Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 28340, 14 pages
doi:10.1155/2007/28340

Research Article

Joint Optimization in UMTS-Based Video Transmission

Attila Zsiros, Attila Fülöp, and Gábor Jeney

Department of Telecommunications, Budapest University of Technology and Economics, Magyar tudósok körútja 2, 1117 Budapest, Hungary

Received 1 May 2007; Revised 1 October 2007; Accepted 11 November 2007

Recommended by Stavros Kotsopoulos

A software platform is exposed, which was developed to enable demonstration and capacity testing. The platform simulates a joint optimized wireless video transmission. The development succeeded within the frame of the IST-PHOENIX project and is based on the system optimization model of the project. One of the constitutive parts of the model, the wireless network segment, is changed to a detailed, standard UTRA network simulation module. This paper consists of (1) a brief description of the projects simulation chain, (2) brief description of the UTRAN system, and (3) the integration of the two segments. The role of the UTRAN part in the joint optimization is described, with the configuration and control of this element. Finally, some simulation results are shown. In the conclusion, we show how our simulation results translate into real-world performance gains.

1. INTRODUCTION

The rapid development of telecommunication networks moves towards the direction of an even integrated, global system. According to the traditional (ISO/OSI) approach, functions of communication are shared between network layers. Thus, every layer can be implemented independently from each other. Nowadays, however, increasing industrial and customer needs can only be fulfilled with convergence of technologies. This means (among others) that the layers cannot be perfectly separated any more: functions of layers interplay with each other. In this paper, we introduce a software demonstration platform, the purpose of which was supporting joint optimization among layers. The software enables performance testing of joint optimized application and other (e.g., physical) layers in wireless video transmission. The platform was developed in the IST-PHOENIX project (http://www.ist-phoenix.org) and is based on the project model.

The system architecture basically follows the traditional ISO/OSI model, but also has the goal of accomplishing a strategy where source coding, channel coding, and modulation parameters are assigned by a common centralized controller intelligence. We call this *joint source and channel coding/decoding (JSCC/D)*. In the traditional model, the source and channel codings are implemented separately, following the well-known separation theorem of Shannon [1]. The results of this rule are complex, but highly transparent systems. These systems are not very effective in the case of such popular applications like audio/video stream transmission [2]. Modern applications, however, often have requirements that cannot be perfectly satisfied using the traditional ISO/OSI approach. Such requirements are (among others) the real-time transmission or the unequal error protection of streams with alternating sensitivity against errors.

The architecture published in this paper can be used with different access techniques, for example, with OFDM or WCDMA. The H.264/AVC [3] and the MPEG-4 video coding are also supported in the application layer. Several transport protocols can be used (UDP, UDP-Lite, DCCP) for the transmission. The model and simulation tool developed by the project provides the opportunity to test the already-mentioned joint optimization principle in a life-like system: in our case, an UMTS network. In this paper, we focus on the system model using the UTRAN WCDMA network, as the detailed UMTS simulation environment has been developed at our university [4]. After functional introduction of the model parts, an analysis of possibilities for optimization with detailed description of configurations offered by UTRAN simulation follows. The effects of optimization on the video transmission (as a function of several different parameter settings) are also shown.

Section 2 gives an overview on the system architecture. Joint optimization issues are discussed in Section 3. Interfaces between blocks and data format at interfaces are described in Section 4. The two parts of protocol hierarchy (i.e., application and transmission modules) are detailed in Sections 5 and 6, respectively. These sections describe the modules briefly from the optimization point of view. Detailed information about RRC layer's mechanisms and control signalling is provided in Section 7; testing architecture, evaluation goals, and some results (with their interpretations) are presented in Section 8.

2. SYSTEM ARCHITECTURE OF SIMULATION MODEL

Rather than keeping the traditional approach of having each network layer work transparently from the other, in the Phoenix project it is proposed to make the endpoints aware of each other in order to perform a joint optimization of the use of available resources in the transmission chain. In practice, this means that the transmission chain components will (via the controllers) exchange information that they previously did not share with each other.

System model using UTRAN WCDMA network is shown on Figure 1: a video transmission is depicted as an example. The modules following each other represent a wired (IPv6) and a wireless (UMTS in this case) network. The wired part is the IP network "cloud," while the wireless medium is the channel block. The data transmission layers of the system correspond to the ISO/OSI model. The system is using JSCC/D control information: this controls, for example, the UEP (unequal error protection) module, or the JSCC/D adapted channel coding. Separate control layer is defined for application and physical layers that in fact only virtually differ: practically they make decisions jointly. The physical control layer is (in our case) the RRC (radio resource control) layer. The feedback controller in the application layer (JSCC/D) joins to the source coder, to the application processing module (ciphering, UEP), and to the streaming, transport and IPv6 protocols. These modules guarantee QoS needed by a real-time multimedia stream, for example, sequential delivery, and so on. The role of these modules is detailed in Section 5. The modules support information about subscriber needs and about network or channel state. This information is forwarded to the control modules that make decisions based on the received feedback information. Adaptation control sets the video coding rate, or the protection level of channel coding.

In Figure 1, the solid lines refer to effective data, payload (video), and joint information flows. The video data flows after source coding and other application processes (e.g., UEP) using streaming and transport protocols through an IPv6 network. Essential control information and protocol headers for optimization are attached to effective data packets. The information that is not synchronized to the data flow is transmitted in a separate flow (see dotted line). Controller interfaces were needed to be built to receive feedback from other modules. This feedback communication is depicted by the dotted line in Figure 1; the interfaces between the radio resource controller and the signal processing layers (PDCP, RLC, MAC and PHY) are drawn similarly. These SAPs (service access points) are defined in the standards [5], but the functionalities had to be extended to allow joint optimization.

Joint optimization is performed at two protocol levels. The first one is the primary at application level, which is a separate layer called "joint controller" in Figure 1. The other is a secondary level in the RRC layer of UTRAN, which is also using the control signals of the application level. The role of the application-level joint controller is described in Section 3; the RRC mechanism is detailed in Section 7.

3. CONTROL OF JOINT OPTIMIZATION

The joint controller (depicted in Figure 1 at both transmitter and receiver sides) plays a key role, being responsible for the optimization of the whole system. The task of the joint controller is to be aware of the global state of the system (which is represented as the union of state information that is present in different layers), exchange this information with other system layers, and jointly optimize different transmission parameters according to the system state in various layers. Note that, in the so-called "preliminary handshaking" phase (connection establishment phase), further information can be exchanged among the system blocks. This information consists of the characteristics of the system, such as the type of available channel encoders, modulator, or security options. In our case, this preliminary phase is not simulated, assuming controllers already know the capabilities of all blocks.

The joint controller on receiver side uses an error-free feedback channel to deliver parameters and measurements for controller of transmitter side. The joint controller makes decisions on the output parameters based on several input parameters (see Figure 2) [6].

The joint controller inputs needed to let the protection allocation run efficiently are the following:

(i) state information, on both network (NSI) and channel (CSI);
(ii) constraint on the total bandwidth available over the wireless channel (i.e., the target channel bitrate for compressed and protected stream, including the network headers size);
(iii) type of joint controller mode (i.e., full or reduced reference method) that will be detailed below;
(iv) feedback coming from the video encoding process (average quantization parameter, PSNR).

