers, $v_i$, with $i = 1, ..., L$. The inputs of the algorithm are the set of available $MCS_j$ schemes at the base station, $\mathcal{MCS}$, with $j = 1, ..., dim(\mathcal{MCS})$, the target average number of users demodulating $MCS_j$ to achieve a target QoS, $t_{MCS_j}$, the application layer bandwidth for a certain scalable video layer $v_i$, $b_{v_i}$, and the tolerance interval $\delta$ when comparing the instantaneous (actual)

number of users demodulating a $MCS_j$, $n_{MC_j}$ with the targeted average number of users $t_{MCS_j}$.

The selection of the MCS to be used is as follows. From the instantaneous user distribution, we compute the sets of users which can successfully demodulate a given $MCS_j$, $n_{MC_j}$. We assume that all users can be served with the lowest order MCS, i.e., $MCS_1$, and the higher the MCS, the smaller the subset of subscribers being able to demodulate such MCS. We also assume that once the scheduler selects a certain MCS for transmitting the base layer, say $MCS_{BL}$, the MCSs to be selected for the enhancement layers should be such that: $MCS_{BL} \leq MCS_{EL1} \leq MCS_{EL2} \leq ...$

For a given video layer to be transmitted, $v_i$, in the first while loop the algorithm skips all the lower order MCSs to avoid the risk of serving many more users than targeted, fulfilling the condition $n_{MC_j} < t_{MCS_j} + \delta$.

In the second while loop, the algorithm selects the candidate MCSs such that the number of users served is within the tolerance interval, i.e., fulfilling also the condition $t_{MCS_j} - \delta < n_{MC_j}$, and computes the required channel rate of the selected MCS (set $\mathcal{U}$), as a function of the bandwidth of the video layer and the MCS, $C_{v_i,j} = b_{v_i}/f(MCS_j)$. The mapping function $f(.)$ was presented in Table 4.1.

Afterwards, the algorithm chooses the $MCS^*$ to send the video layer $v_i$, which minimizes the channel usage within the set of candidate MCS, $\mathcal{U}$.

This scheme can be easily extended to a multi-class scenario, where users are grouped in different priority classes as a function of different subscription rates that they pay to the network operator. In this case, the channel conditions must be tracked for each individual user belonging to a class, and the MCS for transmission can be adapted such that it accommodates even the worst user of the given class. This will act as an extra constraint for the scheduling decisions, while the coarse-time optimization will no longer be based on the average number of users, but rather on the minimum imposed MCS for a certain class.

The simulation setup and the results achieved are presented in Sec. 5.1.

### 4.1.4 Related Work

Prior art focuses on three main aspects. First, in the case of unicast wireless transmissions, opportunistic schedulers are presented which act upon the instantaneous changes of

the channel conditions of the active users [36], [37], [38]. The scheduler then schedules for transmission packets for the user experiencing the best channel conditions at a given time instant. Furthermore, fairness issues may be taken into account in order to mitigate the problem of user starvation, in case a user experiences a bad channel for a prolonged period of time. These solutions mostly focus on the idea of defining fairness metrics among active users, and keeping history information about prior scheduling decisions. In general, the type of content being scheduled, and its relative importance among users, is not taken into consideration.

Second, the problem of opportunistic scheduling is addressed in broadcast scenarios [39]. The type of content usually matters in this context and so does the user preference for a given piece of content. The opportunistic scheduling decision is based on the user preferences/content popularity, and takes into account various constraints, e.g., transmission or storage capacity. From this point of view, the presented scheduling solutions function at a different layer, i.e., the application or control plane, and optimize the type of content that is transmitted at a coarse level. These solutions do not take into account the instantaneous channel variations perceived by the wireless users, and do not attempt an optimization of the channel resource allocation on fine grained time intervals in order to make the wireless transmission process more efficient. Some prior work provides an analytical treatment for the optimization of the user selection ratio in static and dynamic opportunistic multicast scheduling schemes by utilizing the extreme value theory [40, 41].

Finally, the third related area concentrates around the problem of optimizing the resource allocation at the application and network/MAC layer [42], [43], [44]. Usually algorithms are defined so as to choose the right application rate and adaptive modulation and coding scheme in order to maximize/minimize a given metric. These algorithms operate on a coarser time scale, and usually assume the knowledge of average channel conditions and application metrics. While the output of such algorithms can be used as input to our proposed innovation, our opportunistic scheduling mechanism is independent of the existence of such algorithms.

Opposite to the prior work, our scheduler takes into account the instantaneous channel conditions of the groups of users that subscribe to a particular media service and the relative quality difference between the different scalable video substreams, which are competing for

the wireless resources. It makes it possible to select the most convenient MCS to eventually reduce the wireless resource usage, while serving a target average number of users (expected by the operator), i.e., keeping the overall target QoS. This makes the system more robust to events requiring further wireless resources, e.g., leaving room for packet retransmissions or additional AL-FEC.

## 4.2 Mobile Content Delivery Networks and Video Popularity

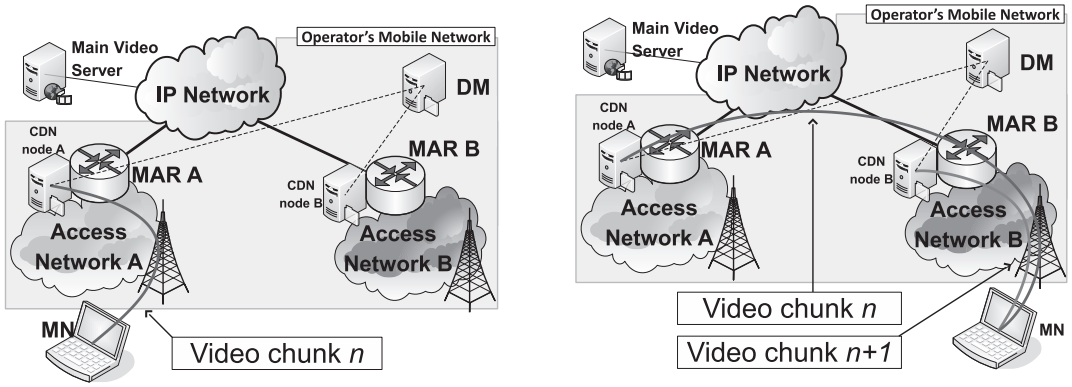
### 4.2.1 Proposed solution

Our mobile CDN system consists of: *i)* the main video server, *ii)* the local caches and *iii)* the DM.

The main video server is a video service provided by the network operator or a third party video service, e.g., YouTube. As such, it must not necessarily be aware of the CDN architecture in the mobile network. We assume that all video content is available in the video server which is reachable by the MN through the IP network and used as service point to download content that is not stored elsewhere in the network closer to the MN. The CDN nodes, upon a request from the DM, can either actively retrieve popular content from the video server or passively cache videos by intercepting the stream delivered to the MN.

The CDN node acting as local cache intercepts the requests received from the users in its coverage area. When the media is stored locally the cache acts as a service point and delivers straightforwardly the media files, otherwise it queries the DM about the best service point that can satisfy the request, and the delivery task is transferred to this latter entity. In parallel, the cache node continuously monitors and ranks during configurable time intervals the popularity of the contents based on the number of requests intercepted. This node registers such requests and the list of files being cached in a local database. At the end of each monitoring time interval, the node uploads the local database to the DM.

The DM uses the information conveyed by the local databases to build a global view of the origin of the requests (i.e., where the content is popular), and of the areas where the video is already available. This allows the DM to track the content and, consequently, to efficiently handle the requests of the users for the selection of the cache from where to retrieve the content. Moreover, the DM controls the files distribution in the system according




(a) Chunk delivery prior to handover.

(b) Chunk delivery after handover.

**Figure 4.2.** *System operations with mobility.*

to the popularity in the region covered by each CDN node. Indeed, if a CDN node currently lacks a content considered popular in its area, then the DM triggers the CDN node to cache the media, from either the video server or another CDN node.

In our scenario, when an MN requests a video content, it can be retrieved either *i)* from the video server, as shown in Fig. 3.2, or *ii)* from the closest cache to the user. Such cache can be exactly the one co-located with the gateway anchoring the MN, like MN2 of the same picture, or a cache co-located with another gateway in case the latter results to be closer than the main video server to the user, as for the video flow delivered to MN3.

In order to illustrate how our system handles the mobility of the users, we consider the example depicted in Fig. 4.2, where a MN moves within the mobile network while playing a popular video. We assume that this popular video is already cached in all the CDN nodes covering the area where the MN is roaming. Hence, the video request of MN is intercepted by the local MAR the MN is currently connected to, say MAR $A$, and the request is imme-

diately served with the delivery of the media by MAR $A$ (Fig. 4.2(a)). Each video chunk is streamed using an HTTP session, which means that each chunk downloading process represents a single IP flow. Therefore, when the MN moves to another access network, i.e., the terminal attaches to another access technology in both static and mobile conditions, the current chunk is redirected using the DMM technology by the old MAR, i.e., MAR $A$, to the new MN location, whereas next chunks are downloaded using the video cache at the new MAR, MAR $B$, as in Fig. 4.2(b)). In the case that the media is stored only at some MARs, or it is not available at all, the Decision Manager selects the CDN node to serve the request. This node can be, respectively, either a MAR in another access network, or the main server. We note that in this scenario, where the MN is currently attached, even if the MAR is not delivering the media it still serves as a mobility anchor, i.e., it ensures that the current chunk is delivered to the MN when the point of access changes.

The simulation and testbed setup and the results achieved are presented in Sec. 5.2.

### 4.2.2 Related Work

The standard solutions proposed for the delivery of videos over a fixed network topology cover the area of CDNs, which make it possible to delocalize the video contents to local caches disseminated in the network rather than storing them at a video source (server). Open source CDN systems such as [45] focus on the development of static CDNs for specific video streaming applications, for instance Darwin Streaming Server by Apple [46] or Helix Universal Server by Real [47].

When dealing with mobility, CDNs are detrimental to performance due to the user's dynamics which causes quick changes of the popularity distribution of the video contents in the network. The design of MCDNs [10] goes along with the need for software tools that allow to split the video into chunks so that the content provider gains the flexibility to route the requests from the mobile user to the closest local cache from where to retrieve the file [48]. To this end, DASH [29, 49] is a novel streaming technology providing a well recognized solution in the community. It is widely used, due to its capability to deliver the partitioned video segments to the client via HTTP and to adapt the streaming bitrate to the link capacity.

The advantages of content caching in mobile networks have already been discussed

in [50], where the authors analyze the impact of deploying caches at different hierarchy levels of a General Packet Radio Service (GPRS) network. The work provides a cost metric to evaluate the effects of placing caches in both national data centers, where usually an operator's Gateway GPRS Support Node (GGSN) is placed, and regional data centers, where the Serving GPRS Support Node (SGSN) entities are usually located. The authors highlight the result that it is beneficial to install caches in the regional data centers.

We consider the EPC architecture [23] rather than the GPRS one, and, in such context, we can translate the above concept by placing the caches co-located with the S-GW. Nevertheless, according to the mobility protocol used in the EPC, when a MN starts a packet data session, the IP connections are topologically anchored at the P-GW by means of tunneling procedures. Therefore, the current model for mobile networks forces the user plane to always traverse the operator's core network, and the advantages of having a flatter content distribution are lost. According to this reasoning, we propose an architectural framework where a CDN system is integrated in a mobile network operated by a mobility protocol that follows the DMM approach. This new mobility paradigm is currently matter of research in the IETF DMM Working Group [51]. The goal of it is to find alternative solutions to the current standard protocols for IP mobility (e.g., Mobile IPv6, Proxy Mobile IPv6, etc.), aiming at flattening the mobile network, which allows to anchor the data sessions to entities at lower hierarchy levels, e.g., the S-GW. We further design a novel MCDN architecture which makes use of the novel mobility protocol DMM to efficiently deliver the video contents requested by the mobile users with the target of keeping the continuity of the video session in the presence of handovers among heterogenous RANs (for instance employing WiFi and LTE radio technologies), thus serving the video consumer with the expected quality, while reducing, by means of DASH technology, the network costs associated to the video transport mechanisms.

## 4.3 Path Selection

In video applications the user requests a video to the operator, which is in charge of providing such service at an agreed-upon target video quality. In the mobile network, the requested video may be stored in more than one CDN video caches and the operator selects one of these video sources to provide the requested service. Once a CDN cache is selected,

the path for the delivery of the video from the source to the end user is established. It is necessary to define the network metrics which are used to assess a video path from the source to the end user. Once the set of metrics is identified, we can represent each of the $N$ available video paths with a tuple (vector) $V_j$, $j = 1, ..., N$. For a given path $j$, the $i$-th element of $V_j$, i.e., $P_i^j$, is the value taken by the $i$-th metric, $i = 1, ..., M$. Therefore, the input to the path selection algorithm is the set of $N$ tuples as follows:

$$V_j = \{P_1^j, P_2^j, ..., P_M^j\}, \quad j = 1, ..., N.$$

### 4.3.1 Optimization problem

The goal of our work is to find the $M$-tuple, i.e., the video path, that maximizes a measure of the proximity to an ideal $M$-tuple formed by the most desirable values taken by each component. In order to make tuples comparable, a first step is to adapt the values taken by the metrics, so that the lower the value, the better the performance (possibly taking the inverse value when the contrary holds). Then, we map the values in the tuples to the interval [0,1], in order to make the resulting values associated to the original tuples indicate the proximity to the best available value, i.e., the closer the values to 1, the better the tuple and the more likely the corresponding path to be selected, and the closer to 0, the worse the tuple. Hence, we design our optimization algorithms with the aim of maximizing a utility function defined on such "normalized" $M$-tuples.

### 4.3.2 Proposed Solution

The first optimization algorithm we design reflects the network operator's point of view. The aim of a network operator is to offer a certain video service while taking care of the overall performance of the network. Thus, a suitable operator utility function maximizes the sum of the proximity values associated to the metrics in the $M$-tuples (*max-sum* criterion). The second optimization algorithm reflects the user's point of view, for whom it is desirable to reduce the impact of possible weak points in the delivery chain, e.g., bottlenecks, and in general to ensure that the worst-case performance is maximized. Thus, a suitable user utility function maximizes the minimum proximity value associated to the metrics in the $M$-tuple (*max-min* criterion).

The set of network metrics, involved in the optimization procedures, needs to be (i) identified, to specify each possible video path, and (ii) associated to application performance metrics, in order to evaluate the performance of the optimization algorithms. In the first step, we separately define the metrics which characterize the Core Network (CN) side and the Wireless Access Network (WAN) side of the video path. For this, we assume that in the CN the delivery delay of the video is impacted by the number of links and nodes in the path and by the number of requests to the caches, while the channel can be assumed to be without losses. In the WAN, on the contrary, the delay has a minor influence on videos (last hop), while the data rate is severely impacted by the wireless channel conditions experienced by the user.

Thus, in our framework, we associate (i) the CN-related metrics to the application performance metric (APM) of video delivery delay, here defined as the response time of a CDN video cache for the delivery of the requested video and the time to travel through the selected path, and (ii) the wireless metrics to the APM of channel capacity offered to the end user in the wireless hop. This mapping is detailed later in this section.

We now present the *max-sum* and *max-min* optimizations, as shown in Algorithm 2.

### 4.3.3 Algorithm

In every time slot, a set of tuples $\{V_j\}$, with $j = 1, ..., N$, is given as input. The values taken by the network metrics are such that the higher the value, the worse the performance (e.g., the inverse of the channel capacity value is taken). As a next step, both algorithms compute the minimum (ideal) value taken by each of the $i$-th element of the tuples, expressed as $P_i^*$. The collection of these ideal values gives the ideal tuple (ideal path) $V^* = \{P_1^*, ..., P_M^*\}$. Hence, the ratios of the value taken by each element of the ideal tuple to the value taken by the corresponding element of the evaluated tuple ($\overline{P_i^j}$) give the "mapped" version, i.e., within the interval [0,1], of the original set of vectors, i.e., $\overline{V_j}$, with respect to the ideal tuple. In order to make metrics comparable within the tuple, the linear mapping is such that the ideal target value of a metric is associated to 1 and the worst value is associated to 0. Thus, $\overline{V_j}$ can be seen as the proximity vector to the ideal $V^*$. The metrics involved in the optimization problem are then weighted with coefficients $a_i \in [0, 1]$ to tune the algorithms to meet the operator's preferences. For instance, when the weighting coefficients of the CN-related

**Algorithm 2** Max-sum and max-min optimization algorithms.

**Input:** $\{V_j\}$, with $V_j = \{P_1^j, ..., P_M^j\}$, $\{a_i\}$;

**Procedure:**

**for** $i = 1 \rightarrow M$ **do**

· $j^*$= $\underset{j}{\text{argmin}}\{P_i^j\}$;

· $P_i^* = P_i^{j^*}$;

**end for**

Ideal tuple: $V^* = \{P_1^*, ..., P_M^*\}$;

**for** $j = 1 \rightarrow N$ **do**

· Compute $\overline{V_j} = \{\overline{P_1^j}, .., \overline{P_M^j}\}$;

· Update $\overline{V_j} = \{a_1\overline{P_1^j}, .., a_M\overline{P_M^j}\}$

**end for**

**1) Max-sum selection: (Operator)**

· $j_o = \underset{j\in\{1,...,N\}}{\text{argmax}} \sum_{i=1}^{M} a_i\overline{P_i^j}$;

· $V_o^s = V_{j_o}$;

**2) Max-min selection: (User)**

· $j_u = \underset{j\in\{1,...,N\}}{\text{argmax}} \underset{i\in\{1,..,M\}}{\min} a_i\overline{P_i^j}$;

· $V_u^s = V_{j_u}$;

**Output:** Selected vectors $V_o^s$ and $V_u^s$.

metrics are set to $0$, only the wireless access matters, thus the framework optimizes the selection of the wireless access technologies available at the mobile and thus prioritizes the APM of the channel capacity. If the weighting coefficients of the wireless metrics are set to $0$, then the core network status determines the optimal path selection, no matter what wireless access is selected, and priority is given to the APM of delivery delay. The two algorithms now select the tuple $V_j$ maximizing their own proximity function. The *max-sum* algorithm selects the path $V_o^s$ which maximizes the sum of the proximity values whereas the *max-min* algorithm uses the minimum proximity component value to find $V_u^s$.

The optimization algorithms compute the video path to be used for the video delivery with a complexity that grows linearly with the number of metrics $M$ and with the number of available unique paths $N$. We keep the computational costs of the algorithms as low as possible while meeting the mobile phone limited capabilities and maintaining the minimum necessary amount of information to distinguish each unique video path. Hence, in our framework, we restrict the set of metrics to $M = 3$ as follows. We define two metrics on the CN side of the video path. The first metric is the routing distance to a specific End Point (EP), i.e., a CDN cache, expressed as the number of hops between the mobile device and the EP. The second CN metric is the EP memory occupancy information, i.e., the ratio of used storage. The values taken by the CN metrics are communicated by the ALTO server in the core network to the ALTO client in the mobile, possibly via ALTO protocol extensions enabling joint transmission of multiple metric values and supporting such an EP occupancy as proposed in [52]. Then, we define a wireless-related metric which takes into account the channel quality, e.g., the Signal-to-Noise Ratio (SNR), for both cellular and WLAN access. This value can be directly measured by the mobile terminal from the wireless interface. Further metrics can be defined to better represent core and access network status on one hand and to better evaluate the performance metrics of interest on the other hand. However, the values taken by most of these metrics cannot be communicated to/from the mobile terminal in real-time, opposite to the aforementioned 3 network metrics. Moreover, adding network metrics to the optimization problem further increases the complexity of the system, running the undesirable risk of growing the computational time of the algorithm beyond the hardware capabilities, making it infeasible in practice. The analysis of the impact of the number of metrics $M$ on the practical feasibility of our algorithms is out of the scope of this work

and is left for future work.

In the following, we explicitly map the network metrics involved in the optimization problem to the application performance metrics (APM) evaluated in our Matlab simulator, namely channel capacity and response time.

**Channel capacity**

The wireless access metric, defined as the SNR of the wireless channel, is mapped to the upper bound of the achievable transmission rate:

$$C = B \log_2(1 + SNR)$$

where the channel capacity $C$ is the product of the bandwidth of the system $B$ with the base 2 logarithm of $(1 + SNR)$.

**Response time**

The two CN-related metrics, i.e., cache load and routing distance, are mapped to the response time of the network for releasing the video requested by the end user. That is, the time for transmitting a video from the CDN node to the mobile user is assumed to be proportional to the number of hops $N_{hops}$ in the path by a unit measure of response time per hop[1], while the cache load is translated to the time needed to retrieve the video from the storage in the cache (hence, proportional to the fraction of storage in use, $CDN_{storage}$). Thus, the response time $T$ is the sum of these two components: the time to travel through the whole path $T_{hops}$ and the time spent to get the file requested from the cache $T_{retrieve}$. The latter can be also thought as the time to travel through the first node in the core network, i.e., the video source (CDN cache).

$$T = \underbrace{\alpha\, N_{hops}}_{T_{hops}} + \underbrace{\beta\, CDN_{storage}}_{T_{retrieve}}$$

$\alpha$ and $\beta$ are chosen by the operator to best fit the network characteristics and status at the time of the video request. In our experiments, for the sake of simplicity, $\alpha$ is set to the time unit of response time, i.e., 1 ms (average time to travel through one hop). $\beta$ reflects the

[1]For the sake of analysis, we set to 1 ms the time unit of response time per hop. This is an average value we observed for a range of experiments run with the trace-route tool.

characteristics of the Solid State Drive (SSD) and it is set as well to 1ms. If the Hard Disk Drive (HDD) technology is implemented instead, then $\beta$ increases by a factor $\sim$15 due to the slower random access time compared to the SSD [53].

### 4.3.4 Validation

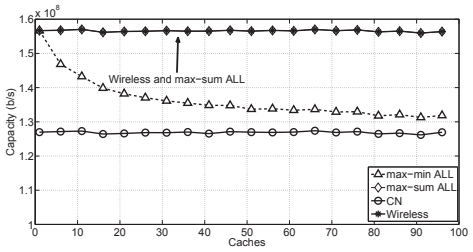
In this section we investigate how the two optimization algorithms perform with respect to the number of CDN sources deployed in a mobile network. Starting from the results achieved in this section we further develop our delivery framework in ns-3 to take into account user mobility and network load, as it will be shown in Sec. 5.3.

We implement in Matlab a mobile network where we deploy a number of CDN caches on the core network side, at a given number of hops (within the range $[1, 20]$) from the wireless access networks available and with a given fraction of memory occupancy, and we implement the LTE and WiFi modules to simulate the wireless access network part. On the mobile user side we implement the module in charge of selecting online the path for the delivery of the video from the source (CDN video cache) to the user. Thus, the module can run the *max-sum* and the *max-min* optimization algorithms based on the aforementioned metrics.

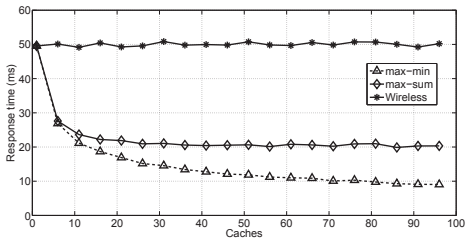
Our test scenario is implemented as follows. Consider a mobile user moving in an LTE micro-cell, with 20 MHz of bandwidth and a coverage radius of up to 3 km. We further deploy 3 WiFi 802.11n spots, with a bandwidth of 20 MHz, at regular intervals of 0.5 km from the LTE base station along the same radius. The user moves from around 2 km to 500 m away from the base station, then turns back and moves up to 3 km away from the base station. The simulation time is slotted (1 s per slot), and the total number of slots needed to travel through this path is set to $S = 800$.

We consider path loss and shadowing effects in both wireless models. Clearly, once the user is in the range of a WiFi hot spot the exploitation of this additional access technique may be beneficial to both the user and the network operator. For the latter, it will result in additional available capacity to redistribute to the users that are not covered by any WiFi hot-spot. For the former, video quality can be increased when keeping an arbitrarily high target video rate.

As mentioned before, the wireless metric in use (SNR) is mapped to the channel capacity,

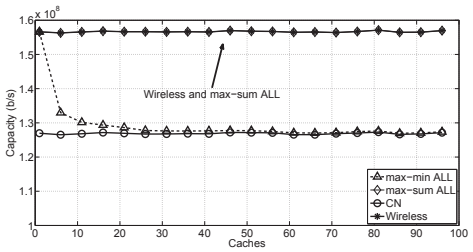



(a) Channel capacity vs. number of caches.

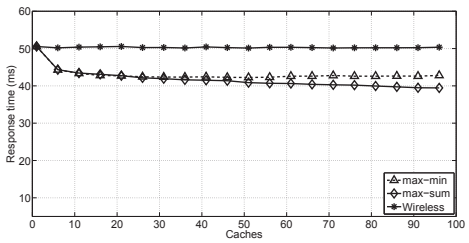



(b) Response time vs. number of caches.

