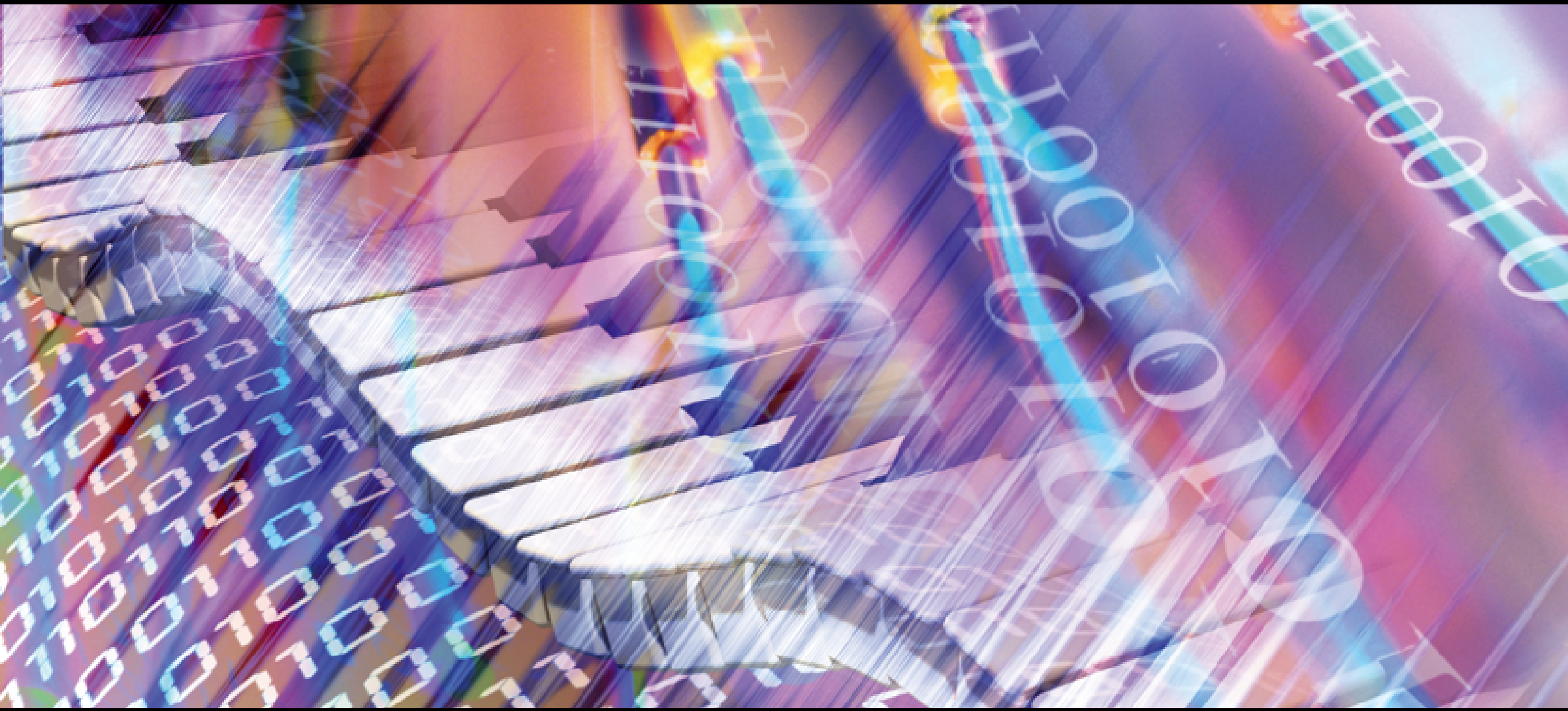
Advances in Multimedia

Multimedia Transmission over Emerging Wireless Technologies

Guest Editors: Stavros Kotsopoulos and Tasos Dagiuklas



Multimedia Transmission over Emerging Wireless Technologies

Advances in Multimedia

Multimedia Transmission over Emerging Wireless Technologies

Guest Editors: Stavros Kotsopoulos and
Tasos Dagiuklas

Contents

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 43739, 2 pages
doi:10.1155/2007/43739

Editorial

Multimedia Transmission over Emerging Wireless Technologies

Stavros Kotsopoulos[1] and Tasos Dagiuklas[2]

[1] *Department of Electrical and Computer Engineering, University of Patras, 26504 Patras, Greece*
[2] *Department of Telecommunication Systems and Networks, TEI of Mesolonghi, 30300 Nafpaktos, Greece*

Received 8 January 2008; Accepted 9 January 2008

The efficient delivery of multimedia applications and services over emerging diverse and heterogeneous wireless networks (e.g., 2.5G/3G/4G, Wi-Max, WLANs, PANs, ad hoc, etc.) is a challenging research objective. The research effort for the 3G/4G vision of interworking among heterogeneous technologies to achieve multimedia session continuity, retain multimedia QoS characteristics, and so forth amplifies the need to evaluate the conditions and restrictions under which delivery of such services can be accomplished. Furthermore, there are several research issues such as encoding techniques, cross-layer optimization, error control, security require considerable researcheffort so that appropriate solutions are designed and developed. The objective of this Special Issue is to present state-of-the-art research and developing activities contributing to all facets of multimedia across heterogeneous emerging wireless technologies (e.g., 4G, ad hoc, Wi-Max, etc.). This Special Issue aims to disseminate state-of-the-art research, development, and novel solutions of multimedia transmission over wireless/mobile-networked multimedia ad hoc systems and technologies.

The first paper entitled "WLAN technologies for audio delivery" by N. Tatlas et al. presents audio delivery over WLAN and investigates the synchronized and robust real-time streaming of multiple audio channels to multipoint receivers, for example, wireless active speakers. A novel synchronization scheme is also introduced, allowing optimized playback for multiple receivers. The perceptual audio performance is assessed for both stereo and 5-channel applications based on either PCM or compressed audio signals.

The second paper entitled "Efficient TTI for 3G multimedia applications" by C. Chaikalis addresses time transmission interval (TTI), which is a very important task for implementing optimum UMTS turbo coding in flat Rayleigh fading environment. Different multimedia scenarios are investigated by C. Chaikalis, using maximum UMTS frame length. It is shown that different operating environments require appropriate TTI in terms of BER performance for a variety of data rates.

The impact of data transfer service, multimedia streaming service, web service, and traffic load on speech quality in an environment of WLANs (IEEE 802.11) has been examined at the third paper entitled "Impact of background traffic on speech quality in VoWLAN" by P. Pocta et al. In their paper, a new method for improving the detection of the critical conditions in wireless networks from the speech quality point of view is presented.

The forth paper entitled "Fuzzy logic control of adaptive ARQ for video distribution over a Bluetooth wireless link" by R. Razavi et al. considers adaptive ARQ, suitable for video transmission over Bluetooth links. The authors present a fuzzy logic control of ARQ, based on send buffer fullness and the head-of-line packet's deadline. The scheme considers both the delay constraints of the video stream and at the same time avoids send buffer overflow. Tests explore a variety of Bluetooth send buffer sizes and channel conditions.

The fifth paper entitled "Utilizing cross-layer information to improve performance in JPEG2000 decoding" by H. Persson et al. focuses on wireless multimedia communication and investigates the scenario of how the cross-layer information can be used to improve the performance in image quality at the application layer using the IPEG2000 standard.

The sixth paper entitled "Joint optimization in UMTS-based video transmission" by A. Zsiros et al. presents a new software platform to enable demonstration and capacity testing within the framework of IST-PHOENIX project. The developed platform simulates a joint optimized UMTS video transmission.

The seventh paper entitled "Energy-constrained quality optimization for secure image transmission in wireless sensor networks" by W. Wang et al. proposes a new cross layer approach to optimize image transmission quality for secure digital image delivery in wireless sensor networks (WSNs) with strict energy budget constraint. First, a selective encryption approach favorable for UEP communication schemes with cipherplain-text diversity as well as position-value (P-V) diversity is proposed. Network resources such as desirable BER, ARQ retry limit, and transmission rates are jointly optimized across PHY, MAC, and APP layers regarding distortion reduction distribution bounds and inter-segment correlation for image data transmission.

Metamodeling comparison issues are involved in the eight paper entitled "A comparison performance analysis of QoS WLANs: approaches with enhanced features" by I. Papapanagiotou et al. The contribution of their paper is to compare and enhance known methods for performance analysis of the IEEE 802.11e MAC layer such as Markov chains, queuing theory, and probabilistic analysis. The proposed analyses carry by themselves scientific interest because they are extended enhancements with the latest enhanced distribution coordination (EDCA) parameters.

The ninth paper entitled "Video broadcasting using queue proportional scheduling" by D. Toumbakaris and S. Kotsopoulos examines the use of queue proportional scheduling for video broadcasting. First, the behavior of QPS is examined as the scheduling frequency is reduced and a method is proposed that uses statistics on the arrival rates to improve its performance. The reduction of the scheduling frequency simplifies the scheduler and decreases the required operations. Using a Markov chain to model packet delay distribution, it is discussed how the video encoding rate can be chosen in order to reduce the expected distortion of streams transmitted through broadcast channels.

The tenth paper entitled "Distortion optimized packet scheduling and prioritization of multiple video streams over 802.11e networks" by I. Politis et al. presents a generic framework solution for minimizing video distribution of multiple video streams transmitted over 802.11e wireless networks. Their work includes intelligent packet scheduling and channel access differentiation mechanisms. Two intelligent scheduling algorithms have been proposed that will select which packet or combinations of packets over all video streams will be dropped based on the available transmission rate of the communication network, which in this case is 802.11e WLAN.

Stavros Kotsopoulos
Tasos Dagiuklas

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 12308, 16 pages
doi:10.1155/2007/12308

Research Article

WLAN Technologies for Audio Delivery

Nicolas-Alexander Tatlas,[1] Andreas Floros,[2] Thomas Zarouchas,[1] and John Mourjopoulos[1]

[1] *Audio Technology Group, Department of Electrical and Computer Engineering, University of Patras, 26500 Patras, Greece*
[2] *Department of Audio Visual Arts, Ionian University, Plateia Tsirigoti 7, 49100 Corfu, Greece*

Received 21 April 2007; Revised 30 August 2007; Accepted 27 December 2007

Recommended by Tasos Dagiuklas

Audio delivery and reproduction for home or professional applications may greatly benefit from the adoption of digital wireless local area network (WLAN) technologies. The most challenging aspect of such integration relates the synchronized and robust real-time streaming of multiple audio channels to multipoint receivers, for example, wireless active speakers. Here, it is shown that current WLAN solutions are susceptible to transmission errors. A detailed study of the IEEE802.11e protocol (currently under ratification) is also presented and all relevant distortions are assessed via an analytical and experimental methodology. A novel synchronization scheme is also introduced, allowing optimized playback for multiple receivers. The perceptual audio performance is assessed for both stereo and 5-channel applications based on either PCM or compressed audio signals.

1. INTRODUCTION

In typical home and professional applications, digital audio can be delivered from any source to single or multiple receivers, through local area networks (LANs). Thus, the interconnection between devices may be simplified, and the communication between audio sources, receivers, and other multimedia devices may be optimized [1]. Additionally, in this case, an Internet connection could transparently be considered as an additional source with enhanced features, such as audio on demand. A further improvement would be the employment of wireless local area networks (WLANs). The first obvious practical benefit of using a WLAN is that interconnection cables are eliminated and, depending on the application, a number of wireless transceivers (access points—APs and wireless Stations—STAs) can be installed and appropriately configured for realizing any required audio delivery scenario. An additional advantage is that the same WLAN infrastructure can also service data transmissions between personal computers and other digital devices; hence, such systems will be compatible with a wide range of applications and eventually will present extremely flexible and cost-effective alternative to the present home entertainment chain.

A number of wireless audio products already exist in the market, starting from analog systems operating in the area of 800–900 MHz (e.g., wireless microphones, in-ear monitors, and loudspeakers) up to proprietary wireless digital streaming technologies. In such systems, the wireless transmission protocol is application specific for reducing the implementation complexity and cost. This restricts equipment compatibility and raises interoperability issues between different manufacturing designs, often to the extent that the concept of networking is defied. To overcome such compatibility issues, established wireless networking standards should be employed, such as Bluetooth [2], HyperLAN/2 [3], HomeRF [4], and the IEEE 802.11 family of protocols [5]. Among them, the latter specification currently represents the most promising scheme for wireless audio applications, due to its wide adoption and the continuous ratification process which will provide significant enhancements in many state-of-the-art networking aspects, such as security and adaptive topology control.

Despite the recent advances on transmission rates (the 802.11 g specification [6] offers a theoretical maximum of 54 Mbps, while the upcoming 802.11n draft extends this rate up to 270 Mbps), the existing "best-effort" nature of WLAN protocols introduces practical limits for real-time audio open-air streaming. To overcome such constraints, the transmission protocol must provide quality-of-service (QoS) guarantees [7]. Additionally, although QoS represents the major requirement for point-to-point multimedia streaming applications, the development of wireless multichannel audio products (e.g., for home theater) raises the challenging issue of synchronization between the wireless receivers.

A number of techniques for network time synchronization have been proposed for audio over IP networks [1]. Generally, there are two such synchronization issues: (a) local clock synchronization problems due to a (variable) offset between the hardware clocks of the remote receivers, and (b) intrastream (packet ordering and timing in one stream) and interstream (temporal relationship among different streams) synchronization, both defined in [8]. Considering a two-channel system, usually local clock synchronization issues may be manifested as audio pitch modification, change of source directional perception, or even audible noise for rapidly changing clock offset [9]. Additionally, loss of intrastream synchronization may be perceived as gaps and phase mismatches during reproduction, while interstream synchronization may lead to two channels being perceived as separate sources and shifts in the acoustic image spatial position.

A previous published study [9] has shown that local clock synchronization can be efficiently achieved in the application layer using the available hardware, leading to a maximum clock drift of less than 0.1 ms. On the other hand, interstream synchronization of high-quality multichannel audio still represents a very challenging research field, since the methods already presented mainly focus on intrastream synchronization for single-channel voice applications or interstream synchronization for point-to-point multichannel applications [10].

The aim of this study is to give an in-depth overview of all the issues related to audio WLAN delivery, focusing on two areas: (i) the effect of the wireless environment on the overall audio playback quality and (ii) the issue of interstream synchronization for real-time playback of high quality, (uncompressed and compressed) audio multichannel streams over a WLAN platform. The wireless transmission protocol considered in this work is the well-established 802.11b standard with the QoS enhancements defined in the latest version of the IEEE802.11e draft amendment [11]. Additionally, a novel interstream synchronization technique is introduced for synchronizing discrete audio channel playback among a number of wireless loudspeakers, using typical, off-the-self, transceiver hardware. In this way, the study is not bounded by product-specific protocols and implementations.

The rest of the paper is organized as follows. Section 2 provides a general background on wireless networking and QoS for audio applications. The architecture of the system employed and a theoretical timing and distortion playback analysis is presented in Section 3. In the same Section, a novel method (termed as Consistent Delay Synchronization—CoDeS) is introduced, which compensates for any playback distortions introduced by any network interchannel variable delay. Section 4 presents the test methods employed and examines the timing error results and the audibility for the various tests described. Finally, the conclusions of this work are summarized in Section 5.

2. WIRELESS NETWORKS FOR AUDIO APPLICATIONS

There are two classes of audio applications that can be supported by a WLAN.

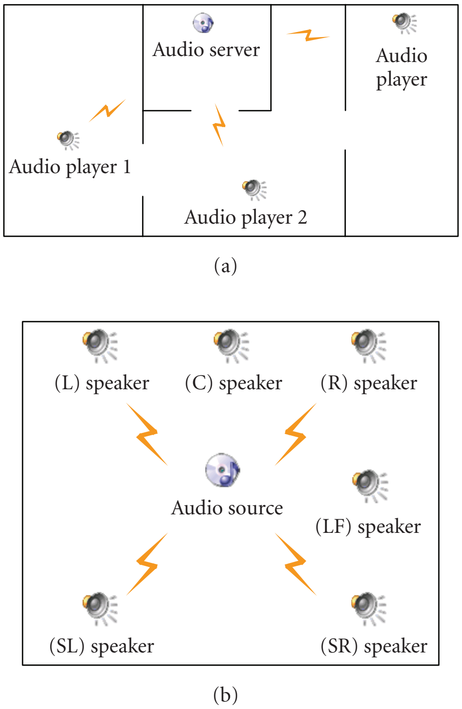


Figure 1: Typical WLAN digital audio systems: (a) simple point-to-point wireless audio delivery, (b) wireless multichannel playback setup.

(a) Simple point-to-point home audio delivery (see Figure 1(a)) where an audio server wirelessly transmits in real time the same or different audio streams to a number of wireless audio players/receivers. No synchronization between the wireless receivers is necessary, while the maximum allowed number of remote players is dynamically adjusted by the QoS bandwidth reservation algorithms. Using the above setup, any certified WLAN-enabled audio device (including portable audio playback, laptop computers, and consumer electronics equipment) can be directly connected to the network and receive audio data on user demand. Currently, a number of products operating in the S-Band ISM (2.40–2.48 GHz, available worldwide) and in the C-Band ISM (5.725–5.875 GHz, available in some countries) exist, including Bluetooth audio applications [12], complete wireless home networking setups [13, 14], as well as wireless headphone solutions [15]. Moreover, integrated home theater systems employing wireless surround channel reproduction via a single point–to–point link have been recently introduced in the consumer electronics market [16]. However, most of these systems employ compressed quality or even analog audio and are based on proprietary transmission protocols and technologies, and hence are incompatible to the emerging WLAN standards.

(b) Wireless point-to-multiple receivers (see Figure 1(b)), where typically 6 loudspeakers of a 5.1 channel home theater setup can be wirelessly connected to an audio source. In this case, the digital audio source transmits audio data to the appropriate wireless loudspeaker which should perform simultaneous and synchronized (relative to all other receivers) playback in real time. Hence, local (i.e., hardware) clock as well as packet playout synchronization methods are required for eliminating unpredicted channel shifts and phase distortions. Additionally, in this case, the pre- and power amplification modules can be assumed to be integrated within the loudspeaker and it is likely that digital audio amplifiers can be employed for greater power efficiency, better system integration, as well as reduced size and cost [17].

A WLAN-based multichannel audio system will additionally benefit from high-level procedures for automatic receiver position discovery. These will be able to take into consideration lower protocol layer metrics currently being defined by the 802.11k enhancement in order to allow each wireless loudspeaker to define its' function (e.g., left, right channel, etc.) within the multichannel setup. Moreover, application-layer mechanisms must be also supported for allowing the user to control several playback parameters, such as relative channel volume, delay, and so on. All these aspects represent challenging topics which may allow in the near future the replacement of typical wired speakers with novel WLAN-enabled active speakers.

2.1. *Quality-of-service over WLANs*

As it was previously mentioned, the accepted term for providing time-critical services over a network is quality of service (QoS), which refers to the capability of a network to constantly provide specific service guarantees. An introduction to QoS issues for streaming audio over WLANs can be found in [18], where a very early version of the 802.11e draft amendment is described. However, during a long ratification process, additional QoS characteristics were introduced and are briefly described here. The current 802.11e specification [11] defines two access types. Using "priority differentiation," all transmitting devices contend for the wireless medium under the rules defined by the enhanced distributed channel access (EDCA). On the other hand, when "resource reservation" is used, all wireless transmissions are centrally controlled, following the so-called hybrid controlled channel access (HCCA) rules.

A previous work [19] has defined the minimum transmission requirements for wirelessly distributing CD-quality and multichannel audio using EDCA. In this work, the HCCA mechanism is considered which, to the best of the author's knowledge, is not yet exploited for high-quality audio applications, while, according to [20], it achieves QoS performance for real-time multimedia traffic. A detailed description of HCCA is out of the scope of this work and can be found in [21]. Briefly, under HCCA, an STA transmits only upon the reception of a polling frame sent by the AP. The allowed transmission time lengths are calculated using a number of traffic specifications (TSPECs) declared by the STAs upon their service initialization. Typical TSPEC parameters are the traffic mean/maximum data rate, the corresponding packet length, and the physical (PHY) transmission rate. 802.11e additionally includes the simple scheduler (SiS) description for defining the minimum requirements of any HCCA service scheduler. An alternative scheduler [22] is also considered here, termed scheduling based on estimated transmission times-earliest due date (SETT-EDD), which aims to improve QoS performance under variable transmission conditions.

2.2. *Wireless multichannel audio system topology*

The general architecture of the wireless multichannel system considered here is shown in Figure 2 and consists of (a) one digital audio source integrated with a wireless QoS AP transceiver forming a wireless digital audio source (WiDAS). Appropriate buffering stages are also employed in order to transform the digital audio samples stream into packets of appropriate length. (b) A number ($M \geq 2$) of wireless digital audio receivers (WiDARs). Typically, these systems can be considered as wireless self-powered loudspeakers, containing a wireless subsystem which recreates the audio stream from the received packets.

Briefly, the basic functionality of the above setup is as follows. The WiDAS transmits the digital audio and control information to the WiDARs. In case of linear PCM-coded audio, each audio channel is transmitted to the appropriate WiDAR, based on the identification information provided by the topology detection procedure. For compressed-quality audio, it is likely that all the audio channels will be multiplexed in a single digital stream. Hence, the WiDAS can broadcast the audio information to all WiDARs, each being responsible for decoding the transmitted stream and for selecting the appropriate audio channel for playback. However, the lack of an adaptive 802.11 medium access control (MAC) layer retransmission mechanism when broadcasting, renders the overall transmission quality inadequate. Thus, unicast transmissions are generally preferred.

The WiDAS controls all wireless transmissions and attempts to sustain the appropriate mean data rate necessary for real-time audio reproduction. However, due to the stochastic link conditions, the instantaneous throughput and transmission delay values may significantly differ. As human hearing is highly delay sensitive, prebuffering mechanisms must be employed [23]. Long buffering queues can better compensate for variable channel conditions; however, there is a trade-off between the perceived user interaction latency and the buffering length. These restrictions force the buffering stages to be of predetermined and finite size, which, as it will be explained in the next section, may result in audible distortions during real-time playback.

3. WIRELESS TRANSMISSION TIMING ANALYSIS

To study the wireless packet-oriented transmission of digital audio streams and any possible errors induced, a source-to-receiver timing analysis will be now introduced. Initially, assuming uncompressed audio PCM streams, the WLAN audio source produces audio samples at a constant rate of $N f_s$ (bytes/s), where N is the byte resolution per sample and f_s (Hz) is the sampling rate. The transmitter software assembles

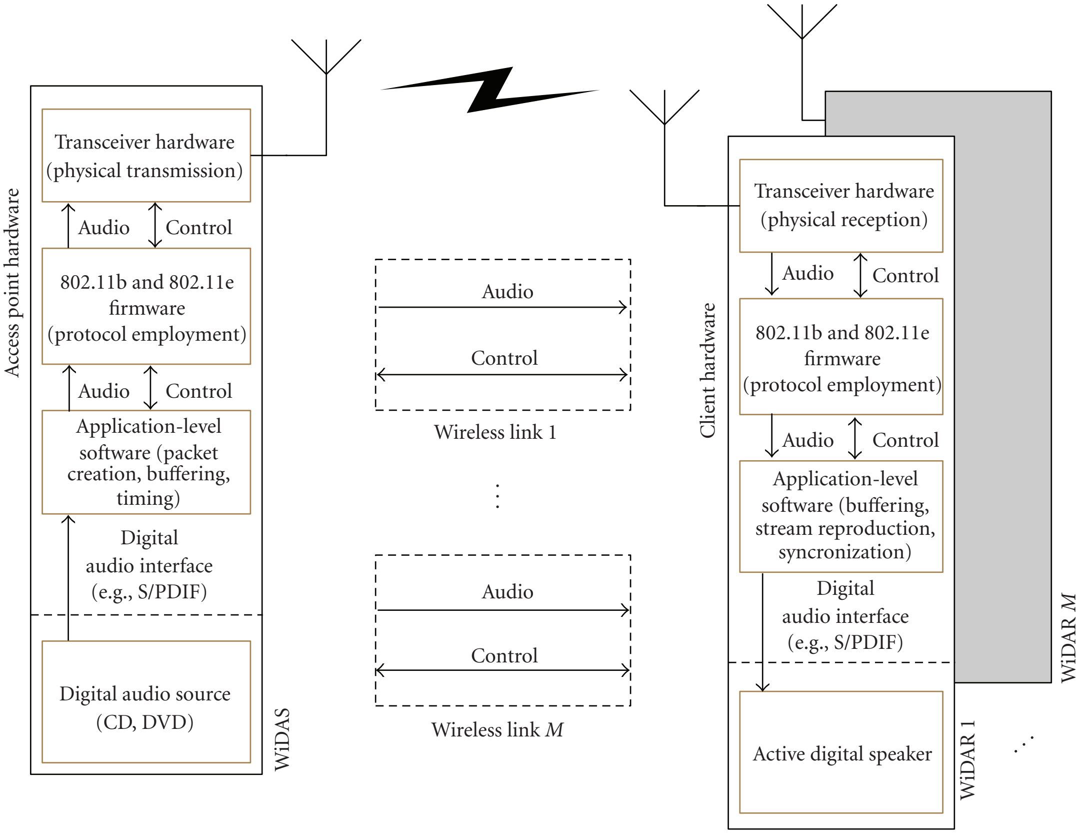

Access point hardware
Transceiver hardware (physical transmission)
Audio
Control
802.11b and 802.11e firmware (protocol employment)
Audio
Control
Application-level software (packet creation, buffering, timing)
Digital audio interface (e.g., S/PDIF)
Digital audio source (CD, DVD)
WiDAS
Audio
Control
Wireless link 1
⋮
Audio
Control
Wireless link M
Client hardware
Transceiver hardware (physical reception)
Audio
Control
802.11b and 802.11e firmware (protocol employment)
Audio
Control
Application-level software (buffering, stream reproduction, syncronization)
Digital audio interface (e.g., S/PDIF)
Active digital speaker
WiDAR 1
WiDAR M

Figure 2: Architecture of the WLAN multichannel playback system considered in the study.

appropriate number of samples, to form packets of length L_p (bytes). Thus, each packet is generated every

$$T_g = \frac{L_p}{Nfs}(s). \tag{1}$$

For compressed audio, such as MPEG-1 Layer III streams, the audio source produces data at predefined intervals of

$$T_g = \frac{L_p}{b}(s), \tag{2}$$

where L_p is the frame length in bytes and b is the predetermined compressed audio total bitrate (bps). Hence in both cases, the ith packet generation time is quantized in multiples of T_g.

3.1. Wireless audio transmission

Upon generation of each audio packet, the packet is inserted into the WiDAS transmission (Tx) buffer at instances $t_g(i)$. However, assuming that no upper-layer recovery mechanism is employed, if the buffer is full, then the packet will be dropped. This is statistically equivalent to applying packet-aging functions during transmissions, meaning that each packet in the queue will be deleted if it is not successfully transmitted in a predefined time interval. Thus, buffer overflows caused by transmission errors lead to permanent packet losses. It should be also noted that, despite the lossy wireless transmission, out-of-order packet arrival cannot occur, as no alternative routing nodes exist. Hence no packet reordering mechanisms are necessary, and a buffered packet will be transmitted after the transmission of all the previously queued packets. Assuming that $d_{\mathrm{Txbuffer}}(i)$ is the time delay between the packet insertion in the Tx queue and its transmission to the receiver, then the transmission time instance $t_{\mathrm{Tx}}(i)$ for the ith packet will be equal to

$$t_{\mathrm{Tx}}(i) = t_{\mathrm{g}}(i) + d_{\mathrm{Txbuffer}}(i). \tag{3}$$

Furthermore, the delay $d_{\mathrm{Txbuffer}}(i)$ between the packet insertion in the Tx queue and its successful transmission depends on the number of packets already in queue, and hence the actual transmission delay of a given packet will be $d_{\mathrm{Tx}}(i)$, as shown in Figure 3,

$$d_{\mathrm{Txbuffer}}(i) = d_{\mathrm{Txbuffer}}(j) - (i-j)T_{\mathrm{g}} + d_{\mathrm{Tx}}(i), \tag{4}$$

where j is the index of the immediately preceding packet successfully inserted in the Tx queue.

$d_{\mathrm{Tx}}(i)$ represents the time interval elapsed between the movement of the ith packet to the first position of the Tx queue and its successful transmission. This delay depends on the subsequent polling of the corresponding stream, resulting into successful packet transmissions, depending on the transmission physical rate, transmission time overheads, retransmission delays, and the scheduler type. Here, the propagation delay between the transmission and the reception

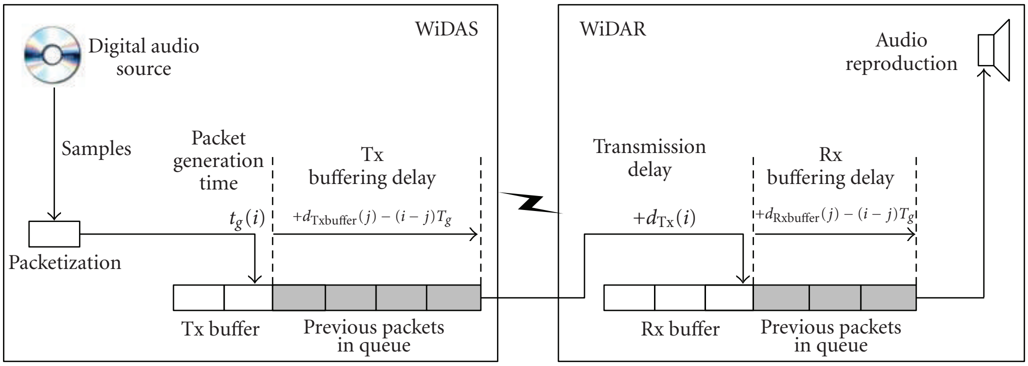


FIGURE 3: Representation of delay generation in packet-oriented WLAN audio transmission.

point will be considered to be negligible compared to such delays (e.g., retransmission, etc.), something that is largely true especially for high physical rates.

3.2. *Wireless audio reception*

Upon reception, each packet will be placed in the reception (Rx) buffer, as long as there is sufficient space, at instances $t_{\text{Rx}}(i)$ equal to $t_{\text{Tx}}(i)$. If the Rx buffer is full, then this incoming packet will be permanently disregarded. Each buffered packet will be successfully reproduced provided that all previous packets successfully inserted in the Rx queue have been reproduced. Then, the time instance for the ith packet reproduction is equal to

$$t_p(i) = t_{\text{Rx}}(i) + d_{\text{Rxbuffer}}(i), \tag{5}$$

where $d_{\text{Txbuffer}}(i)$ is the delay between the packet insertion in the queue and its reproduction (also shown in Figure 3)

$$d_{\text{Rxbuffer}}(i) = d_{\text{Rxbuffer}}(j) - (i-j)T_{\text{g}} + T_{\text{g}}, \tag{6}$$

where j is the index of the last packet in the queue before the insertion of the ith packet.

To denote the successful packet insertion, either in the Tx or Rx queue, a function $\delta_s(i)$ can be *defined* such that

$$\delta_s(i) = \begin{cases} 0, & \text{if either queue is full} \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

Using the above equations, the packet reproduction time $t_p(i)$ can be defined as

$$t_p(i) = \delta_{\text{s}}(i)\left[t_{\text{g}}(i) + d(i)\right], \tag{8}$$

where

$$d(i) = d_{\text{Txbuffer}}(i) + d_{\text{Rxbuffer}}(i) \tag{9}$$

is the total delay induced by the WLAN between packet generation and final playback. Packet losses will cause the playout delay of succeeding packets to fluctuate, since the corresponding t_{Txbuffer} and t_{Rxbuffer} values will appropriately change, as will be described below.

If playback prebuffering is employed, then the first packet reproduction will be delayed in the reception queue for a predefined amount of time, $t_{\text{prebuffering}}$, causing the first packet received to be delayed by this time so that

$$d(k) = t_{\text{prebuffering}}, \tag{10}$$

where k is the first packet successfully inserted into the reproduction queue; typically, $k = 1$. It is obvious that this delay propagates to all subsequent packets.

3.3. *Time-related network distortions*

If a packet with index i is dropped from either the Tx or Rx queue ($\delta_s(i) = 0$), a discontinuity in the reproduction will occur, which may be audible during audio playback. Moreover, in the case of a stereo or multichannel audio setup, this would additionally cause loss of channel synchronization with at least one channel leading, unless a compensation strategy is employed in the system, such as inserting an empty packet in the queue. As will be shown in Section 4, this condition can appear under heavy network load. In practice, for a given PHY transmission rate, excessive MAC layer retransmissions may occur and/or, for the case in which the transmitting application is attempting to compensate for previous packet losses through retransmissions, the overall required channel data rate will increase. If the reproduction has actually been halted when receiving the packet, that is,

$$t_p(i) - t_p(j) > T_{\text{g}}, \tag{11}$$

where j *is* the previously reproduced packet, then another kind of audible distortion will take place, since playback interrupts (silence gaps) will be introduced. In this case, at least one channel will be lagging, unless the other receivers momentarily stop playing audio data as well. For a given packet, the above two distortions can be combined, by writing

$$t_p(i) - t_p(j) > T_{\text{g}} \quad j \neq i-1. \tag{12}$$

In this case, depending on the value $(i-j-1)$ giving the packets lost and the value $(t_p(i) - t_p(j) - T_g)$ giving the time elapsed from stopping the playback, the receiver will be either leading or lagging, or even in phase with the other receivers.

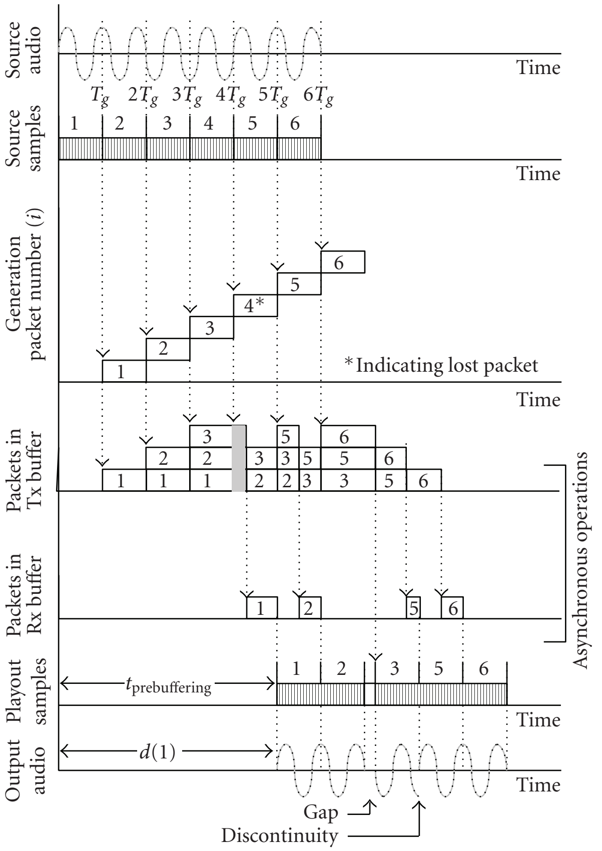


Figure 4: Example of timing analysis in packet-oriented digital audio WLAN transmission.

In order to assess the total amount of data excessively delayed, the concept of delayed throughput is introduced. In general, throughput is described as the sum of data successfully sent by a source over the transmission time. According to the 802.11e, a transmission delay bound must be defined for any traffic source. Hence, the amount of data arriving after this bound, over their transmission time, is termed here as delayed throughput (T_D, measured in Mbps) and defined to be equal to

$$T_D = \frac{1}{I_{\text{total}}} \sum_{i=1}^{I_{total}} \frac{L_p}{d_{\text{Tx}}(i)} \quad \text{for } d_{\text{Tx}}(i) > \max \text{SI}, \tag{13}$$

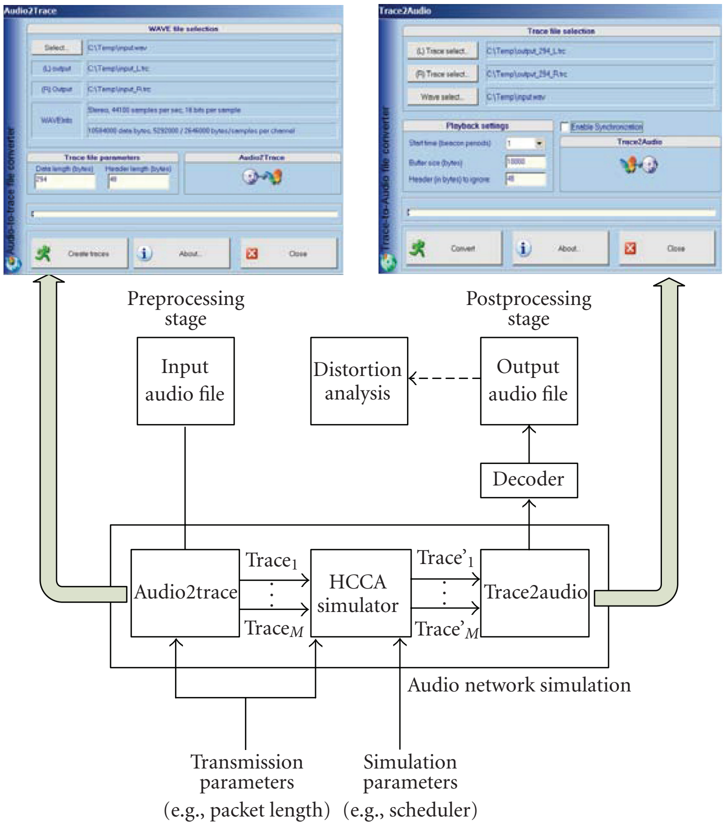
where I_{total} is the total number of packets transmitted and max SI is the maximum service interval, defined in [11]. In case that $T_D \neq 0$, reproduction distortion will be introduced (i.e., relative channel delay, discontinuities, and silence gaps).

An example for reproduction gaps and discontinuities due to the wireless transmission is shown in Figure 4. Here, for illustrative purposes, the Tx and Rx buffer size is set to 3 packets. As shown, playback commences after the predetermined prebuffering time $t_{\text{prebuffering}}$, and while packets #1 and #2 are accurately reproduced, excessive transmission delay for packets #1, #2, and #3 causes a gap in reproduction. Moreover, packet #4 is disregarded because at the time it is pushed in the transmitter buffer, the Tx Buffer is full, causing a discontinuity in audio reproduction. Note that the transmission delay is *not taken under consideration*, thus a received packet is available for reproduction on the instance it is extracted from the Tx buffer and inserted in the Rx buffer.

3.4. Synchronization strategy

The preceding analysis indicates that in the case audio data packets are lost or excessively delayed, an application-level compensation strategy is necessary in order to ensure synchronized reproduction of all receivers at any given time, even if prebuffering is employed. The algorithm should have low signaling complexity (expressed in terms of additional packet exchange required for achieving synchronization) in order to ensure that no substantial network overhead is induced. As already mentioned, local (hardware) clock synchronization is not addressed here, since it has been shown to be efficiently achieved in the application layer using the available wireless transmission hardware [9].

The consistent delay synchronization (CoDeS) strategy proposed here is based on the adjustment of the packet delay so it remains consistent and independent of networking parameters and conditions. Ideally, when prebuffering is employed, the playout delay for each packet should be constant and equal to $t_{\text{prebuffering}}$. The following paragraphs explain how the CoDeS strategy compensates for variable delays utilizing information from each packet header, for the case of (a) buffer overflow and (b) buffer underflow.



Audio2Trace

WAVE file selection

Select... C:\Temp\input.wav

(L) output C:\Temp\input_L.trc

(R) Output C:\Temp\input_R.trc

WAVEInfo: Stereo, 44100 samples per sec, 16 bits per sample; 10584000 data bytes, 5292000 / 2646000 bytes/samples per channel

Trace file parameters: Data length (bytes) 294; Header length (bytes) 48

Trace2Audio

Trace file selection

(L) Trace select... C:\Temp\output_294_L.trc

(R) Trace select... C:\Temp\output_294_R.trc

Wave select... C:\Temp\input.wav

Playback settings: Start time (beacon periods) 1; Buffer size (bytes) 10000; Header (in bytes) to ignore 48

□ Enable Synchronization

FIGURE 5: Block diagram of the WLAN audio delivery simulation.

(a) If any packet drop occurs due to buffer overflows, then the delay of the next packet in the reception queue will be reduced, that is,

$$d(i) < t_{\text{prebuffering}}. \tag{14}$$

The corresponding lost packet playback time must be then compensated for, by adjusting accordingly $d_{\text{Rxbuffer}}(i)$ to $d'_{\text{Rxbuffer}}(i)$ ensuring synchronized reproduction of all subsequent packets, that is,

$$d'_{\text{Rxbuffer}}(i) = d_{\text{Rxbuffer}}(i) + d(i) - t_{\text{prebuffering}}. \tag{15}$$

(b) If excessive delay is introduced in the transmission path causing buffer underflow, then a packet might be reproduced after its predetermined playback time. In this case, the packet has to be disregarded. Thus, if

$$d(i) > t_{\text{prebuffering}}, \tag{16}$$

then, it must be $\delta_s(i) = 0$.

After adjusting the delays according to the above equations, the receiver may employ digital audio editing signal processing for error concealment, such as proposed in [24, 25], in order to ensure acceptable signal continuity.

4. TEST METHODOLOGY AND RESULTS

In order to realize the complete real-time wireless transmission and playback process and to evaluate any playback distortions, a computer-based test methodology was developed [26], which is illustrated in Figure 5. It consists of three main subsystems: (a) the simulation preprocessing stage, which converts the digital audio data into inputs to the HCCA simulator, (b) the HCCA simulator [27], conforming to the mandatory HCCA functionality defined in the latest 802.11 e draft specification [11], and (c) a simulation postprocessing stage which processes the simulator's output and produces a new file containing a "reproduced" PCM version of the original digital audio data. The proposed CoDeS synchronization algorithm is also incorporated in the postprocessing subsystem to derive the "synchronized" version of the received data.

The reproduced audio file can be used for evaluating distortions for each receiver, by comparing the reproduced data to the corresponding original input waveform, as well as for evaluating possible synchronization loss between these data. The methodology described here for stereo (compressed and uncompressed) digital audio data can be easily extended to more channels (e.g., 6 audio channels as used in the DVD format). A detailed description of the functionality of the three subsystems is provided in the following paragraphs.

4.1. Simulation parameters

The HCCA simulator (see Figure 5) uses trace files for external traffic modeling of any required traffic flow. A trace file describes a traffic stream in terms of the resulting packets as a function of time, a technique used in the past to model variable video traffic [27]. In this work, although the audio data rate is always constant (for either compressed or linear PCM), trace file modeling was employed for mapping the transmitted data packets to specific segments of typical audio files.

The mapping of the original audio file to trace files was performed using an application developed by the authors of [26] called Audio2Trace, as shown in Figure 5. For the case of uncompressed (PCM) audio transmission, the conversion parameters include the total duration, the audio channel bitrate, the packet header length (see below), and the user-defined pure audio data packet length L_p (in bytes) which was set equal to 294 and 882 bytes during this work. Note that the above packet length selection was implied by the requirement of deriving a transmission schedule in whole submultiples of the beacon interval. On the other hand, for compressed transmission (e.g., MPEG-1 Layer III (mp3)), the conversion parameters are all extracted from the input file header, with the pure audio data packet length being equal to the MPEG frame length. In both cases, the UDP transport protocol was employed, which adds an 8 byte header on all transmitted packets [10], while 40 additional header bytes are reserved for future control purposes (e.g., RTP encapsulation). Taking into account the derived data bitrates for stereo linear PCM signal (1.4 Mbps for uncompressed CD-quality audio), the legacy, low-cost IEEE802.11b protocol was selected with a PHY rate equal to 11 Mbps, while a custom retransmission scheme is employed in the MAC layer. However, as explained in Section 3, packet losses may occur due to buffer overflow.

The wireless channel models employed during the simulations were obtained through measurements of real-world 802.11b-based transmission patterns in a controlled environment, with the wireless stations being at 2 m distance, under no interference (also referred as good channel) and medium interference (also referred as medium channel) induced by two neighboring stations in the 2.4 GHz frequency band. More specifically, the number of the total transmissions (including retransmissions) and the total transmissions that resulted into successful delivery were measured within every beacon period and the successful delivery probability was calculated as a function of time (expressed in multiples of the beacon period). A channel model plug-in for the HCCA simulator was finally implemented, that takes into account the resulting probability values and applies them to the transmissions taking place within each simulated HCCA TXOP.

The HCCA simulator models the WiDAS and WiDAR 802.11 MAC layer functionality, assuming that the WiDAS device has a 5 Kbyte prebuffering stage, typically used in such applications, for every serviced audio stream. The simulator produces one output trace file per serviced traffic (audio channel) stream containing information for all the corresponding data packets sent, such as packet transmission confirmation as well as the packet delay induced. This information is used by the simulation postprocessing stage (implemented by the Trace2Audio application) for deriving a new wave file representing the received (playback) version of the source audio data.

Using the Trace2Audio application, the receiver buffering stage is implemented using a first-in first-out (FIFO) reception queue with user-defined length. This queue is gradually filled with the successfully received data packets and is getting empty in a sample-by-sample basis, as the audio samples are read at a rate equal to the original PCM sampling frequency f_s (Hz). If the Rx queue is empty, then the output sample values are set to zero. Furthermore, a user-selectable initial latency for prebuffering purposes has been considered, in order to decrease the audible consequences of jitter in packet arrival. This latency was set in multiples of 100 ms, which is the beacon transmission period defined by the legacy IEEE 802.11 specification. For the test cases considered here, the Rx queue length was always equal to 10000 bytes and the initial latency was set equal to 1 beacon interval (100 ms).

In the case of compressed audio transmission, an external decoder was employed in the final post processing stage. The application initially discriminates between correctly received frames and erroneous data caused by excessive delays. The decoder is used to process the correctly received frames and decode them to PCM samples, while erroneous or missing data are directly mapped to zero-value PCM digital audio samples, depending on the initial encoding bitrate and sampling rate.

The proposed CoDeS synchronization scheme described in Section 3, in practice, requires only metadata information for the packets, included within each header and timing information from the WiDAR. CoDeS has been included as an option in the Trace2Audio application modifying (when necessary) the Rx queue contents.

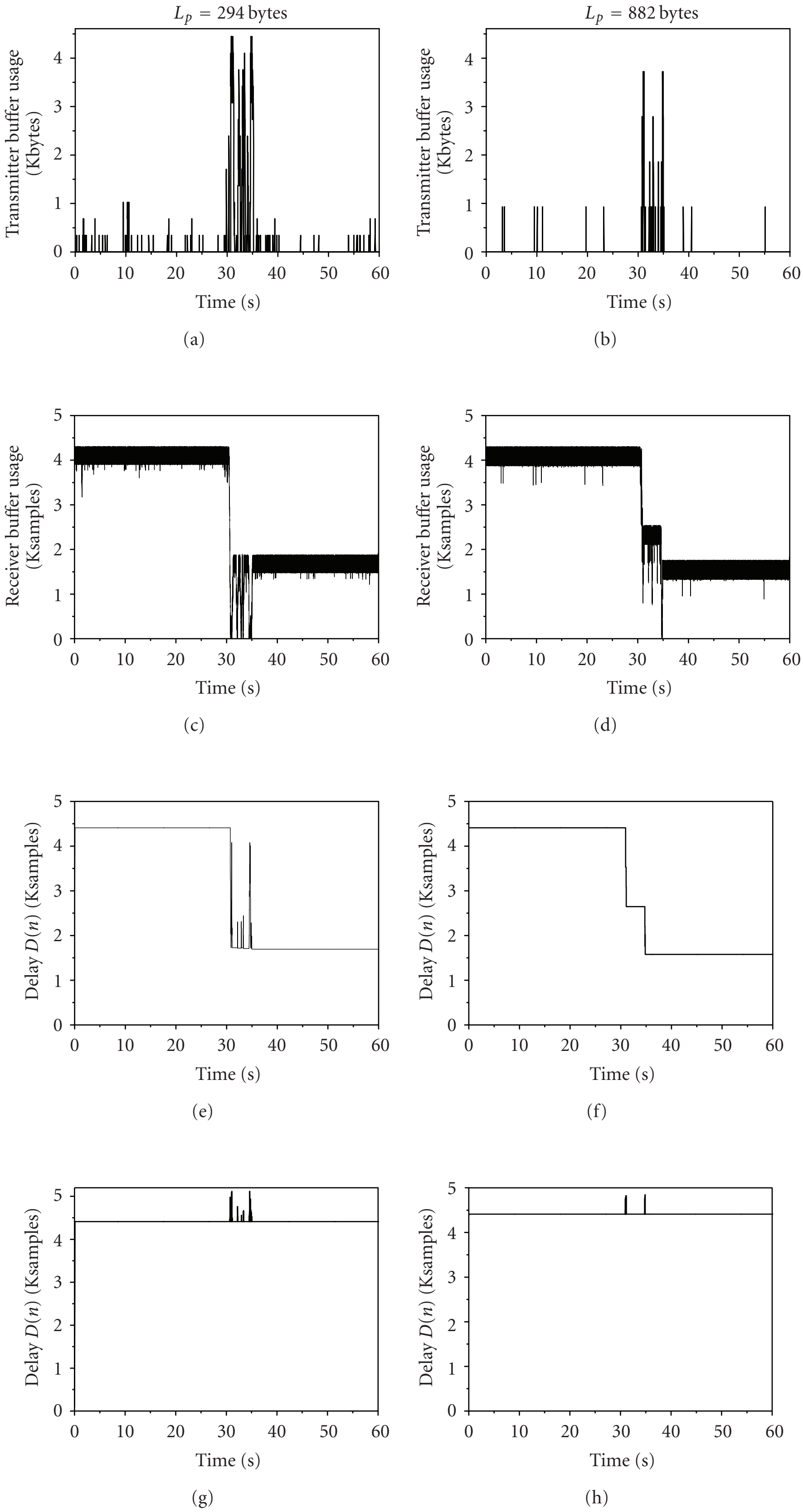
4.2. Network-induced distortion evaluation

4.2.1. End-to-end playback delay

In order to detect the end-to-end delay for each audio channel under all possible parameters for PCM transmission, a periodic audio test signal was selected as the system input. By comparing the original input signal to the wirelessly reproduced version, the end-to-end delay can be estimated as seen in the following figures.

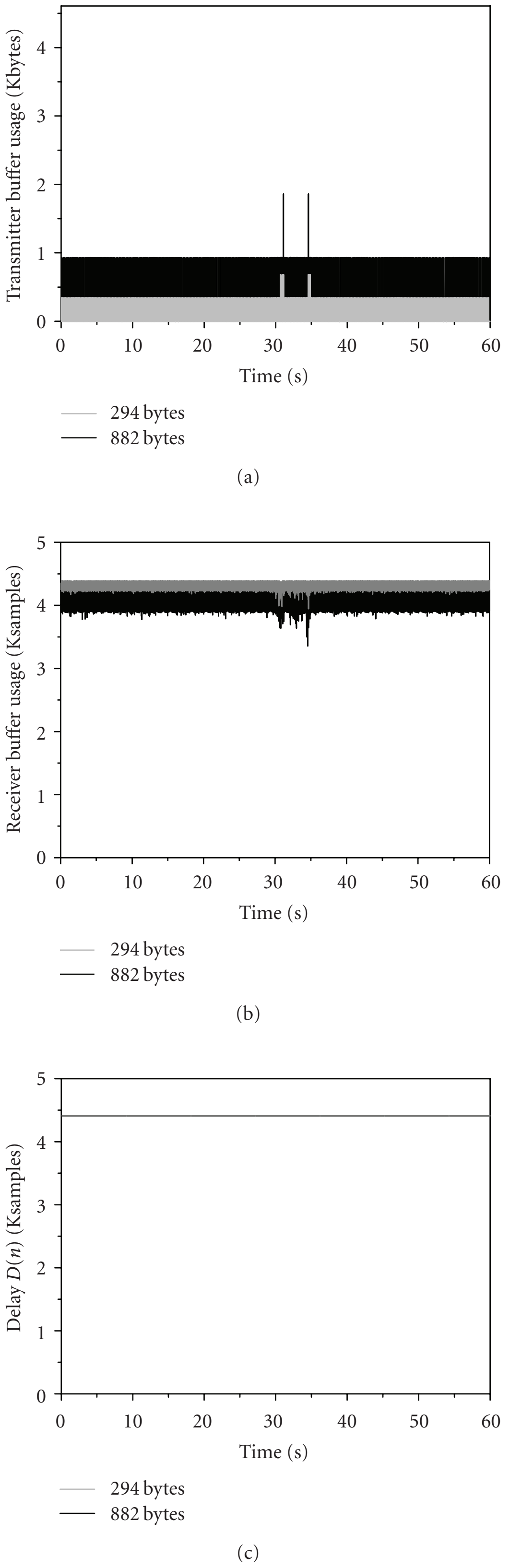
Figure 6 shows typical Tx and the Rx usage for a single digital audio traffic stream, as well as the playout delay for the corresponding receiver over a 60-second simulation interval, for packet sizes $L_p = 294$ and 882 bytes and for medium interference channel conditions. For these results, the SiS scheduler is employed, while the effect of the CoDeS synchronization algorithm to the playout delay is shown in the lower diagrams. Figure 7 shows the corresponding Tx and Rx buffer usage and playout delay for $L_p = 294$ and 882 bytes, when the SETT-EDD scheduler is employed, under similar medium interference channel conditions.

Provided that the total Tx and Rx buffer lengths for each audio traffic flow were selected equal to 5 Kbytes and 5 K



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6: Typical example for time evolution in Tx and Rx buffer and end-to-end delay. One channel of a stereo setup is shown for different packet sizes (for SiS scheduler and medium interference): (a), (b) Tx buffer usage; (c), (d) Rx buffer usage; (e), (f) delay $D(n)$ without CoDeS synchronization; (g), (h) delay $D(n)$ with CoDeS synchronization.



(a)

(b)

(c)

Figure 7: Typical example for time evolution in Tx and Rx buffer and end-to-end delay. One channel of a stereo setup is shown for different packet sizes (SETT-EDD scheduler and medium interference): (a) Tx buffer usage, (b) Rx buffer usage, (c) delay $D(n)$ without and with CoDeS synchronization (identical lines).

samples, respectively; the maximum data that can be inserted into the corresponding queue is equal to the nearest, lowest multiple integer of the packet size employed. It should be also noted that when the playback starts, the initial Rx queue filling size equals to 4410 samples, due to the 100 ms prebuffering applied.

From the above tests, the following conclusions can be drawn.

(a) The SiS scheduler introduces overflows in the Tx buffer and significant data losses. Accordingly, the Rx buffer remains empty of data at many instances during the same interval, causing gaps in reproduction.

(b) For the case of the SETT-EDD, the minimum Rx buffer usage value for all tests is nearly 2500 samples for the largest packet considered. Thus, the prebuffering time can be reduced, without generating gaps.

(c) The Tx buffer is optimally utilized when using the SETT-EDD scheduler. For medium wireless channel conditions, higher buffer usage will be required; however, no overflows have occurred during the tested interval. It can be deduced that even for a smaller Tx buffer, the system would operate without data losses.

(d) Although the SiS scheduler performance is poor for all test cases, less erratic—but still not acceptable—playback is performed for $L_p = 882$ bytes. On the other hand, no playback distortions occur when using the SETT-EDD scheduler, while improved operation is achieved for smaller packet sizes.

(e) The proposed application-level CoDeS synchronization algorithm generally ensures that the reproduction is kept synchronized for all test cases, since the delay for each traffic stream is constant throughout the simulation time.

The effect of the data overflows in the TxQ and the wireless variable packet delay transmission are clearly shown in Figure 8, where the original transmitted and the wirelessly reproduced waveforms are shown for a single audio channel. Apart of the silence gaps, a significant shift of the original waveform to the right side of the plot diagram is observed, which introduces relative channel phase delay. The audibility of both types of distortions introduced (silence gaps and relative channel delay) was verified through a sequence of tests, analyzed in the following section.

4.2.2. Audibility of distortions

Over the past years a number of psychoacoustic models and methods had been proposed to measure the perceived quality of both speech and audio signals [28]. The emergence of these approaches formulated up, to a certain degree, the ITU-R recommendation on perceptual evaluation of audio quality (PEAQ) [28, 29]. In the present study, two different methods were employed to assess the audibility of the wireless network distortions and to evaluate the performance of the interstream synchronization algorithm, on both PCM and compressed data: (a) the well-accepted noise-to-mask (NMR) criterion [30] and (b) a number of subjective listening tests.

Although the NMR criterion was initially developed for the purposes of perceptual audio coding, it can be also

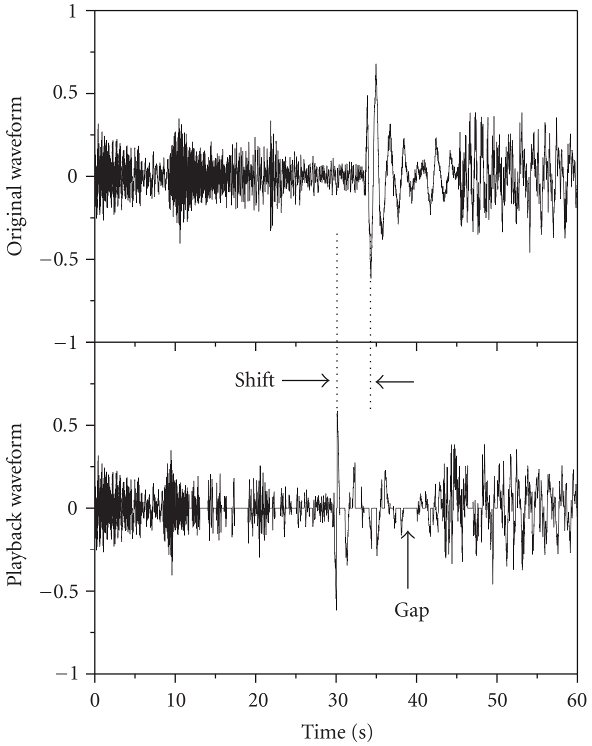


Figure 8: (a) Original digital audio source waveform, (b) reproduced (playback) waveform.

utilized to any audio processing system [31]. For NMR estimation (dB) in frame *i*, the ratio between the "error" energy and masked threshold, grouped in a number of critical bands, was calculated in a frame by frame basis, that is,

$$\mathrm{NMR}(i) = 10\cdot\log_{10}\left(\frac{1}{27}\sum_{\mathrm{cb}=1}^{27}\frac{\mathrm{error}_{\mathrm{cb}}(i)}{\mathrm{mask\_thr}_{\mathrm{cb}}(i)}\right). \quad (17)$$

The objective metric utilized for the quality assessment of audio streams was based on the averaged NMR(i) values, for a total number of K frames. It should be noted that NMR values above 0 dB indicate the presence of audible distortions, while NMR values below 10 dB indicate an audio signal free of audible distortions [30] and that the reference signal used in all test cases was the original PCM audio track (prior to wireless transmission and any encoding/decoding).

