MEDIEVAL
INFSO-ICT-258053

MEDIEVAL

Deliverable D5.2

Final specification for transport optimisation components and interfaces

Editor:	Daniele Munaretto (CFR)
Deliverable nature:	Public
Due date:	June 30, 2012
Delivery date:	June 30, 2012
Version:	1.0
Total number of pages:	58
Reviewed by:	Michelle Wetterwald (EURECOM)
Keywords:	MEDIEVAL, Quality-of-Experience, transport optimisation, cross-layer optimisation

Abstract

Deliverable D5.2 presents the final specification for transport optimisation components and interfaces and focuses on the scientific key research results and advancements made in Task 5.2 since the delivery of the previous deliverable D5.1. This deliverable provides the detailed specifications for video transport optimisation (Task 5.2), taking into account the modules and interfaces involved in the design of the optimisation algorithms. Its objective is to present in details the algorithms which ensure the efficient transport of the video flow through the transport network. All components in this subsystem interact with upper and lower layers in order to provide a cross-layer optimisation and trigger traffic engineering and content adaptation in different layers based on the current condition of the network.

List of authors

Company	Author
ALBLF	Bessem Sayadi
CFR	Daniele Munaretto
DOCOMO	Gerald Kunzmann, Bo Fu
IMDEA Networks	Joerg Widmer

Executive Summary

The purpose of deliverable D5.2 is to provide the final specification for transport optimisation components and interfaces, focusing on Task 5.2. In this document we present the ongoing research activities and key results since the delivery of deliverable D5.1 [3]. D5.2 provides the detailed specifications of the video transport optimisation taking into account the modules and interfaces involved in the design of the optimisation algorithms.

The scientific key contributions are described in this deliverable along with the corresponding scientific results in terms of publications and contributions to standardization fora. Moreover, the ongoing implementation and simulation efforts are described in detail.

A first optimisation scheme, namely PathSelection, optimally selects the video paths based on multiple network metrics (core and access networks). A second algorithm, called TrafficManagement, efficiently allocates the data rates of multiple video streams. A third algorithm, SVCFiltering, dynamically filters video packets on a video frame level, the finest granularity that could react to the fast variations of wireless channel conditions. They adapt the video transmission with a common objective: the QoE is maximized while the network resources are efficiently utilized. A fourth algorithm, called FECAdaptation, can optimise for reliability independently without interworking with the other three schemes.

One of the key components characterizing the MEDIEVAL Transport Optimisation subsystem is the optimisation of the video delivery chain. The target of our research work is to address transport layer issues for mobile video and integrate the improvements achieved with our novel algorithms with other techniques implemented throughout the protocol stack. Thereby, a decentralized cross-layer optimisation aims at solving congestion events in the mobile network by means of various traffic engineering methods, as well as providing an optimised Quality-of-Experience (QoE) for all users. In the present document we outline the scenarios and motivations for our optimisation, and detail proposed solutions. Finally, simulation results are presented to support our advancements.

Table of Contents

List of Figures

Abbreviations

3GPP	3rd Generation Partnership Project
ALTO	Application-Layer Traffic Optimisation
AL-FEC	Application Layer – Forward Error Correction
APM	Application Performance Metric
ARQ	Automatic Repeat request
AS	Application Server
AVC	Advanced Video Codec (H.264)
BL	Base Layer (H.264/SVC Codec)
BS	Base Station
BBSE	Basic Bit Stream Extractor
CDN	Content Delivery Network (component / module)
CGS	Coarse Grain Scalability
CN	Core Network
CNM	Core Network Monitoring (module)
CRC	Connection Relay and Cache
DM	Decision Manager (component / module)
eNB	eNodeB
E2E	End-to-End
EL	Enhancement Layer (H.264/SVC)
EPC	Evolved Packet Core
FEC	Forward Error Correction
FM	Flow Manager (component / module)
GoP	Group of Pictures
HARQ	Hybrid ARQ
JND	Just Noticeable Difference
JSVM	Joint Scalable Video Model
L25	L2.5 abstraction module (Wireless Access subsystem)
LTE	Long Term Evolution
MAC	Media Access Control
MCS	Modulation Coding Scheme
MDP	Markov Decision Process
MEDIEVAL	MultimEDia transport for mobIlE Video AppLications
MGS	Medium Grain Scalability
MOS	Mean Opinion Score
MSC	Message Sequence Chart
MSE	Mean Squared Error

OAI	Open Air Interface
PHY	Physical layer (of the OSI model)
PoA	Point of Attachment
PSNR	Peak Signal-to-Noise Ratio
QoE	Quality of Experience
QoEVC	QoE & Video Control module (Video Services subsystem)
QoS	Quality of Service
RAN	Radio Access Network
RL	Reinforcement Learning
SNR	Signal to Noise Ratio
SSIM	Structure SIMilarity index
SVC	Scalable Video Codec (H.264)
TE	Traffic Engineering (module)
TO	Transport Optimisation (subsystem / component)
UE	User equipment
UEP	Unequal Error Protection
UPCON	User Plane Congestion Management
VoD	Video on Demand
WAN	Wireless Access Network
XLO	Cross-Layer Optimisation (module)

1 Introduction

This deliverable aims to present the final specification for transport optimisation components and interfaces, focusing on the ongoing research activities in Task 5.2 of WP5, starting from the optimisation algorithms and cross-layer mechanisms presented in deliverable D5.1 [3].

In deliverable D5.1 [3] the initial MEDIEVAL Transport Optimisation architecture was presented. It supplements the general description of the MEDIEVAL system given in deliverable D1.1 [2] with detailed description of the Transport Optimisation architecture, technologies, new functionalities and internal interfaces between the different modules in this subsystem. When designing the subsystem, we envisioned a novel dynamic transport architecture for next generation mobile networks that is adapted to video service requirements. Our approach is to follow a QoE-oriented and cross-layer enabled redesign of networking mechanisms as well as the integration of Content Delivery Network (CDN) techniques in order to optimise the transport of the video inside the mobile core network.

This deliverable focuses on the specification of the modules and internal interfaces of the MEDIEVAL Transport Optimisation subsystem and on the advances of the research in Task 5.2. The specification work related to the external interfaces between the Transport Optimisation subsystem and the other MEDIEVAL subsystems (Video Services, Wireless Access, Mobility) within the overall MEDIEVAL architecture is reported in D1.3 [27]. The progress on the advanced CDN mechanisms (Task 5.1) is reported in deliverable D5.3 [34].

The structure of the deliverable is the following:

- in Chapter 2 we summarize the key achievements for the scientific work performed in the period between June 2011 and June 2012 and present our dissemination activities (resulting publications and standardization work);
- in Chapter 3 we report on the scientific advancements, including the detailed design of the cross-layer algorithms in Task 5.2 to address different issues of the video delivery chain at different levels of the mobile network; the main areas are the QoE-based traffic management, QoE-based scalable video layer filtering based on MAC buffer management, QoE-based optimisation with CDN and network layer awareness on hybrid wireless network, and FEC rate-adaptation;
- in Chapter 4 we present in detail the module and interface specifications. In this chapter the connection between the modules and interfaces and the algorithms presented in Chapter 3 is discussed;
- in Chapter 5 we report on the status of our evaluation work, i.e., on the simulation tools we use to validate our research work at different levels of the video delivery chain. Moreover, in this chapter we present our preliminary simulation results and our future evaluation plans.
- the deliverable is summarized and conclusions are drawn in Chapter 6.

2 Key contributions

This chapter presents the key contributions of Task 5.2 since the submission of the Deliverable 5.1 [3] in June 2011. We highlight the main achievements in terms of scientific work and also provide a brief description of the dissemination activities.

The key research results will be presented in detail in Chapter 3, where we will describe the optimisation algorithms designed to address the transport optimisation issues at different levels of the mobile operator network. Each of the designed algorithms aims at providing a solution to a specific issue of the video delivery chain, from the video source, e.g., video cache, to the mobile terminal at the end user. A first combination of the algorithms will be discussed.

The main key research contributions and dissemination activities are as follows:

- Concerning the QoE-based traffic management, a study item for User Plane Congestion Management (UPCON) was started in SA WG1 3GPP in November 2011 [Annex B]. We are contributing to this study item to identify requirements for handling user plane traffic when RAN congestion occurs. Use cases were proposed and being studied in SA1 meetings. We presented the MEDIEVAL project and highlighted the QoE-based optimisation in the ITG 5.2.4 Workshop on Traffic Management for Mobile Networks in March 2012. We extended our QoE-based cross layer optimisation to consider temporal QoE fluctuations [10]. Due to the variations of the network environments, users may experience the video quality with fluctuations which produce negative effects on QoE. We introduced a new parameter in order to smooth the temporal QoE fluctuations. To overcome the drastic quality fluctuation due to handovers, we integrated resource reservation based on mobility prediction to the QoE-based optimisation. We submitted the solution to IEEE Globecom 2012 [40].
- We are working on a control process for scalable video delivery over a wireless channel that maximizes the average received video quality according to the channel variation. A cross-layer control mechanism is proposed to control the queues level at the medium access control (MAC) layer in the PoA and at the application layer in the proxy (or in the server depending on the scenario) to efficiently estimate the quality of the decoded frames. The idea of adapting the source rate based only on the observation of the fullness of the last buffer in the stream path (MAC buffer) was published [16] and filed as a European patent [17]. The proposed algorithm has been submitted to IEEE Globecom 2012.
- We are working on two key objectives with respect to QoE-based optimisation with network layer awareness on hybrid (LTE and WiFi) wireless network: quasi-real time evaluation of the perceived QoE by end users and application requirements in terms of network resource usage. We propose to exploit the ALTO [22] protocol and its extensions for the mobile core and to evaluate key metrics (QoE-based video metrics and network metrics) in simulations and live experiments. The idea of designing this QoE-based online optimisation algorithm was published and presented at the 5th Workshop (Fachgespraech) NG SDP on Oct. 11, 2011, in Munich, Germany [8]. A first design of a QoE-online evaluation tool at the mobile terminal was submitted to the IEEE MMTC E-letter (published in March 2012), [21]. We further investigate how video applications can be served through several video delivery paths, resulting in different video quality experienced by the end user. Thus, we designed an optimisation framework which runs online algorithms to select the best path for video delivery based on a set of performance metrics evaluated for a range of settings by means of simulation. This optimisation work was submitted to IEEE Globecom 2012 [23] and the architectural design to ACM MobiArch 2012 [24]. We foresee a journal paper to finalize the architectural and algorithmic design by the end of the year or beginning of 2013.
- With respect to FEC rate-adaptation mechanisms, we designed a fast decision algorithm which is able to compute the amount of error protection needed by the video transmission based on feedback about the channel conditions. While the optimal solution to this problem requires a full search through the complete solution space, our heuristic approach takes a much faster decision based on a careful look at the properties of the scalable encoded bit-stream, and of the unequal error protection

mechanism. The FEC rate-adaptation framework is detailed in [4]. A first design of a robust opportunistic scheduler was presented at IEEE WoWMoM 2011 [7]. The complete version of the design and the framework evaluation was presented at IEEE WCNC 2012 [9]. We plan to collect all these components in a journal paper later in 2013.

- We designed a first joint optimisation framework, i.e., a combination of some of the optimisation algorithms of this deliverable. The benefits provided by the different algorithms will be studied with respect to several performance metrics, such as the video quality perceived at the end user side and delay. Such combination makes it possible to obtain a multi-level optimisation framework. We plan to submit a first paper on the combined algorithm to a journal/magazine by the end of the year.

3 Scientific Work

The purpose of this chapter is to present the updates in terms of scientific work introduced in deliverable D5.1 [3]. The optimisation algorithms designed in task T5.2 are presented in this section, while the mapping to modules and interfaces and simulation results are shown in Chapter 4 and Chapter 5, respectively.

In the following sections we present four algorithms which are designed to address the transport optimisation problems at different levels of the mobile operator network. Each algorithm aims at providing a solution to a specific issue of the video delivery chain, from the video source, e.g., video cache, to the mobile terminal.

In Section 3.1, an algorithm for the core network is designed to find the optimal QoE resource allocation among multiple end users by considering their video sensitivities and the data rate. The algorithm is applied to maximize the overall QoE and smooth the QoE fluctuation.

In Section 3.2, an algorithm to optimise the QoE of a video flow according to the channel variation is proposed. The algorithm exploits the dynamic characteristics of the encoded videos as well as the channel state. The goal is to design an optimal filtering policy for the different quality layers in case of scalable videos. We formulate the wireless video transmission problem as a Markov Decision Process (MDP) that explicitly considers the cooperation at the application layer and the MAC layer, the heterogeneity of the video data, and the varying network conditions. Learning techniques which allow an online update of the filtering decision according to the changing system characteristics are also considered. The impact of using a delayed channel state information is analyzed comparing to the case where the channel state information is not know. In addition, an online estimation of the decoded video quality at each time is performed corresponding to the reward function to be maximized by the MDP framework.

In Section 3.3, an online QoE-based computation algorithm that bridges CDN and wireless oriented metrics is presented. This work represents a common effort of Tasks 5.1 and 5.2 to compute the best video path available in the network (from a CDN cache down to the wireless access). We give the freedom to the operators to place an instance of the algorithm in the TO subsystem (Task 5.2), whenever CDN and wireless metrics should be jointly taken into account, or in the CDN subsystem (Task 5.1), if the only core network metrics are required to decide which video path to go for, i.e., when it is needed to update the CDN source.

In Section 3.4, a robust rate allocation algorithm via error correction is designed for video streaming applications, that is placed in the core and as well at the edges of the network.

Finally, in Section 3.5, we present a first integration of the aforementioned optimisation algorithms in the Transport Optimisation (TO) component. When handling different issues, the above optimisation algorithms are able to overcome their limitations by interworking with each other. We show how the optimisations can be combined to work in a common framework in a distributed way. We believe that the video transport can benefit from this multi-level optimisation framework that addresses most of the critical issues in the network, i.e., congestions, user mobility and wireless channel variations.

3.1 QoE-based traffic management (TrafficManagement)

Video services are rapidly growing with the advances of mobile communications technology and the capacity of mobile terminals, which challenging the networks with constraint resources. Although LTE systems provide higher data rates, in loaded cells congestion may still occur due to the resource-demanding video services. Thus, user satisfaction of video services is difficult to be maintained with resource constraints.

Another challenge is the variations of the wireless channel conditions. With the mobility of users, the wireless channel can change drastically. Consequently, video quality fluctuates, thereby causing undesirably negative evaluation of the service from the user perspective.

To keep a satisfactory and smooth perception of service experience by the users, we exploit information of video applications within a properly designed cross-layer framework to efficiently handle multimedia traffic. Video applications have different characteristics. Under congestion, the traffic needs to be managed

intelligently to maximize user satisfaction. By taking into account the video characteristics, the traffic is shaped in the network under the constraint of wireless resource availability. Furthermore, the fluctuation of the video quality also needs to be controlled to avoid perceivable quality degradation. We performed subjective tests to reveal the noticeable threshold of quality fluctuation by human beings. By managing the traffic, the video quality fluctuation is smoothened to maintain high user satisfaction.

The QoE-based optimisation scheme

The scheme aims at adapting the video streams of multiple users in order to optimise their overall QoE. The optimisation module in the network runs the optimisation algorithm periodically to calculate the optimal data rate for each user within its scope. The resulting target data rates are indicated to the traffic engineering module in the network to shape the traffic that goes through the wireless links per traffic engineering schemes, which are typically transcoding and packet dropping.

The optimisation is performed periodically, e.g., every second. In each period, the optimisation module retrieves the information of video sensitivities from the applications. It also retrieves the average channel conditions from the eNodeB. Both types of information are required by the optimisation module to understand the impact on QoE and the availability of resources in order to allocate an optimal data rate for each user in its scope. Figure 1 depicts the cross-layer framework with involved components and cross-layer information.