**Figure 4.3.** *Impact of the* max-sum *and* max-min *algorithms on response time and wireless channel capacity in the* balanced *scenario when considering: i) "ALL": all metrics; ii) "CN": only the CN-related metrics and iii) "Wireless": only the wireless metric.*




(a) Channel capacity vs. number of caches.

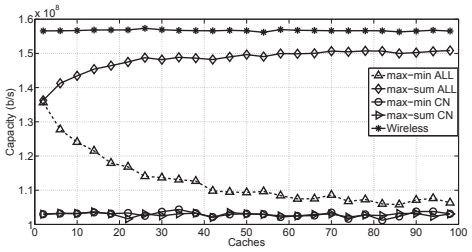



(b) Response time vs. number of caches.

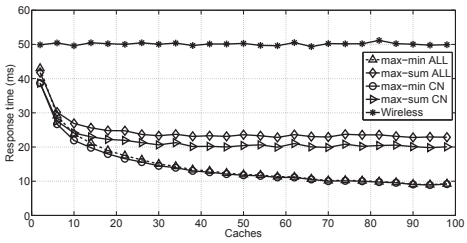
**Figure 4.4.** *Impact of the* max-sum *and* max-min *algorithms on response time and wireless channel capacity in the* unbalanced *scenario when considering: i) "ALL": all metrics; ii) "CN": only the CN-related metrics and iii) "Wireless": only the wireless metric.*

and the two CN-related metrics are mapped to the response time of the network. We evaluate the trade-off between channel capacity and response time as the number of available caches in the core network increases, given that the best path is selected within the range of available paths (tuples) by using either the *max-sum* or *max-min* optimization algorithms as detailed in Section 4.3.2.

Both optimization algorithms work at two levels of time-granularity. The wireless channel quality, due to its nature and to the user mobility, quickly fluctuates at a fine-grained scale, thus varying at each time slot in our simulator. On the contrary, the metrics associated to the core network change at a coarser-grained scale, say, every 10 slots. In fact, the storage of the cache and the number of hops of a video path vary more slowly in time (e.g., switching caches on/off for energy saving purposes and so on).




(a) Channel capacity vs. number of caches.




(b) Response time vs. number of caches.

**Figure 4.5.** *Impact of the* max-sum *and* max-min *algorithms on response time and wireless channel capacity in the* restricted *scenario when considering: i) "ALL": all metrics; ii) "CN": only the CN-related metrics and iii) "Wireless": only the wireless metric.*

We set the two CN-related network metrics to draw 3 network scenarios, named the *balanced*, *unbalanced* and *restricted* scenarios. The first two scenarios are such that the CDN caches deployed in our simulator are connected to both LTE (base station) and WiFi (hotspot) access networks, while in the third scenario the caches can only have either an LTE or a WiFi connection, not both. In the *balanced* scenario, the values of cache load and routing distance slightly change around an average value set for all CDN caches, i.e., the video load is shared among the caches. In the *unbalanced* scenario we set the values of the CN-related metrics so that the closer the cache to the base station, the lower the routing distance and the higher the video load in the storage; on the contrary, we assume that the farther the caches from the base station, the higher the routing distance and the lower the cache load. In the *restricted* scenario, the caches are set similarly to the *balanced* scenario but, opposite to it, can only have either an LTE or a WiFi connection (randomly set).

We compare the performance of the two optimization algorithms when: (i) the network metrics in the tuples all have the same importance (label "ALL" in the plots of Figs. 4.3,4.4,4.5), (ii) only the CN-related metrics or (iii) only the wireless metric are used for the optimization (label "CN" and "Wireless", respectively, in Figs. 4.3,4.4,4.5).

### 4.3.5 Discussion

The *max-sum* and *max-min* algorithms, when used for optimizing the wireless access part ("Wireless"), show the same performance in terms of both response time and wireless capacity as the number of caches increases in all scenarios. Since the optimization problem

is reduced to one element of the tuples, both algorithms select any tuple (path) under the condition that the wireless access in use is the best available one, and therefore give the same result. Thus, the wireless capacity is always maximized, while the response time is as high as 50 ms, the maximum value in our simulations. In each plot, we grouped the curves into one labeled "Wireless" to represent both algorithms when only the access part is optimized. For better clarity in the plots, we have also grouped the overlapping[2] curves into one, labeling it according to the common feature. For instance, in the *max-sum* case "ALL" and "CN" curves overlap, so we group them into the curve *max-sum* (the same holds for the equivalent case of *max-min*). The same happens for *max-sum* "CN" and *max-min* "CN", grouped into the curve "CN".

We compare the performance of the two optimization algorithms by studying the trade-off between wireless capacity and response time as the number of deployed caches increases in the network, for the *balanced*, *unbalanced* and *restricted* scenarios in Figs. 4.3,4.4,4.5, respectively.

In the *balanced* scenario (Figs. 4.3(a) and 4.3(b)), the *max-min* algorithm performs worse than the *max-sum* algorithm in terms of wireless capacity when all metrics are used for the optimization, otherwise they perform similarly (for this reason we present one curve labeled "CN" for both algorithms). In this scenario the best trade-off is given by the *max-sum* algorithm when few caches are deployed (minimum response time, average wireless capacity) while the *max-min* "ALL" algorithm is preferable for a higher number of caches (maximum wireless capacity, low response time).

Similar conclusions can be drawn in the *unbalanced* scenario in terms of wireless capacity, except for a steeper decrease of the capacity for the *max-min* "ALL" algorithm. In terms of response time, we notice that the *max-sum* algorithms switch the performance with the *max-min* algorithms when compared to the *balanced* scenario. This is due to the fact that when the cache load and routing distance are well distributed in the network, i.e., on average the caches are equally convenient to request a video, the *max-sum* criterion does not discriminate among caches, while for the *unbalanced* scenario the joint selection (*max-sum* function) makes it possible to find a compromise between load and routing distance, thus reducing the overall response time. In general the response time of the proposed algorithms in this

[2]Here overlapping means that the difference between the values taken by the curves is sufficiently small.

scenario is detrimental to performance (only 15 % less than the maximum delay) compared to the *balanced* scenario.

In the *restricted* scenario (Figs. 4.5(a) and 4.5(b)), both *max-min* and *max-sum* "ALL" algorithms perform in between compared to the respective "CN" and "Wireless" algorithms, in terms of both wireless capacity and response time. We notice that for the wireless capacity, with the increase of the number of caches, the *max-min* and *max-sum* "ALL" algorithms perform similarly to the "Wireless" (top, Fig. 4.5(a)) and to the "CN" (bottom, Fig. 4.5(a)) algorithms, respectively.

A general conclusion we can draw from these simulation results is that, compared to the baseline solutions, i.e., "CN" and "Wireless", the algorithms that jointly optimize core and access network related metrics are able to find a path with a good trade-off between wireless capacity and response time for a broad range of settings. The *max-sum* criterion gives the best performance in the *unbalanced* scenario, but in general it privileges the wireless capacity over the response time compared to the *max-min* criterion. As mentioned before, it is possible for the operator to fine-tune the weights of the metrics in the tuples to adapt the performance of the proposed algorithms to the actual network settings and to its own targeted performance and network resources.

### 4.3.6 Storage cost and cache deployment

In this section we re-consider the plots in Figs. 4.3,4.4,4.5 to determine the number of caches to be deployed by the operator in order to get a satisfactory trade-off between wireless channel capacity and network response time. Moreover, we consider the real costs associated to the cache storage as a further criterion for selecting the proper number of caches to be deployed. Table 4.2 reports some data taken from [54] about the costs associated to the amount of storage in the cache per GB per month/year, where we assume, as a concrete example, that each video cache can store 10TB.

Based on the plots in Figs. 4.3,4.4,4.5, we can say that a good trade-off between wireless channel capacity and network response time is achieved when the number of caches is limited, with small differences among the scenarios under study. Operating with few caches can already reduce the network response time to low values in our simulations, with a negligible impact on the wireless channel capacity when using the *max-sum* algorithm. At the

**Table 4.2.** *Costs associated to the storage of a cache.*

| GB | $ per month/GB | $ per year |
|---|---|---|
| 10240 | 0.1 | 12288 |
| 102400 | 0.093 | 118579.2 |
| 1024000 | 0.070 | 976281.6 |

same time, the monetary costs are kept low compared to solutions that enjoy slightly better performance but are significantly more costly due to the higher number of caches required for the deployment. In addition, as already presented in [55], the distribution of CDN caches in nodes implementing other protocol functions such as access routers, access points or base stations is beneficial to reduce costs. In [56], based on monetization considerations for the CDN market, the authors show that CDNs are an important component in the video supply chain for mobile operators, and that the most profitable way of dimensioning a CDN is to have a small number of caches each serving a large number of users, which is consistent with our results in Figs. 4.3,4.4,4.5.

We evaluate in Sec. 5.3 the performance of the framework by means of simulation in ns-3 [57], an event-based network simulator that provides a detailed implementation of the LTE core and access networks. We refer the interested reader to [58, 59] for further information on the LTE modules implemented in ns-3.

### 4.3.7 Related Work

In a mobile network, a video application can be served through different paths crossing both the core and the access networks, resulting in different QoE at the user side. The main factors affecting the quality delivered by each path are related to error propagation phenomena [60], the latency response time [55] and the available application video rate [61]. In prior works, access and core network metrics are decoupled and studied independently from each other. This way, the benefits of optimizing one side of the network might be wasted in the other side of the network. Metrics related to the access network can be measured at the end user via the wireless interface and fed back to the network, possibly with tight granularity so as to track user mobility and channel variations. Based on this information, the mobile operator is in charge of managing the video service to keep a target QoE and to optimize

the network resource usage. CDN is a crucial design choice for network operators, which enables large scale content distribution at a reasonable cost and without overloading the operator's core network. Distributing video caches close to the users partly compensates for the significant increase of video traffic in mobile networks, which is reported to double every year [62]. A mobile CDN solution for video delivery includes network based caching, network guided optimization of content delivery and advanced multicast solutions. This requires a continuous monitoring of the current conditions of the entire system, in particular the status and distribution of the CDN nodes [32], as well as the popularity of contents. Using the collected data the system dynamically maintains an optimal configuration of a set of servers for content distribution and selects optimal sources for transmitting the video to the user.

The problem of optimal cache management is investigated in the community and, for instance, its impact on the QoE of the user is studied in [63]. The authors encode a video into multiple copies and select the optimal set of playback rates (in response to varying channel conditions) to be made available in the network for a fixed number of cache copies and then find the optimal number of cached files to achieve a target QoE. In contrast to this work, our goal is to select a video path in response to a set of network metrics which impact the final QoE of the user and depict the whole video delivery chain, i.e., from the selection of the video source to the choice of the access network technology.

The framework proposed in [64] aims at optimizing CDN caching by taking into account the server storage status, network path status and content utility. By monitoring these metrics, the system is able to update the overlay delivery chain (CDN caches) and instructs the mobile clients from where to optimally download contents. However, the paper does not take into account the wireless hop, which is very often the bottleneck of the entire delivery chain. In [65] authors leverage the storage and computing power of the core network nodes to collect the network status information necessary to compute multi-path scalable video delivery strategies. The authors make simultaneous use of multiple paths to deliver a set of video layers increasing the QoE perceived at the end user, whereas in our framework we select a single video path based on an optimal selection of network metrics from the core to the access network nodes in order to deliver the best video quality to the user.

Conversely, [66] focuses on the wireless performance analysis (3G, 4G, WLAN) of an

Internet Protocol Television (IPTV) delivery architecture when changing the placement and characteristics of the streaming servers at the access network. The goal of [66] is to make traditional IPTV available to users anywhere, anytime, on any device and through any network. The authors propose a new CDN-based architecture to distribute content to different access networks by placing video streamers in the access networks (including mobile), thus reducing bandwidth consumption and quality of service (QoS) restriction due to the IP core network. Our architectural assumptions for content distribution across wireless heterogenous technologies are in line with this work; however, contrary to our framework, the performance analysis in [66] is limited to the wireless side of the framework when changing the servers placement.

The work described in [55] deals with the distribution of the CDN caches and aims at optimizing the QoS and latency response time and the energy consumption. Although in our work we do not specifically focus on energy aspects (rather on network resource savings), similar considerations can be derived when distributing the caching system. First of all, it has been demonstrated that local caches reduce the transport costs to a great extent, secondly mobile CDNs help the service providers to address their scalability issues. The work proposed in [55], however, differs significantly from our approach, since the video delivery chain is considered to be wireless medium agnostic, lacking an end-to-end viewpoint on the video transport optimization.

We further stress the point that prior solutions for video transport optimization over cellular networks cover different areas, where each of them is investigated and developed independently. For instance, in [67], a cross-layer scheduler for video transmission is presented, focusing on the wireless access side only, without taking into account the network resources as a whole. Network operators target the efficient utilization of the overall network resources while providing high quality video transport, thus requiring novel designs and raising multidisciplinary topics [68]. A combination of video transport optimization mechanisms considering the entire delivery chain over mobile networks was proposed only very recently in [12], where interworking traffic shaping mechanisms are conceptually discussed. In contrast to the aforementioned related works, we optimize the selection of the entire video delivery chain taking into account both access and core network metrics and we develop a delivery framework with the goal of reducing the packet delivery delay while

keeping the requested data rate of the video services and thus the perceived video quality. Eventually, we make use of traffic shaping techniques at the access points to further reduce the delivery delays while preserving the requested throughput.

## 4.4 QoE-based video transport

At the current stage, mobile network operators face the issue of supporting high quality video services with the available network resources. The widespread high-speed wireless coverage will likely increase the number of users that require high quality video services, making the support of such a traffic in the access networks a challenging question.

An attractive solution in this scenario consists in dynamically adapting the QoE perceived by the final video consumers to the available transmission resources by adjusting the video code rates. As observed in [69], in fact, reducing the encoding rate of a video is much less critical in terms of QoE degradation than increasing the packet loss probability or the delivery delay. However, the perceived QoE at a certain encoding rate depends on the specific characteristics of the video, e.g., scene and source dynamics and frame-by-frame motion.

In this section we propose a novel approach to handle under-provisioned video traffic scenarios that is based on the possibility of dynamically adjusting the video encoding rate and that takes into account the different QoE behaviors of video sequences, expressed in terms of Structural Similarity (SSIM) index [70]. We indeed measure the SSIM of a large set of H.264-AVC [71] video clips [72,73] coded at different rates, which correspond to different perceived quality levels. After a suitable normalization and rescaling of the metrics of interest, we are able to analytically approximate the perceived quality characteristic of each video by means of a simple 4-degree polynomial expression. We hence associate each video to its polynomial coefficients in order to compactly describe how the SSIM degrades for lower transmit rates. To reduce the state space, we also proposed a class-based approach where videos showing similar SSIM *vs* rate relations are grouped in a class and tagged with a set of polynomial coefficients that characterize that class. We then define Resource Management (RM) and Video Access Control (VAC) algorithms that take into account this QoE-based information associated to each video to maximize a certain utility function. As a proof of

concept, we apply our approach to a simple scenario with a congested link shared by multiple video flows.

### 4.4.1 Evaluation Setup

We evaluate the objective QoE of the videos with the SSIM index, which is a full reference metric that allows to measure the structural similarity between two frames of the same video. From the human eye perspective, SSIM improves the representation of the perceived video quality compared to traditional metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE). These pure mathematical metrics estimate the perceived distortion based on analytical pixel-by-pixel comparison. On the contrary, SSIM measures the image degradation in terms of perceived structural information change, thus taking into account the tight inter-dependence between spatially close pixels which contain the information about the objects in the visual scene. SSIM is calculated via statistical metrics (mean, variance) computed within a square window of size $N \times N$ (typically $8 \times 8$), which moves pixel-by-pixel over the entire image. The measure between the corresponding windows $X$ and $Y$ of two images is computed as follows:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{4.1}$$

with $\mu$ and $\sigma^2$ the mean and variance of the luminance value in the corresponding window, and $c_1$ and $c_2$ variables to stabilize the division with weak denominator (we refer the interested reader to [70] for more details on the computation). For practical reasons, we take the average values of the SSIM index for each video.

The range of the SSIM index goes from $0$ to $1$, which represent the extreme cases of totally different or perfectly identical frames, respectively. Table 4.3 shows the mapping between SSIM and Mean Opinion Score (MOS) scale, as reported in [74].

### 4.4.2 Video evaluation

We consider a pool of $V = 38$ CIF video clips, taken from standard reference sets, e.g., [72]. Each video has been encoded with the Joint Scalable Video Model (JSVM) reference software [75] into H.264-AVC format at $C = 18$ increasing compression rates, which correspond to as many quality levels. The list of video names, full quality transmit rate, duration and classification of video motion is provided in Table 4.4.

**Table 4.3.** *Mapping SSIM to Mean Opinion Score scale*

| SSIM | MOS | Quality | Impairment |
|---|---|---|---|
| $\geq 0.99$ | 5 | Excellent | Imperceptible |
| $[0.95, 0.99)$ | 4 | Good | Perceptible but not annoying |
| $[0.88, 0.95)$ | 3 | Fair | Slightly annoying |
| $[0.5, 0.88)$ | 2 | Poor | Annoying |
| $< 0.5$ | 1 | Bad | Very annoying |

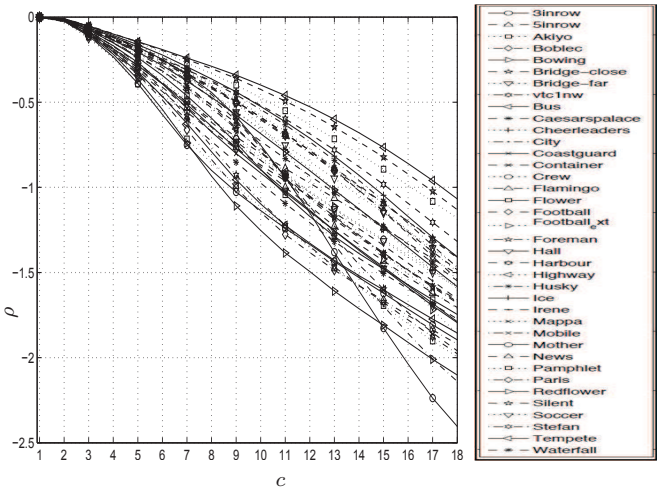



**Figure 4.6.** *Logarithm of the normalized rate* $\rho_v(c)$ versus *compression level* $c$ *for different video clips.*

We denote by $c \in \{1, \ldots, C\}$ the available compression rates and by $r_v(c)$ the transmit rate of video $v \in \{1, \ldots, V\}$ encoded at rate $c$, with $r_v(1)$ being the maximum (i.e., full quality) rate. To ease the comparison between different video clips, it is convenient to normalize the video rates to the full quality rates. Moreover, following the Weber-Fechner's law that speculates a logarithmic relation between the intensity and the subjective perception of a stimulus, we introduce a logarithmic measure of the normalized rate, here named *Rate Scaling Factor* (RSF), which is defined as

$$\rho_v(c) = \log(r_v(c)/r_v(1)) \,, \tag{4.2}$$

and shown in Fig. 4.6 when varying the compression level $c$ for the different videos.

We can see that the compression level, i.e., the number of quantization points considered in the H.264-AVC encoding, determines the rate of the video sequence depending on the content of the video itself. For a given compression level $c$, the larger the RSF $\rho_v(c)$ the more dynamic the video sequence. Indeed, dynamic sequences exhibit lower spatial and temporal correlation of consecutive video frames and, hence, are less amenable to compression.

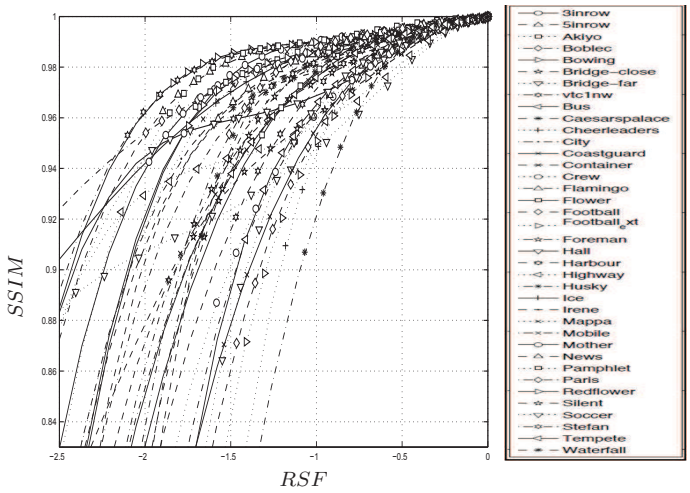



**Figure 4.7.** *SSIM of the different video clips when varying the RSF.*

The dynamics of the video content also impact the perceived QoE for a certain RSF value, as clearly shown in Fig. 4.7 that shows the average SSIM of each video clip when varying $\rho_v$. The markers correspond to empirical SSIM values for the $V$ videos in the test set, while the lines are obtained from a $4$-degree polynomial approximation of such values.

We observe that the polynomial approximation is, in general, acceptably accurate for the range of $\rho$ of practical interest. Therefore, the SSIM characteristic of a video $v$ can be approximated as

$$F_v(\rho) \simeq 1 + a_{v,1}\rho + a_{v,2}\rho^2 + a_{v,3}\rho^3 + a_{v,4}\rho^4 \,. \tag{4.3}$$

The vector of coefficients $\mathbf{a}_v = \{a_{v,i}\}$ provides a compact description of the relation between the perceived QoE and the RSF of a video $v$. It is hence conceivable to tag each video with such a compact representation of its QoE characteristic that can then be used by RM and VAC algorithms, as discussed in the next section.

### 4.4.3 SSIM-based RM and VAC Algorithms

In this section we investigate how the QoE characterization of a video sequence can be used to optimally allocate transmission resources to different video sessions and to decide whether or not a new video shall be admitted into the system.

We consider a framework where different video clips are multiplexed into a shared link of capacity $R$ by a control unit that performs RM and VAC. More specifically, the RM module detects changes of the link capacity (e.g., due to concurrent data flows or fading phenomena in wireless channels) and triggers an optimization procedure that adapts the video

rates to maximize a certain QoE-related utility function. Similarly, the VAC module determines whether or not a new video request can be accepted without decreasing the QoE of any videos below a threshold $F^*$ negotiated, for instance, between operator and video consumers. To this end, the VAC invokes the RM module to get the best resource allocation policy for all the videos potentially admitted into the system and, then, computes the SSIM of each video through Eq. (4.3) and checks whether it is above the aforementioned quality threshold. If not, the last video admission request is refused, otherwise the new video is accepted into the system, transmission resources are reallocated as determined by the RM module, and video sources are required to adapt their source rate to such a new allocation.

To alleviate the RM from the burden of computational costs when dealing with a high number of active videos in the channel, we further consider a clustering approach where videos are grouped based on their SSIM vs. rate similarity. Hence, we partition the videos in $K$ classes, depending on the value of $\rho$ for which the SSIM crosses the threshold $F^*$. Each class is then associated to a reference $F$ curve, expressed as in Eq. (4.3), with polynomial coefficients equal to the mean of the coefficients of the videos in that class:

$$F_k^{\mathcal{C}}(\rho) = 1 + \sum_{i=1}^{4} a_{k,i}^{\mathcal{C}} \rho^k \,, \quad \text{with} \quad a_{k,i}^{\mathcal{C}} = \frac{\sum_{v \in \mathcal{C}_k} a_{v,i}}{||\mathcal{C}_k||} \,,$$

where $\mathcal{C}_k$ denotes the set of videos in the $k$th class, and $||\mathcal{C}_k||$ is the cardinality of the class. Class-based QoE functions $F^{\mathcal{C}}$ can be used by the RM and VAC algorithms in place of the actual $F$ of each video, thus reducing the computational complexity at the cost of suboptimal resource allocation and possible violation of the SSIM constraints, due to the coarse approximation of the QoE characteristic of the videos.

### 4.4.4 Optimal resource allocation problem

Let $f_i$ denote the SSIM function associated to a video $i$, which can correspond to either $F_i^{\mathcal{C}}$ or $F_i$, depending on whether or not the class-based solution is considered. Furthermore, let $R$ denote the transmission capacity that needs to be allotted to the videos, and by $\Gamma = \{\gamma_i\}$ an allocation vector that assigns to the $i$th video a fraction $\gamma_i$ of $R$, with $\gamma_i = 0$ indicating that the video is not accepted into the system. Although the H.264 encoding can only offer a discrete set of transmit rates (see Fig. 4.6), in the formulation of the optimization problem we assume that video rates can change in a continuous manner. Under this assumption, the

RSF of the $i$th video can be expressed as

$$\tilde{\rho}_i = \log\left(\frac{\gamma_i R}{r_i(1)}\right) . \tag{4.4}$$

The optimization problem addressed by the RM module can then be defined as follows:

$$\begin{aligned} \Gamma_{\text{opt}} &= \arg\max_{\Gamma} U(\Gamma, R, \{f_i\}) \\ \text{s.t.} \quad & \sum_i \gamma_i \leq 1 \end{aligned} \tag{4.5}$$

where $U(\cdot)$ denotes the *utility function* considered by the optimization algorithm.