The most important input parameter from the source coder is SSI (source significance information). The derivation and use of SSI for joint optimization is detailed in [7]; this section only gives a short introduction for H.264/AVC codec solely. A simple semi-analytical method is proposed to optimize the protection levels on the different parts of an H.264/AVC bitstream for transmission over an error-prone channel. The model used for simulating video stream sensitivity allows the prediction of the resulting distortion depending on the channel errors (experienced by the video decoder). The model proposes to estimate the average expected end-to-end distortion $\hat{D}_{S+C}$ after the source and channel

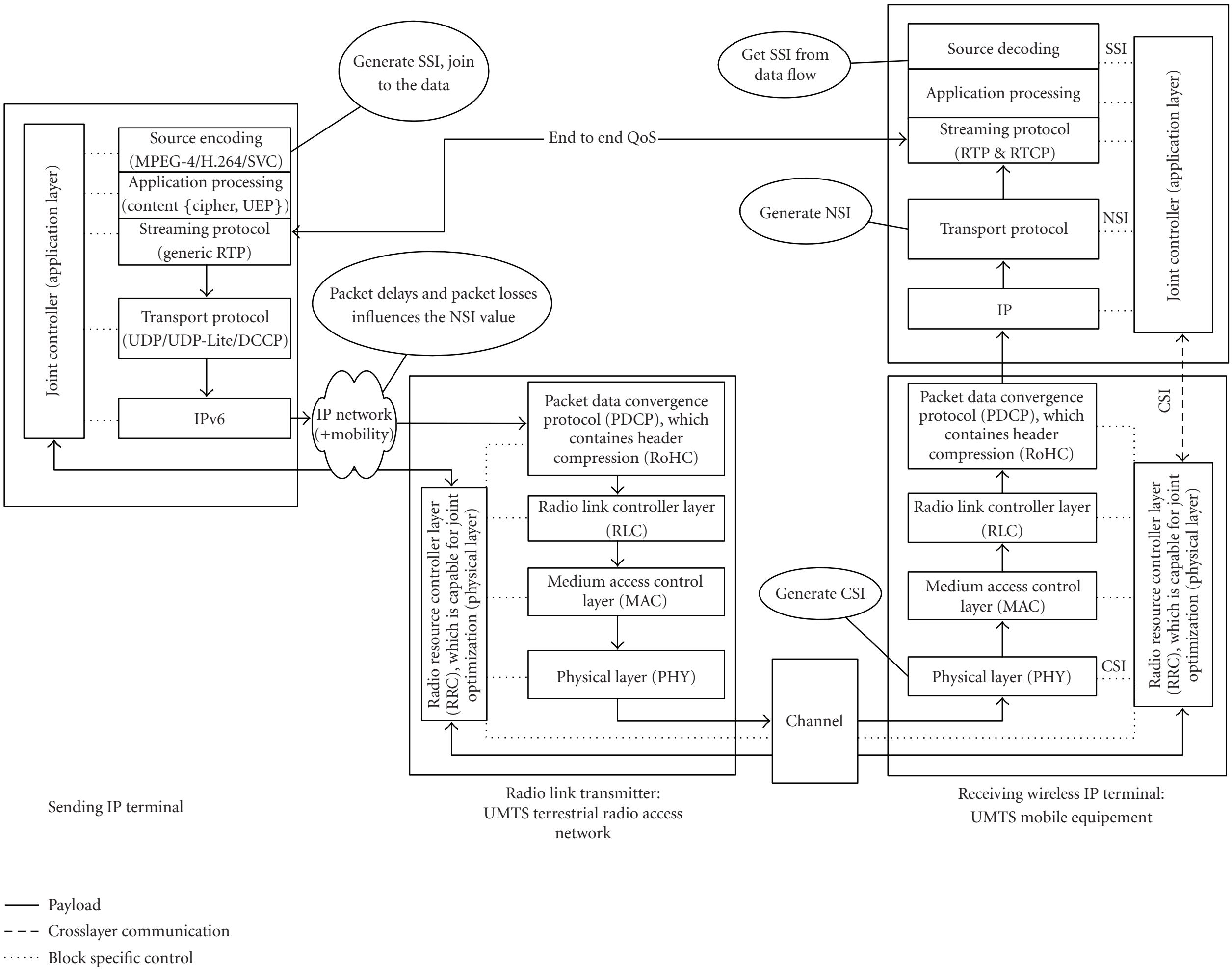


FIGURE 1: Overall system model.

coding operations for a video sequence. For the sake of simplicity, each frame is assumed coded into a single slice (or NAL (network abstraction layer) in the H.264/AVC standard).

The distortion $\widehat{D}_{\mathrm{S+C}}$ for a frame (or NAL) transmitted over an error-prone channel can be derived by taking into account the different distortion $D_i\psi$ values corresponding to the respective associated error event probability P_i:

$$\widehat{D}_{\mathrm{S+C}} = \sum_{i \in N} D_i \cdot P_i. \tag{1}$$

Instead of taking into account the impact of every single bit error and also all of their combinations, it is proposed to assume that errors can be grouped and averaged. The distortion resulting from errors in the frame can lead to the loss of the NAL with D_{loss}, or to partial corruption of the NAL with D_{corr}, and the distortion inherent to compression operation, impacting even correctly received NALs with D_o.

For $P_c\psi$ (resp., P_l) the probability to receive correctly (resp., to loose completely) an NAL, the following joint source and channel distortion, or *sensitivity* is obtained as

$$\widehat{D}_{\mathrm{S+C}} = P_c \cdot D_o + P_l \cdot D_{\mathrm{loss}} + \left(1 - P_c - P_l\right) \cdot D_{\mathrm{corr}}. \tag{2}$$

The resulting distortion is expressed in terms of MSE (mean square error):

$$\mathrm{MSE} = \sum_{i=1}^{M} \sum_{j=1}^{Q} \frac{\left(\mathrm{pl}^*(i,j) - \mathrm{pl}(i,j)\right)^2}{M \times Q}, \tag{3}$$

where M, Q are the width and height of the video frame, and $\mathrm{pl}(i,j)$, $\mathrm{pl}^*(i,j)$ are the luminance of original and reconstructed frames' pixels. Peak signal-to-noise ratio (PSNR) can be expressed as

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right). \tag{4}$$

Note that the goal is the minimization of the end-to-end distortion $\widehat{D}_{\mathrm{S+C}}$, defined as MSE, which means maximization of PSNR.

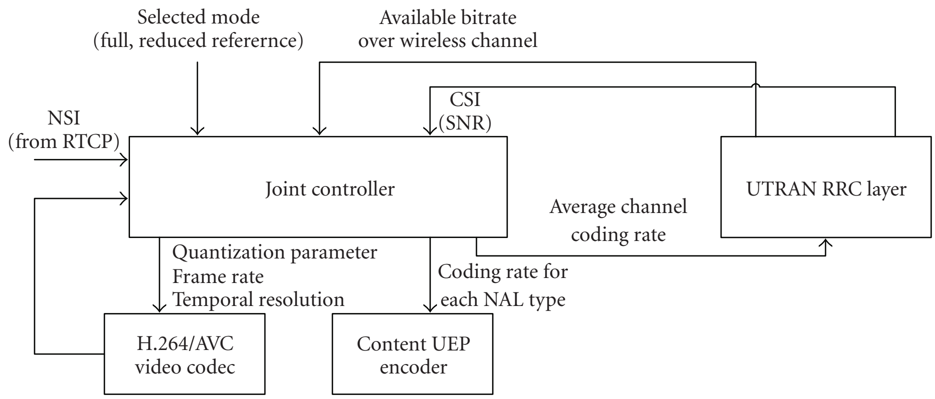


Figure 2: Joint controller inputs and outputs.

Table 1: Example state sets for joint controller.

	QP_I, QP_P	Frame rate [fps]	GOP size [frames]	Average source bitrate [kbps]
State 1	(14, 16)	7.5	8	232.92
State 2	(14, 16)	15	15	266.39
State 3a	(14, 16)	30	15	384.16
State 3b	(14, 16)	30	30	335.62
State 4	(8, 12)	15	15	414.63
State 5a	(8, 12)	30	15	607.61
State 5b	(8, 12)	30	30	532.08

Considering a memoryless erroneous channel and taking into account empirical observations, the resulting equations (see [7]) of sensitivity are derived with solely estimating the obtained distortion for the best (no transmission error) and the worst (frame lost) transmission conditions, and the frame length.