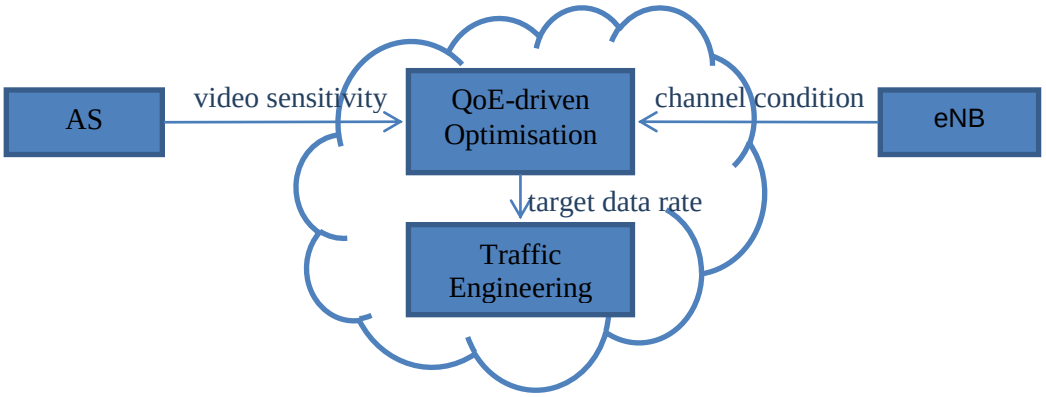


Figure 1: QoE-based traffic management

Formulating and solving the optimisation problem

The following algorithm finds the QoE-optimised resource allocation among multiple users by considering their video sensitivities and channel qualities.

Models

For each video streaming application, there are two models used to link the relation between the resource given to the application and the estimated QoE of the application. The QoE levels are represented by the Mean Opinion Score (MOS). It gives a score ranging from 1 (bad) to 5 (excellent) about the perceived quality.

- Utility function

The utility function describes the relation between the data rates and the estimated MOS values.

$$U = f(R)$$

where R is the data rate and U is the utility which is directly the MOS value. The utility function is derived per video per traffic engineering scheme. A source video is assumed to have the maximum quality in its original data rate. Traffic engineering schemes adapt this data rate, for instance by transcoding or packet

dropping. The correspondent QoE in different data rates is estimated by objective video quality metrics, e.g., SSIM. Examples can be found in [30].

- Radio link layer model

$$R = g(Q, x)$$

where R is the data rate, Q is the channel quality, e.g., CQI values, and x is the allocated resources, e.g., physical resource blocks. The radio link layer model gives the data rate to be achieved given the channel quality and the allocated resource.

Resource allocation vector

The algorithm searches for the optimal resource allocation vector which maximizes the objective, e.g., average QoE.

- Objective function and the optimal resource allocation vector are given by

$$F(x) = \frac{1}{K} \cdot \sum_{k=1}^{K} U_k(x_k)$$

$$x_{opt} = \arg\max F(x)$$

where K is the number of users inside the optimisation scope, x_{opt} is optimal resource allocation among the users. The objective function aims to maximize the overall QoE of the users in the optimisation scope.

The greedy search is applied to find the optimal resource allocation in a fast manner.

Simplified algorithm: a greedy search algorithm of optimal resource allocation

Input: Utility function U, number of user K, channel qualities Q, resource allocation step size Δx, maximum number of iterations $I_{\max}$.

Output: Optimal resource allocation x_{opt}.

Initialization: initial equal resource allocation x, iteration index $I = 0$.

for $k = 1$ to K do

Get $R_k = g(Q_k, x_k)$;

Compute $U_k = f(R_k)$;

end for

loop

for $k = 1$ to K do

increase and decrease resource for k: $x_k + \Delta x, x_k - \Delta x$;

compute $\Delta U_{k,inc}, \Delta U_{k,dec}$;

end for

$k_{inc} = \arg\max_k \Delta U_{k,inc}$; // the utility increases the most by getting Δx

$k_{dec} = \arg\max_k \Delta U_{k,dec}$; // the utility decreases the least by releasing Δx

$x_{k_{inc}} = x_{k_{inc}} + \Delta x$; // set resources

$x_{k_{dec}} = x_{k_{dec}} - \Delta x$; // set resources

$I{+}{+}$;

if $I > I_{\max}$ **then**

break;

end if
end loop
output: x_{opt}

Improving fluctuation of QoE during handovers

As the network conditions as well as the user location are changing over time, the video transmission is subject to fluctuations and the video quality perceived by users varies within each video session. Handovers are a critical challenge as they introduce strong fluctuations to the perceived video quality. During handovers, the bandwidth for a video session could change dramatically depending on the congestion status of the source cell and the target cell. The quality of the video could drop greatly if the user moves from an idle cell to a busy cell. Moreover, handovers can also affect the other existing users in the cell. For example, when a user accessing a high rate video streaming is handing over from a lightly loaded cell to a congested cell, the existing users in the congested cell will be affected due to the resource limitations.

Humans are able to recognize a video quality change if it exceeds a specific threshold. To decide the value of this threshold, a subjective test with 30 persons was done in [10], by applying the Just Noticeable Difference (JND) concept. Two video sequences were used in the test: 'Mother and Daughter' and 'Foreman', with static and dynamic video scenes, respectively. The results of the subjective test showed that the JND for 'Mother and Daughter' and for 'Foreman' is 0.21 MOS and 0.26 MOS respectively. In the following work the average value (0.23 MOS) was used as threshold value for all video sequences.

To include the QoE fluctuation into the consideration in the optimisation problem, a penalty parameter is introduced to the objective function:

$$F(x) = \frac{1}{K} \cdot \left(\sum_{k=1}^{K} U_k\left(x_k\right) - \beta \sum_{k=1}^{K} (\varepsilon_k - \varepsilon_{th})\right)$$

$$\beta = \begin{cases} 0, \text{if } \varepsilon_k < \varepsilon_{th} \\ 1, \text{if } \varepsilon_k \geq \varepsilon_{th} \end{cases}$$

where ε_{th} is the lower bound threshold of a perceivable quality fluctuation, it is set to be the MOS value of 0.23, ε_k is the fluctuation of MOS the user experiences. β is a weighting factor used for giving priority for the smoothness of temporal video quality. By introducing this penalty parameter, which negatively affects the overall perceived quality, if the temporal change of video quality exceeds the threshold ($\varepsilon_k \geq \varepsilon_{th}$), the overall perceived quality is considered to be degraded. The scheme is denoted as Temporal Quality Smoothness Maximization (TQSM).
To overcome QoE fluctuation due to resource shortage during handovers, a resource reservation scheme is applied. Users with mobile terminals are likely to watch videos on their way outside. Mobility prediction technologies provide the possibility to get the knowledge of a handover a short time before it happens [41]. In most cases they are watching videos in transporters, e.g., cars and trains, along the roads, which make the mobility prediction more realistic [42]. With the availability of user mobility prediction, the target cell is informed of an oncoming handover. It evaluates how much resources the handover user requires to keep the QoE degradation within the perceivable threshold. If the required resources are not available, it starts to gradually reserve resources for the oncoming user. The resource reservation is performed step by step without making the existing user to perceive the reduction. The scheme is denoted as Dynamic Unperceivable HandOver (DUHO).

Scalability
For a scalability study, we analysed signalling overhead. From our analysis, the algorithm is scalable in a worst case study. Details of the analysis will be presented in deliverable D1.3 [27].
Future work
We are applying SVC for a more flexible rate shaping tool. We are focusing on the temporal QoE aspects in order to present to the users a satisfactory experience of a whole video session. Besides our current work to

improve the instantaneous QoE during a video session, we are considering the final QoE upon the end of a video session, when the users would have an overall impression of the whole session.

3.2 QoE-based video scalable layer filtering process based on MAC buffer management (SVCFiltering)

The video content is assumed encoded by using the quality or SNR (Signal to Noise Ratio) scalability of a H.264/SVC video which allows a frame to be coded with identical frame sizes for Base Layer (BL) and Enhancement Layer (EL), but providing a graceful degradation of quality among SNR layers at the same spatial/temporal resolution. The SNR scalability offers various granularities for different applications: Coarse-Grain quality Scalability (CGS), which supports bit rate adaptation at the level of SNR layers, and Medium-Grain quality Scalability (MGS), which supports bit rate adaptation at the frame level. With the MGS scalability, any EL can be discarded from a quality scalable bit stream. This scheme is considered in our study where we propose an online cross-layer control mechanism to maximize the quality of the received video while accounting for the variations of the characteristics of the video content and of the channel.
This control problem is addressed with Markov Decision Processes (MDP). The optimal actions to apply to the system are learned via Reinforcement Learning (RL). For that purpose, the quality of the decoded frames at receiver has to be inferred via an observation *(i)* of the quality of the various scalability layers and *(ii)* of the level of queues at the Application layer of the proxy (or the server depending on the scenario) and Medium Access Control (MAC) layer of the PoA.

Our goal is to design an optimal scheduling policy of quality scalable layers.

Figure 2 depicts a synoptic scheme for the transmission of SVC stream over wireless channel.

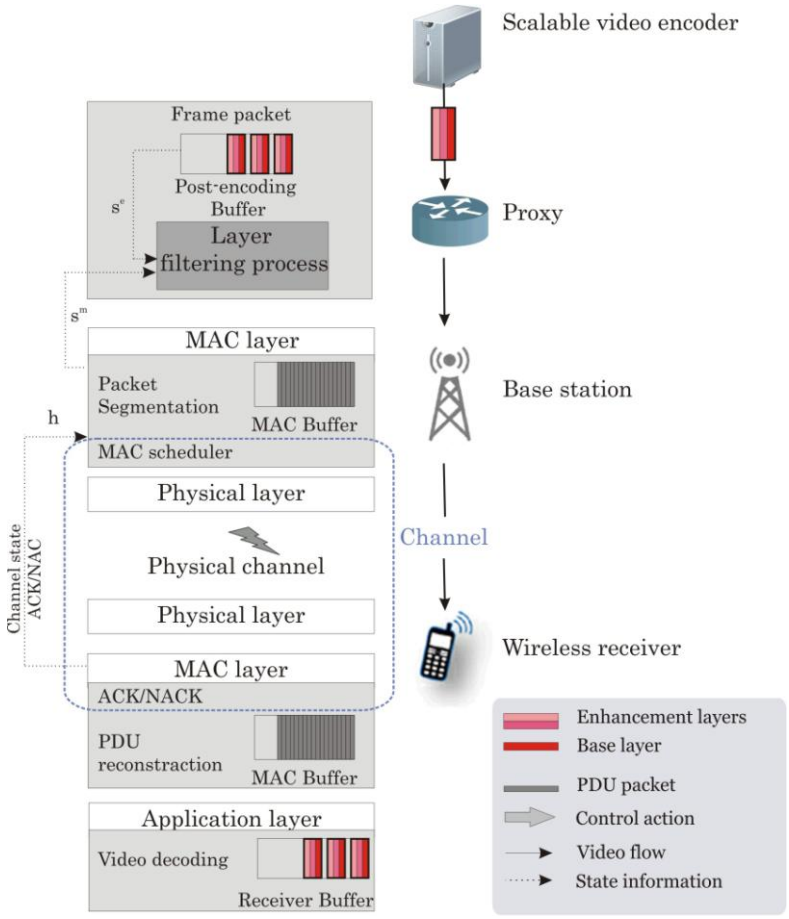


Figure 2: SVC stream transmission over wireless channel, case of LTE network

The mapping of each block to the MEDIEVAL architecture is detailed below.

Streaming server (Video Services side)

The video sequence is segmented into frames and encoded into L layers: a base layer and a set of L − 1 enhancement layers. Frames are generated with a constant period of time T and are identified by their temporal index t. The encoding parameters (quantization step, frame rate, etc.) are controlled by the streaming server.

Proxy (WP5 side, XLO module)

The L SNR layers are packetized by the streaming server and fed to the post-encoder buffer. The proxy is assumed not to control the encoding parameters. At each time, the controller performs *layer filtering* within the proxy so that for each frame, SNR layers may be sent, kept, or dropped. The layer filtering process, performed by the proxy, should be designed to maximize the video quality at receiver side. To perform this task, the decoded video quality is maximized by taking into account most factors impacting directly or indirectly the received video quality: frame type, number of SNR layers, error concealment, and packet loss due to post-encoder and MAC buffer overflow. The post-encoder is controlled in order to limit the delay for delay sensitive streams. Otherwise, this control point could be deactivated.

Base station and channel (Wireless Access side)

The base station contains additional buffers for each user connected to it to perform the rate and bandwidth allocation among users. Packets transmitted by the layer filtering are fed to the MAC buffer of the base station after being segmented into Packet Data Units (PDUs) of constant size. One has to control the MAC buffer in order to avoid overflow state to prevent PDUs from being dropped. PDUs are then transmitted to the mobile receiver via a wireless channel characterized by time-varying conditions.

If the channel state is considered in the filtering control, the control process should exploit some feedback from the MAC buffer and from the mobile client to estimate the channel conditions. Without channel state observation, the control has to rely on the observation of the level of the MAC buffer only.

Three hypotheses concerning the knowledge of the state of the channel are considered.

- *Hyp. 1*: *instantaneously available channel state*, where $s^h{}_t$ is assumed available when choosing the action to apply between time t and t+1; this is realistic only when feedback with very short delay is possible.
- *Hyp.2*: *Delayed channel state*, where the channel state is assumed available after a delay δ when choosing the action to apply between t and t+1; this represents a more realistic situation.
- *Hyp.3*: *unknown channel state* which is a scenario where no channel state feedback is considered.

User equipment (Wireless Access side)

The mobile receiver stores correctly received PDUs in its own MAC buffer. Packet de-encapsulation and buffering in the buffer at application layer in the proxy is done as soon as all corresponding PDUs have been received. Complete or incomplete frames are then processed by the video decoder. Outdated packets are dropped, without being decoded. Some packet-loss concealment may be put at work at the receiver side.

The mobility issue of the terminal is captured by the proposed algorithm since we are monitoring the last buffer in the stream path.

Model description

MDP formulation

The problem of designing an optimal scheduling policy of L SNR scalable layers over a wireless channel is translated in the framework of discrete time Markov Decision Process (MDP).

We formulate the stochastic optimisation problem as an MDP and solve it online using reinforcement learning. The advantages of the online method are that it does not require a priori knowledge of the traffic arrival and channel statistics to determine the scheduling policies. To solve our proposed optimisation problem, all components of the tuple (S, A, P, r) have to be identified.

We assume that time is slotted into discrete-time intervals of length T such that the t-th time slot is defined as the time interval [tT, (t + 1)T[. T can be set at the frame level corresponding to the cadence of the encoder or at the PDU level corresponding to cadence of the scheduler at the MAC layer. Filtering decisions are made at the beginning of each interval and the system state is assumed to be constant throughout each interval.

System states

The set of states of the systems are: s^h is the channel state, s^I is the frame type state, s^e is the post-encoding buffer state, and s^m is the MAC buffer state. The vector gathering all state is $\mathbf{S} = \left(s^h, s^I, s^e, s^m\right) \in S$. The details of the states are explained further below.

- **Channel state**

The channel state s_t^h describes the channel conditions (rate, probability of error, capacity, *etc.*) assumed constant between t and t + 1. The varying channel rate is modelled here as an N_h-state Markov chain. At time t, the state s_t^h with values in the set $H = \{1, \ldots, N_h\}$ represents a rate within the set $R^c = \left\{R_c^0, \ldots, R_c^{N_h}\right\}$ expressed in bit/s.

State transition probability $p_{k,l} = p\left(s_t^h = l \mid s_{t-1}^h = k\right)$ from moving from state $k \in \{1, \ldots, N_h\}$ to state $l \in \{1, \ldots, N_h\}$ may be estimated online.

- **Frame type state**

Video streams are typically compressed into GoP structures containing intra-predicted (I), inter-predicted (P), and bi-directionally predicted (B) data units.

We consider frame types I, P, and B to illustrate the dependencies between frames as well as the impact of each frame type on the video quality. It has been shown that transitions among data unit types in a GoP structure can be modelled as a stationary Markov process. We assume that the choice of frame type is set constant by the encoder for the whole video sequence; therefore, the frame type transition probabilities depend on the desired ratio of I, P, and B data units (*e.g.*, IBPBPBP . . .) set by the video coder. Let $s^I \in S^I$ denoting the frame state.

- **Buffer state**

The state of the post-encoder buffer (in the proxy) is denoted by $s^e \in S^e$ and the state of the MAC buffer is denoted by $s^m \in S^m$. Here, the state of the post-encoder buffer describes the number of frames stored in the buffer. This helps to regulate the delay introduced within the system. The state of the MAC buffer corresponds to the number of PDUs or of bits (PDUs have all the same size) in the buffer.

- **Actions**

The layer filtering process is in charge of deciding the number of SNR layers to send among the two oldest frames in the post-encoder buffer.