We consider two utility functions that reflect different optimization purposes:

**Rate Fairness (RF)**

Resources are distributed to all active videos proportionally to their full quality rate, without considering the impact on the perceived QoE. In this case,

the optimal rate allocation for the $i$th video is simply given by

$$\gamma_{\text{opt},i} = \frac{r_i(1)}{\sum_j r_j(1)} \tag{4.6}$$

so that the RSF of each video equals $\tilde{\rho} = \log(R/\sum_j r_j(1))$.

**SSIM Fairness (SF)**

Resources are allocated according to a max-min fairness criterion with respect to the SSIM of the different videos:

$$U(\Gamma, R, \{f_i\}) = \min_i f_i(\rho_i) . \tag{4.7}$$

Note that, under the assumption of continuous rate adaptation, the SF criterion yields the same SSIM, say $\sigma$, to all active videos. Given this target SSIM, the RSF for each video can be easily found as $\tilde{\rho}_i = f_i^{-1}(\sigma)$, where $f_i^{-1}$ is the inverse of the QoE monotonic function $f_i$. Therefore, the optimization problem can be easily solved by searching for the maximum $\sigma$ that satisfies the rate constraint in Eq. (4.5), i.e., such that

$$\frac{1}{R}\sum_i r_i(1) 10^{f_i^{-1}(\sigma)} \leq 1 . \tag{4.8}$$

Since the left-hand side expression is monotonic, the maximum $\sigma$ can be easily and quickly found via numerical method (e.g., binary search). The optimal resource allocation is then given by

$$\gamma_{opt,i} = \frac{r_i(1)}{R} 10^{f_i^{-1}(\sigma)} .$$

### 4.4.5 RM and VAC algorithms

Based on these utilization functions, we can define three possible RM algorithms, namely:

**RF,** based on Eq. (4.6);

**SFE** based on Eq. (4.7) with exact (E) QoE characterization. i.e., $f_i = F_i$;

**SFC** based on Eq. (4.7) with class-based (C) QoE characterization., i.e., $f_i = F_i^{\mathcal{C}}$.

Given the channel capacity $R$ and the set of videos to be allocated, each tagged with the polynomial coefficients associated to $\{f_i\}$ and the available encoding rates $\{r_i(c)\}$, the RM algorithm finds the optimal allocation $\Gamma_{\text{opt}}$ under the continuous rate assumption and, then, looks for a feasible rate allocation at minimum distance from $\Gamma_{\text{opt}}$ that satisfies Eq. (4.5).

As mentioned before, the VAC algorithm can be built upon any such RMs. The VAC, in fact, will simply call the RM to perform its computations on the set of active videos.

### 4.4.6 Related Work

Prior works on video classification mainly focus on extracting objective networking and quality metrics. In [76] the authors classify videos based on selected common spatial-temporal audio and visual features described by the MPEG-7 compliant content descriptors. Due to the complexity of the method, the authors make use of the principal component analysis (PCA) to reduce the set of features under study. Nevertheless, this work is strictly dependent on the MPEG-7 multimedia format. Scene detection mechanisms were developed in recent years based on predictive analytical models. In [77], the authors propose a scene-change detector for video-conference traces that works based on the average number of bits generated during the scenes, and it modeled with a two-state Markov chain. The proposed low complexity method comes at the cost of requiring full knowledge of the type of video to properly set the thresholds for the scene recognition.

Further related works focus on quality prediction models to capture the behavior of video scenes. In [13], an objective model to predict the quality of the lost frames for 3D videos is designed based on the header information of the video packets at different ISO/OSI layers. This model is able to roughly capture the SSIM of some video clips based on the size of the lost frames and via deep packet inspection (DPI), which is usually avoided by operators in cellular deployments due to the complexity and national privacy rules. Nevertheless, in [78], the authors claim that the frame loss probability, which is mainly a network metric, provides only a limited insight into the video quality perceived by the user. Moreover, the authors state that the rate distortion curves drawn using the PSNR provide a limited representation of the perceived video quality, thus improved quality metrics to better represents videos are needed.

In our work, we analyze and group video test sequences based on the relation between video compression rate and SSIM. As widely recognized, SSIM index improves traditional objective QoS metrics like PSNR and MSE, which have been proven to be inconsistent with the human eye perception. Although the SSIM characterization of a video sequence is computationally expensive, we show that it can be compactly represented by means of four polynomial coefficients which can be associated to the video. Tagged videos can then be handled by simple traffic shaping mechanisms in case of network congestion or under-provisioned network resources.

**Table 4.4.** *Video test set*

| Name | Full quality rate [kbit/s] | Duration [s] | Class # |
|---|---|---|---|
| 3inrow | 11856 | 12 | 1 |
| 5row1 | 11135 | 12 | 1 |
| Akiyo | 5387 | 10 | 2 |
| Boblec | 11504 | 12 | 2 |
| Bowing | 10325 | 10 | 1 |
| Bridge_close | 18246 | 66 | 4 |
| Bridge_far | 18304 | 70 | 4 |
| Vtc1nw | 11210 | 12 | 1 |
| Bus | 16954 | 5 | 4 |
| CaesarsPalace | 17001 | 12 | 3 |
| Cheerleaders | 21757 | 12 | 4 |
| City | 14139 | 10 | 3 |
| Coastguard | 16570 | 10 | 4 |
| Container | 12229 | 10 | 3 |
| Crew | 16179 | 10 | 4 |
| FlamingoHilton | 25622 | 12 | 4 |
| Flower | 16335 | 8 | 3 |
| Football | 15806 | 3 | 4 |
| Football_ext | 18092 | 12 | 4 |
| Foreman | 14642 | 10 | 3 |
| Hall_Monitor | 16291 | 10 | 4 |
| Harbour | 17929 | 10 | 3 |
| Highway | 17529 | 66 | 4 |
| Husky | 24065 | 8 | 4 |
| Ice | 9517 | 8 | 2 |
| Sign_Irene | 14091 | 18 | 3 |
| Washdc | 12948 | 12 | 2 |
| Mobile | 19172 | 10 | 3 |
| Mother_Daughter | 11348 | 10 | 2 |
| News | 7824 | 10 | 2 |
| Pamphlet | 10917 | 10 | 1 |
| Paris | 12450 | 35 | 2 |
| Redflower | 14168 | 12 | 2 |
| Silent | 11586 | 10 | 3 |
| Soccer | 14063 | 10 | 4 |
| Stefan | 17589 | 3 | 3 |
| Tempete | 17850 | 8 | 3 |
| Waterfall | 14950 | 8 | 3 |

Chapter 5

# Simulation and Experimental Results

In this chapter we present the simulation and experimental results of our video transport mechanisms designed in Chap. 3 and implemented in Chap. 4. Thus, in Sec. 5.1 the simulation results for our robust opportunistic scheduling are presented, whereas Sec. 5.2 shows simulation and experimental results from a real testbed of our MCDN framework, in Sec. 5.3, the simulation results of the E2E video path selection mechanism are shown and in Sec. 5.4 we report the simulation results for our QoE-based resource allocation schemes.

## 5.1 Robust Opportunistic Broadcast/Multicast Mechanisms

### 5.1.1 Simulation Setup

We encode two yuv video sequences in CIF resolution (352x288), *Foreman_cif* and *News_cif* [72], using the JSVM encoding software [75], at 30 frames per second with GoP equal to 16. We use only one I and 15 P frames, where each I frame starts a new GoP. For the sake of simplicity, we use only the SNR scalability offered by H.264/SVC, thus we encode the video sequences into 3 video layers, i.e., one base quality layer and two enhancement layers. The quantization points for each video layer, the corresponding cumulative encoding rates and the ideal PSNR achieved when correctly receiving each video layer are shown in Table 5.1 for both video sequences.

We consider a broadcast scenario where the BS serves $N = 100$ users distributed in the corresponding area of service.

We assume a simple distance-dependent path loss model, where the channel conditions

**Table 5.1.** *Quantization points, rates and ideal PSNR.*

| Layer | Q Point | Rate (Kbps) | PSNR (dB) |
|---|---|---|---|
| *News_cif* | | | |
| Base | 42 | 90.4 | 31.5992 |
| Enhancement 1 | 32 | 291.9 | 37.4076 |
| Enhancement 2 | 22 | 846.8 | 43.7088 |
| *Foreman_cif* | | | |
| Base | 42 | 117.1 | 29.9432 |
| Enhancement 1 | 402.5 | 402.5 | 34.7884 |
| Enhancement 2 | 1506.3 | 1506.3 | 40.7380 |

of each user depend on the distance between the BS and the user. In this way we can straightforwardly evaluate the coverage area of each MCS, thus we can design as many regions as the cardinality of the set of MCSs available at the BS for the packet transmissions. The most robust MCS, i.e., $MCS_1$, covers the whole area of service of the BS, while the second MCS, $MCS_2$, covers an area which is smaller but included in the previous area. The higher the MCS, the smaller the area covered by the transmission scheme, thus the smaller, on average, the number of users being served by such MCS. In our simulations we further set at the base station the carrier frequency to 2 GHz, the transmission bandwidth to 10 MHz, the transmission power to 46 dBm, the penetration loss to 20 dB, the noise figure to 9 dB and the thermal noise density to -174 dBm/Hz. If we use the slowest MCS, i.e., QPSK 1/8, the communication area is up to $\sim$1 Km from the base station.

A first assignment of the MCS to be used to send each video layer is performed a priori, based on the average user distribution (baseline solution), by an optimizer,[1] given the total channel constraint. The optimization procedure works at a coarser granularity with respect to the time domain when compared to the granularity required to opportunistically tune the scheduling mechanism (instantaneous). This is mainly due to the computational time of the heuristic algorithm.

Before discussing the simulation results, we shortly describe two tunable parameters in our simulator. We define a first metric to take into account the user mobility, called *mobility*, which is the measure of how quickly a user changes MCS serving area. This translates into

[1]The interested reader can find the details of such optimizer in [27, 35]

an index spanning the interval $[0, 1]$, where $0$ and $1$ indicate a static and highly dynamic scenario, respectively. Hence, if the *mobility* index is set to 0, no user will jump into a neighboring MCS serving area in the time slot, on the contrary users will change MCS area in each time slot when the index is set to 1. We define another user mobility-related metric, the *group size*, as the user correlation when switching from the current to a neighboring MCS serving area (i.e., the number of users changing MSC area in the same time slot). Users in the cell are equally divided in groups of the same size once this parameter has been set.

### 5.1.2 Discussion

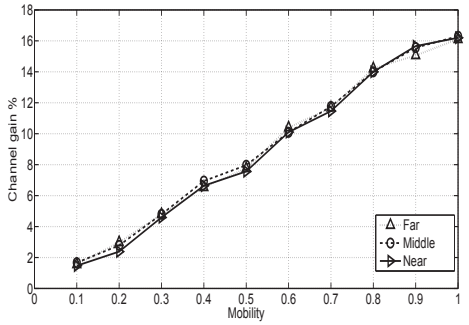
We now investigate the impact of *mobility* and *group size* of users on the wireless channel gain, i.e., the ratio of channel rate (ksymb/s) saved by our algorithm compared to the baseline solution. Our experiments are averaged over more than 1000 runs and are run on a range of scenarios, as presented in Table 5.2. In the *far* scenario most users are far from the base station, thus they can successfully decode packets only if sent with low order MCSs. In the *middle* scenario most users are placed in the middle region while in the *near* scenario most users are close to the base station, experience good SNR levels and thus can decode packets sent with high order MCSs. The cumulative distribution of users with respect to the MCSs is reported in Table 5.2.

**Table 5.2.** *Static user distribution for each scenario.*

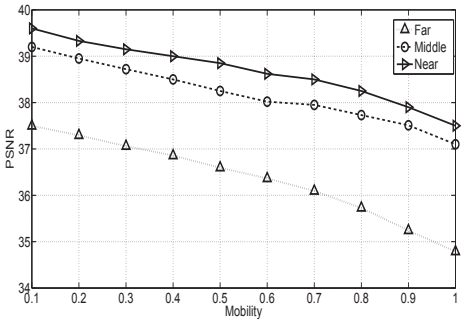
| MCS | Far | Middle | Near |
|---|---|---|---|
| 1 | 100 | 100 | 100 |
| 2 | 60 | 95 | 95 |
| 3 | 35 | 85 | 90 |
| 4 | 20 | 50 | 80 |
| 5 | 10 | 15 | 65 |
| 6 | 5 | 5 | 40 |

In our work we performed simulations using these 3 scenarios and the 2 aforementioned yuv sequences, but due to the weak dependency of the results (quality-wise) from the scenario and the video in use, we now continue our discussion focusing only on the *far* scenario using the *Foreman_cif* video sequence.

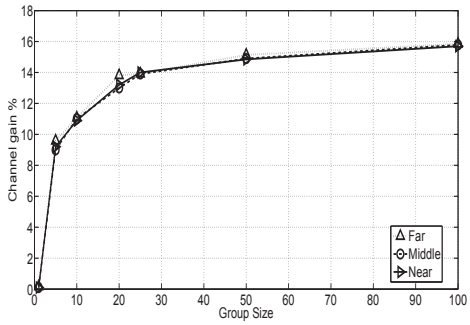
In Fig. 5.1(a) we plot how the normalized channel gain varies, i.e., the percentage of



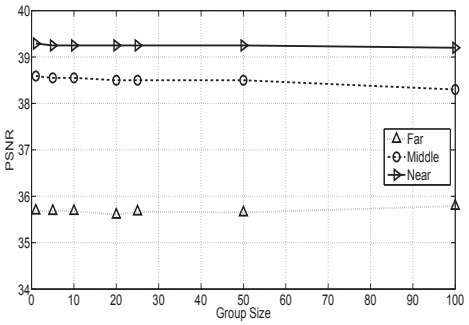

(a) Normalized channel gain, *group size* set to 20.




(b) QoS for the 3 scenarios, *group size* set to 20.




(c) Normalized channel gain, *mobility* set to 0.8.




(d) QoS for the 3 scenarios, *mobility* set to 0.8.

**Figure 5.1.** *Impact of* mobility *and* group size *on the channel gain and on the average video quality perceived (PSNR) for far,* middle *and* near *scenarios.*

wireless resources saved by our proposed broadcast scheduling mechanism compared to the baseline solution, with respect to the *mobility* in the cell, while keeping the average overall QoS of the mobile users as close as possible to the quality level guaranteed by the baseline approach, see Fig. 5.1(b).

The PSNR degrades with *mobility* due to the higher probability of moving to different MCS serving areas and of not being able to receive enough video packets from the lower video layers (which also makes any higher-layer packet correctly received useless).

Further conclusions can be drawn from Fig. 5.1(c), where we fix the *mobility* index in the cell to 0.8, i.e., highly dynamic scenario, and we let users change the MCS serving area either individually or jointly (i.e., within a group) in a certain time slot. Looking at the extreme case of group size equal to 1, i.e., individual changes, the channel gain is negligible. This is due to the heterogeneity of the network, since the selection of the highest possible MCS to

guarantee the target average of users being able to decode a given video layer cannot track each single user's behavior. In case users jointly change MCS area, chances of selecting a higher MCS to serve the video layers and of letting users benefit from it are enhanced. As a result, the higher the homogeneity of users behavior in the cell, the larger the set $\mathcal{U}$, and the higher the chances of saving wireless resources.

The above discussion and conclusions focused on the simulation results for the *far* scenario, but equally apply to the *middle* and *near* scenarios as well: the channel gain increases with the increase of the *mobility* and of the *group size* as shown by Figs. 5.1(a) and 5.1(c), respectively. The only noticeable difference is that the target QoS for both *near* and *middle* scenarios is a few dBs higher compared to the *far* scenario, due to the larger number of users in the proximity of the base station (see Table 5.2).

## 5.2 Mobile Content Delivery Networks and Video Popularity

We first examine the impact of the DASH streaming in terms of traffic load on the network by measuring the amount of data traversing a MAR when a video is requested by a mobile user in a real scenario. Next, we use such measurements to build a large scale network scenario in which we simulate the behavior of the system for a range of network settings where we vary the level of occupancy of the caches and the mobility of the users.

### 5.2.1 Video streaming measurements

A DASH video file consists of the Media Presentation Description (MPD) file and the set of segments building the media content. The MPD collects the information that characterizes the video file and the segment locations. The MPD may contain multiple *representations* for the same media, that is, multiple versions with different resolutions and bitrates. A DASH client is able to dynamically select a representation which better fits, for instance, the network conditions.

In our experiments, we measured the traffic generated by streaming a DASH video from a server to a client through an IPv6 infrastructure that build our testbed. We employed the 597 seconds long Big Buck Bunny DASH video[2] segmented into a set of chunk sizes, i.e., 1, 2, 4, 6, 10 and 15 seconds, and a single representation with average bitrate of 800 Kbps. In

[2] `http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/`

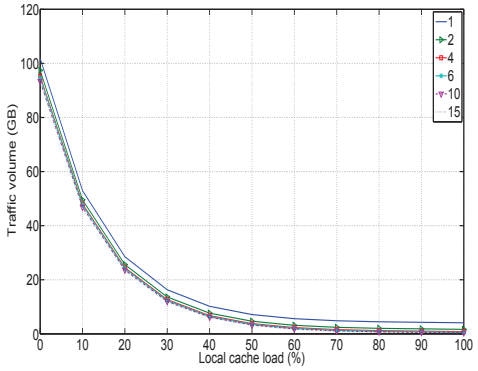
**Table 5.3.** *Experimental measurements*

| DASH Video Streaming Statistics | | | | | | | |
|---|---|---|---|---|---|---|---|
| Chunk length [s] | | 1 | 2 | 4 | 6 | 10 | 15 |
| No. of chunks | | 597 | 299 | 150 | 100 | 60 | 40 |
| Video size [B] | Chunk sum | 60 254 052 | 58 987 593 | 58 344 196 | 58 138 660 | 57 509 612 | 57 373 964 |
| | MPD file | 51 958 | 26 586 | 13 921 | 9 671 | 6 541 | 4 739 |
| | MP4 file | 865 | 865 | 865 | 865 | 867 | 867 |
| | Total | 60 306 875 | 59 015 044 | 58 358 982 | 58 149 196 | 57 517 020 | 57 379 570 |
| Data TX [B] | no tunnel | 64 699 689 | 62 262 514 | 61 251 667 | 60 903 590 | 60 096 154 | 59 873 730 |
| | per chunk | 108 375 | 208 236 | 408 344 | 609 036 | 1 001 603 | 1 496 843 |
| | with tunnel | 69 464 550 | 67 967 699 | 66 837 895 | 66 487 790 | 65 679 527 | 65 454 149 |
| | per chunk | 116 356 | 227 317 | 445 586 | 664 888 | 1 094 659 | 1 636 356 |
| Overhead [%] | no tunnel | 7.28 | 5.50 | 4.96 | 4.74 | 4.48 | 4.35 |
| | with tunnel | 15.19 | 15.17 | 14.53 | 14.34 | 14.19 | 14.07 |
| CDN Node - DM interface Statistics | | | | | | | |
| Control packets TX | | 5 955 | 2 990 | 1 500 | 1000 | 600 | 400 |
| per chunk | | 9.97 | 10 | 10 | 10 | 10 | 10 |
| Control data TX [B] | | 570 465 | 286 300 | 143 575 | 95 680 | 57 583 | 38 388 |
| per chunk | | 955,55 | 957,53 | 957,17 | 956,80 | 959,72 | 959,70 |
| Overhead [%] | | 0.88 | 0.46 | 0.23 | 0.16 | 0.10 | 0.06 |

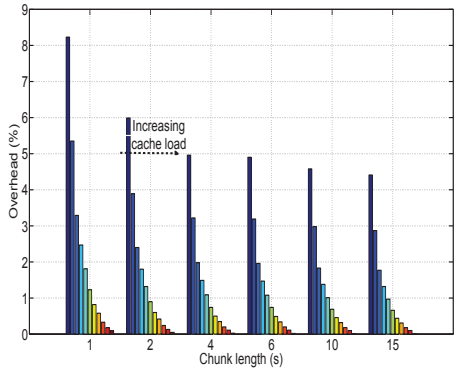
this setup, the DASH video is an MPEG-4 file, thus the MPD contains the pointers to the initialization file, namely the "MP4 file", and to the video chunks locations. The streaming was captured at an intermediate node, acting as a MAR. The values in Table 5.3 summarize the results obtained from the captures.

We performed the measurements for two scenarios: in the first one, called *no tunnel*, the video is forwarded by the MAR working as a plain IPv6 router, that is, without encapsulating the packets. In the second scenario, called *with tunnel*, the packets are forwarded through a tunnel before being delivered to the MN, thus we simulate the encapsulation that would occur in case of mobility[3]. We computed the overhead as the ratio between the extra bytes transferred due to the streaming session and the total video size (payload). As expected, the shorter the chunks, the higher the overhead, because the video requires more HTTP and TCP sessions to be downloaded. Thus, the streaming requires a 7.28% overhead for the 1 second chunks video, and 4.38% for the 15 seconds version. This behavior is exacerbated by the presence of a mobility tunnel: from Table 5.3 we can infer that the encapsulation boosts

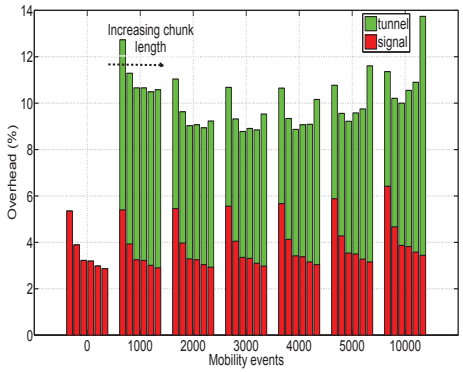
[3]The IPv6-in-IPv6 encapsulation requires 40 bytes for the extra IPv6 header. We recall that with current mobility protocols, like Proxy Mobile IPv6, the packets are always encapsulated, while in our DMM architecture, the packets are tunneled during the handover event until the next chunk is requested.




(a) Total traffic volume vs. cache load.




(b) Signalling overhead for static scenarios in the core network.




(c) Signaling and tunneling overhead in the core network with 10% cache load.

**Figure 5.2.** *Traffic volume and overhead generated for static and mobile scenarios.*

the overhead, and, conversely to the previous observation, it penalizes more the longest chunks.

The lowest part of Table 5.3 describes the amount of control data transferred in the interaction between the CDN node and the DM. Such interaction serves to provide the best chunk source location, and it costs less than 1 KB for each chunk.

### 5.2.2 Discussion

Fig. 5.2(a) shows the in- and outbound traffic volume at the joint entity CDN-MAR node as a function of the percentage of chunks stored in the local cache. The general shape of the curve is a direct consequence of the video popularity distribution introduced earlier. As

expected, as more content is cached in the CDN node, the less traffic is transported through the core network, and more traffic is directly served from the CDN node to the user. If the cache of the CDN node is empty (left side of Fig. 5.2(a)), all data needs to be retrieved from the main video server (traffic from/to main server and from/to MN). Providing just a few of the most popular chunks in the CDN node can already significantly reduce the traffic load in the core network. Providing further contents with low popularity in the CDN node will still reduce the traffic volume in the core network, but at a lower rate. In the extreme case where all contents are cached locally (right side), all requests can be served from the CDN node and the traffic is only between the CDN node and the MN (i.e., mainly on the only access network side). Note that the CDN node cannot reduce the traffic volume on the path from the MAR to the MN. Thus, the traffic load observed at $100\%$ cache load is exactly the share of the traffic volume on the access network, whereas the delta observed for lower cache loads is the traffic volume in the core network. This shows that $65\%$ of the overall network traffic (i.e., core and access network) can be saved by providing all data from the local CDN nodes.However, the higher the storage capacity of the CDN, the more the associated costs will be. Storage costs approximately increase on a linear scale, whereas the savings in traffic volume follow an exponential shape. The network architects goal is to find the optimal tradeoff between the two.

We can further observe that the shorter the chunk length, the larger the generated overhead traffic. This overhead is caused by *A)* the packet overhead, *B)* the HTTP session setup for each chunk, and *C)* the messages querying the DM for the optimal location of the requested chunk. Overhead *A+B* is smaller for larger chunks as fewer packets are necessary to handle the network protocol operation at all layers of the IP stack.

Finally, overhead *C* obviously increases with *i)* shorter (i.e., more) chunks and *ii)* smaller cache sizes, i.e., less hits for cached content. Chunks of 1 s length will result in 15 times the number of "*get_optimal_location*" requests to the DM than chunks containing 15 s of video data.

As a consequence, we can argue that longer chunks can reduce the traffic volume in the network. This fact is in line with the observations driven by the values of Table 5.3 and it is also shown in Fig. 5.2(b), which depicts the share of the measured overhead in the core network for different chunk lengths when varying the cache load from $0$ to $100\%$ in steps of

$10\%$. In addition, if all chunks are cached in the local CDN nodes (right-most bar of each group), nearly no overhead is measured in the core network, as only requests for the MPD and MP4 files pass the CDN nodes.