Not only the sensitivity of an H.264/AVC encoded intra- or predicted frame is deduced in [7], but also the sensitivity for a GOP, that is, group of pictures (made of an intraframe followed by N Predicted (P) frame) and the sensitivity of a data-partitioned GOP.

When the stream is data partitioned, each P frame is carried over up to three slices (NAL-A, NAL-B, NAL-C) with each slice depending on the same frame previous ones for correct decoding.

The application of the above-mentioned semi-analytical expressions is to select the best tradeoff between protection and compression for a given working point (i.e., channel SNR value), by comparing the sensitivities resulting from the different configurations of source and channel coding for a global fixed bitrate over the channel. Practically, when the formulas used with FEC protection such as RCPC (rate-compatible punctured convolutional) codes, they allow to minimize the video sequence distortion. RCPC codes offer a low complexity and allow to reach different coding rates thanks to predefined puncturing tables, offering an error event probability over an AWGN channel bounded by [8]. Consequently, the video distortion can be estimated by using this error event probability ψ value in the established expressions.

Practically, at a given channel SNR (e.g., 3 dB) the PSNR values at various source quantization parameters or at various channel coding rates can be calculated (using formulas in [7]). This allows maximizing PSNR and controlling source coding frame rate and/or RCPC codec coding rate in the UEP module. In Figure 2, channel quality appears as NSI for the wired channel and CSI for the wireless channel.

In order to reduce the dimension of the possible configurations, the joint controller has been modeled as a finite state machine (FSM) with 7 states. Each state is defined by a fixed set of parameters which control the operations performed by various blocks of the chain. Periodically, the JSCC/D controller tries to establish the best state to operate in, in order to maximize the video quality perceived by the end user while also respecting the constraints imposed by the system (e.g., block capabilities, supported data rate, etc.) [9]. (Video quality is quantified with the objective PSNR measure.)

Each state of the joint controller corresponds to a frame-rate, GOP size, and a set of quantization parameters for intra- and predicted frames (these parameters determine average source bitrate). Table 1 shows information about example state sets used in one of our simulation scenarios. When the appropriate setting has been decided, the controller launches its sensitivity estimation as described above to determine the recommended bitrate for each frame, and the corresponding protection rate to apply.

The different refinement levels generated with the frame shuffle or data partitioning approach (or any scalable coding method in practice) have different sensitivities [9]. Using the corresponding overall distortion expressions, that is, (2), it is possible to choose the best parameters of puncturing rate of RCPC for each refinement level (this is called unequal error protection) or each frame (this is called equal error protection). The SSI specifies the priority of a certain part of the bitstream and the length of that part. The video codec provides the SSI-information by marking the video stream layers according to their importance for the decoded image quality.

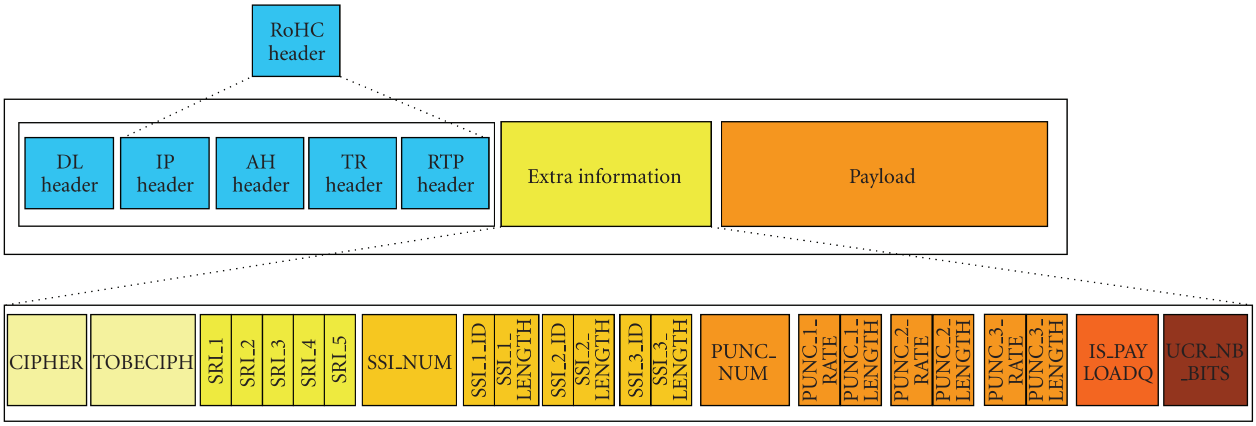


Figure 3: Binary data stream structure and extra information definition.

Summarizing, the joint controller outputs based on the inputs aforementioned are the following:

(i) source encoding parameters, namely quantization parameters (QP), bitrate, temporal resolution (i.e., frame rate), normal/frame shuffle/data partitioning mode;

(ii) content UEP/EEP coding rate for each partition, that is, for each network abstraction layer (NAL) type or refinement level;

(iii) average channel-coding rate for UTRAN RRC, to control protection of stream at physical layer level (see Section 7).

According to the input information collected from the system, the status (represented by source frame rate, quantization parameters, etc.) may be modified at each controller time step. The controller step duration can be chosen according to the selected scenario, considering wireless channel coherence time. The channel conditions should be constant in one controlling step. On the other side, the time step has to be long enough to allow source adaptation frame by frame. Furthermore, reaction time of controlling have to be considered at wireless UTRAN segment. On both transmitter and receiver sides, the setting of RRC layer must be changed using control messages. (Note that the presented 1-second value proved appropriate for joint controller time step.)

Based on the considerations detailed in [6], we will take the following limits:

max {*Frame duration, RRC controlling time*} <
Joint Controller time step <
shadowing channel coherence time.

Selected mode input parameter of joint controller (see Figure 2) is corresponding to video-quality assessment, the aim of which is twofold. It is necessary to provide real-time feedback to the sender, but more importantly it is used in the designing phase to be able to judge the effect of encoding and network parameters on the quality. Full reference methods measure "fidelity" between a corrupted and a reference undistorted image. An example of this approach is the commonly used PSNR (peak signal-to-noise ratio). PSNR assumes that the received signal is the sum of original undistorted signal and an error signal.

Mathematically this can be formulated as follows:

$$y_i = x_i + e_i, \quad (5)$$

where y_i, x_i, and e_i indicate the luminance of the corrupted, the original, and the error pixels, respectively.

Mean square error is expressed as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} e_i^2, \quad (6)$$

where N indicates the number of pixels in a video frame. The PSNR index results from (4).

"Full knowledge," for which the APP controller has full knowledge on the bitrate obtained for various quantization parameters (QP) and can as consequence set without doubt the best compromise in terms of compression versus protection by means of the sensitivity estimation function given in [6]. This mode is realistic when considering broadcasting of existing sequences that have been precoded at various bitrates, for which the controller will then choose the most adapted one for transmission over the channel at time t.

Full reference quality assessment models require to access all original image information, which need cannot always be satisfied. In the project, various reduced reference metrics are introduced as well. These methods are still very complicated and set on specific applications. In this kind of systems, two blocks, one on transmission side and one on receiver side (in our case the source encoder and decoder blocks), extract some features from original and corrupted signals and uses them to build a video quality index (for us made by joint controller). Source parameters used for quality evaluation are ideally transmitted in an undistorted channel. Practically, this information should be strongly protected from channel errors. Accordingly, our simulation uses error-free feedback channels.

Note that, in reduced reference mode, joint controller needs previous state information from the source encoder (i.e., average QP value).

4. CROSS-LAYER COMMUNICATION AND SIGNALLING INFORMATION

The reality of cross-layer communication for our simulation chain implies that the different signalling information (SSI, cipher key, etc.) is indeed transmitted together with the bitstream. In practice, the extra data is being transferred directly into a binary packet which is made of the payload obtained after video encoding and application processing (content cipher and UEP), with the addition of an extra information field viewed as an additional header, as illustrated by Figure 3. Extra information is exchanged using the IPv6 header by Hop-by-Hop option. Payload contains ciphered protected video frame (or NAL).