$\mathbf{a}_t = (a_{1,t}, \ldots, a_{L,t}, a_{L+1,t}, \ldots, a_{2L,t}) \in A$ is the vector of actions taken in the post-encoder buffer between time t and t+1, where $a_{l,t} = \{-1, 0, 1\}$ for $\ell \in \{1, \ldots, 2L\}$ is the filtering decision for the $\ell-$th SNR layer of the last two frames in the buffer. $a_{\ell,t}$ represents the number of transmitted packets from the post-encoder buffer, when its value is positive, or the number of dropped packets when it is negative. If $a_{\ell,t} = 0$, packets are neither transmitted nor dropped.

- **Reward function**

The layer filtering process should be designed to maximize the video quality at the receiver side. At each regulation time, a reward value should be calculated for each transmission action. The reward should indicate to the control system how good or how bad the chosen action is. Ideally, one should estimate and maximize the video quality of the decoded frame. QoE is the key criteria for evaluating the video service, however, it is difficult to efficiently estimate it at the controller due to the delay caused by the MAC and the receiver buffers. To tackle the problem of transmission delay, the proposed layer filtering process should be able to estimate the QoE of the decoded video at each time based on the chosen action, the environment condition, and the actual buffer state. Thus, the reward function to be maximized by the MDP framework is the estimated QoE done at the transmitter that captures the relation between the video coding as well as the network parameters affecting the video quality at time t. Several objective and subjective video quality measurement techniques are available, see, *e.g.*, [37][38][39] and the references therein. The QoE value can

be derived from the PSNR, SSIM, or any other metric, as in [30], using automatic QoE measuring tool, corresponding to the resulting quality of the transmitted frames.

A frame may be dropped by the layer filtering process or if the post-encoder buffer is full, or if the layer filtering decide to transmit the frame but the MAC buffer has no more space to store it. We propose to decompose the QoE estimation into two steps. In the first step, the estimation is performed based on the chosen action regardless of the buffers states. In the second step, the estimator calculates the next state of the MAC buffer using the chosen action in order to estimate whether the frame will be safely stored or dropped from the buffer. We assume that the receiver buffer is filled in a progressive way which allows the viewer to decode as a new frame arrives in the buffer.

In case of non reception of the base layer of the reference frame, error concealment techniques, in order to reconstruct the missing frame, are used. If the *Frame copy* [33] algorithm is performed when a frame is lost, the video quality is assumed to be reduced by $\lambda(s_t^I)$ due to the loss of the current frame depending on the frame type. These values can be estimated experimentally off-line.
The total reward function is

$$\begin{cases} r_t(s_t, a_t) = r_{t-1}(s_{t-1}, a_{t-1}) - \lambda(s_t^I) & \textit{if the frame is lost} \\ r_t(s_t, a_t) = r_q^t(s_t^I, a_t) & \textit{else} \end{cases}$$

Where $r_q^t(s_t^I, a_t)$ is the QoE value estimated at the controller corresponding to that of the decoded frame at time t when correctly received depending on the action and the frame type.

In order to learn the optimal action to choose at each state, an online learning algorithm is used.
Learning algorithm for the optimal policy

In practice, the reward and transition probability functions are unknown *a priori*. Consequently, the optimal policy cannot be computed using value iteration. Thus, we adopt a model-free reinforcement learning algorithm.
Q-learning (Q^*: S × A → R) is an RL technique that works by learning an action-value function that gives the expected reward of taking a given action in a given state and following a fixed policy thereafter.

We define the optimal action-value function Q^*: S × A → R, which satisfies:

$$Q_{t+1}(s_{t+1}, a_{t+1}) \leftarrow (1-\alpha_t) Q_t(s_t, a_t) + \alpha_t \left[r_t + \gamma \max_{a \in A} Qt(s_{t+1}, a'_t) \right]$$

Where a'_t is the greedy action in state s_{t+1}, which maximizes the current estimate of the action-value function; $\alpha_t \in [0, 1]$ is a time-varying learning rate parameter; and, $Q_0(s, a)$ can be initialized arbitrarily for all $(s, a) \in S \times A$. Here, $Q_0(s, a)$ is initialized to zero for each state action pair.

When the XLO is triggered by a congestion notification for the Core Network or the Wireless Access, the control process starts by doing exploration. The controller goes through the same environment many times in order to learn how to find the optimal actions. During the exploration phase, the Q-learning rules are performed by executing each action in each state a number of times until the Q values converges to the optimal value. To reduce the execution time and learning time, we use the virtual experience method proposed in [33] to update the state-action space. More details about the method are available in [33].

In order to make the controller aware about the different changes that may happen in the system (channel condition, video characteristics, etc.), we perform ε-greedy algorithm where the controller identifies the best action according to the state-action values.

Algorithm : Q-learning algorithm

Initialize $Q_0(s, a)$

For each time step **do**:

Choose a_t from s_t using policy derived from Q_t(ε -greedy)

Take action a_t,

Observe r_{t+1}, s_{t+1}

$$Q_{t+1}(s_{t+1}, a_{t+1}) \leftarrow (1-\alpha_t)Q_t(s_t, a_t) + \alpha_t\left[r_t + \gamma \max_{a \in A} Qt(s_{t+1}, a'_t)\right]$$

End for

$s_{t+1} \leftarrow s_t$

Future Work

The results obtained from the proposed layer filtering algorithm are promising. However, we assume that when the PDU packet is transmitted from the MAC buffer to the client, it is well received. To test this assumption, we plan to study the impact of the PDU losses on the already obtained results and to analyse how the reward function should be updated in order to capture this phenomenon.
In parallel, we started implementation on the OpenAirInterface (OAI) platform [35]. Our goal is to assess the performance of our algorithm by using realistic 3GPP stack of an LTE network. The integration of the QoE curve in the reward function is in progress.

3.3 QoE-based optimisation with network layer awareness on hybrid wireless network (PathSelection)

In a mobile network, a given video application can be served through different paths, resulting in different QoE, measured at the end user by any method of choice and fed back to the network. Thus, the mobile operator is in charge of guiding the service i) to keep a target QoE and ii) to optimise the network resource usage.

The diversification of wireless access technologies allows addressing application requirements in potentially different ways. This is particularly true when a mobile user can access the same content in the Internet via both cellular and wireless local area networks. A user might perceive a different QoE over different paths and should have the possibility of getting the best available QoE. In this view the mobile operator needs a tool to provide the network costs associated to the different possible paths and guide the applications to better optimise network resources. Such a tool is being defined by the IETF Working Group ALTO (Application Layer Traffic Optimisation) which is designing the ALTO client-server protocol.

Starting from current trends in standardization, we are working on two key objectives: quasi-real time evaluation of the perceived QoE by end users and application requirements in terms of network resource usage. The online QoE computation algorithms combined with application layer optimisations give a promising solution. To this end we propose to exploit the ALTO protocol and its extensions for the mobile core and to evaluate key metrics (QoE-based video metrics and network metrics) in simulation and live test experiments.

We further investigate how video applications can be served through several video delivery paths, resulting in different video quality experienced by the end user. This means that the mobile operator is in charge of guiding the video services to keep a target video quality and to optimise the network resource usage, taking into account network metrics related to both core and wireless access networks. Thus, we designed a optimisation framework which runs online algorithms to select the best path for video delivery based on a set of performance metrics.

We propose a new concept of QoE online computation where we take into account metrics impacting the quality perceived by the user from the Core Network (CN), i.e., involving CDN-related metrics, and from the Radio Access Network (RAN), i.e., related to wireless channel metrics. The combination of these two sets of metrics opens the door to a promising research direction taking into account issues related to the whole video

delivery chain. Moreover, based on these metrics we intend to develop an online no-reference QoE computation method suitable for mobile devices. The key idea is to combine metrics that specify a typical CDN environment [11] with metrics collected at the mobile operator side. The mobile operator has the unique possibility of combining both worlds and of proposing fast adaptive algorithms to optimise the perceived QoE [21]. Figure 3 depicts a simplified vision of the Evolved Packet Core (EPC) enhanced for optimal CDN integration [12].

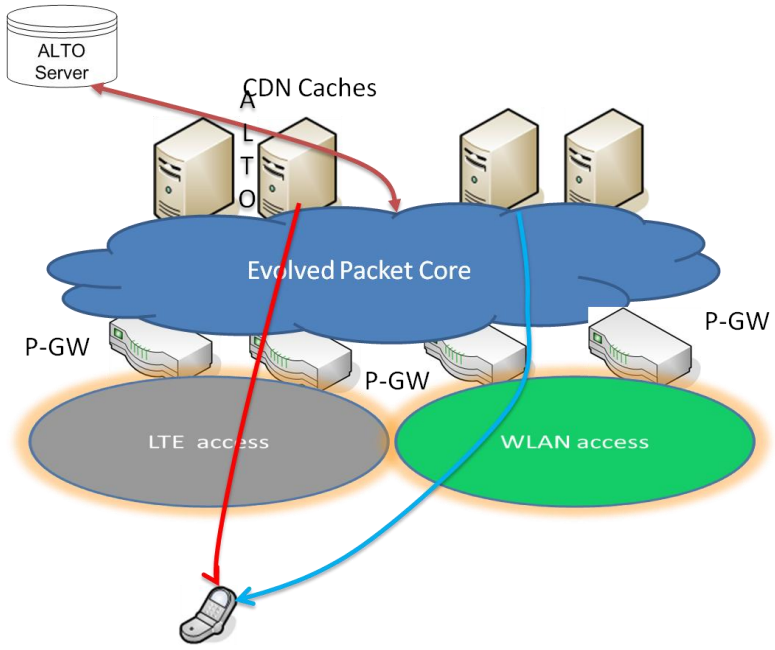


Figure 3: Considered network architecture.

Our idea is to build vectors reflecting each possible path of the whole video delivery chain, from a video source (cache) to the mobile user (terminal).
The goal of our work is to find the optimal vector of values for both CDN and wireless related metrics. Optimality here has a different meaning whether we consider the network operator's or the user's point of view.
Thus, we aim at optimising the set of vectors based on two different criteria reflecting the requirements of service providers and users. The common set of solutions is the optimal set taking into account both network operator and user's sides.
The aforementioned procedure is due to the fact that it is likely that the two solution spaces do not coincide, since the cheapest solution for a network operator in terms of resource usage unlikely gives the best QoE to the end user. Enhancing the perceived quality of a video comes at a cost, which usually results in higher bit rates to be provided by the network operator.
We foresee the design of a framework, to be implemented on the mobile terminal, which runs a computationally light optimisation algorithm in real-time to select the best path for the video delivery, ensuring a target user's QoE level under the constraints of the available network resources, i.e., network operator's costs.

Algorithm

In video applications the user requests a video from the operator, which is in charge of providing such service at a given target video quality agreed with the user. In the mobile network, the requested video may be stored in more than one CDN video cache and the operator selects one of these video sources to provide the requested service. Once a CDN cache has been 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 sufficient to identify a video path from the source to the end user.

Once the set of metrics is identified, we can represent each available video path j with a tuple (vector) V_j, where the i-th element of V_j is the value taken by the i-th metric P_j^i, with i=1,...,M. Assuming N available tuples of values taken by M metrics (M-tuple), we can represent the set of tuples $\{V_j\}$, with j=1,...,N, as follows:
$V_j = \{P_j^1,...,P_j^M\}$.

Optimisation 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 value 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 details of the procedure are given later in this section) 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 closer to 0, the worse the tuple. Hence, we design our optimisation algorithms with the aim of maximizing a utility function on the "normalized" M-tuples.

Objective function

The first optimisation algorithm 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 optimisation 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 a minimal network performance level. Thus, a suitable user utility function is such as to maximize the minimum proximity value associated to the metrics in the M-tuple *(max-min* criterion).
The set of network metrics, involved in the optimisation procedures, needs to be (i) identified, to specify each possible video path, and (ii) associated to application performance metrics, used to evaluate the performance of the optimisation algorithms. In the first step, we separately define the metrics which characterize the Core Network (CN) side and the metrics for 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, while the data rate can be assumed to be infinite (no error propagation phenomena nor losses). In the WAN, on the contrary, the delay has a negligible influence on streamed 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 pseudo-code of the two algorithms, performing the max-sum and max-min optimisations, as shown in Algorithm 1.

Every time slot, a set of tuples $\{V_j\}$, with j=1,...,N, is given as input to both optimisation algorithms. The values taken by the network metrics are such that the higher the value, the worse the performance.
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 give the "mapped"' version, i.e., within the interval (0,1], of the original set of vectors, i.e., $\underline{V_j}$, with respect to the ideal tuple. In order to distribute the values in the whole unit interval and to make

metrics comparable within the tuple, the linear mapping is such that the ideal value of a metric is associated to 1 and the worst value of a metric is associated to 0.
Thus, $\underline{V_j}$ can be seen as the proximity vector to the ideal V*.

Algorithm 1 Max-sum and max-min optimization algorithms.

Input: $V_j = \{P_1^j, ..., P_M^j\}$, with $j = 1, ..., N$; fixed a_i, with $i = 1, ..., M$;
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 the mapping to the unit interval, $\overline{V_j} = \{\overline{P_1^j}, .., \overline{P_M^j}\}$;
· Multiply each metric i in $\overline{V_j}$ with the weighting coefficient a_i:
· $\overline{V_j} = \{a_1\overline{P_1^j}, .., a_M\overline{P_M^j}\}$
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.

The metrics involved in the optimisation problem, after being mapped to the interval (0,1] with respect to the ideal values, are weighted with coefficients a_i in [0,1], i.e., the weight given to the i-th metric, in order to appropriately tune the optimisation algorithms to meet the actual network settings and the operator's preferences. For instance, when the weighting coefficients of the CN-related metrics are set to 0, only the wireless access matters, i.e., the framework optimises 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 optimisation 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's 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), 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 [25]. Then, we define a wireless-related metric which takes into account the channel quality as a Signal-to-Noise Ratio (SNR) of both cellular and WLAN access. This value can be 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 the mobile terminal in real-time, opposite to the aforementioned 3 network metrics. Moreover, adding network metrics to the optimisation problem further increases the complexity of the system, running the undesirable risk of growing the computational time of the algorithm beyond the hardware capabilities and 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 scope of this work, thus, this analysis is left for future work.
In the following, we explicitly map the network metrics involved in the optimisation problem to the application performance metrics (APM) evaluated in our Matlab simulator: the channel capacity and the response time.

Channel capacity
The wireless access metric, defined as the SNR of the wireless channel, is mapped to the upper bound of the wireless channel by using the Shannon-Hartley theorem [26]:

$$C = B \cdot \log(1 + SNR)$$

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, while the cache load is translated to the time needed to retrieve the video from the storage in the cache (hence, proportional to the ratio 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 $T_{retrieve}$.

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

α and β 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, α is set to the time unit of response time, i.e., 1 ms (average time to travel through one hop); β reflects the characteristics of the Solid State Drive (SSD) and it is set as well to 1ms. If the Hard Disk Drive (HDD) technology is implemented, then β increases by a factor around 50 due to the slower random access time compared to the SSD.

The simulation results are presented in Section 5.3.

Due to the modularity of the framework here presented, we also envision the possibility of running an instance of the algorithm in the DM module when the wireless metrics are not required to select the best video path available in the core network, whereas the algorithm runs in the XLO when both CN and RAN metrics are required. Furthermore, as explained in Section 3.5, this algorithm fits well in a multi-level optimisation algorithm, which is the combination of the optimisation algorithms running in the TO component.

Future Work

The design of a heuristic algorithm for the optimisation of the network and quality metrics that affect the QoE of a user at the mobile terminal is currently work in progress.

3.4 FEC rate-adaptation algorithm (FECAdaptation)

We design a fast decision algorithm which is able to compute the amount of error protection needed by the video transmission based on the wireless channel feedback. Starting from a theoretical distortion model for the streamed video sequence, we derive an optimisation problem whose aim is to find the optimal rate allocation for video packets and error protection packets, given an instantaneous channel realization. While the optimal solution to this problem requires a full search through the complete solution space, our heuristic approach takes a much faster decision based on a careful look at the properties of the scalable encoded bit stream, and of the unequal error protection mechanism. During an iterative process we take a step-wise utility-optimal decision which increases the transmission rate until the channel rate is achieved, either by adding redundant packets to an already scheduled video layer, or by adding packets of a higher enhancement layer. Our FEC optimisation mechanism consists of two modules. At each eNB, a local optimiser determines the number of video layers to be sent, the corresponding modulation and coding scheme (MCS) to be used for each video layer, as well as the required amount of FEC. Per-layer FEC data is generated at a FEC module located at or close to the video source and these packets are multicast along with the SVC video data. Based on the instantaneous channel feedback, the local optimiser runs our proposed algorithm, obtaining the transmission policy appropriate for the given channel. Based on this policy, the video and redundant packets are scheduled for transmission or simply dropped from the queues, thus achieving our UEP scheme for the scalable encoded bit stream.