In the following set of simulations we add the user mobility into the system. We consider the range from $0$ to $10K$ mobility events during the simulated time period, i.e., on average one mobility event per played video. Note that a mobility event is a change in the IP point of attachment, caused by handovers resulting in a change of the serving MAR. If a mobility event is triggered, the DMM will forward the currently streamed chunk to the new MAR to avoid the break of the ongoing HTTP session. This causes additional "tunneling overhead" (D), as the packets of the ongoing chunk will be transported along additional network paths to the new MAR, and thus they are counted as traffic at both MARs. The following chunk is then directly requested from the new MAR and thus it does not need to be transferred through the DMM tunnels.

Fig. 5.2(c) shows the percentage of overhead for different numbers of mobility events and different chunk lengths. The group of bars at the left (no mobility) depicts the same scenario and values as in Fig. 5.2(b) in the second bar of each group, i.e. for a cache load of 10%. We distinguish between signaling overhead (A+B+C) and tunneling overhead (D), as they show an opposite behavior. The signaling overhead gets smaller for larger chunk sizes as fewer chunks are requested per video, whereas the tunneling overhead increases with larger chunks due to a larger chunk being forwarded. In the static scenario (no mobility), there is no tunneling overhead. Then, for increasing user mobility, we can see that the additional overhead due to the tunneling of the ongoing chunk to the new MAR overcomes the reduction of signaling overhead when large chunks are used. In addition, for short chunks, the lookup tables in the CDN nodes must hold more entries and the data exchanged between CDN nodes and the DM to synchronize the popularity databases. Moreover, more lookup requests per time must be processed, requiring more processing power and thus expensive hardware.

As a final observation, in Fig. 5.2(c), we note that a chunk length of $4$ seconds represents a good trade-off in our system between signaling and tunneling overhead, outperforming the other chunk sizes in most scenarios. Moreover, a $4$ seconds chunk length offers a good granularity for the video player to promptly adjust the video quality.

## 5.3 Path Selection

The aim to maximize a utility function on the "normalized" $M$-tuples representing all the possible paths (where each of the $M$ values in a tuple refers to a given network metric) is now further investigated from a real deployment point of view. To do so, we investigate the feasibility of the algorithm in an LTE-compliant network implemented in ns-3 and we eventually add traffic engineering mechanisms to support the video path selection in congested scenarios.

In Section 4.3, two criteria for the selection of the video paths were investigated, reflecting the operator's and the user's points of view. Due to the preferable performance achieved by the *max-sum* criterion, we select the utility function maximizing the sum of the proximity values associated to the metrics in the $M$-tuples. Thanks to LTE core and access networks features implemented in ns-3, we can extend the set of network metrics already considered for the sake of analysis of the number of caches to be deployed in the network in Section 4.3.2. In addition to the storage occupancy of a CDN video server, the routing distance from the server to the points of access, and the channel quality of the wireless link, we also take into account the available data rate on the link between the video source and the P-GW (thus, taking into account concurrent sessions) and the length of the queue of packets at the access points, which is a further useful indicator of the network conditions, i.e., the longer the queue, the higher the probability of congestion on the access side. The storage occupancy of the CDNs is computed as the fraction of the memory in use. The routing distance is expressed as the number of hops between the CDN cache and the end user. The data rate on the link between the CDN video source and the P-GW is also taken into account, since once a CDN is selected for transmission, the available bit-rate on the link should count for the corresponding bandwidth consumption due to concurrent sessions, previously established by other nodes, on the same link. The values taken by these core network metrics can be communicated in the network via ALTO protocol extensions enabling joint transmission of multiple metric values as proposed in [52]. Furthermore, we consider the SNR of both cellular and WiFi access as a quality measure of the wireless link, observed by the mobile terminal from the wireless interface, complemented by the measure of the queue size at the access points as mentioned before.

The goal is to investigate the impact of our video delivery framework on throughput and

packet delivery delay for video streaming services. This way, we provide the guidelines for fine-tuning the network metrics involved and for steering the video path to best meet the requirements of the end users and of the service providers.

In the rest of the section our algorithm will be referred to as Path Selection (PS) and will be investigated both as a stand-alone mechanism and jointly with the Packet Dropping with Threshold mechanism (PDT), described as follows.

Existing traffic engineering mechanisms are enabled in the mobile network when congestion occurs. Based on the video specifications, operators might decide to reduce the data rate from the source, in the core network nodes or at the access points. The selection of the video path for a set of end users might cause the temporary congestion of some preferred links/nodes, leading to a degradation of the network performance and of the quality perceived by the user. We thus shape the video traffic at both the eNodeB and the WiFi spot by dropping the video packets before they are queued in the buffers of the access points. We notice that the congestion lasts until a new available video path is selected. Since the procedure could involve several links and nodes throughout the mobile network, it is assumed to be solved after a considerable number of packet transmissions. The mechanism of packet dropping is implemented as follows. When the end-to-end delay of the packet received by a given end user exceeds a fixed threshold (delay constraint), a mechanism in the corresponding access point is triggered such that the incoming packets from the core network and meant for any other node are dropped before being queued in the buffer(s) of the access points. This makes room for packets to be sent to the node observing higher delivery delays. Once the measured packet delay falls below the threshold, the packet dropping procedure is disabled (otherwise a timer is triggered when the congestion occurs and expires after a fixed time interval).

As a remark, we point out that while PS works for both UDP and TCP video traffic, PDT should be restricted only to the UDP flows supported by H.264/SVC codecs to make it possible to drop video frames at the data link layer. In TCP connections, video packets cannot be dropped as described above, thus the operator would need a proxy in the core nodes, e.g., the P-GW, allowing the dynamic resizing of the TCP window (adaptive streaming). However, at the time of writing, dynamic adaptive streaming over HTTP (DASH) is not yet supported in ns-3. For the sake of simplicity, we focus our study on a network perfor-

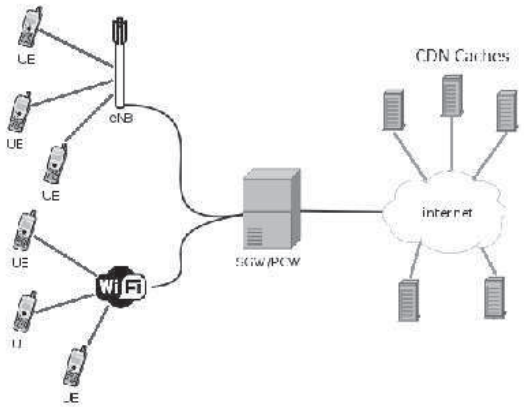



**Figure 5.3.** *Reference architecture for the simulations.*

mance level, from which we derive the impact of the streaming service on the video quality at the user side. Readers interested in more accurate models, for instance, with focus on the receiver side, are referred to, e.g., [79].

### 5.3.1 Simulation Setup

We implemented the PS and PDT algorithms in the LTE module in ns-3. In our reference scenario, Fig. 5.3, we deploy from 5 to 10 CDN sources connected to the P-GW in the core network, as this number of CDN nodes is already acceptable in our previous analysis and at the same time feasibly implementable in ns-3, and we deploy 6 mobile users with different mobility patterns in the wireless access network. The LTE access network is implemented as follows. We consider an LTE macro-cell with bandwidth ranging from 5 to 20 MHz (i.e., number of resource blocks between 25 and 100) and a coverage radius of up to 10 km. The transmission power of the eNodeB is set to 30 dBm, the noise figure to 5 dB, and we use the Friis' free space propagation model. The buffers dedicated to each user to be served by the eNodeB have limited size set to $2^{21}$ Bytes. We implement a WiFi 802.11n spot, with bandwidth set to 20 MHz, at a distance of 3 Km from the eNodeB and with coverage radius of up to 200 m. The transmission power of the access point is set to 16 dBm, the noise figure to 7 dB, and the channel propagation model is the log-distance model. The common buffer to serve all the users attached to the access point is limited to the size of $2^{20}$ Bytes. We implement in the EPC module the CDN caches such that each CDN has its own storage capacity and the link between each CDN and P-GW is set to the same channel rate, but

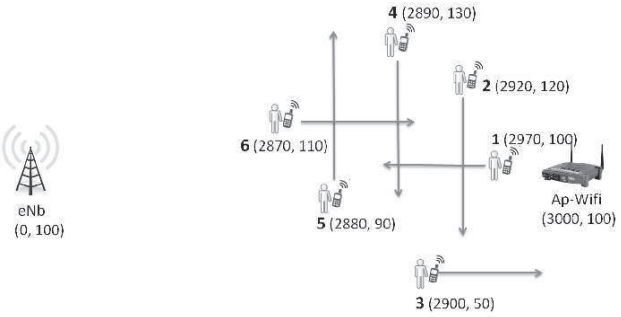



**Figure 5.4.** *Users mobility for the congested scenario.*

to different propagation delays due to the geographic locations. Since our focus is on the performance of a congested network as a whole, all video streaming sources, for the sake of simplicity, generate video packets at the same rate of 500 KB/s. The packet size is set to 1024 Bytes, while the delays on each CDN-P-GW link is randomly chosen in the interval $[1, 500]$ ms, that includes the delays of the core links, the response time of the cache (load) and the latency of the HDD. We let the delays of the links be slowly time-varying during the simulation time, in order to take into account the dynamics of the caching nodes, the traffic and the core links and nodes between the source and access point (we consider different time granularities of the dynamics between core and wireless sides) in real networks.

We assess the performance of our PS algorithm under congested and generalized scenarios. In the former, as for Fig. 5.4, we place six users in the cell and we let them move following a way point mobility model, making sure that they cross the WiFi area (that becomes congested already with this small number of users and for most of the simulation time) within each run and that they are in the LTE coverage area. The average speed of the users is set to 2 m/s (urban area). In the generalized scenario, users move according to a random way point model with average speed set to 2 m/s. Here, we average the performance of the system over a wide range of network topologies thanks to the random placement and movement of the users at each instance of the ns-3 simulator.

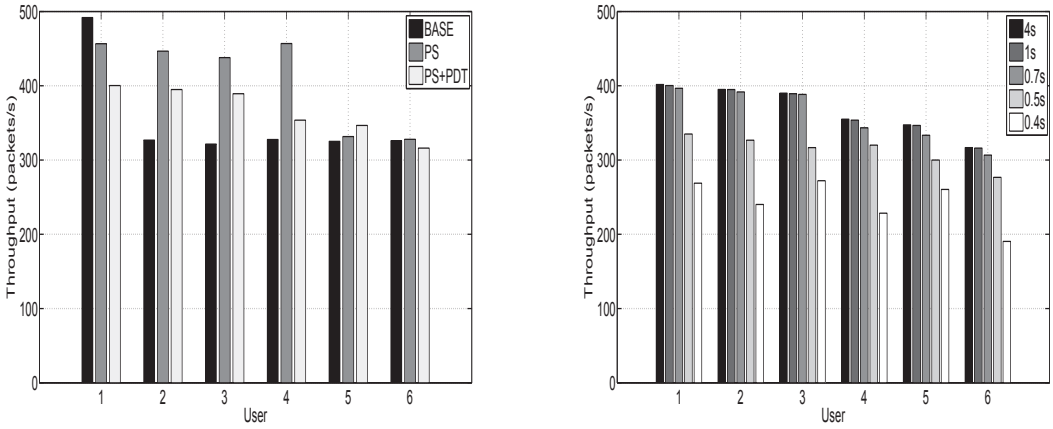
We notice that, once the user is in the range of the WiFi hot spot, an additional access technique is made available, thus increasing the chances of maintaining a good performance of the ongoing video session. In particular, we set a minimum SNR (threshold) as WiFi sensitivity level when connecting to the WiFi access, making sure that the wireless link holds at

the selected user's speed[4]. Moreover, from an operator's point of view, multiple technologies lead to an excess capacity that can be redistributed to the users that are not covered by the WiFi spot. For what concerns the mobility management within our delivery framework, IP session continuity features are not yet implemented in ns-3, thus we handle the user mobility at the application layer to avoid out-of-order decoded packets at the receiver side. Examples of deployed mobility management solutions at the application layer are those based on IP multimedia subsystems (IMS), which use IETF protocols such as the session initiation protocol (SIP) to integrate with the Internet. As a final remark, while PDT runs instantaneously, i.e., as soon as the measure of the packet delay exceeds the threshold, the PS algorithm operates every 2 s, a value that takes into account the signaling propagation and collection of the ALTO messages containing the network metrics.

### 5.3.2 Congested scenario

In this section we discuss the performance of our proposed algorithms for a scenario designed to represent a congested network, i.e., the video packets coming from the CDN video sources exceed in number the buffer size at the access points. We compare the throughput and packet delivery delay of our proposal to those of a baseline solution, named "Base", implemented in the latest release of ns-3. There, a mobile user is connected to either the eNodeB or the WiFi spot and receives the video packets uniquely from the first video source selected at the time of the session initialization. In our simulations we take into account the impact of mobility and load of the system on throughput and packet delivery delay, that are the key metrics to assess the performance of real-time video streaming applications. In the plots, we compare Base, PS and the combination of PS and PDT, namely "PS+PDT", algorithms. This way, we provide a set of network management solutions, which offer different trade-offs in terms of throughput and delay. We argue that the increasing performance of the network as we enable in sequence the PS and the PDT traffic engineering mechanisms comes at the cost of an increasing computational complexity of the system. Depending on the service and users' requirements, the mobile operator might deploy only the PS algorithm instead of the combination of PS and PDT, when a fine-tuning of the parameters involved in the framework can already meet the video service requirements.

[4]The interested reader can find more details in [80].

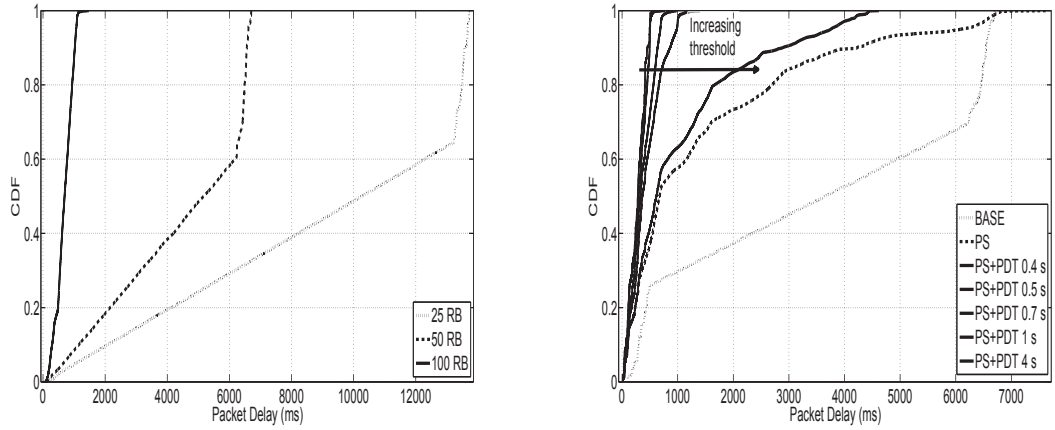



(a) Throughput measured at the nodes for Base, PS and PS+PDT (from left to right in each group), with threshold set to 1s.

(b) Throughput measured at the nodes for PS+PDT, with different thresholds (descending order from left to right in each group).

**Figure 5.5.** *Impact of the Base, PS and PS+PDT algorithms on the throughput for the six mobile users.*

In Fig. 5.5, we compare the throughput measured at the six mobile users when using the Base, PS and the PS+PDT algorithms in a congested scenario. In Fig. 5.5(a), we plot the average throughput of the mobile users setting the threshold of the PDT to 1 s. To make the algorithms comparable, in the baseline case we let 5 users always connect to the eNodeB (and to the same CDN by definition of baseline algorithm) and 1 user to the WiFi access point. This way, the eNodeB is congested since the users cannot modify their initial choice of the best video path (otherwise the video session would be dropped) nor benefit from the packet dropping at the access points. The results show that PS is beneficial in terms of throughput compared to the baseline case, since the nodes can switch from one access point to the other, improving the average quality of the channel, thus the packet delivery rate. The combination of PS and PDT is slightly worse than PS in terms of throughput, as expected, but still acceptable for GBR applications, when setting, for instance, the target rate to $\sim$ 330 KB/s. However, as plotted in Fig. 5.5(b), setting a higher threshold corresponds to an increase of the throughput to values similar to the PS case, even though PDT drops packets from the video streams. When congestion occurs, PDT drops the packets before being queued in the buffers of the access points. The congested nodes, after a short time interval, benefit from such packets dropped in the buffers of the eNodeB or in the common




(a) Cumulative distribution function of the average packet delays measured at the nodes for number of RBs set to 25, 50 and 100, baseline case.

(b) Cumulative distribution function of the average packet delays measured at the nodes for number of RBs set to 50.

**Figure 5.6.** *Impact of the Base, PS and PS+PDT algorithms on the packet delay of the mobile users.*

buffer when considering the WiFi spot. Once the congestion is solved, more packets can flow again through the access points before the end of the video session, set to 30 s. When the threshold is set to around 1 s, the percentage of dropped packets still makes it possible to increase the number of packets being delivered in time to the user before the end of the video session. This turns into a gain in throughput of the nodes for such high thresholds, at the cost of a slight packet delivery delay increase. Setting the threshold below 1 s, the percentage of dropped packets outweighs the ratio of packets "gained" by the system once the congestion is over.

Before discussing the impact of the algorithms on the packet delivery delay, we present in Fig. 5.6(a) the cumulative distribution function (CDF) of the packet delivery delays at the end users when the number of resource blocks (RBs) in use at the eNodeB by the baseline solution is equal to 25, 50 and 100. Hence, we analyze at which bandwidth the eNodeB is congested in the reference scenario. With 100 RBs the network delivers the packets avoiding congestions, while with 50 RBs the eNodeB is congested and half of the packets are delivered with delays above 6 s (and even more with 25 RBs). For the sake of comparison, we target a scenario where the access points, at some point in time during the simulation, are fairly congested, thus we set the number of RBs to 50 in the simulation runs. The WiFi access point, with a bandwidth of 20 MHz and with six users crossing its coverage area, is rarely

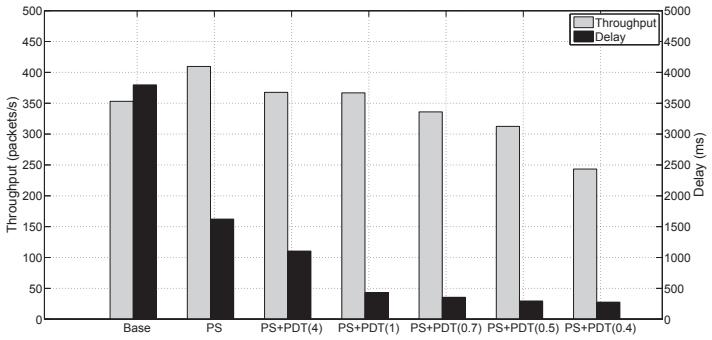
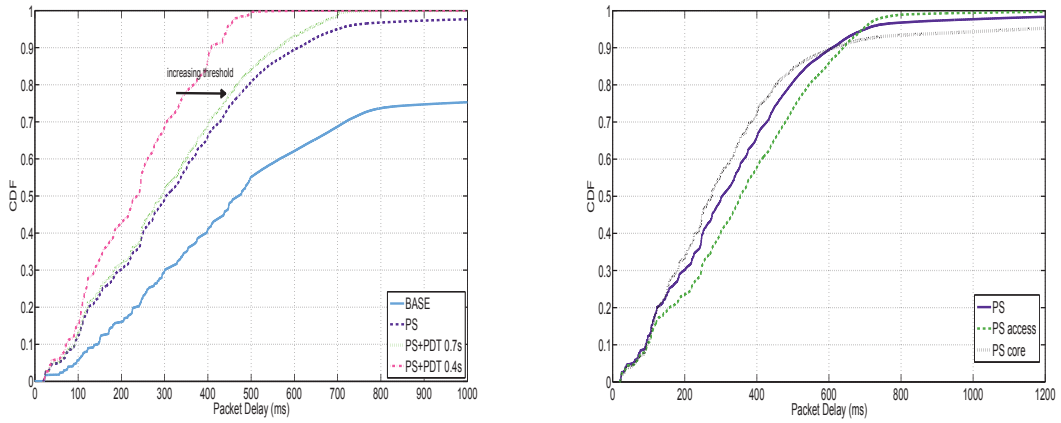



**Figure 5.7.** *Average throughput and delay comparison. In brackets the threshold value for PS+PDT.*

congested, giving the operator a chance to offload the traffic when congested or, from the user's point of view, to re-direct the video path to the next best available technology. In Fig. 5.6(b) we compare Base, PS and PS+PDT by showing the CDF of the packet delivery delay experienced by the users. As expected, in the baseline case the users experience the worst delays due to the pre-selected video path which is kept during the whole video session, thus congested or slow links between the CDN selected and the P-GW affect the performance. For instance, the flexibility offered by PS makes it possible to deliver 90% of the total number of video packets (which ensures a near-optimal quality, assuming a logarithmic relation between data rate and perceived video quality as described in [61]), with a 40% delay reduction compared to the baseline case. Further improvements are brought when PDT is enabled at the access points, leading to remarkable gains in terms of delivery delay, but at the cost of some throughput performance degradation for stringent delay constraints, such as 0.4 and 0.5 s (see Fig. 5.5).

We compare in Fig. 5.7 the average throughput and packet delivery delay experienced by the users when using the different algorithms proposed. It can be observed that PS alone allows to significantly cut the delays, which is crucial for delay stringent streaming applications, compared to the baseline case, without affecting the throughput, which instead benefits from the video path redirection. PDT jointly with PS further cuts the delays in the delivery system while keeping the delivery ratio, unless the threshold is set to very low values, thus making the network unstable. We remark that PDT alone is out of the scope of this work since our main goal is to evaluate the PS algorithm in ns-3 to assess the impact of the algorithm in an LTE-compliant network. Moreover, the target is to improve the




(a) Comparison among PS, PS+PDT and baseline solution.

(b) Comparison when tuning the metrics in the PS algorithm.

**Figure 5.8.** *Cumulative distribution function of the average packet delays.*

responsiveness of the network without significantly affecting the throughput to meet the highly-demanding requirements of the latest emerging mobile services. We recall that it is possible for the operator to fine-tune the weights of the metrics taken into account in the PS algorithm to adapt the performance of throughput and delivery delay to the dynamics of the network and to the targeted performance and available network resources (as it will be shown later).

### 5.3.3 Generalized scenario

Based on the analysis of the results achieved in the congested scenario, we now investigate the performance of our algorithm, compared to the baseline scheme, for a wide range of network settings where congestions might temporarily occur. We report the performance of PS in terms of throughput and packet delivery delay. By averaging over a high number of network topologies, where users are randomly placed and move following a random way point model, we assess the robustness of our algorithms for a range of settings and evaluate the overall performance for long-term video sessions. We start comparing the packet delivery delay of our algorithms to the baseline solution in Fig. 5.8(a). We notice that PS already achieves remarkable results, whereas some slight further improvement is obtained, especially in terms of maximum delivery delay when PDT is used in addition. The throughput

is significantly reduced only for PS+PDT with 400 ms delay constraint (Table 5.4), showing that congestion does not occur very often.

**Table 5.4.** *Delays and throughput for PS, generalized scenario.*

| | Base | PS | PS+PDT 1s | PS+PDT 0.7s | PS+PDT 0.4s |
|---|---|---|---|---|---|
| Throughput (pck/s) | 374 | 462 | 459 | 454 | 344 |

### 5.3.4 Impact of core and access networks

In this section we investigate the impact of core and access network metrics by tuning the weights of the coefficients in the optimization algorithm. Thus, in Fig. 5.8(b) we compare the original PS algorithm, where each metric has the same weight, with PS algorithms where we increase the weight of either the core or the access network metrics. We notice that by giving the core network metrics 10 times more weight with respect to the wireless access metrics reduces the delivery delay for 90% of the packets, while the remaining 10% need more time to reach the destination with respect to the original PS. This is due to the fact that preferring the core to the access can lead to temporary congestion on the wireless channel. On the other side, increasing the weight of the access network metrics makes it possible to quickly complete the individual packet delivery, at the cost of an increased average packet delay, as shown in Table 5.5. The maximum packet delivery delay is heavily reduced when the access is preferred in the optimization algorithm, with a negligible loss in throughput. This result is crucial and shows the limits of core network-based optimization approaches such as those proposed in [30].

**Table 5.5.** *Delays and throughput for PS when tuning the metrics.*

| Algorithm | Max delay (ms) | Average delay (ms) | Throughput (packet/s) |
|---|---|---|---|
| PS | 5668 | 352 | 462 |
| PS core | 5141 | 381 | 477 |
| PS access | 1978 | 365 | 434 |

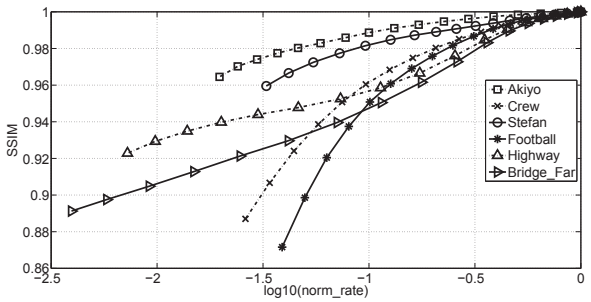



**Figure 5.9.** *SSIM vs. $log_{10}$ of the normalized rates of the selected videos.*

### 5.3.5 Impact on the QoE

In order to evaluate the impact of our path selection strategies on the QoE, we encoded a set of video clips from [72] into H.264-AVC [71] sequences using the JSVM software [75]. The sequences count 300 frames encoded at 30 frames per second with a GoP size of 16. The encoding scheme is such that the I frame starts the corresponding GOP and is followed by P frames. The quantization points of the videos are selected to achieve an application rate of about $500KB/s$. We then encode the same videos at lower bitrates, i.e., with coarser quantization points, and compute the SSIM [81] (already presented in Sec.4.4.1) index values corresponding to each coding rate shown in Fig. 5.9, using the MSU quality measurement tool [82].