First two parameters are cipher key and cipher mode (enabled or disabled). The SRI contains information from the source known a priori. In practice, SRI_1, ..., SRI_5 are used only by the soft-input H.264 decoder and are neither used mandatory for hard-input H.264 decoding. IS_PAYLOADQ and UCR_NB_BITS fields also carry useful information for soft decoding. UTRAN module is not capable of transferring soft information, instead of bits of packets; accordingly, SRI is not applied. SSI_NUM means the number of SSI fields, which contain ID for priority class and length of the data part belonging to the class. PUNC fields refer to the puncturing parameter for UEP module to adjust data rate.

CSI describes the wireless channel state using measured signal-to-noise ratio. This unsynchronized feedback information can be forwarded by the ICMPv6 protocol (internet control message protocol version 6), because it entails low overhead. NSI, which contains inter-arrival jitter, average delay for packets, and packet loss rate, should be exchanged using the RTCP packets, even if the overhead introduced is slightly higher than with other schemes (e.g., ICMPv6 messages), because the RTCP packets are already exchanged between the receiver and the sender and because their format do not require any modification to include NSI information. Video quality, that is, PSNR measure is also feedback information produced after video decoding, mentioned in Section 3.

5. APPLICATION PART

In this section, layers controlled directly by the joint controller (JSCC/D) are detailed.

The source coding and decoding modules on the top of Figure 1 are using MPEG-4 or H.264/AVC codecs. Although features of these encoders are beyond standard capabilities, in this paper only H.264/AVC (advanced video coding) codec is detailed. The reason behind is that this codec fits well in our UTRAN simulation environment, and real-time wireless services with low latency and bitrate below 1 Mb/s. The rationale for choosing H.264/AVC is its design, which provides a more efficient compression when compared to the former standards (such as MPEG-2, H.263, MPEG-4), while presenting a reasonable implementation complexity versus coding efficiency ratio, and that is easily adaptable to networked applications, in particular wireless networks and internet, thanks to its network abstraction layer (NAL) structure.

The integration of the H.264 codec into the simulation chain meant the adaptation of the H.264 joint verification model (JM) version 10.1 [10] that had been developed by the ITU-T and MPEG joint video team. The H.264/AVC video codec implemented in Phoenix project using frame shuffle and data partitioning techniques in addition to standard operation. The difference introduced by the frame shuffle operation when compared to classical GOP ordering and coding process is the introduction of different dependencies among frames. Frame shuffle technique allows with a large set of shuffling patterns to envisage the adaptation of the encoding process to the video content features, as well as to the user equipment and transmission channel characteristics. This approach relies on shuffling the frames inside a group of pictures, which led to call it "frame shuffle" [11].

Furthermore, adaptation of the codec has been made in link with the controlling module to ensure that the modification of the source coding parameters can be done at each new application controller decision. The establishment of sensitivity measurements (mentioned in Section 3) allows to apply efficient error-protection scheme by UEP module.

5.1. UEP, ciphering

UEP module can produce equal error protection (EEP) for data or unequal error protection for critical parts of the data than for other less critical parts. The joint controller adjusts the UEP mode based on the information about SSI, NSI, and reduced CSI.

This module relies on RCPC codes with mother code of code rate 1/3, constraint length 5, and number of puncturing tables 9, resulting in the punctured code rates: 8/9, 4/5, 2/3, 4/7, 1/2, 4/9, 2/5, 4/11, 1/3.

Selective video ciphering algorithm is realized in the Phoenix system. It encrypts all the I frames and keeps the other parts untouched. Depending on the GOP structure of the video stream, this algorithm may lead to significant complexity reduction compared to naive algorithm. Ciphering uses a stream cipher, which can be RC4 or AES operation in counter mode.

5.2. Streaming module

On the sending site, the objective of the streaming module is the packetization of data flow into IP packets; on the receiver side, its task is the reconstruction of data flow from received IP packets for upper layers. But the IP packet means not only the IP protocol, besides the transport layer function belong here the RTP and the RTCP protocols.

RTP has been designed for real-time multimedia applications, because it provides timestamps and sequence numbers. Note that RTP itself does not provide any error detection/recovery; it is the application on top of RTP that may provide them. RTCP is used to monitor the quality of service and convey information about the participants in an ongoing session. This is achieved by sending reports between sender(s) and receiver(s). The receiver analyzes RTP header information and calculates data rate, inter-arrival jitter, and average delay for packets. The NSI parameter used by joint controller is packet loss rate (PLR) monitored by RTCP.

In the simulation chain UDP, UDP-Lite and DCCP protocols are implemented. The UDP protocol offers connectionless, best-effort service, which means no sequence numbering. Duplicate packets can also occur. The UDP-Lite protocol is the extended version of UDP protocol, which differs from the original UDP protocol with a partial CRC checksum. This partial checksum covers only the header and part of the payload data. If there is an error within the CRC covered part of the data, then the packet will be dropped. If the UDP checksum covers the whole packet, then the behavior of UDP-Lite is the same with the classical UDP. But if we protect only the header field, then can we achieve more effective functionality, because with this technique the number of discards decreases with circa 40%. Another transport layer protocol the DCCP offers is not reliable congestion controlled data flow service with acknowledgement of the correctly received data. The implemented DCCP protocol contains no possibility for retransmission of datagrams, and contains alike to the UDP-Lite protocol a partial checksum. The DCCP partial checksum covers in any case the whole header and the $n \cdot 4$ byte part of the payload. Note that the IPv4/IPv6 packet generation demands that the transport layer protocols have to use checksum, but this checksum can be partial alike to UDP-Lite or to DCCP.

The packets containing the header field of transport protocols are nested into IP packets. The simulation chain supports only the IPv6, the internet protocol version 6, because this protocol is the protocol of future internet. The European Committee pretends from every IST projects to use IPv6.

The existent effective audio/video decoders are capable of processing the erroneous packets, thus increasing the video quality. Consequently, with the above-mentioned partial checksum the number of lost packets decreases and the number of video decoder processed packets increases.

The RTP/RTCP protocols can sit on top of UDP/UDP-Lite protocols, but in the case of DCCP protocol they are superfluous. The DCCP protocol implements all the functions that make the RTP/RTCP protocols essential. The current version of simulation chain uses in every case RTP/RTCP protocols according to practical reasons.

6. TRANSMISSION PART

6.1. IPv6 network

The simulated IP network can be considered as an IP cloud with a bunch of unknown routers. This module represents the wired component of the network. Capacity and buffer size of this virtual network are configurable. The service treatment that a packet can experience at IP interface is characterized by a set of QoS parameters: delay, delay variations (jitter), and loss. End-to-end delay is modeled by gamma distribution, and uniform distribution is employed to represent drop probability. Complexity of wired network, the number of routers, and their parameters are also adjustable. The reliability of wired medium is rather high. The duration of our presented simulation run is relatively low, only 20 seconds. On the other hand, in our simulated case, enough resources are allocated for video transmission, so packet loss probability of wired segments is negligible. IPv6 network typically affects delaying packets; hence order of packets can change and transmission usually becomes more bursty at the output. Since we focus on wireless UTRAN segment, packet loss is set to minimum in wired components. During the transmission of 20-seconds long video, no packet drop occurred at the presented numerical example results.

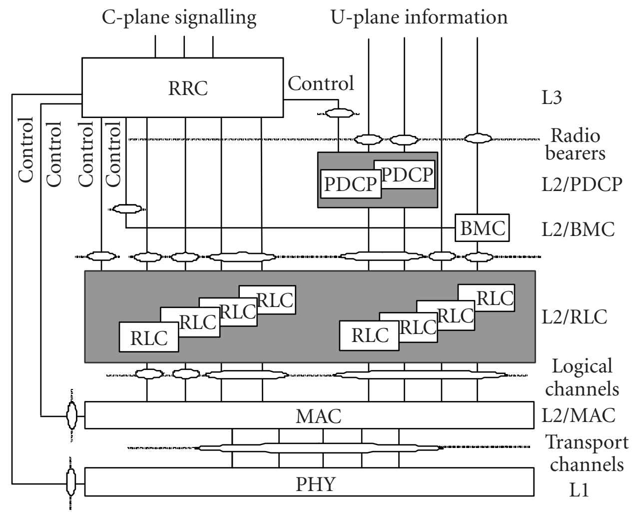


Figure 4: UTRAN protocol architecture.