We further design a robust opportunistic scheduler to be implemented at the base station in a cellular network for multicast media streaming applications. The scheduling mechanism operates based on the average and instantaneous user distributions and radio link channel quality, information obtained through the cellular uplink channel. Multiple scalable video streams are split into video packets, which are opportunistically scheduled with the goal of minimizing the wireless resource usage while keeping the overall target QoS.

For layered multicast streaming using SVC, the server needs to protect the transmission of the video stream against the wireless channel losses. In this context, forward error correction (FEC) techniques are best suited to provide robustness to the transmission process, without the cost of extra added delay of packet retransmissions. Unequal error protection (UEP) schemes, which better protect the more important parts of the video stream, outperform regular robustness schemes, where the error correction algorithm does not take into account the specific video encoding format. The increased quality of the received video, even in adverse channel conditions, comes however at the cost of allocated bandwidth for redundant packets. In the following, we present a fast decision algorithm which is able to compute the amount of error protection needed by the video transmission based on the wireless channel feedback.
Our FEC optimisation mechanism consists of two modules. At each eNB, a *local optimiser* determines the number of video layers to be sent, the corresponding modulation and coding scheme (MCS) to be used for each video layer, as well as the required amount of FEC. Per-layer FEC data is generated at a *FEC module* located at or close to the video source and these packets are multicast along with the SVC video data. Packet marking ensures that in case of congestion in the core or backhaul, the least important multicast packets are dropped first.

Local optimiser at eNB

We assume that the eNB has information about the set of available modulation and coding scheme $MC_j \in \boldsymbol{MC}$ per terminal j as well as the corresponding packet loss probability p_j (without ARQ) on the wireless link. The loss probability p_j is updated periodically as channel conditions change. The eNB further has a maximum capacity C allocated to the video stream.
1) We compute the best modulation and coding scheme MC_j for each individual user j, based on the corresponding bitrate $R(MC)$ and loss probability $p(MC)$. Given MC_j, we can compute the subset of modulation and coding schemes that can be used by the multicast system for transmission, that are decodable by user j.

$$MC_i = \arg\max_{MC} R(MC) \cdot p(MC)$$

2) Next, we compute the appropriate assignment of modulation and coding schemes *MC(k)* for each transmitted video layer *k*, based on the first step. We assume that a video layer *k* transmitted with *MC(k)* can be decoded (possibly with some errors) by all users *j* that support a better or equal MC_j, and hence reduces the video distortion for these users. All other transmitted video layers are considered to be undecodable by user *j*.
We denote the subset of users which can decode a given modulation and coding scheme *MC(k)* by S_k. Our algorithm starts by assigning the lowest MC scheme (supported by all users) to all video layers until the channel capacity is filled. Then we increase the MC scheme sequentially for each layer, and we assess the benefit of this action by looking at the trade-off between the extra video quality achieved by saving capacity using higher MC schemes, and the number of users that are able to decode the video information. Once the MC scheme of one video layer is fixed, the MC scheme for all remaining higher layers is considered to be at least as high. Algorithm 2 presents the sketch of our proposal.

Algorithm 1 Selection of $MC(k)$ for each video layer k.

Input: Video layer rates $\rho_k, \forall k \leq L$, available MC schemes $\mathcal{MC}$, ordered user subsets S_k, channel capacity C;
Procedure: MC selection $\forall$ **video layers.**
Initialization: video layer $k = 1$, MC index $i = 1$;
Assign $MC(u) = i$ to all video layers $u \geq k$ up to capacity C;
Update l based on the assignment of the $MC(u)$;
Compute $D(l) = \sum_{j \in S_i} D_j(l)$; $D_{opt} = D_l$;
while video layer $k \leq L$ **do**
  while $i \leq M$ **do**
    $i := i + 1$;
    Test $MC(u) = i$ for all video layers $u \geq k$ up to capacity C;
    Update l based on the assignment of $MC(u)$;
    Compute $D_l = \sum_{u=1}^{l} \sum_{j \in S_{MC(u)}} D_j(u)$;
    if $D_l \leq D_{opt}$ **then**
      $D_{opt} = D_l$; $MC(u) = i$ for all video layers $u \geq k$;
    else
      Break;
    end if
  end while
  $k = k + 1$;
end while
Output: l, $MC(u)$ $\forall u \leq l$.

Here, D_j represents end-to-end video distortion, as perceived by one media client *j*, as an additive metric depending on both the source distortion and the distortion (in terms of MSE) presented in [20]. As network packets contain in general data referring to the same amount of video information (e.g., one frame, one slice, or one encoded video layer of a frame), the distortion is proportional to the number of lost packets, and is differentiated by the importance of the video layer containing the lost packets. Finally the total distortion of our multicast scenario can be computed as the sum of the individual distortions of all users in the system:

$$D(l) = \sum_{i=1}^{N} D_j(l)$$

3) Once we have established the appropriate MC scheme for each video layer, we need to establish the final transmission scenarios in which we protect the video information with application-layer FEC. To this end, we explore the trade-off between sending additional video layers, or better protecting the already scheduled layers, given the total available application rate. The MC scheme used for each video layer defines the capacity needed for transmitting the respective video layer. We associate to each *MC(k)* chosen for layer transmission the packet loss probability p_k of the worst user assigned to the given subset S_k. Within this framework, we present a fast algorithm, which explores at each optimisation step the trade-off between adding another video layer for transmission, or increasing the FEC protection of the previously scheduled layers. The decision of the algorithm is taken based on a utility function which assesses the decrease in overall video distortion of these two actions. The output of the algorithm consists in the number of video layers scheduled for transmission and their associated rate, plus the amount of additional FEC protection to be scheduled for each layer.

Our algorithm performs on a per-GoP basis; however, its functionality remains intact also on smaller or larger time frames. The algorithm takes as input the parameters *C* and *p*, and the rates ρ_k of the video layers *k* available for scheduling. This algorithm solves an iterative optimisation problem, in which at each step it decides whether to increase the transmission rate of the video, possibly scheduling another enhancement layer, or to increase the FEC protection of an already scheduled video layer. The decision is based on a utility function, which computes the decrease in distortion of a possible action (between adding more FEC redundancy or scheduling a new enhancement layer), relative to the channel resources needed by this action.

For easiness of presentation we assume that the algorithm has already scheduled the entire video base-layer for transmission. At each iteration the algorithm computes the increase of the total video distortion by either *i)* increasing the error protection of the currently scheduled video layer *k* by one additional FEC packet, e.g., $n_k=n_k+1$ or, *ii)* adding the first packet of an additional enhancement layer. Note that according to our linear distortion model for the enhancement layers, once the algorithm decides to add packets of a new video layer, it will proceed until the full layer is scheduled for transmission. The algorithm iterates until the capacity *C* is reached. Assume that action *a* achieves an increase in distortion *D(a)*, with the cost of *R(a)* of rate consumed. The step-wise optimal decision taken by the algorithm is:

$$a^* = \arg\max_a \; U(a) = \frac{\Delta D(a)}{\Delta R(a)}$$

The sketch of the algorithm is formalized in Algorithm 2.

Algorithm 2 Fast Rate-Adaptation Alg. for SVC with UEP.

Input: Channel params. C and p, video layers ρ_k, $\forall k \leq L$;
Procedure: Rate Allocation
Initialization: $l = 1$, $R_v = \rho_{temp} = \rho_1$, $n_j = g$, $\forall k \leq L$;
while $R_v \leq C$ **do**
  if $\rho_{temp} == \rho_l$ **then**
    Compute $U(1)$ and $U(2)$ according to the actions described above;
    if $U(1) \geq U(2)$ **then**
      $n_l \rightarrow n_l + 1$, $R_v \rightarrow R_v + \frac{\rho_l}{g}$;
    else
      $l \rightarrow l + 1$, $R_v \rightarrow R_v + \frac{\rho_l}{g}$, $\rho_{temp} = \frac{\rho_l}{g}$;
    end if
  else
    $R_v \rightarrow R_v + \frac{\rho_l}{g}$, $\rho_{temp} \rightarrow \rho_{temp} + \frac{\rho_l}{g}$;
  end if
end while
Output: Rate allocation tuple: $(R_v(l), FEC(n_k, g))^*$.

Here, *g* is the size of the GoP in number of packets, i.e. the minimum number of packets that can be scheduled for this GoP.

FEC module

The FEC module located at or close to the video source uses multiple multicast streams to distribute video and FEC data to the base stations. The base stations periodically provide feedback on the maximum number of video layers transmitted to the associated terminals as well as the amount of FEC per layer as determined by the algorithms above. The FEC module then re-computes the multicast distribution tree to avoid unnecessarily transmitting video layers and FEC information to parts of the network where they are not needed. Such adaptations occur over longer time scales of maybe minutes. Short term congestion is dealt with through the packet marking and preferential packet dropping mechanisms from the traffic engineering module.

Future Work

We do not foresee further research advancements on this topic within the MEDIEVAL project.

3.5 Towards a joint optimisation framework

The aforementioned optimisation algorithms address different problems on the video delivery chain in the network and adapt the video streams in different ways. The first three optimisations, namely TrafficManagement (described in section 3.1), SVCFiltering (described in section 3.2), and PathSelection (described in section 3.3), impact each other when working together in the network. FECAdaptation (described in section 3.4), on the other hand, can optimise for reliability independently without interworking with the other three schemes. Therefore, in this task the first three optimisations are considered to be combined into a joint framework for video transport optimisation. This section presents the preliminary idea of why and how the three optimisation schemes could work jointly.

PathSelection optimally selects the video paths given multiple metrics (core and access networks). The optimisation works on a video chunk level, a granularity that could react to user mobility and traffic pattern variations. TrafficManagement efficiently allocates the data rates of multiple video streams. The optimisation works on a GOP level, a finer granularity that has an impact on the perceptual video quality. SVCFiltering dynamically filters video packets on a video frame level, the finest granularity that could react to the fast variations of wireless channel conditions. By working on different time granularity, the three optimisation schemes adapt the video transport subject to the network dynamics with appropriate timing. They adapt the video transmission with a common objective: the QoE is maximized while the network resources are efficiently utilized.

When handling different issues, the three optimisations are able to overcome their limitations by interworking with each other. SVCFiltering has the finest time granularity but is restricted to single streams, losing the benefit from multiplexing users. TrafficManagement addresses this issue leveraging the diversity of video sensitivities. They are both limited to the assigned path of the videos. PathSelection provides the possibility of offloading the traffic and redirecting the users, leaving the stream adaptation to TrafficManagement and SVCFiltering in the best selected video path.

The three optimisations can be combined to work in a common framework in a distributed way. PathSelection determines the optimal end-to-end paths to transport the video streams. TrafficManagement manages multiple streams over a network node provided the conditions of their paths. SVCFiltering performs a finer filtering of each stream at eNodeB given more dynamic status of the wireless paths.

Let us take the transportation of an SVC stream as a toy example. The end-to-end path is first selected optimally to carry the SVC stream by the PathSelection algorithm. TrafficManagement is assumed to be placed in the P-GW and it calculates the target data rate for a series of GOPs in order to maximize the overall QoE within its scope. SVC layers may be dropped if the target data rate is not met. SVCFiltering works in the eNodeB in a more dynamic fashion. For each frame, packets may be dropped upon SVCFiltering's

decision. During the streaming session, PathSelection may redirect the path for the SVC stream whenever better available video paths can be selected. TrafficManagement and SVCFiltering are triggered upon updated network conditions which do not require the change of the current video path. As a result, the SVC stream is adapted at different granularities, places and time scales.

The video transport benefits from the three-level optimisation framework in order to overcome most of the critical issues in the network, e.g., congestions, user mobility and wireless channel variations.

4 Update on specification work

The purpose of this chapter is to give the final specification of the modules and of the internal and external interfaces designed in Task 5.2 (Transport Optimisation component, TO, Figure 4), since the first description given in deliverables D1.1 [2] for the external interfaces and in D5.1 [3] for the internal interfaces and modules.

In this section we describe the TO modules, i.e., the XLO, TE and CNM modules, taking into account the optimisations designed in Chapter 3. We highlight how the two subsystems in WP5, CDN and TO, interact via an internal interface and we present how the TO subsystem interacts with the Mobility, Wireless Access and Video Services sub-systems through the external interfaces.

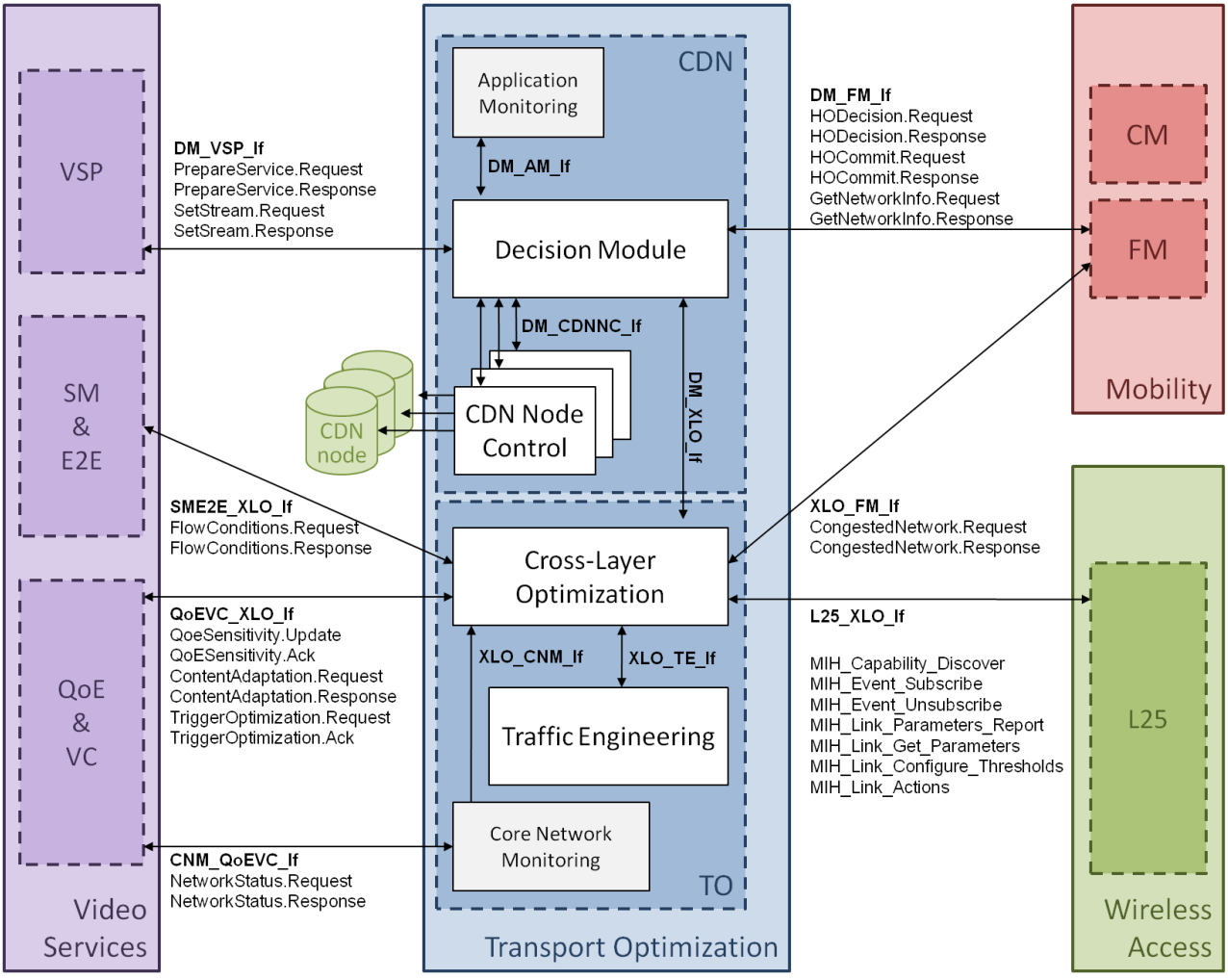


Figure 4: Modules and Interfaces of the Transport Optimisation subsystem

As mentioned in Deliverable D5.1 [3], in the Cross-Layer Optimisation module (XLO) we aim at computing the actions to be performed in the Traffic Engineering (TE) module, as the outcome of heuristic algorithms designed to solve an optimisation problem.