**Table 5.6.** *SSIM values for each algorithm, generalized scenario.*

| Videos | Base | PS | PS+PDT 1s | PS+PDT 0.7s | PS+PDT 0.4s |
|---|---|---|---|---|---|
| Akiyo | 0.9986 | 0.9992 | 0.9992 | 0.9992 | 0.9983 |
| Crew | 0.9780 | 0.9823 | 0.9822 | 0.9820 | 0.9761 |
| Stefan | 0.9801 | 0.9842 | 0.9841 | 0.9839 | 0.9782 |
| Football | 0.9703 | 0.9765 | 0.9763 | 0.9760 | 0.9675 |
| Highway | 0.9668 | 0.9717 | 0.9716 | 0.9713 | 0.9649 |
| Bridge_Far | 0.9599 | 0.9658 | 0.9657 | 0.9653 | 0.9576 |

We average the SSIM values computed for each video frame. The range of SSIM values goes from 0 to 1, which represent the case of completely different or identical frames, respectively.

The throughput measured in the generalized scenario for the algorithms under study (Sec. 5.3.3) is mapped to SSIM as shown in Table 5.6, and then to MOS, which assesses

the subjective perceived video quality on a scale of 5 values, from 1 (bad) to 5 (excellent), as in [61]. The differences of the MOS values computed for each video when using the same delivery algorithm are negligible, thus we present a single mean value of the MOS for each algorithm, i.e., $3.42$ (Base), $4.57$ (PS), $4.53$ (PS+PDT, 1s), $4.47$ (PS+PDT, 0.7s) and $2.96$ (PS+PDT, 0.4s). The results show that the PS algorithm already improves the quality perceived by the video consumer from fair (i.e., MOS=3) to good quality perceived (i.e., MOS=4), whereas imposing too tight a threshold to the system (0.4 s) can drop the quality perceived even below the one achieved by the Base algorithm.

For what concerns the network latency, we note that users adopting the Base algorithm may incur in congestion events without being able to recover from them within the simulation time, while PS and PS+PDT algorithms let users recover from congestion but at the cost of experiencing a jitter which decreases the perceived quality as the motion intensity of the video increases. Based on [83], we argue that the impact of jitter on the QoE is mitigated by properly setting the buffering time at the receiver side. Our experimental traces show that PS and PS+PDT can generate a maximum jitter of a few hundred ms which can be compensated at the user side with a buffering time of 2-3 s [83].

## 5.4 QoE-based video transport

We implement in Matlab a video delivery framework in which a shared transmission channel with capacity $R$ is controlled by an access unit that provides i) resource allocation to the different video sources and ii) video admission control functionalities.

To begin with, we compare the performance of the RM algorithms in Section 4.4.3 by assuming that all $V = 38$ videos in our test set are simultaneously active when varying the capacity $R$ of the channel. Fig. 5.10 reports the average SSIM when varying $R$ for the three RM schemes. Note that the channel capacity has been normalized to the maximum aggregate traffic rate generated by the video sources, given by

$$G = \sum_{v=1}^{V} r_v(1)\,. \tag{5.1}$$

The vertical bars represent the standard deviation of the SSIM for the different videos. As expected, SFE and SFC show smaller standard deviation than RF. However, we note that the approximation introduced by the class-based approach is paid in terms of a larger variability

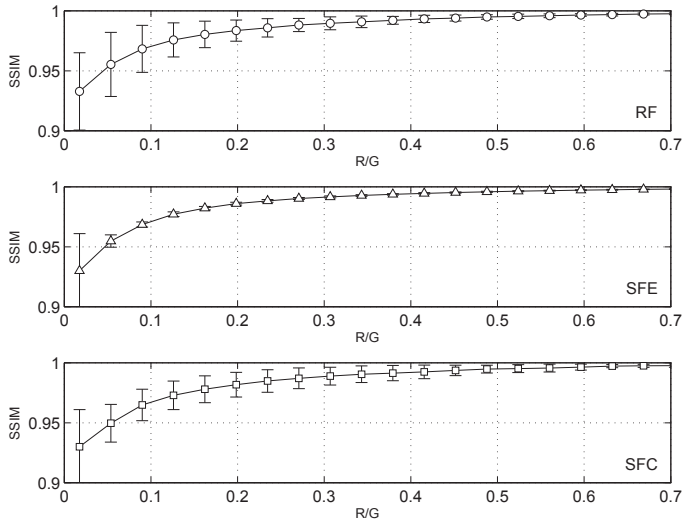
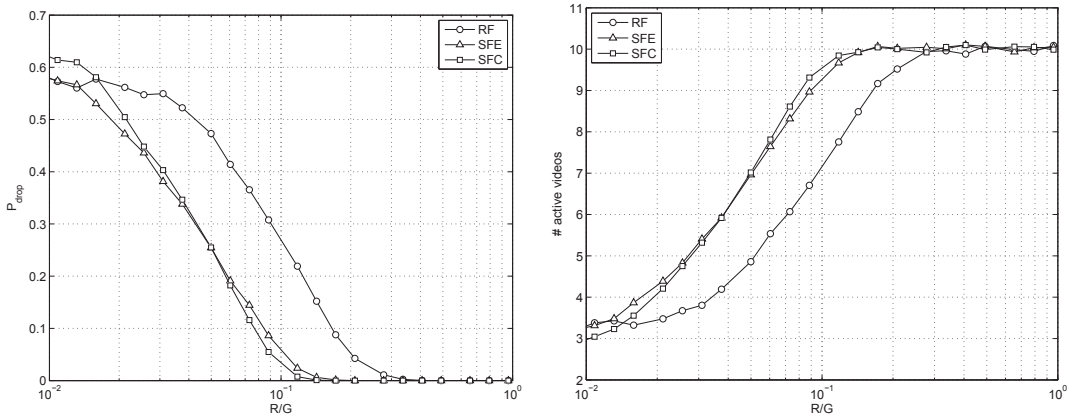



**Figure 5.10.** *Mean and standard deviation of videos' SSIM when varying the normalized channel rate $R/G$, for different RM algorithms.*

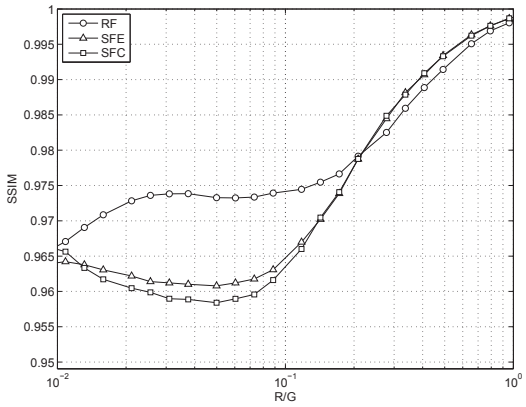
of the SSIM of the active videos compared to the exact per-video approach of SFE. Overall, SSIM-based resource allocation schemes tightly reflect the expected behavior of the system for any given SSIM threshold and can further benefit in terms of computational savings when partitioning videos into classes.

Successively, we test the VAC algorithm with the different RMs. To this end, we simulate a Poisson video request process with $\lambda = 0.66$ requests/s. Each video request refers to a video randomly and uniformly picked among the $38$ videos of our test set, so that the average offered load is $\lambda T = 10$ videos, where $T$ is the average duration of a video, with an aggregate rate request (at full video quality) of about $G = 150$ Mbit/s. At every new video access request, the VAC invokes the RM algorithm to get the optimal rate allocation in case the new video flow gets accepted. Then, the VAC estimates the SSIM for each active video when applying such allocation policy to the system and checks whether the quality of any video drops below the threshold that we set to $F^* = 0.95$, i.e., the minimum SSIM value to reach a MOS value of 4 (good). In this case, the new video access request is rejected and that video flow is *dropped*, i.e., not activated in the network. When an active video session is over, the related channel resources are released and immediately reallocated by the RM to the active video sequences. Note that, in case of SFC, the class-based SSIM approximation is only considered in the VAC and RM algorithm, while simulations are performed considering the

real SSIM characteristics of each video.




(a) Video dropping probability *vs* channel rate. (b) Average number of active videos *vs* channel rate.




(c) Average SSIM of active videos *vs* channel rate.

**Figure 5.11.** *VAC performance with different RM algorithms, when varying the normalized channel transmit rate $R/G$.*

Fig. 5.11(a), Fig. 5.11(b), and Fig. 5.11(c) report video dropping probability $P_{\mathrm{drop}}$, average number of active videos, and average SSIM of the active videos, respectively, for the three RM algorithms when varying the rate $R$ of the transmission channel. We observe that when $R$ is of the same order of magnitude of the offered traffic (rightmost side of the graphs), the three algorithms perform in a similar manner. When progressively decreasing the channel rate, the RF algorithm scales uniformly down the RSF $\rho$ of all active videos, which determines a *non-uniform* decrease of the SSIM of the different videos. In this way, the SSIM of more dynamic videos get close to the SSIM threshold when others are still enjoying much higher SSIM. Considering again the curves in Fig. 4.7, we observe that the most dynamic

videos reach the SSIM threshold $F^* = 0.95$ for values of $\rho$ from $-1.5$ to $-1$, which correspond to a normalized channel rate $R/G$ in the range from $0.02$ to $0.1$. In these conditions, a single high dynamic active video will prevent the entrance of new videos, since any further reduction of the rate allocated to the ongoing video will decrease the SSIM of some of them below the threshold. The average SSIM remains approximately constant and equal to the mean SSIM of the videos in the test set for $\rho \in [-1.5, -1]$. For even smaller channel rates, the most dynamic videos are likely not accepted into the system, and the rate of the others will be proportionally reduced, determining a progressive decrease of the average SSIM.

On the other hand, the SFE and SFC algorithms can admit new videos even at low channel rates (lower dropping probability compared to RF), at the cost of a slight decrease of the quality delivered to the users, though within the constraint of minimum acceptable SSIM. Actually, this constraint cannot be strictly enforced by SFC, whose admission decision is based on the class-based approximation of the SSIM characteristic of a video and, indeed, the average SSIM of active videos is slightly lower for SFC than for SFE. This non-ideal behavior of SFC, however, can be dealt with by slightly increasing $F^*$.

Chapter 6

# Conclusions

In this thesis a set of video delivery mechanisms were designed to optimize the transmission of video applications over next generation cellular networks. We first designed from an architectural point of view our proposed schemes addressing issues related to the core network and the radio access network, and we further detailed them at the different layers of the protocol stack which were taken into account for the optimization. Upon the architectural choices, resource allocation schemes were implemented to support a range of video applications, including video broadcast/multicast streaming, video on demand, real-time streaming, video progressive download and video up-streaming.

We first proposed a robust opportunistic algorithm for scalable video broadcast streaming. From a range of realistic settings, we draw the conclusion that the wireless resource usage can be reduced while keeping the same expected average QoS at the clients, allowing an operator to reuse the saved channel capacity either for other sessions or for enhancing the current streaming (i.e., adding AL-FEC, retransmissions and so on). A possible extension is to integrate this opportunistic scheduler at the base station with the coarse-grained optimizer designed in [27, 35] in order to build a complete delivery framework, from the media server to the end users. Moreover, we considered only the quality scalability feature of H.264/SVC. The spatial and temporal scalability features of such codec would bring different dimensions and more flexibility to our scheduling mechanism, e.g., addressing the problem of heterogeneous mobile devices (smartphones, tablets, etc..). As a further future research avenue, we foresee the challenge of enhancing the proposed scheduling scheme by selecting video packets from a specific queue so that the number of users that can receive a

video quality layer with a given MCS at a certain time instant is maximized. This method enhances the overall video quality of the whole system, by increasing the size of the individual groups of users receiving a specific video layer. This procedure will give a wider range of tunable scheduling opportunities to a mobile operator.

We also designed a MCDN-based video delivery framework which serves the mobile users video requests by taking into account the popularity and proximity to the users of the video contents. The simulation results show that the system can significantly reduce the traffic volume in an operator network. In particular, the larger the cache, the less the data volume transported through the network, but the higher the storage costs. A good trade-off between amount of traffic and costs is achieved when the cache size is set to low values, i.e., $10 - 20\%$. We further showed that our MCDN solution reduces the signaling in the mobile network and the overall video traffic (video chunks) between video server and the radio access network. In particular, the longer the chunk, the less amount of video data should be transmitted, and the less signaling overhead is measured. When we consider also the tunneling overhead due to the mobility of the user, we note an effect in contrast to the previous observations with a good trade-off found for a chunk length of $4$ s. Overall, we validated our design choice and by means of simulation we evaluated the network resources consumed when providing the video services for a range of network settings, achieving promising results in terms of limited overhead and reduced video traffic generated in the network. Thanks to our large-scale network simulations, we got an insight into the potential of our MCDN-based approach, which is under testing at the time of writing, on a real testbed [1]. As future work, we plan to implement our framework on realistic platforms to evaluate the impact of our approach on wide-area practical deployments.

Starting from the latest trends in the mobile video industry and standards development, we proposed an E2E video delivery framework. We have designed two algorithms for the selection of a video path from the source to the end user, based on a set of network metrics that describe both the core and the access network performance, and we have evaluated the trade-off between the response time of the network and the capacity of the wireless channel for a broad range of settings. We have further implemented and evaluated in ns-3 our delivery system to assess the performance of the video transport over a realistic LTE cellular network model, with the goal of keeping the system throughput while improving the

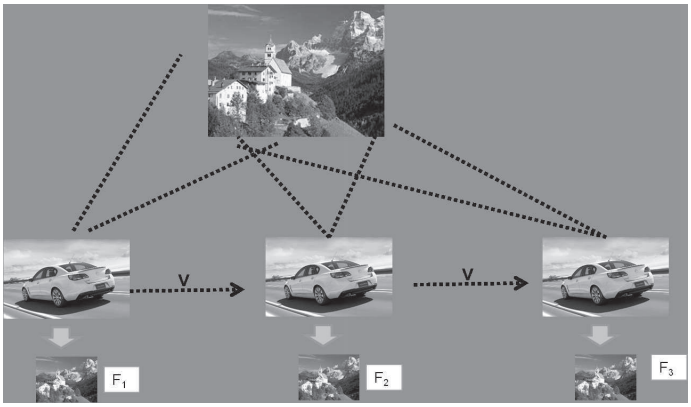



**Figure 6.1.** *Reference scenario.*

responsiveness, i.e., the packet delivery delay, experienced by the mobile users. By selecting online the best video path available for a set of network metrics, PS achieves remarkable gains in terms of delivery delay, while keeping or even slightly increasing the overall throughput of the system. When PDT is enabled jointly with PS, the delivery delay can be further decreased but at the cost of lowering the throughput and of additional computational complexity. The mapping of the QoE-related performance metrics was evaluated to eventually assess the impact on the perceived quality, and it reveals that our proposed mechanisms enhance the QoE by up to 1.2 MOS points compared to the baseline solution. We finally foresee that multimedia delivery systems over cellular networks would benefit from software-defined networking [16], i.e., controlling the network hardware remotely via software (as proposed in this work), which is enabled by OpenFlow-like [17] management tools for the exchange of information among core nodes, thus making the network infrastructure smarter and able to support better performance.

## 6.1 Work in progress

At the time of writing a new topic is currently under study, which will be briefly introduced in this section.

From an architectural point of view, we move further down to the client side. We focus on the real-time upstreaming use case as, for instance, in Fig. 6.1, where a mobile user shoots a video from the camera or the smartphone, and upstreams such video to the Internet for

sharing the contents with friends. The way the video frames generated should be encoded and scheduled to guarantee a certain expected video quality at the consumer side is crucial when considering the users mobility, the dynamics of the scene and the channel variations. Thus, our main contribution will be to model the frame correlation due to the double effect of the users mobility and scene changes in order to properly select the encoding rate and scheduling policies for a sequence of frames. We are designing a delivery framework where the decisions on the encoding rate of the video frames and on the scheduling policy are made based on the frame correlation and channel conditions. The mapping of user and scene dynamics to the frame correlation is challenging due to the simultaneous effect of the two sources of mobility on the video frames, which has not been explicitly addressed in the community so far.

Appendix A

# QoS and QoE

QoS represents a combination of several objective attributes of services, typically the bitrate, delay and packet error rate. There has been a common belief that by improving QoS the operators could provide higher levels of quality to users. In recent years this thinking has evolved to the concept of QoE. Rather than the performance statistics of the service, QoE tries to quantify the user experience and how it is impacted by the service performance. Specifically for video applications, experience of the application is more sensitive and has more dimensions compared to traditional applications. For this reason, for video applications there could be a broad definition of QoE, covering all aspects of a video application, e.g., satisfaction of video quality, user interfaces and devices.

As an original video is subject to several impairments during the delivery, the video quality perceived by users might be degraded. The quality of impaired videos can be measured by performing subjective tests, in which end users are asked to rate the videos. This method is not feasible in service and network development work. Objective video quality assessment methods are therefore extensively developed to be applied in multiple scenarios where the perceptual quality of videos is demanded without performing time-consuming subjective test. Based on the type of input data being used for perceptual quality assessment, the objective video quality assessment methods can be classified into several categories. One of them, that is widely used, is a media-layer method analyzing video signals to assess QoE. Many methods of this type need reference videos in order to assess QoE by comparing the distortion or similarity between the reference and the impaired videos. Traditionally, MSE and PSNR are point-based methods but they are not perfectly correlated with perceptual

quality measurements due to the non-linear behavior of the human visual system. SSIM [81] considers the perceptual structural information loss as the cause of quality degradation. The Video Quality Metric (VQM) [82,84] extracts visual features and combines their impairments to compute the QoE. The perceptual quality of videos is rated numerically by MOS levels, see Table A.1, via subjective assessment tests as recommended in [85, 86]. The goodness of the performance of the individual objective assessment method is revealed by comparing the MOS levels rated by subjects with the computed quality values via the aforementioned objective assessment methods.

**Table A.1.** *Mapping SSIM to Mean Opinion Score scale*

| SSIM | MOS | Quality | Impairment |
|---|---|---|---|
| $\geq 0.99$ | 5 | Excellent | Imperceptible |
| $[0.95, 0.99)$ | 4 | Good | Perceptible but not annoying |
| $[0.88, 0.95)$ | 3 | Fair | Slightly annoying |
| $[0.5, 0.88)$ | 2 | Poor | Annoying |
| $< 0.5$ | 1 | Bad | Very annoying |

Appendix B

# H.264/SVC

One major challenge that streaming video applications need to mitigate is the diversity in network performance as well as device and end-user equipment capabilities. Consider the scenario where a given HD input video stream is encoded and streamed to multiple end users. Some users may suffer from poor bandwidth and will not be able to receive the streamed video. Some users may have old receivers and decoders, which are unable to decode HD videos, and some users may have perfect conditions for the HD experience. In that case, the video service provider will have to perform multiple encoding of the same video stream, and deliver the most suitable stream to each client. H.264/SVC [7] is a set of extensions that promises to encode once, and play everywhere. When using SVC, the video stream is encoded in multiple video quality layers. The first layer provides the most basic video quality (often referred to as base layer) and other layers enhance the overall quality of the video (called enhancement layers). The more scalable enhancement layers the SVC stacks, the more diverse bit-rates, frame-rates, and resolutions it is possible to support. The base layer is encoded using a full standard compatible H.264 AVC and can be decoded independently. SVC offers three types of scalability (Fig. B.1):

- Spatial scalability: the video frame size and resolution can be adapted to match the capabilities of heterogenous mobile devices, e.g., HD, CIF or QCIF screens;

- Temporal scalability: the frame rate can be increased/decreased to take into account the motion smoothness required at the user side;

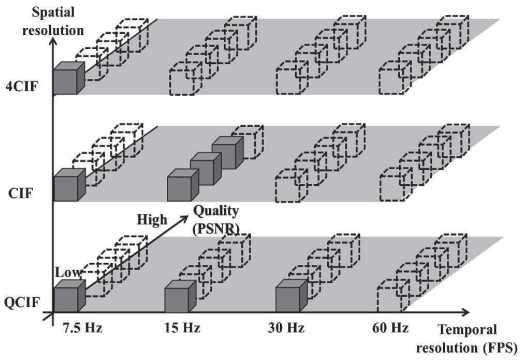



**Figure B.1.** *Example of H.264/SVC scalability in three dimensions: spatial, temporal and quality.*

- Quality scalability: the quality perceived by the user can be adapted to, for instance, the level negotiated with the mobile operator.

Appendix C

# Realtime Redundancy Allocation for Time-Varying Underwater Acoustic Channels

The recent development and employment of autonomous underwater vehicles, underwater sensors, and acoustic buoys motivate the interest in the design of reliable and energy-efficient underwater acoustic (UWA) communication strategies. On the one hand, reliable communications translate into an efficient utilization of the available bandwidth, which is a scarce resource in the UWA channel. On the other hand, energy-efficiency makes it possible to extend the lifetime of the aforementioned devices, which are usually battery-powered. Since transmission is the most energy consuming activity of an acoustic modem [87, 88], energy can be saved by reducing as much as possible unsuccessful transmissions and the amount of unnecessary redundancy.

Following this reasoning, in this work we propose an optimization framework and a realtime algorithm, able to jointly address reliability and energy-efficiency over time-varying conditions in a communication link. For the sake of simplicity, we model the UWA channel as a time-varying Binary Symmetric Channel (BSC) and we support this assumption by showing experimental UWA channel conditions and the corresponding communications performance. Moreover, we compute the amount of redundancy which maximizes a metric, suitable for representing how efficiently the information is encoded in terms of both spectral

efficiency and energy consumption. Furthermore, we design a realtime algorithm, able to allocate the precomputed optimal amount of redundancy. Finally, we evaluate its performance in both rapidly and slowly time-varying channel conditions, such as those measured during the KAM11 experiment [89].

As widely investigated in the past few years for terrestrial wireless communications, e.g., see [90–94], energy saving can be achieved by designing scheduling schemes characterized by low-power and long (low-rate) transmissions. However, these energy-efficient transmission schemes assume block-fading or additive white Gaussian noise channel models, which may not be suitable for representing the doubly selective UWA channel over intervals of time covered by long packets. Therefore, further analysis and validation is needed to tailor these energy-efficient schemes to UWA communications.

In the literature, the problem of reliable UWA communications has been studied, e.g., in [95–100]. In particular, the authors of [95, 96] investigate the performance of rateless coding schemes used for broadcast communications. In [97], the authors study the tradeoff between delay and reliability in a multi-hop sensor network. The authors of [98] perform a simulation study on the reliability achieved by the proposed Automatic Repeat reQuest (ARQ) technique, tailored to UWA communications. Even though these studies provide insight on the reliability associated to a multi-user channel in different networking scenarios, the presented results are limited to the case of time-invariant channel fading, since only the dependence on distance is considered.

In contrast to this previous work, we focus on the unreliability due to time-variability in an acoustic link. In particular, we design and evaluate a framework which allocates in realtime the redundancy required to protect UWA transmissions, based on limited channel side information (CSI). This CSI is provided by either an acknowledgement (ACK) or not-acknowledgement (NACK) sent by the receiver to the transmitter, upon the receipt of a packet.

In [100], the authors design a super-Nyquist modulation and rateless coding scheme suitable for doubly-selective underwater acoustic channels. However, they do not explicitly consider optimality allocating the redundancy to decrease the number of retransmissions. The authors of [101] propose a rateless coding scheme, whose soliton distribution adapts to the fading conditions and which runs based on limited CSI (ACK/NACK) available at the

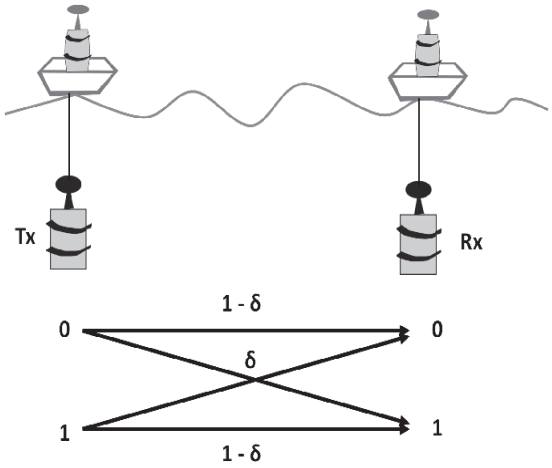



**Figure C.1.** *Typical UWA scenario, single-user communication channel. Below, the corresponding binary symmetric channel, with crossover probability $\delta$.*

transmitter. They show that this adaptive rateless code outperforms the standard rateless codes in terms of throughput. In this work a similar adaptive coding scheme is considered. However, here we calculate in real time the precomputed optimal amount of redundancy without introducing extra control messages.