The subjective listening tests considered both raw PCM and MPEG-1 Layer III audio streams, both wireless channel conditions (no and medium interference), as well as the proposed CoDeS synchronization algorithm. The tests were organized as follows: a total of 9 listeners participated in two successive sessions where identical test files were reproduced randomly in two *phases*. In *phase A,* the uncompressed audio signals were presented to the subjects and in *phase B,* the encoded/decoded audio test signals were presented. For both cases, the listeners were informed about the type of signals being reproduced, thus, to consider inherent degradation of the encoded/decoded audio signals. The listeners ranked the quality of the audio material in the scale from 1 to 5 where 1 was described as "bad," 2 as "poor," 3 as "fair," 4 as "good," and 5 as "excellent."

Figures 9 and 10 show the average NMR values, while Figures 11 and 12 show the corresponding subjective listening results as a function of the test parameters, for the case of raw PCM and mp3-coded audio.

From these figures, the following conclusions can be drawn.

(a) For raw PCM audio streams, the effect of the wireless network conditions in most cases is inaudible. However, using the SiS scheduler under medium interference wireless channel conditions (for both choices of packet lengths) introduces a notable audible degradation.

(b) This degradation is reduced when the proposed CoDeS synchronization algorithm is employed, which compensates for such distortions and reduces the average NMR values by 23 dB (for 294 bytes packet length) and 34 dB (for 882 bytes packet length). This can be also observed at Figure 11, where the CoDeS synchronization algorithm was ranked with higher score values.

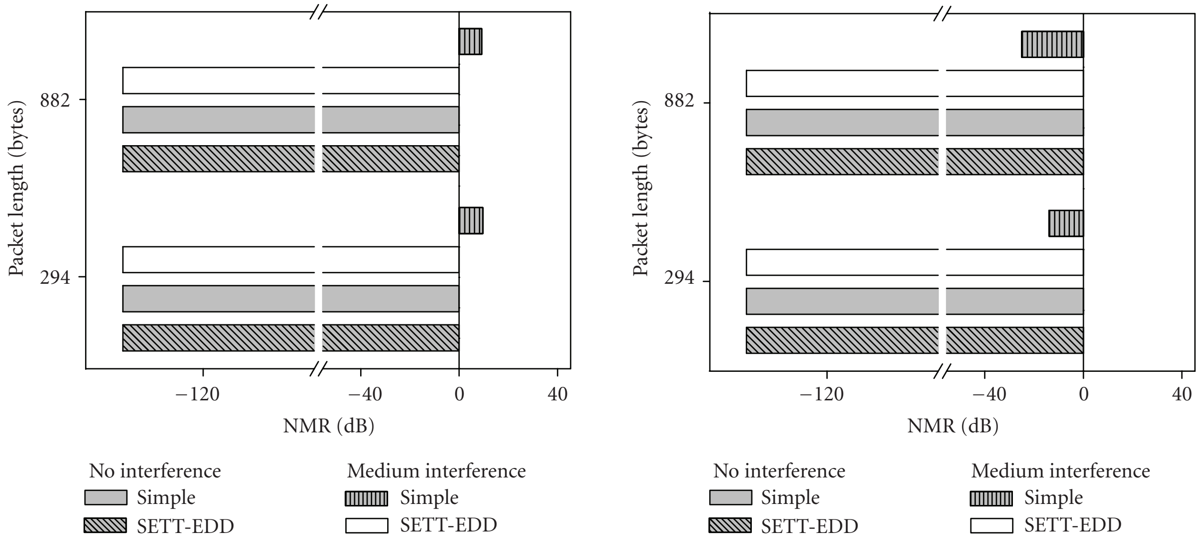
(c) For compressed audio streams, it is clear that the inherent distortions due to the lossy data compression bias the NMR measurements, as it is also depicted in Figure 12 where generally lower score values are observed. However, it is clear that the wireless network imposes additional significant degradation to the overall audio quality, for almost all cases of using the SiS scheduler, especially during medium interference wireless channel conditions.

(d) For a coding rate of 256 kbps and even for good channel conditions, the SiS scheduler introduces significant quality degradations, which again are reduced by the CoDeS synchronization algorithm (see Figures 10(b) and 12(b)). In these cases, the channel bandwidth usage is suboptimal due to the packet size employed (equal to the mp3 frame length).

(e) In Figure 10(b), NMR values for three of the test cases (i.e., 160 kbps bitrate) are close to 10 dB, which ideally indicate an audio signal free of audible distortions. However, Figure 12(b) indicates audio quality below "fair" for the test cases considered. As it is clear, full compliance between the subjective and objective tests is difficult to succeed, that is, instant audible distortions may cause the listener(s) to rank the entire audio segment as "fair" or even "poor." According to this approach, it is possible to have a biased subjective ranking (towards to low-grade audio quality) even for the highest bitrate (i.e., 256 kbps).

(f) Using the CoDeS synchronization algorithm in compressed audio (mp3) streams, an overall perceptual improvement equal to 7 dB can be achieved. More specifically, the distortions for the SiS scheduler, under medium interference wireless channel conditions, are compensated for both bitrates considered here (i.e., 160 kbps and 256 kbps). This is also stated in Figure 12(b) as the score values for the test cases considered are above 2 indicating a slightly better perceived audio quality.

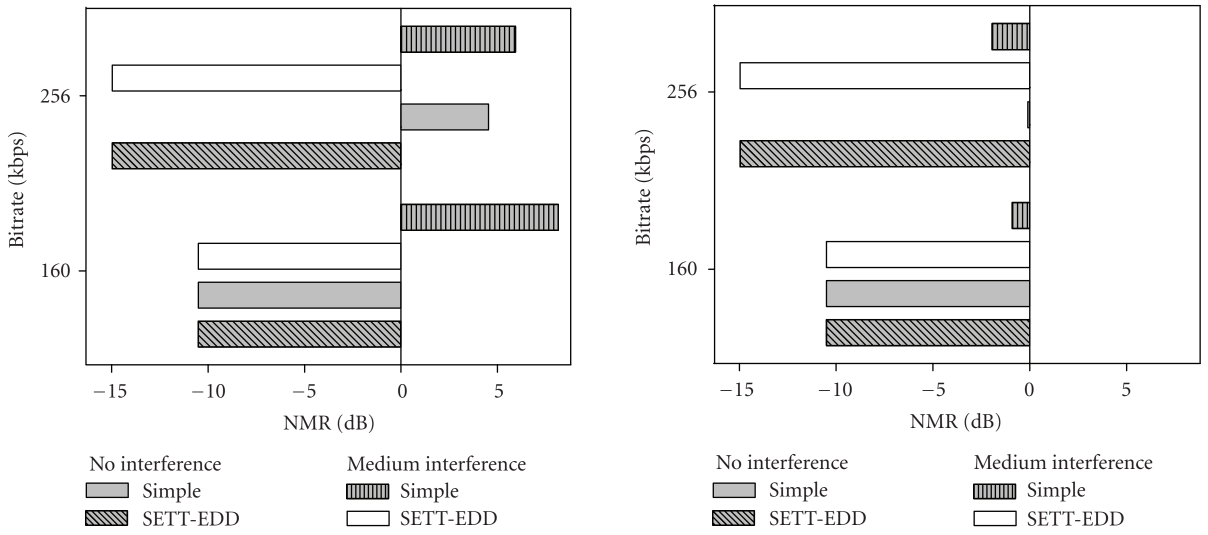
(g) As expected, the adaptive nature of the SETT-EDD scheduler leads to an overall better performance compared to the SiS scheduler, for all channel configurations and audio material.



(a)

(b)

Figure 9: NMR values for 1-channel PCM audio WLAN streaming delivery: (a) CoDeS disabled; (b) CoDeS enabled.



(a)

(b)

Figure 10: NMR values for MPEG-1 Layer III audio streaming delivery: (a) CoDeS disabled; (b) CoDeS enabled.

(h) Clearly, the results of the listening tests are in close agreement with the audio quality assessment performed using the NMR criterion.

Summarizing the above results, it is obvious that the wireless channel has a significant impact on the overall perceived playback audio performance. Nevertheless, from a networking point of view, the choice of the scheduler represents a critical decision, as it can render the presence of any wireless channel interference, transparent to the application. Moreover, it is clear that the proposed CoDeS synchronization strategy significantly improves the playback quality of both compressed and uncompressed audio for the case of low wireless link quality.

4.2.3. Overall audio WLAN performance

Error-free playback can be achieved when the output audio streams match sample-accurately the input streams; after taking out the initial delay caused by the prebuffering stage, Figure 13 shows the error-free stereo PCM and mp3 playback test cases examined under good and medium wireless channel conditions. The above methodology was extended to a wireless 5-channel PCM (16 bit/44.1 KHz) playback system and the number of error-free reproduced audio channels for this case is shown in Figure 14. It can be deduced that when interference is present, smaller packet lengths and the employment of an adaptive scheduler (such as the SETT-EDD) should be preferred.

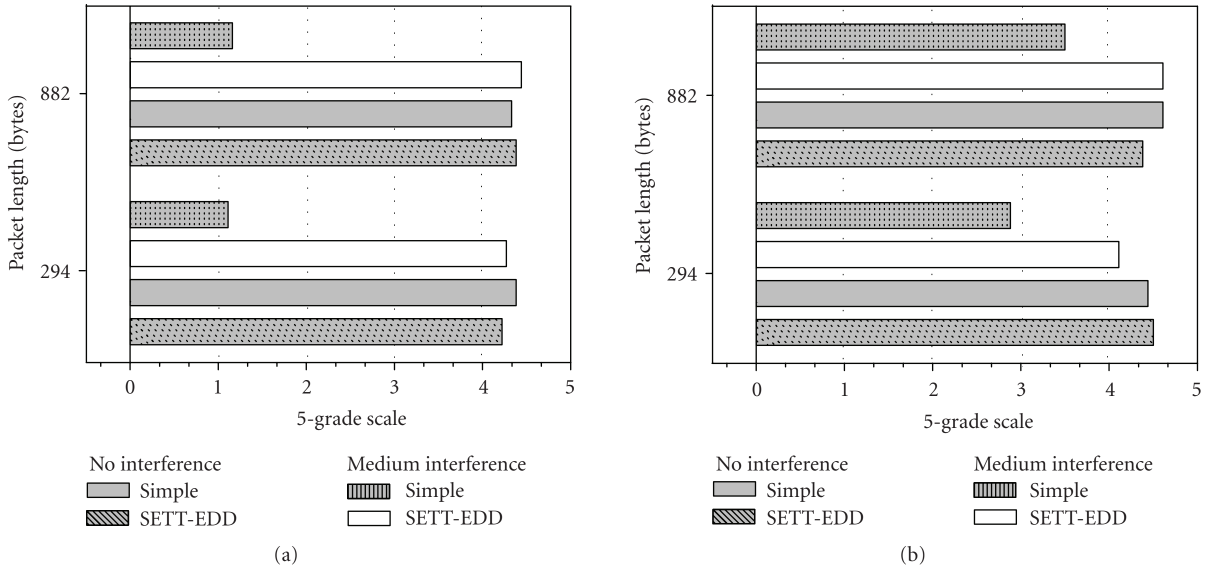


FIGURE 11: Subjective test results for PCM audio WLAN streaming delivery: (a) CoDeS disabled; (b) CoDeS enabled.

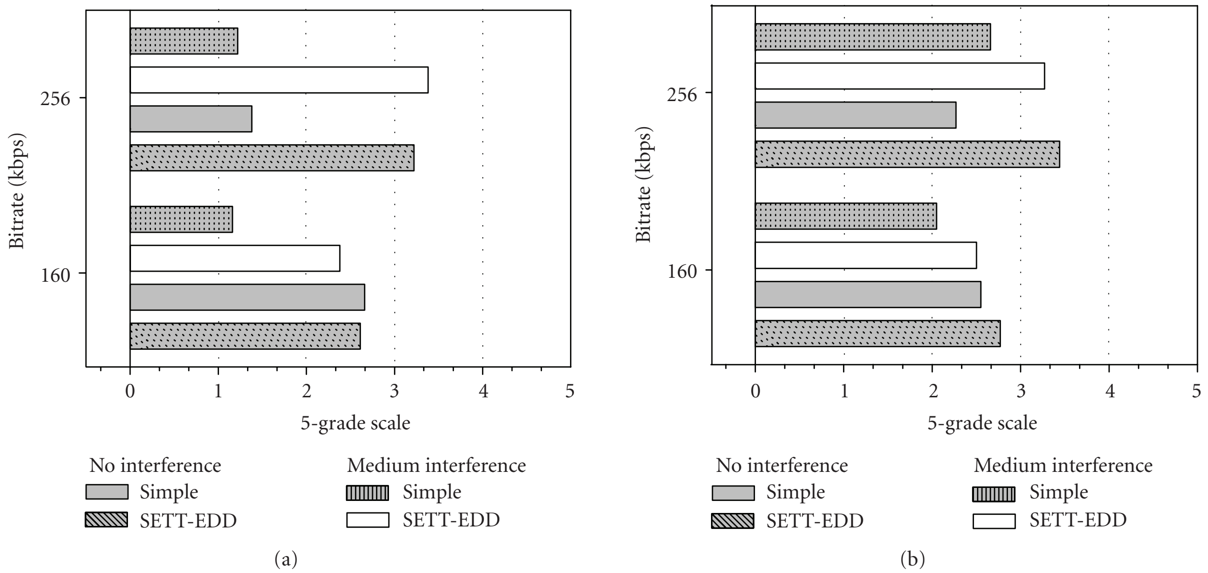


FIGURE 12: Subjective test results for MPEG-1 Layer III audio streaming delivery: (a) CoDeS disabled; (b) CoDeS enabled.

Finally, Figure 15 shows the measured delayed throughput (T_D) values (see (13)) for the wireless transmission of stereo mp3, stereo PCM 16 bit/44.1 KHz, 5-channel PCM 16 bit/44.1 KHz, and stereo PCM 24 bit/96 KHz audio, using the simple and the SETT-EDD scheduler, under medium interference wireless channel conditions. As can be deduced from (13), T_D (Mbps) indicates the mismatch between the requested (from the audio source) and the on-time WLAN delivered data. Note that for mp3 and stereo CD-quality transmission, the delayed throughput for SETT-EDD is zero. It appears that for noncompressed PCM audio and for increasing source bitrate, a corresponding increase of T_D can be expected. However, for mp3 compressed audio transmission, due to the additional overhead caused by the mismatch between the mp3 frame length and the network packet length, a strong increase of the measured T_D values must be expected.

Clearly, while the packet length does not seem to have a direct impact on the playback performance for PCM audio transmission, smaller packet sizes are more robust, since the Tx and Rx buffers are optimally used. The test cases prove that the selection of the service scheduler, especially

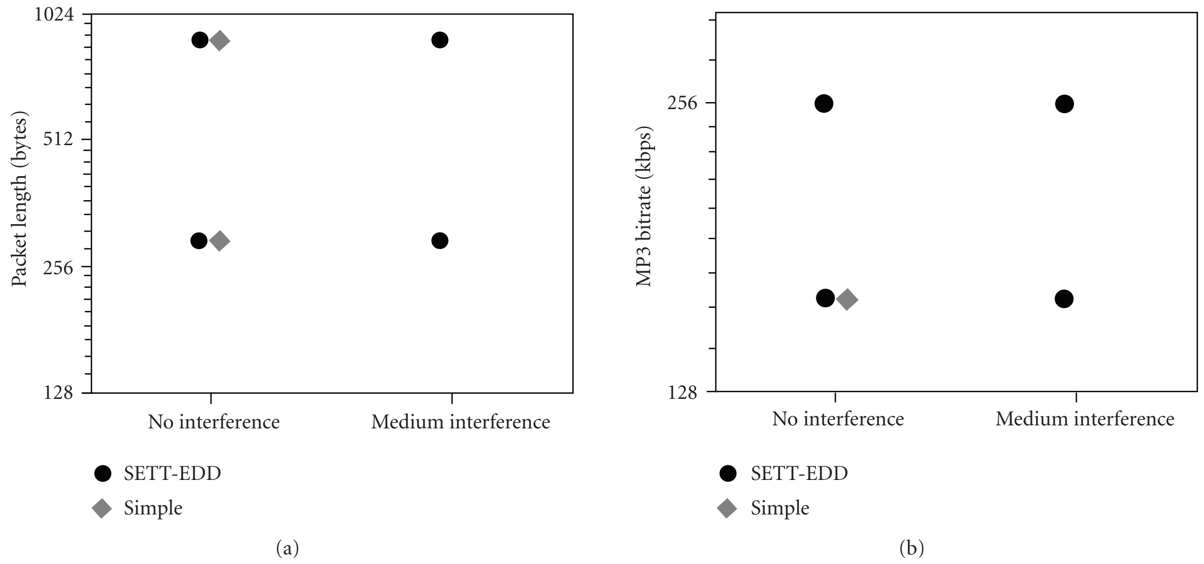


Figure 13: Parameter map for error-free stereo digital audio WLAN delivery: (a) PCM $f_s = 44.1$ KHz, $N = 16$ bit; (b) mp3 coded at 160 Kbps and 256 Kbps.

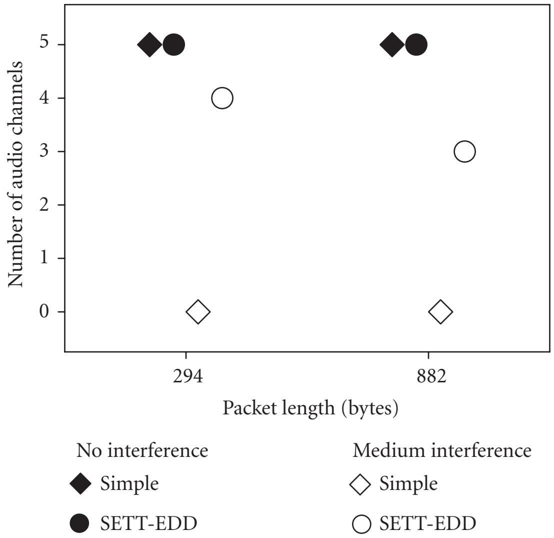


Figure 14: Number of serviced audio channels versus packet length for error-free digital WLAN delivery.

under the presence of channel interference, as in probable real-life conditions plays a crucial role on the overall playback performance. More specifically, it was found that the simple scheduler defined by the 802.11e specification as a minimum requirement reference design introduces significant reproduction distortions for most test cases, even for mp3 transmission under medium channel interference. Note that the simple scheduler induces distortions in all receivers of a 5-channel PCM system and completely fails to service high quality 96 KHz/24 bit PCM stereo.

On the other hand, largely error-free reproduction is achieved for the case of the SETT-EDD scheduler, although some distortions can be expected for the cases of 5-channel PCM and 96 KHz/24 bit stereo reproduction, largely due to the higher bitrate required. The trends obtained from the error-free results for the SETT-EDD scheduler would apply even if the Tx and Rx buffer lengths are reduced and starter prebuffering time is applied. This can enhance the overall performance of the WLAN digital audio system, especially for applications where a maximum delay limit is imposed, such as for typical audiovisual applications.

5. CONCLUSIONS

The most challenging aspect of integrating digital audio and WLAN technologies appears to be the robust and synchronized real-time streaming of multiple channels to multipoint receivers. Although efficient, cost-effective, and well-established network topologies exist for such applications, it appears that these solutions are not yet transparent for audio applications. An initial conclusion derived from this study is that successful operation can be achieved only if the wireless protocol provides strict QoS guarantees.

Main sources of distortion are due to audio WLAN packets, being permanently lost, generating gaps and loss of synchronization between the reproduced audio channels. The results presented in the previous section describe the optimal parameters for error-free stereo CD-quality and mp3 real-time audio playback. While it may be assumed that the 802.11b 11 Mbps throughput should be sufficient for uncorrupted mp3 and PCM stereo playback, such comparisons between the audio source bitrate and the physical channel throughput do not suffice for concluding that the application

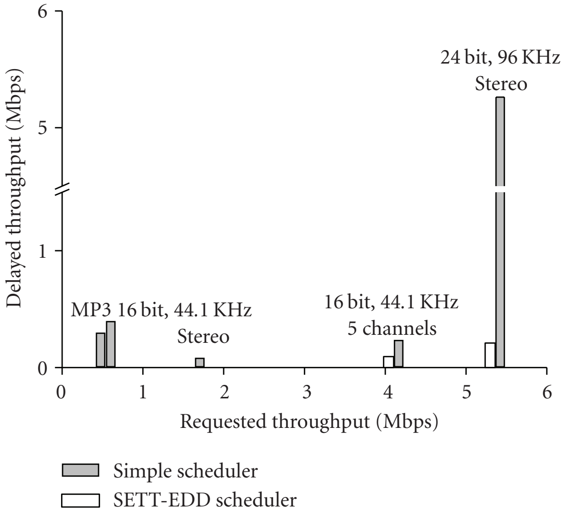


Figure 15: Delayed throughput for digital WLAN audio delivery.

will not suffer from such WLAN distortions. The largely error-free reproduction achieved for the case of the SETT-EDD scheduler shows that the employment of an adaptive scheduler, which dynamically adjusts the service schedule based on the networking conditions, represents a fundamental requirement for practical real-time audio streaming applications, at the expense of increased implementation complexity and processing power. However, excessive channel interference may always introduce additional transmission errors and channel congestion. Under such conditions and due to the wireless network bandwidth limitations, any dynamic scheduling algorithm (such as the SETT-EDD scheduler) may also fail to service real-time streaming traffic flows.

An application-level mechanism such as the proposed CoDeS synchronization algorithm will generally ensure that the reproduction is kept synchronized for WLAN transmission-induced errors by keeping the delay for each traffic stream constant, for each serviced device. Additionally, although it is out of the scope of this work, it is expected that the CoDeS synchronization algorithm combined with a packet concealment method [32, 33] will minimize the perceptual effect of the distortions that are introduced by the insertion of silence gaps and proportionally increase the achieved playback quality. Furthermore, by increasing the available bandwidth, the effect of the channel interference on the final playback quality can be reduced. Hence, high-rate wireless protocols (e.g., 801.11g/n) must be preferred for high-quality multimedia and audio applications. Additionally, higher-layer protocols can be developed for dynamically adjusting the number of the playback devices which can be serviced within such a WLAN. For example, under wireless channel degradation, such protocols may temporarily stop servicing playback devices with lower priority on the perceived audio system quality (e.g., the rear speakers in a multichannel DVD setup).

ABBREVIATIONS

AP:	Access point
CoDeS:	Consistent delay synchronization
EDCA:	Enhanced distributed channel access
FIFO:	First-in first-out
HCCA:	Hybrid controlled channel access
ISM:	Industrial, scientific, medicine
LAN:	Local area network
MAC:	Medium access control
PHY:	Physical rate
QAP:	Quality of service access point
QoS:	Quality of service
Rx:	Reception
SETT-EDD:	Scheduling based on estimated transmission times-earliest due date
SiS:	Simple scheduler
STA:	Wireless station
TSPEC:	Traffic specification
Tx:	Transmission
TXOP:	Transmission opportunity
WiDAR:	Wireless digital audio receiver
WiDAS:	Wireless digital audio source
WLAN:	Wireless local area network.

REFERENCES

[1] T. Blank, B. Atkinson, M. Isard, J. D. Johnston, and K. Olynyk, "An internet protocol (IP) sound system," in *Proceedings of the 117th Convention of the Audio Engineering Society*, San Francisco, Calif, USA, October 2004, (preprint 6211).

[2] Bluetooth SIG, "Specification of the bluetooth system," bluetooth core specification version 2.0 EDR [vol 0], November 2004.

[3] European Telecommunications Standards Institute (ETSI), "Broadband radio access networks (BRAN)," HIPERLAN Type 2 Specification.

[4] "The HomeRF^{TM} Technical Committee," HomeRF Specification, Revision 2.01, July 2002.

[5] IEEE802.11 WG and IEEE802.11, "Information technology telecommunications and information exchange between system local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: higher-speed physical layer extension in the 2.4 GHz band," September 1999.

[6] IEEE802.11 WG and IEEE802.11g, "IEEE standard for information technology-telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 4: further higher data rate extension in the 2.4 GHz band," June 2003.

[7] N.-A. Tatlas, A. Floros, and J. Mourjopoulos, "Wireless digital audio delivery analysis and evaluation," in *Proceedings of the 31th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. V201–V204, Toulouse, France, May 2006.

[8] H. Liu and M. Zarki, "A synchronization control scheme for real-time streaming multimedia applications," in *Proceedings of the 13th IEEE International Packet Video Workshop*, Nantes, France, April 2003.

[9] P. Blum and L. Thiele, “Trace-based evaluation of clock synchronization algorithms for wireless loudspeakers,” in *Proceedings of the 2nd Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia ’04)*, pp. 7–12, Stockholm, Sweden, September 2004.

[10] A. Xu, W. Woszczyk, Z. Settel, et al., “Real-time streaming of multichannel audio data over Internet,” *Journal of the Audio Engineering Society*, vol. 48, no. 7, pp. 627–639, 2000.

[11] IEEE802.11 WG and IEEE802.11e/D13.0, “IEEE standard for information technology-telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless medium access control (MAC) and physical layer (PHY) specifications: amendment: medium access control (MAC) quality of service enhancements,” January 2005.

[12] A. Floros, N.-A. Tatlas, and J. Mourjopoulos, “BlueBox: a cable-free digital jukebox for compressed-quality audio delivery,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 2, pp. 534–539, 2005.

[13] http://www.streamium.com/.

[14] http://www.elevenengineering.com/.

[15] http://www.amphony.com/products/h2500.htm.

[16] http://www.pioneerelectronics.com/.

[17] N.-A. Tatlas, A. Floros, P. Hatziantoniou, and J. Mourjopoulos, “Towards the all-digital audio/acoustic chain: challenges and solutions,” in *Proceedings of the AES 23rd International Conference on Signal Processing in Audio Recording and Reproduction*, Copenhagen, Denmark, May 2003.

[18] J. S. Flaks, “Quality of service (QoS) for streaming audio over wireless LANs,” in *Proceedings of the AES 18th International Conference: Audio for Information Appliances*, Burlingame, Calif, USA, March 2001.

[19] A. Floros and T. Karoubalis, “Delivering high-quality audio over WLANs,” in *Proceedings of the 116th AES Convention of the Audio Engineering Society*, Berlin, Germany, May 2004, (preprint 5996).

[20] A. Floros, T. Karoubalis, and S. Koutroubinas, “Bringing quality in the 802.11 wireless arena,” in *Broadband Wireless and WiMax IEC Comprehensive Report*, International Engineering Consortium, Chicago, Ill, USA, 2005.

[21] S. Mangold, S. Choi, G. R. Hiertz, O. Klein, and B. Walke, “Analysis of IEEE 802.11 e for QoS support in wireless LANs,” *IEEE Wireless Communications*, vol. 10, no. 6, pp. 40–50, 2003.

[22] A. Grilo, M. Macedo, and M. Nunes, “A scheduling algorithm for QoS support in IEEE802.11E networks,” *IEEE Wireless Communications*, vol. 10, no. 3, pp. 36–43, 2003.

[23] X. Gu, M. Dick, Z. Kurtisi, U. Noyer, and L. Wolf, “Network-centric music performance: practice and experiments,” *IEEE Communications Magazine*, vol. 43, no. 6, pp. 86–93, 2005.

[24] N.-A. Tatlas, A. Floros, T. Zarouchas, and J. Mourjopoulos, “An error—concealment technique for wireless digital audio delivery,” in *Proceedings of the 5th International Conference on Communication Systems, Networks and Digital Signal Processing (CSNDSP ’06)*, pp. 181–184, Patras, Greece, July 2006.

[25] H. Ofir and D. Malah, “Packet loss concealment for audio streaming based on the GAPES algorithm,” in *Proceedings of the 118th Convention of the Audio Engineering Society*, Barcelona, Spain, May 2005, (preprint 6334).

[26] N.-A. Tatlas, A. Floros, and J. Mourjopoulos, “An evaluation tool for wireless digital audio applications,” in *Proceedings of the 118th Convention of the Audio Engineering Society*, Barcelona, Spain, May 2005, (preprint 6386).

[27] A. K. Salkintzis, G. Dimitriadis, D. Skyrianoglou, N. Passas, and N. Pavlidou, “Seamless continuity of real-time video across UMTS and WLAN networks: challenges and performance evaluation,” *IEEE Wireless Communications*, vol. 12, no. 3, pp. 8–18, 2005.

[28] C. Colomes, C. Schmidmer, T. Thiede, and W. C. Treurniet, “Perceptual quality assessment for digital audio: PEAQ—the New ITU standard for objective measurement of the perceived audio quality,” in *Proceedings of the 17th International Conference on the Audio Engineering Society (AES ’99)*, Florence, Italy, September 1999.

[29] S. Bech and N. Zacharov, *Perceptual Audio Evaluation—Theory, Method and Application*, John Wiley & Sons, New York, NY, USA, 2006.

[30] K. Brandenburg and T. Sporer, “NMR and masking flag: evaluation of quality using perceptual criteria,” in *Proceedings of the 11th International AES Conference: Audio Test & Measurement*, pp. 169–179, Portland, Oregon, May 1992.

[31] J. Herre, E. Eberlein, H. Schott, and K. Brandeburg, “Advanced audio measurement system using psychoacoustic properties,” in *Proceedings of the 92nd AES Convention of the Audio Engineering Society*, New York, NY, USA, March 1992, (preprint 3321).

[32] B. W. Wah, S. Xiao, and L. A. Dong, “A survey of error-concealment schemes for real-time audio and video transmissions over the Internet,” in *Proceedings of International Symposium on Multimedia Software Engineering*, pp. 17–24, Taipei, Taiwan, December 2000.

[33] A. Floros, M. Avlonitis, and P. Vlamos, “Stochastic packet reconstruction for subjectively improved audio delivery over WLANs,” in *Proceedings of the 3rd International Mobile Multimedia Communications Conference (MOBIMEDIA ’07)*, Nafpaktos, Greece, August 2007.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 95474, 7 pages
doi:10.1155/2007/95474

Research Article

Efficient TTI for 3G Multimedia Applications

Costas Chaikalis

Areos 28, 13121 Athens, Greece

Received 24 April 2007; Accepted 23 July 2007

Recommended by Stavros Kotsopoulos

Time transmission interval (TTI) or outer block interleaving is an important task for the implementation of UMTS turbo coding in flat Rayleigh fading environment. An efficient TTI choice can save computational complexity. However, different multimedia scenarios are investigated using the maximum UMTS frame length, and simulation results are presented for the four possible outer block interleaver configurations in the case of flat Rayleigh fading channel. It is shown that different operating environments require an appropriate TTI in terms of bit error rate (BER) performance for the following data rates: 28.8 kbps, 64 kbps, 144 kbps, 384 kbps, and 2 Mbps.

1. INTRODUCTION

Turbo coding [1] offers energy efficiencies close to the limits predicted by information theory with features that include parallel code concatenation, recursive convolutional encoding, interleaving, and an iterative decoding algorithm; while in fading environments, outer block interleaving should also be used. Soft-input/soft-output (SISO) decoder is the significant part of a turbo decoder: the concept of iterative decoding relies on the use of SISO decoders which calculate the a posteriori probabilities based on the received channel sequences and a priori information. One of the main candidate algorithms to be used in a SISO decoder is log maximum a posteriori (log-MAP) algorithm. A detailed description of log-MAP is presented in [2–7], while turbo code basics and performance can be found in [7].

The European 3G standard is called Universal Mobile Telecommunications System (UMTS) and provides data rates up to 2 Mbps giving the opportunity to mobile operators to offer multimedia applications to their customers, according to Table 1. Turbo codes have been adopted as a channel coding scheme in UMTS for data rates higher than or equal to 28.8 kbps [8, 9].

The rest of the paper is organised as follows: Section 2 presents a short literature review on the area together with a brief description of UMTS data stream. Subsequently, Section 3 gives a description of the simulation environment used. The optimum outer block interleaver length is selected through simulation results and different implementation scenarios in Section 4. Finally, after an efficient TTI selection according to simulation results, we conclude in Section 5.

2. RELATED WORK AND UMTS DATA STREAM

The investigation of multimedia applications over mobile cellular networks has been well addressed in published literature [10, 11], while different propagation (mobile channel) issues for such applications, tested by experimental procedures, have been presented in [12, 13].

In mobile communications, turbo codes provide very good coding gains in fading channels. Especially in frequency-flat Rayleigh fading channels, performance can be greatly improved if outer block interleaving is used [7, 14]. This is because turbo encoding without outer block interleaving cannot correct the burst errors induced in a correlated fading channel, since they are more effective with random errors.

In [15], it is shown that the number of columns is the critical parameter in the design of outer block interleavers for turbo codes over such channels. The higher the mobile speed, the larger number of columns needed. In [16], simulation results are presented for the four different UMTS TTIs using flat Rayleigh fading and convolutional coding. Only the coding rate is varied for a terminal speed of 50 km/h, a frame length of 504 bits and a bit rate of 64 kbps. For these parameters and different signal-to-noise ratios (SNRs), a TTI of 80 milliseconds is shown to achieve the best FER

Table 1: UMTS multimedia services.

Category	Multimedia applications
Entertainment	Internet, video, text, picture, and multimedia messaging, datacast, personalisation applications (ring tone, screen saver, desk top)
Work	Call with image and data stream, IP telephony, notepad, 2-way video conferencing, directory services, work group, telepresence, FTP, instant voicemail, colour fax
Shopping	E-commerce, e-cash, e-wallet, credit card, telebanking
Education	Online libraries, search engines, remote attendance, field research
Health	Telemedicine, remote diagnose, and health monitoring
Automation	Home automation, traffic telematics, machine, telemetry
Travel	Location sensitive information and guidance, e-tour, location awareness, time tables, e-ticketing
Additional services	TV, radio, PC, access to remote computer, MP3 player, camera, GPS, remote control unit

performance. Similar to the approach described in [16], but for turbo codes, in this paper, we discuss the effect of outer block interleaving on UMTS turbo codes performance for a constant frame length, log-MAP algorithm for turbo decoder, five different bit rates, and also different operating environments.

In [17], a reconfigurable block interleaver for block codes over flat Rayleigh fading is presented. A formula is derived to calculate the interleaving depth according to the Doppler frequency. It is also mentioned that the bit error rate (BER) will be higher unless the right number of columns is employed. In this paper, we will show that the same conclusion is also drawn for turbo codes. Finally, in [5], we discuss different UMTS implementation scenarios for an outer block interleaver, different frame lengths, operating environments, and a constant bit rate. This paper considers the same approach but constant frame length and different bit rates.

UMTS data stream is described in [8, 9]. Thus, a UMTS transport channel transfers the information over the radio interface from the medium access control sublayer of layer 2 to the physical layer. The characteristics of a transport channel are determined by its transport format set, which consists of different transport formats. The transport formats must have the same type of channel coding and TTI, while the transport block set or data frame size can vary. The transport block set determines the number of input bits to the channel encoder and can be transmitted every TTI, with possible values for TTI of 10, 20, 40, and 80 milliseconds. Every transport channel is also assigned to a radio access bearer with a particular data rate.

Table 2: Implementation scenarios, corresponding bit rates, and multimedia examples.

	Implementation scenario				
	1	2	3	4	5
Bit rate R_b	28.8 kbps	64 kbps	144 kbps	384 kbps	2 Mbps
Multimedia application	Fax service	Packet data services	Packet data services	Packet data services	Packet data services

3. SIMULATION ENVIRONMENT

A flat Rayleigh fading channel is described by

$$y_k = \alpha_k \cdot x_k + n_k, \quad (1)$$

where k is an integer symbol index, x_k is a binary phase shift keying (BPSK) symbol amplitude (±1), n_k is a Gaussian random variable, and y_k is a noisy received symbol. The fading amplitude a_k is a sample from a correlated Gaussian random process with zero mean and is generated using the sum of sines or Jakes model [18].

In our simulations, a carrier frequency $f_c = 2\,\text{GHZ}$ is considered. It is also assumed that 1 000 000 information bits are transmitted and grouped into frames whose length must be ≥40 and ≤5114, according to UMTS specifications [19, 20]. For a particular transport channel, every TTI, the data with the characteristics specified in a transport format of the transport channel, is turbo-encoded at the transmitter. After turbo encoding and block interleaving, the bits are BPSK-modulated and transmitted through the mobile channel.

At the receiver, outer block deinterleaving and turbo decoding is performed. Floating point arithmetic is also used, while the receiver is assumed to have exact estimates of the fading amplitudes (perfect channel estimation without side information). The iterations of the turbo decoder are assumed to be eight.

4. EFFICIENT TTI SELECTION

The frame length used in our simulations is chosen to be the maximum length specified for UMTS, 5114 bits. Five typical input bit rates R_b are considered according to UMTS specifications: 28.8 kbps, 64 kbps, 144 kbps, 384 kbps, and 2 Mbps [21]. These bit rates represent five different UMTS implementation scenarios, as Table 2 illustrates. Particularly, according to [21], scenario 1 can be applied to fax service, while scenarios 2, 3, 4, and 5 can be applied to any multimedia packet data service like conventional internet services (web-browsing, electronic mail, file transfer, video/audio streaming, e-commerce, videoconference), interactive data or combination with location information and mobility (location-based services, navigation).

Furthermore, four different mobile speeds are considered for each implementation scenario: 4 km/h (corresponds

TABLE 3: Normalised fade rates for different Doppler frequencies, mobile terminal speeds, and implementation scenarios.

Terminal speed (km/h)	Doppler freq. f_d (Hz)	Scenario				
		1	2	3	4	5
4	7.4	0.000085	0.000038	0.000017	0.0000064	0.0000012
50	92.5	0.00107	0.00048	0.00021	0.00008	—
100	185.1	0.0021	0.00096	0.00042	0.00016	—
300	555.5	0.0064	0.0028	0.00128	—	—

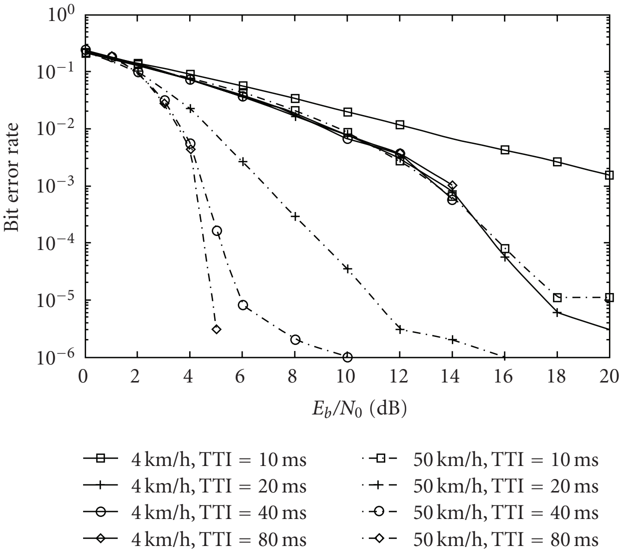


FIGURE 1: BER versus SNR for scenario 1 and f_dT_S values 0.000085 and 0.00107.

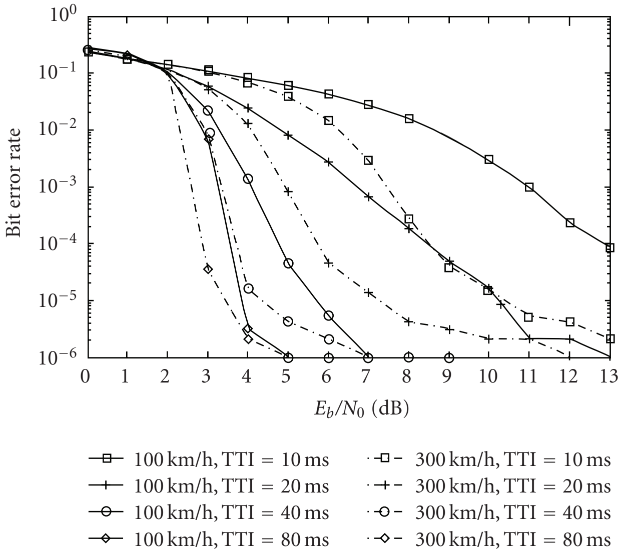


FIGURE 2: BER versus SNR for scenario 1 and f_dT_S values 0.0021 and 0.0064.

to indoor or low-range outdoor operating environment), 50 and 100 km/h (urban or suburban outdoor operating environment), and 300 km/h (rural outdoor operating environment). In Table 3, the different values of normalised fade rate f_dT_S are presented for each scenario and terminal speed. For the first 3 scenarios (data rates 28.8 kbps, 64 kbps, and 144 kbps), all 4 mobile terminal speeds can be applied according to [22], whereas for scenarios 4 and 5, there are some limitations. Thus, according to [22], for scenario 4 (data rate 384 kbps), the mobile terminal speed of 300 km/h cannot be considered, and for scenario 5 (data rate 2 Mbps), the mobile terminal speeds of 50, 100, and 300 km/h cannot be considered as well.

A scaling factor $s = 0.7$ is also applied in log-MAP turbo decoding algorithm in our simulation model because, according to [6], $s = 0.7$ and $s = 0.8$ give the best performance improvement for log MAP in a flat-fading channel. In the following sections, BER performance is evaluated for each scenario and our goal is to find the optimum TTI value.

4.1. Scenario 1: bit rate 28.8 kbps

In Figure 1, the BER performance of the simulated system for mobile speeds 4 and 50 km/h is presented. For 4 km/h and at a BER of 2×10^{-3}, there is a gain of approximately 6 dB for a TTI transition from 10 to 20 milliseconds. Thus, for 4 km/h, the optimum TTI value is 20 milliseconds: there is no BER improvement for larger TTI values. On the contrary, as Figure 1 shows, BER becomes worse for values larger than the optimum value of 20 milliseconds. Additionally, at a BER of 10^{-3}, 0.5 dB performance loss can be observed for 80 milliseconds compared to the optimum TTI value of 20 milliseconds. For 50 km/h and at a BER of 10^{-3}, there is a gain of approximately 6 dB for a TTI transition from 10 to 20 milliseconds. The gain decreases to 2 dB and 0.5 dB for a TTI transition from 20 milliseconds to 40 milliseconds and from 40 milliseconds to 80 milliseconds, respectively. Thus, for 50 km/h, the optimum TTI value is 80 milliseconds.

For Figure 2, the mobile terminal speeds are 100 and 300 km/h. Here, as Figure 2 shows, the optimum TTI value for both terminal speeds is 80 milliseconds. However, at a BER of 10^{-3} for 100 km/h, a gain of 4 dB, 2.5 dB and 1 dB is seen for a TTI transition from 10 to 20, 20 to 40, and 40 to 80 milliseconds, respectively. At the same BER for 300 km/h terminal speed, a TTI increase from 10 to 20, 20 to 40, and 40 to 80 milliseconds gives performance gains of 2.5 dB, 1.5 dB, and 0.8 dB, respectively.

4.2. Scenario 2: bit rate 64 kbps

Figure 3 presents the performance for the four different TTI values considering terminal speeds of 4 km/h and 50 km/h.

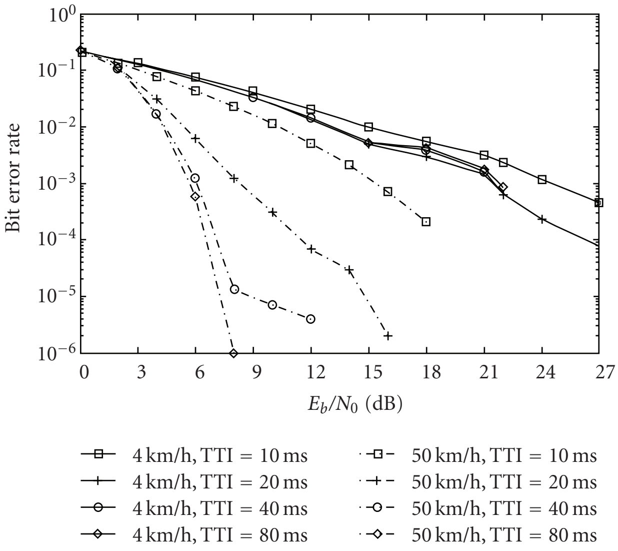


Figure 3: BER versus SNR for scenario 2 and $f_d T_S$ values 0.000038 and 0.00048.

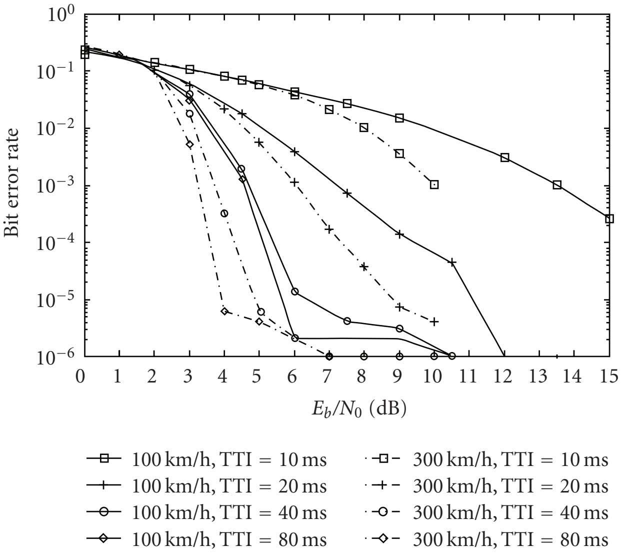


Figure 4: BER versus SNR for scenario 2 and $f_d T_S$ values 0.00096 and 0.0028.

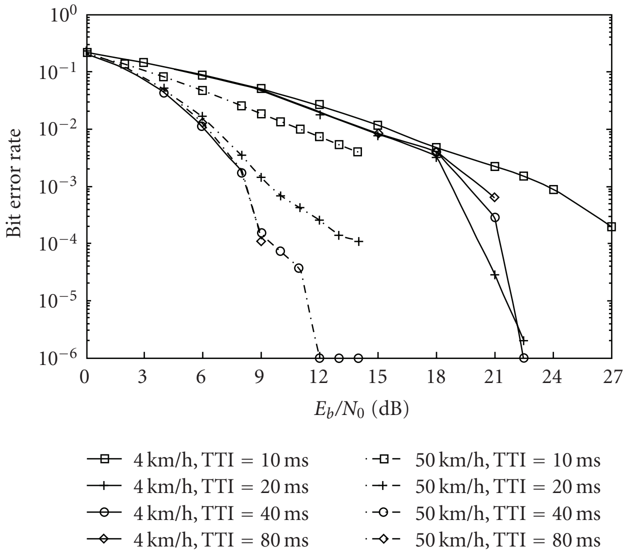


Figure 5: BER versus SNR for scenario 3 and $f_d T_S$ values 0.000017 and 0.00021.

At a BER of 10^{-3} for 4 km/h, we see that a TTI increase from 10 to 20 milliseconds gives 4 dB gain. It is obvious that a TTI = 20 milliseconds is the optimum choice: a further TTI increase gives no BER improvement but a gain loss of 0.6 dB at a BER of 10^{-3} and a loss of 2.5 dB at a BER of 3×10^{-3}. For 50 km/h, 80 milliseconds represents the ideal TTI choice in terms of performance and complexity. Particularly, at a BER of 10^{-3} performance gain of 6 dB, 2 dB, and 0.8 dB is observed for TTI increase from 10 to 20 milliseconds, 20 to 40 milliseconds, and 40 to 80 milliseconds.

Figure 4 illustrates the BER performance for speeds of 100 km/h and 300 km/h. For 100 km/h at a BER of 10^{-3}, a TTI of 20 milliseconds gives around 6 dB gain over 10 milliseconds, a TTI of 40 milliseconds gives 2.5 dB gain over 20 milliseconds, while a TTI of 80 milliseconds gives 0.25 dB gain over 40 milliseconds. For 300 km/h terminal speed at a BER of 10^{-3}, for a TTI increase from 10 to 20, 20 to 40, and 40 to 80 milliseconds, the corresponding improvements are 4 dB, 2 dB, and 0.6 dB. However, the optimum choice for both speeds is the maximum TTI value (80 milliseconds).

4.3. *Scenario 3: bit rate 144 kbps*

As can be observed from Figure 5, for 4 km/h mobile terminal speed, a TTI of 20 milliseconds is the optimum choice. Furthermore, at a BER of 10^{-3}, the performance improvement using 20 milliseconds is 5 dB compared to 10 milliseconds. On the other hand, it is obvious that for TTI = 40 milliseconds, there is a loss of 1 dB at a BER of 10^{-3}, while for TTI = 80 milliseconds there is a loss of 1.8 dB at the same BER compared to the optimum TTI value. For a terminal speed of 50 km/h, a TTI of 40 milliseconds represents the optimum solution. Consequently, the performance gain for a TTI increase from 10 to 20 milliseconds at a BER of 4×10^{-3} is 6 dB, while at a BER of 10^{-3} for a TTI increase from 20 to 40 milliseconds, the gain is 1.3 dB.

Figure 6 illustrates the performance of the simulated system for terminal speeds 100 km/h and 300 km/h. It is clear that for both speeds, the optimum value is 80 milliseconds. Among the three TTI values, a TTI of 20 milliseconds gives the highest improvement: at a BER of 10^{-2}, the gain using 20 milliseconds compared to 10 milliseconds is 5 dB, while using 40 milliseconds gives 2.2 dB gain compared to 20 milliseconds at a BER of 10^{-3}. Also at a BER of 10^{-3}, using 80 milliseconds gives 0.6 dB gain compared to 40 milliseconds. For a terminal speed of 300 km/h, again the highest performance improvement among the three possible TTI values (20, 40, 80 milliseconds) is 20 milliseconds:

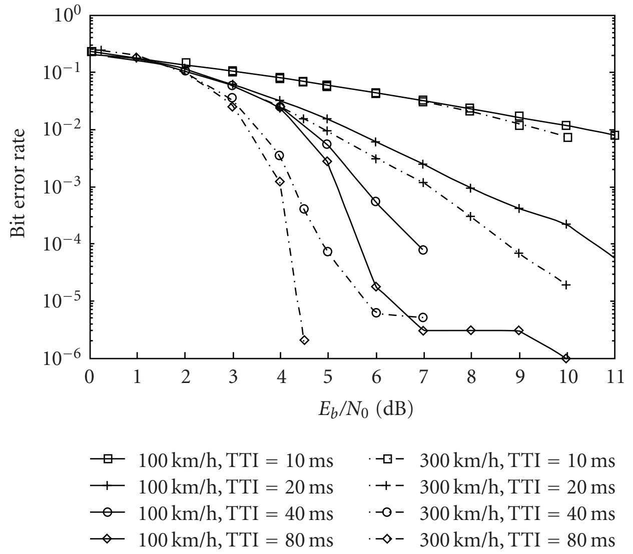


FIGURE 6: BER versus SNR for scenario 3 and f_dT_S values 0.00042 and 0.00128.

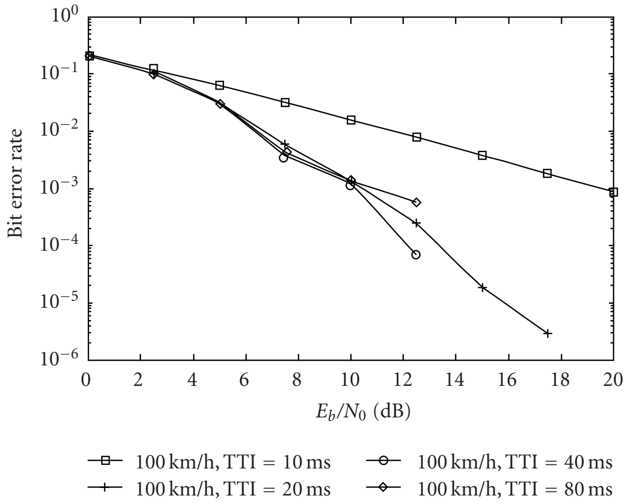


FIGURE 8: BER versus SNR for scenario 4 and $f_dT_S = 0.00016$.

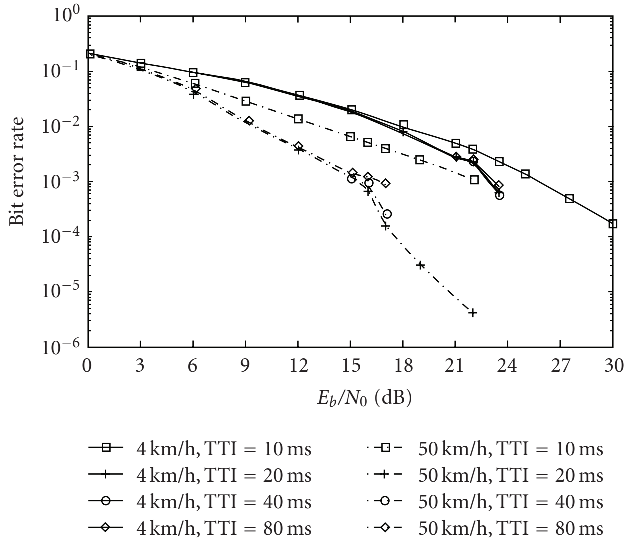


FIGURE 7: BER versus SNR for scenario 4 and f_dT_S values 0.0000064 and 0.00008.

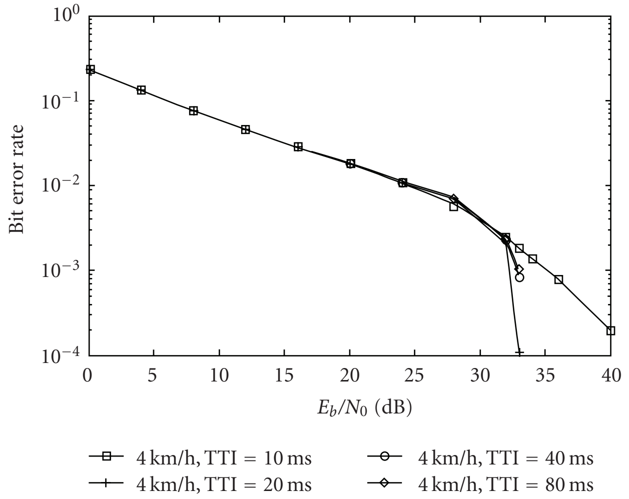


FIGURE 9: BER versus SNR for scenario 5 and $f_dT_S = 0.0000012$.

at a BER of 10^{-2}, the gain using TTI = 20 milliseconds compared to 10 milliseconds is 4.3 dB. At a BER of 10^{-3}, using 40 milliseconds instead of 20 milliseconds gives a gain of 2.5 dB, while using 80 milliseconds instead of 40 milliseconds gives a gain of 0.4 dB.

4.4. Scenario 4: bit rate 384 kbps

Considering a large bit rate of 384 kbps, Figure 7 presents the BER of the simulated system for mobile terminal speeds of 4 and 50 km/h. As can be seen, 20 milliseconds is the optimum interleaver length for both speeds. For the first terminal speed and at a BER of 10^{-3}, an increase of the TTI from 10 to 20 milliseconds gives a gain of 3 dB. An increase of the TTI to 40 milliseconds at the same BER results in no performance loss, whereas a further increase (80 milliseconds) can cause 0.5 dB loss. For the second terminal speed and at a BER of 10^{-3}, an increase of the TTI from 10 to 20 milliseconds gives a gain of 7 dB. At the same BER for a TTI increase to 80 milliseconds, we observe performance losses of 1.5 dB.

In Figure 8, the performance of the simulated system for a mobile terminal speed of 100 km/h and bit rate 384 kbps is evaluated. It is clear that 40 milliseconds is the optimum TTI value. Particularly, at a BER of 10^{-3}, a performance gain of 9.5 dB and 0.8 dB is seen for a TTI increase from 10 to 20 and 20 to 40 milliseconds, respectively. Moreover, at the same BER, a performance loss of 1.5 dB is observed for further TTI increase (40 to 80 milliseconds).

Table 4: Efficient TTI for different implementation scenarios and mobile terminal speeds based on BER performance.

Mobile terminal speed (km/h)	Scenario				
	1	2	3	4	5
4	20 ms	20 ms	20 ms	20 ms	20 ms
50	80 ms	80 ms	40 ms	20 ms	—
100	80 ms	80 ms	80 ms	40 ms	—
300	80 ms	80 ms	80 ms	—	—

4.5. *Scenario 5: bit rate 2 Mbps*

In Figure 9, the BER of the system for a mobile terminal speed of 4 km/h and a high bit rate of 2 Mbps is evaluated. A value of 20 milliseconds is shown to be the optimum TTI value. Particularly, at a BER of 10^{-3}, a performance gain of 3 dB is seen for a TTI increase from 10 to 20 milliseconds. It is remarkable that at the same BER, a performance loss of 0.9 dB is observed for further TTI increase (20 to 40 milliseconds).