The XLO module also leverages the adaptation ability provided in the QoE and Video Control module of the Video Services subsystem [30] and in the Flow Manager module of the Mobility subsystem [31], when the transport issues cannot be solved via TE.

The Core Network Monitoring (CNM) is the module in charge of monitoring the core network and of detecting eventual congestions in the network. These modules are shortly described in the next section, highlighting the relation with the algorithms in Chapter 3, followed by the description of the internal and external interfaces.

4.1 Modules

In this section we provide the final specification of the modules in Task 5.2.

4.1.1 Cross-Layer Optimisation (XLO)

This module resides in the MAR. It reacts to the events and problems in the network and it cooperates with the other layers upon their requests. Inside the XLO module the four optimisation algorithms described in Chapter 3 find the solution to best optimise the transport under the given constraints. The cross-layer information is used by the algorithms for application- and network-aware optimisations. The solutions computed by the XLO module are not limited to the adaptation within the core network side but also impact the other layers.

The XLO can be triggered by congestion in the wireless links detected by the Wireless Access, or congestion inside the network detected by the CNM module. When the Video Services discovers that the network conditions do not meet the requirements of the applications, the XLO module is requested by the Video Services subsystem to optimise the transport layer. For what concerns the handover events, the XLO also gets information from the Mobility subsystem about the completion of handovers upon the request of moving a video flow (e.g., due to the impossibility of solving the issue via XLO optimisations).

When the XLO is triggered, the problems are addressed in different scopes and optimised by the selected algorithms at different levels, which is described in more details in Section 3.5. Cross-layer information is needed to best fit the specific requirements and characteristics of the applications and of the network conditions. The XLO module takes application-layer information from the Video Services subsystem, e.g., the video sensitivities, to maximize the QoE of users in the optimisations. For legacy application servers where the application-layer information might not be available, the module can use default parameters to achieve an average QoE level of performance. The XLO module retrieves the wireless link conditions from the Wireless Access subsystem in order to manage the transport under the constraints of the wireless access. The whole cross-layer information enables the XLO module to efficiently manage the resources in the network and provides satisfactory QoE to the users.

When the XLO computes the optimal solution, it requires the other modules to take actions accordingly. If the outcome is to adapt the video traffic, the XLO module triggers the TE module to execute an action. Either rate shaping schemes or packet marking could be performed in the TE module according to the instructions of the XLO module. If the XLO module is unable to solve the issue through an action that can be performed by the TE, it can request the Video Services subsystem to perform some content adaptation at the video source. Another solution that can be taken by the XLO is to offload the video traffic, thus it requests the Mobility subsystem to offload some video flows. Besides, for more dynamic and fast rate shaping, the packets can be marked by the TE before they are delivered through the Wireless Access subsystem. The latter subsystem is able to drop the packets according to the marking. Thus, in addition to the TE actions, the XLO module leverages a multi-level (Video Services - Wireless Access - Mobility subsystems) set of criteria to address the issues raised from the network and that cannot be solved by one of the TE actions.

4.1.2 Traffic Engineering (TE)

This module is placed in the MAR. It executes the traffic engineering actions dictated by the XLO module. The actions taken are listed as follows:

- Scalable layer filtering (solution in Section 3.2):

- Frame dropping (B or P): in case of non scalable encoded streams, the traffic engineering may decide to drop low priority packets like bi-predictive in order to reduce the bitrate. This drop is controlled by the XLO module (solution in Section 3.2);
- Frame scheduling: the SVC layer filtering scheme proposed in this document decides to skip a frame and to wait for the next period to check if there is room to send it in order to maximize the QoE. This action implies a re-prioritization of the different video packets before being sent to the Wireless Access;
- Transcoding (solution in Section 3.1): according to the calculation by the QoE-based optimisation algorithm of Section 3.1, we can choose to transcode a stream in order to improve the QoE;
- AL-FEC adaptation (solution in section 3.4): we can decide to increase the reliability of the video flow.

4.1.3 Core Network Monitoring (CNM)

The CNM module monitors the core network and triggers the XLO module in case network congestion is detected; as well it provides useful information to the video control in order to adapt the content taking into account the network status, buffer states, delays, lost and momentary service throughput.

The required features in the MEDIEVAL project are already supported by some commercial tools (e.g., the wireless network guardian of Alcatel Lucent **Errore. L'origine riferimento non è stata trovata.**). The following features are implemented:

- Subscriber's quality of experience monitoring;

- Analysis and identification of root-cause issues that are contributing to a subscriber's degraded experience;

- Checking whether an issue originates within a cell site, on a backhaul link, within the packet core network, across devices, or with a misbehaving application;

- Identifying a wide range of performance issues associated with the subscriber's data session, including issues such as poor cell performance, inability to launch a data session, DNS failures, poor-performing device and more.

The existing tools are sufficient to fulfil the different requirements of the project, i.e., the core network monitoring is out of scope from the MEDIEVAL research point of view.

4.2 Internal interfaces

In this section we provide a high level description of the internal interfaces. For more details, please see Annex A.1.

4.2.1 DM_XLO_If

This interface is used by the XLO to request the list of End-Points (EPs) available in the core network from the DM.

The DM provides the list of available EPs for a given session of a user to the XLO, which then runs the optimisation algorithm for selecting a video path in the network with better performance metrics, if available and necessary, using also the wireless metrics coming from the interface L25_XLO_If (4.3.2).

Moreover, this interface is used by the DM to request the XLO to perform an optimisation run (thus, using also wireless metrics) and send back the best available video path computed.

List of primitives:


```
XLO_DM_ALTO.request(from XLO to DM)
XLO_DM_ALTO.response(from DM to XLO)
DM_XLO_Optimise.request(from DM to XLO)
DM_XLO_Optimise.response(from XLO to DM)
```

4.2.2 XLO_CNM_If (conceptual interface)

The main objective of this interface is to provide information about congestion in the core network, detected by the CNM, to the XLO module. The XLO is triggered to select the best optimisation solution to be adopted for such detected issue. This interface is not implemented within the project.

List of primitives:

```
CNM_XLO_Congestion_Report.Indication(from CNM to XLO)
```

4.2.3 XLO_TE_If

This interface is used by the XLO to communicate to the TE module the action to take according to some criterion based on QoE (maximize the global QoE in the congested area, QoE fairness): it can be used to signal to drop layers, to drop frames, to re-prioritize packets, etc. This interface is activated if the cross layer algorithm succeeded to find a solution to solve the congestion within the Transport Optimisation subsystem.

List of primitives:

```
XLO_TE_TrafficAdaptation.Request (from XLO to TE)
XLO_TE_TrafficAdaptation.Response (from TE to XLO)
```

4.3 External Interfaces

In this section we provide a high level description of the external interfaces used by T5.2 modules to interact with the Video Services, Wireless Access and Mobility sub-systems. The details of the external interfaces are provided in deliverable D1.3 [27].

4.3.1 FM_XLO_If

The XLO module is activated when it receives a trigger of congestion and starts finding a solution by exploiting the dependency and the heterogeneity of the video flows in the congested area. In some cases, it is sufficient to move a flow to make room for the remaining flows to benefit from an overall improved QoE at the end user side. To do that, XLO_FM_If is used to trigger the Flow Manager to find a better path for a selected flow. The selection of the flow is an output of the XLO module. One can select the worst QoE experienced flow to be moved, for instance. The XLO is also informed of the handover events via this interface, upon the prediction or the completion of the handovers. The XLO is able to perform the optimisation considering the impact of the incoming users.

4.3.2 L25_XLO_If

The objective of this interface is to enable an efficient and optimised video transport by exchanging information through the abstract interface (L25). From the X-layer optimisation module, it receives

information to enhance the configuration of the wireless access: flow requirements (for QoS and multicast mechanism), flow identification, marking criteria.

All the functionalities of this interface are already included in the MIH_SAP through the extension of the MIH_Link_Actions and the reporting capabilities of IEEE 802.21. See details in D1.1 [2], Sections 8.3.1.1 and 8.3.1.2, for the primitives used in this interface, and as well in D1.3 [27].

4.3.3 SME2E_XLO_If

This is the interface between the session management & E2E network monitoring module (SME2E) and the cross-layer optimisation module (XLO).
This interface is used to communicate E2E measurement of monitored flows to the XLO. The XLO therefore has the knowledge of the end-to-end quality of the transportation in order to correct its optimisation. The SME2E module is able to measure the packet loss ratio, end-to-end delay, delay jitter. They give the XLO module an overview of the result of the end-to-end transportation.

4.3.4 QoEVC_XLO_If

This is the interface between the QoE Engine & Video Control module (QoEVC) and the cross-layer optimisation module (XLO).
Firstly, it is used to communicate video sensitivity information: the QoE Engine provides the XLO with video sensitivity information in the form of a utility function. This function describes, for instance, the relationship between the perceived quality and a set of objective parameters, such as data rate and packet loss. Transport Optimisation subsystem uses this function to estimate the perceived quality when computing the optimisation problem.
Secondly, it is used to communicate the requests between the two modules when one of them needs the other for appropriate adaptation in that layer. As soon as the XLO is not able to maintain a certain QoE level for a video flow, the XLO signals to the QoEVC a request for content adaptation (i.e., send video in a different format) in order to avoid or limit the amount of packets being dropped at the network. On the other hand, if the QoEVC finds that the application requirements are no longer satisfied, it could request the XLO to perform optimisation in the network.

4.3.5 CNM_QoEVC_If

This is the interface between the Core Network Monitoring (CNM) and the QoE & Video Control (QoEVC).

This interface is used to provide information about the core network status from the network monitoring block to the Video Services subsystem. The network in WP5 monitors its status and provides useful information, such as buffer states, delays, loss rate and current service throughput. The Video Services subsystem will collect that information, process it, and decide about the best adaptation mechanisms with respect to the specific service. In the case of live video, it may adapt the encoding rate, or change the amount of FEC packets.

5 Status of the evaluation work

The purpose of this chapter is to present the first results of the algorithms introduced in Chapter 3 and mapped for the final specification to the XLO-TE modules and to the interfaces to the other subsystems in Chapter 4. The simulation work is described for each proposed algorithm, except for Section 5.4 where we extend the simulation results for broadcast scalable video rate allocation and scheduling under mobility. For the implementation work, we focus our effort on the XLO module and its interactions with the other subsystems, as described also in deliverable D6.3[36].

5.1 QoE-based traffic management

Simulator

We use OPNET LTE model [28] to simulate video delivery over LTE interface. The OPNET LTE model supports Release 8 of the 3GPP standard. We are running experiments in LTE FDD 5MHz cells. More system parameter settings are listed in Table 1.

Table 1 QoE-based traffic management simulation parameters

Parameter	Value
Operation bandwidth	5 MHz
Carrier frequency	2 GHz
Channel model	ITU vehicular and ITU pedestrian
User mobility scheme	Vehicular and pedestrian speed
EPS bearer type	Best effort
Packet scheduler	Proportional fair
Video sequences	Foreman, Football, Soccer, in QCIF
Video codec	H. 264 AVC
Traffic engineering	Transcoding

Results

Single cell scenario

In this scenario, a cell fully loaded by video application users is set up. To make the cell loaded, video users are added one by one until the downlink of the system is fully utilized. Three videos, namely Foreman, Soccer, and Football, are used by the video streaming applications. Each user in the cell is receiving one of the video streams.

The optimisation algorithm to maximize the sum of QoE (described in Section 3.1) is applied. Transcoding is used to shape the streams according to the target data rates calculated by the optimisation. The following figures show the results of the QoE-based optimisation. Figure 5 shows the average MOS value of all UEs in the loaded cell. MOS values are around 3.7, which is in a satisfactory range. Figure 6 shows the MOS values of three video applications. It can be observed that the Foreman video streaming receives the best QoE and the Football video streaming receives the worst QoE. This reflects the objective of the optimisation algorithm, which is to maximize the overall QoE. The optimisation identifies the different sensitivities of the videos and allocates the resources based on the utility. As Foreman video has the least sensitivity and the Football video has the largest sensitivity, Foreman video could achieve better QoE with less resources, which contributes to reach a higher overall QoE.

Figure 5: Average MOS of all UEs in the cell

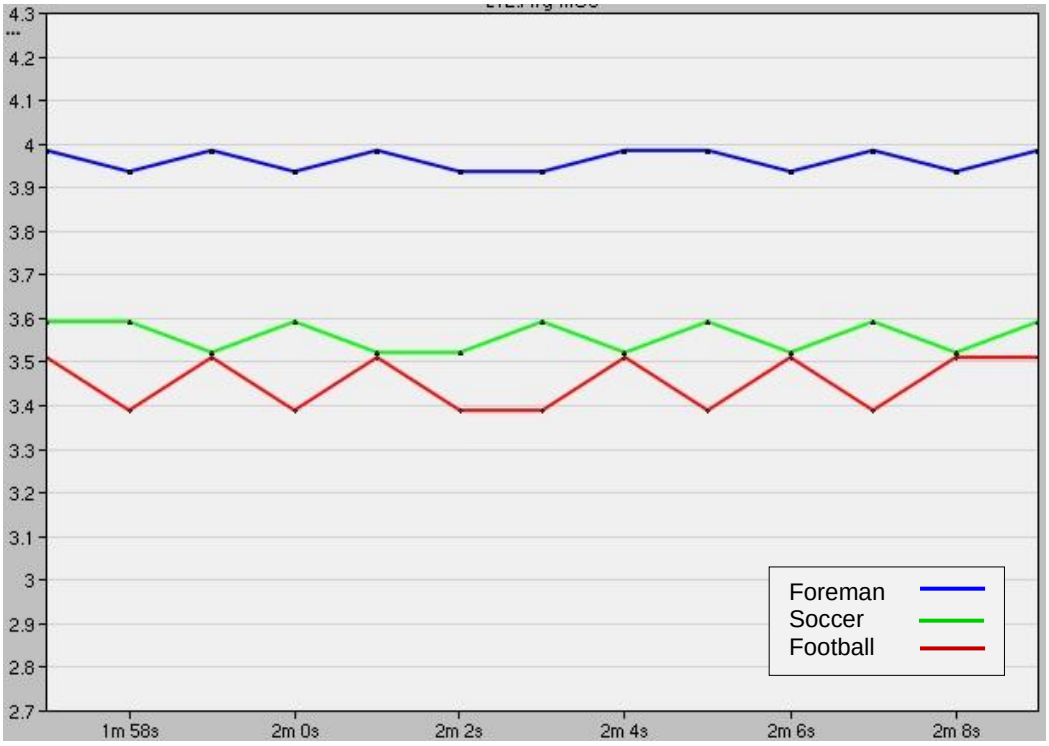


Figure 6: MOS of UEs with different video applications

Two-cell scenario

Furthermore, two-cell scenario is set up to simulate quality fluctuations when users are moving from a non-congested cell to a highly congested cell. The change of congestion status between the cells makes bandwidth of the user with a handover change drastically. The target cell is loaded until congestion in the same manner as in the single cell scenario. As the Football video is most demanding on bandwidth and has the largest sensitivity among the three videos, the handover user is set to receive this video stream for a worst

case scenario. The two schemes (TQSM and DUHO) to improve fluctuation of QoE during handover (described in Section 3.1) are applied.
The following two figures show the results of the optimisation during handover. The perceivable threshold is considered in the first scheme (denoted as "TQSM" in the following two figures) by applying the objective function with the penalty parameter (described in Section 3.1). The resource reservation is applied in the second scheme (denoted as "DUHO" in the following two figures) which reserves resources for the handover user (described in Section 3.1). In Figure 7 for TQSM scheme, due to the drastic lack of bandwidth in the congested target cell and the handover user just comes into the edge, the handover user experiences a drop of around 1.0 MOS after the handover. This is the best it can do with the constrained resources in the congested cell. In Figure 7 for DUHO scheme, resources are reserved in the target cell to keep the drop of MOS within 0.21 to avoid a perceivable degradation of video quality. In Figure 8, the CDF of the drop of MOS after handing over to the target cell is shown. Resource reservation helps to reduce the QoE fluctuation due to drastic bandwidth change during handovers.