The structure of the work is summarized as follows. In Sec. C.1, we present the considered scenario and channel model. In the same section, we define a metric representing the encoding efficiency. In Sec. C.2, we formulate the optimization problem, from which we compute the amount of redundancy that maximizes the aforementioned metric. In Sec. C.3, by leveraging on these results, we design a realtime algorithm, able to allocate the redundancy in an actual UWA scenario and we evaluate its performance by considering both rapidly and slowly time-varying channel conditions. Sec. C.4 concludes the work.

## C.1 System model

The scenario consists of an acoustic communication system as represented in Fig. C.1, where a single transmitter-receiver pair is considered and no multi-user interference occurs. This is an appropriate model for a deterministic medium access control scheme, such as Time Division Multiple Access (TDMA), which has been proposed and studied for UWA networking scenarios [102, 103]. Moreover, we assume a fixed traffic generation rate. In fact, differently from the power control schemes that aim at maximizing the amount of informa-

tion for a given average power constraint, here we aim at maximizing the spectral efficiency per transmission, subject to a fixed amount of information to be successfully transmitted, which is a practical scenario for UWA communications.

More specifically, we consider the scenario of an adaptive coding scheme, which can adjust the amount of redundancy per transmission. At each transmission, the amount of redundancy is chosen according to the channel conditions, by solving the optimization problem presented in Sec. C.2 with the algorithms proposed in Sec. C.3.

### C.1.1 Metric and channel model

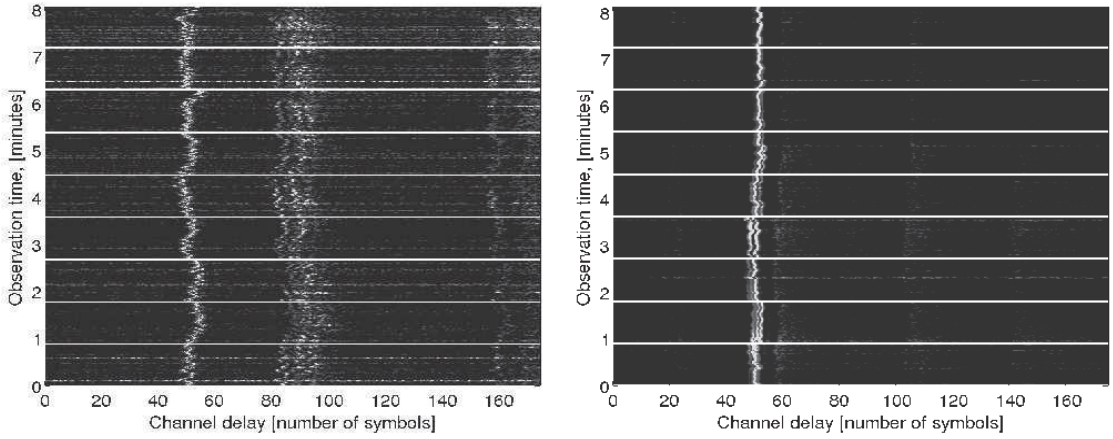
In the following, we indicate the amount of information to be transmitted and the corresponding redundancy as $x$ and $y$, respectively. The overall codeword length, $n$, is given by $n = x + y$. The efficiency metric to be maximized is defined as:

$$\eta(x, y, \epsilon) = \frac{x(1-\epsilon)}{x+y}, \tag{C.1}$$

where $\epsilon$ is the codeword error probability. This error probability (and its approximation) was derived in [104] for several channel models in the finite block-length regime. In particular, in this work, we consider the BSC model, represented in Fig. C.1. The choice of this model is mainly due to the fact that it makes it possible to obtain close-form results, and therefore simplifies the analysis of the system, while also matching the collected experimental data for a point-to-point UWA channel sufficiently well.

The efficiency metric, $\eta$, represents the trade-off between reliability, indicated by the factor $1-\epsilon$, and spectral efficiency, expressed by the ratio $x/(x+y)$. In fact, $\eta$ is a decreasing function of $y$, and an increasing function of $1-\epsilon$. However, $1-\epsilon$ itself is an increasing function of $y$, thus revealing a trade-off, which depends on how rapidly $\epsilon$ decreases as redundancy increases. This trade-off gives rise to an optimum value $y_{opt}$ that maximizes $\eta$, and should be used to optimally allocate redundancy. In fact, transmitting more redundancy would only increase the energy consumption at the transmitter, since unnecessary bits would be sent. On the other hand, if insufficient redundancy is transmitted, the receiver will request a retransmission by sending a NACK, thus increasing both energy consumption and delay.

The codeword error probability, $\epsilon$, depends on $x$, $y$, and the channel conditions. In case of a BSC model, fully described by the crossover probability $\delta$, $\epsilon$ depends on $x$, $y$, and $\delta$.




(a) Deployment A.

(b) Deployment B.

**Figure C.2.** *Time series of the amplitude estimates of the channel impulse response, during Julian dates 181 at 4 p.m. (UTC) (deployment A) and 187 at 4 a.m. (UTC) (deployment B). The $x$-axis corresponds to the channel delay, whereas the $y$-axis represents the recording time, which spans 9 minutes.*

In particular, in [105] and in [104, pag. 51-54] the author derives an upper bound for the achievable rate, $\log M(\epsilon, n)$, from which we derive the expression for $\epsilon$ as a function of $x$, $y$, and $\delta \notin \{0, \frac{1}{2}, 1\}$, which can be written as:

$$\epsilon(x, y, \delta) = Q\Big(\frac{nC(\delta) - x + \frac{1}{2}\log n}{\sqrt{nV(\delta)}}\Big), \tag{C.2}$$

where

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-\frac{w^2}{2}} dw. \tag{C.3}$$

The capacity, $C(\delta)$, of a BSC with crossover probability $\delta$ is:

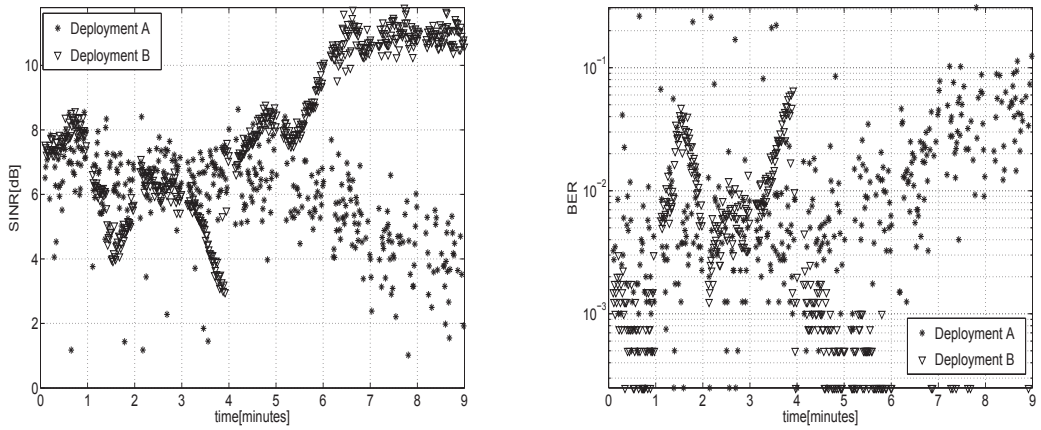
$$C(\delta) = 1 - h(\delta) \tag{C.4}$$

where $h(\delta)$ is the entropy equal to:

$$h(\delta) = -\delta \log(\delta) - (1 - \delta) \log(1 - \delta) \tag{C.5}$$

whereas $V(\delta)$ is called channel dispersion, defined in [104, pag. 12], and in case of a BSC becomes:

$$V(\delta) = \delta(1 - \delta) \log^2 \frac{1 - \delta}{\delta}. \tag{C.6}$$




(a) SINR time series (b) BER time series ($y$-axis is in logarithmic scale)

**Figure C.3.** *SINR and BER time series, from the KAM11 experimental campaign during Julian dates 181 (4 p.m.) and 187 (4 a.m.), indicated with stars and triangles, respectively.*

This quantity indicates the coefficient, $V$ in the approximation valid for different channel models:

$$\log M(n, \epsilon) = nC - \sqrt{nV}Q^{-1}(\epsilon) + O(\log n). \tag{C.7}$$

The rationale behind the choice of the BSC model lies in its simplicity as well as in its suitability to provide insights on how to optimally allocate the redundancy over subsequent packets affected by time-varying channel conditions. Such approximation is verified using experimental data. However, it is also worth noticing that our framework is independent of the channel model, as soon as the codeword error probability can be expressed as a function of the channel conditions, $x$ and $y$. Understanding which model would be more accurate for real underwater acoustic communications is under study. In order to qualitatively support this model, we present some experimental results in terms of Signal to Interference plus Noise Ratio (SINR)[1] and Bit Error Rate (BER) during two different deployments, subject to different time-varying channel conditions.

### C.1.2 Experimental channel evaluation

In order to support the suitability of the BSC model, we analyze a set of acoustic signals transmitted under water and collected during the KAM11 experimental trial [89]. In particular, we focus on a train of nine almost one-minute long acoustic signals, BPSK modulated at center frequency $13$ kHz and rate $6250$ bps. We consider the signals recorded at $3$ km from the transmitter. Furthermore, in order to show different time-varying conditions, we present the results for two deployments, where the shallowest transducer was deployed at A) $15$ m and B) $45$ m below the surface. The transmitter was $45$ m below the surface for both deployments. We remark that these two deployments also refer to different time intervals of the experimental campaign.

However, since consistent channel behaviors were observed in each deployment for several consecutive days, and the two deployments were tested in adjacent time periods, we may conclude that the critical factor affecting the observed behavior is the different position of the receiver, rather than the time at which the measurements were taken. As an example, we represent in Figs. C.2(a) and C.2(b) the time series of the amplitude of the channel impulse responses for A and B, respectively.

When the receiver moves due to surface fluctuations, this produces impulse noise represented by tiny horizontal lines in the channel impulse response estimates in Fig. C.2. In deployment A, since the receiver is closer to the surface, this effect is more visible (see Fig. C.2(a)), whereas in deployment B, as shown in Fig. C.2(b), the channel exhibits a more stable structure of arrivals and the impulse noise is negligible. We remark that the horizontal thick white lines in Figs. C.2(a) and C.2(b), separating each minute of observation, represent the silent time interval between subsequent transmissions of the acoustic signals.

Similar observations hold for Figs. C.3(a) and C.3(b), representing the time series of the BER, $\delta$, and of the SINR, estimated over subsequent chunks of the transmitted signal, $5190$ symbols long, of which $1100$ are used for synchronization and initialization of the Decision Feedback Equalizer (DFE)[2]. We remark that the quantization effect in the low BER regime

[1]Note that interference here means Inter-Symbol Interference (ISI), since the considered communication system is ultra-wide band in a communication link.

[2]The equalizer is used in training mode. It combines the signals measured at four channels spanning half a meter of the column water. Moreover, in order to compensate for time-varying channel durations, we adapt accordingly the length of the feedback filter.

(below $10^{-3}$) is due to the fact that we estimate BER over limited sized packets. Whenever the receiving system is unable to correctly synchronize and to estimate the channel, the chunks are dropped from the analysis.

As a final note, we want to stress that the BSC model could be more accurate for deployment B, for which slower time-varying channel conditions were measured. In fact, if a packet lies within a channel coherence time, the error statistics are time-invariant, and thus can be represented by a single crossover probability $\delta$. On the other hand, if we consider longer packets, e.g., $10\,\mathrm{s}$, which possibly span multiple channel coherence times, sometimes of the order of a few seconds, a more complex model, such as a Markov model with memory, should be taken into account. This extension is left for future work.

## C.2 Optimization problem

In this section, we develop the optimization framework that, based on a given number of information bits $x$ and on the bit error rate $\delta$, infers the optimal amount of required redundancy as follows:

$$y_{opt} = \underset{y}{\operatorname{argmax}}\, \eta(x, \epsilon, y), \tag{C.8}$$

where $x$ is constant and application-dependent and $y$ is chosen so as to maximize $\eta(x, \epsilon, y)$, under given channel conditions. In order to find a maximizer, we study the sign of the first derivative of $\eta(x, \epsilon, y)$ with respect to $y$. Such derivative can be expressed as:

$$\frac{\partial \eta}{\partial y} = \partial_y \Big( \frac{x(1 - \epsilon(x, y, \delta))}{x + y} \Big) \tag{C.9}$$

$$= \frac{-(x + y)x\partial_y \epsilon(x, y, \delta) - x(1 - \epsilon(x, y, \delta))}{(x + y)^2}. \tag{C.10}$$

In the numerator, the first term has only positive factors, with the exception of $\partial_y \epsilon(x, y, \delta)$, which is negative, since $\epsilon$ is a strictly decreasing function of $y$. This makes the first term positive. Conversely, $x(1 - \epsilon(x, y, \delta))$ is non-negative, since $x > 0$ and $\epsilon(x, y, \delta) \in [0, 1]$, thus making the second term non-positive. These considerations highlight that there exists a trade-off between the two terms, which determine the sign of $\partial_y \eta(x, y, \delta)$ as a function of $y$.

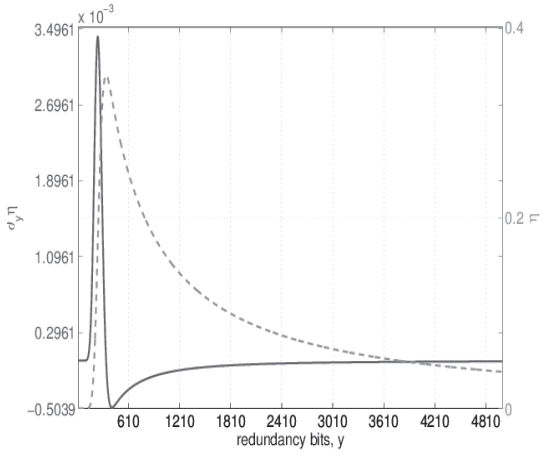



**Figure C.4.** *$\eta$ and its first derivative are represented in dashed and solid curves, respectively. In this case, $x = 200$ bits and $\delta = 0.046$.*

### C.2.1 Numerical results

We numerically evaluate the solutions provided by the optimization framework. In particular, we consider $x$ varying in the interval $[200, 1000]$, with an increment of $50$ bits, and $\delta$ spanning the interval $[10^{-3}, 0.1]$, with increments of $0.005$. In the considered range for $\delta$, there exists a $y_{opt}$ maximizing $\eta$ (an example of this behavior is shown in Fig. C.4). For very small values of $\delta$, $\eta(x, y, \delta)$ turns out to be a monotonically decreasing function of $y$, so that the optimal value is $y_{opt} = 0$ and no redundancy should be used (i.e., it is better to take a chance and then retransmit whenever needed instead of investing resources to provide a priori error protection). On the other hand, for values of $\delta$ close to 0.5 the maximum efficiency would occur for very large values of $y$, which would lead to impractical values of the packet size. In this case one may either limit $y$ to the maximum value allowed, or refrain from transmission altogether.

As an example of the obtained results, in Fig. C.4, we show $\eta(x, y, \delta)$ and its first derivative with respect to $y$, when $x$ is $200$ and $\delta$ is $0.046$. It can be noticed that $\eta$ slowly increases for small $y$ ($\partial_y \eta \sim 0$), whereas as $y$ increases, it also steeply increases since the successful decoding probability tends to $1 (\partial_y \eta > 0)$. Afterwards, $\eta$ decreases for larger values of $y$ ($\partial_y \eta < 0$).

In Figs. C.5 and C.6, we show the computed $y_{opt}$ for the considered intervals of $x$ and $\delta$. In particular, each curve in Fig. C.5 is obtained by cutting the 3D plot in Fig. C.6 with a vertical

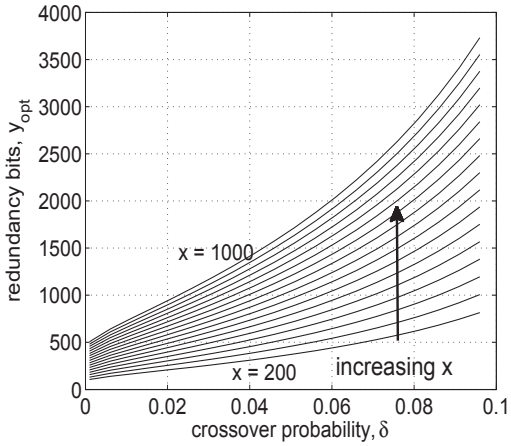



**Figure C.5.** *The $y$-axis represents the optimal number of redundancy bits, obtained from the optimization framework, for varying $x$ (represented by the different curves) and $\delta$ (in the $x$-axis).*

plane for a constant value of $x$. We make use of these results in the realtime algorithm for redundancy allocation as they become precomputed values of $y_{opt}$ in a look-up table, as explained in Sec. C.3.

## C.3 Algorithms and evaluation

**Algorithm 3** Algorithm to compute the optimal FEC, as for Sec. C.2.

**Input:** $\delta$ bit error rate, and $x$ information bits;

**Computation optimal FEC, $y_{opt}$ :**

$\cdot\ V(\delta) = \delta(1-\delta)log^2(\frac{1-\delta}{\delta})$;

$\cdot\ C(\delta) = log2 - \delta log\frac{1}{\delta} - (1-\delta)log\frac{1}{1-\delta}$;

$\cdot$ Compute $y_{opt} = \underset{y}{\operatorname{argmax}}\ \eta(y)$;

**Output:** $y_{opt}$ .

In this section, we design an allocation algorithm, starting from the transmitter side, which is in charge of allocating the redundancy based on the ACK/NACK fed back by the receiver.

Algorithm 3 is responsible for computing the optimal redundancy needed for the next transmission based on $\delta$, the measured bit error rate, and $x$, the number of information bits to be transmitted (see the optimization problem in Sec. C.2). Beforehand, in an offline

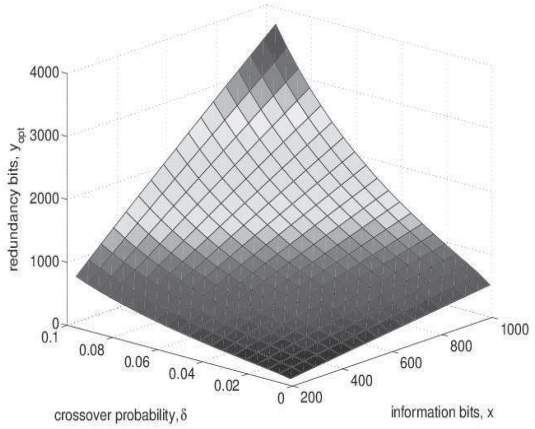



**Figure C.6.** *Surface plot of the numerical results obtained for* $x \in [200, 1000]$, *and* $\delta \in [10^{-3}, 0.1]$.

fashion, the operator should build a look-up table, containing couples of BER, $\delta$, and the corresponding optimal amount of Forward Error Correction (FEC), $y_{opt}$. The number of entries is fixed based on the granularity and range of the levels of quality of service required by the application. The more the couples, the finer the tuning of the redundancy to be allocated in the packets to be transmitted. As an extreme case, one entry in the table corresponds to fixing a constant packet length for any value of $\delta$.

In detail, in order to pre-build the look-up table at the transmitter side, algorithm 3 runs over a set of pre-computed $\delta$s (offline computation). For each $\delta$, it calculates the dispersion $V(\delta)$ and the capacity $C(\delta)$, and thus the corresponding optimal number of redundant bits $y_{opt}$. This procedure is repeated for each entry of the look-up table.

At this point, it is worth noticing that, since $\delta$ cannot be easily made available to the transmitter, during the realtime redundancy allocation phase, we need to map somehow a metric, e.g., the SINR, that the transmitter can estimate to the corresponding value of $\delta$. In order to do so, we preset a mapping function between the values of SINR and BER, estimated during an initial channel probing session. As an example, in Fig. C.7 we show the estimates of the couples (SINR, BER) for deployment A, KAM11. This plot is representative of a general behavior observed over most of the experimental data. Thanks to this behavior, the mapping function is inferred by associating to each SINR the corresponding average BER value.

Algorithm 4, implemented at the transmitter side, is responsible for allocating the redun-

**Algorithm 4** Algorithm at the transmitter side.

**Input:** $R$ channel realizations, BPSK modulation, $L_{packet}$ maximum packet length, $x$ information bits and look-up table ($BER_i$, $FEC_i$);

**Procedure:**

· Estimation of SINR upon chirp receipt;

· Mapping of SINR to BER, getting $\delta$;

· Selection of the $i-th$ level from the look-up table ($BER_i$, $FEC_i$), for which $BER_i$ is the closest value to the measured $\delta$:

· $i = \underset{j}{\operatorname{argmin}} |\delta - BER_j|$;

· Selection of $FEC_i$;

**if** $L_{packet} \geq (x + FEC_i)$ **then**

**Output: send a packet of length** $x + FEC_i$.

**else**

**Output: send probe packet**.

**end if**

dancy during subsequent transmissions. In particular, the transmitter is aware of the SINR seen at the receiver, $\hat{\gamma}$. This measure of the channel quality can be collected through a chirp sent by the receiver. We propose to use, e.g., an ascending chirp as ACK and a descending chirp as NACK. This solution is more efficient than sending a feedback message, containing only one bit of ACK/NACK and including large overhead in order to cope with the harsh UWA channel conditions. Once the SINR is measured, it can be mapped to an average BER, $\delta$, as aforementioned. Finally, the transmitter selects the corresponding value of FEC, thus resulting into a new packet. If the total length exceeds the maximum allowed packet length, denoted as $L_{packet}$, no transmission is performed, otherwise the packet is sent out.

At the transmitter side, we can also distinguish two cases, according to whether the look-up table is computed i) online, and ii) offline. In case i), every $T_{clock}$ ticks, the receiver sends a train of probing signals, known at the transmitter side, in order to be able to estimate the corresponding $\delta$. In case ii), before deployment each link is characterized through a look-up table, in the same way but for longer periods of time.

During the communication session, the receiver's tasks are to decode the packet and send an ACK if the decoding is successful, or a NACK otherwise. It can be noticed that

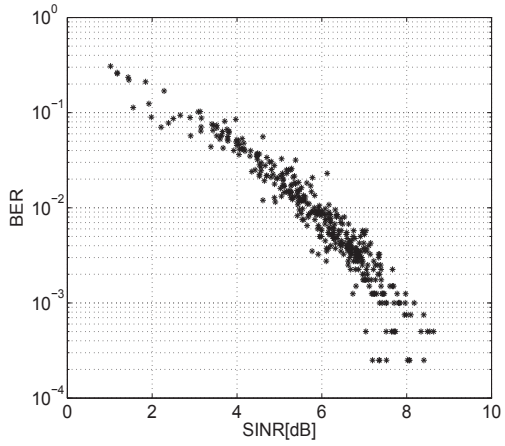



**Figure C.7.** *BER vs. SINR for the collected data during Julian date 181 (4 p.m.), deployment A.*

the proposed communication system leaves the burden of selecting the suitable amount of redundancy as well as the channel probing responsibility to the transmitter side. This is justified by the fact that a more complex feedback procedure (other than ACK/NACK, e.g., SINR estimation at the receiver fed back to the transmitter) may not pay off in terms of performance in the presence of long propagation delays that would make the feedback too outdated to be useful.

### C.3.1 Results

In this section, we evaluate the proposed redundancy allocation algorithm by means of simulation. In particular, we consider the experimental data shown in Figs. C.2(a) and C.2(b), which are representative of the different deployment depths.

First, we compute a finer SINR time series over the $9$ minutes. Each SINR is estimated by processing packets consisting of 1300 symbols ($208$ ms), so as to make such time series suitable for the simulation of multiple packet transmissions. At every simulation run, we read the estimated SINR, based on which we compute the proper amount of redundancy $y_{opt}$. Then, we run the receiving decoding processing over the chunk of data immediately subsequent to the time of the SINR estimate. If the currently measured bit error rate, $\delta(i)$[3], is greater than the BER assumed in the computation of $y_{opt}$, the packet is considered un-

[3]We remark that we focus on a general result, independent of the type of implemented encoder/decoder, therefore here we do not implement a specific coding scheme.

| | $\eta_{0.001}$ | $\eta$ | $\eta_{0.1}$ | average SINR dB |
|---|---|---|---|---|
| Dep. A | 0.028 | 0.318 | 0.201 | 5.8459 |
| Dep. B | 0.371 | 0.465 | 0.207 | 8.1168 |

**Table C.1.** *Results in terms of $\eta$, $\eta_{0.001}$, and $\eta_{0.1}$ evaluated over deployments A and B.*

successfully decoded. At the next iteration, the subsequent considered SINR is separated in time by the last packet duration plus the round trip time (which here is $4$ seconds). In this way, we could estimate the metric $\eta$, as:

$$\hat{\eta} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\delta(i) \leq BER(i)\}\frac{x}{x+y(i)} \tag{C.11}$$

where $N$ is the number of iterations and $\mathbf{1}\{\delta(i) \leq BER(i)\}$ is the indicator function. Similarly, we compute $\eta$ in two cases of constant packet lengths obtained for i) $BER = 10^{-3}$, and ii) $BER = 0.1$. We denote the obtained evaluation of $\eta$ as i) $\eta_{0.001}$ and ii) $\eta_{0.1}$.