5. CONCLUDING REMARKS

Multimedia services represent the main novel services that 3G mobile communications can offer compared to 2G. Furthermore, high-data rates accommodate the introduction of multimedia services to 3G mobile phones. Thus, according to UMTS specifications, turbo codes represent the recently developed technology, which facilitates the efficient introduction of high-data rates in general and especially multimedia applications.

An efficient TTI choice in Rayleigh fading channels for different operating environments for UMTS 3G systems can reduce complexity and latency. These latency savings are very important, especially in the case of real-time multimedia services with tight delay constraints. Particularly, in Table 4, which summarises the simulation results presented in the previous section, the optimum TTI values are presented for the different scenarios and operating environments (mobile terminal speeds). Furthermore, our analysis shows that in terms of BER, an efficient TTI choice depends on bit rate and operating environment.

However, as can be seen in Table 4, for all five implementation scenarios considered, as a compromise between lowest BER and complexity (which means the best BER performance for the lowest TTI) and for a constant frame length of 5114 bits, a TTI = 20 milliseconds (outer block interleaver with 2 columns) is recommended for indoor or low-range outdoor operating environment.

For an urban or suburban outdoor environment and for a terminal speed of 50 km/h, a TTI = 80 milliseconds is optimum for relatively small bit rates like 28.8 kbps and 64 kbps (implementation scenarios 1 and 2), while as bit rate increases, the optimum TTI value decreases. Particularly, for increased bit rates like 144 kbps and 384 kbps (implementation scenarios 3 and 4), 40 milliseconds and 20 milliseconds are the suggested optimum values, respectively. For the same operating environment but a terminal speed of 100 km/h, 80 milliseconds is the right choice for bit rates 28.8 kbps, 64 kbps, and 144 kbps (scenarios 1, 2, and 3). For bit rate 384 kbps (scenario 4), 40 milliseconds should be used, while scenario 5 (2 Mbps) cannot be established. Again, the optimum TTI value decreases as bit rate increases, but after 144 kbps. As a general conclusion for this operating environment, we can say that for low bit rates, the optimum TTI value is the maximum value of 80 milliseconds. For increased bit rates, the optimum TTI value decreases. The decrease is faster for lower mobile terminal speeds. Moreover, in a rural outdoor operating environment, a TTI of 80 milliseconds is proposed for the first 3 scenarios, since scenarios 4 and 5 cannot be implemented.

Furthermore, our simulation results also show that choosing higher TTI than the optimum value causes negative effects on BER. Particularly, in indoor or low-range outdoor operating environment for all bit rates, the choice of a TTI value which is larger than the optimum one can cause up to 1.8 dB BER loss. This value occurs at a medium bit rate of 144 kbps. The same effect is also seen at high-bit rates (384 kbps) in urban or suburban outdoor operating environment with a performance loss of 1.5 dB. However, as in [17] for block codes, the optimum choice of block interleaver length in different fading environments is also essential for turbo codes. Finally, the simulation results show that a TTI increase from 10 to 20 milliseconds gives the highest performance gain compared to the other TTI increases (20 to 40 and 40 to 80 milliseconds) for all bit rates considered. On the other hand, the lowest gain is given by a TTI increase from 40 to 80 milliseconds. This is what we expect, since performance gain decreases as the number of columns of outer block interleaver increases [15, 16].

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1261–1271, 1996.

[2] S. S. Pietrobon, "Implementation and performance of a turbo/MAP decoder," *International Journal of Satellite Communications*, vol. 16, no. 1, pp. 23–46, 1998.

[3] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proceedings of the IEEE International Conference on Communications (ICC '95)*, vol. 2, pp. 1009–1013, Seattle, Wash, USA, June 1995.

[4] C. Chaikalis and J. M. Noras, "Reconfigurable turbo decoding for 3G applications," *Signal Processing*, vol. 84, no. 10, pp. 1957–1972, 2004.

[5] C. Chaikalis, "Reconfiguration aspects and a reconfigurable outer block interleaver for 3G applications," *Wireless Personal Communications*, vol. 41, no. 1, pp. 77–97, 2007.

[6] C. Chaikalis, *Reconfigurable structures for turbo codes in 3G mobile radio transceivers*, Ph.D. thesis, Department of Electronics & Telecommunications, University of Bradford, Bradford, UK, 2003.

[7] J. P. Woodard and L. Hanzo, "Comparative study of turbo decoding techniques: an overview," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2208–2233, 2000.

[8] H. Holma and A. Toskala, Eds., *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*, John Wiley & Sons, Chichester, UK, 2000.

[9] F. Muratore, Ed., *UMTS: Mobile Communications for The Future*, John Wiley & Sons, Chichester, UK, 2001.

[10] S. Kotsopoulos and D. Lymberopoulos, "A new medical data management concept in a hybrid cellular mobile radio communication network," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '91)*, vol. 1, pp. 674–680, Phoenix, Ariz, USA, December 1991.

[11] S. Kotsopoulos and D. Lymberopoulos, "Communication protocols and on-board processor for a new national scale private mobile radio service," in *Proceedings of IEEE International Conference on Selected Topics in Wireless Communications*, pp. 147–150, Vancouver, BC, Canada, June 1992.

[12] S. Bouzouki, S. Kotsopoulos, G. Karagiannidis, K. Chassomeris, and D. Lymberopoulos, "On optimal cell planning: case study for a DCS 1800 system," *International Journal of Communication Systems*, vol. 14, no. 9, pp. 857–870, 2001.

[13] K. Ioannou, I. Panoutsopoulos, S. Koubias, and S. Kotsopoulos, "A new dynamic channel management scheme to increase the performance index of cellular networks," *IEE Electronics Letters*, vol. 40, no. 12, pp. 744–746, 2004.

[14] E. K. Hall and S. G. Wilson, "Design and analysis of turbo codes on Rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 160–174, 1998.

[15] K. Tang, P. H. Siegel, and L. B. Milstein, "On the performance of turbo coding for the land mobile channel with delay constraints," in *Proceedings of the 33rd Asilomar Conference on Signals, Systems and Computers (ACSSC '99)*, vol. 2, pp. 1659–1664, Pacific Grove, Calif, USA, October 1999.

[16] F. Poppe, D. De Vleeschauwer, and G. H. Petit, "Guaranteeing quality of service to packetised voice over the UMT Sair interface," in *IEEE 8th International Workshop on Quality of Service (IWQOS '00)*, pp. 85–91, Pittsburgh, Pa, USA, May-July 2000.

[17] K. I. Chan and J. C.-I. Chuang, "Required interleaving depth in Rayleigh fading channels," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '96)*, vol. 2, pp. 1417–1421, London, UK, November 1996.

[18] M. Pätzold, U. Killat, F. Laue, and Y. Li, "On the statistical properties of deterministic simulation models for mobile fading channels," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 254–269, 1998.

[19] 3GPP TS 25.212 V3.9.0, "Multiplexing and channel coding (FDD)," Release 1999, March 2002.

[20] 3GPP TS 25.222 V3.8.0, "Multiplexing and channel coding (TDD)," Release 1999, March 2002.

[21] 3GPP TR 25.944 V3.5.0, "Channel coding and multiplexing examples," Release 1999, June 2001.

[22] 3GPP TS 25.201 V3.3.0, "Physical layer—general description," Release 1999, March 2002.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 57423, 9 pages
doi:10.1155/2007/57423

Research Article

Impact of Background Traffic on Speech Quality in VoWLAN

Peter Počta, Peter Kortiš, and Martin Vaculík

Department of Telecommunications, FEE, University of Žilina, Univerzitná 1, 01026 Žilina, Slovakia

Received 24 April 2007; Accepted 26 July 2007

Recommended by Stavros Kotsopoulos

This paper describes measurements of the impact of background traffic on speech quality in an environment of WLANs (IEEE 802.11). The simulated background traffic consists of three types of current traffics in telecommunication networks such as data transfer service, multimedia streaming service, and Web service. The background traffic was generated by means of the accomplished Distributed Internet Traffic Generator (D-ITG). The impact of these types of traffic and traffic load on speech quality using the test sequence and speech sequences is the aim of this paper. The assessment of speech quality is carried out by means of the accomplished Perceptual Evaluation of Speech Quality (PESQ) algorithm. The proposal of a new method for improved detection of the critical conditions in wireless telecommunication networks from the speech quality point of view is presented in this paper. Conclusion implies the next application of the method of improved detection of critical conditions for the purpose of algorithms for link adaptation from the speech quality point of view in an environment of WLANs. The primary goal of these algorithms is improving speech quality in the VoWLAN connections, which are established in the competent link.

1. INTRODUCTION

Voice over Internet Protocol (VoIP), the transmission of packetized voice over internet protocol networks, has gained much attention in recent years. It is expected to carry more and more voice traffic for its cost-effective service. However, the current Internet, which was originally designed for data communications, provides best-effort service only, posing several technical challenges for real-time VoIP applications. Speech quality is impaired by packet loss, delay, and jitter. Assessment of perceived speech quality in the IP networks becomes an imperative task to manufacturers as well as service providers.

Speech quality is judged by human listeners and hence it is inherently subjective. The Mean Opinion Score (MOS) test, defined by ITU-T P.800 [1], is widely accepted as a norm for speech quality assessment. However, such subjective test is expensive and time-consuming. It is impractical for frequent testing such as routine network monitoring.

Objective test methods have been developed in recent years. They can be classified into two categories: signal-based methods and parameter-based methods. Signal-based methods use two signals as the input to the measurements, namely, the reference signal and the degraded signal, which is the output of the system under test. They identify the audible distortions based on the perceptual domain representation of two signals incorporating human auditory models. These methods include Perceptual Speech Quality Measure (PSQM), Measuring Normalizing Blocks (MNB), Perceptual Analysis Measurement System (PAMS), and Perceptual Evaluation of Speech Quality (PESQ). Among them, PSQM and PESQ [2] were standardized by ITU-T as P.861 and P.862, respectively. Parameter-based methods predict the speech quality through a computation model instead of using real measurements. The typical model is the E-model as defined by ITU-T recommendation G.107. The E-model includes a set of parameters characterizing the end-to-end voice transmission as its input, and the output can be transformed into a MOS scale for prediction.

The algorithm PSQM is based on comparison of the power spectrum of the corresponding sections of the reference and the degraded signals. The results of this algorithm more correlate with the results of listening tests, in comparison with E-model. At present, this algorithm is no longer used because of a raw time alignment. Instead of it the algorithm PESQ is rather used. The algorithm PESQ is facilitated with very fine time alignment and one single interruption being also taken into account in the calculation of MOS. It is possible to use PESQ in mobile networks as well as in networks based on packet transmission. The disadvantages include impossibility to use it for codec with data rate lower than 4 kbps, and higher calculation load caused by recursions employed in the algorithm.

To provide person-to-person (instead of place-to-place) connections anywhere and anytime, the Internet is expected to penetrate the wireless domain. One very promising wireless network is the Wireless Local Area Network (WLAN), which has shown the potential to provide high-rate data services at low cost over local area coverage. Working in the license-exempted 2.4 GHz industrial, scientific, and medical (ISM) frequency band, the IEEE 802.11b WLAN and IEEE 802.11g WLAN offer a data rate up to 11 Mbps and 54 Mbps, respectively, while IEEE 802.11a WLAN and European Telecommunications Standard Institute (ETSI) Hiperlan/2 can support data rates up to 54 Mbps in the 5 GHz frequency band. Nowadays, the work on upcoming IEEE 802.11n WLAN standard is in progress. Its publication is currently expected in September, 2008 [3]. The IEEE 802.11n will be able to support a data rate up to 248 Mbps (2 streams) in the 5 GHz or 2.4 GHz frequency band. As a wireless extension to the wired Ethernet, WLANs typically cover a small geographic area, in hotspot local areas where the traffic intensity is usually much higher than in other areas.

The promising VoIP technology and wide deployment of WLANs are assumed to drive the application of Voice over WLAN (VoWLAN), which is expected to experience a dramatic increase in the near future [4].

One major challenge for VoWLAN is Quality of Service (QoS) provisioning. Originally designed for high-rate data traffic, WLANs may experience bandwidth inefficiency when supporting delay-sensitive and low-rate voice traffic. Hence, it is essential to enhance the QoS support capability of current WLAN standards, such as the most popular IEEE 802.11 standard. The assessment of speech quality in the current WLAN standards is very desirable opportunity for evaluation, verification, and testing of enhanced QoS methods for supporting voice services in this type of very fast developing wireless telecommunication networks.

Some works have studied the effects of packet loss and jitter on speech quality. Particularly, [5–7] examined these effects in the MOS domain for certain packet loss rate and packet sizes in the environment of IP networks. Only in [8] was examined the effect of packet loss, jitter, and number of users on perceived speech quality based on PESQ algorithm in the wireless networks, especially in 3G networks. In [9], the effect of free bandwidth on VoIP performance in IEEE 802.11b WLAN was investigated. The E-model has been used for the purpose of speech quality prediction in IEEE 802.11b WLAN. The close relationship between wireless bandwidth utilization and call quality was found. A lot of works have been performed in the area of evaluation of IEEE 802.11e Enhanced Distribution Coordination Access (EDCA) for enhanced VoWLAN performance, for example, [10, 11]. In [10], the legacy IEEE 802.11 Distribution Coordination Function (DCF) and the IEEE 802.11 EDCA were compared to show that the EDCA can provide differentiated channel access among different priority traffics. In [11], the impact of adjustment Arbitration InterFrame Space (AIFS), Contention Window (CW), and Transmission Opportunity (TXOP) on the delay and the throughput of two competing saturated stations were examined. The scheme for prioritizing voice was suggested for IEEE 802.11 infrastructure mode networks. This scheme is based on increasing AIFS value used by other stations. Some works focus on the improving of current QoS support methods, for example, [12]. In [12], a new Medium Access Control (MAC) protocol is proposed for QoS support in WLAN. The protocol is an alternative to the recent enhancement IEEE 802.11e. A new priority policy provides the system with better performance by simulating Time Division Multiple Access (TDMA) functionality.

Here, we focus on the impact of background traffic on speech quality of transmission sequences in the environment of IEEE 802.11 networks. The background traffic was generated by means of Distributed Internet Traffic Generator (D-ITG) [13]. The simulated background traffic consists of three types of current traffics. The current traffics are: data transfer service, multimedia streaming service, and Web service. Increasing traffic load causes the increasing jitter and packet loss. In general, speech quality drops with increasing packet loss and jitter. The impact of these types of traffic and traffic load on speech quality is studied in this paper. The speech quality is assessed by means of the accomplished PESQ algorithm. The proposal of a new method for improved detection of the critical conditions in telecommunication networks from the speech quality point of view is presented at the end of this paper.

The rest of the paper is organized as follows. Section 2 briefly reviews the limitations of IEEE 802.11 in supporting of voice and IEEE 802.11e EDCA. Section 3 describes the measurement scenario. Section 4 presents the measurement results. In Section 5, we propose a method for improved speech quality-centered detection of critical conditions in telecommunication networks. Section 6 concludes the paper and suggests some future studies.

2. VOICE SERVICE IN IEEE 802.11

2.1. Limitations of IEEE 802.11 in supporting voice service

As a real-time application, VoWLAN is delay-sensitive but can tolerate a certain level of packet loss. Hence, delay and jitter are the main QoS measures. Each voice packet should be transmitted within delay bounds. Also, the jitter (i.e., variation of voice packet delay) should be carefully controlled as it may degrade speech quality more severely than delay. Traditionally, an appropriately designed jitter buffer is an effective way to deal with jitter and make the voice understandable [14]. Therefore, the investigation of the impact of network performance parameters change (jitter, packet loss) on speech quality is the main goal of this paper. The change of network performance parameters is realized by means of background traffic. The investigation of these influences allows designing the methods for improving speech quality in wireless networks.

As the most popular WLAN standard, IEEE 802.11 defines a mandatory DCF and an optional centralized Point Coordination Function (PCF). DCF is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), where collision is resolved by binary exponential backoff. The optional Request-To-Send (RTS)/Clear-To-Send (CTS) dialog can also be applied to further deal with the hidden terminal problem. Mainly designed for data transmission, DCF does not take into account the delay-sensitive nature of real-time services. On the other hand, with PCF, Contention-Free Period (CFP) and a Contention Period (CP) alternate periodically. During CFP, when polled, a station gets permission to transmit its data frames. The main drawbacks of PCF include bandwidth waste when two stations in the same Basic Service Set (BSS) (which is composed of an Access Point (AP) and a number of stations associated with the AP) try to communicate with each other, uncontrolled transmission time of a polled station, and unpredictable CFP start time [15].

In order to support applications with QoS requirements, the IEEE 802.11e standard [16] enhances the original IEEE 802.11 MAC sublayer by introducing the Hybrid Coordination Function (HCF), which includes two medium access methods: Enhanced Distributed Channel Access (EDCA) and HCF Coordination Channel Access (HCCA). In this paper, we will focus only on the EDCA method because a major part of currently available APs is enabled only for the EDCA method.

2.2. IEEE 802.11e EDCA

EDCA is designed to enhance the DCF mechanism and to provide a distributed access method that can support service differentiation among traffic categories. The service differentiation is provided by assigning different contention parameters to different Access Category (AC). A QoS station can support at most eight user priorities, which are mapped into four ACs. Each AC contends channel access with different AIFS and CW settings. Compared with DCF, where DCF InterFrame Space (DIFS) is used as the common InterFrame Space (IFS) for a station to access the channel, EDCA uses different AIFS for each AC to achieve the access differentiation, where the AIFS for a given AC is defined as

$$\text{AIFS[AC]} = \text{SIFS} + \text{AIFSN} \times \sigma. \quad (1)$$

The AIFSN denotes the number to differentiate the AIFS for each AC, and σ is the time interval of a slot for IEEE 802.11 standard, which is determined according to the physical medium used. The AC with the smallest AIFS has the highest priority. EDCA assigns smaller CWs to ACs with higher priorities to bias the successful transmission probability in favor of high-priority ACs in a statistical sense. Indeed, the initial CW size (CW_{min}) can be set differently for different priority ACs, yielding higher priority ACs with smaller CW_{min}. Figures 1 and 2 illustrate these EDCA parameters and the access procedure, respectively.

In the EDCA, both the physical carrier sensing and the virtual sensing methods are similar to those in the DCF. However, there is a major difference in the countdown procedure when the medium is determined to be idle. In the EDCA, after the AIFS period, the backoff counter decreases by one at the beginning of the last slot of the AIFS (shown as the crossed time slot in Figure 2), while in the DCF, this is done at the beginning of the first time slot interval following the DIFS period [10, 16, 17].

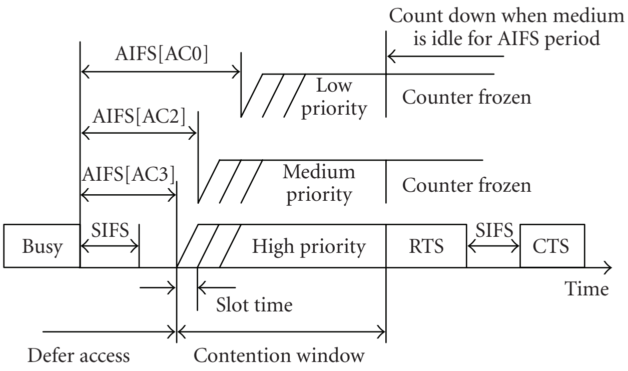


Figure 1: IEEE 802.11e EDCA mechanism parameters.

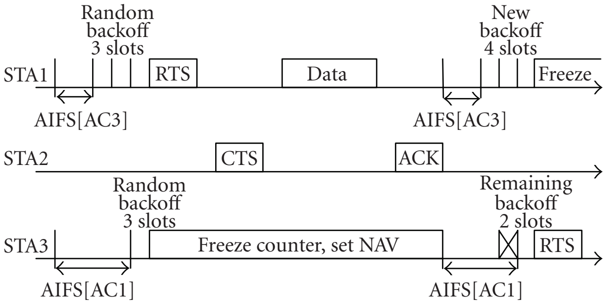


Figure 2: IEEE 802.11e EDCA channel access procedure.

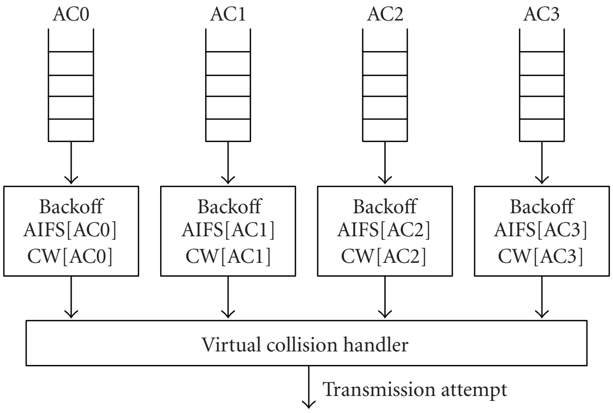

AC0
AC1
AC2
AC3
Backoff AIFS[AC0] CW[AC0]
Backoff AIFS[AC1] CW[AC1]
Backoff AIFS[AC2] CW[AC2]
Backoff AIFS[AC3] CW[AC3]
Virtual collision handler
Transmission attempt

Figure 3: ACs and virtual collision.

For a given station, traffics of different ACs are buffered in different queues as shown in Figure 3. Each AC within a station behaves like a virtual station; it contends for access to the medium and independently starts its backoff after sensing that the medium is idle for at least AIFS period. When a collision occurs among different ACs within the same station, the higher-priority AC is granted the opportunity for physical transmission, while the lower-priority AC suffers from a virtual collision, which is similar to a real collision outside the station [10, 16, 17].

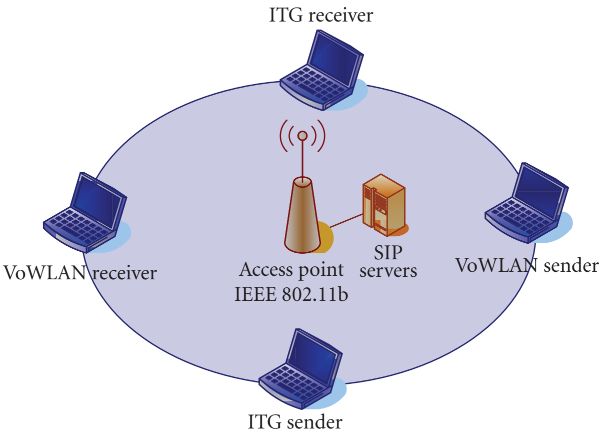


Figure 4: Measurement setup.

IEEE 802.11e also defines a TXOP limit as the interval of time during which a particular station has the right to initiate transmissions. During an EDCA TXOP, a station may be allowed to transmit multiple data frames from the same AC with a Short InterFrame Space (SIFS) gap between an ACK and the subsequent data frame [10, 16, 17].

3. MEASUREMENTS

3.1. Experimental setup

One-way VoWLAN session was established between two wireless hosts (VoWLAN sender and VoWLAN receiver), via the AP, in IEEE 802.11b WLAN (see Figure 4).

Two wireless stations (ITG sender and ITG receiver) equipped with the accomplished D-ITG were used to generate and receive background traffic. ITG sender generated the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) packets of the length 1024 bytes. Background traffic is described in Section 3.3. Voice traffic was generated using VoIP clients. Session Initiation Protocol (SIP) was used for established VoWLAN connections. For the measurement we chose the ITU-T G.729A encoding scheme [18]. In the measurement, each frame was encapsulated into a packet in turn, corresponding to a packet size of 10 milliseconds. Adaptive jitter buffer, packet loss concealment, and voice activity detector are implemented using VoIP clients.

The ITU-T recommendation P.862.3 [19] recommends to use a sequence in the range from 8-second to 30-second duration for the purpose of speech quality measurement. We decided to use the 30-second sequence for the needs of our measurements. The 30-second sequence enables to realize the precise speech quality measurement. The duration of each measurement was set equal to the length of sequence. The measurements were performed for six different testing conditions. The sequences described in Section 3.2 were utilized for transmission through the given VoWLAN connection. Finally, PESQMOS was measured by PESQ algorithm.

The adjusting of station positions and antennas was being performed until signal strength and link quality achieved by all stations were roughly similar. All stations had fixed position (no mobility) during all measurements. The signal strength and link quality were kept in the range from 85% to 100% (excellent) for all performed measurements.

Table 1: EDCA parameters used for measurements.

AC	CW_{min}	CW_{max}	AIFSN	TXOP(b) [μs]
Voice	3	7	1	3264
Video	7	15	1	6016
Best Effort	15	1023	3	0
Background	15	1023	7	0

As we aim to use IEEE 802.11 technology for the transmission of real-time services, EDCA has to be applied to support this type of services. This fact is reviewed in Section 2. The EDCA parameters are described in Table 1. The values of CW and AIFSN represent the number of time slots; each slot is of 20 -microsecond duration. The beacon frame was periodically transmitted every 1000 milliseconds.

3.2. Description of sequences

Two types of sequence were used for the purpose of the measurement. The first type of sequence is the test sequence composed of simple signals. Speech sequences are the second type of sequence. The speech sequences are composed from speech records.

3.2.1. Description of test sequence

The test sequence consists of non-speech-like (fully artificial) signals. These signals are defined in ITU-T recommendation P.501 [20] which divides them into deterministic and random signals. Development of the method for more precise detection of the critical conditions in the telecommunication networks from the speech quality point of view was our motivation for using simple signals. The duration of the test sequence is set to 30 seconds. The test sequence is composed of the following signals introduced and evaluated in [21]:

(i) sinusoidal signal of frequencies 300, 800, 1000, 1700, 2400, 3000 Hz,

(ii) square bipolar signal of frequencies 300, 400, 500, 600, 635, 670 Hz,

(iii) Gaussian white noise with $\mu = 0$ and $\delta = 0.0001; 0.001; 0.005; 0.01; 0.025; 0.05$.

The principle of the creation of the final test sequence is based on arranging the parts of the test sequence, which are shown in Figures 5 and 6. The final test sequence consists of six sections. Each section consists of five parts. The arrangement shown in Figure 5 is used once and subsequently the arrangement shown in Figure 6 is used four times to form the first section of final test sequence. The arrangement shown in Figure 6 is used five times to form the other sections of the final test sequence. The signals step by step have got the values defined above; for example, in the second section of the test sequence (from 5 seconds to 10 seconds), the signals have the following values: square bipolar signal $f = 400$ Hz, Gaussian

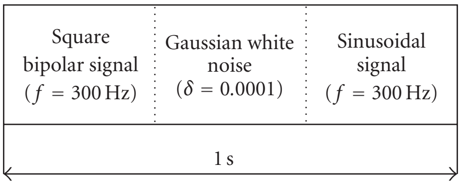

Square bipolar signal (f = 300 Hz)
Gaussian white noise (δ = 0.0001)
Sinusoidal signal (f = 300 Hz)
1 s

Figure 5: Initial part of test sequence.

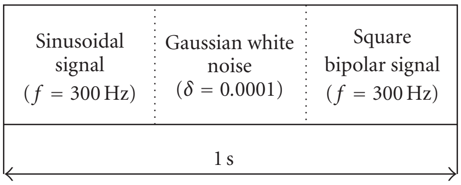

Sinusoidal signal (f = 300 Hz)
Gaussian white noise (δ = 0.0001)
Square bipolar signal (f = 300 Hz)
1 s

Figure 6: Second part of test sequence.

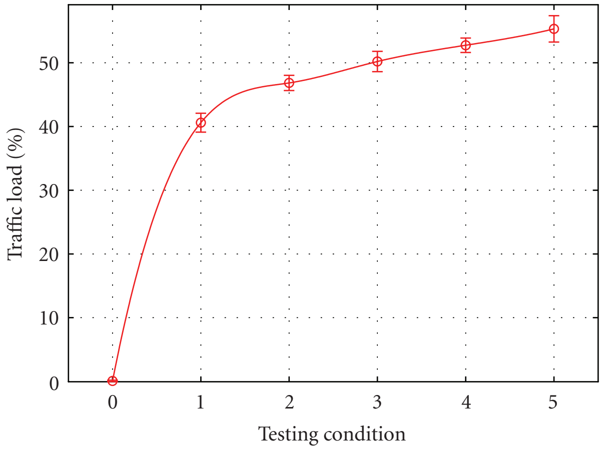


Figure 7: Traffic load for given testing conditions. The vertical bars show 95% CI (derived from 10 measurements) for each testing condition. The testing condition numbers correspond to Table 2.

white noise $\delta = 0.001$, and sinusoidal signal $f = 800$ Hz. The values of the signals in the first section of the test sequence (from 0 second to 5 seconds) are the same as those in Figures 5 and 6. The test sequence was stored in 16-bit, 8000 Hz linear PCM.

The arrangement of test sequence was derived from the criteria given by ITU-T recommendations for speech quality measurements such as P.862.3. The very important requirement is that the sequence for speech quality measurements includes active speech and silence periods. In ITU-T recommendation P.862.3 is defined that the sequence should be active in the range from 40% to 80% of its duration. The active interval of proposed test sequence is about 66% of its duration. We decided to emulate active interval by means of simple (non-speech-like) signals such as sinusoidal signal and square bipolar signal. The silence period was modelled by means of Gaussian white noise.

The choice of the test sequence arrangement for intrusive measurement of Voice Transmission Quality of Service (VTQoS) was published in [22]. The optimization of the test sequence for ITU-T G.729 encoding scheme was published in [23].

It was found that the test sequence composed of simple signals is more sensitive to the transmission impairments (jitter, packet loss, etc.) in environment of IP network than a sequence composed of speech samples. Hence, such test sequence is more suitable for more precise detection of the qualitative changes in the IP networks. We compared these two types of sequences in [24].

3.2.2. Description of speech sequences

The speech sequences selection should follow the criteria given by ITU-T recommendation P.830 [25] and ITU-T recommendation P.800 [1]. The speech sequences should include bursts separated by silence periods. They are normally of 1–3 seconds long, although this does vary considerably between languages. Certain types of voice activity detectors are sensitive only to silent periods that are longer than 200 milliseconds. Also the speech sequences should be active for 40%–80% of their duration. The speech sequences are composed of speech records. In our experiments, these speech records come from a Slovak speech database. In each set, two female and two male speech utterances were used. The speech sequences, which are of 30-second long with 57% average value of active speech interval, were stored in 16-bit, 8000 Hz linear PCM.

3.3. Background traffic

Background traffic has been generated by D-ITG. The primary goal of background traffic is to simulate standard traffic that appears in WLANs, which includes data transfer via Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP), multimedia streams for real-time applications.

The simulated background traffic includes three types of communication.

(i) “Data transfer service”, which includes FTP and other nonspecified services, is represented as information stream with constant bit rate based on TCP;

(ii) “Multimedia streaming service” represents real-time multimedia applications and therefore is based on information stream with constant bit rate. The UDP is used in this case;

(iii) “Web service” that is simulated as a sequence of separated data bursts with Poisson distribution of packet rate. The active period of burst is 400 milliseconds and the bursts appear periodically every two seconds. TCP was used for the purpose of this service.

The measurements have been performed for six different testing conditions. The selected bit rates of the three above-mentioned types of communication and average traffic load of background traffic are described in Table 2 and Figure 7. The calculation of average traffic load was based on the 100% channel rate, which is 11 Mbps for IEEE 802.11b technology.

Table 2: Performance evaluation of testing conditions.

Testing condition	Data transfer service [Mbps]	Multimedia streaming service [Mbps]	Web service [Mbps]	Average traffic load [%]
0	0	0	0	0
1	2	2.5	0.5	40.7
2	2.25	2.82	0.56	47.1
3	2.5	3.14	0.61	50.6
4	2.75	3.45	0.68	53.5
5	3	3.76	0.74	55.7

The simulation of the multimedia streaming service was carried out focusing on the impact of the service traffic on speech quality. Note that D-ITG does not allow the simulation of the multimedia streaming service using Real-time Transport Protocol (RTP), but the RTP-based streaming service has the same impact on speech quality as the streaming using UDP. The aim of this measurement was to investigate how these types of traffic and traffic load affect speech quality. The traffic load was measured by means of Wireshark network analyzer [26].

3.4. *Assessment of speech quality*

PESQMOS was evaluated by the PESQ metric [2], which is the recent ITU-T standard for objective speech quality assessment. PESQ combines merits of PAMS and PSQM99 (an updated version of PSQM), and adds new methods for transfer function equalization and averaging distortions over time. It can be used in wider range of network conditions, and gives higher correlation with subjective tests and the other objective algorithms [2, 27, 28]. Unlike the conversational model, PESQ is a listening-only model; the degraded sample is time-aligned with the reference sample during preprocessing. The PESQMOS values do not reflect the effects of delay on speech quality.

4. MEASUREMENT RESULTS

The measurement was independently performed 10 times under the same testing conditions. The PESQMOS results were averaged out and the standard deviation was kept within 0.07 PESQMOS for the test sequence and 0.15 PESQMOS for the speech sequences.

Figure 10 shows the measurement results for the test sequence and speech sequences. The graphs represent the dependence of PESQMOS change on the testing conditions. The testing conditions represent a few types of network conditions. Each network condition is described by traffic load. The increasing traffic load causes jitter and also packet loss to increase. In general, speech quality drops with increasing packet loss and jitter. Figure 7 shows the traffic load for given testing conditions. The transmission rates for given testing conditions are described in Table 2. Figure 11 shows the measurement results for the speech sequences. We can see some differences in the impact of background traffic on speech quality among the individual speech sequences in Figure 11. It is caused by a different arrangement of the speech sequences as well as by differences in duration of their active speech intervals among speakers. The impact of background traffic on the jitter (delay variation) and packet loss in VoWLAN connections is shown in Figures 8 and 9. The 1550 voice packets were approximately transmitted during one 30-second long VoWLAN connection. The average value of jitter ranged from 1.72 to 42.76 milliseconds and the total packet loss ranged from 0.81% to 23.03% for these measurements. The total packet loss consists of two components. The first component is the lost packets and the second component is the dropped packets.

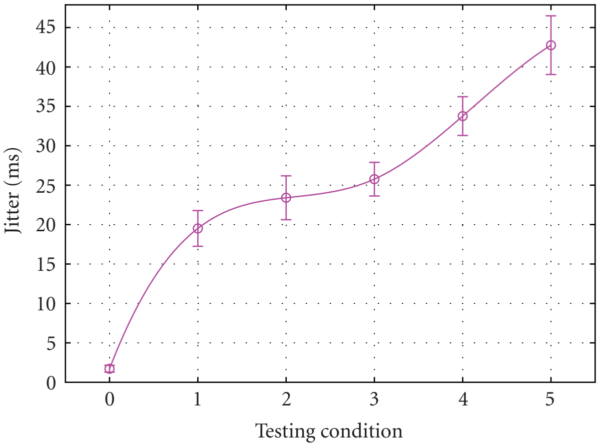


Figure 8: Impact of background traffic on average value of jitter in VoWLAN connections. Other detailed descriptions of Figure 7 apply appropriately.

Figure 10 represents only average values separately for female and male speech sequences. It can be seen that the test sequence has smoother characteristic than the speech sequences. From the speech quality point of view, the test sequence responds to network performance parameters change (jitter, packet loss, etc.) more sensitively than the speech sequences do. The effect of network performance parameters change causes higher difference in energy of the particular FFT points by using the stationary spectra signals such as sinusoidal signal and square bipolar signal. That is a reason of the increased sensitivity of the test sequence to the network performance parameters change. The enhanced sensitivity allows for carrying out the detection of qualitative changes in telecommunication networks more precisely. We aim to use it for the development of a method for improved speech

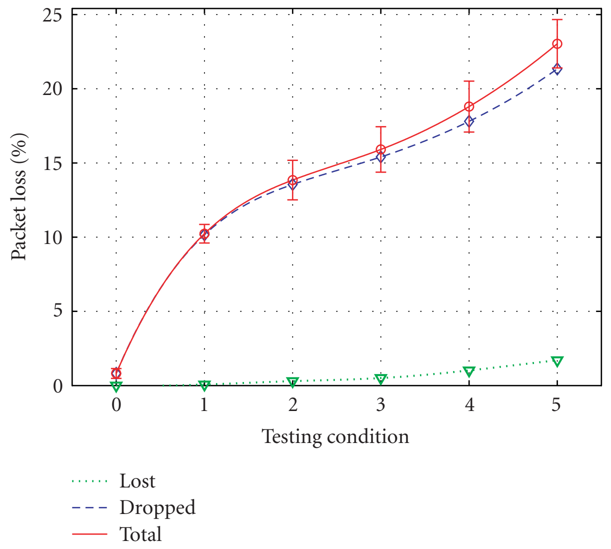


Figure 9: Impact of background traffic on packet loss in VoWLAN connections. Other detailed descriptions of Figure 7 apply appropriately.

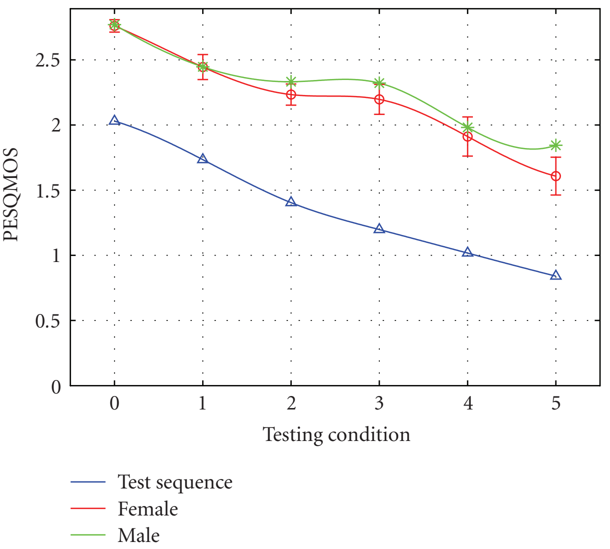


Figure 10: Impact of background traffic on speech quality of speech sequences and test sequence. Other detailed descriptions of Figure 7 apply appropriately.

quality-based detection of critical conditions in telecommunication networks.

5. PROPOSAL OF METHOD FOR IMPROVED DETECTION OF CRITICAL CONDITIONS IN WIRELESS NETWORKS FROM THE SPEECH QUALITY POINT OF VIEW

The proposal of the method for improved speech quality-centered detection of critical conditions in telecommunication networks follows the experiment described in this paper. Such improved detection allows for realizing earlier response

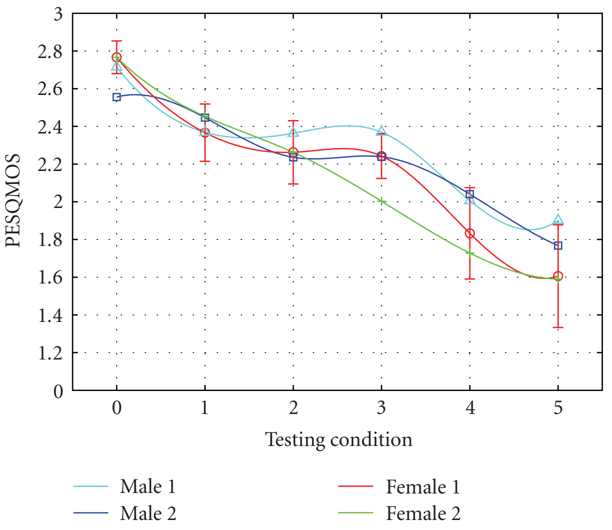


Figure 11: Impact of background traffic on speech quality of speech sequences. Other detailed descriptions of Figure 7 apply appropriately.

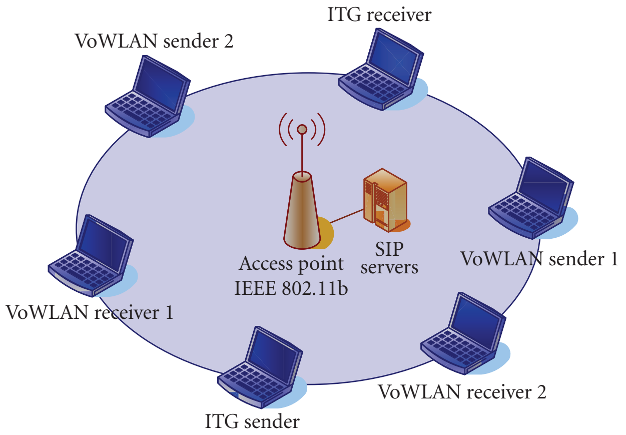


Figure 12: A typical network scenario in the environment of WLANs.

to the change of network performance parameters and it may avoid the impairment of speech quality. Figure 12 represents a typical network scenario in the environment of WLANs. VoWLAN connection no. 1 is used for the real VoWLAN transmission, and VoWLAN connection no. 2 is used for the purpose of improved detection of critical conditions in wireless telecommunication networks. This method is based on the intrusive measurement of VTQoS. The test sequence is transmitted through the VoWLAN connection no. 2. ITG stations represent the data traffic stations. The principle of the improved detection of the critical conditions is based on the simple scheme.

The measurement by using the test sequence is carried out in successive 30-second steps and the duration of the measurement is set to 30 seconds. The PESQMOS value is computed for each measurement. The PESQMOS is used as a threshold parameter for the network conditions. The

Table 3: Decision level for the relevant network conditions.

PESQMOS value	The network condition
>1.275	Standard
<1.275	Critical

threshold value for network conditions is set to 1.275 PESQMOS for the test sequence. This critical decision threshold has been derived from the results, which were obtained by the measurements described in this paper.

As seen from the measurement results (Figure 10), the PESQMOS value 1.275 for the test sequence corresponds to about 48% value of traffic load. The PESQMOS values of all the speech sequences (Figure 11) rapidly decline for the traffic load exceeding 48%. The jitter and packet loss parameters (Figures 8 and 9) increase fast above 48% of the traffic load value. Table 3 figures out the decision level for the relevant network conditions.

When the PESQMOS value for test sequence is higher than the threshold value, the network is situated in the standard network conditions from the speech quality point of view. It means that speech quality is kept within a tolerable range and the adjustment of network parameters (e.g., AIFS, CW, packet size, etc.) for low-priority services is not necessary. The critical network conditions are expected when the PESQMOS value for test sequence is below the threshold value. The impairment of speech quality is expected and the adjustment of network parameters for low-priority services is required. The acceptable speech quality can not be expected when the adjustment of the network parameters is omitted. The network parameters will be adjusted by means of upcoming algorithms for link adaptation from the speech quality point of view as a future extension of this method. These algorithms are outlined in the conclusion.

6. CONCLUSION AND FUTURE WORK

This paper investigated the impact of background traffic on speech quality in VoWLAN applications. Different traffic testing conditions were used for the purpose of the measurements. Each testing condition consists of the three types of current traffics, which exist in the telecommunication networks. The current traffics are data transfer service, multimedia streaming service, and Web service. The results show that test sequence responds to the change of the network performance parameters (jitter, packet loss, etc.) more sensitively than the speech sequences from the speech quality point of view. It allows for performing an improved detection of qualitative changes in telecommunication networks. We propose the method for such speech quality-centered detection of critical conditions in wireless networks. In the future, we aim to expand this method to algorithms for link adaptation from the speech quality point of view in the environment of IEEE 802.11 networks. The primary goal of these algorithms is improving speech quality in VoWLAN connections, which are established in the competent link. The first type of the algorithm will be based on the fragmentation of data packets. The fragmentation of the large datagrams shall create packets of a size small enough to allow for satisfying the delay requirements of the delay-sensitive traffic. Small delay-sensitive packets could be interleaved between fragments of the large datagram. The second type of the algorithm will be based on prioritizing voice by means of increasing a station’s AIFS value. The scheme for prioritizing voice, which was suggested in [11], is based on increasing AIFS value used by other stations. Increasing a station’s AIFS value results in an increased delay after every transmission on network before that station can continue decrementing its counters. This scheme can be used to protect a voice call against large numbers of data stations, maintaining throughput, mean delays, and delay distributions in a range where high voice call quality can be expected. We suppose that the increasing AIFS value will have an impact on speech quality. Future work will also focus on developing and verifying these algorithms for such speech quality-centered link adaptation.

REFERENCES

[1] ITU-T Rec. P.800, “Methods for subjective determination of transmission quality,” International Telecommunications Union, Geneva, Switzerland, 1996.

[2] ITU-T Rec. P.862, “Perceptual evaluation of speech quality,” International Telecommunications Union, Geneva, Switzerland, 2001.

[3] Official IEEE 802.11 working group project timelines, http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm.

[4] “Voice over wireless LAN: 802.11x hears the call for wireless VoIP,” Market research rep., in-stat, April 2002.

[5] L. A. R. Yamamoto and J. G. Beerends, “Impact of network performance parameters on the end-to-end perceived speech quality,” in *Proceedings of Expert ATM Traffic Symposium*, Mykonos, Greece, September 1997.

[6] B. Duysburgh, S. Vanhastel, B. De Vreese, C. Petrisor, and P. Demeester, “On the influence of best-effort network conditions on the perceivedspeech quality of VoIP connections,” in *Proceedings of the 10th International Conference on Computer Communications and Networks (ICCCN ’01)*, pp. 334–339, Scottsdale, Ariz, USA, October 2001.

[7] L. Ding and R. A. Goubran, “Assessment of effects of packet loss on speech quality in VoIP,” in *Proceedings of the 2nd IEEE Internatioal Workshop on Haptic, Audio and Visual Environments and Their Applications (HAVE ’03)*, pp. 49–54, Ottawa, Ontario, Canada, September 2003.

[8] J. Zhang, D. Yang, and Z. Quan, “Voice quality of VoIP in mobile communication systems,” in *Proceedings of the IEEE Radio and Wireless Symposium (RWS ’06)*, pp. 131–134, San Diego, Calif, USA, January 2006.

[9] M. Narbutt and M. Davis, “Effect of free bandwidth on VoIP performance in 802.1lb WLAN networks,” in *Proceedings of the Irish Signals and Systems Conference (ISSC ’06)*, pp. 123–128, Dublin, Ireland, June 2006.

[10] S. Choi, J. del Prado, N. S. Shankar, and S. Mangold, “IEEE 802.11 e contention-based channel access (EDCF) performance evaluation,” in *Proceedings of the IEEE International Conference on Communications (ICC ’03)*, vol. 2, pp. 1151–1156, Anchorage, Alaska, USA, May 2003.

[11] I. Dangerfield, D. Malone, and D. J. Leith, "Experimental evaluation of 802.11e EDCA for enhanced voice over WLAN performance," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1–7, Boston, Mass, USA, April 2006.

[12] G. S. Paschos, I. Papapanagiotou, S. A. Kotsopoulos, and G. K. Karagiannidis, "A new MAC protocol with pseudo-TDMA behavior for supporting quality of service in 802.11 wireless LANs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, Article ID 65836, 9 pages, 2006.

[13] Distributed Internet Traffic Generator, http://www.grid.unina.it/software/ITG/.

[14] P. Wang, H. Jiang, and W. Zhuang, "IEEE 802.11E enhancement for voice service," *IEEE Wireless Communications*, vol. 13, no. 1, pp. 30–35, 2006.

[15] Q. Ni, L. Romdhani, and T. Turletti, "A survey of QoS enhancements for IEEE 802.11 wireless LAN," *Wireless Communications and Mobile Computing*, vol. 4, no. 5, pp. 547–566, 2004.

[16] IEEE 802.11e-2005, "IEEE standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: wireless medium access control (MAC) and physical layer (PHY) specifications: amendment 8: medium access control (MAC) quality of service (QoS) enhancements," November 2005.

[17] Z.-N. Kong, D. H. K. Tsang, B. Bensaou, and D. Gao, "Performance analysis of IEEE 802.11e contention-based channel access," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 2095–2106, 2004.

[18] ITU-T Rec. G.729, "Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-exited linear prediction (CS-ACELP)," International Telecommunications Union, Geneva, Switzerland, 1996.

[19] ITU-T Rec. P.862.3, "Application guide for objective quality measurement based on recommendations P.862, P.862.1 and P.862.2," International Telecommunications Union, Geneva, Switzerland, 2005.

[20] ITU-T Rec. P.501, "Test signals for use in telephonometry," International Telecommunications Union, Geneva, Switzerland, 2000.

[21] P. Počta and M. Vaculík, "Method of choice of test signals for automatic intrusive measurement VTQoS," in *Proceedings of the 4th International Conference on Measurement of Speech and Audio Quality in Networks (MESAQIN '05)*, Prague, Czech Republic, June 2005.

[22] P. Počta and M. Vaculík, "Determination of optimal test sequence for automatic intrusive measurement VTQoS on environment of fixed telecommunication network," in *Proceedings of the 6th International Conference RTT*, Ostrava, Czech Republic, 2005.

[23] P. Počta and M. Vaculík, "Researching of coders influence on basic measurement signals used in optimal test sequence," *Advances in Electrical and Electronic Engineering*, vol. 5, no. 3, pp. 377–380, 2006.

[24] P. Počta and M. Vaculík, "Comparison of test sequence for intrusive measurement of VTQoS with speech sequences in the environment of IP networks," in *Proceedings of the 5th International Conference on Measurement of Speech and Audio Quality in Networks (MESAQIN '06)*, Prague, Czech Republic, June 2006.

[25] ITU-T Rec. P.830, "Subjective performance assessment of digital telephone-band and wideband digital codecs," International Telecommunications Union, Geneva, Switzerland, 1996.

[26] Wireshark network analyzer, http://www.wireshark.org/.

[27] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone network and codecs," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '01)*, vol. 2, pp. 749–752, Salt Lake City, Utah, USA, May 2001.

[28] J. Holub, R. Šmíd, and M. Bachtík, "Child listeners as the test subject—comparison with adults and P.862," in *Proceedings of the 2nd International Conference on Measurement of Speech and Audio Quality in Networks (MESAQIN '03)*, Prague, Czech Republic, May 2003.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 45798, 13 pages
doi:10.1155/2007/45798

Research Article

Fuzzy Logic Control of Adaptive ARQ for Video Distribution over a Bluetooth Wireless Link

R. Razavi, M. Fleury, and M. Ghanbari

Department of Electronic Systems Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Received 27 April 2007; Accepted 23 July 2007

Recommended by Tasos Dagiuklas

Bluetooth's default automatic repeat request (ARQ) scheme is not suited to video distribution resulting in missed display and decoded deadlines. Adaptive ARQ with active discard of expired packets from the send buffer is an alternative approach. However, even with the addition of cross-layer adaptation to picture-type packet importance, ARQ is not ideal in conditions of a deteriorating RF channel. The paper presents fuzzy logic control of ARQ, based on send buffer fullness and the head-of-line packet's deadline. The advantage of the fuzzy logic approach, which also scales its output according to picture type importance, is that the impact of delay can be directly introduced to the model, causing retransmissions to be reduced compared to all other schemes. The scheme considers both the delay constraints of the video stream and at the same time avoids send buffer overflow. Tests explore a variety of Bluetooth send buffer sizes and channel conditions. For adverse channel conditions and buffer size, the tests show an improvement of at least 4 dB in video quality compared to nonfuzzy schemes. The scheme can be applied to any codec with I-, P-, and (possibly) B-slices by inspection of packet headers without the need for encoder intervention.

1. INTRODUCTION

The enhanced data rate (EDR) of IEEE 802.15.1, Bluetooth [1] version 2.0 [2] now has a peak user payload of 2.2 Mb/s, which is the same average rate offered by some implementations of IP-TV. Therefore, a bottleneck free way exists of distribution encoded video clips from a server across an IP network to a Bluetooth master node, and, thence, over a Bluetooth wireless interconnect. Moreover, many cellular phones are also equipped with a Bluetooth transceiver and larger resolution screens of CIF (352×288) and QCIF (176×144) pixel size. Compared to IEEE 802.11 (Wi-Fi)'s typical current usage of 100–350 mA [3], Bluetooth's consumption is 1–35 mA, implying that for mobile multimedia applications Bluetooth is preferable. Nokia's proprietary Wibree technology, with a similar design to Bluetooth, uses lower-power button-cell batteries but its throughput is apparently restricted to a gross air rate of 1 Mbps. IEEE 802.15.4 (ZigBee) also has similarities to Bluetooth but as it is intended for sensor applications, its capacity is limited to 250 kbps. Bluetooth availability as a low-cost transceiver (<$5 US) makes it an attractive proposition for bespoke mobile video streaming applications, such as cordless TV within a variety of vehicles [4] or augmented reality for wearable computers [5].

However for video transmission, as in a group of pictures (GoP), slices within one picture are predicted from previous ones, noise and interference on the wireless channel may corrupt slice-bearing packets as they make the final hop before decoding and display on a mobile device. This suggests retransmission of corrupted packets should occur. Unfortunately, the default Bluetooth infinite retransmission limit for stop-and-wait automatic repeat request (ARQ) is unsuitable for delay-sensitive video streaming. This is a significant weakness, because, in general, ARQ has proved more effective than forward error correction (FEC) [6] in ensuring statistically guaranteed quality of service (QoS) over wireless networks. In Bluetooth, fast ARQ comes for free by virtue of time division duplex (TDD) polling, which is necessary for transmit/receive recovery, allowing a single-chip implementation.

Real-time delivery of video is delay-sensitive, as a frame cannot be displayed if its data arrive after their decoded deadline. A further deadline exists for reference picture types if their presence contributes to decoding of future frames [7]. In practice, a play-out buffer exists on a mobile device to account for start-up delay and also absorbs delay jitter (variation of delay). Therefore, the maximum delay permissible corresponds to the start-up delay deemed tolerable to the user. Packets may arrive too late for the frame to be displayed, and, as error concealment at the decoder is implementation dependent, the net result is poor quality video. Not only do packets arrive after their display deadline, but while retransmission takes place, other packets may either wait too long in the send buffer or in extreme cases arriving packets may find the send buffer full. ARQ adds to delay and, therefore, the number of retransmissions should be minimized. A side effect of reducing the number of retransmissions is that power usage is reduced, which is especially important when there is an imbalance of activity in Bluetooth's centralized packet scheduling scheme between a master node and its slaves.

As an alternative to the default ARQ, research reported in [8] appears to have first introduced to Bluetooth priority-based retransmission for video picture packets, though that paper went no further than a static scheme favouring Intra-coded pictures (I-pictures) at the link layer, based on similar application layer techniques to those in [9]. The need to adjust adaptively Bluetooth's default retransmission timeout for multimedia applications was established in [10], with adaptation by relatively conventional means. In our work, ARQ adaptation allows picture importance, channel conditions, and buffer fullness to be accounted for in retransmission decisions. However, adaptive ARQ is not a complete solution, as it fails to account for deadline-expired packets remaining in the send buffer while retransmission takes place. The danger is that these packets will then be transmitted simply to be discarded at the receiver. The presence of expired packets in the send buffer, just like excessive ARQ delay, contributes to queuing delay to other packets and possibly buffer overflow. Therefore, an active discard policy for deadline-expired packets is required as an addition to adaptive ARQ. In our case, the active discard policy is implemented as a deadline-aware buffer (DAB) and is also based on picture type. Picture type can be ascertained by inspection of application packet headers or real-time transfer protocol (RTP) headers, whereas accounting for picture content rather than picture importance may require intervention at the source encoder.

This paper introduces fuzzy logic control (FLC) to adaptive ARQ over Bluetooth. To the best of our knowledge, FLC has not been used for this purpose for Bluetooth and, in general, has not been applied in this way to video distribution over a wireless network. The aim of the current work is to retransmit a packet as many times as needed to ensure error-free reception but without delaying that packet beyond its deadline and without leading to send buffer overflow. The main reason for introducing FLC is that we found that its performance in terms of delivered video quality is simply better than a conventional scheme, as the tests in Section 4 illustrate. The fuzzy scheme also reduces the average number of retransmissions, its key advantage being that it can directly adapt to delay conditions, rather than simply by indirect means through active discard of expired packets. In general, a fuzzy scheme is more easily tuned by adjustment of its membership functions. By introducing two control inputs, a fuzzy scheme can trim its response. The two inputs in our scheme were buffer fullness *and* the deadline margin of the packet at the head of the Bluetooth send queue (the direct delay input). A fuzzy scheme is also well-suited to implementation on a mobile device, because not only are the decision calculations inherently simple (and can be made more so by adoption of triangular membership functions) but also by forming a look-up table (LUT) from the fuzzy control surface, its operation can be reduced to simple LUT access.

The remainder of this paper is organized as follows. Section 2 surveys related work on implementing video quality of service (QoS) through ARQ and FLC for wireless networks. Section 3 gives details of Bluetooth ARQ, FLC ARQ, and the evaluation methodology. Section 4 contains the results of the evaluation, while Section 5 draws some conclusions.

2. RELATED WORK

Fixed-size play-out buffers at the receiver are liable to underflow given that variable-bit-rate (VBR) encoded video is inherently "bursty." The burstiness occurs at multiple time scales, owing to changes in picture type within a GOP, within a scene, with variable motion, and between scene cuts. Though in fixed networks large play-out buffers (at up to several seconds of start-up delay) may be applied in video-on-demand applications, Web-based video clip distribution with click-level interactivity is less tolerant to start-up delay. On a mobile device, memory contributes significantly to the power budget [11], resulting in relatively small buffers. For example, the experiments in [12] assumed a send buffer size of fifty packets. The complexity of an adaptive rather than fixed-size play-out buffer [13], which can subsequently vary its size according to network conditions, may also deter mobile device implementation. Video smoothing was transferred to wireless networks in [14], given that the available bandwidth is even more subject to fluctuations than a fixed network. In [14] also, selected packets are given priority transmission, rather than enforce rate changes at the encoder, which discriminates against pre-encoded video. However, layered encoding is assumed, while much content exists in nonlayered format. For single-layer video, the packet type is a simple way of applying either a delay- or a loss-priority packet transmission. Packet type indicates content importance without the need for content awareness at the link layer. In [9], simple packet type discrimination is proposed as a means of implementing differentiated services QoS on the fixed Internet.