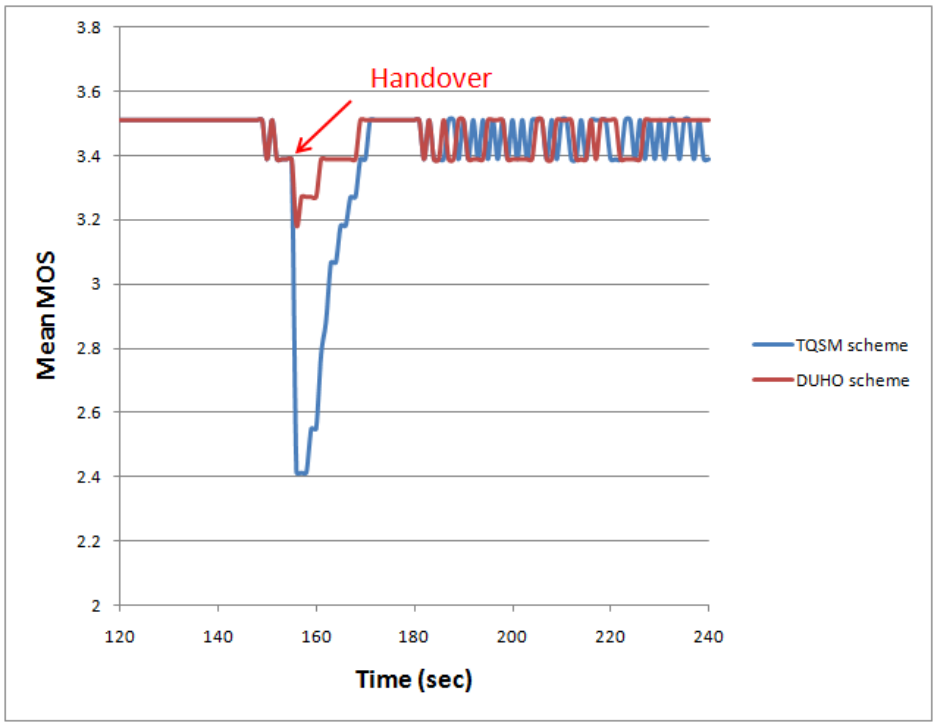


Figure 7: Mean MOS of the handover user

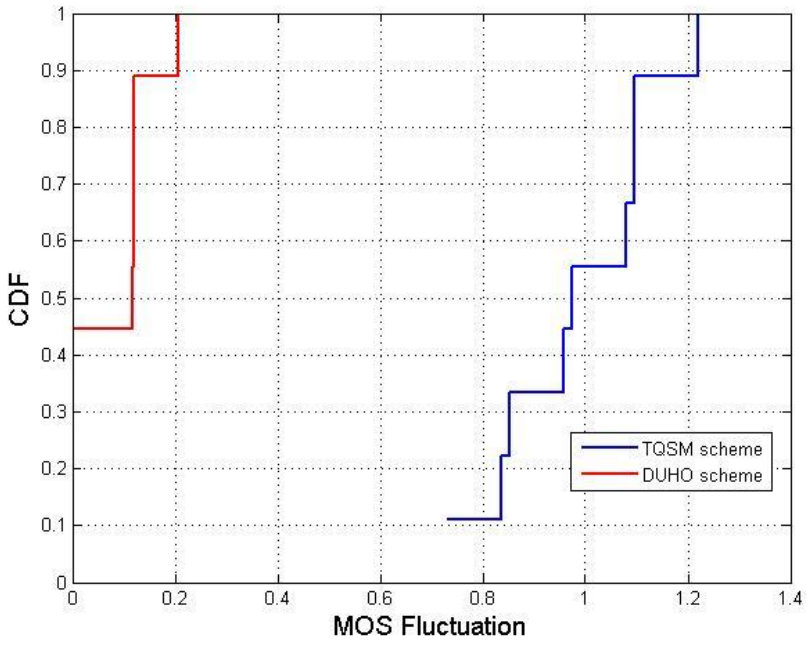


Figure 8: CDF of MOS degradation during handover

Future work
We will use the testbed to evaluate the performance of the QoE-based optimisation. We are applying SVC in order to take the advantage of its flexibility for rate adaptation.

5.2 QoE-based video scalable layer filtering process based on MAC buffer management

Simulator

We currently use Matlab to simulate the wireless interfaces available at the point of access (e.g. eNodeB) and the mobile terminal.

The performance of the proposed layer filtering process has been evaluated on *Foreman.qcif* and *Mother & Daughter.qcif* sequences at F = 30 fps (Frame Per Second). Experiments are performed using the H.264/SVC encoder (in version JSVM 9.19 [33]). The temporal period at which the control system is operating is taken as T = 1/F. The video sequences are divided into GoPs of N_G = 16 frames. The first frame in the GoP is encoded in Intra mode and the remaining frames are encoded as P-frames. Video sequences are encoded using three MGS scalability layers per frame (L = 3) corresponding to different video qualities.

Concerning the channel model, a two-state Markov model, which state switches between *bad* (B) state with instantaneous channel rate $R_c = 0$ and *good* (G) state with instantaneous channel rate $R_c = 1$ within a period T, is considered. The channel state transition are governed by the following transition probabilities P(G|G) = 0.9 and P(B|B) = 0.8 and stationary probabilities P(G) = 0.66 and P(B) = 0.33. A quite large value of P(B|B) has been chosen to simulate the bursty nature of an error-prone wireless channel.

The post-encoder buffer and the MAC buffer are assumed having respectively a maximum size Be = 25 in terms of number of frames and Bm = 500 in terms of number of PDU packets. As specified in the Radio Link Control (RLC) protocol specification of the the 3rd Generation Partnership Project (3GPP), we consider the PDU to be static with a PDU size equal to 336 bits.

In order to reduce the model state space and accelerate the convergence of the RL process, we consider only two states to quantify the fullness of the post-encoder located in the proxy (or in the server depending on the scenario), namely, overflow state and normal state. Otherwise, MAC buffer states are quantized into five intervals. The fifth interval is considered smaller than the other intervals in order to avoid MAC buffer overflow and prevent PDUs from being dropped. We assume that when the post-encoder buffer is full, frames are dropped in a head of line order (the oldest frame in the buffer is dropped first), however, when the MAC buffer is full, frames are dropped not in a head of line order but the last introduced frame in the MAC buffer causing overflow will be dropped so that the layer filtering process could know according to the buffer state if the frame will be dropped or not.

The online learning is performed using the Q-learning technique over 10000 time slots (Foreman and Mother & Daughter sequences by repeating the sequences from the beginning after 300 frames). Here, Q(s, a) is initialized to zero for each state action pair. Q-learning algorithms works by estimating the values Q(s, a) of state-action pairs (s, a). Once these values have been learned, the optimal action from any state is the one with the highest Q-value.

Results

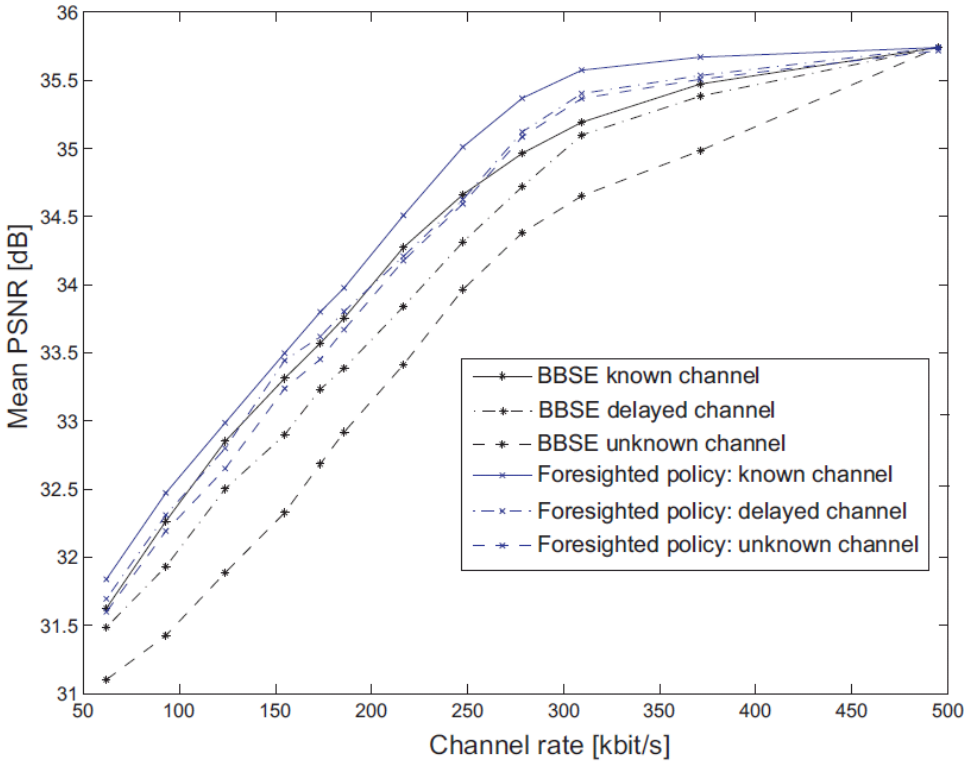
The proposed stochastic scalable layer filtering process is implemented for the three levels of knowledge of the channel states: the channel state is immediately available when choosing an action, the channel state is available with a unit delay, and no channel state is available to the controller. These three cases are performed using myopic and foresighted policies and are compared to the Basic Bit-Stream Extraction (BBSE) in the same three mentioned cases.

- *Basic bit stream extractor*

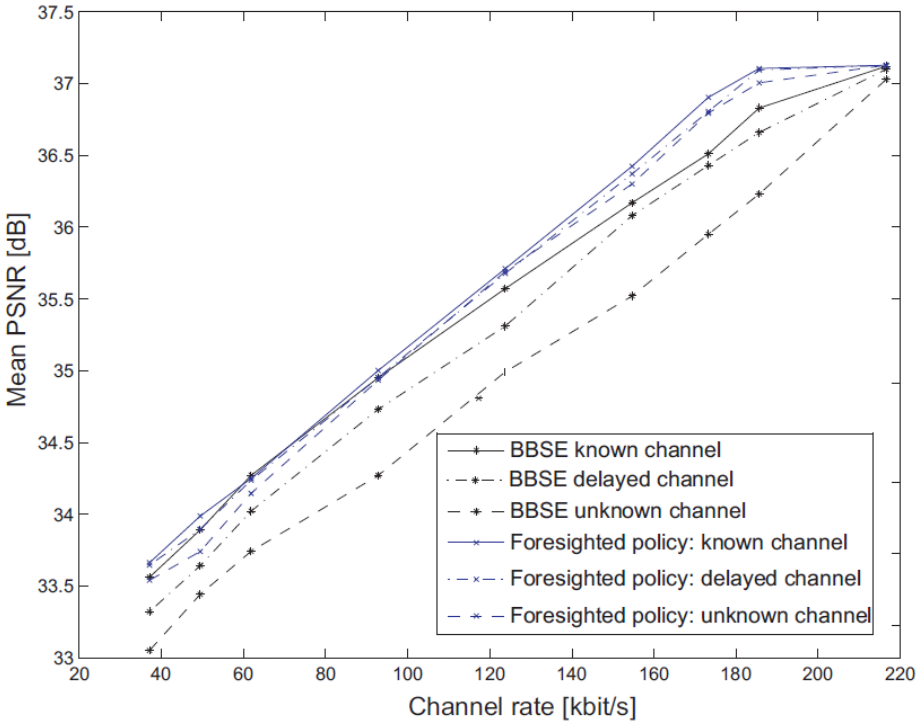
A BBSE method is a content independent extraction method provided in JSVM (Joint Scalable Video Model). The BBSE process consists in extracting SVC layers according to a specific priority order. The prioritization is done according to the high-level syntax elements dependency id: dependency, temporal, and quality id. The number of SVC layers allowed to be transmitted for each frame is selected according to the defined prioritization order and to the MAC buffer state. NAL units are ordered based on their quality level and are included in the MAC buffer in order to fill the most the empty space in the buffer without reaching the overflow state.

- *Performance*

Figure 9 shows the performance of the system under the three hypotheses concerning the channel state knowledge in the proposed scalable layer filtering process for *Foreman* and *Mother & Daughter* sequences using the foresighted policy and compared to the BBSE process. Different channel rates are considered [50−500] kbit/s for Foreman sequence and [40 − 220] kbit/s for Mother and Daughter sequence.



(a)



(b)

Figure 9: PSNR of the decoded *Foreman* (a) and *Mother & Daughter* (b) sequences under the 3 hypothesis

Channel state	Foreman		Mother & Daughter	
	PSNR (dB)	Rate (Kbps)	PSNR (dB)	Rate (Kbps)
Known	0.4	60	0.39	22
Delayed	0.54	70	0.43	33
Unknown	0.91	120	0.84	42

Table 2: Average gain in PSNR (dB) and rate (Kbps) of the proposed foresighted SVC filtering policy compared to the BBSE scheme for *Foreman* and *Mother & Daughter* sequences

As shown in Figure 9a and Figure 9b and Table 2, when the channel state information is not considered in the system model, our proposed foresighted scheme outperforms the BBSE scheme. This proves that we can improve the QoE while reducing the use of uplink resources from the terminal to the network. This feedback is typically used to convey the channel state information.
The provided gain of the proposed scheme compared to the BBSE scheme is mainly due to the accurate SNR layers selection choice which relies on the contribution of each layer to the video quality

Future Work

The next step in our research work will be the implementation on the OAI platform (ongoing). The goal is to assess performance of the proposed SVC layer filtering algorithm using a real test bed in order to overcome the weaknesses of the simulations, where the channel is modelled as a two-states Markov model, the segmentation in PDU are not compliant to 3GPP (no header insertion), etc. OAI platform offers us the advantage of benefiting from the real implementation of LTE compliant to 3GPP standard. Second, the impact of PDU losses will be analyzed on the given results. Some modification in the reward function will be conducted in order to be robust against these PDU losses.

5.3 QoE-based optimisation with network layer awareness on hybrid wireless network

Simulator

We currently use Matlab to simulate the wireless interfaces available at the mobile terminal. Thus, an LTE and a WiFi (802.11n) module have been implemented to simulate the wireless channel of a mobile user moving in a LTE cell with some WiFi hotspots available at a given distance from the LTE base station.
We implement a mobile network where we deploy a number of CDN caches on the core network side, at a given number of hops from the wireless access networks available and with a given memory occupancy.
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 optimisation algorithms based on the metrics mentioned in section 3.3.
Our test scenario is as follows. Consider a mobile user moving in an LTE micro-cell, having 20 MHz of bandwidth, and a coverage radius of up to 3 km. We further deploy 3 WiFi 802.11n APs, 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, resulting in an average speed of around 5m/s.
Values of average SNR with respect to the distance from the LTE base station are collected as plotted in Figure 10. We consider path loss and shadowing effects in both wireless models. It is evident that, once the user is in range of a WiFi hot spot (blue, red and black peaks in Figure 10), 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. It is worth noting that the figure represents average values, therefore there is no guarantee that the LTE access can always supply a good QoE, which is a good reason to exploit also WiFi whenever available. This capability of WiFi of serving as both an offload option and an overall

QoE improvement mechanism justifies the presence of WiFi hot spots even when the offered SNR is apparently not much higher than that of LTE, as happens to the first one (blue traces) in Figure 10. Finally, assuming the delivery of a scalable video, i.e., encoded with the video compression standard H.264-SVC (Scalable Video Coding) [6], we might choose to deliver the base video quality through the LTE channel, which is always available, leaving the enhancement video layers to the WiFi channel, when available.

As mentioned before, the wireless metric in use (SNR) is mapped to the channel capacity, 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 maxmin optimisation algorithms as detailed in section 3.3.

Both optimisation 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 or more. 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).

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 is to the base station, the lower the routing distance and the higher the video load in the storage; the farther the cache is 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 optimisation algorithms of Section 3.3 when: (i) the network metrics in the tuples all have the same importance (label “ALL” in the plots of Figure 11), (ii) only the CN-related metrics or (iii) only the wireless metric are used for the optimisation (label “CN” and “Wireless”, respectively, in Figure 11).