The results are shown in Table C.1. It can be noticed that the proposed optimal allocation provides the highest encoding efficiency for both deployments, which have different communication channel qualities, as highlighted in the last column. The constant allocation assuming the smallest BER provides higher efficiency than the other constant allocation for deployment B, in which the channel conditions were favorable. Vice-versa, for deployment A, the more robust constant allocation outperforms the other. In both cases, an adaptive redundancy allocation, based on the proposed BSC model, would gain around $58\%$ and $25\%$, with respect to the corresponding suboptimal constant allocation in deployment A and B, respectively.

These results motivate further analysis on how to encode larger packets, which are more sensitive to the time-varying conditions of the arrival structure of the channel.

## C.4 Conclusions and future work

In this work, we built an optimization framework, well supported by experimental evidence. We used a BSC channel model to reflect in a simple way but without loss of generality the channel conditions measured in the collected data. We defined a metric representing how efficiently the information is encoded in terms of both spectral efficiency and energy

consumption and we formulated the optimization problem to maximize such metric. Finally, we designed a realtime algorithm to compute the redundancy required in a UWA communication link.

The presented study and results pave the way for future work. In particular, we plan to evaluate the efficiency of the proposed algorithm as a function of average SINR and channel coherence times, estimated in a more extensive data set. Moreover, we want to investigate how to allocate in realtime the redundancy over longer packets, for which Markov channel models should be validated. As a final goal, we want to understand whether short or long packets are more efficient (in terms of both bandwidth and energy) in UWA communication systems.

Appendix D

# Performance Evaluation of FEC techniques based on BCH codes in Video Streaming over Wireless Sensor Networks

## D.1 Introduction

Wireless Sensor Networks (WSNs) technology has experienced a rapid growth in the last several years, supported both by the research community and private stakeholders. WSNs are nowadays extensively adopted in a wide range of applications, where they replace old wired and wireless systems, needing power and connection cables, thus more expensive and hard to setup. A reduced set of WSNs applications include climatic monitoring [106], structural monitoring of buildings [107, 108], human tracking [109], military surveillance [110], and, more recently, multimedia related applications [111, 112].

The development of the so-called Wireless Multimedia Sensor Networks (WMSNs) has been fostered by a new generation of low-power and very performant microcontrollers, able to speed-up the processing capabilities of a single wireless node, as well as the development of new micro-cameras and microphones induced by the mobile phones industry. The WMSNs application fields are various and quickly growing. As matter of example we cite active

surveillance of sensitive ambients [113] and Intelligent Transport Systems (ITS) scenarios in which tiny devices equipped with a camera can be used to collect traffic related data as in the IPERMOB [114] project. In the above mentioned examples the use of multimedia information (particularly in respect of video) is completely different. While in ITS related applications the multimedia data can be processed on-board, thus transmitting aggregated information only (e.g., number of cars per unit of time passing through a checkpoint), in case of video surveillance a full stream of images is usually sent through the network.

Multimedia streaming in WMSNs is a challenging application. Considering wireless sensor networks based on the IEEE802.15.4 [115] standard, a very big issue is that if matching with the transmission bandwidth lower than 250 Kbps. This limitation implies the transmission of small size images (e.g., 320x240, 160x120) at very low frame rates to avoid network congestion [116]. Furthermore, where image compression techniques can not be applied due to their computational complexity, raw images must be sent, thus reducing again the image size, the frame rate or both of them. Taking into account these limitations, it is significant to stress that, proportionally to a decrease in the image size and in the frame rate, each image acquires more significance (e.g., a malicious behavior must be discovered using a smaller sequence of images). Another point to consider for developing an effective multimedia streaming application in such a scenario is the variability of the wireless channel which causes unexpected errors in the transmitted bits and consequently packet losses. The image corruption, due to packet losses, produces a reduction of the available amount of the information at the receiver side. Due to the importance of a single data packet, it is highly recommendable to apply error protection techniques where possible. As a matter of example in [117] the use of Unequal Error Protection (UEP) based on network resource allocation (e.g., transmission power) for compressed images is evaluated.

In this work we propose and evaluate the performance of Forward Error Correction (FEC) techniques based on BCH [118] codes for recovering bit errors in case of multimedia streaming over IEEE802.15.4 networks. In a real outdoor scenario we first measure the Bit Error Rate (BER) for one-hop transmissions (network organized in a star topology) and we show its impact on the video streaming quality in terms of Peak Signal-to-Noise Ratio (PSNR). Then we detail the use of the BCH codes as a solution fully compliant with the IEEE802.15.4 standard, presenting a simulated performance evaluation, based on the col-

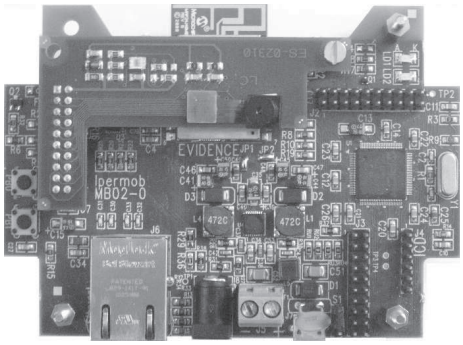



**Figure D.1.** *The IPERMOBv2.0 board developed within the IPERMOB project.*

lected loss traces, as a function of the error correction capability of the code.

The rest of the work is organized as follows: in Section D.2 we present the BER data collection results in a real scenario, as well as the impact of losses on video streaming quality. In Section D.3 we discuss how BCH codes can be applied in IEEE802.15.4 as a fully compatible extension of the standard. The performance evaluation of the proposed error recovery strategy is presented in Section D.4, conclusions follow in Section D.5.

## D.2 BER evaluation in IEEE802.15.4 networks and impact of losses on video streaming quality

### D.2.1 Hardware and software

The experimental BER for wireless transmissions based on the IEEE802.15.4 standard has been evaluated by using the IPERMOBv2.0 board, depicted in Fig. D.1. More in detail the board is composed by:

- *Micro-Controller Unit.*
  The MCU is the PIC32MX795F512L, a 32 bits, 80 MIPS low cost integrated circuit produced by Microchip$^{TM}$.

- *Transceiver.*
  The transceiver is the Microchip$^{TM}$MRF24J40MB which is fully compliant with the IEEE802.15.4 standard and characterized by an omnidirectional diagram pattern and a transmission power up to +20 dBm.

- *Camera.*

  The camera mounted on the device is the HV7131GP, a CMOS based camera which can be configured via MCU for acquiring images at various resolutions (up to 640x480) and frame rates (up to 30 fps).

- *Ethernet interface.*

  An IEEE802.3 interface is present on the board, implemented exploiting the MCU MAC functionality and an external physical layer adapter (LAN8720).

- *Serial interface.*

  A serial interface (RS-232) can be used with an external TTL-to-RS232 converter which is connected to the UART interface provided by the MCU.

The board has been specifically designed to support high demanding multimedia applications while requiring low power consumption during image acquisition (75 mA for 160x120 images at 1 fps).

The data acquisition firmware for collecting real BER traces has been developed as a custom application on the top of the ERIKA RTOS [119,120], an innovative real time operating system for small microcontrollers that provides an easy and effective way for managing tasks. Furthermore, the transceiver driver allows to access the MRF24J40MB functionality with a minimal time overhead.

### D.2.2 Data collection scenario and BER results

The data collection scenario selected for evaluating the BER is a parking area in the Pisa Airport. The full scenario with nodes positions and obstacles is depicted in Fig. D.2. The


**Figure D.2.** *The data collection scenario within the Pisa Airport area.*

| Position | Distance [m] | BER | BL [bits] | LQI | RSSI |
|---|---|---|---|---|---|
| 1 | 64 | $2.18 \cdot 10^{-4}$ | 1.66 | 108.38 | 120.97 |
| 2 | 53 | $1.69 \cdot 10^{-3}$ | 1.61 | 99.86 | 122.09 |
| 3 | 43 | $6.56 \cdot 10^{-7}$ | 2.23 | 116.38 | 147.88 |
| 4 | 68 | $1.43 \cdot 10^{-4}$ | 1.69 | 110.86 | 131.14 |
| 5 | 57 | $1.24 \cdot 10^{-2}$ | 1.65 | 93.97 | 104.18 |
| 6 | 43 | $9.12 \cdot 10^{-3}$ | 1.65 | 94.61 | 121.96 |

**Table D.1.** *BER results for the selected positions in the Airport scenario.*

scenario is at of a typical video surveillance system in which a set of nodes (six in our case) are installed to monitor malicious events inside the parking. Each node of the system can in turns acquire and send an image to the network coordinator (at the left side of the picture) which works as a point of service for the backhauling network. The data collection environment is heavily affected by reflections as well as Non-Line-Of-Sight (NLOS) transmissions.

In the performed experiments the hardware devices have been installed at 2.5 m height and data packets have been sent at the maximum bitrate (transmission bandwidth saturated) collecting three traces for each position. The transmission power has been set to 9dBm to fulfil the ETSI requirements [121]. The results of the experimental analysis are reported, for each position, in Table D.1 in terms of BER, Burst Length (BL), Link Quality Indicator (LQI) and Received Signal Strength Indicator (RSSI). All results in the Table D.1 have been averaged among the three collected traces.

Due to the high variability of the selected scenario the BER values span in a range from $10^{-2}$ to $10^{-7}$. While on one hand we expected higher values of BER when the distance between sender and receiver increased (e.g., positions 1 and 3), on the other one, we faced, on the BER value, unpredictable effects due to obstacles in the scenario. As a matter of example, a trace collected in position 2 was acquired in condition of partial NLOS (a truck stopped for half of the data collection time in the middle of the transmission path) so that the associated BER sharply increased. Where the transmission in NLOS conditions is permanent

the BER values is dramatically high, as it happens for positions 5 and 6. Regarding the BL this is close to 1.6 for each position, with the exception of position 3, although for the low number of corrupted packets this value is not statistically significant. Concerning the LQI and the RSSI, BER is dependent on the LQI (when the LQI increases the BER decreases), while the RSSI is not simply related to the BER because its value is the sum of both effective signal and its reflections: higher values of RSSI do not coincide with lower values of BER (e.g., positions 1 and 2).

### D.2.3 Impact of bit errors on video quality

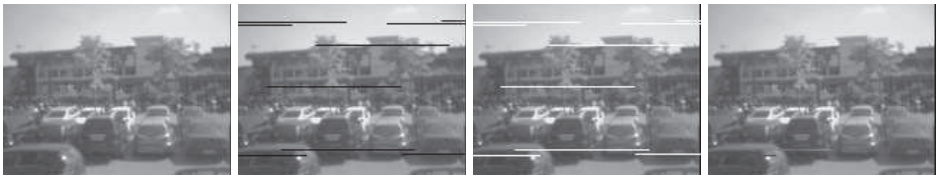
The impact of the bit errors on the transmitted video streaming has been evaluated, for each node position, simulating the transmission of grayscale (8 bits depth) raw images (no compression is applied) with a resolution of 160x120 pixels and frame rate equal to 1 fps. In the simulation the videos used for the performance evaluation are a set of real videos, with high motion, acquired with the hardware presented in Section D.2.1 and previously stored in a database.

In the simulations each image is fragmented according to the maximum payload allowed by the IEEE802.15.4 standard, and a data packet is considered lost if at least one error occurs (wrong frame check sequence value and packet discarded by the receiver). In the results an evaluation of the impact of three low-complexity concealment algorithms is shown, black insertion, white insertion and copy frame. In case of black insertion concealment the lost part of a video frame is replaced with black pixels, while in case of white insertion concealment the color of the replaced pixels is white. The copy frame concealment is more complicated with respect to the previous ones and it consists in replacing the lost parts of a video frame with the ones of the previous received frame. All results for the applied concealments are reported in Table D.2, in terms of PSNR, for a numeric comparison. In Table D.2 the PSNR in case of black concealment is labeled PSNR-bc, the one for the white concealment as PSNR-wc, while PSNR-cc is the video quality in case of copy frame concealment.

A very trivial and intuitive result from Table D.2 is that the video quality strictly depends on the experienced BER. Higher values of BER produce low values of PSNR for all the considered concealment techniques. In case of BER values higher than $10^{-3}$ the video quality reduction is bigger than 70% with respect to the highest value, while for BER of $10^{-4}$

| Position | BER | PSNR-bc [dB] | PSNR-wc [dB] | PSNR-cc [dB] |
|---|---|---|---|---|
| 1 | $2.18 \cdot 10^{-4}$ | 43.14 | 44.13 | 57.61 |
| 2 | $1.69 \cdot 10^{-3}$ | 14.30 | 16.47 | 37.12 |
| 3 | $6.56 \cdot 10^{-7}$ | 126.92 | 126.98 | 127.07 |
| 4 | $1.43 \cdot 10^{-4}$ | 61.21 | 61.90 | 69.75 |
| 5 | $1.24 \cdot 10^{-2}$ | 10.81 | 13.46 | 27.17 |
| 6 | $9.12 \cdot 10^{-3}$ | 27.45 | 31.78 | 32.54 |

**Table D.2.** *PSNR as a function of the BER and concealment techniques.*



(a) Reference image (b) Black concealment (c) White concealment (d) Copy concealment

**Figure D.3.** *Impact of the selected concealment algorithms for a selected loss trace.*

the video quality reduction is close to 45%. Even if a BER of $10^{-4}$ can not be considered as a critical value for IEEE802.15.4 wireless communications [122], it produces a substantial quality reduction of the received video stream. The selection of an appropriate concealment technique mitigates the effects of the packet losses, guaranteeing a gain in the quality of the received video. More in particular, the copy frame concealment outperforms all the other concealment techniques under test. A graphical output of the applied concealment solutions is depicted in Fig. D.3, where the frame with copy concealment is almost completely reconstructed.

In a WMSN scenario, in which tiny devices send image frames with low resolution and low frame rate, error recovery techniques must be applied in order to avoid poor video quality and possible artifacts in the reconstructed dynamics of the scene, as it could happen in video surveillance systems.

## D.3 Error recovery strategy based on BCH codes

In this work we consider the use of BCH codes as a possible FEC strategy for recovering bit errors. According to the coding theory, the BCH codes is a class of error-correcting block codes in which the coding and decoding procedures are characterized by low complexity, thus making these codes very suitable for a real implementation in low-power devices. In a simplistic statement the aim of the coding process is to add a certain number of redundancy bits to the initial block of information bits, thus providing the capability of correcting a certain number of wrong bits. Each BCH code is characterized by three parameters:

- *n:* the total number of bits after the coding procedure. Its value is given by the number of information bits plus the number of redundancy bits;
- *k:* is the number of the information bits which must be protected *(k<n)*;
- *t:* is the error correction capacity of the code. Each BCH code can correct up to $t$ errors in each block on $n$ bits, while adding $n - k$ bits of redundancy.

In literature a BCH code is identified by the above introduced parameters with the labelling BCH(n,k,t). The value $R_c = k/n$ is the code rate and is related to the redundancy level and overhead introduced by the code. Lower values of $R_c$ mean higher protection levels and higher additional overhead in terms of bits to transmit.

The use of FEC strategies based on BCH codes within IEEE802.15.4 networks require to define new policies in accepting corrupted packets at the receiver. According to the standard a transmitted data packet is composed, at the Medium Access Control (MAC) level, by


(a) Standard IEEE802.15.4 data message.


(b) Proposed MAC data message with FEC protection.

**Figure D.4.** *Standard and proposed MAC data messages.*

three main fields: header, payload and Frame Check Sequence (FCS) (Fig. D.4(a)). Once the packet is received a new FCS is evaluated and the packet is discarded in case of the transmitted and received FCSs are unequal. This approach can not be pursued if FEC strategies are applied, because it will result in neglecting the effects of the protection strategy itself. The proposed approach to apply FEC based on BCH codes within IEEE802.15.4 networks is depicted in Fig. D.4(b). The FCS field is evaluated only on the packet header, thus avoiding to deliver packets to the wrong node, while in the payload no check on the correctness of the data is applied. From the payload the $n$ bits of each code word are considered and the $k$ bits of the original information are extracted. The error correction capacity of the code guarantees the correction of up to $t$ errors for each code word. If a higher number of errors is experienced these can not be revealed and corrected. The proposed protection strategy is especially indicated for the transmission of multimedia data (e.g., speech, video) in which the residual errors do not affect the validity of the full packet, but slightly affect the quality of the received media stream.

The proposed FEC strategy does not require to change the MAC layer of the IEEE802.15.4 standard for a specific application, being fully compatible with it. The new data messages can be defined using the bits reserved from the standard for specifying new possible messages. The basic messages of the IEEE802.15.4 standard, as well as the reserved bits combinations are reported in Table D.3. The use of the reserved bits to extend the types of data messages does not impose changes to the network stack in the device firmware, although only devices with the new features will be able to use them.

| **Frame type value b2 b1 b0** | **Description** |
|---|---|
| 000 | Beacon |
| 001 | Data |
| 010 | Acknowledgment |
| 011 | MAC command |
| 100-111 | Reserved |

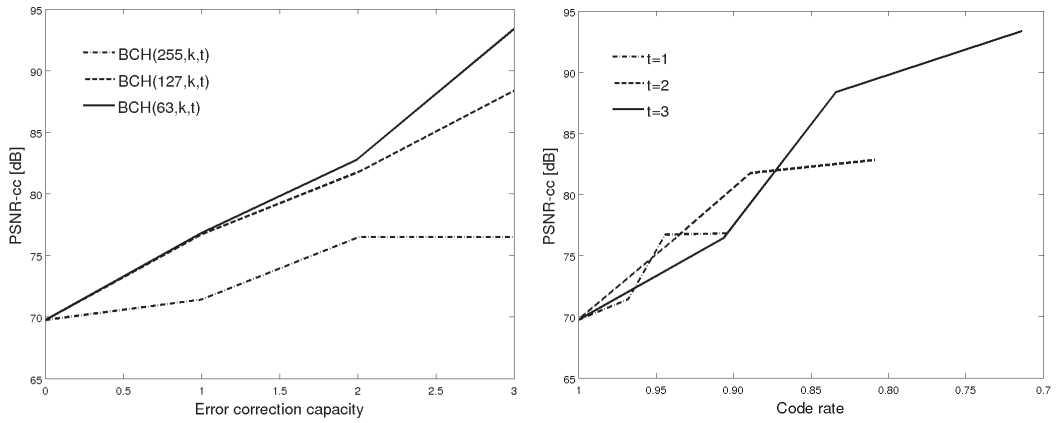
**Table D.3.** *Standard IEEE802.15.4 messages.*

| Protection | Rc | Overhead [%] | Error recovered [%] | PSNR-cc [dB] |
|---|---|---|---|---|
| No protection | 1 | 0 | 0 | 69.75 |
| BCH(255,247,1) | 0.968 | 19.31 | 3.54 | 71.41 |
| BCH(255,239,2) | 0.937 | 23.14 | 20.38 | 76.46 |
| BCH(255,231,3) | 0.906 | 27.36 | 20.38 | 76.46 |
| BCH(127,120,1) | 0.944 | 15.30 | 4.67 | 76.71 |
| BCH(127,113,2) | 0.889 | 22.51 | 24.82 | 81.75 |
| BCH(127,106,3) | 0.834 | 30.61 | 50.50 | 88.35 |
| BCH(63,57,1) | 0.904 | 17.94 | 5.71 | 76.81 |
| BCH(63,51,2) | 0.809 | 31.90 | 27.29 | 82.81 |
| BCH(63,45,3) | 0.714 | 40.74 | 53.62 | 93.35 |

**Table D.4.** *Video quality performance results of the proposed protection strategy for three classes of BCH codes.*

## D.4 Performance evaluation of the proposed error recovery strategy

The performance of the error recovery strategy proposed in Section D.3 has been evaluated with a simulative study as a function of the error correction capacity of the code ($t$) and its code rate ($R_c$). In the performed simulations real loss traces gathered in the scenario described in Section D.2.2 have been used. In particular we adopted the ones collected in position 4, characterized by a $1.43{\cdot}10^{-4}$ BER and by a video quality reduction of about 45% with respect to the highest value of Table D.2. In the simulation, in order to evaluate the PSNR, we used video streams composed by grayscale (8 bits depth) raw images (no compression is applied) with 160x120 pixels resolution at 1 frame per second (i.e. the same set used to evaluate the effects of bits errors in case of transmission without protections in Section D.2.3). In the performed analysis the copy frame concealment is the only one considered, due to its better performance in terms of PSNR with respect to black and white insertion concealments. All the results of the performed analysis are reported in Table D.4, where for each code is reported, together with the code rate, the additional overhead, the percentage of recovered errors and the obtained PSNR value. Three main classes of BCH codes have been considered, with $n$ equal to 255, 127 and 63 bits respectively. The overhead introduced by the code has been evaluated as the additional number of bits required




(a) PSNR as a function of the Error correction capacity of the code.

(b) PSNR as a function of code rate.

**Figure D.5.** *Traffic volume and overhead generated for static and mobile scenarios.*

(a) Reference image (b) No FEC applied (c) FEC applied with t=1 (d) FEC applied with t=3

**Figure D.6.** *Impact of the selected concealment algorithms for a selected loss trace.*

(including the packet header) to send the video frames.

The quality of the received video stream depends on two main factors: the error correction capacity of the code and the code rate. Increasing the error correction capacity ($t$) a higher percentage of errors is recovered, thus increasing the PSNR. This behaviour is shown numerically in Fig. D.5(a) and graphically in Fig. D.6, and it is common to all the three classes of codes analyzed. Low percentage of errors are recovered with $t = 1$, despite the high value of the additional overhead, this is because the code is not able to recover all the burst error, which is in average equal to 1.69. With $t \geq 2$ the improvement in percentage of recovered errors, and in PSNR, becomes more significant. In case of BCH(255,k,t) codes, no gain is experienced increasing $t$ from 2 to 3, the reason is due to the presence of a bigger number of burst errors in the block of bits protected by the code. Regarding the PSNR performance as a function of the code rate ($R_c$), its behaviour is depicted in Fig. D.5(b) for the considered error correction capacities. As much the code rate decreases, a bigger data

redundancy is applied and higher values of PSNR are reached.

The use of the proposed protection technique for video streaming over IEEE802.15.4 network guarantees significant performance improvements in terms of PSNR. Considering BCH codes with error correction capacity bigger than the average burst length and minimum overhead, BCH(127,113,2), the performance improvement in terms of PSNR is equal to 17.20% at the cost of an increased overhead of 22.51%. In case of the maximum considered error correction capacity and lowest code rate, BCH(63,45,3), the gain in PSNR is of 33.83% with an additional overhead of 40.74% in number of transmitted bits.

## D.5 Conclusions

In this work the problem of improving perceptual video quality in multimedia streaming over wireless sensor networks is analyzed. Using real video traces collected using IPER-MOBv2.0 board, and exploiting experimental loss traces gathered in a complex real-world scenario, we first evaluate the impact of IEEE802.15.4 packet losses on the received video quality. Presented results show as BER values higher than $10^{-4}$ lead to a quality reduction in terms of PSNR close to 45% with respect to the highest quality value experienced. Moreover, we propose a forward error correction strategy based on BCH codes and fully compliant with the IEEE802.15.4 standard by the definition of a new MAC data message specifying the bits kept reserved by the standard for different type of messages. Performance results as a function of the error correction capacity of the code and its code rate show how the proposed technique guarantees to improve the final PSNR. When the error correction capacity of the code is greater than the experienced burst length the performance improvement in terms of PSNR, with respect to a plain transmission (no protection applied), is equal to 17.20%. This performance improvement is reached with a minimum additional overhead of 22.51% in number of transmitted bits. Furthermore, when higher protection levels can be applied, video quality improvement equal to 33.83%, with an additional overhead of 40.74%, has been experienced.

# List of Publications

The work presented in this thesis has appeared in the articles reported below.

### Journal papers and Magazines

[J1] **D. Munaretto**, M. Zanforlin, M. Zorzi, "Online Path Selection: a Video Delivery Framework for Next Generation Cellular Networks", submitted to *IEEE Transactions on Multimedia*, Dec. 2013.

[J2] B. Fu, **D. Munaretto**, T. Melia, B. Sayadi, W. Kellerer, "Analyzing the Combination of Different Approaches for Video Transport Optimization for Next Generation Cellular Networks", *IEEE Network Magazine, Special Issue on Video over Mobile Networks*, Mar.-Apr. 2013.

[J3] T. Melia, **D. Munaretto**, L. Badia, M. Zorzi, "Online QoE Computation for Efficient Video Delivery over Cellular Networks", *IEEE COMSOC MMTC E-letter*, Mar. 2012.