As IEEE 802.11 has no built-in QoS mechanism, there has been interest in closed-loop error control through ARQ. Though the 802.11 point coordination function (PCF) access protocol is centralized to limit delay, weaknesses in its specification [15, 16] have meant little attention has been paid to it, unlike Bluetooth's centralized control. An exception is the work in [17], where centralized control is considered most appropriate to multimedia applications and there is brief consideration (among other QoS techniques) to MAC-level adaptation of the IEEE 802.11a retry limit. Despite the title of the paper, it is not the retry limit but the ARQ mode in IEEE 802.11e's distributed coordination function (DCF) that is adapted according to channel conditions in [15]. The available modes are No ACK, stop-and-wait ARQ (as in Bluetooth), and block ARQ, whereby a number of successfully received packets are collectively acknowledged. The IEEE 802.11e variant of IEEE 802.11 also includes a buffer discard threshold, though this is not adaptive. The hybrid coordination function in IEEE 802.11e is another way of reintroducing centralized control whereby during contention-free periods, the access station assigns mobile stations transmit times. In IEEE 802.11 in general, larger frames are sent by a request-to-send (RTS)/clear-to-send (CTS) handshake [16], which reduces the impact of contention. With or without RTS/CTS, it is possible to alter the maximum number of retransmissions and in IEEE 802.11e, it is also possible to set a maximum limit to the time spent in the transmitter buffer [15]. In [12] for IEEE 802.11b, the packet loss rate over the wireless link is balanced with the loss rate from buffer overflow by incremental adjustments to the retry limit. Packet purging is also employed in [12], whereby packets dependent on lost packets are removed from queues. The problem with purging, as opposed to deadline-aware active discard, is that it appears only actionable when I-picture packets have been lost. The scheme was tested for a six-layered video stream, which increases the time taken in searching queues for packet purging, while the computational cost is less for the single queue nonscaleable video assumed in our work.

Mention should also be made of hybrid ARQ [18] in which the ARQ also contains a notification of uncorrectable errors. Further redundancy is then added to a packet before retransmission. A related technique [19], which was also made deadline aware, employs selective-repeat ARQ to control the bit-rate at the encoder. Both hybrid ARQ and ARQ bit-rate control appear not to be suitable for pre-encoded video and for the latter, the close proximity of the encoder also is needed to avoid delay in bit-rate adjustments. In [19], the error propagation impact of packets is found at the encoder and a retry limit with active discard is set for IEEE 802.11 DCF retries. This would appear to not be a general solution, as it requires collusion between the encoder and the link layer transmitter. Turning to Bluetooth, apart from [8, 10] mentioned in Section 1, adaptive ARQ seems to have been little explored except in previous research by us [20], which introduced a conventional adaptive ARQ scheme according to wireless channel conditions. Channel conditions are inferred from send buffer fullness and priority is given according to B- and P- and I-picture types and not just to I-picture packets. In Section 4, the adaptive ARQ scheme in [20] is compared with that based on FLC.

In [21], FLC was applied to Bluetooth packet scheduling on a piconet, in which multiple Bluetooth slaves are present. The work in [22] applied FLC to Bluetooth rate control through tandem controllers in an open loop system. Outside Bluetooth, FLC has found applications [23] in wireless networks with access control of a time division multiple access system. In [24], FLC is used in a random early drop (RED) router again as a form of access control. In TCP, any packet losses cause the TCP source to reduce its sending rate. Of course, deliberate and random packet losses are unsuitable for encoded video and the unbounded delay introduced by TCP's reliability mechanism also makes it unsuitable for video display. Nonetheless, in [25], an interesting application of FLC to tandem network (wired and wireless links) controlled the retransmission rate and the RED rate.

3. METHODOLOGY

Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping time-slots of 625 microseconds in duration. Every Bluetooth frame consists of a packet transmitted from a sender node over 1, 3, or 5 timeslots, while a receiver replies with a packet occupying at least one slot, so that each frame has an even number of slots. Therefore, in the case of master-to-slave transmission, a single slot packet serves for a link layer stop-and-go ARQ message whenever a corrupted packet payload is detected. The timeout or retransmission limit value by default is set to an infinite number of retransmissions. On general grounds, this is unwise in conditions of fast fading causing by multipath echoes, as error bursts occur. Another source of error bursts is cochannel interference by other wireless sources, including other Bluetooth piconets, IEEE 802.11b,g networks, cordless phones, and even microwave ovens. Though this has been alleviated to some extent in version 1.2 of Bluetooth by adaptive frequency hopping [26], this is only effective if interference is not across all or most of the 2.402 to 2.480 GHz unlicensed band. IEEE 802.11b operating in direct sequence spread spectrum mode may occupy a 22 MHz subchannel (with 30 dB energy attenuation over the central frequency at ±11 MHz) within the 2.4 GHz band. IEEE 802.11g employs orthogonal frequency division multiplexing to reduce intersymbol interference but generates similar interference to 802.11b. Issues of interference might arise in apartment blocks with multiple sources occupying the 2.4 GHz band or when higher-power transmission occurs such as at WiFi hotspots.

3.1. Bluetooth ARQ

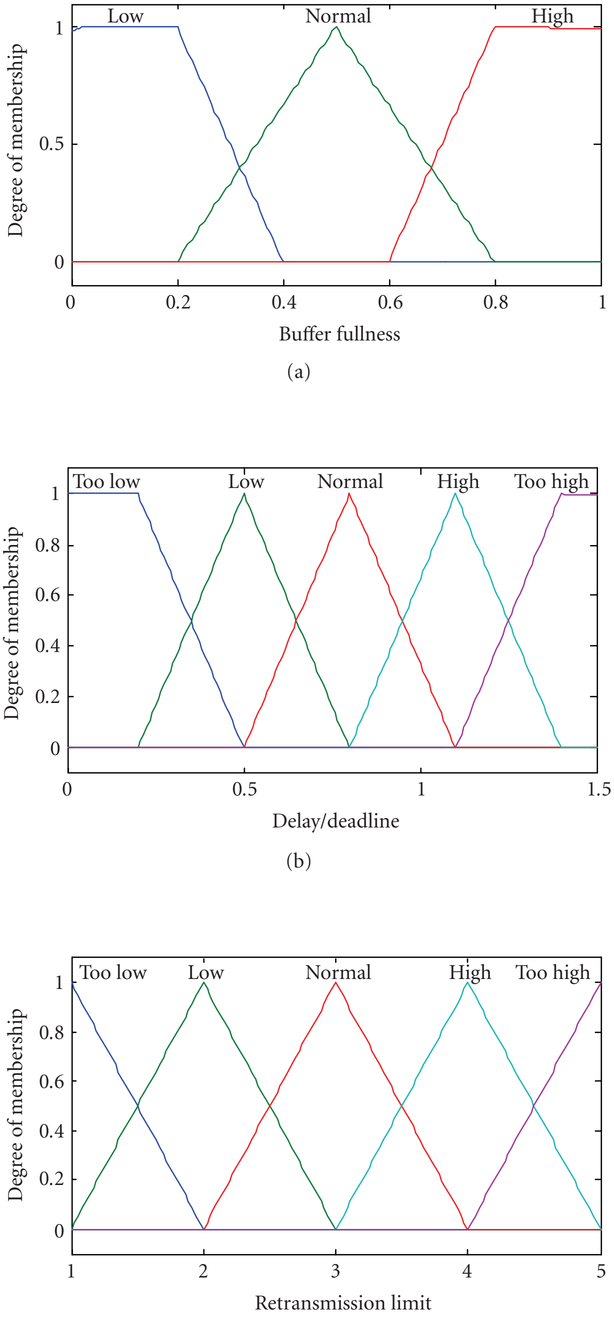
For Bluetooth, an ARQ may occur in the following circumstances [27]: (a) failure to synchronize on the access header code; (b) header corruption detected by a triple redundancy code; (c) payload corruption detected by CRC; (d) failure to synchronize with the return packet header; (e) header corruption of the return packet. Notice that a faulty ARQ packet can itself cause retransmission. The main cause of packet error [27], however, is (c) payload corruption. As mentioned in Section 1, the default value of the ARQ retransmission timeout in most Bluetooth chipsets [10] is set to infinity, resulting in unlimited retries. In [10], a fixed retransmission timeout and an adaptive retransmission timeout were considered. The disadvantage of a fixed retransmission timeout is that it is difficult to arrive at a value that avoids either excessive delay or excessive packet drops in *all* circumstances. The adaptive retransmission timeout, which was upper and lower bounded, was based, in [10], on a smoothed round-trip time. The retransmission timeout was adapted downwards or upwards if the new smoothed round trip time, respectively, is less than or more than the previous smoothed round trip time.

3.2. *Fuzzy logic control of ARQ*

This section briefly introduces FLC before introducing FLC ARQ. In a fuzzy subset, each member is an ordered pair, with the first element of the pair being a member of a set S and the second element being the possibility, in the interval $[0, 1]$, that the member is in the fuzzy subset. This should be compared with a Boolean subset in which every member of a set S is a member of the subset with probability taken from the set $\{0, 1\}$, in which a probability of 1 represents certain membership and 0 represents nonmembership. In a fuzzy subset of (say) "buffer fullness," the possibility that a buffer with a given fullness taken from the set S of fullness may be called high is modeled by a membership function, which is the mapping between a data value and possible membership of the subset. Notice that a member of one fuzzy subset can be a member of another fuzzy subset with the same or a different possibility. Membership functions may be combined in fuzzy "if then" rules to make inferences such as if x is high and y is low, then z is normal, in which high, low, and normal are membership functions of the matching fuzzy subsets and x, y, z are linguistic variables (names for known data values). In practice, the membership functions are applied to the data values to find the possibility of membership of a fuzzy subset and the possibilities are subsequently combined through defuzzification, which results in a crisp (nonfuzzy) value.

For the adaptive ARQ FLC, there are two inputs: buffer fullness and the normalized delay of the head of the queue packet. Bluetooth buffer fullness is a preferable measure (compared to delay or packet loss) of channel conditions and of buffer congestion, as was established in [28]. Buffer fullness is available to an application via the host controller interface (HCI) presented by a Bluetooth hardware module to the upper layer software protocol stack. Retransmissions avoid the effect of noise and interference but also cause the master's send buffer queue to grow, with the possibility of packet loss from send buffer overflow.

The retransmission timeout of the packet at the head of the Bluetooth send queue will affect the delay of packets still to be transmitted. Therefore, the second FLC input moderates the buffer fullness input. The assigned membership functions, which were arrived at heuristically, are shown In Figures 1(a) and 1(b), and once found were fixed. The buffer fullness range in Figure 1(a) is $[0, 1]$ corresponding to a percentage fullness. In Figure 1(b), the horizontal axis represents the delay time of the packet at the head of the queue divided by the display deadline. In Figure 1(b), unit delay/deadline corresponds to expiration of playout deadline. It is important to note that any packet in the send buffer is discarded if its deadline has expired (Section 3.3). However, this takes place after the fuzzy evaluation of the desired ARQ retransmission timeout. In practice, the inputs to the FLC were sampled versions of buffer fullness and packet delay/deadline to avoid excessive ARQ retransmission timeout oscillations over time. The sampling interval was every 20 packets. Table 1 shows the "if $\cdots$ then" rules that allow input fuzzy subsets to be combined to form an output. Notice that more than one rule may apply because of the fuzzy nature of subset membership.



(a)

(b)

(c)

Figure 1: Fuzzy membership functions: (a) input buffer fullness (b) input delay/deadline (c) output retransmission limit.

The inputs were combined according to the well-known Mamdani model [29] to produce a single output value. The standard center of gravity method was employed to resolve to a crisp output value according to the output membership functions shown in Figure 1(c). Notice that the output in Figure 1(c) corresponds to the full range of possibilities, whereas if a deadline-aware-buffer (Section 3.3) is incorporated, then discard of expired packets will mean that the higher end of the output range will not occur. In Figure 1(c),

Table 1: FLC If · · · then rules used to identify output fuzzy subsets from inputs.

		Delay/Deadline				
		Too low	Low	Normal	High	Too high
Buffer Fullness	High	Normal	Normal	Low	Too low	Too low
	Normal	High	High	Normal	Low	Too low
	Low	Too high	High	Normal	Low	Too low

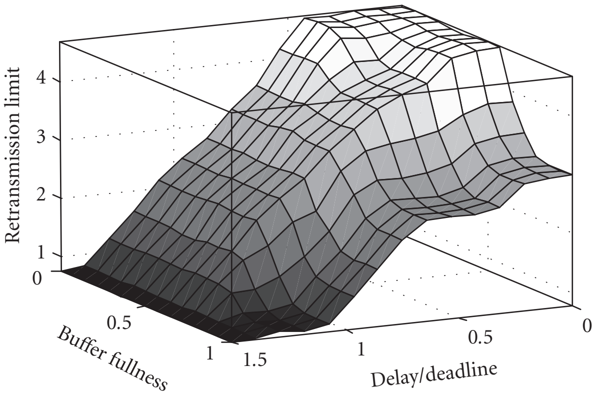


Figure 2: Control surface resulting for FLC ARQ.

the retransmission limits correspond to the retransmission timeout of the current packet. Clearly, a packet can only be retransmitted an integer number of times but the crisp output may result in a real-valued number. This difficulty was resolved by generating a random number from a uniform distribution. If the random number was more than the fractional part of the crisp output value, then that value was rounded down to the nearest integer, otherwise it was rounded up. The advantage of this procedure over simple quantization is that, in the long term, the resolution of the number of transmissions will be higher and the mean value will converge to a desired output level. The output value was subsequently scaled according to the priority of the packet's picture type.

For reasons of error resilience, encoded video is transmitted as a repeating sequence of GOP [30], with the start of each GOP formed by an I-picture. An I-picture is the basis for prediction of all other pictures in the GOP (usually 12 to 15 pictures in all) and, hence, its loss has drastic consequences for all other pictures. P-pictures also form the basis for predictions but are not essential for the reconstruction of other pictures within the GOP (as other I- or P- anchor pictures retained in a decoded buffer can be applied). Lastly, the third type of picture, the bipredictive B-picture, has no predictive value.

A simple scaling of 5 : 3 : 1 was applied, respectively, for I-, P-, B-pictures, given a choice of five maximum retransmissions. Normalizing this scaling to a factor of one for I-picture packets, results in a ratio of 1 : 0.6 : 0.2, giving for maximum retransmissions five, just $1 \times 5 = 5$ retransmissions for I-picture packets but $0.6 \times 5 = 3$ maximum retransmissions for a P-picture packet, and $0.2 \times 5 = 1$ maximum retransmissions for a B-picture packets. In practice, the scaling is applied to the crisp value output after defuzzification. For example, if the crisp output value was 2.3, and a P-picture packet was involved, then the value after scaling is $2.3 \times 0.6 = 1.38$. Then, the random-number-based resolution results in two retransmissions if the random number is less than or equal to 0.38 and one retransmission otherwise. It should be mentioned that a maximum value of five retransmissions was also adopted in the priority queueing tests in [12], albeit for an IEEE 802.11 wireless network.

The fuzzy control surface is represented in Figure 2, as derived from the Matlab fuzzy toolbox v. 2.2.4. A scaled version of this output surface is applied with scaling dependent on picture type. As mentioned in Section 1, by means of an LUT derived from the surface, a simple implementation becomes possible.

3.3. Deadline-aware buffer

In the conservative send buffer discard policy of this paper, all packets of whatever picture type have a display deadline which is the size of the play-out buffer expressed as a time beyond which buffer underflow will occur. In a conservative policy, in which there is no need for play-out buffer fullness updates, the deadline is set as the maximum time that the play-out buffer can delay the need for a packet. Play-out buffers are normally present to smooth out jitter across a network path (if the Bluetooth master was also an access point) and in this paper the size is assumed to be constant. In the simulations of Section 4, the display deadline was set to 0.10 second.

In addition to the display deadline, all I-picture packets have a decoded deadline, which is the display time remaining to the end of the GOP. This is because reference pictures (I- or P-) are still of value to the receiver as they serve in the decoding of subsequent pictures, even after their display deadline has elapsed. Thus, for a 12 frame GOP, this is the time to display 11 frames, that is, 0.44 second at 25 frame/s. For P-picture packets, the time will vary depending on the number of frames to the end of the GOP. For B-pictures the decoded deadline is set to zero.

The decoded deadline is added to the display deadline and a packet is discarded from the send buffer after its total deadline expires. By storing the GOP end time, an

IBBP$_1$BBP$_2$BBIBBP$_3$ $\cdots$

(a)

IP$_1$BBP$_2$BBIBBP$_3$BB$\cdots$

(b)

Figure 3: I-, B-, P-picture reorderings: (a) display order, (b) send buffer output order.

implementation performs one subtraction to find each decoded deadline. Account has been taken of I- B- P-picture reordering at encode and send buffer output, Figure 3, which has an effect on buffer fullness. Reordering is introduced to ensure that reference pictures arrive and can be decoded before the dependent B-pictures. In the discard policy, packet handling and propagation delay is assumed (optimistically) to be constant. In all experiments, the buffer queue discipline is assumed to be first-in-first-out.

In analytical terms, consider a buffer filled with packets from just one type of picture, with total deadline time D second. (For I- and B-picture packets D is fixed but for P-pictures a mean total deadline time might be substituted.) Assume that the Bluetooth frame (outgoing packet together with incoming single time-slot acknowledgement packet) handling and propagation time is t so that total time before packet expiration is $(D-t)$. As dynamic packetization is used in the simulations in Section 4, the Bluetooth frame-size, S, can be taken as constant. Unfortunately, for VBR streams, although the display rate is constant, the output data rate varies by time. Nevertheless, the per packet consumption time is S/R if only one transmission is necessary and $(S \times N)/R$ if a maximum of N retries are necessary. Additionally, the time before packet expiration is further reduced by the need to wait for $(N-1)$ prior transmissions without consumption, that is, reduced to $(D-t-(N-1)t) = D - Nt$. Dividing the time before expiration by the packet consumption time gives the sustainable send queue length, Q, before packet discard becomes necessary:

$$Q = \left\lfloor \frac{((D-Nt) \times R)}{(N \times S)} \right\rfloor. \tag{1}$$

With suitable adjustments to take account of different packet types, (1) might serve to regulate the flow from a compliant encoder (or transcoder) but, in case of a VBR video stream, adaptive ARQ is a convenient way to increase the available queue length by varying N in (1).

3.4. Channel model

A number of studies [31, 32] have established that the validity of employing a first-order Markov chain is a good approximation in modeling the packet-level error process in a fading channel. A Gilbert-Elliott [33, 34] two-state discrete-time, ergodic Markov chain modeled the wireless channel error characteristics between a Bluetooth master and slave node. By adopting this model it was possible to simulate burst errors of the kind that cause problems to an ARQ mechanism. The Gilbert-Elliott model was also employed for modeling the channel in a study of go-back-n and selective ARQ [35] in a CDMA spread-spectrum system and in [36] was applied to the same version of Bluetooth as herein. The mean duration of a good state, T_g, was set at 2 seconds and in a bad state, T_b was set to 0.25 second. In units of 625 microseconds (the Bluetooth time slot duration), $T_g = 3200$ and $T_b = 400$, which implies from

$$T_g = \frac{1}{1 - Pgg}, \qquad T_b = \frac{1}{1 - Pbb} \tag{2}$$

that, given the current state is good (g), Pgg the probability that the next state is also g is 0.9996875 and Pbb, given the current state is bad (b), the probability that the next state is also b is 0.9975. The transition probabilities, Pgg and Pbb, as well as the BER, are approximately similar to those in [37], but the mean state durations are adapted to Bluetooth. At 3.0 Mb/s, the bit error rate (BER) during a good state was set to $a \times 10^{-5}$ and during a bad state to $a \times 10^{-4}$, where a is a scaling factor.

3.5. Simulation setup

This research employed the University of Cincinatti Bluetooth (UCBT) extension (download is available from http://www.ececs.uc.edu/~cdmc/UCBT) to the well-known ns-2 network simulator (v. 2.28 used). The UCBT extension supports Bluetooth EDR but is also built on the air models of previous Bluetooth extensions such as BlueHoc from IBM and Blueware. All links were set at the maximum EDR 3.0 Mbps gross air rate. Simulation runs were each repeated ten times and the results averaged to produce summary statistics.

The simulations were carried out with input from an MPEG-2-encoded bitstream at a mean rate of 1.5 Mbit/s for a 30-second video clip with moderate motion, showing a newsreader and changing backdrop, which we designate "News." Peak signal-to-noise ratio (PSNR) was found by reconstructing with a reference MPEG-2 decoder. The display rate was 25 frame/s resulting in all in 750 frames in each run. The source video was common intermediate format (CIF)-sized (366 × 288 pixel) with a GOP structure of $N = 12$, and $M = 3$ (M is the number of pictures from the I-picture to the first P-picture, i.e., including two B-pictures). In [38], fully filled Bluetooth packets were formed using maximal bandwidth five time-slot packets, regardless of slice boundaries. While this results in some loss in error resilience, as each MPEG-2 slice contains a decoder synchronization marker, in [38], it is shown that the overall video performance is superior to the choice of smaller packet sizes.

Figure 4 summarizes the testing environment. The source videos are encoded to act as traces, which are introduced into the NS-2 with UCBT extension simulator. The simulator is configured to output a trace from which packet statistics are extracted. The trace determines which of the video-bearing packets were lost. Together with the input video bitstream and its packetization details, this serves to recreate an encoded video bitstream as it would have been received according to the simulations conditions. The bitstream is then decoded and the resulting video is compared with the original to find the delivered video quality.

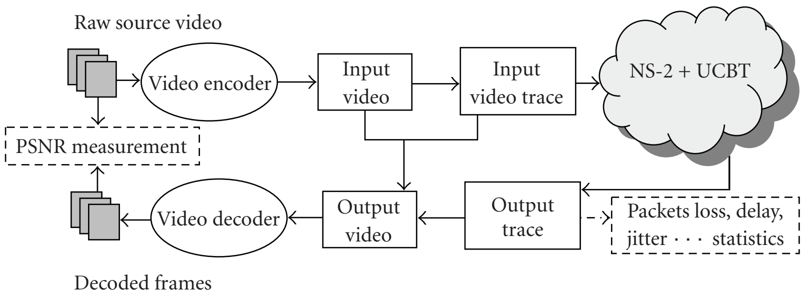


Figure 4: The testing environment for the experiments of Section 4.

4. SIMULATION RESULTS

This section examines FLC ARQ's performance in three different directions: (1) when the wireless channel conditions are varied; (2) when the send buffer size is altered; and (3) when different types of video are transmitted in addition to the "News" sequence described in Section 3.5.

As a point of comparison, FLC ARQ is compared with an adaptive ARQ scheme [20]. In this scheme, the ARQ retransmission timeout can be adaptively selected in terms of number of retransmissions allowed, to avoid further delay after the packet enters the tail of the send buffer. A threshold is set, that is, the maximum number of retransmissions allowed when the buffer is empty. The maximum number of retransmissions is subsequently changed by a factor depending on the buffer fullness reported by the Bluetooth HCI. The formula employed is summarized as

$$N = \frac{m(c-f)}{c}, \tag{3}$$

where N is the maximum number of retransmissions allowed—the retransmission timeout, m is the maximum integer-valued number of retransmissions allowed when the buffer is empty, f is the number of packets buffered in the send buffer (buffer fullness), and c is the buffer capacity. Notice that N is real valued, but is further adjusted in the same manner as the FLC output (Section 3.2), that is, by generating a random number and rounding up or down according to a comparison with the fractional part. When the buffer is empty, $f = 0$, then the maximum number of retransmissions occurs, whereas when the buffer approaches full occupation, then no retransmissions may occur. The smaller the value of m becomes the sooner this latter event occurs. In the comparative tests, the value of m varies according to the picture type of the packet, allowing a form of priority-based adaptive ARQ. The same maximum retransmission weighting was applied as for the FLC scheme (Section 3.2), namely, $m = 5, 3, 1$. respectively, for I-, P-, B-picture packets.

To examine the response to changing channel conditions, the BERs for the good and bad states of Section 3.4 were scaled by an integer-valued factor a, while Pgg and Pbb retained the values set in Section 3.4. In this way, the effect of differing (deteriorating) channel conditions could be assessed. Figure 5 plots number of transmissions needed to achieve a successful transmission according to factor a, with a buffer size of 150. The superioriority of the FLC with DAB scheme is confirmed in Figure 5, as it is in terms of mean packet delay in Figure 6. A feature of this plot is that for the worst channel conditions ($a = 5$), the average delay actually extends beyond the display deadline when the infinite ARQ with DAB scheme is employed. This is explained by the weighting given to the average by delayed I- and P-picture packets that attract an extra decode deadline, given that many B-picture packets are delayed up to their display deadline.

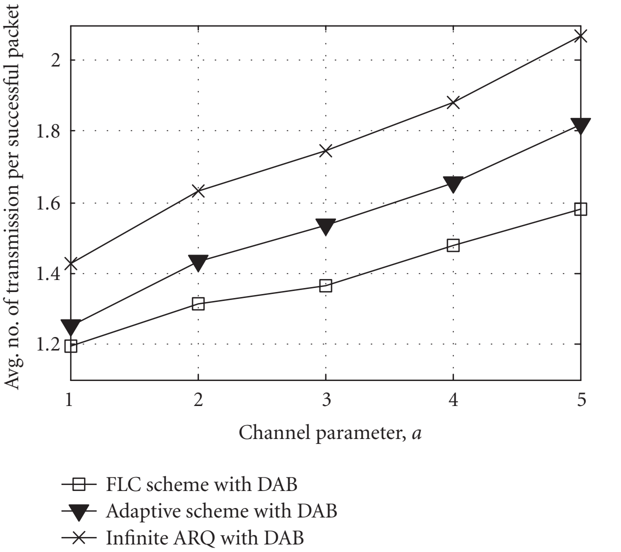


Figure 5: Mean number of transmissions by ARQ with DAB scheme, according to channel conditions.

Table 2: Various ARQ schemes applied in Figure 9.

Index	ARQ control scheme
1	FLC with DAB
2	Adaptive ARQ with DAB
3	Adaptive ARQ without DAB
4	Infinite ARQ with DAB
5	Infinite ARQ without DAB
6	No ARQ

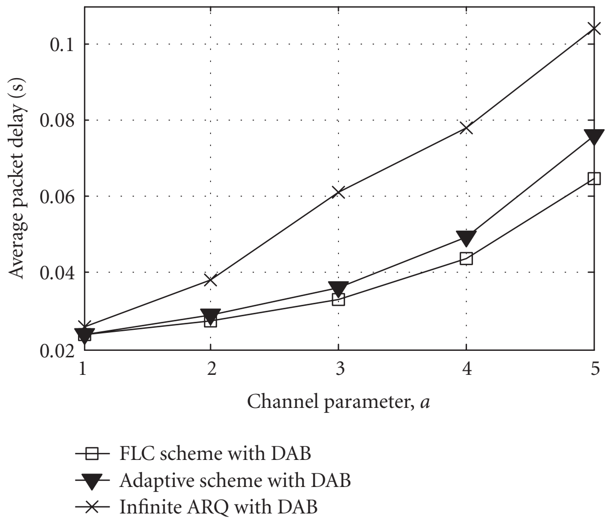


FIGURE 6: Mean packet delay ARQ with DAB scheme, according to channel conditions.

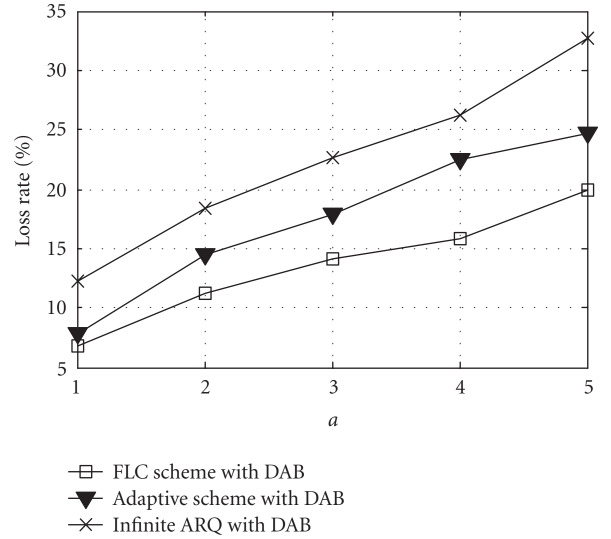


FIGURE 7: Mean packet loss rates for three DAB-based schemes, with changing channel conditions.

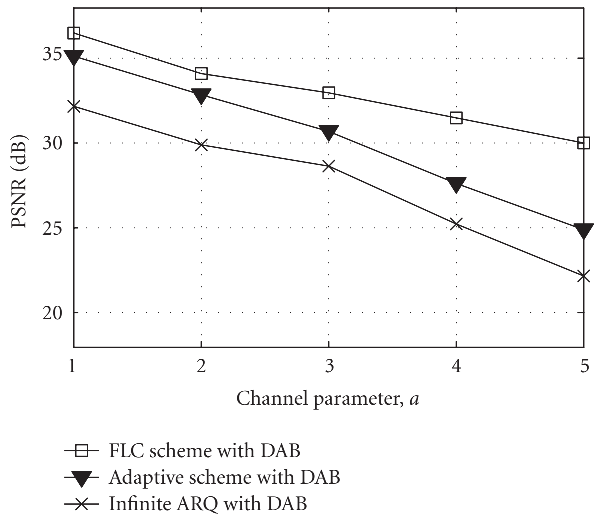


FIGURE 8: Mean PSNR for three DAB-based schemes, with changing buffer size.

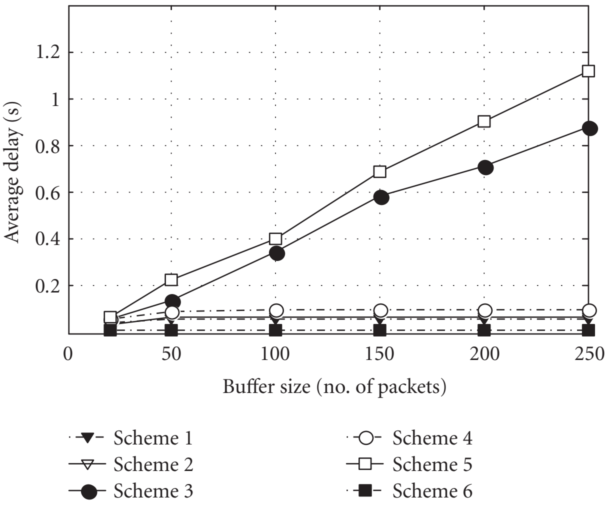


FIGURE 9: The impact of buffer size on the various ARQ control schemes in Table 2.

The superiority of the FLC with DAB scheme according to channel conditions is confirmed by Figure 7, again with a large number of simulation runs (fifty) to approach converged rates. As one might expect, mean PSNR follows a similar trend to packet loss rate, as illustrated by Figure 8. For our purposes, PSNR suffices as a measure of received video quality, as it certainly indicates an improvement (or little to no improvement) from applying a technique. PSNR is, of course, a relative technique and only applies to comparisons for the same video sequence.

There is no guarantee that the Bluetooth send buffer size will be set favourably, given the need for other types of traffic to utilize the link. In general, as the send buffer size is increased then more packets accumulate during a bad state, leading to an increase in the number of packets retained up to their deadline. In Figure 9, the delay as a consequence of six ARQ control schemes is compared with the schemes listed in Table 2. In the buffer-size experiments, factor a was set to 3. Whether adaptive ARQ or infinite ARQ is employed, if there is no DAB, then, as buffer size increases, mean delay also increases. Other schemes in the mean do not differ greatly, and in Figure 9, the plots for schemes 1 and 2 partially overlap. The time to recover from a bad state to subdeadline levels of delay is significant, as the larger the buffer the slower the recovery, as Figure 10 shows for adaptive ARQ without DAB. Therefore, one can conclude that inclusion of a DAB is clearly vital to any scheme transporting video.

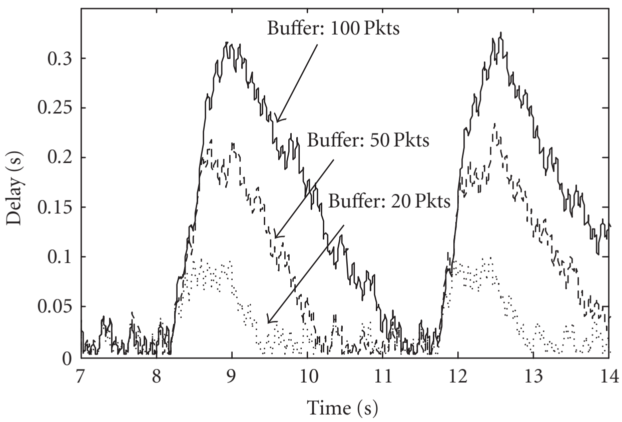


Figure 10: Example of delay recovery behavior for scheme 3 in Table 2.

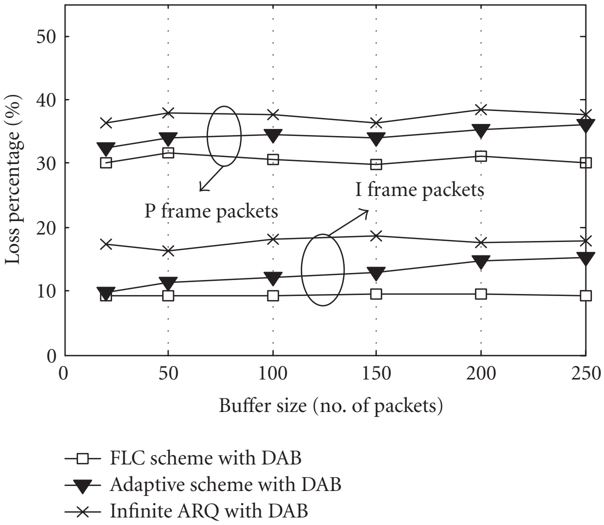


Figure 11: Packet loss rates for three DAB-based schemes, with changing buffer size.

Table 3: Distribution of packets by picture type in the test video.

Picture type	Percentage packets
I	17.97
P	37.93
B	44.10

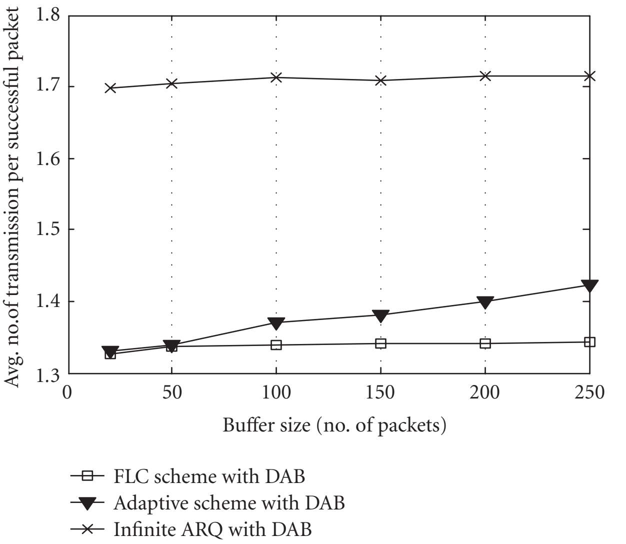


Figure 12: Mean number of transmission for three DAB-based schemes with changing buffer size.

Turning from delay to received video quality for DAB-enabled schemes, in Figure 11 the loss rates are analyzed by packet picture type. The input video is the same as in Section 3.5. The original distribution of packets is shown in Table 3. For infinite ARQ with DAB, no distinction is made in terms of the ARQ policy between different picture types, and, therefore, the discard rate reflects the ratio of picture types recorded in Table 3. As the distribution of bad states in the two-state channel model is erratic, it is only when fifty independent runs were simulated that the loss rate distribution for the infinite ARQ with DAB converged, as in Figure 11. From Figure 11, clearly the performance of the FLC with DAB scheme is significantly improved upon adaptive ARQ with DAB, and becomes more so as the buffer size is increased. In the adaptive ARQ with DAB scheme, the sole control upon retransmission is the buffer fullness ratio. As the buffer size increases, the DAB policy keeps the number of buffered packets to previous levels, and hence the buffer fullness factor reduces. Therefore, packets can be retransmitted a greater number of times and if these are B-picture packets then retained I- and P-picture packets are more likely to pass their expiration deadline. This understanding is confirmed by Figure 12, which shows that, as buffer size increases, the mean number of transmissions increases under the adaptive ARQ scheme but remains stable under FLC ARQ. The resulting impact on PSNR is recorded in Figure 13. While, the default Bluetooth scheme, even with a DAB, results in poor-quality received video, the adaptive ARQ with DAB scheme's performance is buffer-size-dependent. The video quality under the FLC with DAB scheme is reasonable and fairly constant in the mean, despite the particularly poor channel conditions that occur in bad states.

Figure 14 is a timewise comparison of PSNR of infinite ARQ, adaptive ARQ, and FLC ARQ, all with a DAB in place. The buffer size was 50 and factor a was set to 3 in the results of Figure 14. From Figure 14(a) it is clear that infinite ARQ even with a DAB in place, represents a much poorer experience for the viewer, especially from frame 300 to 700. The level of adaptive ARQ with DAB video quality is generally closer to 30 dB rather than 40 dB, whereas under FLC ARQ approaches a level of 40 dB, though there are some drops in quality.

BER is related to packet error rate (PER) according to packet payload length L (assuming that payload corruption

Table 4: Summary statistics for various ARQ control schemes, all with DAB, giving mean PSNR in dB, mean packet loss rate as a percentage, mean delay in seconds (s), with buffer size B in numbers of Bluetooth packets, and channel condition factor a being higher for worse wireless channel conditions. The results are for three different video sequences.

	ARQ Scheme	News			Friends			Football		
		PSNR	Loss	Delay	PSNR	Loss	Delay	PSNR	Loss	Delay
$B = 50$ $a = 2$	**FLC**	**34.11**	**11.45**	**0.027**	**33.56**	**12.28**	**0.028**	**32.97**	**12.41**	**0.028**
	Adaptive	33.08	13.85	0.029	32.90	15.02	0.029	32.14	15.65	0.029
	Infinite	29.92	18.05	0.038	28.79	19.89	0.040	27.80	20.31	0.041
$B = 150$ $a = 2$	Fuzzy	34.09	11.29	0.027	33.62	12.17	0.028	33.09	12.38	0.028
	Adaptive	32.87	14.59	0.029	32.55	15.83	0.029	32.03	16.31	0.029
	Infinite	29.89	18.41	0.038	28.82	19.81	0.040	27.86	20.30	0.041
$B = 50$ $a = 3$	**FLC**	**32.91**	**14.17**	**0.033**	**32.14**	**16.24**	**0.035**	**31.44**	**17.10**	**0.036**
	Adaptive	31.19	16.90	0.035	31.02	18.76	0.037	30.67	19.60	0.038
	Infinite	28.66	22.57	0.061	26.87	24.30	0.066	26.22	25.14	0.068
$B = 150$ $a = 3$	Fuzzy	32.97	14.05	0.033	32.20	16.19	0.035	31.45	17.07	0.036
	Adaptive	30.68	17.84	0.036	30.76	19.11	0.039	30.17	20.11	0.040
	Infinite	28.67	22.58	0.062	26.91	24.27	0.066	26.23	25.10	0.069
$B = 50$ $a = 5$	**FLC**	**29.89**	**20.12**	**0.065**	**28.76**	**21.90**	**0.068**	**27.63**	**22.30**	**0.072**
	Adaptive	25.32	22.25	0.073	24.11	23.57	0.076	23.00	25.41	0.081
	Infinite	22.04	33.14	0.104	20.80	34.97	0.110	19.46	35.10	0.112
$B = 150$ $a = 5$	Fuzzy	30.01	19.83	0.065	28.80	21.85	0.068	27.63	22.28	0.073
	Adaptive	24.89	24.70	0.076	23.71	24.12	0.079	22.67	26.08	0.085
	Infinite	22.11	32.97	0.105	20.89	34.70	0.111	19.51	25.33	0.112

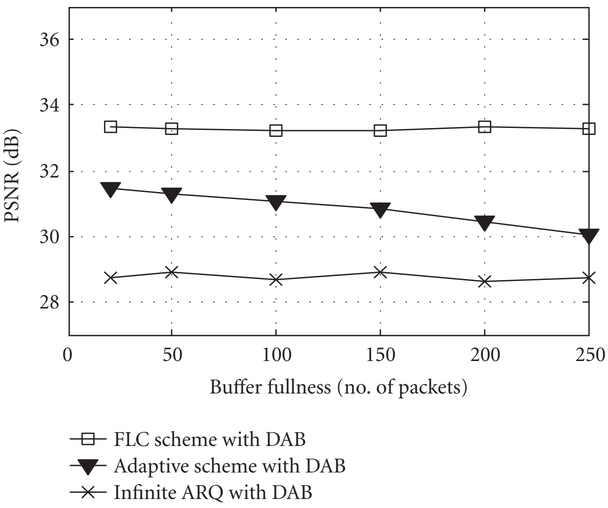


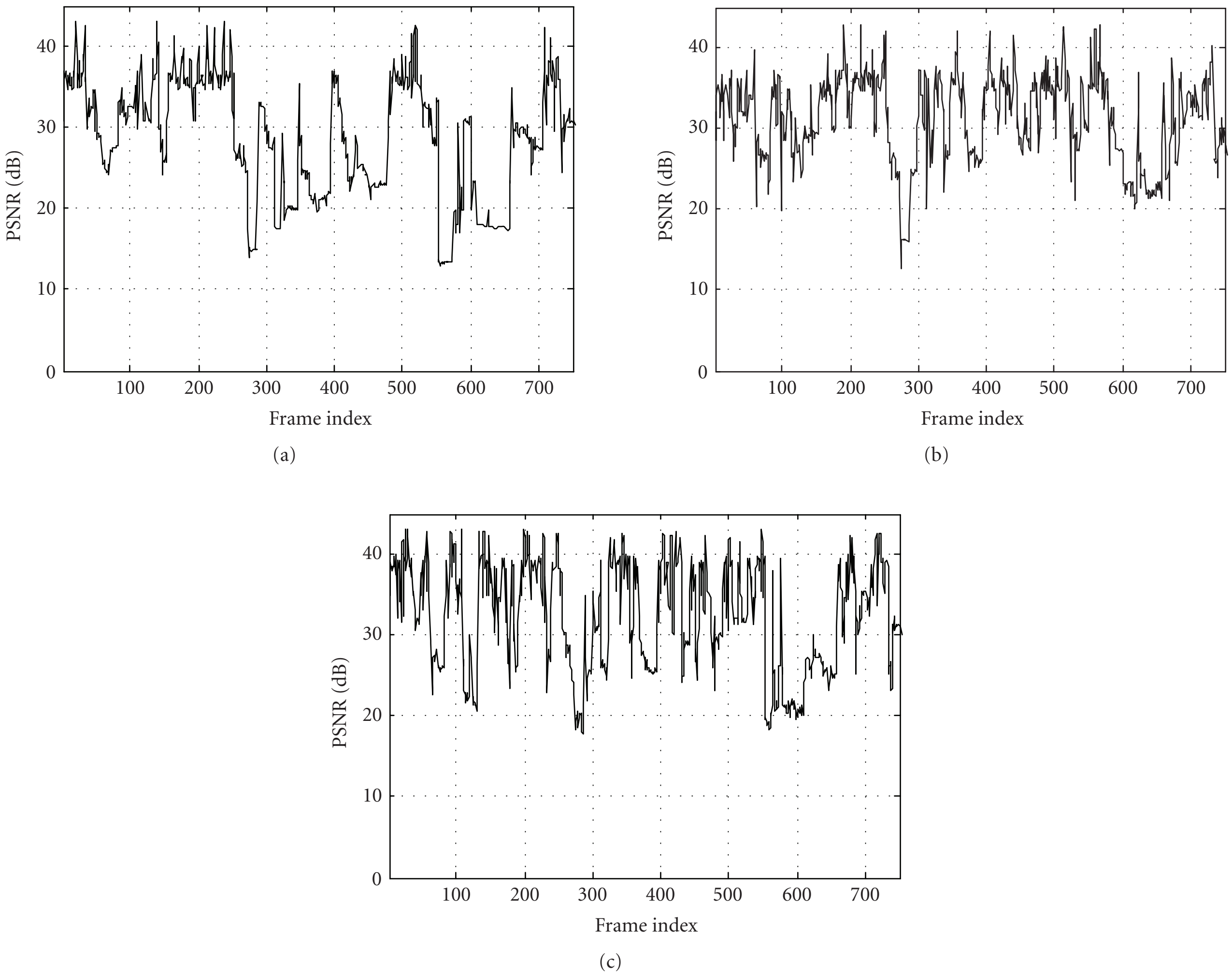
Figure 13: Mean PSNR for three DAB-based schemes with changing buffer size.

is the dominant form of PER). Set BER = p then for m retransmissions in addition to the first attempted transmission, the PER is calculated as

$$\text{PER} = \left(1 - (1 - p)^L\right)^{m+1}. \tag{4}$$

Plotting PER against BER for various retry counts (m), with fixed L appropriate to 5-time-slot Bluetooth packets in 3.0 Mbit/s EDR mode (Section 3), Figure 15 shows in general terms that there are variations in the range of BERs for which adaptive ARQ is appropriate. As the BER becomes low, ARQ becomes less appropriate as all of the plotted retry limits result in very low PERs. For high retry BERs, all retry limits result in impossible PER levels.

Table 4 provides summary statistics (mean of 50 runs) for the different schemes, with three input video sequences: (1) “News” as in previous experiments in this section, (2) “Friends” from the well-known American situational comedy, with more “action” than in “News,” and (3) “Football” with rapid movement. The additional clips had the same GOP structure as the “News” sequence and similarly were CIF-sized at 25 frames/s. A DAB was employed for all the schemes, resulting in similar mean packet delay times for the two priority-based ARQ schemes (adaptive and FLC), according to channel condition. However, it is important to realize that delay is only recorded for delivered packets, whereas the presence of the DAB results in packet discard before delay is recorded. Obviously, delay increases as the wireless channel BERs increase through the application of scaling factor a to the good and bad state BERs of Section 3.4. Across the different video clips, packet loss rates increase and PSNR decreases approximately according to the degree of motion in each video, ranked in order “News,” “Friends,” and “Foot-



(a)

(b)

(c)

Figure 14: PSNR for the test video under (a) infinite ARQ with DAB; (b) adaptive ARQ with DAB; (c) FLC ARQ with DAB.

ball.” Increasing the buffer size results in a negative impact on packet loss rates in respect to the adaptive ARQ scheme, in the sense that the video quality for all three clips deteriorates as the buffer size is increased (owing to the adverse effect on buffer queue waiting times when the ARQ scheme is regulated by a buffer fullness factor). The default Bluetooth ARQ scheme never results in a mean PSNR above 30 dB. Lastly, the FLC with DAB results are emboldened as, in all cases, the received video quality is superior compared to the other schemes, whatever the channel conditions, buffer size, or type of input video clip.

5. CONCLUSIONS

This paper compared various data-link layer schemes for control of ARQ timeouts and found that fuzzy logic control results in a quite considerable improvement in received video quality over a traditional scheme. Though adaptive control of ARQ at the link layer is known in the literature, mainly for other than Bluetooth, the identification of a near optimal scheme is not. Fuzzy logic control can readily be tuned but once the operating parameters are established, no further modifications are required. Though the detailed experiments in this paper are specific to Bluetooth, there is no reason why the same approach should not be applied to other wireless technologies that employ ARQ as a form of error control. Equally, though the scheme was tested with the widely deployed MPEG-2 codec, I and P slices are present in the more recent H.264 and B slices occur in all but H.264’s baseline profile.

Summary results found that in poor channel conditions fuzzy logic control of adaptive ARQ resulted in at least 4 dB improvement in video quality. A secondary finding of the paper was that by the addition of a deadline-aware buffer, delivered packet delay is reduced, though this is only significant if the number of discarded packets through deadline expiration is not high. The delivered video quality of the fuzzy logic controlled scheme is relatively immune to change in send buffer size, whereas adaptive ARQ using buffer fullness to judge the number of retransmissions is buffer-size-dependent, with larger buffer sizes having a negative effect on received video quality. Fuzzy logic control of ARQ in this paper adjust the number of retransmissions in a way that time

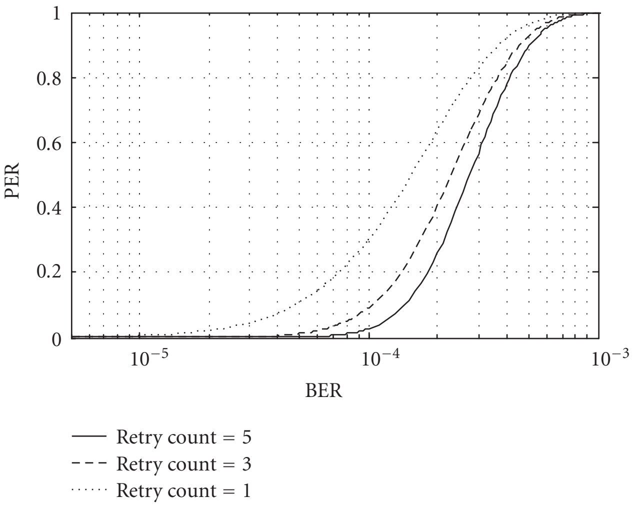


FIGURE 15: PER against BER for differing ARQ retry limits.

delay constraints are honored and buffer overflow is sharply reduced. As fuzzy logic control reduces the number of retransmissions it also reduces power consumption. The possibility of regulating power consumption in mobile devices by an additional power-control factor is open to a fuzzy logic scheme, whereas such an enhancement is less obvious in a tradional ARQ scheme.

ACKNOWLEDGMENT

This work was supported by the EPSRC, UK, under Grant no. EP/C538692/1.

REFERENCES

[1] J. C. Haartsen, "The Bluetooth radio system," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, 2000.

[2] "Specification of the Bluetooth System—2.0 + EDR," 2004, http://www.bluetooth.com.

[3] E. Ferro and F. Potortì, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[4] A. K. Arumugam, S. M. D. Armour, M. F. Tariq, and A. R. Nix, "Proposed evolution technologies for Bluetooth," in *Proceedings of the IEEE 54th Vehicular Technology Conference (VTC '01)*, vol. 4, pp. 2523–2527, Atlantic City, NJ, USA, October 2001.

[5] R. Razavi, M. Fleury, and M. Ghanbari, "Low-delay video control in a personal area network for augmented reality," in *Proceedings of the 4th Visual Information Engineering (VIE '07)*, pp. 1245–1300, London, UK, July 2007.

[6] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "End-to-end QoS for video delivery over wireless Internet," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 123–134, 2005.

[7] M. Kalman, P. Ramanathan, and B. Girod, "Rate-distortion optimized video streaming with multiple deadlines," in *Proceedings of the International Conference on Image Processing (ICIP '03)*, vol. 2, pp. 661–664, Barcelona, Spain, September 2003.

[8] R. Kapoor, M. Cesana, and M. Gerla, "Link layer support for streaming MPEG video over wireless links," in *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN '03)*, pp. 477–482, Dallas, Tex, USA, October 2003.

[9] W. Tan and A. Zakhor, "Packet classification schemes for streaming MPEG video over delay and loss differentiated networks," in *Proceedings of the 11th International Packet Video Workshop*, Kyongju, Korea, May 2001.

[10] L.-J. Chen, R. Kapoor, K. Lee, M. Y. Sanadidi, and M. Gerla, "Audio streaming over Bluetooth: an adaptive ARQ timeout approach," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW '04)*, pp. 196–201, Hachioji, Japan, March 2004.

[11] M. Yokotsuka, "Memory motivates cell-phone growth," *Wireless Systems Design*, vol. 9, no. 3, pp. 27–30, 2004.

[12] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

[13] N. Laoutaris and I. Stavrakakis, "Adaptive playout strategies for packet video receivers with finite buffer capacity," in *Proceedings of the International Conference on Communications (ICC '01)*, vol. 3, pp. 969–973, Helsinki, Finland, June 2001.

[14] Z. Jiang and L. Kleinrock, "A packet selection algorithm for adaptive transmission of smoothed video over a wireless channel," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 494–509, 2000.

[15] J. Wall and J. Y. Khan, "An adaptive ARQ enhancement to support multimedia traffic using 802.11 wireless LANs," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 3037–3041, Dallas, Tex, USA, December 2004.

[16] H. Zhu, M. Li, I. Chlamtac, and B. Prabhakaran, "A survey of quality of service in IEEE 802.11 networks," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 6–14, 2004.

[17] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1752–1763, 2003.

[18] M. Chen and G. Wei, "Multi-stages hybrid ARQ with conditional frame skipping and reference frame selecting scheme for real-time video transport over wireless LAN," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 158–167, 2004.

[19] C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Packet scheduling for video streaming over wireless with content-aware packet retry limit," in *Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing (MMSP '06)*, pp. 409–414, Victoria, BC, Canada, October 2006.

[20] R. Razavi, M. Fleury, and M. Ghanbari, "Adaptive timeout for video delivery over a Bluetooth wireless network," in *Proceedings of the International Conference on Applied Computing (IADIS '07)*, pp. 245–254, Salamanca, Spain, February 2007.

[21] J. Bandara, X. Shen, and Z. Nurmohamed, "A fuzzy resource controller for non-real-time traffic in wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC '00)*, vol. 1, pp. 75–79, New Orleans, La, USA, June 2000.

[22] H. B. Kazemian and L. Meng, "An adaptive control for video transmission over Bluetooth," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 2, pp. 263–274, 2006.

[23] Y. Xiao, P. Chen, and Y. Wang, "Optimal admission control

for multi-class of wireless adaptive multimedia services," *IEICE Transactions on Communications*, vol. E84-B, no. 4, pp. 795–804, 2001.

[24] L. Rossides, C. Chrysostomou, A. Pitsillides, and Y. A. Sekercioglu, "Overview of fuzzy-RED in diffServ networks," in *Proceedings of the 1st International Conference on Computing in an Imperfect World*, vol. 2311 of *Lecture Notes In Computer Science*, pp. 1–13, Belfast, Ireland, April 2002.

[25] C. Luo and C. Ran, "An adaptive retransmission and active drop mechanism based on fuzzy logic," in *Proceedings of Asia-Pacific Radio Science Conference (APRASC '04)*, pp. 162–165, Qingdao, China, August 2004.

[26] N. Golmie, N. Chevrollier, and O. Rebala, "Bluetooth and WLAN coexistence: challenges and solutions," *IEEE Wireless Communications*, vol. 10, no. 6, pp. 22–29, 2003.

[27] M. C. Valenti, M. Robert, and J. H. Reed, "On the throughput of Bluetooth data transmissions," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '02)*, vol. 1, pp. 119–123, Kowloon, China, March 2002.

[28] R. Razavi, M. Fleury, and M. Ghanbari, "Detecting congestion within a Bluetooth piconet: video streaming response," in *London Communications Symposium*, pp. 181–184, London, UK, September 2006.

[29] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[30] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, Institution Electrical Engineers, Stevenage, UK, 2003.

[31] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 1–11, 1996.

[32] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445–455, 1997.

[33] E. N. Gilbert, "Capacity of burst-noise channel," *The Bell System Technical Journal*, vol. 39, no. 8, pp. 1253–1265, 1960.

[34] E. O. Elliott, "Estimates of error rates for codes on burst noise channels," *The Bell System Technical Journal*, vol. 42, no. 9, pp. 1977–1997, 1963.

[35] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, 1999.

[36] L.-J. Chen, T. Sun, and Y.-C. Chen, "Improving Bluetooth EDR data throughput using FEC and interleaving," in *Proceedings of the 2nd International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '06)*, vol. 4325 of *Lecture Notes In Computer Science*, pp. 725–736, Hong Kong, December 2006.

[37] R. Fantacci and M. Scardi, "Performance evaluation of preemptive polling schemes and ARQ techniques for indoor wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 248–257, 1996.

[38] R. Razavi, M. Fleury, E. Jammeh, and M. Ghanbari, "Efficient packetisation scheme for Bluetooth video transmission," *Electronics Letters*, vol. 42, no. 20, pp. 1143–1145, 2006.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 24758, 10 pages
doi:10.1155/2007/24758

Research Article

Utilizing Cross-Layer Information to Improve Performance in JPEG2000 Decoding

Hannes Persson,[1] Anna Brunstrom,[1] and Tony Ottosson[2]

[1] Department of Computer Science, Karlstad University, 651 88 Karlstad, Sweden
[2] Department of Signals and Systems, Chalmers University of Technology, 412 96 Gothenburg, Sweden

Received 1 May 2007; Accepted 28 July 2007

Recommended by Stavros Kotsopoulos

We focus on wireless multimedia communication and investigate how cross-layer information can be used to improve performance at the application layer, using JPEG2000 as an example. The cross-layer information is in the form of soft information from the physical layer. The soft information, which is supplied by a soft decision demodulator, yields reliability measures for the received bits and is fed into two soft input iterative JPEG2000 image decoders. When errors are detected with the error detecting mechanisms in JPEG2000, the decoders utilize the soft information to point out likely transmission errors. Hence, the decoders can correct errors and increase the image quality without making time-consuming retransmissions. We believe that the proposed decoding method utilizing soft information is suitable for a general IP-based network and that it keeps the principles of a layered structure of the protocol stack intact. Further, experimental results with images transmitted over a simulated wireless channel show that a simple decoding algorithm that utilizes soft information can give high gains in image quality compared to the standard hard-decision decoding.

1. INTRODUCTION

Multimedia is predicted to be one of the main applications in future wireless systems. When multimedia is transmitted over a noisy channel, the quality of the multimedia can unfortunately be affected by transmission errors. To mitigate the transmission errors a common technique at the link and the transport layer is to retransmit the erroneous or lost data. This is however not always the best solution, especially when considering delay sensitive applications, because retransmissions introduce delays and result in inefficient use of the channel.