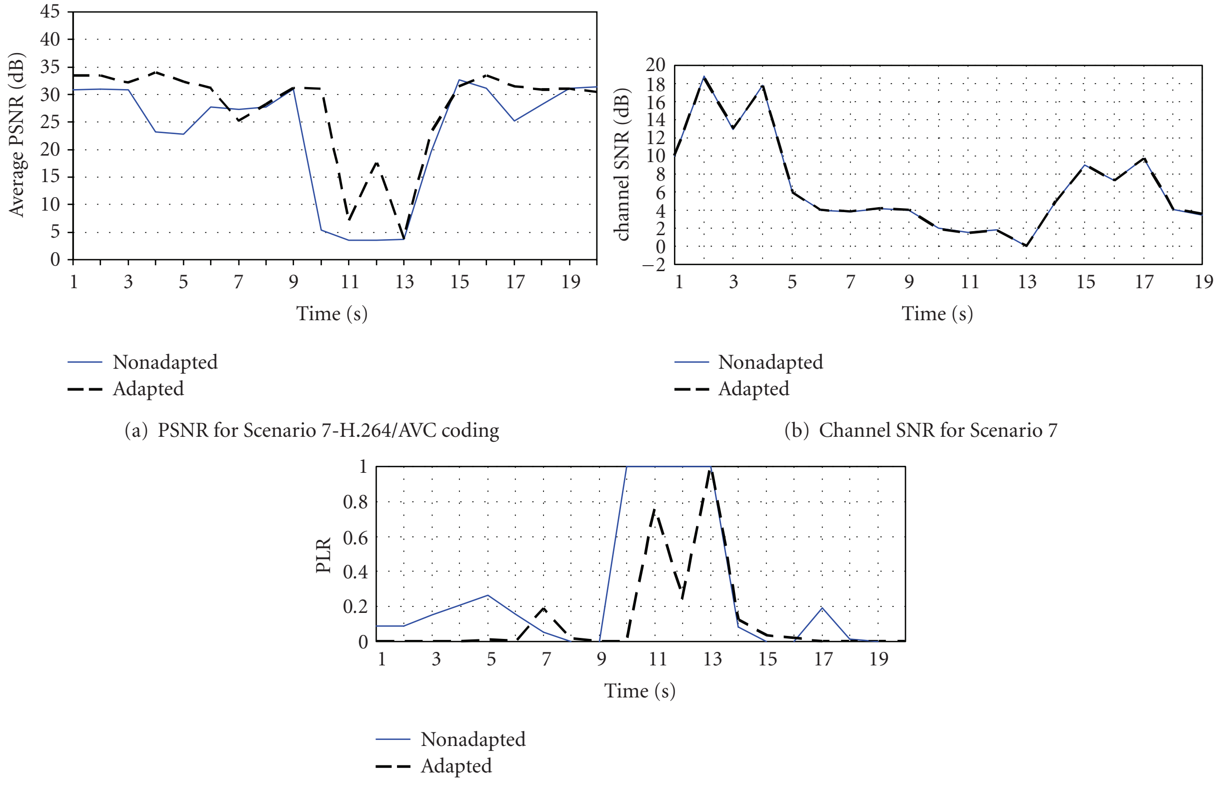
Futhermore, a mobility model is adopted at IP layer. The results show that the higher the handover frequency is, the higher the end-to-end packet loss and the packet loss rate (PLR) becomes. Its impact can be summarized into two aspects. There will be an increase in the packet loss rate, because during a handover all packets will be lost. This means that the higher the handover rate is, the higher the PLR becomes. However, according to reality when there are not handovers, a delay will be introduced due to mobility effects. The presence of application controller can improve the perceived quality also when mobility is present, because when the losses are high the source coding rate is properly reduced. We considered in this paper a low handover frequency scenario, so no handover has occurred.

When there is a network congestion, indicated by a high value for the PLR feedback in the NSI, the controller sets immediately the state to the first, characterized by the lowest source bitrate, in order to reduce as much as possible the amount of data which have to flow through the IPv6 network.

6.2. UTRAN

6.2.1. Structure and functionalities

Main considerations of designing our UTRAN modules were the following: (1) flexibility, (2) platform independent code, (3) efficient implementation, and (4) compliance with



(a) PSNR for Scenario 7-H.264/AVC coding

(b) Channel SNR for Scenario 7

(c) PLR for Scenario 7

Figure 5: Simulation results with scenario 7 [16], video sequence duration 20 seconds: average PSNR (a), radio channel SNR (b) and packet loss ratio (c).

standards. Concentrating on dedicated data transfer, functionalities are realized through [5, 12–15] standards. Additional functions, such as connection establishment, are not relevant for us, and thus they are not implemented. Figure 4 illustrates UTRAN protocol architecture. RRC layer is modified to be capable of receiving and handle JSCC/D control information and to properly configure PDCP (packet data convergence protocol), RLC (radio link control), MAC (medium access control), and physical layers.

6.2.2. *Data flow transfer*

Horizontal layers of UMTS can be divided into two planes called C- (control-) and U- (user-) planes. C- and U-planes are responsible for control and user data transfer, respectively. At the highest level, there are separate layers for control and user data transfer; at lower levels, the same layer handles both streams.

The RRC (radio resource control) layer located in the third layer (L3) controls all layers of the UTRAN, containing RLC, MAC, physical layers. RRC layer implements signalling of existing connections towards upper layers, thus making appropriate data transfer possible through the UMTS radio interface. The PDCP layer is located also in L3, but in the U-plane; this layer receives data packets from upper layers and, after robust header compression (RoHC), it forwards them to lower layers as SDUs (service data units).

Standards of the RLC layer describe three supported transfer modes, namely, AM (acknowledged mode), UM (unacknowledged mode) and TM (transparent mode). The usage of the acknowledged mode would contradict to joint optimization principle as results obtained by using JSCC/D controller could not introduce any improvement if all corrupted data packets were retransmitted. Usage of UM and TM modes is adequate for our purposes. The RLC layer segments packets received from upper layers (segmentation is the main difference between UM and TM), assigns sequence numbers (only when using UM), and forwards them to the MAC layer. UM and TM modes do not include retransmission of corrupted packets. The MAC layer maps the received PDUs (protocol data units) to transport channels (using padding if necessary) and selects an appropriate transport format, which is used to forward the data to the physical layer.

Physical layer calculates CRC for the data packets and, after channel coding, maps streams to physical channels. Data on physical channels is sent to the radio interface with QPSK modulation. Transmitted data is modeled as a complex baseband equivalent signal and is passed through a radio channel module that simulates multipath fading and adds AWGN noise to the transmitted signal. The receiver is a coherent

RAKE receiver that estimates the attenuation of the channel main signal paths optimally. After the RAKE receiver, inverse signal- and data-processing algorithms of the layers MAC, RLC, PDCP are performed.