[J4] **D. Munaretto**, D. Jurca, J. Widmer, "A Resource Allocation Framework for Scalable Video Broadcast in Cellular Networks", *Springer Mobile Networks and Applications*, 16 (6): pp 794-806, Dec. 2011.

### Conference papers

[C1] **D. Munaretto**, F. Giust, G. Kunzmann, M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks", accepted for publication in *IEEE ICC 2014*.

[C2] M. Zanforlin, **D. Munaretto**, A. Zanella, M. Zorzi, "SSIM-based video admission control and resource allocation algorithms", submitted to *IEEE WiOpt (WiVid) 2014*.

[C3] I. Ahmed, L. Badia, **D. Munaretto**, M. Zorzi, "Analysis of PHY/Application Cross-layer Optimization for Scalable Video Transmission in Cellular Networks", *IEEE WoWMoM*, June 2013.

[C4] **D. Munaretto**, M. Zanforlin, M. Zorzi. "Performance evaluation in ns-3 of a video delivery framework for next generation cellular networks", *IEEE ICC (IIMC)*, June 2013.

[C5] B. Tomasi, **D. Munaretto**, M. Zorzi. "Realtime redundancy allocation for time-varying underwater acoustic channels", *ACM WUWNet*, Nov. 2012.

[C6] **D. Munaretto**, T. Melia, S. Randriamasy, M. Zorzi. "Online path selection for video delivery over cellular networks", *IEEE Globecom (QoEMC)*, Dec. 2012.

[C7] R. Costa, T. Melia, **D. Munaretto**, M. Zorzi. "When Mobile Networks meet Content Delivery Networks: challenges and possibilities", *ACM MobiArch*, Aug. 2012.

[C8] **D. Munaretto**, M. Zorzi. "Robust opportunistic broadcast scheduling for scalable video streaming", *IEEE WCNC*, Apr. 2012.

[C9] T. Melia, S. Randriamasy, **D. Munaretto**, M. Zorzi. "QoE optimization with network layer awareness on hybrid wireless network", *Next Generation Service Delivery Platforms (NG SDP), GI/ITG Workshop*, Oct. 2011.

[C10] **D. Munaretto**. "Opportunistic Scheduling and Rate Adaptation for Scalable Broadcast Video Streaming", *IEEE WoWMoM 2011*, June 2011.

[C11] N. Amram, B. Fu, G. Kunzmann, T. Melia, **D. Munaretto**, M. Zorzi. "QoE-based Transport Optimization for Video Delivery over Next Generation Cellular Networks", *IEEE ISCC (MediaWiN) 2011*, June 2011.

[C12] M. Petracca, M. Ghibaudi, C. Salvadori, P. Pagano, **D. Munaretto**. "Performance Evaluation of FEC techniques based on BCH codes in Video Streaming over Wireless Sensor Networks", *IEEE ISCC (MediaWiN) 2011*, June 2011.

[C13] A. E. Essaili, E. Steinbach, **D. Munaretto**, S. Thakolsri, W. Kellerer. "QoE-driven resource optimization for user generated video content in next generation mobile networks", *IEEE ICIP 2011*, Sep. 2011.

# Bibliography


[1] "Medieval Home Page." [Online]. Available: http://www.ict-medieval.eu/

[2] D. Munaretto, D. Jurca, and J. Widmer, "A Resource Allocation Framework for Scalable Video Broadcast in Cellular Networks," *Springer Mobile Networks and Applications*, vol. 16, pp. 794 – 806, Dec. 2011.

[3] I. Ahmed, L. Badia, D. Munaretto, and M. Zorzi, "Analysis of PHY/Application Crosslayer Optimization for Scalable Video Transmission in Cellular Networks," in *IEEE WoWMoM 2013*, Madrid, Spain, June 2013.

[4] D. Munaretto and M. Zorzi, "Robust opportunistic broadcast scheduling for scalable video streaming ," in *IEEE WCNC 2012*, Paris, France, April 2012.

[5] D.Munaretto, "Opportunistic Scheduling and Rate Adaptation for Scalable Broadcast Video Streaming ," in *IEEE WoWMoM 2011*, Lucca, Italy, June 2011.

[6] D. Jurca, D. Munaretto, and J. Widmer, "Method and apparatus for scheduling packets," Patent 10 163 495.4, Oct 3, 2012.

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technology*, vol. 17, pp. 560–576, 2003.

[8] T. Melia, D. Munaretto, L. Badia, and M. Zorzi, "Online QoE Computation for Efficient Video Delivery over Cellular Networks," *IEEE COMSOC MMTC E-letter*, Mar. 2012.

[9] D. Munaretto, F. Giust, G. Kunzmann, and M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks," in *accepted at IEEE ICC 2014*, Sidney, Australia, June 2014.

[10] R. Costa, T. Melia, D. Munaretto, and M. Zorzi, "When Mobile Networks meet Content Delivery Networks: challenges and possibilities," in *ACM MobiArch, 2012* , Istanbul, Turkey, Aug. 2012.

[11] D. Munaretto, M. Zanforlin, and M. Zorzi, "Online Path Selection: a VideoDelivery Framework for Next Generation Cellular Networks," *submitted to IEEE Transactions on Multimedia*, 2013.

[12] B. Fu, D. Munaretto, T. Melia, B. Sayadi, and W. Kellerer, "Analyzing the Combination of Different Approaches for Video Transport Optimization for Next Generation Cellular Networks," *IEEE Network Magazine, special issue on video over mobile networks*, vol. 27, pp. 8 – 14, March-April 2013.

[13] B. Feitor, P. Assuncao, J. Soares, L. Cruz, and R. Marinheiro, "Objective quality prediction model for lost frames in 3D video over TS," in *IEEE ICC 2013*, Budapest, Hungary, June 2013.

[14] D. Munaretto, T. Melia, S. Randriamasy, and M. Zorzi, "Online path selection for video delivery over cellular networks," in *Proc. of QoEMC, IEEE Globecom 2012*, Anheim, CA, USA, Dec. 2012.

[15] T. Melia, S. Randriamasy, D. Munaretto, and M. Zorzi, "QoE optimization with network layer awareness on hybridwireless network ," in *Next Generation Service Delivery Platforms (NG SDP), GI/ITG Workshop* , Munich, Germany, Oct. 2011.

[16] Software-Defined Networking: The New Norm for Networks, ONF White Paper, Apr. 2012.

[17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 69 – 74, Apr. 2008.

[18] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms ," in *submitted to IEEE WiOpt (WiVid) 2014*, Sidney, Australia, June 2014.

[19] A. E. Essaili, E. Steinbach, D. Munaretto, S. Thakolsri, and W. Kellerer, "QoE-driven resource optimization for user generated video content in next generation mobile networks," in *IEEE ICIP 2011*, Bruxelles, Belgium, Sep. 2011.

[20] M. Petracca, M. Ghibaudi, C. Salvadori, P. Pagano, and D.Munaretto, "Performance Evaluation of FEC techniques based on BCH codes in Video Streaming over Wireless Sensor Networks ," in *IEEE ISCC (MediaWiN) 2011*, Corfu, Greece, June 2011.

[21] B. Tomasi, D.Munaretto, and M. Zorzi, "Realtime redundancy allocation for time-varying underwater acoustic channels," in *ACM WUWNet 2012*, Los Angeles, CA, Nov. 2012.

[22] "3GPP Specification Detail." [Online]. Available: http://www.3gpp.org

[23] "3GPP Specification Detail: LTE." [Online]. Available: http://www.3gpp.org/LTE

[24] "3GPP TR 25.913." [Online]. Available: http://www.3gpp.org/DynaReport/25913.htm

[25] P.Lescuyer and T.Lucidarme, *Evolved Packet System (EPS): The LTE and the SAE Evolution of 3G UMTS*. John Wiley & Sons Ltd, 2008.

[26] CISCO, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017*. White Paper, 2013.

[27] D. Munaretto, D. Jurca, and J. Widmer, "Broadcast Video Streaming in Cellular Networks: An Adaptation Framework for Channel, Video and AL-FEC Rates Allocation," in *WICON 2010*, Singapore, Mar. 2010.

[28] D. Liu, J. C. Zuniga, P. Seite, H. Chan, and C. J. Bernardos, "Distributed Mobility Management: Current practices and gap analysis," Internet-Draft (work in progress), draft-ietf-dmm-best-practices-gap-analysis-01.txt, June 2013.

[29] "DASH." [Online]. Available: http://www-itec.uni-klu.ac.at/dash/

[30] Akamai Technologies Inc., *Akamai Media Analytics*. White Paper, 2009.

[31] Y. Fakhri, B. Nsiri, D. Aboutajdine, and J. Vidal, "Throughput Optimization for Wireless OFDM System in Downlink Transmission Using Adaptive Techniques ," in *IEEE WiCOM 2006*, Wuhan City, China, Sep. 2006.

[32] "ALTO Status Pages." [Online]. Available: http://tools.ietf.org/wg/alto/

[33] "3GPP TS 23.401, 3GPP Specification Detail." [Online]. Available: http://www.3gpp.org

[34] J. Kim, T.-W. Um, W. Ryu, and B. S. Lee, "Heterogeneous Networks and Terminal-Aware QoS/QoE-Guaranteed Mobile IPTV Service," *IEEE Communications Magazine*, vol. 46, pp. 110 – 117, May 2008.

[35] D. Munaretto, D. Jurca, and J. Widmer, "A Fast Rate-Adaptation Algorithm for Robust Wireless Scalable Streaming Applications," in *IEEE WiMob 2009*, Marrakech, Morocco, Oct. 2009.

[36] M. Mehrjoo, M. Dianati, X. Shen, and K. Naik, "Opportunistic fair scheduling for the downlink of IEEE 802.16 wireless metropolitan area networks," in *ACM International conference on Quality of service in heterogeneous wired/wireless networks (QShine)*, Waterloo, Ontario, Canada, Aug. 2006.

[37] M. Dianati, R. Tafazolli, X. Shen, and K. Naik, "Call admission control with opportunistic scheduling scheme," *Wireless Journal on Communications and Mobile computing*, vol. 10, pp. 372–382, Mar. 2010.

[38] X. Liu, E. K. P. Chong, and N. B. Shroff, "A Framework for Opportunistic Scheduling in Wireless Networks," *The International Journal of Computer and Telecommunications Networking*, vol. 41, pp. 451 – 474, Jan. 2003.

[39] V. Vukadinovic and E. Drogou, "Opportunistic fair scheduling for the downlink of IEEE 802.16 wireless metropolitan area networks," in *IEEE International Conference on Communications (ICC)* , Glasgow, UK, June 2007.

[40] T.-P. Low, M.-O. Pun, and C.-C. Kuo, "Optimized Opportunistic Multicast Scheduling Over Cellular Networks," in *IEEE Globecom 2008*, Los Angeles, CA, Dec. 2008.

[41] T.-P. Low, M.-O. Pun, Y.-W. P. Hong, and C.-C. Kuo, "Optimized Opportunistic Multicast Scheduling (OMS) over Wireless Cellular Networks," in *Technical Report TR2010-008*, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, Mar. 2010.

[42] H. Hu, J. Liu, and J. Liang, "Downlink Scheduling for Multimedia Multicast/Broadcast over Mobile WiMAX: Connection-oriented Multi-State Adaptation," *IEEE Wireless Communications*, vol. 16, pp. 72–79, Aug. 2009.

[43] J. Kim, J. Cho, and H. Shin, "Resource Allocation for Scalable Video Broadcast in Wireless Cellular Networks," in *IEEE WiMob 2005*, Montreal, Canada, Aug. 2005.

[44] P. Pahalawatta, R. Berry, T. Pappas, and A. Katsaggelos, "Content-Aware Resource Allocation and Packet Scheduling for Video Transmission over Wireless Networks," *IEEE Journal on Selected Areas in Communication*, vol. 25, pp. 749–759, May 2007.

[45] "openCDN." [Online]. Available: http://labtel.ing.uniroma1.it/opencdn/

[46] "Darwin Streaming Server ." [Online]. Available: http://dss.macosforge.org/

[47] "Helix Universal Server ." [Online]. Available: http://www.realnetworks.com/helix/index.aspx

[48] D. Munaretto, M. Zanforlin, and M. Zorzi, "Performance evaluation in ns-3 of a video delivery framework for next generation cellular networks," in *IEEE ICC (IIMC) 2013*, Budapest, Hungary, June 2013.

[49] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP-Design Priciples and Standards," in *ACM MMSys*, New York, NY, Feb. 2011.

[50] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To Cache or Not to Cache: The 3G Case," *Internet Computing, IEEE*, vol. 15, no. 2, pp. 27–34, 2011.

[51] "IETF DMM Working Group." [Online]. Available: http://datatracker.ietf.org/wg/dmm/

[52] "Multi-Cost ALTO." [Online]. Available: http://tools.ietf.org/id/draft-randriamasy-alto-multi-cost-05.txt

[53] "Seagate." [Online]. Available: http://www.seagate.com/internal-hard-drives/enterprise-hard-drives

[54] Amazon Web Services, *How AWS Pricing Works*. White Paper, Dec. 2011.

[55] D. Boscovic, F. Vakil, S. Dautovic, and M. Toic, "Pervasive wireless CDN for greening video streaming to mobile devices," in *IEEE MIPRO, 2011* , Opatija, Croatia, May 2011.

[56] K. Hosanagar, R. Krishnan, M. Smith, and J. Chuang, "Optimal Pricing of Content Delivery Network (CDN) Services," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, vol. 7. Washington, DC, USA: IEEE Computer Society, Jan. 2004.

[57] "ns-3 Network Simulator." [Online]. Available: http://www.nsnam.org

[58] M. R.-E. N. Baldo, M. Miozzo and J. Nin-Guerrero, "An Open Source Product-Oriented LTE Network Simulator based on ns-3," in *Proc. of ACM MSWiM*, Miami Beach, FL, USA, November 2011.

[59] "LENA documentation." [Online]. Available: lena.cttc.es/manual

[60] K. Stuhlmller, N. Frber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Sel. Areas Commun.*, vol. 18, pp. 1012 – 1030, June 2000.

[61] S. Kahn, S. Duhovnikov, E. Steinbach, and W. Kellerer, "MOS-based multiuser multiapplication cross-layer optimization for mobile multimedia communication," *ACM Journal on Advances in Multimedia*, vol. 2007, pp. 1 – 11, Jan. 2007.

[62] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010 - 2015, White Paper, 2011.

[63] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-Driven Cache Management for HTTP Adaptive Bit Rate Streaming Over Wireless Networks," *IEEE Transactions on Multimedia*, vol. 15, pp. 1431 – 1445, Oct. 2013.

[64] G. Kandavanam, D. Botvich, and S. Balasubramaniam, "PaCRA: A Path-aware Content Replication Approach to support QoS guaranteed video on demand service in metropolitan IPTV networks," in *IEEE Network Operations and Management Symposium (NOMS)2010* , Osaka, Japan, June 2010.

[65] Z. Zhu, S. Li, and X. Chen, "Design QoS-Aware Multi-Path Provisioning Strategies for Efficient Cloud-Assisted SVC Video Streaming to Heterogeneous Clients," *IEEE Transactions on Multimedia*, vol. 15, pp. 758 – 768, June 2013.

[66] C. E. Palau, J. Mares, B. Molina, and M. Esteve, "Wireless CDN video streaming architecture for IPTV," *ACM Journal on Multimedia Tools and Applications*, vol. 53, pp. 591 – 613, July 2011.

[67] W. Kumwilaisak, Y. T. Hou, Q. Zhang, W. Zhu, J. Kuo, and Y.-K. Zhang, "A cross-layer quality-of-service mapping architecture for video delivery in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, pp. 1685 – 1698, Dec. 2003.

[68] Q. Zhang, W. Zhu, and Y. Q. Zhang, "End-to-End QoS for Video Delivery Over Wireless Internet," *Proceedings of the IEEE*, vol. 93, pp. 123 – 134, Jan. 2005.

[69] N. Amram, B. Fu, G. Kunzmann, T. Melia, D.Munaretto, and M. Zorzi, "QoE-based Transport Optimization for Video Delivery over Next Generation Cellular Networks ," in *IEEE ISCC (MediaWiN) 2011*, Corfu, Greece, June 2011.

[70] Z.Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600 – 612, Apr. 2004.

[71] "Advanced Video Coding for Generic Audiovisual Services," *ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC*.

[72] "Test media repository." [Online]. Available: http://media.xiph.org/video/derf/

[73] "Yuv qcif and cif video file." [Online]. Available: http://trace.eas.asu.edu/yuv/index.html

[74] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Workshop on Quality of Multimedia Experience (QoMEX)*, Trondheim, Norway, June 2010.

[75] "Joint scalable video model - reference software." [Online]. Available: http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

[76] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," in *IEEE ICME*, Baltimore, MD, July 2003.

[77] I. Spanou, A. Lazaris, and P. Koutsakis, "Scene change detection-based discrete autoregressive modeling for MPEG-4 video traffic," in *IEEE ICC 2013*, Budapest, Hungary, June 2013.

[78] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, pp. 58 – 78, Oct-Dec 2004.

[79] V. Singh, J. Ott, and I. Curcio, "Predictive Buffering for Streaming Video in 3G Networks," in *IEEE WoWMoM 2012*, San Francisco, CA, USA, June 2012.

[80] A. de la Oliva, T. Melia, A. Vidal, C. J. Bernardos, I. Soto, and A. Banchs, "IEEE 802.21 enabled mobile terminals for optimized WLAN/3G handovers: a case study," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, pp. 29 – 40, Apr. 2007.

[81] Z. Wang, A. C. Bovik, H.R.Sheikh, and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600 – 612, Apr. 2004.

[82] "VideoLAN." [Online]. Available: http://compression.ru/video/quality_measure/info_en.html#ssim

[83] C. N. Taylor and S. Dey, "Run-time allocation of buffer resources for maximizing video clip quality in a wireless last-hop system," in *IEEE ICC 2004*, Paris, Frane, June 2004.

[84] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, pp. 312 – 322, Apr. 2004.

[85] "Methodology for subjective assessment for television pictures." [Online]. Available: http://www.itu.int/rec/R-REC-BT.500/en

[86] "Methodology for subjective assessment for multimedia." [Online]. Available: http://www.itu.int/rec/T-REC-P.910/en

[87] "WHOI umodem." [Online]. Available: http://acomms.whoi.edu/umodem/

[88] "Evologics acoustic modems." [Online]. Available: http://www.evologics.de/en/products/acoustics/index.html

[89] W. Hodgkiss and J. Preisig, "Kauai Acomms MURI 2011 (KAM11)Experiment," in *Proc. 11th European Conference on Underwater Acoustics (ECUA 2012)*, 2012, pp. 993 – 1000.

[90] E. Uysal-Biyikoglu, B. Prabhakar, and A. E. Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 487 – 499, 2002.

[91] E. Uysal-Biyikoglu and A. E. Gamal, "On adaptive transmission for energy efficiency in wireless data networks," *IEEE Transactions on Information Theory*, vol. 50, pp. 3081 – 3094, 2004.

[92] M. Zafer and E. Modiano, "Optimal rate control for delay-constrained data transmission over a wireless channel," *IEEE Transactions on Information Theory*, vol. 54, pp. 4020 – 4039, 2008.

[93] J. Lee and N. Jindal, "Energy-efficient scheduling of delay constrained traffic over fading channels," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 1866 – 1875, 2009.

[94] R. Srivastava and C. Koksal, "Energy optimal transmission scheduling in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1550 – 1560, 2010.

[95] P. Casari, M. Rossi, and M. Zorzi, "Towards optimal broadcasting policies for HARQ based on Fountain codes in underwater networks," in *Proceedings of IEEE/IFIP WONS*, 2008.

[96] P. Casari and A. F. H. III, "Energy-efficient reliable broadcast in underwater acoustic networks," in *Proceedings of ACM WUWNet*, 2007.

[97] W. Zhang and U. Mitra, "A delay-reliability analysis for multihop underwater acoustic communication," in *Proceedings of ACM WUWNet*, 2007.

[98] Z. Haojie, T. Hwee-Pink, A. Valera, and B. Zijian, "Opportunistic ARQ with bidirectional overhearing for reliable multihop underwater networking," in *Proceedings of IEEE OCEANS*, 2010.

[99] Q. Fengzhong and L. Yang, "Rate and reliability oriented underwater acoustic communication schemes," in *Proceedings of IEEE DSP/SPE*, 2009.

[100] U. Erez and G. Wornell, "A super-Nyquist architecture for reliable underwater acoustic communication," in *Proceedings of Conference on Communication, Control, and Computing (Allerton)*, 2011, pp. 469 – 476.

[101] N. Bonello, Z. Rong, C. Sheng, and L. Hanzo, "Reconfigurable rateless codes," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 5592 – 5600, 2009.

[102] L. Badia, M. Mastrogiovanni, C. Petrioli, S. Stefanakos, and M. Zorzi, "An optimization framework for joint sensor deployment, link scheduling and routing in underwater sensor networks," in *Proceedings of ACM WUWNet*, 2006.

[103] K. Kredo, P. Djukic, and P. Mohapatra, "STUMP: Exploiting Position Diversity in the Staggered TDMA Underwater MAC Protocol," in *Proceedings of IEEE INFOCOM*, 2009.

[104] Y. Polyanskiy, "Channel coding: non-asymptotic fundamental limits," in *Princeton Univ.*, Princeton, NJ, USA, 2010.

[105] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, pp. 2307 – 2359, 2010.

[106] K. Martinez, R. Ong, and J. Hart, "Glacsweb: a Sensor Network for Hostile Environments," in *Proc. of the IEEE Sensor and Ad Hoc Communications and Networks Conference*, October 2004, pp. 81–87.

[107] R. Lee, K. Chen, S. Chiang, C. Lai, H. Liu, and M. Wei, "A Backup Routing with Wireless Sensor Network for Bridge Monitoring System," in *Proc. of the Communication Networks and Services Research Conference*, May 2006, pp. 161–165.

[108] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "PermaSense: Investigating Permafrost with a WSN in the Swiss Alps," in *Proc. of the Fourth Workshop on Embedded Networked Sensors*, June 2007, pp. 8–12.

[109] S. Feller, Y. Zheng, E. Cull, and D. Brady, "Tracking and imaging humans on heterogeneous infrared sensor arrays for law enforcement applications," in *Proc. SPIE Aerosense*, 2002, pp. 212–221.

[110] I. Bekmezci and F. Alagoz, "New TDMA based sensor network for military monitoring (MIL-MON)," in *Proc. IEEE Military Communications Conference*, October 2005, pp. 2238–2243.

[111] I. Akyildiz, T. Melodia, and K. Chowdury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks (Elsevier)*, vol. 51, no. 4, pp. 921–960, March 2007.

[112] M. Petracca, G. Litovsky, A. Rinotti, M. Tacca, J. D. Martin, and A. Fumagalli, "Perceptual based Voice Multi-Hop Transmission over Wireless Sensor Networks," in *Proc. IEEE Symposium on Computers and Communications*, July 2009, pp. 19–24.

[113] Y. Jiang, X. Yao, W. Wang, and L. Gu, "New Method for Weighted Coverage Optimization of Occlusion-Free Surveillance in Wireless Multimedia Sensor Network," in *Proc. IEEE International Conference on Networking and Distributed Computing*, October 2010, pp. 21–24.

[114] "IPERMOB:A Pervasive and Heterogeneous Infrastructure to control Urban Mobility in Real-Time," http://www.ipermob.org, July 2009.

[115] I. C. Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPAN)," *The Institute of Electrical and Electronics Engineers, Inc.*, October 2003.

[116] E. Culurciello, J. Park, and A. Savvides, "Address-Event Video Streaming over Wireless Sensor Networks," in *Proc. IEEE International Symposium on Circuits and Systems*, June 2007, pp. 849–852.

[117] W. Wang, M. Hempel, D. Peng, H., H. Sharif, and H. Chen, "An Energy Efficient Encryption for Video Streaming in Wireless Sensor Networks," *IEEE Transactions on Multimedia*, vol. 12, no. 5, pp. 417–426, August 2010.

[118] T. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley, 2005.

[119] P. Gai, E. Bini, G. Lipari, M. D. Natale, and L. Abeni, "Architecture For A Portable Open Source Real Time Kernel Environment," in *Real-Time Linux Workshop and Hand's on Real-Time Linux Tutorial*, November 2000.

[120] "The Erika Enterprise Real-time Operating System," http://erika.tuxfamily.org.

[121] "ETSI EN 300 328 V1.7.1 (2006-10)," www.etsi.org.

[122] K. Shuaib, M. Alnuaimi, M. Boulmalf, I. Jawhar, F. Sallabi, and A. Lakas, "Performance Evaluation of IEEE 802.15.4: Experimental and Simulation Results," *Journal of Communications*, vol. 2, no. 4, pp. 29–37, June 2007.

I want to thank Prof. Michele Zorzi for the opportunity he gave me and to make me feel part of his group. Special thanks for the great atmosphere and the unforgettable moments together go to my friends and colleagues Marco (x2), Francesco, Irene, Davide, Giorgio, Federico, Angela, Giulio, Leonardo, Michele, Andrea, Chiara, Beatrice, Luca, Nicolo'.

*Grazie*