Even though the multimedia user has high demands on the delivered data (e.g., constraints on delay and quality), limitations in human perception allow minor quality degraded multimedia to be forwarded to the user as long as the main information is conveyed in time. Hence, a tradeoff between the transmission delays and the quality of the received multimedia can be made to be able to reach the play out deadline. The tradeoff between delay and quality causes the sender and the receiver to have strategies apart from retransmissions for dealing with transmission errors. These strategies include, for example, error concealment or applying forward error correction (see review in [1]).

The multimedia applications possess knowledge about the structure of the data. In addition, many multimedia codecs that exist today, for example, JPEG2000 and MPEG-4, support features for error resilience [2] allowing a graceful multimedia quality degradation upon errors. Adding lower-layer knowledge about the channel conditions at the application layer can strengthen the error handling capabilities of the application even further. An example of this approach is joint source channel decoding (JSCD), which is an optimized technique to achieve high gains in multimedia quality. A comprehensive survey of JSCD is found in [3] and novel applications are presented, for example, in [4, 5]. However, the JSCD technique is impractical in a layered protocol stack due to the iterative communication between channel and source decoders. A general and simplified decoding technique would ease the deployment in an IP-based wireless network.

In this paper, the considered scenario concerns multimedia traffic transferred from a server in the fixed network to a mobile wireless host (see Figure 1). As an example

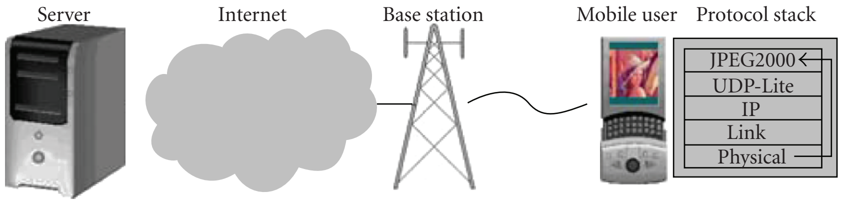


Figure 1: Considered scenario.

application the international standard for still image compression JPEG2000 [6] is used. The target environment concerns wireless channels where multiple bit errors can occur. Images that are sent over these channels should contain some error-resilient information to be able to detect the bit errors. The error resilience is provided by the JPEG2000 standard and is encoded at the sender side. Two soft input iterative JPEG2000 decoders that combine the error resilience and the soft information have been implemented. Experiments performed with these decoders and a simulated wireless channel illustrate that image quality can be improved significantly even with a simple decoding algorithm and without making retransmissions or applying channel coding. Under favorable conditions, high gains in image quality are observed.

We investigate a general and more simplified decoding technique compared to the JSCD technique, where the decoding at the source and channel are independent processes and hence are more suitable for a general IP-based network. More specifically, we investigate how lower-layer knowledge can be used as cross-layer information at the application layer to correct transmission errors. We propose that the knowledge about the channel conditions is in the form of soft information, similar to the information used in JSCD. The soft information yields reliability measures for the received bits and is generated from channel observations in the receiver's physical layer. The soft information gives a structured and channel independent representation of the channel condition. Once generated in the physical layer, the soft information should be transferred to the application through well-defined interfaces. This does not violate the principles of the layered structure of the protocol stack. Without any channel specific knowledge the application can be optimized on the receiver side. Hence, this optimization could be used for different channels. Further, through this approach a persistent and error-resilient source does not have to be transcoded each time a new channel is prevailing.

We consider deployment of soft information in an IP-based wireless network. Although this deployment is thoroughly discussed in [7], we will briefly put forward two important aspects: the necessity to modify the interfaces between the layers to facilitate the propagation of soft information from the physical layer and the use of bit error transparent protocols. Soft information should be forwarded to the application layer through a cross-layer framework. Intermediate layers could also take advantage of soft information and adapt accordingly to the channel. Further, the underlying link and transport protocols must allow erroneous payload data to be forwarded to the application layer. For example, UDP-Lite [8], TCP-L [9], and DCCP [10] are transport protocols that are transparent to bit errors (see overview in [11]). Since the modification proposed in this paper is limited to the receiver, we can assume control of the whole protocol stack in the mobile terminal.

The remainder of this paper is organized as follows. Section 2 defines soft information as used in this work and introduces JPEG2000 and its error-resilient mechanisms. A description of the soft input iterative JPEG2000 decoders is conducted in Section 3 followed by a description of the experimental set-up in Section 4. Section 5 presents the experiment results and Section 6 concludes the paper with some remarks.

2. SOFT INFORMATION AND JPEG2000

This section defines soft information as used in this work. A brief introduction to the example application, JPEG2000, is given, followed by a description of its error-resilient mechanisms. The overview description of JPEG2000 relies heavily on [12, 13].

2.1. Definition of soft information

Soft information can be generated directly from channel observations or when channel decoding is used from a soft output channel decoder. As defined in this paper, the soft information is calculated from channel observations and is based on the Euclidean distance between the received symbol, r, and the modulation constellation symbols, s, in the demodulator. The received symbol, r, denotes a noisy observation at the demodulator, $r = s + n$, where n is the channel noise. Assuming M-ary QAM (quadrature amplitude modulation), the number of bits k for a modulation symbol equals $\log_2 M$. The soft information value for bit b_i ($i = 0, \ldots, k-1$) is described by the log-likelihood ratio (LLR) (as described, e.g., in [14–16]):

$$\text{LLR}(b_i) = \log \frac{P(b_i = 0|r)}{P(b_i = 1|r)} = \log \frac{\sum_{\forall s \in b_i = 0} P(r|s)}{\sum_{\forall s \in b_i = 1} P(r|s)}. \qquad (1)$$

The LLR is calculated for each received bit, b_i, and reveals both the binary value and the reliability of the bit. In (1), $P(b_i = 0|r)$ and $P(b_i = 1|r)$ express the conditional probability for receiving a binary value of 0 and 1 respectively, given the received symbol r. Source bits are assumed to be independent and identically distributed. Further in (1), $P(r|s)$ is

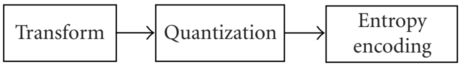


Figure 2: Block diagram of JPEG2000.

the conditional probability, the so-called a posteriori probability (APP), for receiving symbol r, given the transmitted symbol s. The LLR can be interpreted as a quality measure of the estimation that symbol r has been received correctly, given the transmitted symbol s. If an additive white Gaussian noise (AWGN) channel is used, $P(r|s)$ has the form of a Gaussian probability density function [15]. The LLRs for the received bits are assumed to be generated in the physical layer and delivered to the application. It is reasonable to state that the resolution of a soft information value should be x bits per L bit data block [7]. For example, if L equals 1 and x equals 3, the ratio would equal 3 bits of soft information for every data bit. Further, the LLR is transparent to modulation type, thus (1) is applicable when considering systems with arbitrary modulation type.[1] For example, when calculating the LLR for binary phase shift keying (BFSK) and for 64-quadrature amplitude modulation (QAM), (1) will consider the two symbols available for BFSK and the 64 symbols available for 64-QAM, respectively. Since the LLR is transparent to the modulation type in the physical layer, the transparency of the protocol structure is preserved.

2.2. *JPEG2000 compression engine*

The JPEG2000 compression engine consists of three main steps (see Figure 2). Firstly, the encoder applies a wavelet transform on the source image data. Secondly, the transformed data are quantized, and lastly, they are entropy encoded with an arithmetic encoder. The output forms the final code stream preceded by a header containing vital auxiliary image data. Image decoding is conducted by the inverse for each step.

The wavelet transform is a combination of applying a low and a high pass filter vertically and horizontally on the image to be able to explore local frequency characteristics. The wavelet transform produces four subbands for each decomposition level. The wavelet transform is repeatedly applied to the low pass filter output until a desired decomposition level is reached.

To be able to achieve distortion scalability in an image (i.e., an image represented with low quality and hence a low bit-per-pixel value) JPEG2000 involves bit-plane coding of subbands. The subband samples (coefficients) are coded one by one from the most significant bit to the least significant bit. Discarding the least significant bits for a sample will lead to a distorted image due to the information loss. Bits with the same significance form a bit plane. In a bit plane, subband samples contribute only with one bit each. All the bits in a bit plane are coded in only one of a total of three coding passes. Each one of the three coding passes collects contextual information about the bit-plane data.

The spatial restriction of the layered bit planes is the code block dimension. The code block dimension follows the form $2^n \times 2^n$ with a default dimension of 64×64 pixels. The data within a code block are arithmetically encoded and the arithmetic encoding can be performed on an entire code block. However, for error-resilient reasons, the arithmetic encoding is performed on smaller units, that is, on every single coding pass.

2.3. *Error resilience in JPEG2000*

A JPEG2000 code stream starts with a main header. The vital auxiliary data of an image (e.g., size of the image and the number of colors used) are stored in the main header. To be able to start the decoding of an image the main header must be correct. Smaller units called packets store the code block data and each packet has a packet header. The packet header stores auxiliary information about the code block (e.g., number of bytes in each code block and where the code-block is located in the image). To be able to decode the data inside the packet, the packet header must also be protected. This can be achieved by the so-called unequal error protection (UEP) techniques, by first moving the packet headers to the main header and then transferring the main header in a reliable manner over the network. UEP techniques for error robust JPEG2000 header and image data transfer are different kinds of retransmissions strategies, work that is identified in [18], or error correcting codes applied to the packet headers as suggested in [19, 20].

A JPEG2000 decoder, which takes advantage of the error-resilient mechanisms, will not utilize a coding pass that contains a bit error or coding passes in the remaining bit-planes that occur after the bit error. Hence the subband samples will be narrow values. This baseline resilient method of error handling is called error concealment. (The error concealment method is comparable to distortion scalability. The former of them will, however, be conducted in a more uncontrolled fashion.) This behavior can be compared to a baseline decoder with no error resilience, which will use all available data including the samples that are in error. For both cases, quality degradation will take place in the decoded images. However, when applying error concealment, the quality degradation is reduced.

To conceal bit errors, the bit errors must first be detected in the arithmetic decoder. In the JPEG2000 standard, the arithmetic codec has mechanisms to be able to detect bit errors.[2] We use a subset of these mechanisms: (1) restart the arithmetic coder for each coding pass (three coding passes in every bit plane), (2) when restarting the arithmetic coder, use a predictable error-resilient termination policy for every coding pass (consult [13] for an in-depth description) and, (3)

[1] Note that the LLR may not be an optimal representation of the channel. It depends on the modulation whether the LLR provides sufficient statistics or not [17].

[2] The arithmetic decoder has not been tested to see to what extent it lets bit errors pass through undetected.

reset the arithmetic coding context states after each coding pass to decouple the coding passes. Restarting the arithmetic coder and resetting the probability estimation for every coding pass helps to decouple the coding passes. By introducing the termination for each coding pass and explicitly signaling every coding pass length in the packet header, these mechanisms will give some extra data overhead (see [13, pages 509–511]). Resetting the context states also reduces the coding efficiency (see [13, page 503]).

3. UTILIZING SOFT INFORMATION IN JPEG2000

Two algorithms have been developed to process and utilize the soft information. Our goal is to investigate whether soft information is useful at the application layer and whether there could be image quality gains with different algorithms. The following subsections discuss these algorithms in more detail.

3.1. *General idea*

The arithmetically encoded coding passes are the smallest units of coded image data in JPEG2000. If a bit error has occurred in the bit-stream and the arithmetic decoder and the encoder become unsynchronized with each other in the current coding pass, the bit error can be detected if the error-resilient mechanisms discussed above are in use. After a bit error has been detected, the decoder discards (i.e., conceals) the image data in the coding pass that are currently being decoded and thus reduces the effect of bit errors. The baseline resilient decoder will then stop the decoding of the following coding passes and bit planes in the current code block.

By utilizing the soft information and the redundancy added by the arithmetic encoder, it is possible to correct errors. Sometimes the error correction will stop due to limitations in the algorithms, error detection, or in the soft information. However, by salvaging some data from being concealed, the resulting image quality will be improved. Two algorithms have been developed to process the soft information. The main purpose of the algorithms is to evaluate new likely bit-sequences transmitted by the sender.

As described in Section 2, a soft information value is based on the LLR, which reveals the reliability of the bit. From soft information values it is possible to find the most uncertain bits in the received bit-sequence. These bits are in a potential error state when an erroneous bit-sequence is detected. New bit-sequences are formed by swapping the binary values of these bits to their counterparts. When a new bit-sequence has been created, it is evaluated for correctness with the arithmetic decoder, hence making the process iterative. Algorithm 1 outlines the pseudocode for a general iterative decoding algorithm with soft information.

The main difference between the algorithms is how they process the soft information to find new bit-sequences. From a soft information point of view, it is less probable that the new bit-sequences that are generated are correct compared to the originally rejected bit-sequence. Thus, the degree of effectiveness of an algorithm should be based on how well it can find the next most probable bit-sequences. We consider two soft input iterative JPEG2000 decoders implementing an optimal and a heuristic algorithm, respectively.

```
foreach code block
    extract coding passes from code block
    $i = 0$ //code pass counter
    do
        result = arithmetic decode coding pass $i$
        if(result == error)
            if(FIRST_ITERATION)
                $j = 0$ //iteration counter
            else if(MAX_ITERATION)
                conceal remaining coding passes
                break while-loop
            else
                $j = j + 1$
            find new bit-sequence $j$ for coding pass $i$
        else
            $i = i + 1$
    while(result == error || $i <$ MAX_PASSES)
```

Algorithm 1: Pseudocode for a general iterative decoding algorithm.

3.2. *Optimal algorithm*

The soft information values calculated from (1) have the property of being additive. The sum of the adding of the values corresponds to the probability of a bit-sequence being correct. The mathematical background of using a logarithmic scale for the LLR is that it is more convenient to work with summations than multiplications for small values. A calculated sum from adding the soft values of the swapped bits in a bit-sequence represents the accumulated probability of the bit-sequence being correct. This implies that it is possible to develop an algorithm that evaluates the soft information and generates a list of optimally ordered bit-sequences. This optimal algorithm (previously presented in [21]) thus finds the most probable bit-sequences which contain one or several swapped bits.

The implementation of the optimal algorithm builds a binary tree structure considering the m most uncertain bits based on the corresponding soft information values in every n long bit-sequence (where $m \leq n$). When considering the m most uncertain bits, the height of the tree becomes m and is occupied by 2^m leaves. The tree supplies a structured way of finding the m most probable correct bit-sequences. The algorithm recursively traverses the tree, from the top root node down to every leaf node, and calculates the sum of the soft information values for each bit-sequence. Depending on whether a left or a right sub tree is traversed, the algorithm swaps the value of a bit and thus generates new bit-sequences. The implementation of this algorithm is not optimized in terms of efficiency and computational complexity.

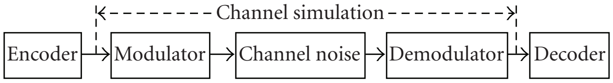


Figure 3: Experiment set-up.

3.3. Heuristic algorithm

We also consider a simple algorithm (previously presented in [22]) that follows a heuristic behavior to evaluate the soft information. On the basis of soft information it picks a small number of bits to swap and generates new probable correct bit-sequences. This algorithm considers a small number of the most uncertain bits and toggles each bit to find a probable correct bit-sequence.

This algorithm first swaps the value of the most uncertain bit and then makes a new decoding pass of the corresponding bit-sequence that previously failed. If the bit-sequence is still in error, the next most uncertain bit is swapped, and an additional decoding pass takes place. This process will continue until a correct bit-sequence is found or until a predefined number of bits have been swapped individually. When a predefined number of bits has been swapped individually the process is extended in such a manner that the decoder permanently swaps the most uncertain bit and then starts to swap the next most uncertain bit.

4. EXPERIMENTAL SET-UP

A number of experiments has been run to evaluate the performance of the algorithms. An overview of the experimental set-up is given in Figure 3. Besides the encoder and the decoder software components, Figure 3 depicts three middle components that are incorporated into the channel simulator software. We assume in our experiments that the underlying layers are able to forward the soft information and accept bit errors in the payload. A description of the experimental set-up of the JPEG2000 codec and the wireless channel follows in this section.

4.1. Image encoder set-up

To evaluate the performance of the algorithms, four image motifs are used in the experiments (Lena, Goldhill, Boat, and Peppers). All the images are JPEG2000 encoded[3] with 1 bit-per-pixel (bpp) including all side information. This setting gives subjectively small image quality degradation compared with the original image. In the experiments, the code block size varies between the values 4×4, 16×16, 32×32 and 64×64. A prerequisite for finding bit errors is that the encoder moves the packet headers to the main header and encodes the error-resilient mechanisms mentioned in Section 2.

4.2. Channel set-up

Simulations of the wireless channel are made by modulating the bits using 16-QAM and transmitting the resulting symbols over an AWGN channel.[4] The signal-to-noise ratio per bit (SNR = E_b/N_0) for the channel ranges from 5 to 16 dB. This ratio measures the relative power of the signal and the noise. An SNR of 5 dB implies a high noise level and 16 dB implies an almost error free channel.

The software for the channel simulates the middle three components shown in Figure 3. The channel simulation is only applied to the image data and no bit errors will thus occur in the vital main header where the packet headers are stored. No retransmissions, channel coding, or interleaving are applied to the image data. The QAM symbols are Gray coded, however. Finally, the hard decided bits and the corresponding soft information[5] are generated in the demodulator module and then later forwarded to the soft input iterative JPEG2000 decoder.

4.3. Image decoder set-up

Two soft input iterative JPEG2000 decoders[6] integrate the optimal and the heuristic algorithm, respectively. As stated in Section 3.2, the implementation of the optimal algorithm is based on a tree structure; hence it is named the tree decoder. The modified decoders assume that the auxiliary image data, that is, the image header, are transmitted in a reliable manner.

The number of iterations, m, in the tree decoder is varied between 5, 10, and 15. These settings will enable the tree decoder to correct up to 2, 3, and 4 bit errors, respectively. The number of attempts before the heuristic decoder stops to generate new bit-sequences from the soft information is set to 10. The number of attempts, before the heuristic decoder permanently swaps the first bit that is most likely to be in error, is set to 7, thereby enabling correction of 2 bit errors. These numbers have an impact on the chances of finding the bit in error. Depending on which code block size is chosen and which channel conditions are present, the parameters are more or less suitable. Large code blocks in combination with a bad channel imply more uncertain bits in the bit-sequence.

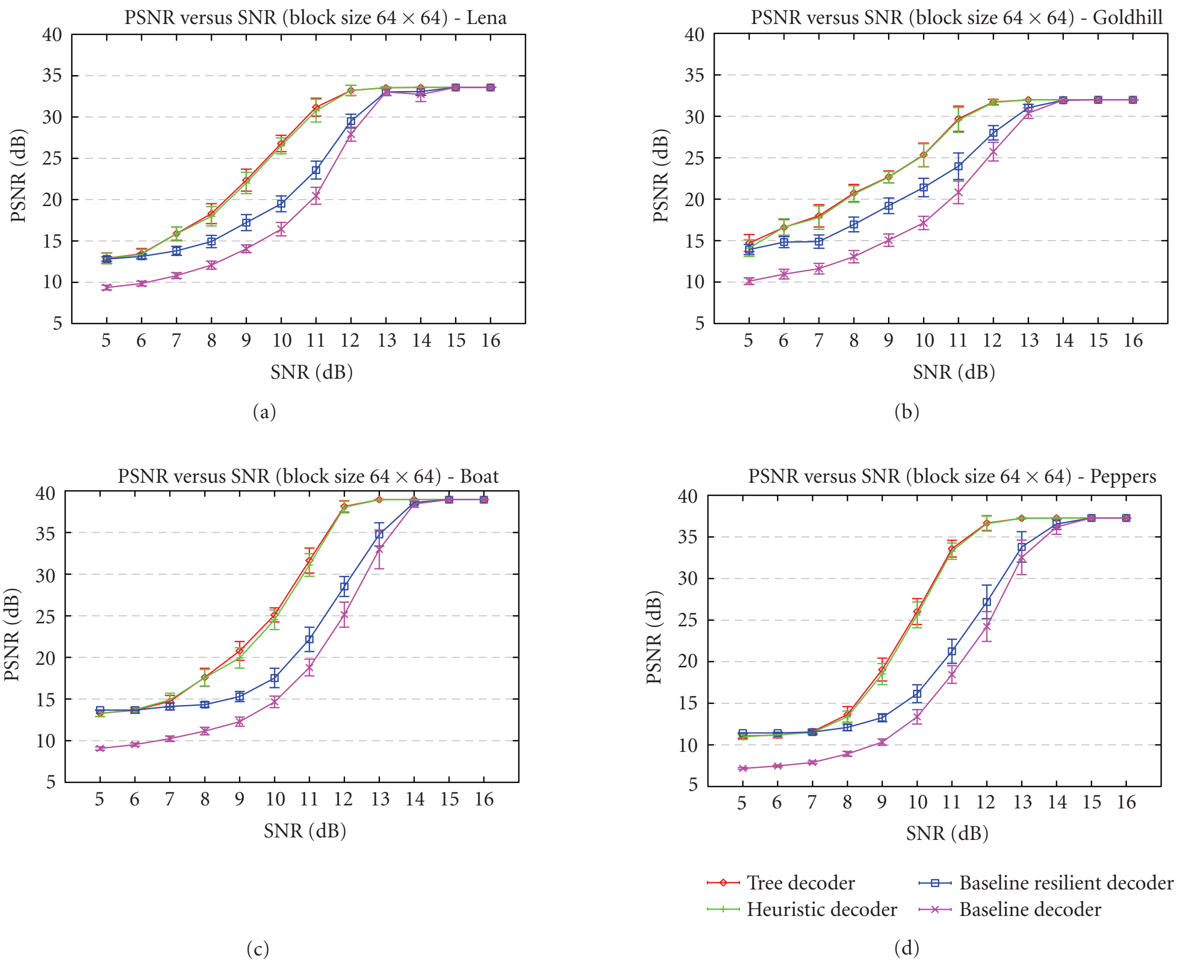
To be able to observe possible gains in image quality with the modified decoders, reference images with two baseline decoders are also decoded. The first baseline decoder is configured to utilize the error-resilient mechanisms in the code-stream and conceal bit errors while the second one is configured not to utilize the error-resilient mechanisms in the code-stream. To compare the quality between the original image and the images decoded by the modified and the baseline decoders, the peak-signal-to-noise-ratio (PSNR) is calculated. To achieve valid statistical results, every image

[3] Encoder software provided by JJ2000 v4.1 (jj2000.epfl.ch).

[4] Modulation software provided by the IT++ library v3.6.6 (itpp.sourceforge.net).

[5] In IT++ soft information values are represented by the data type double, hence 64 bits per data bit.

[6] Based on the Kakadu Software v2.2 (www.kakadusoftware.com).



(a)

(b)

(c)

(d)

Figure 4: Comparison between decoders.

is transmitted over 30 different channels for every channel SNR.

5. EXPERIMENTAL RESULTS

Below follows a discussion of the experimental results. The graphs show image quality, measured with the PSNR metric, on the y-axis versus different choices of channel SNRs on the x-axis. The results are primarily presented for the code block size 64 × 64. The mean values and confidence intervals of 95 percent are used to present the results.

5.1. Tree versus heuristic decoder

The first experiment compares the tree and the heuristic decoder. The maximum number of iterations used is set to 10. The results are depicted in Figure 4 with the results for the two baseline decoders included as reference.

The most important observation made in Figure 4 is that the modified decoders outperform the baseline decoders for a range of channel SNRs. The highest gain observed (in PSNR) between the heuristic and the baseline error-resilient decoder is over 12 dB (see Figure 4 peppers image at SNR 11 dB). Another important observation is that the simple heuristic decoder performs well in comparison with the tree decoder. Only very small differences are observed regardless of the channel SNR. Even if the modified decoders give similar results, we still have to point out that the tree decoder performs somewhat better in general than the heuristic decoder. All the decoders behave in a similar manner, independently of image motif.

Taking a detailed look at the results in Figure 4, we find the following. All the decoders, especially the baseline decoder, perform poorly at SNRs 5-6 dB. The performance of the modified decoders and the baseline error-resilient decoder is very similar. In some cases, depending highly on image motif and image size, the baseline error-resilient decoder even performs better than the modified decoders. The modified decoders have a difficult time generating correct bit-sequences since they only consider a small amount of uncertain bits. Still, the advantage of utilizing soft information starts to show already at low SNRs. Depending on the image, this could occur as early as 6 dB. Observe, however, that for SNRs 5–10 dB the resulting subjective image quality is very poor for all the decoders involved, especially for the baseline decoder. Under these bad channel conditions, there are

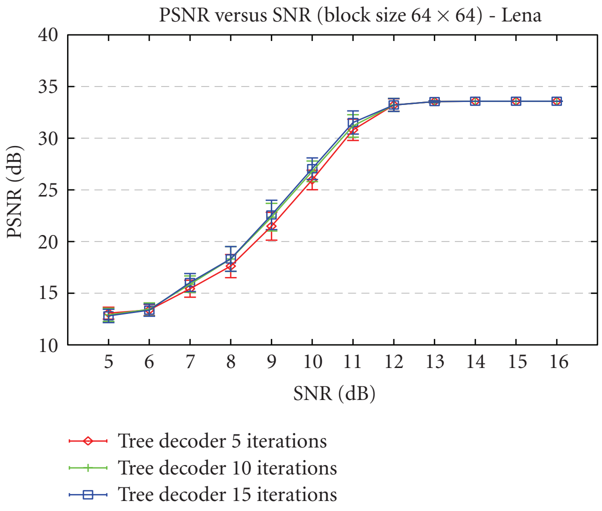


FIGURE 5: Different number of iterations (tree decoder).

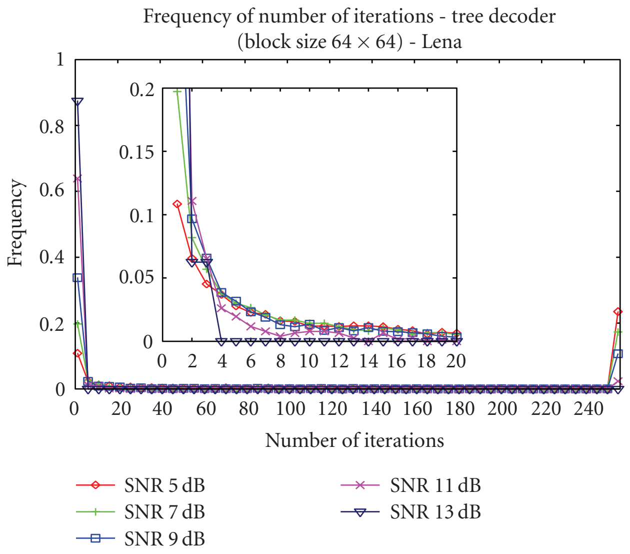


FIGURE 6: Frequency of number of iterations (tree decoder).

too many bit errors and the received images are often useless from a visual perspective. An image with a PSNR value of 30 dB usually gives a good visual experience (see [13]). For SNRs 8–12 dB, the positive effect of utilizing soft information is significant and high image quality gains can be achieved. Here the modified decoders outperform the others. The tree decoder performs somewhat better than the heuristic decoder for SNRs 8–11 dB. However, only gains less than 1 dB in mean values are observed with the tree decoder compared to the heuristic decoder. Furthermore, no statistically significant difference is established between the modified decoders. After SNR 12 dB, there is no obvious quality difference, based both on objective and subjective judgments, between the images decoded from the modified decoders. Hence it is sufficient to use the simpler and less computationally complex heuristic decoder independently of the channel SNR to be able to increase the image quality. For SNR 12 dB and below, it is evident that the baseline decoder performs very poorly compared to the other decoders. Hence taking no resort such as error concealment will lead to a very low image quality. At 12 dB, it is possible to decode an error free image by utilizing soft information. First at 14 dB it is possible to receive an almost error free image independently of decoder choice. No significant difference is present between the decoders at 15-16 dB due to an error free transmission.

Sample images with the Boat motif for SNR 11 dB are depicted in Figures 9(b), 9(d)–9(f). The image quality is very satisfying for the images decoded with the modified decoders while the baseline decoders perform poorly. High gains are also reached with the modified decoders when we investigate the resulting PSNR values.

5.2. *Impact of number of iterations*

Three additional experiments are conducted to investigate the impact of the number of iterations. First, we examine three different maximum numbers of iterations used by the tree decoder (5, 10, and 15 iterations). The results for the Lena image are depicted in Figure 5. The overall observation is that more iterations will only lead to minor image quality gains. The experimental results show that going from 5 to 10 iterations seems to give higher gains than going from 10 to 15 iterations. Hence a limited number of iterations appear to obtain most of the image quality gain. Similar to the results presented in the previous subsection all the different numbers of iterations perform poorly at SNRs 5-6 dB. For SNRs 7–11 dB the highest gain observed is with 15 iterations. However, 10 iterations are sufficient to obtain most of the gain. The image quality gain that 15 iterations introduce is only marginal compared to 10 iterations (less than 1 dB in mean values). The highest PSNR gain observed for 15 iterations over 5 iterations is approximately 2 dB in mean values. When the channel gets better (12–14 dB) no or only minor gains are observed with 15 iterations and for higher SNRs no difference between the different number of iterations is observed due to an error free channel. No statistically significant difference is established between the different numbers of iterations and from a computational overhead perspective it is not motivated to use too many iterations with the tree decoder. Figures 9(a)–9(c) depict sample images with the Boat motif at an SNR of 11 dB. The images show very good quality and it is difficult to detect visual differences between the different numbers of iterations.

To further investigate the impact of number of iterations the tree decoder is allowed to use a maximum of 255 iterations.[7] Each time the algorithm is invoked we record the number of iterations that is needed to correct the bit-sequence. Figure 6 depicts, for different channel SNRs, the

[7] The use of 255 iterations does not necessarily imply an optimal algorithm due to the current implementation of the tree decoder. Only a maximum of 16 bits is under consideration in this decoder.

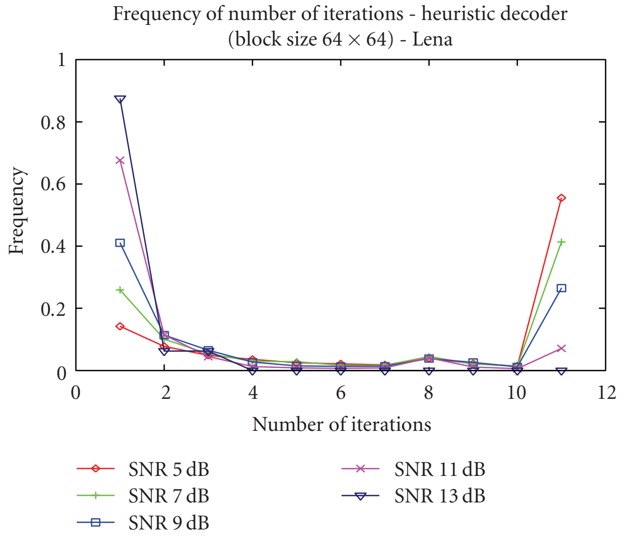


Figure 7: Frequency of number of iterations (heuristic decoder).

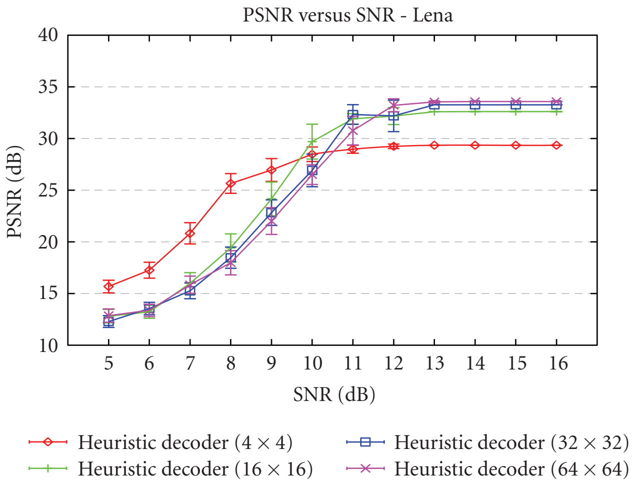


Figure 8: Implications of different code blocks (heuristic decoder).

frequency of the different number of iterations used to correct a bit-sequence relative to the total number of algorithm invocations.[8] It can be concluded that the decoder typically needs very few iterations to correct an erroneous bit-sequence, especially for high channel SNRs. Although not displayed by the graph, no or very low quality gains are observed when 255 iterations are allowed, compared to the results presented above. Allowing up to 10 iterations for SNR 11 dB will resolve 90 percent of the detected errors. This supports our observation above that a limited number of iterations is sufficient to obtain most of the image quality gain. Detailed analysis of the tree decoder's output also reveals that the error-resilient mechanisms in JPEG2000 are not able to detect all bit errors. This can unfortunately result in successive logical errors in the arithmetic decoder's internal states, errors that cannot be corrected with the two proposed algorithms. Thus, the conclusion made from this experiment is that a low number of iterations is sufficient to be able to increase image quality given the robustness the present error-resilient mechanisms can offer.

The last experiment investigates the number of iterations for the heuristic decoder. In this experiment, we again log the number of iterations needed to correct an error. Figure 7 depicts, for different channel SNRs, the frequency of the different number of iterations relative to the total number of algorithm invocations.[9] It can be concluded that the decoder needs only few iterations to resolve an error for SNRs 11 and 13 dB. Similar to the previous experiment for SNR 11 dB, the heuristic decoder resolves 90 percent of the detected errors. For low SNRs the number of iterations needed to correct an error increases and the heuristic decoder is not able to correct many errors.

5.3. *Impact of code block size*

The code block size plays an important role in the combat against errors and has a great impact on how efficiently the proposed algorithms can find new error free bit-sequences. The results in Sections 5.1 and 5.2 are only presented for 64 × 64 large code blocks; thus a discussion of how the code block size affects the outcome of the image quality is conducted below. A discussion of the impact of the code block size for the heuristic decoder is also given in [21]. Figure 8 depicts the influence that different code block sizes have on the image quality for the Lena motif when decoding is done by the heuristic decoder.

Generally when using small code block sizes at low SNRs, we are able to receive images with higher quality compared to larger code blocks because of the bit error enclosing feature of code blocks. The benefit of using small code blocks is clear for all the evaluated error-resilient decoders because they all have less data to process and concealing a bit error does not lead to a large data loss. The pursuit of an error free bit-sequence for the modified decoders is simplified because of the smaller data quantities. For the heuristic decoder, this conclusion is also clear from Figure 8 at an SNR of 5–9 dB. When increasing the code block size at low SNRs all the decoders perform poorly because bit errors in larger code blocks result in more data that are unusable. It will also be harder to pin point potential bit errors with the heuristic and the tree decoders. This is due to the fact that the modified decoders always consider a limited number of uncertain bits and use only a limited number of iterations. When the channel conditions are better (i.e., ≥ 11 dB) it is better to use large code blocks. Compared to small code blocks, large code blocks result in less auxiliary data and, for a fixed image file size, more pure image data are available as a whole. The great overhead of auxiliary information for small code blocks in the file structure reduces the image quality substantially. Thus in an almost error free environment we are better off

[8] A value of 256 for the number of iterations indicates that the error could not be corrected by the algorithm.

[9] The number 11 on the x-axis in Figure 7 corresponds to the case where the decoder is unable to correct the error.

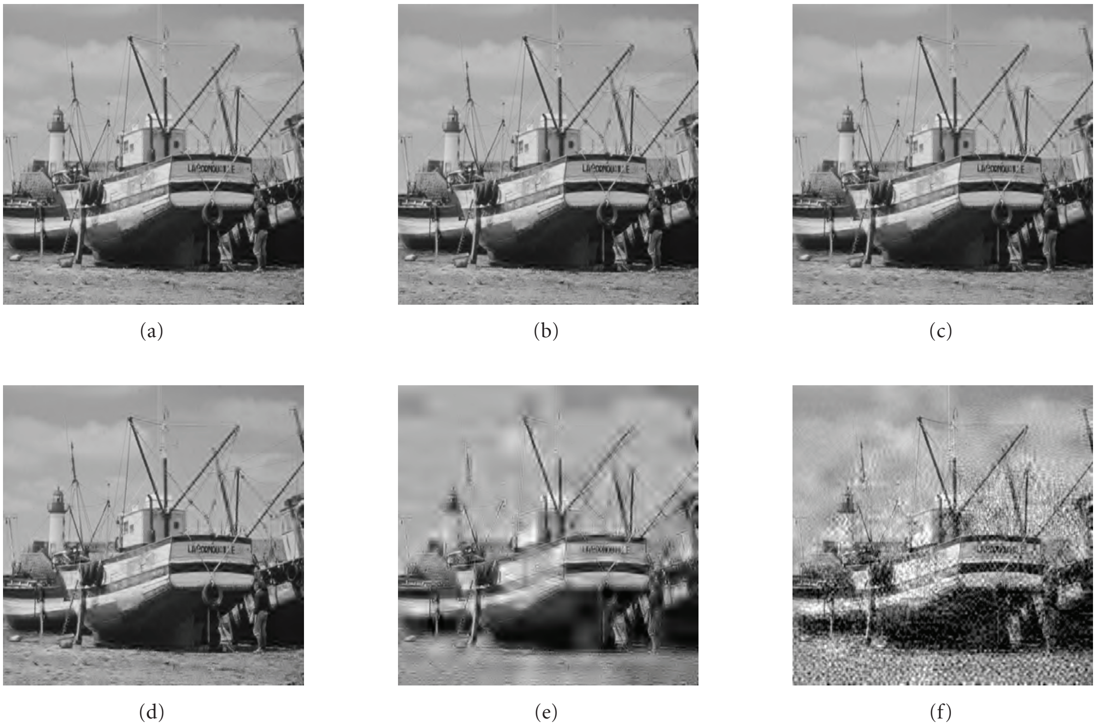


Figure 9: Sample images at SNR 11 dB with code block size 64 × 64 (Boat motif). (a) Tree decoder 15 iterations PSNR 31.4461 dB, (b) tree decoder 10 iterations PSNR 31.4318 dB, (c) tree decoder 5 iterations PSNR 30.3724 dB, (d) heuristic decoder PSNR 30.6885 dB, (e) baseline error-resilient decoder PSNR 23.301 dB, (f) baseline decoder PSNR 20.0612 dB.

using larger code blocks. This observation is again evident for the heuristic decoder from Figure 8, at SNRs between 12 and 16 dB. It is also possible to receive images with large code blocks that contain errors but have higher PSNR values than error free images with small code blocks (see Figure 8 for SNRs 11-12 dB).

6. CONCLUDING REMARKS

The utilization of cross-layer information in the form of soft information from the physical layer is considered in this paper. We have implemented two soft input iterative JPEG2000 image decoders that take advantage of the soft information to improve image quality when combating channel bit errors in a wireless environment. We can state that decoding with soft information performs better than with hard decided bits in terms of gains in image quality and a more versatile decoding process. The gain in channel SNR could be as high as 2 dB. Further, the results of our experiments presented here indicate that a decoder with simple heuristic rules and a limited number of iterations performs well. Both high PSNRs and visual image quality gains are present, especially when the channel SNR lies between 8 and 12 dB. Only marginal additional gains are made with a more complex decoder and when more iterations are allowed.

With our experimental set-up, an acceptable subjective image quality is achieved with soft information approximately at channel SNR 11 dB. When the channel SNR is between 5 and 10 dB, the resulting subjective image quality is not acceptable; thus soft information will not help in the decoding process. When the channel SNR is very high, no image quality gains are observed with soft information due to an error free transmission. Different JPEG2000 code block sizes also have an evident impact on image quality in the occurrence of bit errors and in the context of soft information. At channel SNR 11 dB and above the recommendation is to use larger code blocks.

Comparing a JSCD system to ours, our work differs in the following ways. Firstly, we do not have any demand for applying channel coding in the physical or link layer on the data that soft information is applied on, although it can be included at the sender. Secondly, our work does not involve an iterative process between the channel decoder and the source decoder. This iterative process demands that information can be exchanged between the decoders thus between network protocols. In our proposed decoder, there is an iterative process, but it is solely restricted to the internals of the source decoder. Thirdly, our technique is simplified and not an optimal technique for improving image quality. Work done in, for example, [5] is a novel example of a JSCD system which gives higher gains in image quality compared to our approach but at the expense of higher complexity.

Utilizing soft information from the physical layer in the application layer requires modifications to the intermediate communication layers. The modifications involve firstly the propagation of soft information from the physical layer to the application layer, and secondly, intermediate layers must allow erroneous payload data due to the application's enhanced error resilience. Potential solutions for these demands can be found in the literature (e.g., [23–27] for propagation of soft information and [8–10, 28] for allowing erroneous payload data). In the future, we wish to investigate possible performance improvements with soft information in a system-wide perspective. Combining the results presented here, for example, with a bit error transparent protocol such as UDP-Lite [8] or TCP-L [9] will make it possible to achieve numerical results about the system performance.

ACKNOWLEDGMENTS

This work was supported in part by the Swedish Governmental Agency for Innovation Systems, VINNOVA, and the Swedish Foundation for Strategic Research.

REFERENCES

[1] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[2] I. Moccagatta, S. Soudagar, J. Liang, and H. Chen, "Error-resilient coding in JPEG-2000 and MPEG-4," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 899–914, 2000.

[3] C. Guillemot and P. Siohan, "Joint source-channel decoding of variable-length codes with soft information: a survey," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 6, pp. 906–927, 2005.

[4] C. Guillemot and P. Christ, "Joint source-channel coding as an element of a QoS framework for '4G' wireless multimedia," *Computer Communications*, vol. 27, no. 8, pp. 762–779, 2004.

[5] W. Xiang, S. S. Pietrobon, and S. A. Barbulescu, "Iterative source-channel decoding for robust image transmission," in *Proceedings 4th Australian Communications Theory Workshop (AusCTW '03)*, pp. 61–65, Melbourne, Australia, February 2003.

[6] "JPEG2000 Part 1—JPEG2000 Final Committee Draft Version 1.0, ISO/IEC 15444-1," March 2000.

[7] A. Brunstrom and T. Ottosson, "The introduction of soft information in IP-based wireless networks," in *Proceedings of Nordic Radio Symposium (NRS '01)*, Nynäshamn, Sweden, April 2001.

[8] L.-Å. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "The lightweight user datagram protocol (UDPLite)," Network Working Group, RFC 3828, July 2004.

[9] S. Alfredsson and A. Brunstrom, "TCP-L: allowing bit errors in wireless TCP," in *Proceedings of IST Mobile and Wireless Communications Summit*, Aveiro, Portugal, July 2003.

[10] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," Internet Engineering Task Force, Internet draft expires, March 2006.

[11] M. Welzl, "Passing corrupt data across network layers: an overview of recent developments and issues," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 2, pp. 242–247, 2005.

[12] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, 2001.

[13] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[14] C. Berrou, "The ten-year-old turbo codes are entering into service," *IEEE Communications Magazine*, vol. 41, no. 8, pp. 110–116, 2003.

[15] S. ten Brink, J. Speidel, and R.-H. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '98)*, vol. 1, pp. 579–584, Sydney, Australia, November 1998.

[16] J. Hagenauer, "The turbo principle: tutorial introduction and state of the art," in *Proceedings of the Symposium on Turbo-Codes*, pp. 1–11, Brest, France, September 1997.

[17] M. Skoglund and T. Ottosson, "Soft multiuser decoding for vector quantization over a CDMA channel," *IEEE Transactions on Communications*, vol. 46, no. 3, pp. 327–337, 1998.

[18] M. Grangetto, E. Magli, and G. Olmo, "Error sensitivity data structures and retransmission strategies for robust JPEG 2000 wireless imaging," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 872–882, 2003.

[19] A. Natu and D. Taubman, "Unequal protection of JPEG2000 code-streams in wireless channels," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 534–538, Taipei, Taiwan, November 2002.

[20] D. Nicholson, C. Lamy-Bergot, X. Naturel, and C. Poulliat, "JPEG 2000 backward compatible error protection with Reed-Solomon codes," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 855–860, 2003.

[21] H. Persson, A. Brunstrom, and T. Ottosson, "Utilizing soft information at the application layer: quality enhanced JPEG2000 decoding," in *Proceedings of Radio Vetenskap och Kommunikation (RVK '05)*, Linköping, Sweden, June 2005.

[22] H. Persson, A. Brunstrom, and T. Ottosson, "Utilizing soft information in image decoding," in *Proceedings of the 14th IEEE Personal, Indoor and Mobile Radio Communications (PIMRC '03)*, vol. 3, pp. 2678–2682, Beijing, China, September 2003.

[23] S. Mérigeault and C. Lamy, "Concepts for exchanging extra information between protocol layers transparently for the standard protocol stack," in *Proceedings of the 10th International Conference on Telecommunications (ICT '03)*, vol. 2, pp. 981–985, Tahiti, Papeete, French Polynesia, February-March 2003.

[24] L.-Å. Larzon, U. Bodin, and O. Schelén, "Hints and notifications," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '02)*, vol. 2, pp. 635–641, Orlando, Fla, USA, March 2002.

[25] H. Zheng and J. Boyce, "An improved UDP protocol for video transmission over Internet-to-wireless networks," *IEEE Transactions on Multimedia*, vol. 3, no. 3, pp. 356–365, 2001.

[26] C. Lamy-Bergot and P. Vila, "Multiplex header compression for transparent cross-layer design," in *Proceedings of the IEEE International Conference on Networking (ICN '04)*, pp. 1084–1089, Guadeloupe, French Carribean, March 2004.

[27] Q. Wang and M. A. Abu-Rgheff, "Cross-layer signalling for next-generation wireless systems," in *Proceedings of IEEE Wireless Communications and Networking (WCNC '03)*, vol. 2, pp. 1084–1089, New Orleans, La, USA, March 2003.

[28] G. Fairhurst and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)," Network Working Group, RFC 3366, August 2002.

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 $\hat{D}_{\mathrm{S+C}}$ after the source and channel

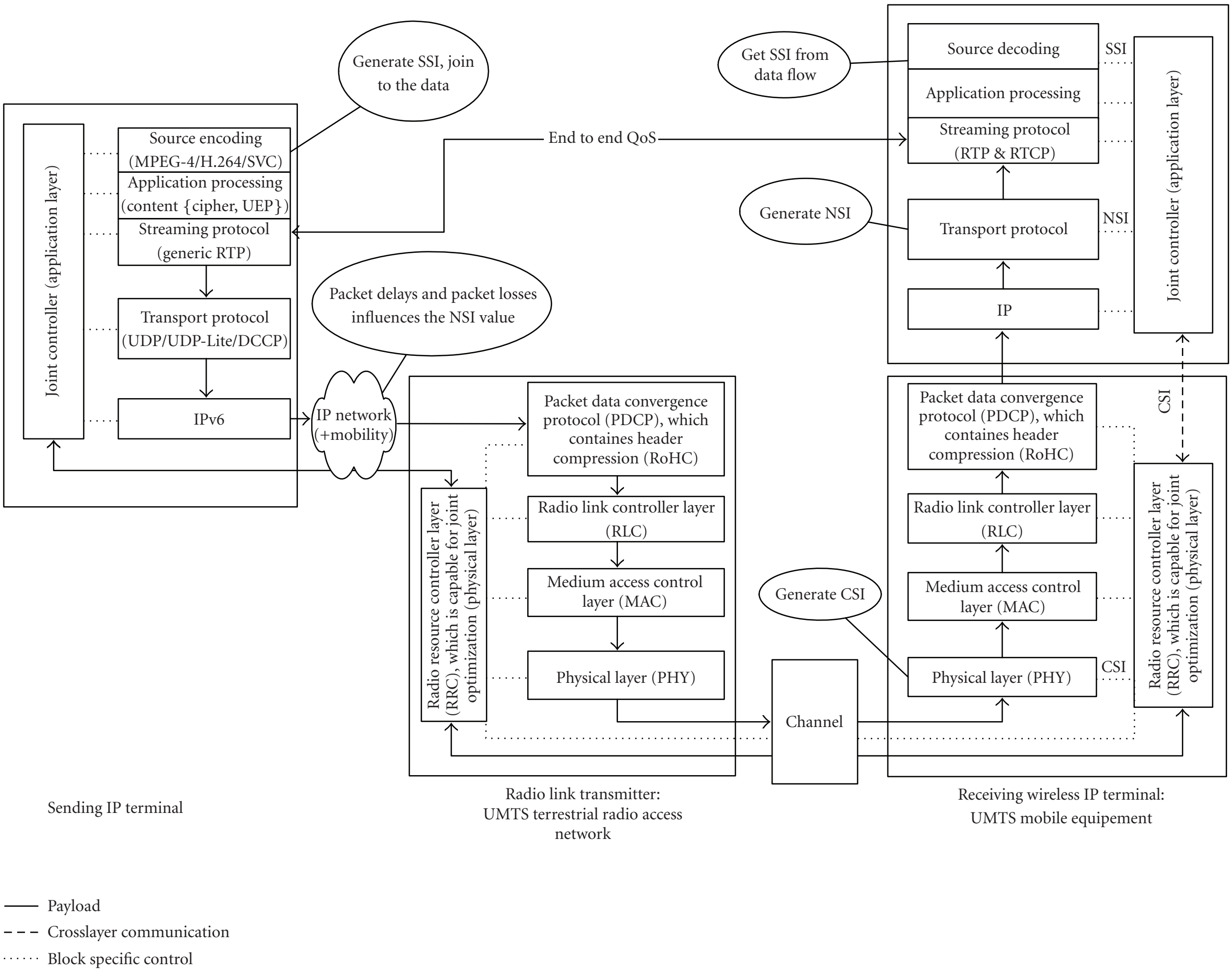

Generate SSI, join to the data
Joint controller (application layer)
Source encoding (MPEG-4/H.264/SVC)
Application processing (content {cipher, UEP})
Streaming protocol (generic RTP)
Transport protocol (UDP/UDP-Lite/DCCP)
IPv6
End to end QoS
Packet delays and packet losses influences the NSI value
IP network (+mobility)
Packet data convergence protocol (PDCP), which containes header compression (RoHC)
Radio link controller layer (RLC)
Medium access control layer (MAC)
Physical layer (PHY)
Radio resource controller layer (RRC), which is capable for joint optimization (physical layer)
Channel
Get SSI from data flow
Source decoding
Application processing
Streaming protocol (RTP & RTCP)
Generate NSI
Transport protocol
IP
SSI
NSI
CSI
Joint controller (application layer)
Packet data convergence protocol (PDCP), which containes header compression (RoHC)
Radio link controller layer (RLC)
Generate CSI
Medium access control layer (MAC)
Physical layer (PHY)
CSI
Radio resource controller layer (RRC), which is capable for joint optimization (physical layer)
Sending IP terminal
Radio link transmitter: UMTS terrestrial radio access network
Receiving wireless IP terminal: UMTS mobile equipement
Payload
Crosslayer communication
Block specific control

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 $\widehat{D}_{\text{S+C}}$ 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:

$$\widehat{D}_{\text{S+C}} = \sum_{i\in N} D_i \cdot P_i. \qquad (1)$$

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

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

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

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

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

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

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

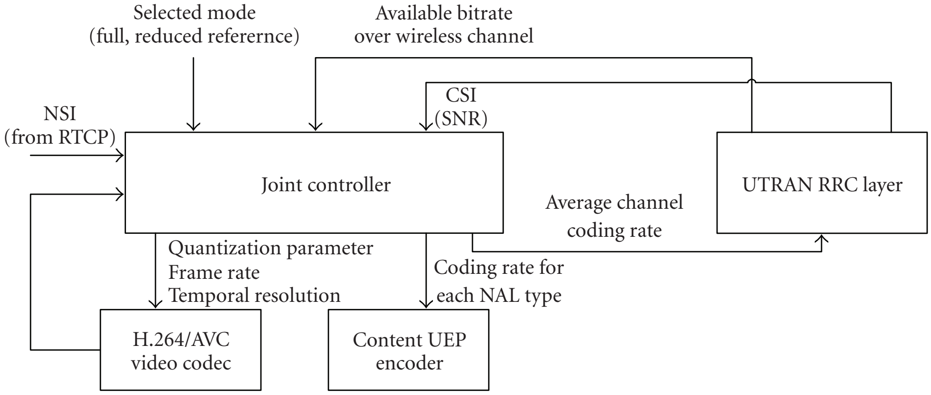


Figure 2: Joint controller inputs and outputs.

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.

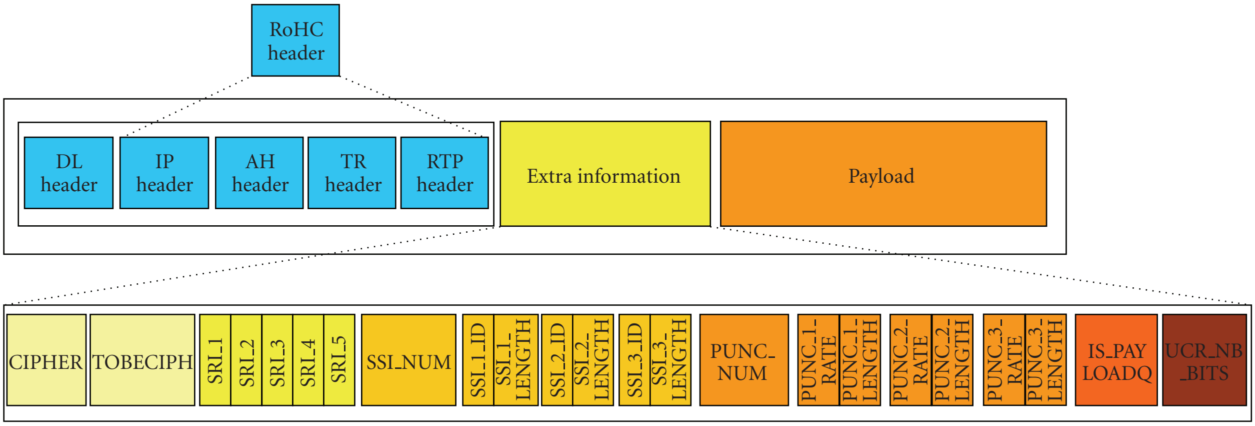


Figure 3: Binary data stream structure and extra information definition.

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:

$$y_i = x_i + e_i, \tag{5}$$

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

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

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.

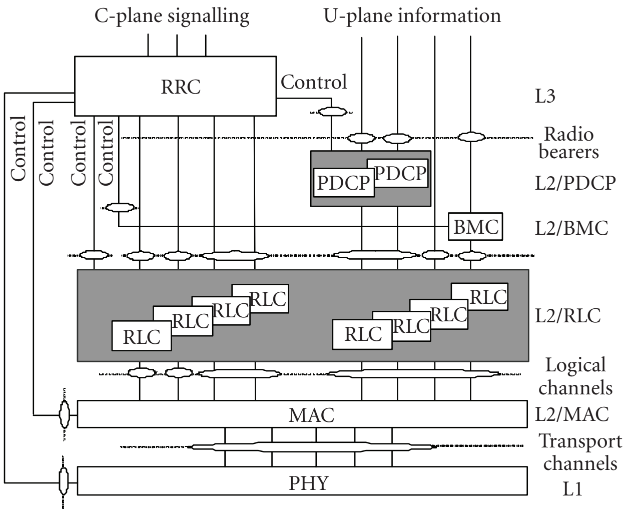


Figure 4: UTRAN protocol architecture.

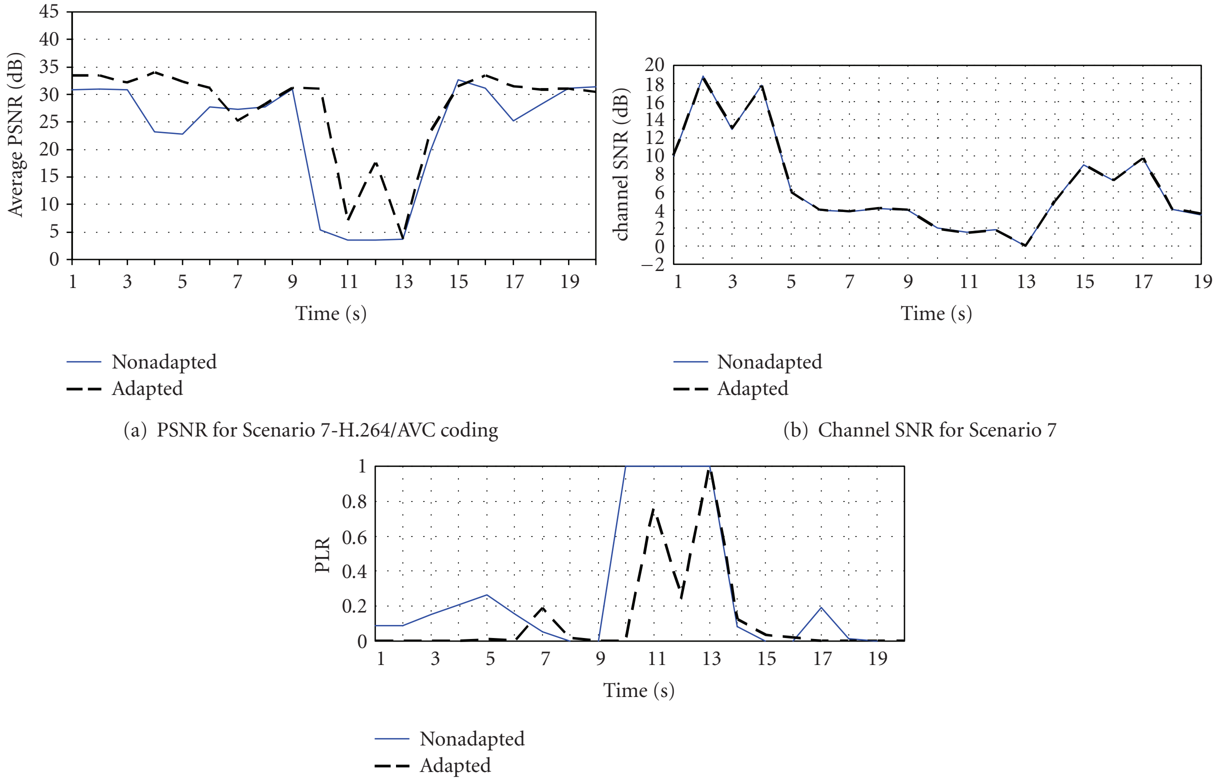
Futhermore, a mobility model is adopted at IP layer. The results show that the higher the handover frequency is, the higher the end-to-end packet loss and the packet loss rate (PLR) becomes. Its impact can be summarized into two aspects. There will be an increase in the packet loss rate, because during 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 decompressing 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.