6.2.3. Control of UTRAN module

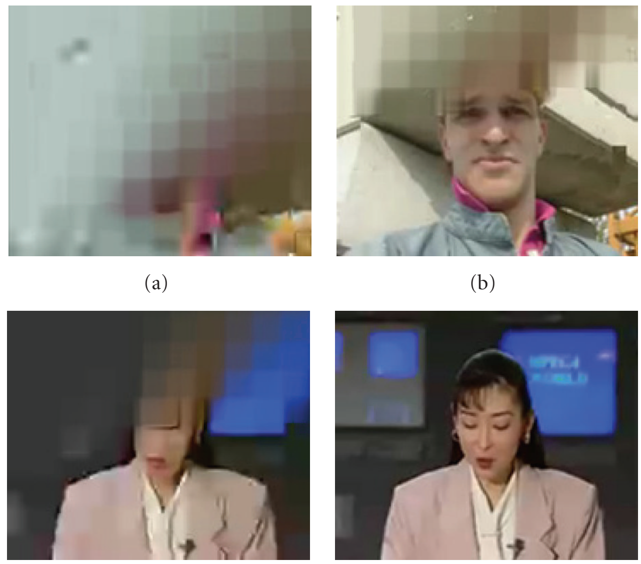
RRC (radio resource control) layer is responsible for controlling other UMTS layers. Joint optimization parameters of Phoenix simulation chain influence the operation of UMTS modules via RRC. Mode of data transfer in UTRAN is defined via the active transport format, which combines the RLC, MAC, and physical layer settings. The actually valid transport format is chosen by MAC from the configured set. The selection algorithm is based on data flow priorities and RLC buffer occupancy [4]. This primary configuration, carried out by MAC, is relatively fast, thus its period time is 10–80 milliseconds. Transport format includes

(i) type of error protection (turbo, convolutional, no coding),
(ii) coding rate,
(iii) TTI (transmission time interval), that is, the interarrival time of transport block sets (10, 20, 40 or 80 milliseconds),
(iv) amount of data in one TTI (transport block size, number of transport blocks),
(v) size of CRC (0, 8, 12, 16, 24 bit),
(vi) rate matching parameter (puncturing).

Besides, RRC layer can change the set of transport formats, from which MAC selects. This secondary configuration can be accomplished more slowly than primary, because it is required to be synchronized sets among sender and receiver. In our simulation, adaptation controlled by JSCC/D is set to one second, which time is comparably needed to enforce new transport format set.

The RRC layer chooses a configuration setting based on control information received from the JSCC/D controller. The set of configuration settings has been determined based on [13]; the set contains transport format sets, RLC layer mode, channel type, payload and header sizes in bits, maximal bitrates, and so on. Upon a single simulation run, none of the above parameters changes except for the transport format set. The above parameters are adjustable in both uplink and downlink directions. Configuration settings used in the simulations are based on values described in standards [13]: 8–2048 kbps in downlink, 8–384 kbps in uplink is available for the system.

Usage of CRC might also affect the system performance. As it has already been mentioned, concatenation of CRC codes to data units (transport blocks) takes place in the physical layer on the transmitter side. On the receiver side, packets with an erroneous CRC checksum are not forwarded to upper PDCP layer by RLC. If CRC check is disabled, UMTS will drop less PDUs, and thus more erroneous packets will reach the source decoder. Clearly, packets can still be lost, even if CRC is completely switched off, because RLC can drop them for invalid control information (sequence number, data length indicator) and PDCP can also drop them while de-



(a) (b) (c) (d)

FIGURE 6: Example of visual results obtained in simulation: non-adapted (left) versus adapted (right).

compressing RoHC header. Performance of adapted case is slightly increased disabling CRC check, but of course, this is not the only adjustment that accounts for the obtainable performance improvements. About the effect of CRC, [16] contains more detailed simulation results.

7. RRC MECHANISM AND CONTROLLING

Established expressions of sensitivity were proposed in Section 3 to select the best compromise between protection and compression for a given working point (i.e., channel SNR) supposing the same overall bitrate. This bitrate is fixed value depending on the amount of data sent at good channel conditions with no distorions, yielding low protection is needed. Source rate depends on spatial resolution, default video coding bitrate, and so on, so that the produced compressed video is satisfactory by means of a quality measure (e.g., the used objective measure PSNR). In UTRAN, the bitrate before wireless channel is directly determined by slot format [TS211] and number of DPDCHs (dedicated physical data channel). The bitrate in UTRAN after spreading the data and summing the DPDCHs is constant 3.84 Mchips/s. Bitrate before spreading R_{PHY} [bits/s] can be calculated as

$$R_{\text{PHY}} = \sum_{i=1}^{\text{Nr\_DPDCHs}} 2\cdot\frac{3840000}{\text{SF}_i}, \tag{7}$$

where Nr_DPDCHs is the number of physical channels and SF_i is spreading factors configured for channel i (determined by used slot formats).

We configure these parameters to match the default source bitrate (i.e., at good channel conditions) take into account the amount of additional data (headers, etc.). Practically, there are only 16 slot formats with rates 15, 30, 60, 120,

Table 2: Recapitulation of the considered simulation parameters.

Parameter	Value
Joint controller	
Mode	Disabled (classical) and full (adapted)
Test video sequence	
Video sequence	Foreman
Video format	CIF (352×288)
Frame rate	15 fps
Duration	10 seconds
Looping	enabled
Source coding	
source codec	H.264/AVC
Initial QP values (I, P)	14, 16
H.264 packet maximum size	180 bytes
Encoding mode	standard (cla.) & frame shuffle with tree configuration (adap.)
Content ciphering	
Mode	RC4 (key length 48 bits)
Content UEP	
Mode	Unequal Error Protection
Encoder type	RCPC with mother code $n = 1$, $k = 3$, $m = 6$
Code rates considered	8/9, 4/5, 2/3, 4/7, 1/2, 4/9, 2/5, 4/11, 1/3
Code generators (in octal)	23; 35; 27
Decoder mode	MAP
IPv6 wired network	
IPv6 network nb of nodes	10
Mean node delay	3 ms
Mean node packet loss	100 ppm
Bottleneck rate	10000 kbps
Buffer size at bottleneck	100000 bytes
IPv6 mobility	
Packet Delay mean	10 ms
Packet Delay sqr. of std. dev.	4 ms
Handover length mean	520 ms
Handover length sqr. of std. dev.	100 ms
Interval between handovers mean	820 s
Interval between handovers sqr. std. dev.	34.5 s
RoHC parameters	
Usage	disabled (cla.) & enabled (adap.)
Network headers considered	RTP/UDP-Lite/IPv6
Compression mode	unidirectional
Compression rate	average (8 bytes)
UTRAN parameters	
Class of Service	Background, streaming, generic IP packet service
RRC protection mode	Equal Error Protection
Bearer channel	384 kbps (classical) 384–64 kbps (adapted)
RLC mode	Unacknowledged
Available TBS sizes	24×336 bits, 16×336, 12×336, 8×336, 4×336, 2×336, 1×336, 0×336
TTI	10 ms
RLC PDU size	320 bits
RLC Header size	16 bits
MAC Header size	0 bit
Spreading Factor	8
Channel Coding	turbo (code rate: 1/3)
CRC size	8 bits (cla.) & 0 bits (adap.)

Table 2: Continued.

Parameter	Value
Slot Format ID	15
Number of DPDCHs	1
R_{PHY}	960 kbps
Radio channel parameters	
Environment	suburban micro-cell
Number of paths	6
Movement speed	30 km/h
Noise type	Gaussian
Noise SNR	10 dB
SNR Estimator at Receiver	Signal-to-Variation Ratio (SVR) Estimator ($N_{\text{sym}} = 4800$)

240, 480, 960, and 1920 kbps. Moreover, only few (max. 3) DPDCHs are used in practice, so the possible configuration is fairly limited.

This approximates the available bitrate over wireless channel well, although data transmitted through DPDCHs not only contain the compressed (and maybe protected) useful information, but additional control information, such as headers, joint controller extra information, and control fields added by UTRAN as well. This rate value can be fedback to the controller in the preliminary handshaking phase.