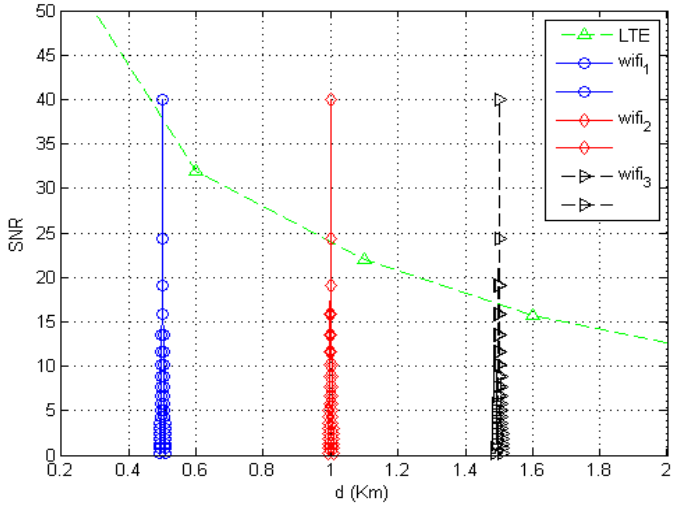


Figure 10: Wireless scenario: single user’ SNR vs. distance. An LTE base station and 3 WiFi spots, at 0.5, 1 and 1.5 Km from the LTE base station are deployed.

Results

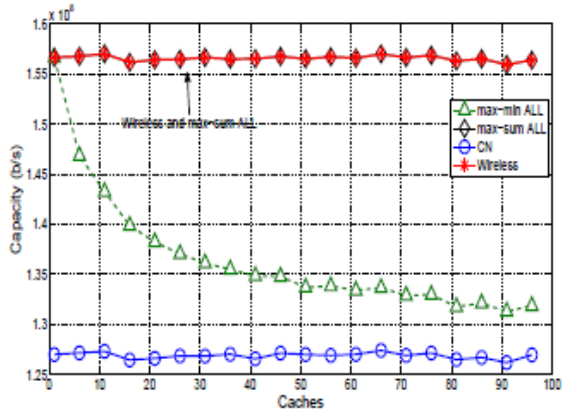
In Figure 11 we compare the performance of the two optimisation algorithms by studying the trade-off between wireless capacity and response time (delay) as the number of deployed caches increases in the network, for the balanced, unbalanced and restricted scenarios.

In the balanced scenario (Figure 11 (a) and (b)), the max-min algorithm performs worse than the max-sum algorithm in terms of wireless capacity when all metrics are used for the optimisation, otherwise they perform similarly (for this reason we present one curve labelled "CN" for both algorithms). In this scenario the best trade-off is given by max-min algorithm when few caches are deployed (minimum response time, average wireless capacity) while the max-sum "ALL" algorithm is preferable for any number of caches (maximum wireless capacity, low response time).

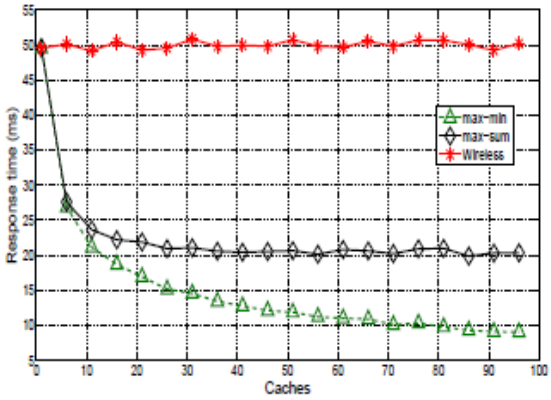
Similar conclusions can be drawn in the unbalanced scenario in terms of wireless capacity, except for a steeper decrease of the capacity for the max-min "ALL" algorithm. In terms of response time, we notice that the max-sum algorithms switch the performance with the max-min algorithms when compared to 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., in 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 striking compromise between load and routing distance, thus reducing the overall response time. In general the response time of the proposed algorithms in this scenario is detrimental to performance (nearly 15 % less than the maximum delay) compared to the balanced scenario.

In the restricted scenario (Figure 11 (e) and (f)), 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. To 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, Figure 11 (e)) and to the "CN" (bottom, Figure 11 (e)) algorithms, respectively.

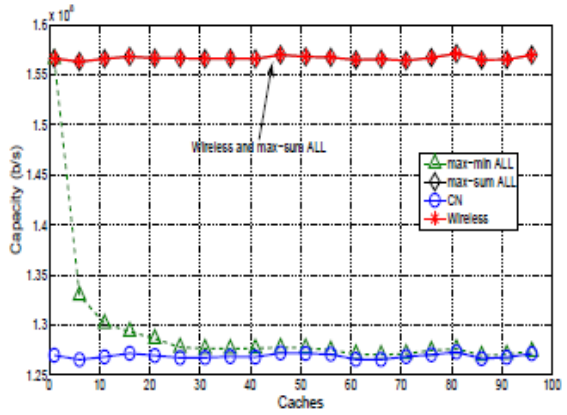
A general conclusion we can draw from the simulation results is that the algorithms optimising jointly core network and access network related metrics better find, with respect to baseline solutions, i.e., "CN" and "Wireless", the path with a reasonable good trade-off between wireless capacity and response time for a 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 aforementioned, it is possible for the operator to finely 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.



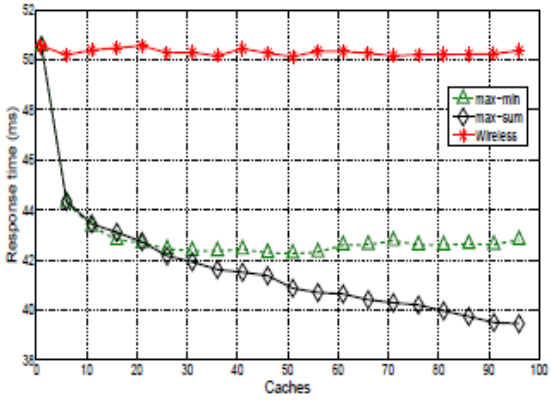
(a) Channel capacity vs. number of caches, *balanced* scenario.



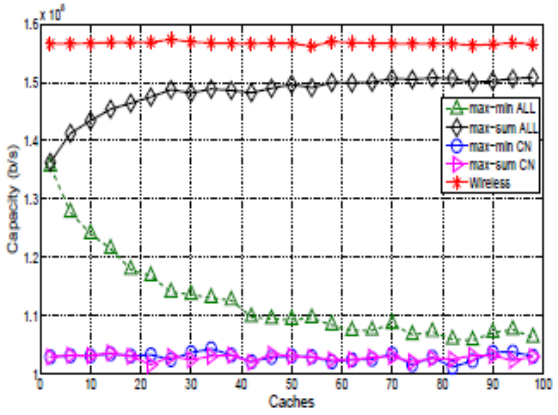
(b) Response time vs. number of caches, *balanced* scenario.



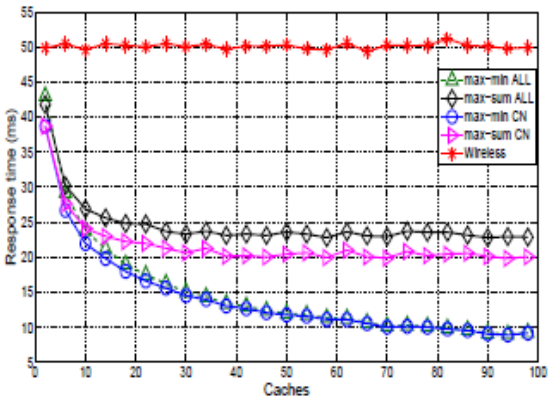
(c) Channel capacity vs. number of caches, *unbalanced* scenario.



(d) Response time vs. number of caches, *unbalanced* scenario.



(e) Channel capacity vs. number of caches, *restricted* scenario.



(f) Response time vs. number of caches, *restricted* scenario.

Figure 11: Impact of the max-sum and max-min optimisation algorithms on response time and wireless channel capacity when considering: i) "ALL": all metrics; ii) "CN": CN-related metrics and iii) "Wireless": only the wireless metric.

Future Work

An immediate next step in our research work will be the implementation of online methods for the evaluation of the most relevant metrics for the QoE and its validation in a real testbed.

5.4 Opportunistic multicast rate allocation and scheduling for scalable video streaming

We evaluate the performance of our algorithm that computes the amount of FEC needed for optimum video transmission by means of simulation. To this end, we encode some video sequences, *Foreman_cif* and *News_cif* that are commonly used for testing video transmissions, using the JSVM encoding software [15], at 30 frames per second with Group of Pictures (GoP) size 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 [6], thus we encode the video sequences into 3 video layers. The quantization points for each layer, the corresponding cumulative encoding rates and the ideal PSNR achieved when correctly receiving each layer are shown in Figure 12 for both video sequences.

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

Figure 12: Quantization points, rates and PSNR for each video layer.

We consider a scenario where the BS serves N users distributed in the corresponding area of service. Assuming a simple path loss model, where the channel conditions of each user depend on the distance between the BS and the user, 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. MCS1, covers the whole area of service of the BS, while the second MCS, MCS2, 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. A first assignment of the MCS to be used for each layer is done before the actual video streaming starts. This assignment is based on the average user distribution (i.e., the distribution of channel quality based on the fraction of users close to the BTS vs. at the cell edge) to derive a baseline solution that matches the channel constraints in terms of rate.

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 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 neighbouring 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 neighbouring MCS serving area (i.e., the number of users changing MCS 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.

Results

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 Figure 13. 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 Figure 13.

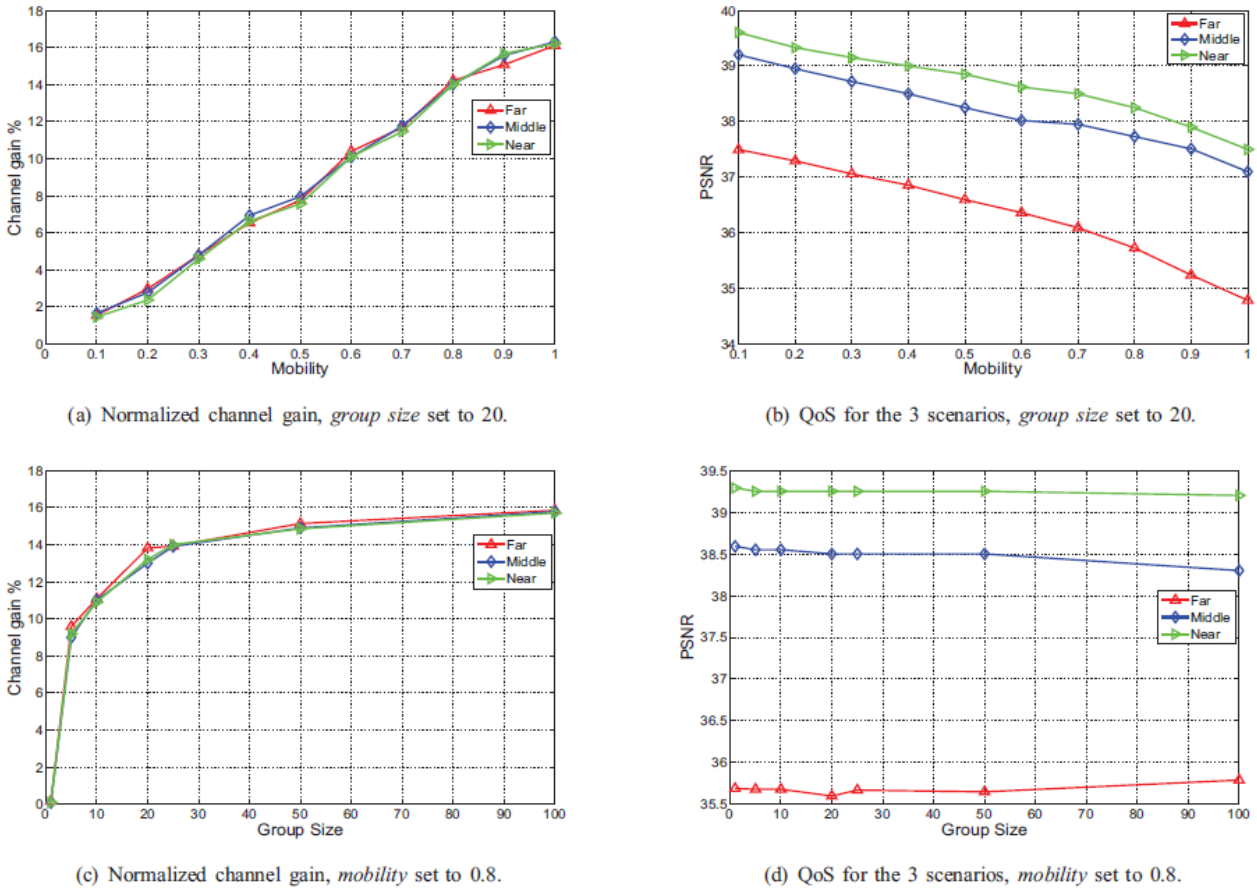
MCS	Far	Middle	Near
1	100	100	100
2	60	95	95
3	35	85	90
4	20	50	80
5	10	15	65
6	5	5	40

Figure 13: Static user distribution for each scenario.

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
Figure 14 (a) we plot how the normalized gain of channel rate varies, i.e., the percentage of 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
Figure 14 (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
Figure 14 (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 behaviour. In the case where 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 behaviour in the cell, the larger the set, 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
Figure 14 (a) and
Figure 14 (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 Figure 13).



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

Future Work

This specific work is concluded, nevertheless we might modify the aforementioned algorithms to best meet the requirements of and adhere to the interface with the other WPs.

6 Summary and Conclusions

This document focused on the final specification of the transport optimisation components and interfaces, on the ongoing research activities in Task 5.2 of WP5 and on the optimisation algorithms and cross-layer mechanisms first presented in deliverable D5.1 [3].

One of the key concepts in the MEDIEVAL project is to improve the QoE when congestion occurs through traffic engineering and cross layer mechanisms. Starting from the current standardization activities in 3GPP and IETF, the project evolves today's mobile operator network architecture, integrating sophisticated and intelligent cross layers network functions able to recognize video characteristics, to classify IP traffic accordingly and to adopt network delivery services of selected policies for improved video traffic delivery. The problem is reflected in both unicast and multicast communications.

Summarizing the key points of the work described in this deliverable, the main scientific work progress is given in Chapter 3, where we present different optimisation algorithms that improve QoE. These cross-layer algorithms, i.e., the QoE-based traffic management, QoE-based scalable video layer filtering based on MAC buffer management, QoE-based optimisation with network layer awareness on hybrid wireless network, and FEC rate-adaptation, address different issues of the video delivery chain at different levels of the mobile network. In addition, we combine several of these mechanisms into a joint optimisation algorithm for an enhanced overall performance. In Chapter 4, we provide the final specification of the WP5 modules and interfaces. Furthermore, we discuss how the algorithms presented in Chapter 3 are integrated in this modular framework. The first results from the performance evaluation of the various mechanisms are presented in Chapter 5. We report on the simulation tools and simulation scenarios we use to validate our research work.

The plan of Task 5.2 in WP5 for the next 6-month period, i.e., towards deliverable D5.4 (due date December 2012), is to provide the following further contributions. Concerning standardization within 3GPP, we plan to contribute to the User plane congestion management (UPCON) item started in SA WG1 in November 2011, to identify requirements for handling user plane traffic when RAN congestion occurs. In terms of scientific work, we will finalize the ongoing simulations and there are also several open questions that still need to be addressed.

For the QoE-driven optimisation algorithm, we are applying SVC for a more flexible rate shaping tool and implementing it in a testbed. Due to the scalability in three dimensions provided by SVC, shaping SVC streams requires an optimal stream shaping in different dimensions. The QoE optimisation will be evaluated in the testbed. We are extending our research on the temporal QoE aspects in order to present to the users a satisfactory experience during a whole video session. The QoE-based video scalable layer filtering process will be implemented on the OAI (Open Air Interface) platform for performance analysis. We will also investigate issues such as buffer mobility issue during handover, the consistency of the predicted reward compared to a delayed one, and the integration of QoE curve in the reward function rather than using PSNR metric. We further plan to implement the algorithms on QoE-based optimisation with network layer awareness in a real testbed reproducing the LTE mobile network, to study the mapping of the QoE-related performance metrics (e.g., based on the QoE models in [32]). Moreover, we will consider the delivery of scalable videos, where an operator delivers the base video quality through, say, the LTE channel (always available), letting the enhancement video layers travel through the WiFi channel (when available). Finally, we will continue the analysis of the interaction among the optimisation algorithms and finalise the joint optimisation algorithm.