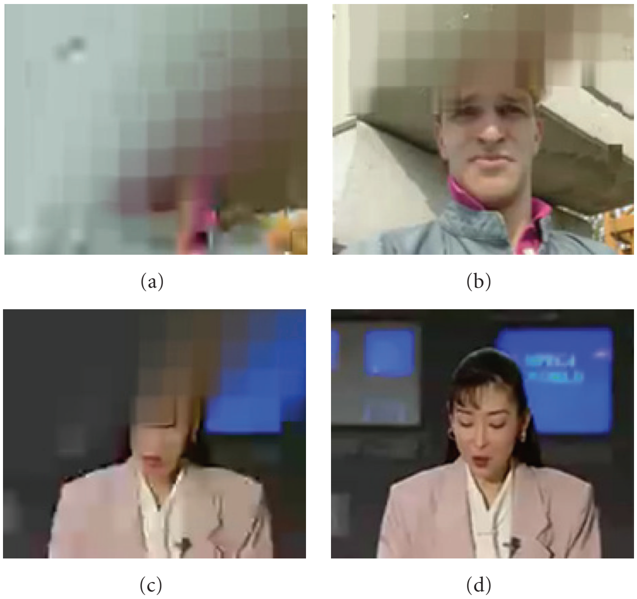


Figure 6: Example of visual results obtained in simulation: non-adapted (left) versus adapted (right).

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 ($N_{\mathrm{sym}} = 4800$)

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

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

The β parameter is expressed as

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

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

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

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 $r_{\mathrm{p}} - r_{\mathrm{C}}$ 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

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

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.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 25187, 9 pages
doi:10.1155/2007/25187

Research Article

Energy-Constrained Quality Optimization for Secure Image Transmission in Wireless Sensor Networks

Wei Wang,[1] Dongming Peng,[1] Honggang Wang,[1] Hamid Sharif,[1] and Hsiao-Hwa Chen[2]

[1] *Faculty of Computer and Electronics Engineering, University of Nebraska-Lincoln, Omaha, NE 68182, USA*
[2] *Institute of Communication Engineering, National Sun Yat-Sen University, 70 Lien-hai Rd., Kaohsiung 804, Taiwan*

Received 1 May 2007; Accepted 22 August 2007

Recommended by Tasos Dagiuklas

Resource allocation for multimedia selective encryption and energy efficient transmission has not been fully investigated in literature for wireless sensor networks (WSNs). In this article, we propose a new cross-layer approach to optimize selectively encrypted image transmission quality in WSNs with strict energy constraint. A new selective image encryption approach favorable for unequal error protection (UEP) is proposed, which reduces encryption overhead considerably by controlling the structure of image bitstreams. Also, a novel cross-layer UEP scheme based on cipher-plain-text diversity is studied. In this UEP scheme, resources are unequally and optimally allocated in the encrypted bitstream structure, including data position information and magnitude value information. Simulation studies demonstrate that the proposed approach can simultaneously achieve improved image quality and assured energy efficiency with secure transmissions over WSNs.

1. INTRODUCTION

The constraints in computation, memory, and energy resources are well-known challenges to sensor network designs especially for multimedia transmission. Although recently there has been a rapid increase of interest on wireless multimedia sensor networks (WMSN), many realistic difficulties are in the way of such advancement, including but not limited to data authentication, secret data protection, image transmission quality, and energy efficiency.

Image encryption and decryption are particularly time-consuming [1], hence justifying a compromise solution where image data are only selectively encrypted to reduce the total computational complexity. This concept proves to be effective and efficient considering that the complex interdependence structure among image compression bit streams [1] can be completely hidden by partial encryption. However, there have been very few papers in literature addressing the correlations among such partial encryption methods and the associated wireless transmission approaches, not to mention the corresponding efforts to meet the resource constraints for sensor networks.

Selective encryption can be effectively performed on the positions of image pixels other than the various values of these pixels conveyed in the natural digital image. Besides layered unequal importance [2], wavelet image compressions such as zerotree-based EZW [3], SPIHT [4], or EBCOT [5] based JPEG2000 produce position information of the objects and the magnitude information of objects. Packet losses of position information destroy the bitstream structure which is crucial for decoding; the bitstream structure is not changed if packet losses of magnitude value information occur. This is called position-value (P-V) diversity inborn with wavelet-based image compression, which provides remarkable potentials for designing multimedia selective encryption algorithms.

This paper proposes a cross-layer approach to deliver selectively encrypted images for minimal distortion with strict energy budget constraints. We first develop a simple but effective position-based selective encryption scheme to reduce encryption overhead by tightly controlling the bitstream structure. Cross-layer optimized UEP strategies are then exploited to allocate the resources among selective encrypted structure information, position information, and magnitude information. Overall, minimized distortion in image transmissions is achieved and the goal of energy efficiency is met.

In recent literature on image selective encryption, most of the popular approaches focus on selecting the important DCT or wavelet coefficients. Research in [6] proposes an effective frequency domain significant coefficients scrambling scheme. To achieve authorized user access control for digital video streaming, a compressed domain scrambling is proposed in [7]. Unfortunately, frequency scrambling techniques usually randomize energy distribution in coefficient matrixes, which sacrifice multimedia compression efficiency according to research in [8]. Similar researches working in frequency domain are found in [9, 10]. Other researches explore selective encryption at entropy coding stage, where the compression performance is not negatively affected. Multiple entropy coding table scheme is proposed in [8] to achieve high-level security. In this approach, entropy coding table is pseudorandomly selected according to the given key. Tree-based selective encryption is proposed in [11], in which the tree structure information is ciphered; without the tree structure information, leaf and children nodes will be put to wrong position, and thus it is impossible to decode the whole image. However, all of these aforementioned works focus on application layer and have not considered the delivery of encrypted images in time-varying wireless channels.

In recent literature on UEP studies for multimedia delivery over wireless networks, most of them focus on rate-distortion or delay-distortion oriented optimization, where different protection levels are applied to different media stream layers. The security factor is largely overlooked. Because selective encryption controls the skeleton of the streaming media and redistributes more importance on cipher-text, traditional optimized UEP schemes are no longer optimal when selective encryption is taken into account. How to transmit image efficiently over WSNs through exploring interdependency and unequal importance nature among selectively encrypted blocks, position information, and value information, has not been extensively discussed in literature.

Wu et al. in [12] proposes an optimized joint source channel coding (JSCC) scheme to achieve minimized total distortion for multiple images over lossy channels simultaneously. The layer-based dependency as well as distortion reduction expectation is well modeled, and combined total distortion is minimized subject to total rate constraint. Hamzaoui et al. survey recent advances in forward error correction (FEC) based scalable image coder in [13], and proposes a local-search-based rate-distortion optimization solution. Li et al. in research [14] develop a real-time link layer retry limit adaptation algorithm for robust video streaming over 802.11-based wireless networks. Multiple video layers are unequally protected by different link layer retry limits. van der Schaar and Turaga in [15] propose cross-layer optimized packtization and retransmission strategies for delay sensitive video delivery over WLANs. The cross-layer optimization problem is formulated as distortion minimization given delay constraints, and significant multimedia quality gain is reported by packtization and retransmission optimization. The aforementioned works are mainly delay-distortion or rate-distortion optimization algorithms suitable for general wireless networks; it is hard to be directly used in WSNs due to the limited energy rather than bandwidth resource in WSNs. One of our preliminary works proposed in [16] shows the energy-distortion gain by considering multimedia selective encryption in resource allocation.

Selective encryption scrambles the intersegment correlation in the final bitstream, leading to significant potential for encryption-oriented cross-layer optimization. In this paper we systematically formulate energy efficient secure image transmission problem, which is significant different from previous layer-based UEP schemes in literature. The paper is organized as follows. In Section 2, position-based selective encryption is proposed. In Section 3, security aware distortion reduction optimization is proposed with energy constraint. In Section 4, frame-level energy consumption and frame-loss ratio are modeled in details for multirate WSNs. Section 5 shows simulation results. The conclusion is drawn in Section 6. Major symbols in equations and notations are defined in Table 1.

2. SELECTIVE ENCRYPTION OF IMAGE DATA

Positions of significant wavelet coefficients are much more important than the magnitudes of those coefficients. Furthermore, the positions of significant coefficients are determined by the clustering model of insignificant coefficients, which is translated into bitstream structure after compression. To effectively cipher the bitstream structure in the proposed approach, position information is packed into p-segments and magnitude information is packed into v-segments bit-plane by bit-plane. This p-segment and v-segment packing processes in each bit-plane are described as follows. In each embedded bit-plane coding iteration, two coding passes are applied to the coefficient matrix with a given reference threshold to determine the significance of wavelet coefficients. In the dominant pass, a coefficient can be coded as one of the four symbols: positive significance, negative significance, tree root, or isolated zero. All the coded symbols in dominant pass are put to p-segment. If the current coding coefficient is in the highest two resolution levels, the coded symbol is marked as Paramount skeleton (PS). PS symbols contain the root information of wavelet decomposition trees, and Morton scanning assures the continuity of PS residing in each p-segment. The very beginning PS symbols to be encrypted are marked as encrypted processions (EP), where the length of EP in each PS can be flexibly configured by users. Because run-length coding and arithmetic coding propagate any single bit error to the rest of the code stream, EP tightly controls PS and PS controls p-segment. Subordinate pass performs magnitude refinement after dominant pass, where the coded magnitude bits of each significant coefficient are put to v-segment. The reference threshold is decreased by half in each iteration, and EP, PS, p-segment, and v-segment are form bit-plane by bit-plane in an embedded manner. Selective encryption is not applied to v-segments because tree structures are only stored in p-segments. The data flow of proposed selective encryption is shown in Figure 1.

The length of each EP can be scalable from zero to the length of the containing PS, and encryption starts from the most significant bit-plane to the least significant bit-plane. The multiple EP indices in entropy coding table are

Table 1: Major symbol summarization for equations.

Equation Sym	Notations
Δd	Distortion reduction of one image packet
$\varepsilon[\Delta D]$	Image distortion reduction expectation
Bk	Association set of the kth EP block
g	Average segment loss ratio of each p-segment or v-segment
E_{MAX}	Energy budget constraint for transmitting one image
M_{MAX}	Link layer ARQ retry limit
$\overline{M}$	Average number of PDU transmissions
L_{RTS}	RTS frame length
L_{CTS}	CTS frame length
L_{DATA}	DATA frame length
L_{ACK}	ACK frame length
T_o	Link layer time-out value for receiving frames
BER_{CTRL}	BER of the control frames
BER	Desirable BER of the data frame
R_{CTRL}	Fixed PHY transmission rate for control frames
R_{DATA}	Scalable PHY transmission rate for data frames
P^{RX}	Power required for receiving circuits
$P_{\text{CTRL}}^{\text{TX}}$	Power required for transmitting RTS, CTS, ACK control frames
$P_{\text{DATA}}^{\text{TX}}$	Power required for transmitting DATA frames
Fitness	The fitness evaluation of a chromosome in genetic evolution

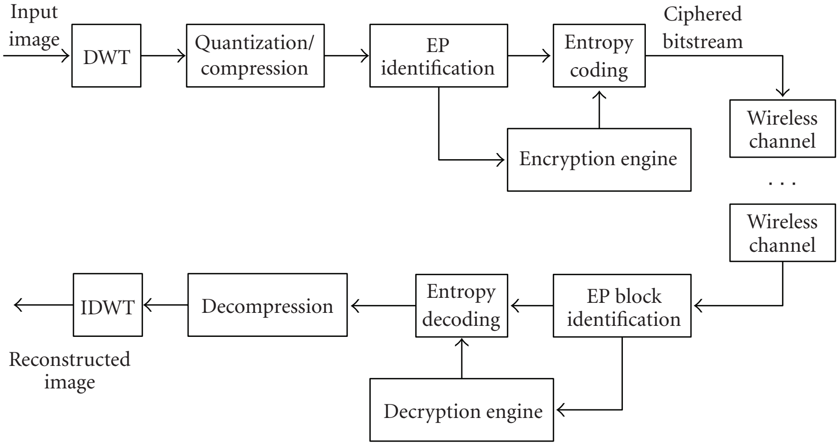


Figure 1: Selective encryption data flow for embedded bitstream.

encrypted and the ciphered code words form the EP blocks which are securely protected. Given the EP length in each p-segment and the encryption block length, one can determine the associated EP blocks of each EP. By this kind of selective encryption, the structure of compressed bitstream is effectively protected. Missing the descriptive information of the structure in p-segments, the magnitude information in v-segments will be placed on totally wrong positions of the wavelet coefficient matrix, which results in chaotic distributions of reconstructed image pixels. The small amount of PS information determines the structure of each p-segment. Modification of PS information scrambles the positions of wavelet coefficients associated with those PS. Strong-block-based robust encryption method, for example, 128-bit advanced encryption standard (AES) [17] applied on EP would make the entire decoding process hardly achievable. Thus, encrypting the tiny amount EP information in each bit-plane can efficiently make the image undecodable without cipher-key. The proposed selective encryption scheme is encryption algorithm independent, and simple low-complexity algorithm such as TEA [18] is applicable. The challenge of key exchange in unsecure networks is effectively solved because the significantly reduced encryption overhead makes the time consuming public-key encryption algorithms such as RSA [19] or ECC [20] algorithms feasible. Finally, image compression codec and entropy coding process in source coding domain can be blind of the existence of selective encryption module, making it format compliant, because p-segments, PS, and EP information are identified directly from the compressed bitstream, and the indices of EP symbols are encrypted after code book lookup. During decryption processes, original indices of EPs are reconstructed after EP block decryption. Then symbols and their runlengths of EP are determined from entropy code book using those decrypted EP indices. PS are reconstructed with the decrypted EPs. Then p-segments are recreated according to decoded PS information, and v-segments are recreated according to the decoded p-segments bit-plane by bit-plane in a progressive way.

3. CROSS-LAYER OPTIMIZATION PROBLEM FORMULATION

Here we formulate the cross-layer optimization as a distortion reduction maximization (distortion minimization) problem with strict energy budget constraint. The distortion of the reconstructed image and the energy consumption of transmitting this image are both related to the network resource parameters including desirable target BER, link layer ARQ retry limit, and physical layer transmission rate (translated to modulation schemes in this paper). These resources are fine tuned among EP blocks, p-segments, and v-segments.

The final bitstream is composed of a ciphertext stream and a plaintext stream. An example of the final bitstream is shown in Figure 2. Each EP block in the ciphertext stream controls several p-segments in the plaintext stream, and each p-segment in the plaintext stream controls all the p-segments and v-segments further down. Here we define $B_k = \{0,1,2,\dots\}$ as the kth EP block set containing the layer number of those encrypted p-segments associated with it. Without the kth EP block, all the p-segments associated with it will be useless for decoding. Referring to the example in Figure 2, $B_0 = \{0,1\}$ and $B_1 = \{2,3\}$. This can be formulated as p-segment0, psegment1 are associated with EP block0, while p-segment2 and p-segment3 are associated with EP block1; if EP block0 packet is dropped during transmission, both p-segment0 and p-segment1 cannot make distortion reduction contribution for decoding. Ciphertext stream is transmitted first, because plaintext stream cannot be reconstructed correctly without ciphertext stream. Then zigzag transmission is applied to the plaintext stream starting from p-segment0. Here two choices can be selected as the next transmitted packet after p-segment0: p-segment1 and v-segment0. Because p-segment1 controls all the p-segments as well as v-segments further down the plaintext stream while v-segment0 controls only all the v-segments further down, p-segment1 is transmitted as the next packet, and then v-segment0. The remaining p-segments and v-segments are transmitted in the same way. The bitstream is truncated if a specific p-segment is erased by wireless channel.

The total expected distortion reduction can be expressed in terms of transmissions error rates for each EP block, important p-segment, and unimportant v-segment respectively. Let N be the number of bitstream layers, $\Delta d_p(j)$ and $\Delta d_v(j)$ be the distortion reduction of the p-segment and v-segment in layer j, and g be the corresponding segment loss probability or segment loss ratio (SLR) of one segment packet during transmission. The total expected distortion reduction $\varepsilon[\Delta D]$ of the reconstructed image can be expressed as

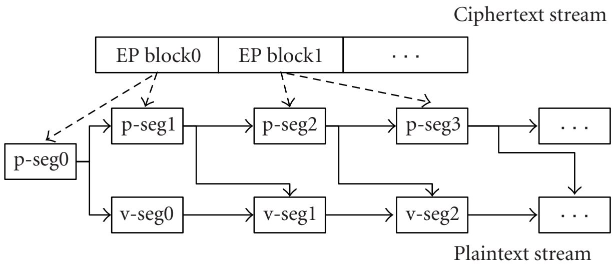


FIGURE 2: Image codestream format after selective encryption.

$$\begin{aligned}\varepsilon[\Delta D] = \sum_{i=0}^{N-1}&\left(\left(\sum_{j=0}^{i}\Delta d_p(j)\right)\cdot g_p(i+1)\right.\\&\left.\cdot\prod_{j=0}^{i}\left((1-g_p(j))\cdot\prod_{k|j\in B_k}(1-g_B(k))\right)\right)\\&+\sum_{j=0}^{N-2}\left(\left(\sum_{j=0}^{i}\Delta d_v(j)\right)\cdot\prod_{j=0}^{i}(1-g_v(j))\cdot g_p(i+1)\right.\\&\left.\cdot\prod_{j=0}^{i+1}\left((1-g_p(j))\cdot\prod_{k|j\in B_k}(1-g_B(k))\right)\right).\end{aligned} \tag{1}$$

In (1), each SLR g can be expressed in terms of link layer average packet loss ratio $\overline{\text{PER}}$ and the number of fragmentations $Q : g = 1-(1-\overline{\text{PER}})^{Q}$. Given the average packet loss ratio and distortion reduction measurement of each segment, the total expected distortion reduction can be expressed in close form in terms of desirable BER, ARQ retry limit. Let H denote the length of one segment, let L denote the link layer fragmentation threshold, the number of link layer fragmentations can be straightforwardly expressed as $Q = \lceil H/L \rceil$. Let N_B denote EP block count, let $\overline{E_B}$, $\overline{E_p}$ and $\overline{E_v}$ denote the energy consumption of transmitting one link layer fragment of EP block, p-segment, and v-segment, respectively. Let E_{MAX} denote the energy budget constraint, the overall optimization problem can be formulated as follows: finding the desirable BER, ARQ retry limit, and transmission rate for each EP block, p-segment, and v-segment, respectively, to achieve maximized overall distortion reduction:

$$\left\{\underset{i\in\{\text{EP}\}\cup\{p\text{-segment}\}\cup\{v\text{-segment}\}}{\text{BER}(i), M_{\text{MAX}}(i), R_{\text{DATA}}(i)}\right\} = \arg\max\{\varepsilon[\Delta D]\}. \tag{2}$$

Subject to the total energy budget constraint E_{MAX},

$$\sum_{i=0}^{N_B-1} Q_B(i)\cdot\overline{E_B}(i) + \sum_{i=0}^{N-1} Q_p(i)\cdot\overline{E_p}(i) + \sum_{i=0}^{N-2} Q_v(i)\cdot\overline{E_v}(i) \le E_{\text{MAX}}. \tag{3}$$

In order to solve the overall optimization problem, we propose a simplified evolution approximation methodology based on genetic algorithm. We assume the channel state changes slowly. Because we use adaptive power control according to desirable BER value, the transmission rate, and hence, the modulation scheme in physical layer is uncorrelated to the distortion reduction expectation. Thus transmission rate can be optimized independently for minimal energy consumption. Letting all the p-segments use one desirable BER and letting ARQ retry limit pair automatically produces a layer based UEP, because the lengths of bitstream segments are almost nondecreasing through all layers. The desirable BER and ARQ retry limit assignments for v-segments work the same way. Furthermore, the lengths of EP blocks are determined by encryption algorithms, which are usually much shorter than p-segments packets. Thus desirable BER and ARQ retry limit assignments for EP blocks can be performed together with those of p-segments, while reducing solution space for optimization. The solutions of overall optimization are jointly simplified as $\{\mathrm{BER}_p, M_{\mathrm{MAX\_}p}, \mathrm{BER}_v, M_{\mathrm{MAX\_}v}\}$. The complexity of the cross-layer optimization problem is significantly reduced by this approximation. The proposed algorithm is formulated as follows. Note that it can be solved offline and various precalculated result patterns can be stored to lookup tables in sensor nodes.

(a) Initialization for gene binary coding and decoding: each element in the solution matrix $\{\mathrm{BER}_p, M_{\mathrm{MAX\_}p}, \mathrm{BER}_v, M_{\mathrm{MAX\_}v}\}$ is coded as a gene, thus each possible solution is coded as a chromosome.

(b) Set the population space size POP_SIZE and maximal generations G_MAX, and randomly generate the first generation with the specified population size.

(c) Calculate the fitness and perform fitness evaluation of each of each chromosome. Fitness is defined as the expected distortion reduction calculated using the chromosome Fitness $= \varepsilon[\Delta D]$ if the total energy consumption is less than or equals to the energy budget; otherwise the fitness is zero. Sort the chromosomes in descending order according to their fitness values.

(d) Select the elitism of parents in current generation according to the fitness of each chromosome. Denote the fitness of the ith chromosome as Fitness(i) where $0 \le i \le \mathrm{POP\_SIZE} - 1$, then the crossover probability of a chromosome with others is expressed as $p(i) = \mathrm{Fitness}(i) / \sum_{i=0}^{\mathrm{POP\_SIZE}-1} \mathrm{Fitness}(i)$. Start chromosome crossover from the chromosome with the highest probability until a new generation with the same population size is created.

(e) Calculate the number of performed generations. If the number of generations > G_MAX, then go to (f), else go to (c) to refine the next generation population.

(f) Output the best chromosome in the current population with the best fitness. This assures the maximum distortion reduction, while the energy consumption is within the budget constraint.

4. ENERGY MODELING WITH OPTIMAL TRANSMISSION POWER AND RATE

To model the link layer energy consumption and transmission quality, to optimize the transmission rate for minimized energy consumption, link layer transmission overhead should be considered. Both payload data transfer errors and overhead frame such as in RTS, CTS, and ACK loss cause the upper PDU delivery failure. A binary event tree illustrated in Figure 3 can be a good model for link layer frame delivery. Each edge in the tree denotes the probability of a specific frame loss event, and the corresponding leaf node denotes the energy consumption penalty of that event. More details have been presented in [21].

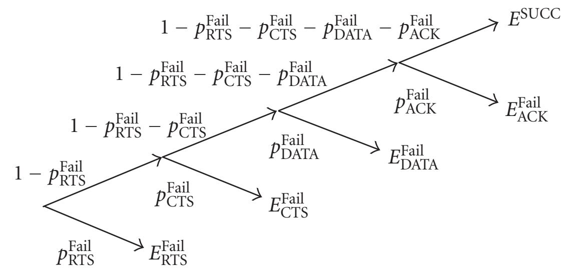

$1 - p_{\mathrm{RTS}}^{\mathrm{Fail}} - p_{\mathrm{CTS}}^{\mathrm{Fail}} - p_{\mathrm{DATA}}^{\mathrm{Fail}} - p_{\mathrm{ACK}}^{\mathrm{Fail}}$
E^{SUCC}
$1 - p_{\mathrm{RTS}}^{\mathrm{Fail}} - p_{\mathrm{CTS}}^{\mathrm{Fail}} - p_{\mathrm{DATA}}^{\mathrm{Fail}}$
$p_{\mathrm{ACK}}^{\mathrm{Fail}}$
$E_{\mathrm{ACK}}^{\mathrm{Fail}}$
$1 - p_{\mathrm{RTS}}^{\mathrm{Fail}} - p_{\mathrm{CTS}}^{\mathrm{Fail}}$
$p_{\mathrm{DATA}}^{\mathrm{Fail}}$
$E_{\mathrm{DATA}}^{\mathrm{Fail}}$
$1 - p_{\mathrm{RTS}}^{\mathrm{Fail}}$
$p_{\mathrm{CTS}}^{\mathrm{Fail}}$
$E_{\mathrm{CTS}}^{\mathrm{Fail}}$
$p_{\mathrm{RTS}}^{\mathrm{Fail}}$
$E_{\mathrm{RTS}}^{\mathrm{Fail}}$

Figure 3: Binary event tree of frame failure and energy cost.

For a single round handshake (without ARQ applied) transmission of an upper layer PDU, the frame error rate (FER) can be expressed as follows, given specific control frame bit error rate $\mathrm{BER}_{\mathrm{CTRL}}$ and desirable bit error rate BER for data frames:

$$\begin{aligned}\mathrm{FER} = 1 &- (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{RTS}}} \\ &+ (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{RTS}}} \cdot (1 - (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{CTS}}}) \\ &+ (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{RTS}} + L_{\mathrm{CTS}}} \cdot (1 - (1 - \mathrm{BER})^{L_{\mathrm{DATA}}}) \\ &+ (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{RTS}} + L_{\mathrm{CTS}} + L_{\mathrm{ACk}}} \\ &\cdot (1 - (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{ACK}}}) \\ = 1 &- (1 - \mathrm{BER})^{L_{\mathrm{DATA}}} \cdot (1 - \mathrm{BER}_{\mathrm{CTRL}})^{L_{\mathrm{RTS}} + L_{\mathrm{CTS}} + L_{\mathrm{ACk}}}.\end{aligned} \tag{4}$$

In this equation $\mathrm{BER}_{\mathrm{CTRL}}$ can be determined according to [22, 23] using the fixed control frame transmission power $P_{\mathrm{CTRL}}^{\mathrm{TX}}$, channel state factor A, noise power density N_0, and control frame transmission rate R_{CTRL}, assuming control frames are transmitted using BPSK modulation with constellation size $b = 1$,

$$P_{\mathrm{CTRL}}^{\mathrm{TX}} = R_{\mathrm{CTRL}} \cdot \frac{N_0}{A} \cdot \left[\mathrm{erfc}^{-1}(2 \cdot \mathrm{BER}_{\mathrm{CTRL}})\right]^2. \tag{5}$$

Also assume data frames are transmitted using scalable QAM-based modulation scheme (constellation size $b > 1$) and power control, the optimized transmission power for data frames is expressed as follows according to [24]:

$$P_{\mathrm{DATA}}^{\mathrm{TX}} = R_{\mathrm{DATA}} \frac{2(2^b - 1)}{3b} \frac{N_0}{A} \left[\mathrm{erfc}^{-1}\left(\frac{(b/2)\mathrm{BER}}{1 - (1/2^{b/2})}\right)\right]^2. \tag{6}$$

The frame error rate is reduced while the number of retransmission is increased if automatic retransmission request (ARQ) is applied. According to [14] the average number of transmissions $\overline{M}$ can be expressed as a nondecreasing function of link layer ARQ retry limit $\mathrm{M}_{\mathrm{MAX}}$:

$$\begin{aligned}\overline{M} &= 1\cdot(1-\mathrm{FER})+2\cdot\mathrm{FER}\cdot(1-\mathrm{FER})\\ &\quad+\cdots+M_{\mathrm{MAX}}\cdot\mathrm{FER}^{M_{\mathrm{MAX}}-1}\cdot(1-\mathrm{FER})\\ &\quad+(M_{\mathrm{MAX}}+1)\cdot\mathrm{FER}^{M_{\mathrm{MAX}}}\\ &=\frac{1-\mathrm{FER}^{M_{\mathrm{MAX}}+1}}{1-\mathrm{FER}}.\end{aligned}\tag{7}$$

Thus the average packet error rate $\overline{\mathrm{PER}}$ provided to upper layer can be approximated as

$$\overline{\mathrm{PER}}=\mathrm{FER}^{(1-\mathrm{FER}^{M_{\mathrm{MAX}}+1})/(1-\mathrm{FER})}.\tag{8}$$

It is clear that the average packet error rate is independent of transmission rate R_{DATA}, thus transmission rate (modulation) optimization can be performed separately from distortion reduction optimization. Let $\overline{E}$ denote the average energy consumption of delivering a PDU with length L_{DATA}, then $\overline{E}$ can be expressed as a function of R_{DATA}. The optimal transmission rate (modulation scheme) can be simply determined by treating $\overline{E}$ as a consecutive function of R_{DATA} and getting the first order derivative $\partial(\overline{E})/\partial(R_{\mathrm{DATA}})$. The discrete transmission rate closest to the zero value first order derivative is selected if there is one R_{DATA} leading to $\partial(\overline{E})/\partial(R_{\mathrm{DATA}})=0$. Otherwise the optimal transmission rate leading to minimal energy consumption must be the highest or the lowest rate depending on the slope of the function $\overline{E}(R_{\mathrm{DATA}})$. According to the binary event tree, the average energy consumption $\overline{E}$ can be expressed as (9) in close form of desirable BER , transmission rate, and ARQ retry limit given channel state information,

$$\begin{aligned}\overline{E} &= \frac{1-\mathrm{FER}^{M_{\mathrm{MAX}}+1}}{1-\mathrm{FER}}\\ &\quad\Bigg(\left(1-\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{RTS}}}\right)\\ &\quad\times\left(P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{RTS}}}{R_{\mathrm{CTRL}}}+P^{\mathrm{RX}}\left(\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}{R_{\mathrm{CTRL}}}+2T_o\right)\right)\\ &\quad+\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{RTS}}}\left(1-\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{CTS}}}\right)\\ &\quad\times\left(P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}{R_{\mathrm{CTRL}}}+P^{\mathrm{RX}}\left(\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}{R_{\mathrm{CTRL}}}+\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}+2T_o\right)\right)\\ &\quad+\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}\left(1-\left(1-\mathrm{BER}\right)^{L_{\mathrm{DATA}}}\right)\\ &\quad\times\left(P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}{R_{\mathrm{CTRL}}}+P^{\mathrm{TX}}_{\mathrm{DATA}}\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}\right.\\ &\quad\left.+P^{\mathrm{RX}}\left(\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}+L_{\mathrm{ACK}}}{R_{\mathrm{CTRL}}}+\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}+2T_o\right)\right)\\ &\quad+\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}}\left(1-\mathrm{BER}\right)^{L_{\mathrm{DATA}}}\\ &\quad\left(1-\left(1-\mathrm{BER}_{\mathrm{CTRL}}\right)^{L_{\mathrm{ACK}}}\right)\left(P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}+L_{\mathrm{ACK}}}{R_{\mathrm{CTRL}}}+P^{\mathrm{TX}}_{\mathrm{DATA}}\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}\right.\\ &\quad\left.+P^{\mathrm{RX}}\left(\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}+L_{\mathrm{ACK}}}{R_{\mathrm{CTRL}}}+\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}+T_o\right)\right)+\left(1-\mathrm{FER}\right)\\ &\quad\times\left(P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}+L_{\mathrm{ACK}}}{R_{\mathrm{CTRL}}}+P^{\mathrm{TX}}_{\mathrm{CTRL}}\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}\right.\\ &\quad\left.+P^{\mathrm{RX}}\left(\frac{L_{\mathrm{RTS}}+L_{\mathrm{CTS}}+L_{\mathrm{ACK}}}{R_{\mathrm{CTRL}}}+\frac{L_{\mathrm{DATA}}}{R_{\mathrm{DATA}}}\right)\right)\Bigg).\end{aligned}\tag{9}$$

Up to now the link layer transmission quality in (8) as well as energy consumption in (9) is modeled as close form functions of network resources including desirable BER, transmission rate, and ARQ retry limit. In the cross-layer optimization algorithm proposed in the previous section, the energy consumption and transmission quality of each packet are jointly fine tuned by adjusting the resource allocation. The optimal transmission rate is also determined independently from the cross-layer optimization algorithm.

5. SIMULATION

In this section, the performance of the proposed UEP scheme as well as the proposed position-based selective encryption is evaluated via simulation studies. The performance of transmission rate optimization is also evaluated, showing its significant energy efficiency gain. T-MAC [25] is selected for WSNs medium access, and multirate plug-in presented in [22] is selected for transmission rate optimization. The simulation parameters are stated as follows. Link layer fragmentation threshold is 36 bytes and MAC header is 11 bytes [26]. Control frame length is 13 bytes. Short preamble is applied with the length of 2 bytes [26], and the receive power is 0.01 mW. The noise power density N_0 is 4×10^{-21} J/Hz and the default value of channel state factor A is −100 dB. Frequency bandwidth is 1 MHz and the modulation is scaled by adjusting constellation size $b = 1, 2, 4, 6, 8$, respectively. Timeout value is set as one-fifth of the RTS transmission time with BPSK modulation. The test image is shown in Figure 4(a) with 64∗64 pixels and 8 bpp. AES standard encryption algorithm is utilized with 128 bits block cipher. The number of EP blocks can be scaled with two p-segments associated to one EP block, starting encryption from the most significant bit-plane (bit-plane 0) to the least significant bitplanes.

Figure 4 shows the original image as well as the decoded images with or without key. The proposed position-based selective encryption scheme is compared with the popular subband selection encryption approach. From these subfigures it is clear that without the correct key for decryption, the qualities of blindly decoded images are very low for both selective encryption schemes. The subband selective encryption renders very coarse images as shown in Figure 4(c) by hiding low-frequency wavelet coefficients, but image information energy concentration may not be directly related to intelligibility. The unprotected wavelet coefficients especially those in middle frequency subbands can still provide significant information for image reconstruction, because the structure of bitstream is unprotected. Unlike subband selection, the position-based selective encryption protects the bitstream structure as well as the positions of wavelet coefficients in all frequency bands. Thus middle- and high-frequency band wavelet coefficients cannot render a blurred image because the positions of those coefficients controlled by bitstream structure are effectively protected.

Figure 5 shows that the position-based selective encryption significantly reduces encryption overhead by reducing the number of encrypted blocks. Subband selection scheme achieves reduced image quality when the number

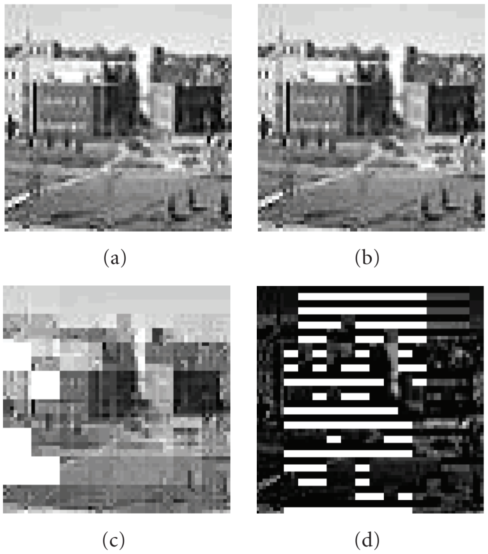


Figure 4: Test image and decoded images. (a) Original "building" image. (b) Correctly decoded image. (c) Blindly decoded image without AES key, for subband selection scheme with 2 blocks of 128-bit AES encryption. (d) Blindly decoded image without AES key, for position-based scheme with 2 blocks of 128-bit AES encryption.

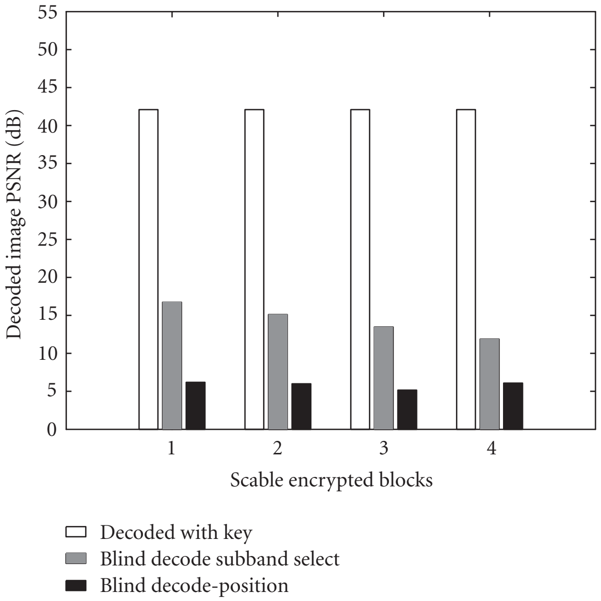


Figure 5: Selective encryption performances. Decoded image quality with different number of encryption blocks for "building" image.

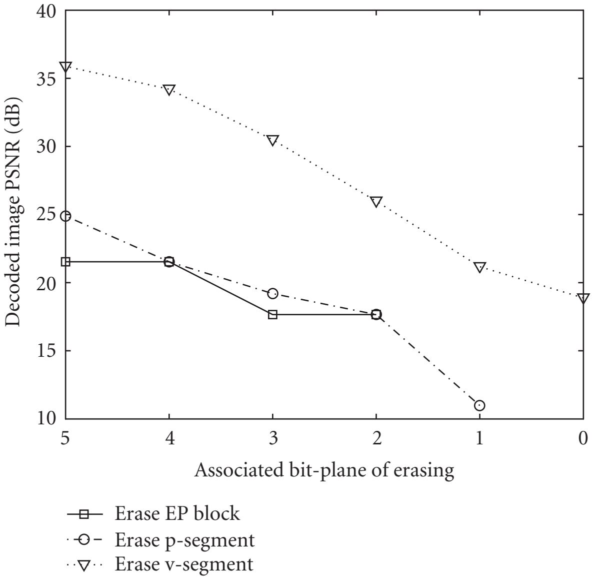


Figure 6: Image quality for "building" with erasing different p-segments, v-segments, or EP blocks in different bit-planes.

of encrypted blocks increases. To achieve acceptable image protection, more blocks need to be encrypted compared with position-based scheme. Again, the reason is due to the distortion reduction contribution of middle and high frequency coefficients. For the position-based selective encryption scheme, the original image is successfully protected even only encrypting one or two blocks of coarser bit-plane EP information. Without the correct EP information in coarser bit-planes, EP information in finer bit-planes can hardly make any contribution for distortion reduction due to wrong positions of significant wavelet coefficients.

The visual effect importance of EP blocks, p-segments, and v-segments for image reconstruction is illustrated in Figure 6, where image qualities with erased EP blocks, p-segments, or v-segments in different bit-planes are shown. The EP blocks contain p-segment structure information, and p-segments contain position information of wavelet coefficients. The magnitude information resides in v-segments. As shown in this figure, EP blocks are more important than p-segments and p-segments are much more important than v-segments. Thus, more robust protection should be applied to EP blocks and p-segments to improve image transmission quality, and less protection can be applied to v-segments to reduce energy consumption.

The energy efficiency gain of transmission rate optimization itself is shown in Figure 7, with different channel state information. Here normalized energy consumption is defined as the energy consumed by transmitting and receiving one bit of pure payload data. The normalized energy cost using optimal transmission rate and modulation scheme is much less than those using nonoptimized ones. For instance, draw a vertical line in Figure 7 at the point where channel state factor is −80 dB, 1.1626e-7 mJ energy is consumed per bit using the worst matched modulation scheme and transmission rate; 0.3265e-7 mJ energy is consumed by transmission using suboptimal matched modulation and transmission rate. However, the optimized transmission rate and modulation scheme achieves only 0.2899e-7 mJ energy consumption for each information bit. In this case, transmission rate and modulation optimization reduces 75% and 11% energy saving than the worst case and suboptimal transmission ones, respectively.

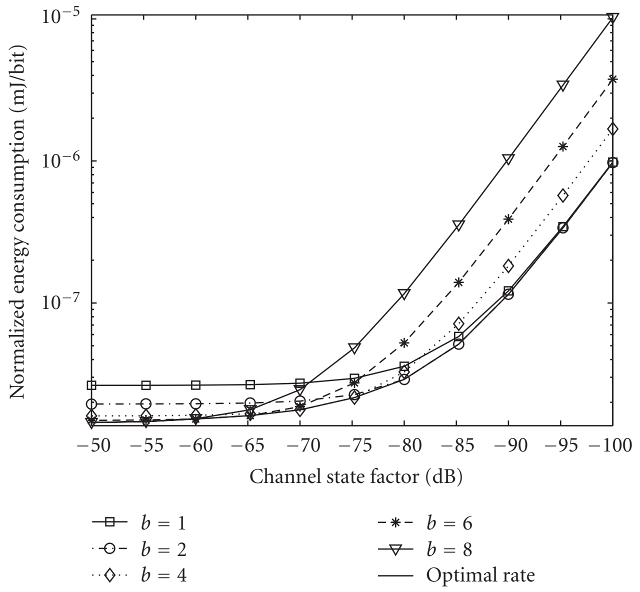
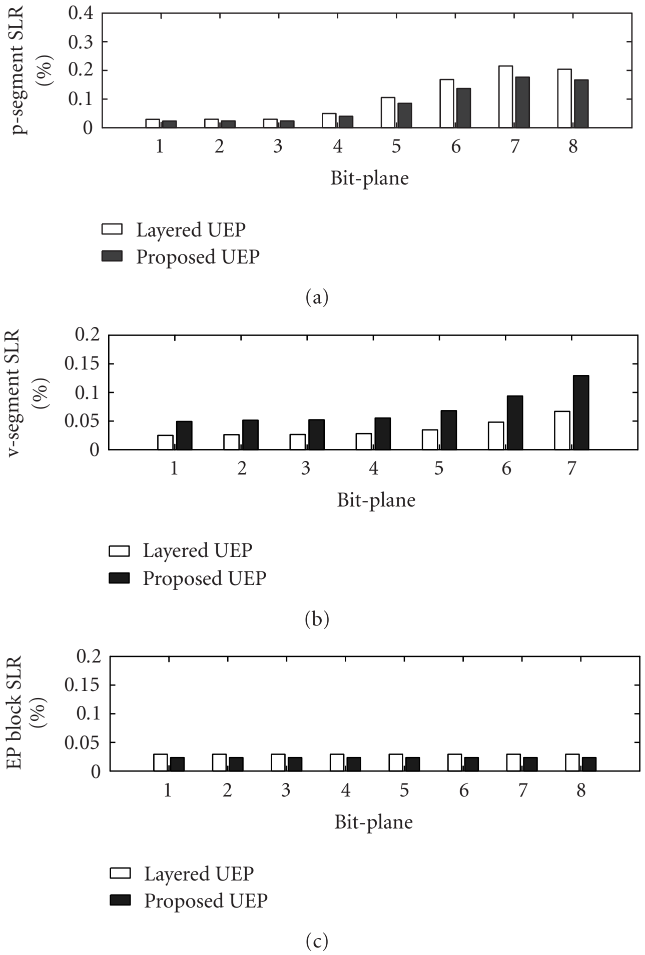


Figure 7: Normalized energy consumption in different channel conditions for different modulation schemes. Link layer fragmentation and payload are both 36 bytes, desirable BER is −50 dB, and retry limit is 3.



(a)

(b)

(c)

Figure 8: Segment loss ratio (SLR) of EP blocks, p-segments, and v-segments, for the scenario of 0.008 mJ energy budget.

To show the quality-energy improvement of the proposed UEP optimization scheme, the performance is compared with traditional layer based optimal UEP approach. Six scenarios are simulated with energy budget from 0.006 mJ to 0.0085 mJ with 0.0005 mJ granularity. The simulation results show that given the same energy budget, the proposed cross-layer optimal UEP approach enhances the image transmission quality while meeting energy budget requirement. The proposed cross-layer optimization approach fine tunes UEP between EP blocks, p-segments, and v-segments as well as the UEP between different bit-plane layers. The segment loss ratio (SLR) of all EP blocks, p-segments, and v-segments for transmitting the encrypted image with 0.008 mJ energy budget constraint is shown in Figure 8. SLR is directly related to the PER of each packet of that segment, which is in turn related to desirable BER and ARQ retry limit allocation of each packet. Compared with layered UEP, the SLRs of encrypted EP blocks and important p-segments are reduced while the SLRs of unimportant v-segments are increased. This is because the proposed UEP allocates more resources to EP blocks and p-segments and less resources to v-segments. The distortion reduction is increased due to more efficient resource allocation.

In the proposed novel UEP method, we have optimized the image quality and confined the energy consumption given the resource budget requirement. The image quality and energy consumption performance is shown in Figure 9. The vertical axis represents the distortion reduction expectation value while the horizontal axis is the corresponding communication energy cost. This figure demonstrates that the proposed cross-layer optimization scheme achieves enhanced image quality in comparison with the traditional layered UEP. At the same time, the proposed UEP can fine turn the network resource allocation, leading to higher energy efficiency under strict resource budget constraints.

6. CONCLUSION

This paper proposed a novel cross-layer UEP optimization approach for wireless image data delivery in sensor networks. Not only does it achieve high-energy efficiency but also image security is protected through creative image data encryption method. The proposed image encryption scenario fits well in the UEP approach resulting in enhancements for both image transmission quality and communication energy efficiency. In our approach, the communication energy efficiency is assured while image quality is optimized by specifically protecting encrypted blocks. A new position-based selective encryption scheme is developed that has very low-computation overhead and is appropriate for this original UEP optimization framework. The security aware cross-layer optimization approach has achieved maximal image transmission quality in wireless channels even though the energy budget constraints are met. Simulation results have demonstrated up to 5 dB image transmission quality improvement

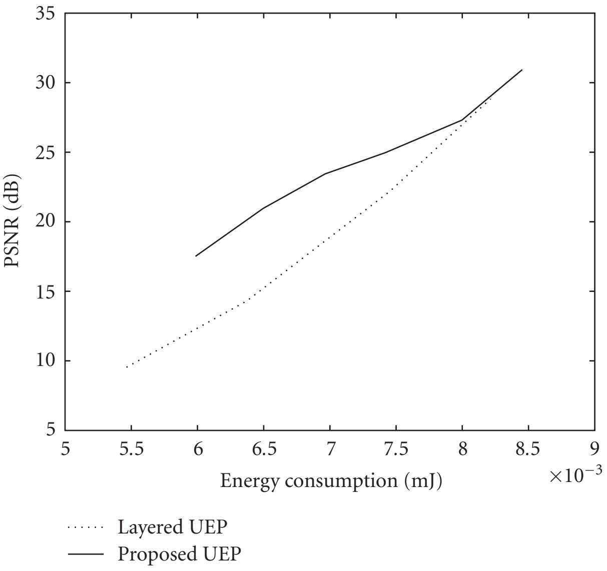


Figure 9: Image quality with different energy consumption.

for energy efficiently transmitting these robustly encrypted image data.

ACKNOWLEDGMENT

This research project was partially supported by the Nebraska Research Initiative grant.

REFERENCES

[1] T. Lookabaugh and D. C. Sicker, "Selective encryption for consumer applications," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 124–129, 2004.

[2] W. Yu, Z. Sahinoglu, and A. Vetro, "Energy efficient JPEG 2000 image transmission over wireless sensor networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 2738–2743, Dallas, Tex, USA, November-December 2004.

[3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[4] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

[5] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[6] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, 2003.

[7] M. S. Kankanhalli and T. T. Guan, "Compressed-domain scrambler/descrambler for digital video," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 356–365, 2002.

[8] C.-P. Wu and C.-C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, 2005.

[9] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proceedings of the 4th ACM International Conference on Multimedia (MULTIMEDIA '96)*, pp. 219–229, Boston, Mass, USA, November 1996.

[10] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905–917, 2006.

[11] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing*, vol. 48, no. 8, pp. 2439–2451, 2000.

[12] Z. Wu, A. Bilgin, and M. W. Marcellin, "Joint source/channel coding for multiple images," *IEEE Transactions on Communications*, vol. 53, no. 10, pp. 1648–1654, 2005.

[13] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bit streams," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 91–107, 2005.

[14] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

[15] M. van der Schaar and D. S. Turaga, "Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission," *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185–197, 2007.

[16] W. Wang, D. Peng, H. Wang, and H. Sharif, "A cross layer resource allocation scheme for secure image delivery in wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 152–157, Honolulu, Hawaii, USA, August 2007.

[17] National Institute for Standard Technology (NIST), "Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES)," Washington, DC, USA, 2001.

[18] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Proceedings of the 2nd International Workshop on Fast Software Encryption (FSE '94)*, vol. 1008 of *Lecture Notes in Computer Science*, pp. 363–366, Springer, Leuven, Belgium, December 1994.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Section 31.7: the RSA public-key cryptosystem," in *Introduction to Algorithms*, pp. 881–887, MIT Press and McGraw-Hill, Boston, Mass, USA, 2nd edition, 2001.

[20] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62–67, 2004.

[21] W. Wang, D. Peng, H. Wang, H. Sharif, and H. H. Chen, "Optimal image component transmissions in multirate wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Communications Conference (GLOBECOM '07)*, Washington, DC, USA, November 2007.

[22] C. Schurgers, O. Aberthorne, and M. B. Srivastava, "Modulation scaling for energy aware communication systems," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '01)*, pp. 96–99, Huntington Beach, Calif, USA, August 2001.

[23] S. Haykin, *Communication System*, John Wiley & Sons, New York, NY, USA, 3rd edition, 1994.

[24] W. Stallings, *Data and Computer Communications*, Prentice Hall, Upper Saddle River, NJ, USA, 7th edition, 2000.

[25] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 171–180, Los Angeles, Calif, USA, November 2003.

[26] http://tinyos.cvs.sourceforge.net/*checkout*/tinyos/tinyos-1.x/contrib/t-mac/tos/system/TMACMsg.h?revision=1.2.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 23817, 13 pages
doi:10.1155/2007/23817

Research Article

A Comparison Performance Analysis of QoS WLANs: Approaches with Enhanced Features

Ioannis Papapanagiotou,[1] Georgios S. Paschos,[2] and Michael Devetsikiotis[1]

[1] *Department of Electrical and Computer Engineering, North Carolina State University, P.O. 27695-7911, Raleigh, NC 27695, USA*
[2] *National Research Institute of Finland VTT, 02150 Espoo, Finland*

Received 30 May 2007; Accepted 13 July 2007

Recommended by Stavros Kotsopoulos

The main contribution of this work is to compare and enhance known methods for performance analysis of the IEEE 802.11e MAC layer, such as the use of Markov chains, queuing theory, and probabilistic analysis. It is the first paper that bases its outputs upon comparison of metrics such as complexity, flexibility, and accuracy, leading to the novel use of a metamodeling comparison. For the analysis, complexity theory and the *L*-square distance method for accuracy are used. In addition, the proposed analyses carry by themselves scientific interest, because they are extended enhancements with the latest EDCA parameters. A form of the PMF of the MAC delay and first-order moments are found using the PGF complex frequency domain function. The analyses incorporate a Gaussian erroneous channel in order to reflect the real conditions of the MAC layer.

1. INTRODUCTION

The wide deployment of WLANs has set an increased pace for extensive scientific studies of the IEEE 802.11 standard [1]. In addition, heterogeneous multimedia applications require advanced editing over the standard, so as to accomplish specific QoS characteristics [2]. Accomplishing such QoS features will make capable transmission of video and voice over new 3G-WLANs [3].

The core 802.11e standard proposes a new hybrid coordination function (HCF), which has the HCF controlled channel access (HCCA) and the enhanced distributed coordination access (EDCA) mechanisms, capable of offering access according to specific QoS features. A typical literature search demonstrates a number of performance analyses for the legacy IEEE 802.11 and 802.11e [4–7] or similar enhancements [8], which incorporate discrete time Markov chains (DTMC). Other models, such as [9–11], use alternative methods of analysis.

A trend has developed, since [4] first presented his case study, to improve and provide more accurate performance values. Due to the subject maturity, the space left for new models that could prompt scientific interest is small. Our novel approach is to propose amendments over these known analytical methods, find their accurate values, and open a new field of performance comparison. It is straightforward that, since new protocols tend to be analyzed by either DTMCs, queuing theory, or general probabilistic methods, the results of the proposed methods can be used to find the best method of analyzing forthcoming or known standards.

We have used three known models [5, 10, 11], which depict the three main methods of analysis, and they are extended according to QoS features proposed in the dot11e standard and error-prone channel. The first model uses DTMC analysis, which takes into account the state of the previous slot. The second one is based on elementary conditional probability arguments, and finally, in the last model, queuing theory and Little's theorem are used to analyze the standard. Accurate values of delay and a way for calculating the PMF of the MAC delay are also given. The proposed analysis alterations carry by themselves scientific interest, and could be studied separately.

Except from some already investigated features of the dot11e, additional ones have been added. These performance enhancements are summarized in the following.

(i) In our models, the effect of different retransmission limits among the access categories is implemented.

(ii) Freezing of backoff counters is taken into account.

(iii) A more accurate equation of saturation throughput is provided in correlation with a way of incorporating the AIFS differentiation phenomenon among the access categories.

(iv) It was also noted that higher ACs monopolized relatively quickly the channel, especially when the type of multimedia traffic was bursty. After revision of D4.0 of the IEEE 802.11e standard [2], the standard defined that after a successful transmission, the AC should get back to backoff, and contest again for the channel in the next time slot. This means that the state 0 of the DTMC cannot be chosen after a successful transmission a feature that older models have omitted or—partly mentioned [12], but not analyzed.

(v) The proposed models include Gaussian erroneous channel for EDCA. Although some works exist, for example, [13, 14], they tend to analyze such problems by implementing the bit error rate (BER) probability in the busy probability, which means that an error can make the channel busy. Since the MAC layer in the backoff level does not see errors and inner codes in the sublayers provide a specific coding gain, the erroneous channel implementation should be made in the performance analysis.

(vi) We analyze in the same environment the effect of Block-ACK and the efficiency of the new IEEE 802.11n introducing much higher transmission rates.

(vii) Most of the aforementioned analyses tend to give results for the basic transfer rate, in which analytical features such as the dual effect of the RTS/CTS in throughput are not shown. In our models, these are corrected while adding new features and giving the exact solution with RTS/CTS, basic access mode in 11 Mbps, and multiples of 24 Mbps transfer rate.

(viii) Z-transform is also used for the PGF of the MAC delay, and first-order moments are found through its first and second derivatives.

The proposed analysis alterations carry themselves scientific interest, and could be studied separately. Additionally, a metamodeling analysis is added, by processing the mathematical performance and simulation outcomes. Thus results are given in terms of

(i) complexity, based on big-O notation;

(ii) accuracy, based on L-square distance;

(iii) depiction of the states of the MAC protocol, nonsaturation easiness of implementation, and flexibility.

The simulation results are based on the HCCA model included in the last version of OPNET modeler 12. The proposed models require advanced knowledge of [2, 4, 10], since formulas and other proved explanations are taken as prerequisites. The paper is organized as follows. In Section 2, we provide numerical analysis of the transmission probability (τ_i) and mean backoff duration ($E[\mathrm{BD}]_i$) of each of the three models. In Section 3, taking into account the values of throughput and delay the transmission rates are extracted, with various conditions of the channel and features enabled or disabled. A third subsection is also given for the analysis of the Block-ACK feature. In Section 4, validation, results, and comparison analysis and evaluation are provided. In Section 5 a conclusive discussion is made upon advantage and disadvantage of each one.

Table 1: 802.11e EDCA standard parameters.

	AC0	AC1	AC2	AC3
Type of service	VoIP	Video	Best effort	Backround
CWmin[i]	7	15	31	31
CWmax[i]	15	31	1023	1023
AIFS[i]	SIFS + 2	SIFS + 2	SIFS + 3	SIFS + 7

2. NUMERICAL ANALYSIS OF THE MODELS

In [2], quality of service succeeded via using four access categories (ACs) with different transmission parameters each. The standard uses different values of AIFS[i], $CW\min_i$, $CW\max_i$ and backoff persistent factor (πf_i), $i = \{0, 1, 2, 3\}$. Finally, transmission opportunity (TxOP) is the maximum amount of bytes that a station is allowed to transmit consecutively before it releases the channel. In this paper, all ACs are considered to send packets with equal number of bytes below TxOP limit and therefore TxOP is not studied. We define $W_{i,0} = CW_{i,\min} + 1$, where $W_{i,j}$ is the contention window size and j is the backoff stage. m_i is defined as the retry limit, after which the contention window remains the same for a number of retransmissions. When the backoff exponential algorithm reaches L_i (long retry limit) times of retransmission and there is a collision, the packet is dropped. In the legacy 802.11 [1], persistent factor (πf_i) has the value of 2, which means that after every collision, the backoff contention window doubles its value. In [2], persistent factor can have different values according to each access category

$$W_{i,j} = \begin{cases} \lfloor (\pi f_i)^j W_{i,0} \rceil, & j = 0, 1, \dots, m_i, \\ \lfloor (\pi f_i)^{m_i} W_{i,0} \rceil, & j = m_i + 1, \dots, L_i, \end{cases} \tag{1}$$

where $\lfloor \cdot \rceil$ is the closest integer function. In Table 1, a summary of the EDCA is presented.

Before defining the mathematical analysis, the following assumptions have been made regarding all models. The number of stations N_i is finite and equal for all ACs and contends only in a single-hop network. There is a constant packet generator and the network is saturated, which means that there is always a packet ready to transmit in each terminal. The channel is erroneous, with uniform distributed errors, and there are no hidden terminal, capture effects, and link-adaptation mechanisms. Finally, as in all existing models, the transmit probability is considered to be independent per station.