The other feedback data is the CSI (see Figure 2). Channel state indicator (CSI) means a signal-to-noise ratio (SNR) estimated with a moments-based method developed for monitoring channel quality in multipath fading channels. The estimator function [17] is valid for M-ary PSK signals (QPSK in UTRAN), so that

$$\hat{\rho}_{\text{SVR}} = \beta - 1 + \sqrt{\beta(\beta - 1)}. \tag{8}$$

The β parameter is expressed as

$$\begin{aligned}\beta = {} & \left(\left(\frac{1}{N_{\text{sym}} - 1}\right)\sum_{n=1}^{N_{\text{sym}}-1} |y_n|^2 |y_{n-1}|^2\right) \Bigg/ \\ & \left(\left(\frac{1}{N_{\text{sym}} - 1}\right)\sum_{n=1}^{N_{\text{sym}}-1} |y_n|^4 \right. \\ & \left. - \left(\frac{1}{N_{\text{sym}} - 1}\right)\sum_{n=1}^{N_{\text{sym}}-1} |y_n|^2 |y_{n-1}|^2\right),\end{aligned} \tag{9}$$

where y_n is the complex symbols after rake receiver and N_{sym} is the length of the frame in complex symbols.

The estimated value is updated for every 10 milliseconds frame, calculating the average SNR for the actual second. Calculated value is sent periodically in every 100 milliseconds to joint controller.

Average channel coding rate (r_{C}) is a feedforward control information produced by the joint controller at every optimization step. As it has already been mentioned, data protection mechanism can be carried out at the level of content by content UEP module, or, alternatively, at the physical layer of UTRAN. Joint controller adjusts the level of protection in UTRAN using this feedforward control information through dedicated signalling (e.g., ICMPv6 messages).

Channel encoding in standard UTRAN can be either switched off or one of the following three encoders can be used: convolutional coder with 1/2 code rate or code rate 1/3 or turbo coder with 1/3 code rate. When sending data through severely erroneous channels, usage of turbo codes is the most efficient. Furthermore, code rate is adjustable through puncturing and repetition functions coupling with unique code properties. So, turbo coder is an efficient choice for our test cases. We can follow equal or unequal error-protection approach, similarly like at the content level. (Protection can be carried at content and/or physical layer level. Unequal optimization can be configured at one of the two levels at a time.)

If equal error protection is followed, the channel-coding rate at physical layer (r_{p}) has to be adjusted to r_{C}. The rate r_{p} [bits/s] can be approximated using (7) with the expression

$$r_{\text{p}} = \frac{(\max\{\text{TBSSize}\}/\text{TTI}) * 1000}{R_{\text{PHY}}}. \tag{10}$$

TTI is the transmission time interval [milliseconds], and max {TBSSize} is the maximum of transport block set (TBS) sizes in transport format set of logical channel. The expression based on considering the coding rate is a quotient of bitrates before and after channel coder. Note that, in this case coder is punctured, so bitrate can be calculated after puncturing operation. Denominator of is (10) close to bitrate after coding if the amount of inserted UTRAN control bits is negligible to the amount of data bits. Nominator gives adequate value if the maximal TBS is selected by MAC in the most of the time.

Adjusting r_{p} to r_{C} means that RRC sets the maximal transport block size in the active TFS, so that $r_{\text{p}} - r_{\text{C}}$ is minimalized. This can be easily carried out while possible values of TBS size are finite. TBS size is a multiple of the RRC PDU (packet data unit) size, which is least data unit and R_{PHY} means an upper bound to it.

UEP approach requires UTRAN to extract SSI fields from arriving packets at PDCP interface. This capability is added as a function of PDCP (controlled also by RRC). Practically, header compression RoHC process is applied in PDCP too, which can separate the headers from the beginning of packets. Extra information is easily accessible while it has fixed structure with fixed length of fields.

Data can be separated to more flows of layers or partitions belonging to the same sensitivity class. The layer architecture of UTRAN depicted on Figure 4 clearly shows that more RLC entities can operate parallel. Creating RLC entities for all data flows makes it possible to transfer the sequences to MAC using different logical channels. The advantage of using different logical channels is that transfer parameters, that is, transport format sets can be distinguished for each channel. Allowed combinations of transport formats for logical channels define transport format combination sets (TFCS). The currently used transfer parameters (i.e., the used combination) are selected by MAC layer. Selection is based on the buffer occupancies of RLC entities and the priority of logical channels, which well fits the priority of sensitivity classes.

When we consider a source represented by the incoming bitstream at UTRAN PDCP interface that may be separated in layers or partitions P_i of different significance, each partition may be protected with a different channel code of rate $r_{\mathrm{p},i}$ according to its sensitivity to channel errors, which can be determined by using (10). Our goal is the minimalization of end-to-end distortion.

Each partition P_i has a source rate

$$R_{\mathrm{S},i} = \phi_i R_{\mathrm{S}} = \frac{B_i}{B} R_{\mathrm{S}}, \tag{11}$$

where R_{S} is the overall source rate, $\phi_i = B_i/B$ is the ration between the number of bits per frame of the ith partition, B_i, and the total number of bits/frame, B. The total source and channel coding rate, $R_{\mathrm{S+C}}$ is given by

$$R_{\mathrm{S+C}} = r \sum_{i=1}^{N} \frac{R_{\mathrm{S},i}}{r_{\mathrm{p},i}}, \tag{12}$$

where N is the number of partitions considered. Here we are interested in a source-dependent choice of channel coding rates for a source coded bitstream with fixed parameters. For each channel condition, for a given source rate R_{S} and a given total channel coding rate r_{C}, the problem consists in finding the channel coding rates $r_{\mathrm{p},i}$ such that the total distortion $D_{\mathrm{S+C}}$ is minimized. The constraint to satisfy [18] is

$$R_{\mathrm{S+C}} \leq \frac{R_{\mathrm{S}}}{r_{\mathrm{C}}}. \tag{13}$$

For analytical deduction and more details see [18] or [6].

8. SIMULATION RESULTS

This section presents experimental tests carried out over the Phoenix end-to-end simulation chain containing UTRAN wireless segment. Our primary goal is to demonstrate the efficiency of an end-to-end optimization of a video transmission over an UTRAN wireless link.

8.1. Practical settings of simulation parameters

In order to validate the usefulness of our approach under realistic conditions, we used seven candidate scenarios defined by the Phoenix project. Results presented here are created by using settings for each module of Scenario 7. The corresponding settings are detailed in project deliverable [16] and summarized in Table 2. This scenario represents pushed video information transfer, such as live news, which corresponds to low delay, multicast, streaming mode and mobile users. Data transfer between a mobile station, as a receiver, and a transmitter station (multicasting news) is simulated.

The raw video sequence “foreman” is CIF resolution YUV format, which is compressed with H.264/AVC picture encoding in the application layer. Frame shuffle mode is activated to ensure scalability of video content. UEP policy is activated for 4 sensitivity classes of video content. The resulting binary stream is then fed to the network layer, which performs RTP/UDP-Lite/IPv6 packetization with the insertion of the extra signalling information as detailed Section 4. This is followed by an IPv6 network emulator (which takes into account possible packet losses and delay due to possible congestions in a wired IP network) and an IP mobility emulator introducing further delays and losses due to the IP wireless mobility. RLC layer (located in UTRAN) is configured in unacknowledged mode (UM) using packet sequence numbering without retransmission of corrupted packets. As it has already been mentioned before, UM is necessary for efficient adaptation. Data errors are needed to be reduced not by retransmission, but joint optimization, reconfiguration of the whole system. In acknowledged mode, RLC layer would hide erroneous packets from higher layers, losing essential information for joint optimization. Radio channel is simulated based on [19] suburban micro-cell environment. As depicted on Figure 1, downlink transmission is carried over in UTRAN.