Acknowledgements and Disclaimer

This work was partially funded by the European Commission within the 7th Framework Program in the context of the ICT project MEDIEVAL (Grant Agreement No. 258053) [1]. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the MEDIEVAL project or the European Commission.

References

[1] European Commission FP7 Project: "MEDIEVAL: MultiMEDia transport for mobIlE Video AppLications", http://www.ict-MEDIEVAL.eu/, retrieved June 2011

[2] MEDIEVAL Project, Deliverable D1.1 – "Preliminary architecture design", June 2011

[3] MEDIEVAL Project, Deliverable D5.1, "Transport Optimisation: Initial Architecture", June 2011.

[4] D. Munaretto, D. Jurca, and J. Widmer, "A resource allocation framework for scalable video broadcast in cellular networks". Springer Mobile Networks and Applications, December 2011

[5] N. Amram, B. Fu, G. Kunzmann, T. Melia, D. Munaretto, S. Randriamasy, B. Sayadi, J. Widmer, M. Zorzi, "QoE-based Transport Optimisation for Video Delivery over Next Generation Cellular Networks", accepted for MediaWiN workshop 2011

[6] 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, no. 9, pp. 560–576, 2003.

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

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

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

[10] S. Thakolsri, W. Kellerer, E Steinbach, "QoE-based cross-layer optimisation of wireless video with unperceivable temporal video quality fluctuation", International Conference on Communications, 2011.

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

[12] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," IEEE J. Sel. Areas Commun., vol. 18, no. 6, pp. 1012–1030, Jun. 2000.

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

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

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

[16] N. Changuel, N. Mastronarde, M. van der Schaar, B. Sayadi, and M. Kieffer. Adaptive scalable layer filtering process for video scheduling over wireless networks based on MAC buffer management. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2352 – 2355, Prague, May 2011.

[17] N. Changuel, B. Sayadi, and M. Kieffer. Video packet filtering algorithm based on only mac buffer management. *European Patent*, no. 808627, Alcatel-Lucent, November 2010.

[18] P. de Cuetos and K. W. Ross. Optimal streaming of layered video: joint scheduling and error concealment. *ACM Multimedia*, pp. 2 – 8, Berkeley, November 2003.

[19] F. Fu and M. Van der Schaar. Structural solutions to dynamic scheduling for multimedia transmission in unknown wireless environments. *Multimedia Systems and Applications Papers in Computer and Information Science*, abs/1008.4406, August 2010.

[20] D. Jurca, W. Kellerer, E. Steinbach, S. Kahn, S. Thakolsri, and P. Frossard, "Joint network and rate allocation for video streaming over multiple wireless networks," in IEEE International Symposium on Multimedia (ISM'07), 2007.

[21] 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.

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

[23] D. Munaretto, T. Melia, S. Randriamasy, and M. Zorzi, "Online path selection for video delivery over cellular networks", submitted to IEEE Globecom 2012.

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

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

[26] T. M. Cover and J. A. Thomas, Elements of Information Theory, 2nd Ed. New York, NY, US: John Wiley & Sons, Inc., 2006.

[27] MEDIEVAL Project, Deliverable D1.3: "Final architecture design", interim version: June 2012 (final version: December 2012)

[28] OPNET LTE model, http://www.opnet.com/LTE/

[29] 9900 WNG, *http://www.alcatel-lucent.com/wps/DocumentStreamerServlet?LMSG_CABINET=Docs_and_Resource_Ctr&LMSG_CONTENT_FILE=Brochures/9900_WNG_Bro.pdf&lu_lang_code=en_WW*

[30] MEDIEVAL Project, Deliverable D2.2: "Final specification for video service control", June 2012

[31] MEDIEVAL Project, Deliverable D4.3: "Final specification for mobility components & interfaces", June 2012

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

[33] N. Mastronarde and M. van der Schaar. Fast reinforcement learning for energy-efficient wireless communications. IEEE Transactions on Signal Processing, abs/1009.5(99), December 2011.

[34] MEDIEVAL Project, Deliverable D5.3, "Advanced CDN mechanisms for video streaming", June 2012.

[35] Open air Interface, Overview, http://www.openairinterface.org/

[36] MEDIEVAL Project, Deliverable D6.3: "First periodic testing report", June 2012.

[37] ITU-T, "Objective perceptual multimedia video quality measurement in the presence of a full reference," ITU-T Rec. J.247 (08/08), Tech. Rep. Rec. J.247, 2008.G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74 – 90, November 1998.

[38] G. Winkler, Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction, 2nd ed., S. Verlag, Ed. Springer, 2005, vol. 27 of Applications of Mathematics.

[39] K. Singh, A. Ksentini, and B. Marienval, "Quality of experience measurement tool for svc video coding," in IEEE ICC, 2011, pp. 1 – 5.

[40] M. Shehada, B. Fu, S. Thakolsri and W. Kellerer, "QoE-based Resource Reservation for Unperceivable Video Quality Fluctuation during Handover in LTE", IEEE GLOBECOM, Apr. 2012.

[41] V. Abdulova and I. Aybay, "Predictive mobile-oriented channel reservation schemes in wireless cellular networks", Wireless. Networks, vol. 17, no. 1, pp. 149 – 166, Jan. 2011.

[42] W.-S. Soh and H. S. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions", IEEE INFOCOM, March 2004.

Annex A Complete Updated Specification

A.1 Internal interfaces specification

In this section we describe the internal interfaces specification for Task 5.2.

A.1.1 DM_XLO_If

This interface is used to request the list of End-Points (EPs) from the XLO to the DM. Also, DM informs XLO of the handover to overcome congestions.

A.1.1.1 XLO_DM_ALTO

A.1.1.1.1 XLO_DM_ALTO.Request

Function

This service primitive is used by the XLO to request the list of EPs available in the network to the DM module, for a specific session of a mobile user.

Semantics of the service primitive

```
XLO_DM_ALTO.request(
    Session_ID,
    MN_ID
)
```

Parameter	Type	Description
Session_ID	UID	Identifier of the session
MN_ID	UID	Identifier of the mobile user

Table: XLO_DM_ALTO.request parameter list

When generated

The XLO is periodically requesting an update of the EPs available with their descriptions.

Effect on receipt

The DM provides the list of available EPs for a given session of a user to the XLO, which then runs the optimisation algorithm for selecting a video path in the network with better performance metrics, if available and necessary, using also the wireless metrics coming from the interface L25_XLO_If (4.3.2).

A.1.1.1.2 XLO_DM_ALTO.Response

Function

This service primitive is used by the DM to provide the list of EPs available in the network to the XLO module.

Semantics of the service primitive

```
XLO_DM_ALTO.response(
    Session_ID,
    ReturneCode,
    End_Points_List
)
```

Parameter	Type	Description

Session_ID	UID	The session identifier.
ReturnCode	UINT	ReturnCode of the required operation: 200 OK 500 ERROR
End_Points_List	List{UID, REAL, UINT}	Vectors of: EP ID, cache load and distance cost (hopcount)

Table: XLO_DM_ALTO.response parameter list

When generated

The DM send the EPs available with their descriptions upon request.

Effect on receipt

The XLO runs the optimisation algorithm for selecting a video path in the network with better performance metrics, if available and necessary, using also the wireless metrics coming from the interface L25_XLO_If (4.3.2).

A.1.1.2 DM_XLO_Optimise

A.1.1.2.1 DM_XLO_Optimise.Request

Function

This service primitive is used by the DM to request the XLO to perform an optimisation run (thus, using also wireless metrics) and send back the best video path available computed.

Semantics of the service primitive

```
DM_XLO_Optimise.request(
    End_Points_List
)
```

Parameter	Type	Description
End_Points_List	List{UID, REAL, UINT}	Vectors of: EP ID, cache load and distance cost (hopcount)

Table: DM_XLO_Optimise.request parameter list

When generated

The DM is periodically requesting an update of the best video path.

Effect on receipt

The XLO run the optimisation algorithm to select the best video path and provide the selected EP

A.1.1.2.2 DM_XLO_Optimise.Response

Function

This service primitive is used by the XLO to provide the best EP available to the DM module, based also on the wireless metrics.

Semantics of the service primitive

```
DM_XLO_Optimise.response(
    ReturneCode,
    End_Point
)
```

Parameter	Type	Description
ReturnCode	UINT	ReturnCode of the required operation:

		200 OK 500 ERROR
End_Point	{UID, REAL, UINT}	EP ID, cache load and distance cost (hopcount)

Table: DM_XLO_Optimise.response parameter list

When generated

It is generated after performing the optimisation algorithm in the XLO module.

Effect on receipt

The DM gets the best EP available among a list of EPs, for specific purposes within the DM.

A.1.2 XLO_CNM_If (Conceptual interface)

The main objective of this interface is to provide information about a congestion in the core network, monitored by the CNM, to the XLO module. The XLO is triggered to select the best optimisation solution to be adopted for such detected issue. This interface is not implemented within the project.

A.1.2.1 CNM_XLO_Congestion_Report

A.1.2.1.1 CNM_XLO_ Congestion_Report.Indication

Function

This service primitive is used by the CNM to notify the XLO about a congestion in the network. This message reports about flows on a congested network node and the related session IDs

Semantics of the service primitive

```
CNM_XLO_ Congestion_Report.Indication(
    Node_ID,
    List_session_flow
)
```

Parameter	Type	Description
Node_ID	UID	Identifier of the node ID

List_session_flow {
SessionID,
FlowID
}
	List{UID, UID}	List of session and flow identifiers.

Table: CNM_XLO_ Congestion_Report.Indication parameter list

When generated

Whenever the CNM detects a congestion in the network.

Effect on receipt

The CNM notifies the XLO about a congestion in the network.

A.1.3 XLO_TE_If

This interface is used by the XLO to communicate to the TE module the action to take according to some criterion based on QoE (maximize the global QoE in the congested area, QoE fairness): it can be decided to drop layers, drop frames, re-prioritize packets…etc. This interface is activated if the cross layer algorithm succeeded to find a solution to solve the congestion within the Transport Optimisation subsystem.

A.1.3.1 XLO_TE_ TrafficAdaptation.Request

Function

As soon as the cross layer algorithm finds a solution to solve the congestion while maintaining a certain level of QoE, XLO sends the new configuration of each flow ID in the congestion area to the TE. The configuration contains the target bitrate and how to achieve it.

Semantics of the service primitive

```
XLO_TE_ TrafficAdaptation.Request (
      Request_ID,
      TE_action_list
)
```

Parameter	Type	Description
Request_ID	ID	The identifier of the request

TE_action_List
{
Session_ID,
Flow_ID,
Type
Target_bitrate
Mask
}
	TLV_List
	{
	Session_ID,
	Flow_ID
	UINT
	integer
	UINT
	}
		This is the list containing the optimal decision of the XLO per flow. For each Flow_ID/Session_ID, we define the type of TE action to be performed {0: Packet scheduling, 1: Packet dropping, 2: SVC layer dropping, 3: Re-prioritize packets, 4: Transcoding, 5..255: for future use} accompanied by the target bitrate (expressed in Kbps) for each flow targeted by the XLO and the corresponding mask to apply in order to do reprioritization step.

When generated

The message is generated by the XLO (triggered beforehand by a congestion notification from the network) after finding a solution (convergence of the algorithm).

Effect on receipt

After receiving this message, the TE applies the XLO decision.

A.1.3.2 XLO_TE_ TrafficAdaptation.Response

Function

This message contains information about the TE behaviour as requested by the XLO.

Semantics of the service primitive

```
XLO_TE_ TrafficAdaptation.Response (
      Flow_Return_List,
)
```

Parameter	Type	Description

Flow_Return_List
{
Session_ID,
Flow_ID,
ReturnCode
}
	TLV_List
	{
	Session_ID,
	Flow_ID
	UINT
	}
		This is the list containing the situation per flow. If the TE succeeded to apply the instructions of the XLO, it returns a ReturnCode equal to 100 (OK) or 200 (not OK).

When generated

The message is generated by the TE to notify the XLO about the status of the action taken.

Effect on receipt

After receiving this message, the XLO decides to trigger or not other layers, e.g., the Flow Mobility to reduce the number of problematic flows (having poor QoE).

Annex B Contributions to 3GPP standardization

In the following we present the contribution to 3GPP standardization. The MEDIEVAL members are contributing to the User Plane Congestion Management (UPCON) study item through the NTT DOCOMO delegates.

B.1 UPCON study item

In the research on the QoE-based traffic management, the user plane congestion is one of the main challenges to be tackled. In 3GPP the problem of user plane congestion is being studied, started from the UPCON study item in SA1 meetings. The use cases are proposed in order to identify the requirements of the 3GPP architecture to handle the specific problems.

B.1.1 Introduction of UPCON study item

Mobile operators are seeing significant increase of the user data traffic. For operators, user data traffic has more than doubled annually for several years. Although the data capacity of networks has increased significantly, the observed increase in user traffic continues to outpace the growth in capacity. This is resulting in increased network congestion and in degraded user service experience. Reasons for this growth in traffic include the rapidly increasing use of "smart phones" and the proliferation of data applications that they support, and the use of USB modem dongles for laptops to provide mobile or nomadic Internet access using 3GPP networks. As the penetration rate of these terminals increases worldwide, this trend of rapidly increasing data traffic is expected to continue and possibly accelerate.

Network operators continue to invest in additional network capacity (network entities and connectivity resources) attempting to cope with user data traffic increases that cause user plane congestion. This additional investment is becoming increasingly costly due to the rapid and continuing increases in user data traffic. From a CAPEX or OPEX perspective, this approach is not sufficient. In addition, existing QoS and PCC mechanisms are being deployed but the full effect is still to be seen. It is therefore necessary to study approaches and mechanisms to manage user plane congestion.

This study item considers scenarios and use cases where high usage levels lead to user plane traffic congestion in the RAN, and proposes requirements for handling user plane traffic when RAN congestion occurs. The aim is to make efficient use of available resources to increase the potential number of active users while maintaining the user experience.

Scenarios that will be considered include handling of user plane traffic when RAN congestion occurs based on:

- the subscription of the user;
- the type of application;
- the type of content.

The technical report for the study item is 3GPP TR 22.805 V1.0.0 (2012-06), Feasibility Study on User Plane Congestion Management (Release 12). The document can be found in the 3GPP website: http://www.3gpp.org/ftp/Specs/archive/22_series/22.805/

B.1.2 Proposal in the study item

The proposal was made by the delegates of NTT DOCOMO based on the contributions from the MEDIEVAL members. The use case was proposed to describe an occurrence of RAN congestion. The requirements for the network functionalities to handle the congestion in this use case were identified.

3GPP TSG-SA WG1 #57	S1-120035
Kyoto, Japan, 13th – 17th February 2012	

Title	UPCON peak traffic offload use case
Agenda Item	9.8 UPCON
Source	NTT DOCOMO, NEC
Contact	Shin-ichi ISOBE(isobes@nttdocomo.co.jp)

Proposal:
Peak Traffic Load Use Case

1. Description

In this use case, a cell is congested due to the high data traffic volume generated by the usage of smart phones and from a large number of users served by the cell. During congestion, the network operator is estimating, deciding and reallocating resources to the active communications while providing sufficient service quality to the users already served by this cell. This applies also for the case of a new user being admitted to the cell.

2. Pre-conditions

Many users are connected to the mobile operators RAN covering the train station, while waiting for their trains. The cell(s) are highly loaded and congestion occurs in the RAN, when John, Mary and many others are using their smart phones for entertainment (e.g. watching videos, web-browsing, etc.). In spite of the congestion, John and Mary are able to watch the videos with good quality.
Bob is also located in the cell but is currently not active.

3. Service Flows

While John and Mary are watching video with high definition, Bob is getting active and starts watching a video. In spite of the congestion, the network shall serve the video of Bob with sufficient resources by reducing the resources previously given to John and Mary. Resources reduction and allocating the freed resources are done such that Bob, John and Mary are provided with a sufficient user-perceived video quality.

4. Post-conditions

The perceived video quality offered to Bob, Mary and John are comparable and sufficient.

5. Requirements

The following requirements are listed for this use case.

- The network shall be able to be informed about RAN congestion situation.
- The network shall be able to assess the quality of the specific communications of the active users.
- If RAN is congested, the network shall be able to decide on the resource allocation for the specific communication in order to provide sufficient service quality perceived by the new active user.
- If RAN is congested, the network shall be able to decide on the resource reallocation to the ongoing specific communications in order to provide sufficient service quality perceived by the existing active users.

The network shall be able to identify specific communication, for which the resources are to be allocated or reallocated.