2.1. Markov chain model (model 1)

A three-dimensional DTMC is proposed, which presents the effect of contending terminals on the channel access probability of each access class (AC), and is described by the stationary probabilities $b_{i,w,j,k}$. The parameter $i = \{0, 1, 2, 3\}$

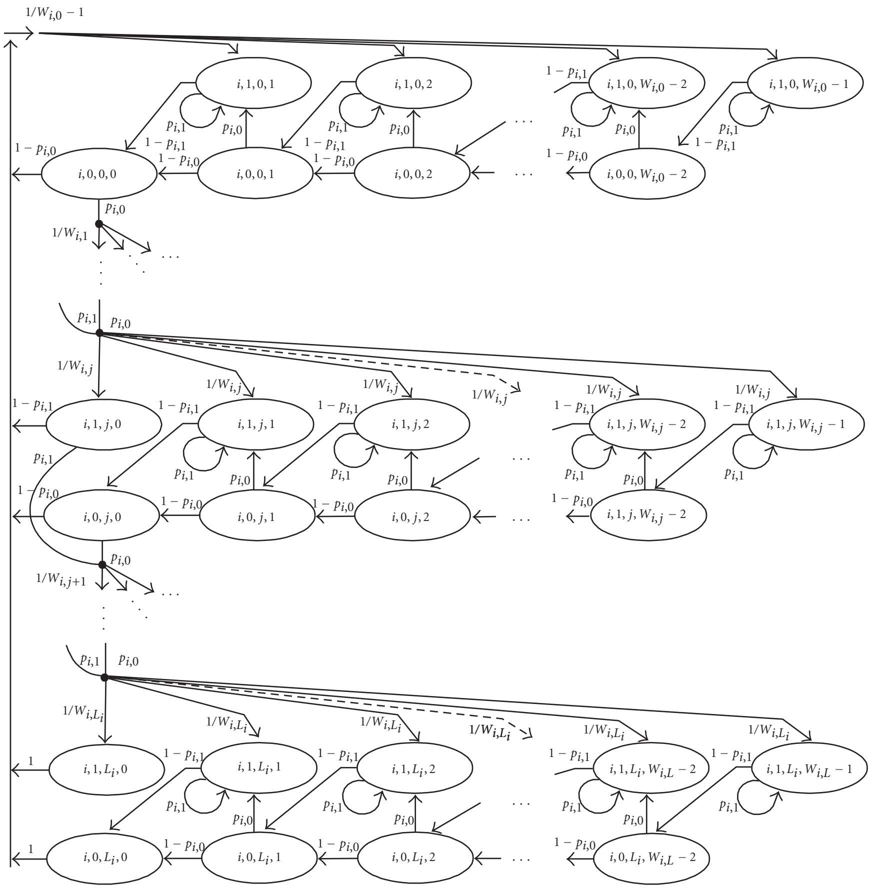


Figure 1: Analytical Markov chain for each access category AC[i].

describes the four access categories, which differentiate the access method according to the dot11e standard.

The first dimension, w, represents the condition of the previous slot, where 1 is for the busy channel and 0 for the idle channel. Similar to [5], a division is needed since special cases exist according to the state of the previous slot. If it was idle, all access categories of all stations may access the channel if their backoff counter is decremented to zero.

On the other hand, if the previous slot was busy, another division must take place. A busy slot can occur if there is a collision or a transmission of another station. In the first case, the stations that did not participate in the collision have frozen their backoff counter and will not be able to transmit. Instead, the stations that collided can transmit in the next slot if they choose a new backoff value equal to 0. In the second case, when there is a successful transmission, none of the stations can transmit in the next time slot. This happens specifically for the standard IEEE 802.11e and not for the legacy IEEE 802.11. The latest defines that after a successful transmission, the contention window starts from 1 and not from 0. All these are considered in the provided analysis and shown in the DTMC of Figure 1, which refers to each access category separately. Note that the state $\{i, 1, 0, 0\}$ is missing.

The other two symbols are j for the backoff stage described above and k which accounts for the backoff delay and takes values $k \in [0, 1, \ldots, W_{i,j} - 2]$ for $w = 0, k \in [0, 1, \ldots, W_{i,j} - 1]$ for $j > 0$ and $w = 1$, and $k \in [1, \ldots, W_{i,j} - 1]$ for $j = 0$ and $w = 1$. In [5], a similar DTMC is used for the legacy dot11. This model is extended considerably so as to include all the new characteristics of dot11e and a finite retry limit. Our analysis also deviates from [5] since the first state of the chain does not exist.

The probability $p_{i,0}$ (or $p_{i,1}$) is that another terminal's access category is transmitting after an idle period (or after a busy period), without errors. The opposite case, that the channel remains idle after an idle period, is represented by q_0 (or after a busy period q_1). After these explanations, all the transitions of the DTMC have been verified and the following equations are accrued:

$$b_{i,1,j,0} = \psi_{i,j} b_{i,0,0,0} \tag{2}$$

for $j = 1, 2, \ldots, L_i$,

$$b_{i,1,j,k} = \frac{1 + p_{i,0}(W_{i,j} - 1 - k)}{1 - p_{i,1}} \psi_{i,j} b_{i,0,0,0} \tag{3}$$

for $k = 1, 2, \ldots, W_{i,j} - 1$ and $j = 0, \ldots, L_i$,

$$b_{i,0,j,k} = (W_{i,j} - 1 - k)\psi_{i,j} b_{i,0,0,0} \tag{4}$$

for $k = 0, 1, \dots, W_{i,j} - 2$ and $j = 1, \dots, L_i$, where

$$\psi_{i,j} = \begin{cases} \dfrac{1}{W_{i0} - 1}, & j = 0, \\ \dfrac{p_{i,0}}{W_{i,1}}, & j = 1, \\ \dfrac{p_{i,0}}{W_{i,1}} \Pi_{i,j}, & j = 2, 3, \dots, m_i, \\ \dfrac{p_{i,0}}{W_{i,1}} \Pi_{i,m_i} P_{i,j}, & j = m_i + 1, \dots, L_i. \end{cases} \tag{5}$$

$\Pi_{i,j}$ and $P_{i,j}$ are defined as

$$\begin{aligned} \Pi_{i,j} &= \prod_{x=2}^{j} \left[\frac{p_{i,1}}{W_{i,x}} + \frac{p_{i,0}}{W_{i,x}} (W_{i,x-1} - 1) \right], \\ P_{i,j} &= \prod_{x=m_i+1}^{j} \left[\frac{p_{i,1}}{W_{i,m_i}} + \frac{p_{i,0}}{W_{i,m_i}} (W_{i,m_i} - 1) \right]. \end{aligned} \tag{6}$$

Applying the normalization condition for each access category's DTMC, as each exponential backoff algorithm runs independently, we have

$$\begin{aligned} &\sum_{k=0}^{W_{i,0}-2} b_{i,0,0,k} + \sum_{k=1}^{W_{i,0}-1} b_{i,1,0,k} \\ &\quad + \sum_{j=1}^{L_i} \left[\sum_{k=0}^{W_{i,j}-2} b_{i,0,j,k} + \sum_{k=0}^{W_{i,j}-1} b_{i,1,j,k} \right] = 1. \end{aligned} \tag{7}$$

After solving this equation, $b_{i,0,0,0}$ is found as

$$b_{i,0,0,0} = \frac{2(1 - p_{i,1})}{K_i + \Lambda_i}, \tag{8}$$

$$\begin{aligned} K_i &= W_{i,0}(1 - p_{i,1}) + p_{i,0}(W_{i,1} - 1)(2 - p_{i,1}) \\ &\quad + 2p_{i,0}(W_{i,0} - 2) + 4, \\ \Lambda_i &= \sum_{j=2}^{L_i} \psi_{i,j} W_{i,j} [(W_{i,j} - 1)(1 - p_{i,1} + p_{i,0}) + 2]. \end{aligned} \tag{9}$$

The probabilities of accessing the channel in a time slot, whether the previous slot was idle or busy, are given by the following equations:

$$\tau_{i,w} = \begin{cases} \dfrac{\sum_{j=0}^{m_i} b_{i,0,j,0} + \sum_{j=m_i}^{L_i} b_{i,0,j,0}}{P_{\text{idle},}}, & w = \text{idle}, \\ \dfrac{\sum_{j=1}^{m_i} b_{i,1,j,0} + \sum_{j=m_i}^{L_i} b_{i,1,j,0}}{1 - P_{\text{idle},}}, & w = \text{busy}, \end{cases} \tag{10}$$

where P_{idle} is derived by the solution of $P_{\text{idle}} = q_0 P_{\text{idle}} + q_1(1 - P_{\text{idle}})$, and describes the probability that the channel is idle in the previous time slot (take notice that this is different from the current idle slot symbolized below as P_{idle}).

2.1.1. *Successful transmission probability*

The probabilities that the channel remains idle after an idle (or busy) time slot can be found in a straightforward manner by supposing that no other station transmits in that time slot:

$$q_w = \prod_{i=0}^{3} (1 - \tau_{i,w})^{N_i}. \tag{11}$$

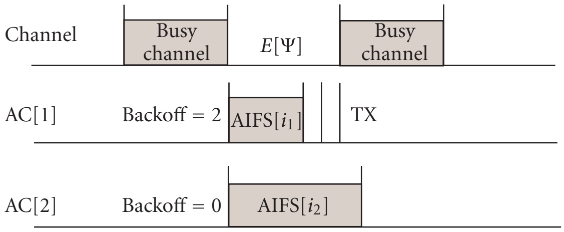


Figure 2: The AIFS differentiation prevents a collision that otherwise would have happened.

The probability of another AC transmitting is relatively complex. Except from the other station's AC transmition, an intercollision handler and virtual collision handler must also be taken into account. In the proposed analysis, such a collision handler is also implemented, adding as well as a correlation measure which gives a close approximation of the intercollision problem.

The phenomenon of intercollision happens when two ACs have different AIFS, and the one with the higher AIFS and higher $E[\Psi]$ has a smaller backoff value. Thus it may happen that these ACs will collide and the differentiation offered from the use of AIFS will be lost. see Figure 2:

$$r(i_1, i_2) = \max\left[1 - \frac{\text{AIFS}[i_1] - \text{AIFS}[i_2]}{E[\Psi]}, 0\right], \quad i_1 \geq i_2, \tag{12}$$

where $E[\Psi]$ is the mean consecutive number of idle slots. The notation min is used to maintain the accuracy of the model:

$$E[\Psi] = \min\left(\frac{P_{\text{idle}}}{1 - P_{\text{idle}}}, 1\right). \tag{13}$$

This specific correlation measure simplifies the analysis, because it does not increase the complexity of the mathematical analysis when trying to solve the DTMC. Therefore, the probabilities of a transmission failure (taking into account the collision probability and error probabilities) after an idle or busy slot are

$$\begin{aligned} p_{i,0} &= 1 - \prod_{z<i} (1 - \tau_{z,\text{idle}})^{\lfloor N_z \cdot r(z,i) \rfloor} \\ &\quad \times (1 - \tau_{i,\text{idle}})^{N_i - 1} \prod_{z>i}^{3} (1 - \tau_{z,\text{idle}})^{N_z}, \end{aligned} \tag{14}$$

$$p_{i,1} = 1 - (1 - \tau_{i,\text{busy}})^{N_i - 1} \prod_{z>i}^{3} (1 - \tau_{z,\text{busy}})^{N_z}. \tag{15}$$

The successful transmission probability in a time slot of an AC is

$$\begin{aligned} P_{s,i} &= P_{\text{idle}} \cdot N_i \cdot \tau_{i,\text{idle}} \cdot \prod_{z<i} (1 - \tau_{z,\text{idle}})^{\lfloor N_z \cdot r(z,i) \rfloor} \\ &\quad \times (1 - \tau_{i,\text{idle}})^{N_i - 1} \cdot \prod_{z>i} (1 - \tau_{z,\text{idle}})^{N_z} \\ &\quad + (1 - P_{\text{idle}}) \cdot N_i \cdot \tau_{i,\text{busy}} \cdot (1 - \tau_{i,\text{busy}})^{N_i - 1} \\ &\quad \times \prod_{z>i} (1 - \tau_{z,\text{busy}})^{N_z}. \end{aligned} \tag{16}$$

2.1.2. *Mean backoff duration*

$E[\mathrm{BD}]_i$ is defined as the mean backoff delay, which is the sum of the backoff transitions $E[X]_i$ when the channel is idle, and the delay due to freezing $E[F]_i$, all of which referring to each AC:

$$E[\mathrm{BD}]_i = E[X]_i\sigma + E[F]_i. \tag{17}$$

The backoff transition delay is $E[X]_i$ defined as the number of slot times k that are needed for the AC to reach state 0 and transmit, considering that the counter is at the state $b_{i,1,j,k}$ or $b_{i,0,j,k}$. The number of times the counter is stopped (freezes) is not taken into account, as they are calculated separately in (20)

$$E[X_i] = \frac{\sum_{j=0}^{L_i}\sum_{k=0}^{W_{i,j}-2} k b_{i,0,j,k}}{P_{\mathrm{idle}}}. \tag{18}$$

After some algebra, the backoff transition delay is found as

$$\begin{aligned} E[X]_i &= \frac{b_{i,0,0,0}\cdot M_i}{12P_{\mathrm{idle}}}, \\ M_i &= \sum_{j=1}^{L_i} \psi_{i,j}(W_{i,j}-1)(W_{i,j}-2)(4W_{i,j}-3), \end{aligned} \tag{19}$$

and the delay due to freezing of the backoff counter is calculated as follows. Note that the denominator of $E[F]_i$ is the exact opposite of the denominator of $E[X]_i$:

$$E[F]_i = \frac{E[N_f]_i}{1-P_{\mathrm{idle}}}\left[\sum_{i=0}^{3} P_{s,i}T_{s,i} + PcT_{c,i}\right], \tag{20}$$

where $E[N_f]_i$ is the number of freezes, and it is analyzed as the fraction of the mean value of the counter $E[X]_i$ divided with the mean consecutive number of idle slots, defined in (13).

In order to find the MAC delay, the mean delay must be subtracted from the dropping delay defined as

$$\begin{aligned} E[\mathrm{Drop}]_i &= b_{i,0,0,0}(T_c + T_{\mathrm{protect}})\psi_{i,L_i} \\ &\quad\times\left[\frac{1+p_{i,0}(W_{i,L_i}-1)}{1-p_{i,1}}p_{i,1} + (W_{i,j}-1)p_{i,0}\right]. \end{aligned} \tag{21}$$

2.2. *Elementary conditional probability analysis (ECPA)*

The proposed probabilistic analysis is simpler than the previous solution of DTMC, because it is based on conditional probabilities of each access category independently [10]. Two events are defined here. The first is called TX_i and means that a station's AC is transmitting a frame into a time slot, and the second is $s = j$ is that the station's AC is in backoff stage j, where $j \in [0, L_i]$, whenever L_i is different in basic and RTS/CTS method according to the short and long retry limit. From Bayes' theorem, we have

$$P(TX_i)\frac{P(s=j \mid TX_i)}{P(TX_i \mid s=j)} = P(s_i = j). \tag{22}$$

2.2.1. *Successful transmission probability*

From [10, equations (2)–(7)], with amendments so as to include the four ACs ($i = \{0,1,2,3\}$), we have that the transmission probability can be written as

$$\tau_i = \frac{1}{((1-p_i)/(1-p_i^{L_i+1}))\sum_{j=0}^{L_i} p_i^j\cdot(1+E[\mathrm{BD}]_{i,j})}. \tag{23}$$

In order to include the freezing of the backoff counters, a distinction must be made. The interruption of the backoff period of the tagged station can occur by three different events and is analyzed as follows. The first is the collision of two or more stations, the second is the transmission of a single station other than the tagged one, and the third is the transmission of a single station. p_i is the probability that the tagged station is interrupted the transmission of any other station (one or more) being

$$p_i = 1 - \prod_{z\geq i}(1-\tau_z)^{N_{m,z}} \tag{24}$$

and $N_{m,z} = N_z - \delta_{m,z}$ ($\delta_{m,z}$ is the Kronecker function [15]).

The probability that the tag station is interrupted by the transmission of a single station (one exactly) is given by

$$p_i' = \binom{N_i-1}{1}\cdot\tau_i\cdot(1-\tau_i)^{N_i-2}\prod_{z>i}(1-\tau_z)^{N_z}. \tag{25}$$

2.2.2. *Mean backoff duration*

In order to find the mean backoff duration, the duration of each exponential backoff must be found, which should include the finite limit of $CW - H[j-1]$ and the freezing of backoff counter each time the slot is detected busy. For example, if there were k freezings, then the delay would be $E[\mathrm{SD}]_{i,j} = \sum_{k=0}^{CW_i-H[j-1]}(kp^k)\cdot(1-p_i)$, which gives finally

$$E[\mathrm{SD}]_{i,j} = \frac{CW_i - H[j-1]}{2\cdot(1-p_i)}. \tag{26}$$

Taking into account all the possible series of the exponential backoff, the mean backoff duration is given from

$$E[\mathrm{BD}]_{i,j} = \begin{cases} \displaystyle\sum_{k=0}^{CW_{,}-1}\frac{\mathrm{BD}_{i,j}^k}{CW_{i,j}}, & 0 \leq j \leq m_i, \\ E[\mathrm{BD}]_{i,m_i}, & m_i \leq j \leq L_i. \end{cases} \tag{27}$$

2.3. *Queuing network (QN) model analysis*

This analysis is based on the Choi et al. [11] queuing model. In our model, the approach towards the network is different than any one proposed before, because it models the behavior of each AC, which contains N_i stations instead of a single station, independently (see Figure 3). Except from that, each backoff stage is modeled by a $G/G/\infty$ queuing system.

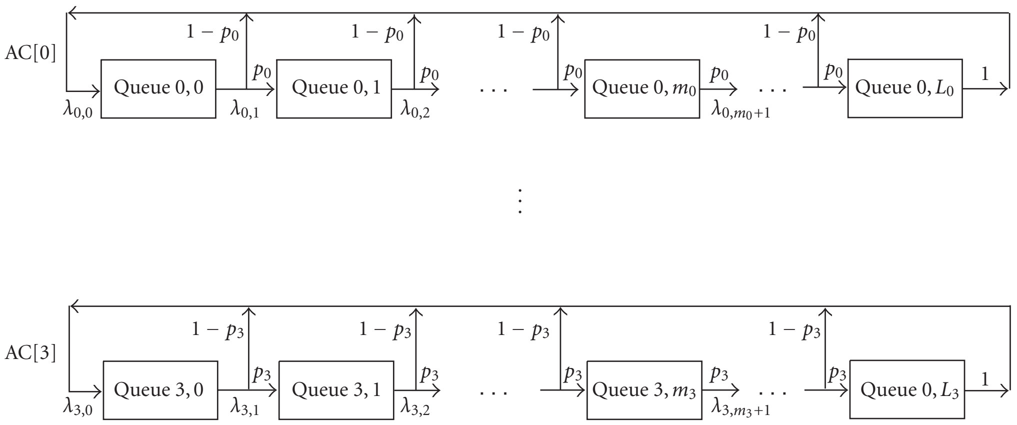


Figure 3: Queuing network model and analyzing the backoff duration with Z-transform.

An infinite number of parallel servers are used so that each queue can serve all stations simultaneously without queuing delay. In addition, the queuing delay is found by taking into account the freezing of backoff counters. Similar to the previous two models, the first queue has less length than the other ones. This solution is based on the assumption that the transmission probability can be expressed as the total attempt rate λ_i, divided by the number of stations of each AC independently:

$$\tau_i = \frac{\lambda_i}{N_i}. \tag{28}$$

Let us define $\lambda_{i,j}$ as the arrival rate and $\mu_{i,j}$ as the average service rate, at each queue of each AC, where $\mu_{i,k}$ is found from the backoff duration of each queue, which is calculated from the Z-transform of each queue given below. From Little's theorem, the number of stations in each queue and in each AC can be found by

$$N_{i,j} = \frac{\lambda_{i,j}}{\mu_{i,j}}. \tag{29}$$

The transition probability from one queue to the next one is related the arrival rates. However, it should be noted that a small difference is found from queue 0 to queue 1, as it has been explained that the value 0 of the first backoff window is not chosen:

$$\lambda_{i,j+1} = p_i\lambda_{i,j}, \quad j = 0, \dots, L_i - 1, \tag{30}$$

where the total attempt rate λ_i is given by

$$\lambda_i = \sum_{j=0}^{L_i} \lambda_{i,j} = \lambda_{i,0}\frac{1 - p_i^{L_i+1}}{1 - p_i} \tag{31}$$

and the average service rate of each queue is found from

$$\mu_{i,j} = \frac{1}{1 + E[\mathrm{BD}]_{i,j}}. \tag{32}$$

The reason for adding 1 with $E[\mathrm{BD}]_{i,j}$ is that to get out of the queue one more slot is required, corresponding to transmission.

2.3.1. Successful transmission probability

Having calculated $\lambda_{i,j}$ and $\mu_{i,j}$, we can use again Little's theorem:

$$N_i = \sum_{j=0}^{L_i} N_{i,j} = \lambda_{i,0}\sum_{j=0}^{L_i} p_i^j\left(1 + E[\mathrm{BD}]_{i,j}\right). \tag{33}$$

In (33), the sum is too complicated to be solved and it needs computer mathematical tools. Finally τ_i is computed from (28):

$$\tau_i = \frac{\lambda_i}{N_i} = \frac{1}{\left((1-p_i)/(1-p_i^{L_i+1})\right)\sum_{j=0}^{L_i} p_i^j \cdot \left(1 + E[BD]_{i,j}\right)}. \tag{34}$$

From the above mathematical results, we can see that (23) and (34) are the same. So both types of solutions give similar results. Thus to find the probability of successful transmission in both models, we use

$$P_{s,i} = N_i\cdot\tau_i\cdot\prod_z \left(1 - \tau_z\right)^{N_{m,z}}. \tag{35}$$

2.3.2. Mean backoff duration

The mean backoff duration is similar to the one analyzed in the ECPA analysis.

3. THROUGHPUT AND DELAY

3.1. Saturation throughput

3.1.1. Block-ACK disabled

The saturation throughput for every AC and for packets with mean length $E[L]$ is given by

$$S_i = \frac{p_{e,i}P_{s,i}E[L]}{T_{\mathrm{slot},i}}, \tag{36}$$

where

$$T_{\mathrm{slot},i} = P_{\mathrm{idle}}\sigma + \sum_{i=0}^{3}\left[\left(1 - p_{e,i}\right)P_{s,i}T_{s,i}\right] + \left(Pc + p_{e,i}P_{s,i}\right)T_{c,i}. \tag{37}$$

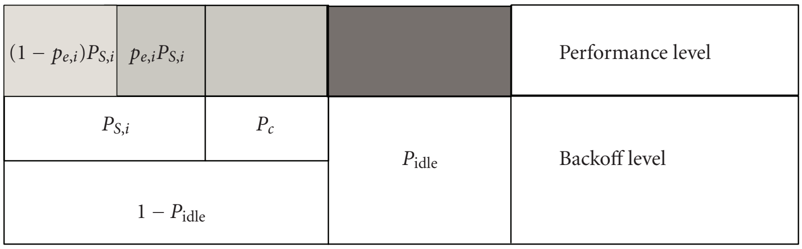


Figure 4: Probability spaces in the analysis of 802.11 standards for error behavior.

As shown in Figure 4, the probability of error affects the successful transmission probability only. Thus whenever both the events of successful transmission probability and error happen, they are regarded as collisions.

Since the errors are uniformly distributed, the error events are independent and identically distributed (i.i.d.) thus the frame error probability is given by

$$p_{e,i} = \left(1 - p_{e,i}^{\text{data}}\right)\left(1 - p_{e,i}^{\text{ACK}}\right)\left(1 - p_{e}^{\text{RTS}}\right)\left(1 - p_{e}^{\text{CTS}}\right). \tag{38}$$

$p_{e,i}^{\text{data}}$ and $p_{e,i}^{\text{ACK}}$ show the uniformly distributed errors in the data packet and in the acknowledgment, and the same applies for the probabilities $p_{e,i}^{\text{RTS}}$ and $p_{e,i}^{\text{CTS}}$ which are used only in RTS and CTS access method. If Basic access method is used, then $p_{e,i}^{\text{RTS}} = p_{e,i}^{\text{CTS}} = 0$.

Then the collision probability is

$$Pc = 1 - P_{\text{idle}} - \sum_{i=0}^{3} P_{s,i}. \tag{39}$$

We must also mention that whenever the retry limit is reached, the packet is dropped. However such a probability is included in Pc, and the retransmissions required after a collision or a drop are based on the upper layer and do not affect the performance of the studied MAC layer.

3.1.2. Block-ACK enabled

Another characteristic of the IEEE 802.11e standard is the Block-ACK feature, which is not obligatory. However Block-ACK can mitigate the overhead problem, especially in higher data rates which are supported by the forthcoming 802.11n. Data rates of nearly 432 Mbps tend to have 10% of MAC efficiency [14].

The Block-ACK feature allows a number of data units to be transmitted and afterwards the sender sends a Block-ACK request (BAR) and receives a Block-ACK (BA) frame. Throughput is increased since less ACK frames are used for a transmission. Analysis of the Block-ACK scheme (BTA) is not within the scope of the paper and more information can be found in the standard [2]. The problem with errors in the BTA scheme is similar to the RTS/CTS and requires to change all of the above equations which include errors in RTS and CTS frames with errors in BAR and BA frames, and to make all the respective errors of ACK equal to zero. However since the errors are uniformly distributed, the probability of error in one of these packets is equal. Finally

$$S_i' = \frac{(1 - p_{e,i})\cdot P_{s,i}'\cdot F\cdot E[L]}{P_{\text{idle}}\sigma + \sum_{i=0}^{3}(1 - p_{e,i})P_{s,i}'T_{s,i} + (Pc + p_{e,i}P_{s,i})}. \tag{40}$$

The time for successful transmission $T_{s,i}$ is thus much bigger since it includes F frames and SIFS time, plus the exchange of the BAR and BA. Moreover H is the physical layer header and δ is the transmission delay:

$$\begin{aligned} T_{s,i}^{\text{basic}} &= T_{E,i} = F\cdot(H + E[L] + \text{SIFS} + \delta) + \text{AIFS}[i] + H \\ &\quad + T_{\text{BAR}} + \text{SIFS} + \delta + H + T_{\text{BA}} + \delta, \\ T_{c,i} &= F\cdot(H + E[L] + \text{SIFS} + \delta) + \text{EIFS}[i] + H + T_{\text{BAR}} + \delta, \end{aligned} \tag{41}$$

where $\text{EIFS}[i] = \text{SIFS} + H + T_{\text{BA}} + \text{AIFS}[i]$.

3.2. MAC delay

3.2.1. Mean value of the MAC delay

In 802.11e [2], two different access mechanisms are provided. The first one is the use of acknowledgments by ACKs (called here "basic") and the other by transmitting request-to-send and clear-to-send packets. The transmission times $T_{s,i}^{\text{Basic}}$ and $T_{s,i}^{\text{RTS/CTS}}$, and the times $T_{c,i}^{\text{Basic}}$ and $T_{c,i}^{\text{RTS/CTS}}$ for a collision can be found in [6].

The mean delay can be defined for each AC by the following equation:

$$\begin{aligned} E[D]_i &= E[N_{cs}]_i(E[\text{BD}]_i + T_c + T_{\text{protect}}) \\ &\quad + E[\text{BD}]_i + T_{s,i}. \end{aligned} \tag{42}$$

The first part of the equation is the delay due to consecutive unsuccessful transmissions, the second part is the mean backoff delay, whenever this transmission will be completed, and the third part is the transmission duration. All are referred to each AC. Following (42), $E[N_{cs}]_i$ can be defined as the mean number of collisions that are followed by a successful transmission:

$$E[N_{cs}]_i = \frac{1 - P_{\text{idle}} - P_{s,i}}{P_{s,i}}. \tag{43}$$

3.2.2. PMF of the MAC delay

Having supposed that the standard refers to an integer number of time slots, then the Z-transform can be used to calculate the delay. Z-transform is well used in the analysis of queuing systems, because its derivatives can provide measures such as mean values, variances, and some other possible moments of the PMF of the MAC delay.

In order to include the freezing of the backoff counters, a distinction has been made in (24) and (25).

The probability that the tag station is interrupted by the transmission of a single station (one exactly) is given by

$$p_i' = \binom{N_i - 1}{1} \cdot \tau_i \cdot (1-\tau_i)^{N_i - 2} \prod_{z>i} (1-\tau_z)^{N_z}. \qquad (44)$$

The phenomenon that the slot is interrupted from a collision or a successful transmission is described by

$$\begin{aligned} P(\text{collision} \mid \text{slot is interrupted}) = p_{c,i} &= \frac{p_i - p_i'}{p_i}, \\ P(\text{successful by one AC} \mid \text{slot is interrupted}) = p_{t,i} &= \frac{p_i'}{p_i}. \end{aligned} \qquad (45)$$

In our case, each state of backoff duration is said to have a delay $\mathrm{SD}_i(z)$. In order to count down to the next state, the slot must remain idle, which is symbolized by the duration of the empty in Z-transform multiplied by the probability of the slot to be idle, $P_{\text{idle}}Z^{\sigma}$. Hence the Z-transform of that delay is

$$\mathrm{SD}_i(z) = \frac{P_{\text{idle}} Z^{\sigma}}{1 - p_i \cdot \left(p_{t,i} Z^{T_{s,i}} + p_{c,i} Z^{T_{c,i}}\right)}. \qquad (46)$$

Then the total delay of backoff duration is given from the geometric sum, since its state is chosen uniformly. Note that the first queue of each AC is smaller since the first state is not chosen:

$$\mathrm{BD}_{i,j}(z) = \begin{cases} \displaystyle\sum_{k=0}^{CW_{i,j}-1} \frac{\mathrm{SD}_i^k(z)}{CW_{i,j}}, & 0 \le j \le m_i, \\ \mathrm{BD}_{i,m_i}(z), & m_i \le j \le L_i. \end{cases} \qquad (47)$$

In the previous subsection, we have shown a unified method to find the mean MAC delay for all the models. In the following subsection, the above metrics for MAC delay will correspond only to the model 3. This happens because the solution of the DTMC after theoretically infinite retries gives mean values. Thus the Z-transform of the MAC delay will be given as a function of $D_i(z)$:

$$\begin{aligned} D_i(z) = (1-p_i) z^{T_{s,i}} \sum_{j=0}^{L_i} \left[\left(p_i z^{T_c}\right)^j \prod_{f=0}^{j} \mathrm{BD}_{i,f}(z) \right] \\ + \left(p_i z^{T_c}\right)^{L_i+1} \prod_{f=0}^{L_i} \mathrm{BD}_{i,f}(z). \end{aligned} \qquad (48)$$

The first part signifies the correct transmission ($(1 - p_i)z^{T_{s,i}}$) having encountered a number of collisions in the previous stages, whereas the second part is the delay associated with dropping of a packet after $L_i + 1$ retries. However, to find the mean value and the standard deviation (SD), the 1st and the 2nd moments of the above equations must be found, respectively, and we need

$$\begin{aligned} E[D]_i &= \left.\frac{\partial D_i(z)}{\partial z}\right|_{z=1}, \\ \mathrm{Var}^2[D]_i &= \left.\frac{\partial^2 D_i(z)}{\partial z^2}\right|_{z=1} + \left.\frac{\partial D_i(z)}{\partial z}\right|_{z=1} - \left\{ \left.\frac{\partial D_i(z)}{\partial z}\right|_{z=1} \right\}^2. \end{aligned} \qquad (49)$$

The last part is to find MAC delay distribution. It is well know that every Z-transform of the PGF can be written as

$$D_i(z) = \sum_{k=0}^{\infty} d_{i,k} z^k. \qquad (50)$$

It seems that from the definition $d_{i,k}$ is the inverse Z-transform of the $D_i(z)$. A method that gives the inverse Z-transform with a predefined error bound is the Lattice-Poisson algorithm [16], which is valid for $|d_{i,k}| \le 1$. Thus the PMF (probability mass function) of the MAC delay is given by

$$dg_{i,k} = \frac{1}{2kr^k} \sum_{h=1}^{2k} (-1)^h \mathrm{Re}\left(DG_i\left(re^{i\pi j/k}\right)\right), \qquad (51)$$

where $DG_i(z)$ is the generating function

$$DG_i(z) = \sum_{k=0}^{\infty} dg_{i,k} z^k = \frac{1 - D_i(z)}{1 - z}. \qquad (52)$$

The following values can be used: $r = -\gamma/(2 * k)$ and $\gamma = 1$.

4. COMPARISON ANALYSIS AND EVALUATION

For validating the correctness of the mathematical analyses, OPNET modeler (version 12) was used with the EDCA simulation model incorporated. For further validation, we also compare the Xiao model [6] which is slightly alternated for reasons of fair comparison. In Table 2, the simulation parameters are summarized. The accuracy values of the figures are explained in Section 4.2.

In Figure 5, saturation throughput comparison is presented for the three models, the Xiao model, and the simulations. The throughput of AC0 is twice the throughput of AC1, which is derived from the half CWmin and CWmax values (see Table 1). It seems that most models have results close to the simulation values, with the DTMC model being the most accurate one. In fact this happens due to the ability of the DTMC to capture possible parameters of the EDCA scheme of 802.11b/e, in each time slot, rather than average values of the other models.

Similar results can be found for the RTS/CTS access method in Figure 6. However one major difference between these methods, which cannot be derived from basic rate analyses (1 Mbps), is that, as the number of nodes (load) increases, RTS/CTS can solve, except from the hidden node terminal, part of the high collision rate. Thus a hybrid system

Table 2: Simulations parameters.

Simulator	OPNET modeler 12
Model	EDCA model august (2006)
Standards	802.11b/e and 802.11a/e
Packet size	Constant 1023 bytes (no segm.)
Interarrival time	Set according to saturation conditions
RTS threshold	512 bytes
Simulation	100 s
Initial seed	128
Space	Square 100 × 100 (single hop)
Mode	No AP functionality (ad hoc)

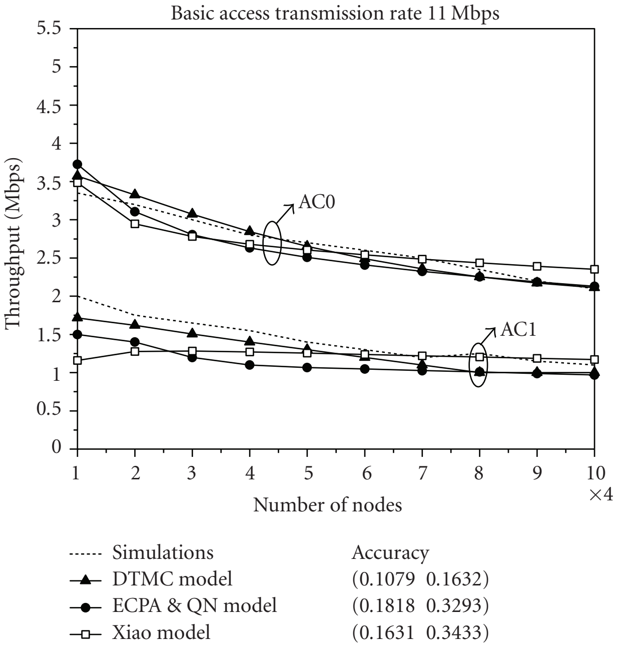


Figure 5: Model comparison in terms of saturation throughput using basic access method at 11 Mbps transmission rate.

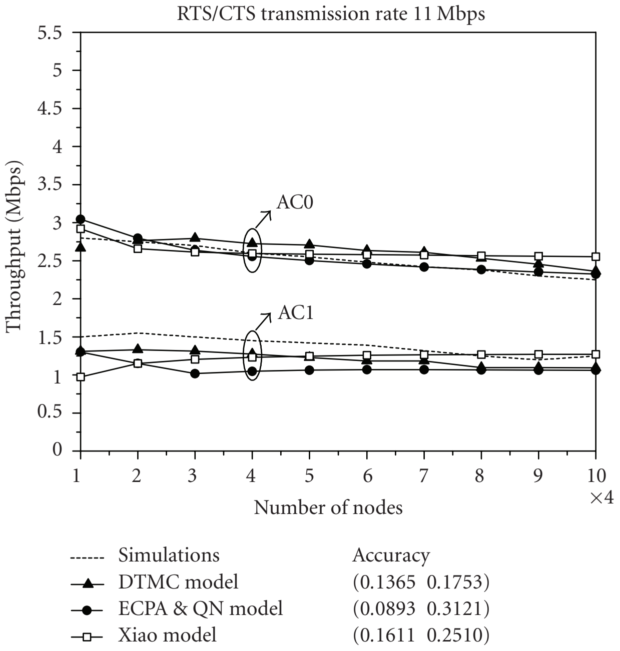


Figure 6: Model comparison in terms of saturation throughput using RTS/CTS access method at 11 Mbps transmission rate.

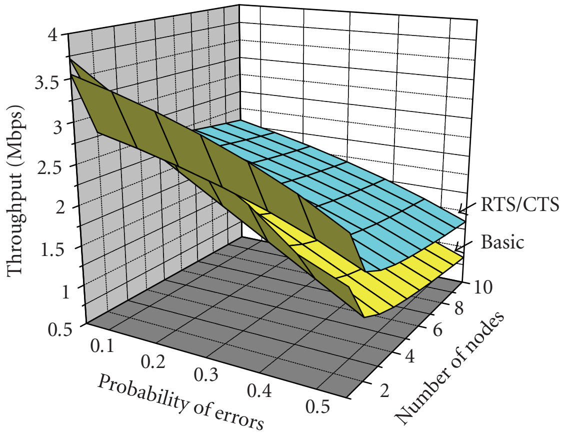


Figure 7: Saturation throughput at 11 Mbps, as a function of probability of error and number of stations for AC0.

that would change the transmission process, according to the load, would provide higher performance of the standard.

In Figure 7, throughput is shown as a function of the number of nodes and probability of error, thus providing a 3D graph. The probability of error is a derivation of cross layer architectures and probabilistic nature of the channel. It is worth seeing that in RTS/CTS method, the degradation of throughput stays in very low values, and shows that RTS/CTS transmission can be a solution in erroneous environments as well. The comparison of the three models in these graphs is avoided since the degradation of the performance due to errors seems to be linear to probability of errors.

According to Figures 8 and 9, the performance analysis shows that our DTMC remains equally accurate as in throughput metric. Xiao's model tends to diverge significantly from the simulation results whereas the elementary probabilistic and queuing models seem to have a good accuracy. Moreover from the ECPA and queuing model, the first derivative (signified as FD in the figures) of the Z-transform PGF of the MAC delay, shows the mean value. Such a mean value seems to be more accurate than the mean values provided by the other models, because it differentiates the freezing probability in collision or busy channel due to a transmission. Similarly, taking the second derivative of the PGF of the Z-transform, the variance is found as shown in Figure 10. Variance of the MAC delay is a significant metric, since it signifies the jitter of the multimedia traffic transmitted over the standard using 11 Mbps bandwidth.

In Figure 12, IEEE 802.11a/e is modeled with a bandwidth of 24 Mbps (analytical and simulation set) and as it is seen when enabled, the Block-ACK mechanism can offer higher throughput in higher load and can even provide

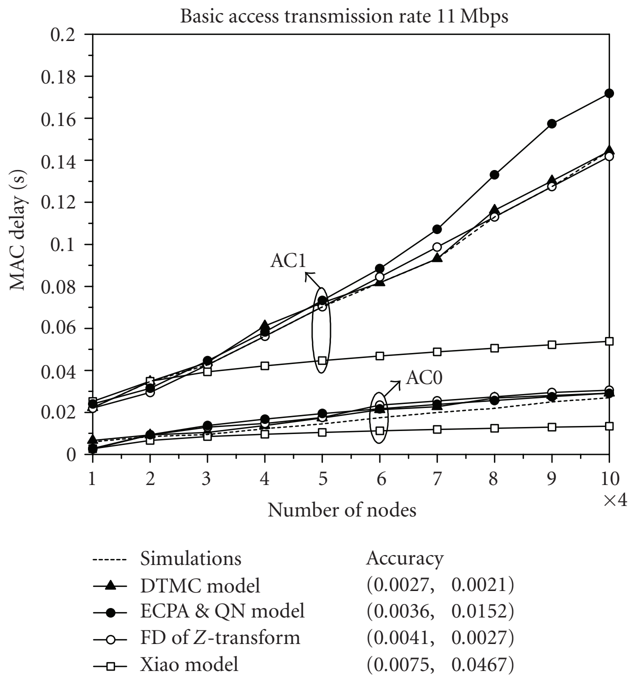


Figure 8: Analysis of MAC delay of the two higher ACs under saturation condition and basic access mode.

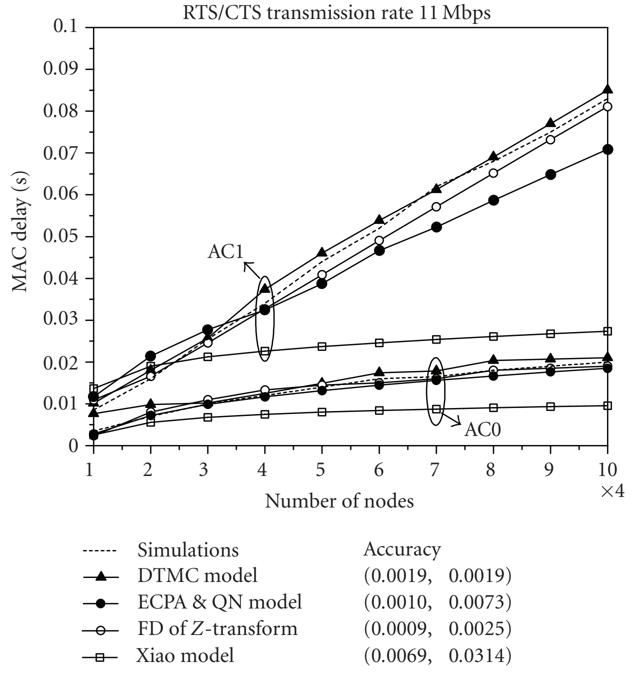


Figure 9: Analysis of MAC delay of the two higher ACs under saturation condition and RTS/CTS access mode.

better results in higher bandwidth occasions. This is due to the reduction of unnecessary ACKs. The reason for modeling IEEE 802.11a is that higher bandwidths are going to be used in 802.11n with Physical (PHY) and MAC layers that do not change significantly. (In Figure 12, accuracy measurements are not taken because the graph is not provided for depiction of Block-ACK—Section 3.1.2.)

Table 3: Big-O notation for computational complexity of the compared models.

	τ	E [BD]	Both τ and E [BD]
DTMC	Com1	Com2	Com2
ECPA	Com3	Com3	Com3
Queuing	Com3	Com3	Com3

4.1. Complexity analysis

Complexity is an important characteristic as regards mathematical analysis and algorithms. Comparing the three approaches in terms of complexity allows for an insight in to the usability and scalability of each one.

The DTMC model is obviously the most complex one. This is due to the independence of each state, which models a state of the BEB, and the correlation with the state of the previous slot. However the state of the previous slot is hardly incorporated in models based on queuing theory or geometric distribution since it does not allow the flexibility to change backoff duration according to the simulation needs. A significant drawback of the proposed DTMC is that nonsaturation throughput analysis becomes a complex problem, whereas in the other analyses, the arrival rate could be changed very easily with simple algebra. On the other hand, the modeling of independent states makes easier to provide amendments in the analysis, such as the one given with the inexistence of the first state. Thus it is easy to observe that the analysis of this first model requires big DTMC and more mathematical formulas to be calculated.

Moreover, the addition of extra features and the incorporation of realistic modeling in this approach inject even more complexity in the final calculations. Apart from this heuristic approach, a computational complexity comparison can be performed in terms of big-O notation. Instead of computer instructions, we use a simple formula calculation as the basic unit of complexity. Each algorithm's order of complexity can be estimated as a function of the number of calculation points N, the number of steps used in the fixed point iteration method M, the retry limits L_i, and the number of ACs calculated i. In Table 3, the results show that all three algorithms have linear complexity relative to M and N, and that the DTMC model is approximately four times more complex than the other two approaches:

$$\begin{aligned}
\text{Com1} &= O\left[NM\left(\sum_i (4L_i+1)\right)\right],\\
\text{Com2} &= O\left[N\left(M\left(\sum_i (4L_i+1)\right)+\sum_i L_i\right)\right],\\
\text{Com3} &= O\left[NM\left(\sum_i (L_i+3)\right)\right].
\end{aligned} \tag{53}$$

Ordinary values for the parameters are: $N = 10$, $M = 20$, $L = 7$, and $i = 4$.

Table 4: Metamodel comparison.

	DTMC	ECPA	Queue
Accuracy	High	Medium	Medium
Flexibility	High	Medium	Medium
Complexity	High	Low	Low
Nonsat	Low	Medium	Low
Depiction	High	Low	Medium

4.2. *Accuracy analysis*

In order to prove the accuracy of each model, we have used the L^2 distance, which is defined as the distance

$$d\left(m_{\mathrm{sim},N_i}(\min q), m_{\mathrm{anal},N_i}\right) = \sqrt{\frac{1}{FN}\sum_{N_i=0}^{FN}\left(m_{\mathrm{sim},N_i}(\min q) - m_{\mathrm{anal},N_i}\right)^2}, \tag{54}$$

where $m_{\mathrm{anal},Ni}$ is the analytical metric (throughput or delay) of each model, $m_{\mathrm{sim},Ni}$ $(\min q\%)$ is the simulation of each metric (we mention it as a function of $(\min q\%)$ which is the closest % quartile—in our case mean values of simulation have been used), both as functions of N_i, which is the number of nodes of each AC.

FN is said to be the maximum number of nodes of each AC, defined for case $FN = 10$. Thus for the comparison analysis, we have the distances, as written in Figures 5, 6, 9, and 10. It is readily seen that the DTMC model is more accurate than the other two models, since it can capture the freezing of the backoff counter, and the effect of the previous slot to the current one. In addition, the first derivative of the MAC delay diverges the freezing probability to busy period due to a transmission or collision, and gives the best possible accuracy.

5. DISCUSSION AND CONCLUSION

The purpose of this work is threefold. The first goal is to present a comparison analysis of the most known analyses in order to find the best method to numerically analyze the standard, while setting the pace for future methods of analysis. Furthermore the modeling techniques, based on complexity and accuracy theory, have not been studied before in wireless networks and they could be a field of great interest, since the computer resources are finite.

The second goal was to extend the already known analyses, introducing features that change considerably the performance analysis. The combination of such features optimizes the MAC protocol outcomes, and makes each mathematical analysis avant-garde by itself.

The third goal was to correct the IEEE 802.11e from general misunderstandings, such as the phenomenon of not providing instant access after a successful transmission and the freezing of backoff counter.

From the evaluation part of the proposed work, the performance (of general BEB algorithms—numerous standards implement BEB) and simulation conclusions are as follows.

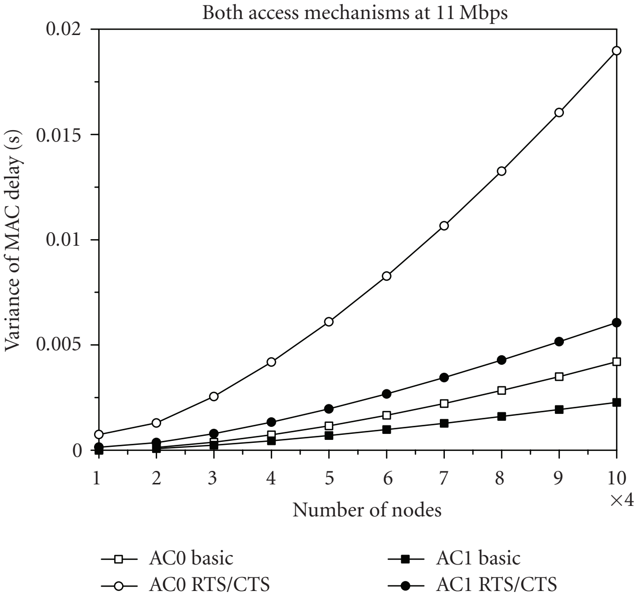


Figure 10: Jitter (second derivative of Z-trasform) is shown for the AC0 and AC1, and for variable number of stations in each AC. The transmission rate is at 11 Mbps 802.11b/e.

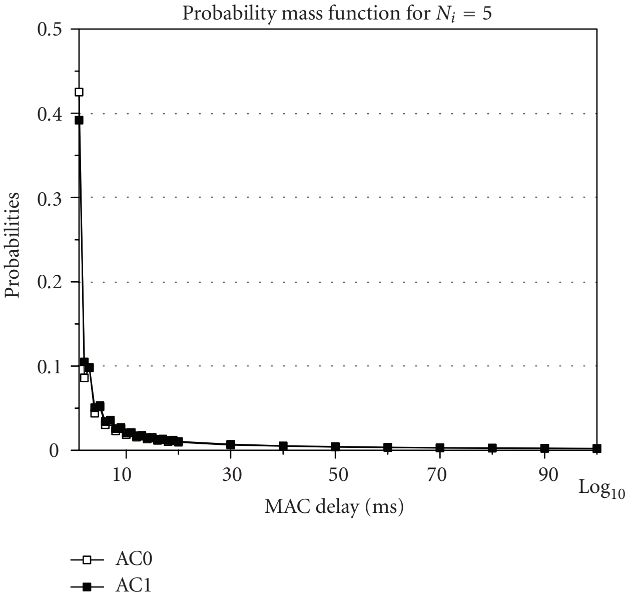


Figure 11: Probability mass function (PMF) for basic access mode and for both ACs. The transmission rate is at 11 Mbps 802.11b/e and the graphs are for 5 numbers of states at each AC[i].

(1) Performance conclusions

(i) In order to have better performance in exponential backoff algorithms, methods that provide fairness of transmission (such as RTS/CTS), apart from aiding in the hidden terminal problem, can reduce the collision rate and improve the performance in erroneous environment. Thus it could be used in WiFi implementations over rural areas.

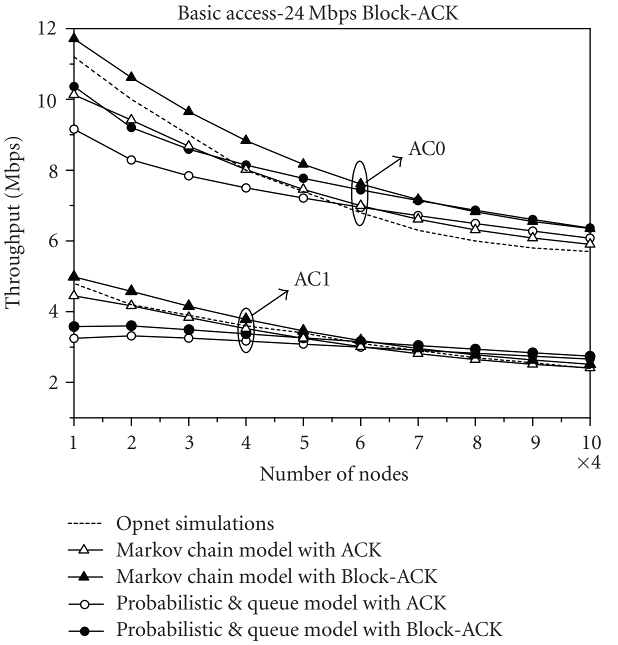


FIGURE 12: Model comparison of saturation throughput with Block-ACK being enabled (F = 64) in 24 Mbps with IEEE 802.11a.

(ii) RTS/CTS and similar access methods should be used only in higher loads, because in lower ones they tend to diminish the throughput performance.

(iii) Analogies on initial windows (e.g., CWmin) in BEB algorithms can give the similar analogies on throughput performance.

(iv) Methods that group packets (e.g., Block-ACK or TxOP) can augment the throughput performance.

(2) Simulation conclusions

As regards the *accuracy*, the DTMC model offers better results, owing to the fact that more EDCA characteristics can be included. This leads to another advantage of the DTMC model which is its *flexibility*. The modeling of each independent state allows for extreme detail in modeling each specific characteristic of the MAC protocol, such as the absence of the first state and the correlation of each state with the previous slot. Moreover the DTMC can also be used as a *depiction of the states* of the MAC protocol.

On the other hand, the other two models demonstrate different advantages. They lead to approximate results bearing less *complexity* compared to the DTMC model, both having the same accuracy and small differences in complexity. Moreover they allow for *nonsaturation conditions* of traffic, whereas in the DTMC model case, this can prove to be a very complex issue.

The Z-transform is used as a method to calculate the accurate delay distribution, while having different types of queues, depicting the heterogeneity of the multimedia applications. Even if such a combination of features and corrections increase, the complexity of the proposed analyses, it is the price that must be paid for the improved accuracy and realism, especially when new machines can solve problems in acceptable time lengths. OPNET modeler 12, was used in two ways: first to verify the correctness of analyses and second as a tool to calculate the accuracy analysis. In Table 4, the above are summarized. Fading channels can be used in order to optimize MAC layer metrics through cross-layer techniques in high-mobility scenarios.

REFERENCES

[1] “Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification,” IEEE, August 1999.

[2] “Medium access control (MAC) enhancements for quality of service (QoS),” IEEE, July 2005.

[3] I. Politis, M. Tsagkaropoulos, S. Kotsopoulos, T. Dagiuklas, and P. Stavroulakis, “On the QoS assesessment of video sessions in heterogeneous 3G-WLANS networks with seamless and secure mobility support,” *Journal of China Communications*, vol. 4, no. 1, 2007, special issue on Communications and Information Security.

[4] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[5] C. H. Foh and J. W. Tantra, “Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters,” *IEEE Communications Letters*, vol. 9, no. 2, pp. 130–132, 2005.

[6] Y. Xiao, “Performance analysis of IEEE 802.11e EDCF under saturation condition,” in *Proceedings of the IEEE International Conference on Communications (ICC ’04)*, vol. 1, pp. 170–174, Paris, France, June 2004.

[7] P. E. Engelstad and O. N. Østerbø, “Delay and throughput analysis of IEEE 802.11e EDCA with starvation prediction,” in *Proceedings of the 30th Anniversary IEEE Conference on Local Computer Networks (LCN ’05)*, pp. 647–655, Sydney, Australia, November 2005.

[8] G. S. Paschos, I. Papapanagiotou, S. A. Kotsopoulos, and G. K. Karagiannidis, “A new MAC protocol with pseudo-TDMA behavior for supporting quality of service in 802.11 wireless LANs,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, Article ID 65836, 9 pages, 2006.

[9] Y. C. Tay and K. C. Chua, “A capacity analysis for the IEEE 802.11 MAC protocol,” *Wireless Networks*, vol. 7, no. 2, pp. 159–171, 2001.

[10] G. Bianchi and I. Tinnirello, “Remarks on IEEE 802.11 DCF performance analysis,” *IEEE Communications Letters*, vol. 9, no. 8, pp. 765–767, 2005.

[11] J. Choi, J. Yoo, and C.-K. Kim, “A novel performance analysis model for an IEEE 802.11 wireless LAN,” *IEEE Communications Letters*, vol. 10, no. 5, pp. 335–337, 2006.

[12] J. Hui and M. Devetsikiotis, “A unified model for the performance analysis of IEEE 802.11e EDCA,” *IEEE Transactions on Communications*, vol. 53, no. 9, pp. 1498–1510, 2005.

[13] Q. Ni, T. Li, T. Turletti, and Y. Xiao, “Saturation throughput analysis of error-prone 802.11 wireless networks,” *Wireless Communications and Mobile Computing*, vol. 5, no. 8, pp. 945–956, 2005.

[14] T. Li, Q. Ni, T. Turletti, and Y. Xiao, “Performance analysis of the IEEE 802.11e block ACK scheme in a noisy channel,” in *Proceedings of the 2nd International Conference on Broadband Networks (BROADNETS ’05)*, vol. 1, pp. 551–557, Boston, Mass, USA, October 2005.

[15] A. Banchs and L. Vollero, “A delay model for IEEE 802.11e EDCA,” *IEEE Communications Letters*, vol. 9, no. 6, pp. 508–510, 2005.

[16] J. Abate and W. Whitt, “Numerical inversion of probability generating functions,” *Operations Research Letters*, vol. 12, no. 4, pp. 245–251, 1992.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 71458, 8 pages
doi:10.1155/2007/71458

Research Article

Video Broadcasting Using Queue Proportional Scheduling

Dimitris Toumpakaris and Stavros Kotsopoulos

Wireless Telecommunications Laboratory, Department of Electrical and Computer Engineering, University of Patras, 26500 Rio, Greece

Received 31 May 2007; Accepted 12 September 2007

Recommended by Tasos Dagiuklas

Queue Proportional Scheduling (QPS) has been shown to be throughput optimal for Gaussian Broadcast Channels. This paper examines the use of QPS for Video Broadcasting. First, the behavior of QPS is examined as the scheduling frequency is reduced and a method is proposed that uses statistics on the arrival rates to improve its performance. The reduction of the scheduling frequency simplifies the scheduler and decreases the required operations. Then, the packet delay variation is modeled using a Markov Chain approach leading to a method for approximating the packet delay distribution. Based on the resulting distribution, it is discussed how the video encoding rate can be chosen in order to reduce the expected distortion of streams transmitted through Broadcast Channels.

1. INTRODUCTION

Wireless systems have been experiencing constant growth and increased popularity during the past decade. Cellular telephones are now part of most people's everyday life. Fueled by their success and the increased appetite of customers for new and improved services, next-generation cellular systems are targeting broadband applications such as data transfers and video streaming. The aim is to provide mobile users with high rates and seamless roaming [1–3]. When the users are stationary, even higher rates can be offered [4].