If the adaptation is “on,” the application layer controller will decide (based on SSI information) on both source coding compression level and radio link protection. Joint controller also takes into account side information signals (CSI and NSI continuously fed back to the transmitter side controller), optimizing the average repartition of bandwidth between compression and protection by using PSNR models for respective channels. Video encoding parameters are set once in a second—this is the time to reconfigure each module in the adapted case. In nonadapted case, JSCC/D is disabled and MAC layer selects the transport format combination to use from a configured set, which is equivalent to a 384 kbps radio bearer [5]. In the nonadapted case, 8 bit CRC checksum is set for each packet data unit. CRC is avoided in the adapted case, and similarly to classical mode, slot format number 15 is set at spreading and modulation. This enables R_{PHY} = 960 kbps overall data rate through the wireless channel, which means a spreading factor of 8 and the usage of only one DPDCH.

Both adapted and nonadapted transmissions use standard turbo channel coding. If we did not apply the same type of error correction, significant difference could be observed, for example, in the case of convolutional coding. If the adaptation is enabled, the level of protection is not static; it is determined by the level of the puncturing mechanism. Equal error protection approach is applied at UTRAN, so RRC determines in every second the available transport format set.

8.2. Numerical example results

Example numerical results are shown in this section based upon 20 seconds of transmission. Parameters and setting of test case are shown in Tables 1 and 2.

In Figure 5, the dashed line curve shows PSNR versus time in the absence of channel effect, corresponding to the coding of the video source according to the APP controller decisions at successive time steps, and representing the maximum PSNR achievable when channel and adaptation are introduced. The solid line curve shows corresponding results obtained for fixed transmission.

Average signal-to-noise ratio curves are depicted on Figure 5(b) on a simulated fading channel; solid line marks nonadapted case and dashed line is for JSCC-adapted case. The packet-loss ratio (PLR) curves—see Figure 5(c)—differ from each other, due to different usage of CRC and variant amount of transferred data. At nonadapted case, source coding operates independently from radio channel, using always the same compression level (50% of the bandwidth). But in the adapted case, video coding parameters can be varied, respectively, the amount of data on radio channel depends on UTRAN physical layer configuration.

Under good channel conditions, the two solutions can be close; however, when fading occurs, adaptation provides an improvement compared to the fixed case. On average, gains of 4 to 5 dB can be observed in this configuration.

Figure 6 shows an illustrative effect of adaptation in accordance with visual impact. “Foreman” CIF and “Akiyo” QCIF video sequences were used for the simulations. The pictures given in Figure 6 are captured for the same frame positions.

It must be noted that the UEP in classical normal mode does not actually offer much gain over the EEP in classical normal mode, in the sense where gain obtained for PSNRs lower than 25 to 30 dB are visually not really interesting for the end user. This is due to the fact that in normal mode, only two partitions exist, that does not provide enough flexibility when considering a reduced discrete number of coding rates, to better protect the intra (more sensitive) class for low SNRs and keep the predicted (less sensitive) still protected enough at medium to good SNRs. When considering more partitions, as was the case in the previous section with Data partitioning, or in the case of three levels of predicted frames (P_1, P_2 and P_3) for “tree” frame shuffle approach for a 15 frames GOP, the number of partitions is large enough to offer the flexibility needed to have the UEP mode always perform better than the EEP mode, yielding a gain of up to 5 dB in PSNR (on AWGN channel) in the range of interest.

Further results can be found in [16] regarding alternative scenarios, environments, and configurations.

9. CONCLUSION

In this paper, we have briefly expounded a system architecture for multimedia transmission over an IP-based wireless network [20]. A novel solution has been shown, where the application world (source coding) and transmission world (channel coding and modulation) interconnect efficiently with the network world (transport services, IP networking), thanks to a joint controller (JSCC/D). In the Phoenix demonstration platform [21], we changed the wireless network segment for a simulated UMTS terrestrial radio access network (UTRAN). Our investigation covered this modified architecture, which is closer to reality due to our detailed, standard-compliant UTRAN simulation environment. We described how to embed the UTRAN network segment into the simulation chain, allowing signalling mechanism between system blocks and the joint controller interface. Out approach is to prove in practice that adaptation is effectively deployable over a system that can be considered as “modern” nowadays, even though some limitations were naturally imposed by existing standards/hardware. Simulation results indicate the gains achievable by applying cross-layer design and show the usefulness of joint source and channel coding when using up-to-date wireless technology.

ACKNOWLEDGMENTS

The authors would like to thank all participants of FP6 IST-001812 PHOENIX project, who have given valuable work contribution for the development of simulation chain and have supported this work. This work was also supported by the MIK 1.2.1 project (http://www.mik.bme.hu).

REFERENCES

[1] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
[2] S. B. Zahir Azami, P. Duhamel, and O. Rioul, “Joint source—channel coding: panorama of methods,” in *Proceedings of the CNES Workshop on Data Compression*, Toulouse, France, November 1996.
[3] *Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264, ISOIEC 14496-10 AVC), Doc JVT-G050r1*, Geneva, Switzerland, May 2003.
[4] Zsiros, A. Fülöp, and G. Jeney, “Easily configurable environment of UTRAN physical layer,” in *Proceedings of the 5th EURASIP Conference Speech and Image Processing, Multimedia Communications and Services*, pp. 93–98, Smolenice, Slovakia, 2005.
[5] “Radio interface protocol architecture,” ETSI Technical Standards TS 125.301 V6.0.0, UMTS, December 2003.
[6] “Deliverable 2.4c: JSCC controller—final design and algorithm optimisation,” *Phoenix project deliverable*, IST-2003-001812, September 2004.
[7] C. Bergeron and C. Lamy-Bergot, “Modelling H.264/AVC sensitivity for error protection in wireless transmissions,” in *Proceedings of the International Workshop on Multimedia Processing (MMSP ’06)*, Victoria, Canada, October 2006.
[8] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their application,” *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 339–400, 1988.
[9] “Deliverable 2.1c: final evaluation and performance of source (de)coding schemes,” *Phoenix project deliverable*, IST-2003-001812, June 2006.
[10] http://iphome.hhi.de/suehring/tml/download, H.264/AVC Software Coordination: JM Reference Software Version 10.1.
[11] C. Bergeron, C. Lamy-Bergot, G. Pau, and B. Pesquet-Popescu, “Temporal scalability through adaptive M-band filter banks

for robust H.264/MPEG-4 AVC video coding," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 21930, 11 pages, 2006.
[12] "Spreading and modulation (FDD)," ETSI Technical Standards TS 125.213 V6.0.0, UMTS, December 2003.
[13] "Common test environments for user equipment (UE) conformance testing," ETSI Technical Standards TS 134.108 V6.0.0, UMTS, October 2005.
[14] "Physical channels and mapping of transport channels onto physical channels (FDD)," ETSI Technical Standards TS 125.211 V6.0.0, UMTS, December 2003.
[15] "Multiplexing and channel coding (FDD)," ETSI Technical Standards TS 125.212 V6.1.0, UMTS, March 2004.
[16] "Deliverable 4.4b: experiments and test results," *Phoenix project deliverable*, IST-2003-001812, December 2006.
[17] A. L. Brandão, L. B. Lopes, and D. C. McLenon, "In-service monitoring of multipath delay and cochannel interferernce for indoor mobile communication systems," in *Proceedings of the IEEE International Conference Communications*, vol. 3, pp. 1458–1462, New Orleans, La, USA, May 1994.
[18] M. G. Martini and M. Chiani, "Rate-Distortion models for Unequal Error Protection for Wireless Video Transmission," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '04)*, pp. 1049–1053, Milan, Italy, May 2004.
[19] "Spatial channel model for multiple input multiple output (MIMO) simulations," ETSI Technical Report TS 125.996 V6.1.0, UMTS, September 2003.
[20] C. Lamy-Bergot, J. Huusko, M. G. Martini, et al., "Joint optimisation of multimedia transmission over an IP wired/wireless link," in *European Symposium on Mobile Media Delivery*, Alghero, Sardinia, Italy, September 2006.
[21] M. G. Martini, M. Mazotti, and C. Lamy-Bergot, "A demonstration platform for network aware joint optimization of wireless video transmission," in *Proceedings of the 15th IST Mobile and Wireless Communication Summit*, Mykonos, Greece, June 2006.