One of the services that is expected to gain popularity over the following years is video broadcasting. Digital Video Broadcasting systems will eventually replace analog transmission. In addition to DVB services, video-on-demand download services will be offered to mobile phone or computer network users. Therefore, a base station that is serving a cell will have to broadcast different video streams to the mobile users of the cell. Cellular system downlinks are typical examples of Broadcast Channels (BCs) [5] where a single transmitter sends data to more than one receivers. A well-known information theoretic result is that the attainable rate vectors in a Gaussian BC form a capacity region that can be achieved using superposition coding at the transmitter and successive interference cancellation (SIC) at each receiver. The performance of practical systems often deviates from the optimal bound. For example, TDMA systems use time division that is sub-optimal, in general, whereas CDMA systems use superposition, but do not use SIC at the receiver where all other users are treated as noise. In this paper it is assumed that the optimal BC performance is achieved by the transceiver architecture. If this is not the case the loss in performance can be taken into account using a nonzero gap value. It is also assumed that both the transmitter and the receiver have perfect Channel State Information (CSI). This is done because the focus of this study is not on how to achieve the Gaussian BC capacity, but on how to manage the available resources in a BC in order to deliver video to the mobile users.

The capacity region of the Broadcast Channel is the union of the rate vectors that can be achieved assuming that the traffic is regular, that is, that during each time period the number of bits transmitted to a user at the physical layer is the same as the number of bits that are sent for transmission by the link layer. However, in practice, the physical layer of a communications system may be receiving data in bursts. For example, if the wireless link is the last hop of a TCP link, the packets may be arriving at irregular intervals at the transmitter due to delays along the data pipe, different routings, packet losses, and so forth. In this case the system may become unstable and the lengths of the queues of some users may not be bounded even if the average rate of each user lies inside the BC capacity region. A significant amount of

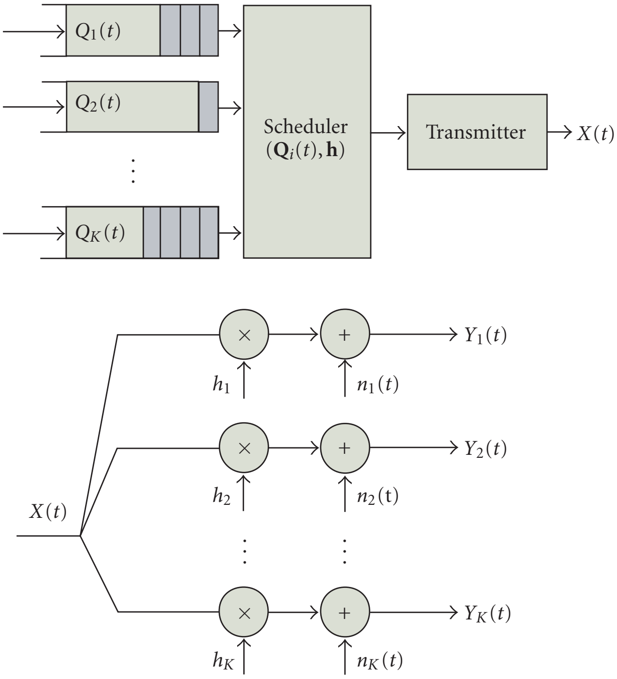


Figure 1: System model.

research effort has been devoted to the problem of achieving the capacity region when the incoming traffic is random. Luckily, it turns out that the set of all arrival rate vectors for which it is possible to keep each queue length finite, referred to as the network capacity region, is the same as the BC capacity region. The scheduling policies that achieve the network capacity region are called throughput optimal. Several throughput optimal policies have been proposed for the BC [6–8]. Their common characteristic is that they rely not only on CSI, but also use Queue State Information (QSI). Therefore, they are cross-layer approaches.

Among the throughput optimal cross-layer approaches, Queue Proportional Scheduling (QPS) [7–9] has been shown to have very desirable delay properties. Although its delay optimality for Gaussian Broadcast Channels has not been proved to date, it results in the smallest average packet delay among the known throughput-optimal algorithms, thus making it a good candidate for video transmission where large delays may lead to packet losses, and, consequently, distortion. As will be explained in more detail in Section 2, QPS allocates resources in the BC based on the channel state as well as the queue lengths. In this paper, a simplified version of QPS is proposed that uses Queue State Information less frequently in order to reduce the computational burden. This way the scheduler becomes simpler, since it does not require access to the queue during each scheduling period. It is shown that, under some conditions on the average arrival rate, the modified algorithm is throughput optimal. However, as is expected from the fact that less information is used, it exhibits performance degradation compared to QPS with continuous use of QSI. This is verified using simulation. Then, the packet delay is modeled using Markov Chains. More specifically, a Markov Chain model is fitted to simulation data and is then used to approximate the probability distribution of the delay of the packets. It is shown that, although the service rate depends on the queue size as well as on the states of the other queues, the approximation is satisfactory. Using information on the expected delay and the corresponding distortion it is possible to choose the video encoder rate in a system employing QPS in order to control the quality of video that is delivered to the users of a BC.

This paper is organized as follows. Section 2 examines the degradation of the performance of QPS as the frequency of using QSI for scheduling decreases and proposes a modification that reduces the performance gap. It is also shown that the modified scheme is throughput optimal under a condition on the average arrival rates. In Section 3 the packet delay is modeled using a Markov Chain model leading to a method that approximates the delay distribution. Section 4 discusses how the distribution of the packet delays can be used to predict the video distortion corresponding to a given encoder rate leading to a discussion on choosing the encoder rate for video streams that are sent to users of a Broadcast Channel. Finally, Section 5 contains concluding remarks.

2. QPS WITH LESS FREQUENT USE OF QUEUE STATE INFORMATION IN GAUSSIAN BROADCAST CHANNELS

Figure 1 depicts the system model that is used in this article. Packets arrive randomly to each queue and are scheduled for transmission. The scheduler allocates the resources of the BC using information on the channel taps h_i (CSI) and the queue states $Q_i(t)$ (QSI). In this article, the scheduler uses Queue State Information only periodically. Moreover, the channel taps are assumed to be constant.

The output signal $X(t)$ is broadcast to the channel, and the signal at each receiver i is equal to

$$Y_i(t) = h_i X(t) + n_i(t), \quad i = 1, \ldots, K. \tag{1}$$

This paper assumes a Gaussian BC, that is, the $n_i(t)$ are i.i.d. zero-mean Gaussian random variables with double-sided power spectral density equal to $\mathcal{N}_0/2$. The capacity region of a Gaussian BC in bits/s, assuming, without loss of generality, that $|h_i|^2 \leq |h_j|^2$ for $i < j$, and that the available bandwidth is equal to $2W$ is given by [5]

$$C_{\mathrm{BC}} = \left\{ R_i : R_i \leq W \log\left(1 + \frac{\alpha_i |h_i|^2 P}{\mathcal{N}_0 W + \sum_{j>i} \alpha_j |h_i|^2 P}\right)\right\}, \tag{2}$$

where P is the average power of $X(t)$ and the $\alpha_i \geq 0$ trace the whole simplex, that is, $\sum_i \alpha_i = 1$. The capacity region is achieved by superposition coding at the transmitter and by successive interference cancellation at each receiver. When the traffic is regular, the transmitter can accommodate any rate vector $\mathbf{R}$ that is inside the capacity region (2). In the following, $\mathbf{R}$ is the number of bits transmitted during the scheduling interval (that is assumed to be equal to 1 for simplicity), so, it is expressed in bits and not in bits/sec.

In this paper it is assumed that traffic arrives irregularly and in packets. The number of packets $A_i(t)$ arriving at queue i during a time period is a Poisson process with rate λ_i, and arrivals at each queue are independent. The packet lengths M_i in bits are assumed to be i.i.d. exponentially distributed with $E[X_i] = \mu_i$ and independent of $A_i(t)$. Therefore, the arrival rate of bits is $\lambda_i \mu_i$. Infinite-capacity queues are considered. At the end of each interval, the scheduler decides on the rate R_i of each queue based on the channel gains $\mathbf{h}$ and the vector of bits (or packets) $\mathbf{Q}(t)$. Other than that, no knowledge of the statistics of the arrival process is required as long as the arrival rate lies inside the capacity region C_{BC}. In this article, $\mathbf{Q}(t)$ may not be used by the scheduler during some scheduling periods as is explained in more detail in this section. The resulting queue state after transmission is

$$Q_i(t + T_S) = (Q_i(t) - R_i(t))^+ + Z_i(t), \tag{3}$$

where $a^+ = \max\{a, 0\}$, and $Z_i(t)$ is the number of bits arriving at queue i during one scheduling period T_s.

Queue Proportional Scheduling (QPS) calculates the rate vector $\mathbf{R}(t) = [\, R_1(t) \quad R_2(t) \quad \cdots \quad R_K(t) \,]$ according to

$$\begin{aligned} \mathbf{R}(t) &= \max_x \{x\mathbf{Q}(t)\} \\ &\text{subject to} \begin{cases} x\mathbf{Q}(t) \in C_{\mathrm{BC}} \\ x \le 1. \end{cases} \end{aligned} \tag{4}$$

Therefore, $\mathbf{R}(t)$ is a scaled version of $\mathbf{Q}(t)$. If $\mathbf{Q}(t)$ is inside the capacity region, $\mathbf{R}(t) = \mathbf{Q}(t)$, else $\mathbf{R}(t)$ is the intersection of the ray $x\mathbf{Q}(t)$ and the boundary of C_{BC}. In [8] the bit-based QPS is considered, and it is shown that $\mathbf{R}(t)$ is the solution of a Geometric Program and is therefore globally optimal. Note that, since $\mathbf{R}(t) \le \mathbf{Q}(t)$ when QPS is used, (3) can be rewritten as

$$Q_i(t + T_s) = Q_i(t) - R_i(t) + Z_i(t). \tag{5}$$

In [8] the bandwidth W, the scheduling period T_s and the average packet length for all queues are set to 1. In this article the scheduling period remains equal to 1, but Queue State Information is only used once every L scheduling periods. Naturally, if, during scheduling period t, the information on $\mathbf{Q}(t - L)$ is used, the scheduling will not be done based on the current needs of the user corresponding to each queue. It is expected (and verified by simulation) that this will lead to larger fluctuations of the service rates, and, consequently, larger average queue sizes and packet delays. However, if the scheduler knows the average arrival rate of each queue it can approximate the queue size $\mathbf{Q}(t)$ by $\mathbf{Q}(t - L) + \boldsymbol{\lambda} L - \sum_{l=0}^{L-1} \mathbf{R}(t - l)$. From this point on, $\boldsymbol{\lambda}$ is the bit arrival rate, that is, the product of the packet arrival rate and the average packet size. This is done for simplicity and for compatibility with the notation used in some of the references. Although not as accurate as the actual $\mathbf{Q}(t)$, this approximation will, on the average, be better than $\mathbf{Q}(t-L)$. As L grows, that is, as use of QSI becomes less frequent, $\mathbf{Q}(t)$ will be close to $\boldsymbol{\lambda}$, assuming that $\boldsymbol{\lambda} \in C_{\mathrm{BC}}$. Based on the above observations, the following heuristic modification of QPS is proposed in order to reduce the use of QSI for scheduling. Let the Queue State Information be used once every L times. Also, assume that the modified QPS starts operating at time $t = 0$. Then, the rate vector $\mathbf{R}(t)$ is equal to

$$\begin{aligned} \mathbf{R}(t) &= \max_x \{x\mathbf{Q}(t)\} \\ &\text{subject to} \begin{cases} x\mathbf{Q}(t) \in C_{\mathrm{BC}} \\ x \le 1 \end{cases} \end{aligned} \tag{6}$$

for $(t \bmod L) = 0$, and

$$\begin{aligned} \mathbf{R}(t) &= \operatorname{vec}\Big(\min \Big\{ \max_x \{\lambda_i x\}, Q_i(t) \Big\} \Big) \\ &\text{subject to } \boldsymbol{\lambda} x \in C_{\mathrm{BC}} \end{aligned} \tag{7}$$

otherwise, where $\operatorname{vec}(x_i) = [\, x_1 \quad x_2 \quad \cdots \quad x_K \,]^T$ the vector with elements x_i. Therefore, the optimization is similar to QPS, with the difference that, for times $nL+l$, $l = 1, 2, \ldots, L-1$ the average arrival rate $\boldsymbol{\lambda}$ is used instead of the actual QSI $\mathbf{Q}(t)$. If the packet arrival rate is constant or changes relatively slowly, that is, if $\boldsymbol{\lambda}$ can be estimated accurately and does not need to be updated often, $\mathbf{R}(t)$ can be precalculated and stored. Therefore, the computational complexity is reduced roughly by a factor of L. However, a practical system will need to update an estimate of $\boldsymbol{\lambda}$, so the reduction in complexity will be less pronounced. Similar to QPS, the rate $\mathbf{R}(t)$ does not exceed $\mathbf{Q}(t)$. This is easily implemented by stopping transmission in a given queue if it becomes empty before the end of a transmission period.

In the following two theorems the throughput optimality of the modified QPS algorithm is established under the condition that the arrival rate $\boldsymbol{\lambda}$ is constant and satisfies a constraint on its distance from the boundary of the capacity region C_{BC}. The proof is constructed using the same approach as in [9]. First, it is shown that $E[\mathbf{Q}(t)]$ becomes proportional to $\boldsymbol{\lambda}$ as $t \to \infty$.

Theorem 1. *Assume that the modified QPS policy is used in a Gaussian BC, and that $\boldsymbol{\lambda}$ is such that $\alpha\boldsymbol{\lambda}$ is at the boundary of C_{BC}. Then, $\alpha < L/(L-1)$, and, as $t \to \infty$, $E[\mathbf{Q}(t)|\mathbf{q}_0] \to w(t)\boldsymbol{\lambda}$, where $\mathbf{q}_0$ is any initial state of the queue and $w(t)$ is a function of time.*

Proof. Given in the appendix. □

Note that, as L increases, the average rate $\boldsymbol{\lambda}$ should be closer to the boundary of C_{BC} for throughput optimality to be guaranteed by the theorem.

Having proved the convergence of $E[\mathbf{Q}(nL)|\mathbf{q}_0]$ to the direction of $\boldsymbol{\lambda}$, throughput optimality is shown along the lines of [9].

Theorem 2. *In a Gaussian BC, the modified QPS policy is throughput optimal, as long as the conditions of Theorem 1 hold.*

Proof. Given in the appendix. □

For the evaluation of the performance of the modified QPS algorithm, a two-user scenario is chosen, similar to the

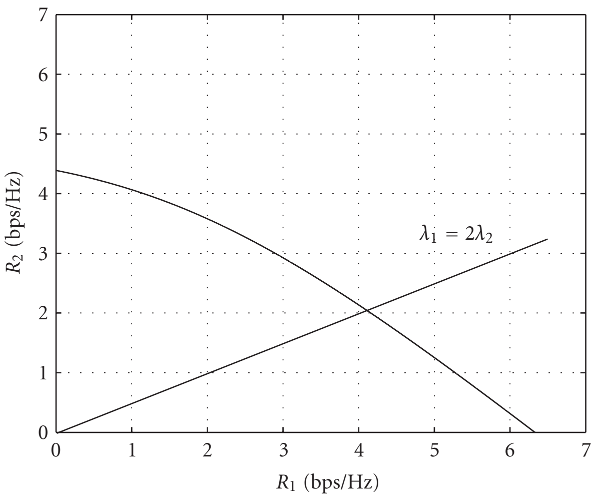
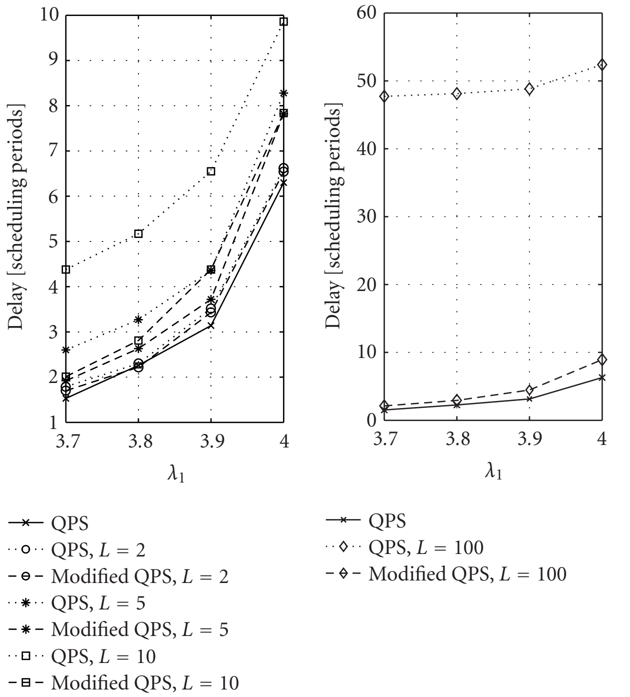


Figure 2: C_{BC} for the 2-user scenario.



(a) (b)

Figure 3: Comparison of performance of QPS, QPS with reduced use of QSI and modified QPS.

one in [9]. The SNR of user 1 is equal to 19 dB, whereas the SNR of user 2 is 13 dB. Moreover, $\lambda_1 = 2\lambda_2$. The capacity region of the Gaussian BC for this scenario is shown in Figure 2. What is also shown is the line $\lambda_1 = 2\lambda_2$. During the periods where QSI is not used, the modified QPS chooses a rate vector along the segment formed by the intersection of the line and the capacity region. Therefore, the maximum average bit rate that can be achieved is equal to $\boldsymbol{\lambda}_{\max} = [\, 4.1 \;\; 2.05 \,]$ bits/s. Figure 3 compares the average packet delay of Queue 1 for different scheduling methods. $\boldsymbol{\lambda}$ is varied from $[\, 3.7 \;\; 1.85 \,]$ to $[\, 4 \;\; 2 \,]$. Due to the nature of QPS, the delays of Queue 2 are similar and their behavior is similar. The dotted lines in Figure 3 depict the degradation of the performance of QPS as the QSI is used less frequently. It is assumed that $\mathbf{Q}(nL)$ is used to compute $\mathbf{R}(nL + l)$, $l = 0, 1, \ldots, L - 1$. The dashed lines correspond to the performance of the modified QPS. The modified QPS obtains an estimate of $\boldsymbol{\lambda}$ by averaging the arrivals during each scheduling period. The performance is evaluated after a sufficient number of iterations of the simulation in order to allow the queues to reach a steady state. Moreover, it is assumed that $\boldsymbol{\lambda}$ does not change during the simulation. A total of 10^5 scheduling periods proved to be satisfactory for simulation. The queue is allowed to converge during the first 10^4 scheduling periods before delay samples are taken.

As can be seen from the figure, as the scheduling based on QSI becomes less frequent, the average packet delay increases for each queue. The modified QPS bridges the gap in performance, especially as L grows. For relatively small values of L use of the modified QPS reduces the average delay by 2 to 3 times compared to the case where $\mathbf{Q}(t - L)$ is used for all L subsequent schedulings. For very infrequent use of QSI ($L = 100$) the improvement is much more pronounced. Note that, for the case of $L = 100$, throughput optimality is not guaranteed by the proofs in this paper, since it only holds for $\lambda_1 > (99/100)4.1 = 4.06$. However, it appears that the modified QPS does not diverge even for average rates less than 4.06.

3. APPROXIMATION OF PACKET DELAY USING MARKOV CHAINS

For video that is transmitted in packet form, what is important is not only the average delay of the packets, but also their delay distribution. More specifically, if a given packet does not arrive within a specific delay window the video decoder may need to decode without using the packet, since either the user cannot tolerate a large delay, or the storage capacity of the receiver buffer will be exceeded. Missing packets result in video quality degradation. Therefore, for the problem of broadcasting examined in this paper, it is useful to be able to obtain the distribution of the packet delay in order to make predictions about the quality of the video stream. A Markov Model is developed in this section whose state denotes the delay of the first (head) packet of the queue in terms of scheduling periods. It is assumed that the only possible transitions are to neighboring states. Again, this is an approximation that is found to work well for QPS.

Clearly, because of queuing, the delays of neighboring packets of a video stream are correlated. During the periods when the queue lengths, and, consequently, the delays become large, it is possible that more than one packet will be delayed. Hence, a model assuming that the delays of neighboring packets of the encoded video stream are independently distributed is not exact unless a sufficient interleaving depth is present. However, in this article it will be assumed that the delays are independent. First, this will provide a lower bound on the video quality that one can expect. Moreover, in order to obtain an accurate estimate of the video

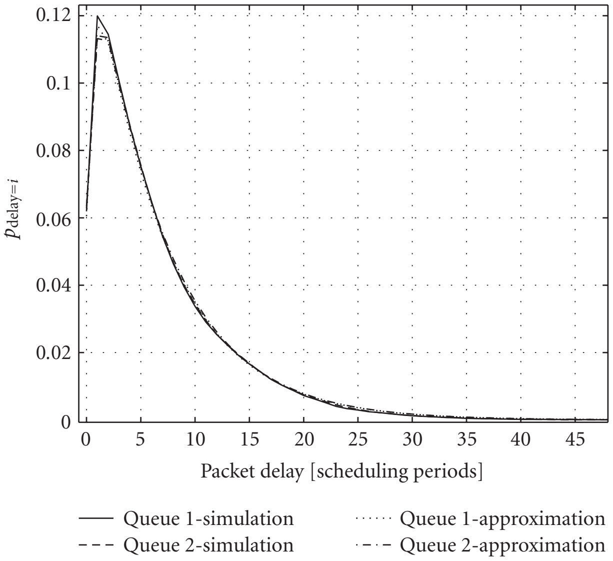


Figure 4: Distribution of the packet delay.

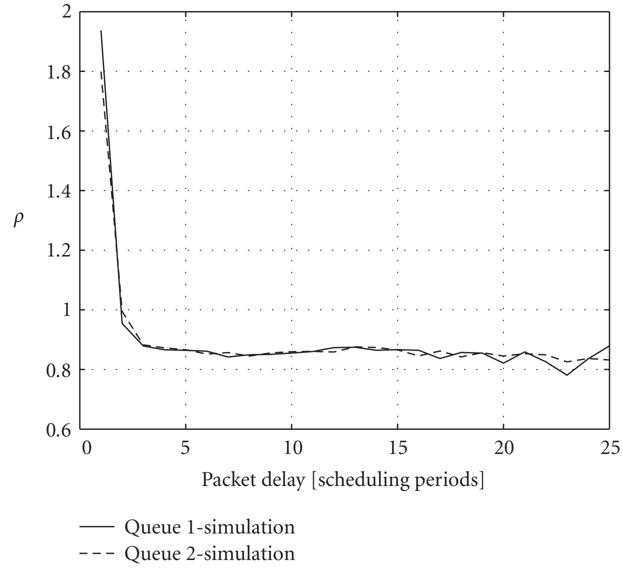


Figure 5: Values of ρ obtained by simulation.

quality, one would need to take into account the particular encoder and decoder that are used, the intra frame ratio, and so forth. One could consider a priority queue where different priorities are given to packets to make sure that enough packets are available for the decoding of a given Group of Blocks (GOB) or frame. Such a scheme will not be accurately described by the Markov Chain presented below, but the approximation may be satisfactory. Another particularity of the system in this article is that the service rate does not depend only on the state of the queue that is being considered, but also on the states of the other queues, since all of them are taken into account for scheduling. However, and despite all the above, simulation results show that the Markov model, albeit simplified, provides a good approximation to the distribution of the delay for a system using a QPS-like scheduler and can therefore be used for the prediction of the video quality.

Figure 4 presents the distribution of the packets delay for the scenario of the previous section, arrival rate $\boldsymbol{\lambda} = [\,4 \quad 2\,]$ and average packet size $\mu = 1$ for both queues. Again, SNR = 19 and 13 dB, respectively. Scheduling uses QSI during all periods. The reason why the probability distribution has a peak at 1 and not 0 is because the scheduler operates only at the end of a period, so packets that have arrived during an interval may have to wait till the end of that interval in order to be able to leave the queue. This skews the peak of the distribution that would otherwise be at 0. In terms of the Markov Chain the service rate is not constant and depends on the queue state.

From Markov Queue theory, and assuming that the service rate depends on the delay of the first packet of the queue,

$$p_{i+1}\mu_{i+1} = \lambda p_i \Longrightarrow \rho_{i+1} = \frac{\lambda}{\mu_{i+1}} = \frac{p_{i+1}}{p_i} \quad i = 0, 1, \ldots. \tag{8}$$

The values of ρ_i are obtained using the p_i's that result from the simulation, and are plotted in Figure 5 for both queues. 2 million scheduling periods are used for the simulation. Note that they converge as i increases. This is expected since, as the queue grows, the scheduled packet will need to wait for a time longer than a scheduling period in order to leave the system. Hence, in this case, the fact that the scheduling happens in specific instants does not influence the service rate. The oscillations as the delay increases are due to the inaccuracy of the p_i's due to the fewer number of samples for the less probable states.

Based on Figure 5, the following approximation is used:

$$\begin{aligned} \hat{\rho}_i &= \rho_i \quad \text{for } 1 \le i \le K-1 \\ \hat{\rho}_i &= \rho_K \quad \text{for } i \ge K. \end{aligned} \tag{9}$$

Then, the delay distribution probabilities are calculated as follows

$$\begin{aligned} \hat{p}_{i+1} &= \hat{p}_{i+1}\hat{p}_i, \quad i = 0, 1, \ldots, \\ \hat{p}_0 &= 1 - \sum_{i=1}^{\infty} \hat{p}_i = 1 - \sum_{i=1}^{\infty}\left(\prod_{j=1}^{i}\hat{\rho}_j\right)\hat{p}_0 \\ &= 1 - \sum_{i=1}^{K-2}\left(\prod_{j=1}^{i}\hat{\rho}_j\right)\hat{p}_0 - \sum_{i=K-1}^{\infty}\left(\prod_{j=1}^{i}\hat{\rho}_j\right)\hat{p}_0 \\ &= 1 - \sum_{i=1}^{K-2}\left(\prod_{j=1}^{i}\hat{\rho}_j\right)\hat{p}_0 - \prod_{j=1}^{K-1}\hat{\rho}_j \textstyle\sum_{m=0}^{\infty}\hat{\rho}_K^m \hat{p}_0 \\ &= 1 - \sum_{i=1}^{K-2}\left(\prod_{j=1}^{i}\hat{\rho}_j\right)\hat{p}_0 - \frac{\prod_{j=1}^{K-1}\hat{\rho}_j}{1-\hat{\rho}_K}\hat{p}_0 \Longrightarrow \\ \hat{p}_0 &= \frac{1}{1 + \sum_{i=1}^{K-2}\left(\prod_{j=1}^{i}\hat{\rho}_j\right) + \prod_{j=1}^{K-1}\hat{\rho}_j/(1-\hat{\rho}_K)}. \end{aligned} \tag{10}$$

The P_i's are approximated using $K = 6$ and $\hat{\rho}_K = 0.86$ for both queues. The resulting approximation of the packet delay distribution is shown in Figure 4. As can be seen, although the queues for the scenario considered in this paper

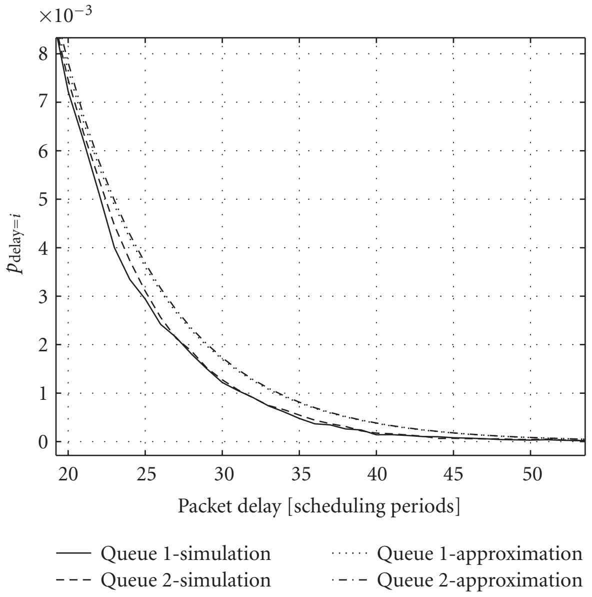


Figure 6: Distribution of the packet delay (detail).

are not independent, the approximation is good, and can be used to predict the delay of the packets scheduled by QPS provided that the arriving traffic rates are known. In Figure 6 more detail is shown for the states corresponding to higher delays. The approximation is slightly pessimistic but still very close to the actual values. In the following section, the distribution of the packet delay is used in order to decide on the encoding rate of video streams that are broadcast to different users and are scheduled using QPS.

4. CHOOSING THE ENCODER RATE FOR VIDEO STREAMS SCHEDULED BY QPS

Video streams transmitted in packets are subject to distortion. Distortion results from several sources such as encoder compression, corrupted data and lost or delayed packets. In [10] the authors develop models and derive expressions for the overall distortion of a video stream $D_d = D_e + D_v$. The first term D_e is the distortion because of signal compression at the encoder. It depends on the INTRA frame rate β and the rate R_e at the output of the decoder. R_e may need to be lowered in order to allow for a more redundant channel code, and, therefore, better protection against noise in the channel. D_v is the distortion occurring at the decoder and is related to the lost or corrupted packets that cannot be used by the decoder to reconstruct the transmitted video stream.

In [10], the channel capacity was assumed to be fixed and the reason for varying R_e was in order to leave more (or less) room to the channel code. The stronger the channel code is the smaller the probability of erroneous packets will be, leading to reduced decoder distortion. Therefore, the choice of R_e (and the associated channel code rate) leads to a tradeoff between D_e and D_v. By choosing R_e, the channel code and the INTRA rate β appropriately, the smallest value of the overall distortion D_d can be found.

In the scenario examined in this paper, a new tradeoff is created between D_e and D_v. In a BC where many users compete for the resources, a larger channel rate also means larger average (and maximum) delays. Hence, allowing the video encoder to send with a faster rate also increases the probability that a packet will not arrive early enough for the decoder to be able to use it. The number of packets with delays that exceed a given threshold adds to the number of packets that are corrupted in the channel and, therefore, the overall number of unusuable packets increases. Consequently, this leads to larger distortion.

As explained in [10] the exact value of the distortion depends on many factors such as the particular stream that is being transmitted, the video encoder and the spatial filters of the decoder, all of which are outside the scope of this paper. Therefore, in this article, it is briefly suggested how the effect of the channel delay can be added to the calculation of D_v. Then, the system optimization can proceed along the lines of [10]. From [10], $D_v = \sigma_{u0}^2 P_L \sum_{t=0}^{T-1}((1-\beta t)/(1-\gamma t))$, where β is the INTRA rate, γ is the leakage parameter that is determined by the loop filter of the decoder, $T = 1/\beta$ is the INTRA update interval, σ_{u0}^2 describes the sensitivity of the video decoder to an increase in the error rate and P_L is the residual packet error rate. When QPS is used, the proportion $(1 - P_L)$ of packets that are not corrupted in the channel or are lost for other reasons, are subject to delays at the queue of the scheduler. The probability of the delay of a packet exceeding a given threshold T_{del} can be found by forming $P_{\text{late}} = \sum_{d=D}^{\infty} p_d = 1 - \sum_{d=0}^{D-1} p_d$, where D is the first value of the delay that exceeds T_{del}. Alternatively, the approximate probabilities $\widehat{p}_d$ that were derived using the Markov Chain model can be used. Hence, the new P_L' that should be used for the calculation of D_v is equal to $P_L' = P_L + (1 - P_L)P_{\text{late}}$.

5. CONCLUSION

In this paper, Queue Proportional Scheduling was considered with video transmission in mind. First, it was shown that, if an increase in the average packet delay can be tolerated, the use of Queue State Information can become less frequent, therefore simplifying the scheduler. A modified QPS scheduler was proposed that performs better than the approach of simply using outdated QSI for scheduling. The modified scheduler performs better than the simplistic approach without increasing considerably the implementation complexity. Moreover, it was proved that, under certain conditions, the modified QPS is throughput optimal. It was also shown using simulation that, in systems using QPS, the distribution of the packet delay can be approximated satisfactorily by a Markov Chain model. This model makes it easier to obtain an estimate for the tail of the probability distribution, and, consequently calculate the video distortion caused by packets whose delay exceeds the buffer size of the decoder. It was also discussed how the effect of late packet arrivals can be included in the calculation of the distortion.

APPENDIX

Proof of Theorem 1

Let $\mathbf{Q}(t) = \mathbf{q}_t = [\, q_{t,1} \;\; q_{t,2} \;\; \cdots \;\; q_{t,K} \,]^T$, where K is the number of users of the BC. Assume, without loss of generality, that $q_{t,1} \neq 0$ and $\lambda_1 \neq 0$. Then, $\mathbf{q}_t$ can be written as $\mathbf{q}_t = w(t)[\lambda_1, \lambda_2 + \Delta\lambda_2, \dots, \lambda_K + \Delta\lambda_K]^T$, where $w(t) = q_{t,1}/\lambda_1$ and $\Delta\lambda_i$ are such that $w(t)(\lambda_i + \Delta\lambda_i) = q_{t,i}$ for $i = 2, \dots, K$. Therefore,

$$E[\mathbf{Q}(t+1) \,|\, \mathbf{Q}(t) = \mathbf{q}_t] = \mathbf{q}_t + \boldsymbol{\lambda} - \mathbf{R}(t). \tag{A.1}$$

When the queue state is used for scheduling, $\mathbf{R}(t) = r(t)(\mathbf{q}_t/w(t))$ where $r(t)$ is such that $\mathbf{R}(t)$ is inside the capacity region, and does not exceed $\mathbf{q}_t$. Hence, $r(t) \leq w(t)$. As is shown in [9],

$$\begin{aligned} &E[\mathbf{Q}(t+1) \,|\, \mathbf{Q}(t) = \mathbf{q}_t] \\ &\quad = (w(t) - r(t) + 1) \\ &\qquad \times [\lambda_1, \lambda_2 + \gamma(t)\Delta\lambda_2, \dots, \lambda_K + \gamma(t)\Delta\lambda_K]^T, \end{aligned} \tag{A.2}$$

where $\gamma(t) = 1 - 1/(w(t) - r(t) + 1)$. If $\mathbf{q}_t \in C_{\mathrm{BC}}$ then $w(t) = r(t)$, $\gamma(t) = 0$ and $E[\mathbf{Q}(t+1) \mid \mathbf{Q}(t) = \mathbf{q}_t] = \boldsymbol{\lambda}$. Else, $\gamma(t)$ is strictly less than 1.

Consider now the case of the modified QPS algorithm, and assume that the queue state information gets used once every $L = 2$ transmission periods, that is, for $t, t+2, \dots, t+2n$, information on $\mathbf{Q}(t)$ is used, whereas during the other periods scheduling is based on $\boldsymbol{\lambda}$

$$\begin{aligned} &E[\mathbf{Q}(t+2) \,|\, \mathbf{Q}(t) = \mathbf{q}_t] \\ &\quad = \mathbf{q}_t + 2\boldsymbol{\lambda} - \mathbf{R}(t) - \mathbf{R}(t+1) \\ &\quad = \mathbf{q}_t + 2\boldsymbol{\lambda} - \frac{r(t)}{w(t)}\mathbf{q}_t - \alpha(t)\boldsymbol{\lambda} \\ &\quad = (w(t) + 2 - r(t) - \alpha(t)) \\ &\qquad \times [\lambda_1, \lambda_2 + \gamma(t)\Delta\lambda_2, \dots, \lambda_K + \gamma(t)\Delta\lambda_K]^T, \end{aligned} \tag{A.3}$$

where $\gamma(t) = 1 - (2-\alpha(t))/(w(t) - r(t) + 2 - \alpha(t))$. If $\mathbf{q}_t \in C_{\mathrm{BC}}$, $w(t) = r(t)$ and $\gamma(t) = 0$. Assuming that $2 - \alpha(t)$ can never become negative, that is, that the average arrival rate is large enough so that it is more than halfway between zero and the boundary of the capacity region, $\gamma(t) < 1$ in all other cases. Therefore, similar to the QPS proof of [9], it can be deduced that the slope of $\mathbf{Q}(t)$ converges to the slope of $\boldsymbol{\lambda}$ in the sense that

$$\theta_{\boldsymbol{\lambda}}(\mathbf{q}_t) \geq \theta_{\boldsymbol{\lambda}}(E[\mathbf{Q}(t+2) \,|\, \mathbf{Q}(t) = \mathbf{q}_t]), \tag{A.4}$$

where $\theta_{\boldsymbol{\lambda}}(\mathbf{x}) = \cos^{-1}(\boldsymbol{\lambda}^T \mathbf{x}/\|\boldsymbol{\lambda}\|_2 \|\mathbf{x}\|_2)$, $0 \leq \theta_{\boldsymbol{\lambda}}(\mathbf{x}) \leq \pi/2$.

For the general case where queue State Information is used every L scheduling periods,

$$\begin{aligned} &E[\mathbf{Q}(t+L) \,|\, \mathbf{Q}(t) = \mathbf{q}_t] \\ &\quad = \mathbf{q}_t + L\boldsymbol{\lambda} - \mathbf{R}(t) - \sum_{l=1}^{L-1} \mathbf{R}(t+l) \\ &\quad = \mathbf{q}_t + L\boldsymbol{\lambda} - \frac{r(t)}{w(t)}\mathbf{q}_t - \sum_{l=1}^{L-1} \alpha_l(t)\boldsymbol{\lambda} \\ &\quad = \left(w(t) + L - r(t) - \sum_{l=1}^{L-1} \alpha_l(t) \right) \\ &\qquad \times [\lambda_1, \lambda_2 + \gamma(t)\Delta\lambda_2, \dots, \lambda_K + \gamma(t)\Delta\lambda_K], \end{aligned} \tag{A.5}$$

where $\gamma(t) = 1 - (L - \sum_{l=1}^{L-1} \alpha_l(t))/(w(t) - r(t) + L - \sum_{l=1}^{L-1} \alpha_l - (t))$. Again, $\gamma(t) = 0$ when $\mathbf{R}(t) = \mathbf{q}_t$. Else, $\gamma(t) < 1$ as long as $L - \sum_{l=1}^{L-1} \alpha_l(t) > 0$. This can be guaranteed if, for each $l, \alpha_l(t) < L/(L-1)$ which means that if the boundary of C_{BC} is at $\alpha\boldsymbol{\lambda}$, then $\alpha < L/(L-1)$.

Since the packet arrivals are Poisson, the queue state of each user is a first order Markov process. From the Chapman-Kolmogorov Equations [11]

$$\begin{aligned} &E[\mathbf{Q}(t+L) \,|\, \mathbf{Q}(0) = \mathbf{q}_0] \\ &\quad = E[E[\mathbf{Q}(t+L) \mid \mathbf{Q}(t)] \mid \mathbf{Q}(0) = \mathbf{q}_0] \quad \text{for} t = 0, L, \dots. \end{aligned} \tag{A.6}$$

From (A.4), $\theta_{\boldsymbol{\lambda}}(E[\mathbf{Q}(t)|\mathbf{Q}(0) = \mathbf{q}_0]) \geq \theta_{\boldsymbol{\lambda}}(E[E[\mathbf{Q}(t+L)|\mathbf{Q}(t)]|\mathbf{Q}(0) = \mathbf{q}_0]) \overset{(6)}{=} \theta_{\boldsymbol{\lambda}}(E[\mathbf{Q}(t+L)|\mathbf{Q}(0) = \mathbf{q}_0])$ for $t = 0, L, \dots$. Thus, if $\theta_n \triangleq \theta_{\boldsymbol{\lambda}}(E[\mathbf{Q}(nL)|\mathbf{Q}(0) = \mathbf{q}_0])$, $n = 1, 2, \dots$, and $\theta_0 \geq 0$, θ_n converges. It can be easily deduced (as, e.g., in [9]), that $\theta_n \to 0$as $n \to \infty$, and, therefore, $E[\mathbf{Q}(nL)|\mathbf{q}_0] \to w(t)\boldsymbol{\lambda}$.

Proof of Theorem 2

It will be shown that, for any $\boldsymbol{\lambda} \in C_{\mathrm{BC}}$ such that $\alpha\boldsymbol{\lambda}$ is at the boundary of C_{BC} and $1 \leq \alpha \leq L/(L-1)$, the queue lengths of all users can be kept finite. The following Lyapunov function is chosen: $L(\mathbf{Q}(t)) = \sum_{i=1}^{K} Q_i(t)$. Then, assuming that t is a scheduling period when the Queue State Information is used, $L(\mathbf{Q}(t+1)) = \sum_{i=1}^{K} Q_i(t+1) = \sum_{i=1}^{K} \{(Q_i(t) - R_i(t))^+ + Z_i(t)\}$. Since $R(t) \leq Q(t)$, $L(\mathbf{Q}(t+1)) = \sum_{i=1}^{K} (Q_i(t) - R_i(t) + Z_i(t))$. $L(\mathbf{Q}(t+2)) = \sum_{i=1}^{K} Q_i(t+2) = \sum_{i=1}^{K} [(Q_i(t) - R_i(t) + Z_i(t) - R_i(t+1))^+ + Z_i(t+1)]$. Again, since the queue lengths cannot become negative, $L(\mathbf{Q}(t+2)) = \sum_{i=1}^{K} Q_i(t+2) = \sum_{i=1}^{K} (Q_i(t) - R_i(t) + Z_i(t) - R_i(t+1) + Z_i(t+1))$. Hence, $L(\mathbf{Q}(t+L)) = \sum_{i=1}^{K} \{Q_i(t) - \sum_{l=0}^{L-1} R_i(t+l) + \sum_{l=0}^{L-1} Z_i(t+l)\}$. Assume that $\mathbf{Q}(0) = \mathbf{q}_0$, where $\max\{q_{i,0}\}$ is sufficiently small. Then, the expected drift of the Lyapunov function conditioned on $\mathbf{q}_t$ is equal to

$$\begin{aligned} &E[L(\mathbf{Q}(t+L)) - L(\mathbf{Q}(t)) \,|\, \mathbf{Q}(t) = \mathbf{q}_t] \\ &\quad = \sum_{k=1}^{K} \Bigg\{ L\lambda_k - (R_k(t) \,|\, \mathbf{Q}(t) = \mathbf{q}_t) \\ &\qquad\qquad - \sum_{l=1}^{L-1} (R_k(t+l) \,|\, \mathbf{Q}(t) = \mathbf{q}_t) \Bigg\}. \end{aligned} \tag{A.7}$$

It will be shown that, as $\|\mathbf{q}_t\|_\infty = \max\{q_{i,t}\}\to\infty$, the Lyapunov drift (A.7) becomes strictly negative. $\|\mathbf{q}_t\|_\infty\to\infty$ also implies that $t\to\infty$ since a queue cannot grow to infinity during a finite time interval. As was shown in Theorem 1, if a condition on $\boldsymbol{\lambda}$ holds, $\mathbf{Q}(nL)$ converges to $w(t)\boldsymbol{\lambda}$ as $t\to\infty$. Hence, at time $t\to\infty$, QPS will use the value of $\mathbf{Q}(t)$ and the rate $\mathbf{R}(t)$ will be equal to $r(t)\mathbf{Q}(t)\to r(t)w(t)\boldsymbol{\lambda} = W(t)\boldsymbol{\lambda}$. Regarding the rates $\mathbf{R}(t{+}l)$, $1 \le l \le L-1$, these are, by definition of the modified QPS, equal to $\alpha(t+l)\boldsymbol{\lambda}$. Therefore, for $t\to\infty$,

$$\begin{aligned} &E\big[L(\mathbf{Q}(t+L)) - L(\mathbf{Q}(t)) \,|\, \mathbf{Q}(t) = \mathbf{q}_t\big] \\ &\quad\longrightarrow \sum_{k=1}^{K}\left\{L\lambda_k - W(t)\lambda_k - \sum_{l=1}^{L-1}\alpha(t+l)\lambda_k\right\}. \end{aligned} \tag{A.8}$$

Since $\boldsymbol{\lambda}$ is in the interior of C_{BC}, $W(t)$ and the $\alpha(t+l)$ will be strictly larger than 1 because the modified QPS algorithm chooses the longest vector along the direction of $\boldsymbol{\lambda}$ that belongs to the BC Capacity region. When $\|\mathbf{q}_t\|_\infty\to\infty$ this vector reaches the boundary of the capacity region, and is, therefore, longer than $\boldsymbol{\lambda}$. Hence, the Lyapunov drift is strictly negative for any $\boldsymbol{\lambda}$ satisfying the condition that $\alpha\boldsymbol{\lambda} \in C_{\mathrm{BC}}, 1 < \alpha \le L/(L-1)$.

REFERENCES

[1] IEEE Std 802.16-2004, "IEEE standard for local and metropolitan area networks, Part 16: air interface for fixed broadband wireless access systems," October 2004.

[2] IEEE Std 802.16e-2005, "IEEE standard for local and metropolitan area networks, Part 16: air interface for fixed broadband wireless access systems, amendment 2: physical and medium access control layers for combined fixed and mobile operation in licensed bands," February 2006.

[3] Long Term Evolution of the 3GPP radio technology, http://www.3gpp.org/Highlights/LTE/LTE.htm.

[4] IEEE Std 802.11n TGn draft 2.0, "Amendment to standard for information technology-telecommunications and information exchange between systems-local and metropolitan networks- specific requirements- Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: enhancements for higher throughput".

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.

[6] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1260–1267, 1999.

[7] A. Eryilmaz, R. Srikant, and J. R. Perkins, "Throughput-optimal scheduling for broadcast channels," in *Modeling and Design of Wireless Networks*, vol. 4531 of *Proceedings of SPIE*, pp. 70–78, Denver, Colo, USA, August 2001.

[8] K. Seong, R. Narasimhan, and J. M. Cioffi, "Queue proportional scheduling via geometric programming in fading broadcast channels," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1593–1602, 2006.

[9] K. Seong, R. Narasimhan, and J. M. Cioffi, "Queue proportional scheduling in Gaussian broadcast channels," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 1647–1652, Istanbul, Turkey, June 2006.

[10] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[11] S. Ross, *Stochastic Processes*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2007, Article ID 76846, 11 pages
doi:10.1155/2007/76846

Research Article

Distortion Optimized Packet Scheduling and Prioritization of Multiple Video Streams over 802.11e Networks

Ilias Politis,[1] Michail Tsagkaropoulos,[2] Thomas Pliakas,[3] and Tasos Dagiuklas[4]

[1] *Department of Electrical & Computer Engineering, University of Patras, Greece*
[2] *Mobility Applications Laboratory - Networks, Ericsson Deutschland GmbH - Eurolab, Germany*
[3] *Department of Information & Communication Systems Engineering, University of Aegean, Greece*
[4] *Department of Communication Systems and Networks, Technological Educational Institute of Messolonghi, Greece*

Received 11 June 2007; Accepted 20 August 2007

Recommended by Stavros Kotsopoulos

This paper presents a generic framework solution for minimizing video distortion of all multiple video streams transmitted over 802.11e wireless networks, including intelligent packet scheduling and channel access differentiation mechanisms. A distortion prediction model designed to capture the multireferenced frame coding characteristic of H.264/AVC encoded videos is used to predetermine the distortion importance of each video packet in all streams. Two intelligent scheduling algorithms are proposed: the "even-loss distribution," where each video sender is experiencing the same loss and the "greedy-loss distribution" packet scheduling, where selected packets are dropped over all streams, ensuring that the most significant video stream in terms of picture context and quality characteristics will experience minimum losses. The proposed model has been verified with actual distortion measurements and has been found more accurate than the "additive distortion" model that omits the correlation among lost frames. The paper includes analytical and simulation results from the comparison of both schemes and from their comparison to the simplified additive model, for different video sequences and channel conditions.

1. INTRODUCTION

The ever growing demand for multimedia services send over wireless access networks leads to cases where multiple users are competing over scarce channel resources. Typical example of such cases is the concurrent media streaming over bottleneck network nodes or the wireless lan-shared channel capacity. Therefore, it is important to optimize the entire transmission system in order to achieve high performance and increase quality of service [1].

In this context, next wireless LAN (WLAN) generation could provide multimedia services to mobiles and fixed users through wireless access, with the development of the high-speed physical (PHY) layers IEEE 802.11g (54 Mbps) [2]. However, wireless channel characteristics such as shadowing, multipath, fading, and interferences still limit the available bandwidth for the deployed applications. Consequently, video compression techniques and transmission techniques are a critical part of multimedia applications over WLAN. The IEEE 802.11 networks are most commonly used due to their low cost and easy deployment. The IEEE 802.11 provides a best effort service, which indicates that every data packet-handed over to the 802.11 interfaces receives similar treatment as other packets in terms of delivery guarantees, that is, available bandwidth, latency, jitter, and so forth. It provides two access mechanisms, in order for the mobile terminals to gain access to shared wireless medium [2]; (1) the *point coordination function* (PCF) and (2) the *distribution coordination function* (DCF). The PCF originally aims at supporting real-time traffic, but is rarely implemented in current commercial products due to its implementation complexity and uncertainty on the efficiency. On the other hand, DCF is a contention-based channel access protocol [3]. In order to deliver real-time video traffic, which is sensitive to packet latency and effective bandwidth characteristic of the underlying network, QoS, and differential service [4–6], the 802.11e [7] seems to be an appropriate solution. It introduces the hybrid coordination function (HCF) that concurrently uses a contention-based mechanism and a pooling-based mechanism, enhanced DCF (EDCF), and HCF controlled channel access (HCCA), respectively. Like DCF, EDCA is very likely to be the dominant channel access mechanism in WLANs because it features a distributed and easily deployed mechanism.

As far as the application plane is concerned, video streaming over emerging wireless networks is becoming ever more popular, hence it is important to develop error resilient source coding and transmission techniques [8, 9]. In detail, the video quality of a decoded video transmitted over a wireless network is often associated with the average pixel-by-pixel distortion of the video frames. This distortion results from both the compression scheme of the source encoder and the transmission channel losses. The latter is commonly referred to as channel distortion and depends on the communication channel loss characteristics, the intra-update period, and the error concealment method applied durring decoding at the receiver. Modeling channel distortion is accurately the key for rate-distortion optimization and end-to-end quality of video communications. There is a large number of research works that has been reported and regards modeling the impact of packet loss on video distortion. Such models can be fall in two categories. In the first category, the models consider that distortion is proportional to the number of losses within video sequences [10, 11]. These studies also suggest that the average distortion of multiple losses can be derived as a superposition of the uncorrelated error signals. However, these models are accurate for low-residual error rates and when such errors are sufficiently apart of each other and there are no burst errors. As an effect, the impact of multiple losses is considered as the superposition of multiple losses. The models in the second category consider the correlation between error signals, giving rise to more complex loss patterns, burst of losses, and losses separated with small lags, than just isolated losses. Evidently, burst losses lead to larger distortion than individual single losses. In this case, the burst length affects the video quality in a distinct way and it has been determined analytically for different packet losses including burst errors and errors with lag [12–14]. However, all the models mentioned above have not considered the inherent feature of H.264/AVC encoder that can select between a number of previously encoded frames highly correlated with the current frame, as reference for motion-compensated prediction of each intermacroblock or macroblock partition. The use of multiple reference pictures allows H.264/AVC to achieve significantly better compression than any previous standard and on the same time, it affects the error propagation in the case of an error frame. Accurate distortion models are very important especially when decisions like rate-distortion optimization and packet scheduling are based on these models.

In the case of concurrent multiple media streams, it is important to develop effective algorithms that will minimize the average distortion of each user. Therefore, there are often cases where users are allocated insufficient transmission bandwidth. In such occasions, the sender has to be able to reduce its video transmission rate by selecting which packets to drop prior to the transmission. Selecting the proper video packets to drop can become very crucial, as it has a very significant effect on the reconstructed video. Several solutions have been proposed for adapting the video characteristics to the transmission channel constraints. Video transcoding techniques [15] are applied in order to reencode the video stream with lower bit rates but increase the complexity of the encoder dramatically. Scalable video coding [16] provides an inherent prioritization of the encoded video packets, thus it allows the sender to quickly select which packets to drop, however scalable coding has not been yet widely accepted. An extended work has been made on the video rate-distortion optimized transmission over wireless networks [17] and has proposed a cross-layer ARQ for H.264 over 802.11 networks which gives priority to important packets during retransmission while [18] describes an adaptive quality-of-service strategy for 802.11 networks that is applied to a single stream and without considering R-D optimization. Finally, [19] studies an R-D optimized bandwidth adaptation of multiple video streams that is performed by a network node that drops packets from all incoming streams.

This paper proposes a novel framework based on two scheduling algorithms named “even-loss distribution” and “greedy-loss distribution” for effective packet scheduling and optimized differentiated channel access that minimizes the overall distortion of the video streams. The framework is based on a new distortion prediction model that takes into consideration the high correlation among reference frames of H.264/AVC encoded video streams, the size of the reference list used and the type of error patterns (single errors, burst of errors, errors separated with lag) that are imposed by the transmission channel. The proposed distortion model has been evaluated against real measurements and is compared with the simplified “additive”-based distortion model that omits the correlation among the frames. This model is power efficient as it requires simplified calculations that can be performed prior to the video transmission. Each of the video senders is informed by the system about the current available capacity of the communication channel and allocates its transmission rate accordingly. The tradeoff between transmission rate and video distortion is different for the two proposed scheduling algorithms.The transmitted packets are then processed through the proposed 802.11e MAC mechanism that is able to differentiate the video streams according to their distortion imposed by the scheduling algorithm. The video sequence that has experienced the greater losses is treated as high-priority traffic and is allocated to the appropriate access category. Analytical results indicate the advantages and disadvantages of the proposed scheduling schemes. It is evident that video streams with different type of context and coding characteristics are affected in different way by the “even-loss distribution” and the “greedy-loss distribution” scheduling algorithms.

The rest of the paper is organized as follows. Section 2 provides an overview of the additive distortion model and the mathematical background of the proposed model. In Section 3, the system model is presented. The setup of the 802.11e system and the configuration of the H.264/AVC encoder are discussed. In Section 4, the “even-loss distribution” and the “greedy-loss distribution” scheduling schemes are presented. The simulation setup and an extended discussion upon the results are provided in Section 5 and the paper concludes in Section 6. Appendices at the end of the paper discuss the mathematical model evaluation against actual measurements and compare the behavior of the proposed model and that of the “additive” distortion model, for several different error patterns (single losses, burst of losses, losses separated with lag).

2. ANALYTICAL DISTORTION MODEL

2.1. Overview of additive model

The following analysis considers a video sequence that begins with an I-frame and is followed by P-frames with an intraframe period N in order to increase error resilience. It is assumed that this intraframe period equals the total error recovery period, in case of packet loss. As an effect, losses that occur outside this period are uncorrelated. Let k be the index of video picture. Then the total increase in MSE distortion that will affect the video if picture k is lost is given by $D(k) = \sum_{i=1}^{L} \Delta d_i$, where L is the number of video pictures in the sequence and Δd_i is the increase in MSE distortion relative to picture i, given that picture k is lost. It is assumed that previous frame concealment is used and there are no prior losses. It has been proved [13] that the MSE of subsequent frames will have a nonzero value, however due to the intraupdate and the spatial filtering, its amplitude decreases gradually until it becomes zero at a point far enough from frame k. Based on this definition, the total video distortion due multiple lost frames M will be the sum of all individual MSEs over all the frames L affected by these losses. This is known as the additive distortion model as presented in [1, 10]:

$$D_{\text{total}} = \sum_{j=1}^{L} D(j) = \sum_{j=1}^{L} \sum_{i=1}^{L} \Delta d_i(j). \tag{1}$$

2.2. Proposed model

We define a list of previously encoded reference frames with size M_{REF} that is used during the encoding and decoding processes for motion-compensated prediction. Moreover, without loss of generality, each frame is coded into a single packet, although this condition can be extended to support different packetization schemes. Finally, in our analysis (similar to previous studies [10, 13, 20]), a simple error concealment mechanism is used that in the case of a frame loss, it replaces it with its previous at the decoder. The error power introduced in a single frame k is denoted by $\sigma_s^2(k)$ and the total video distortion due to error frame k and its error power propagation to the following frames is denoted by $D_s(k)$. Correspondingly, $\sigma^2(k)$ and D are the MSE and the sum of the MSE values over all frames in the intraframe period, of more general loss patterns, respectively. The proposed model includes analytical models for a single frame loss, a burst of losses with variable burst length, and frame losses separated by a lag.

As it has already been determined in [10, 13], the distortion metric consists of two factors: a geometric attenuation factor (due to spatial filter) and a linear attenuation factor (due to intra-update). We have introduced a third parameter M_{REF} that accounts for the impact of the number of reference frames on the distortion propagation. Hence, the error power propagation at frame $k + l$, due to a single frame loss at k, is

$$\sigma^2(k + l) = \sigma^2(k) \cdot \Lambda_l, \tag{2}$$

and the error power propagation effect is

$$\begin{aligned}\Lambda(l) = {} & \left(1 - \frac{l-1}{N}\right) \\ & \times \left(r^{l-1} + \frac{r^{N-l+1}}{\Phi(N,r) + \left((M_{\text{REF}}/N) + 1\right)\left((N-l+1)/N\right)}\right),\end{aligned} \tag{3}$$

where $\Phi(N,r)$ depends on the scene content of the particular video sequence and the coding parameters. This value has been estimated through curve fitting for different isolated errors. The value of this parameter for the video sequence *Foreman*, that was used as reference sequence, is $\Phi(N,r) = (3 - (2 \cdot (N - l + 1)/N)) \times r$. Additionally, $r < 1$ is the spatial filtering factor. Considering a single error at frame k and a period long enough for error recovery, the total video distortion is

$$D_s(k) = \sigma_s^2(k) \cdot A = \sigma_s^2(k) \cdot \sum_{i=0}^{N-1-k} \Lambda_i, \tag{4}$$

where $\sigma_s^2(k)$ is the error power introduced at k and A represents the propagation effect of the initial distortion at k over all the following frames i until the end of the intraframe period N. In order to extend the distortion model for a single loss to a burst of frame losses with a burst length B, we need to consider the correlation between these frames. Assume a correlation coefficient among two error frames $e_s(k-1)$ and $e_s(k)$ to be

$$\rho_{k-1,k} = \frac{\left(e_s^T(k-1) \cdot e_s(k)\right)/M}{\sigma_s(k-1) \cdot \sigma_s(k)}, \tag{5}$$

where $\sigma_s(k-1)$ and $\sigma_s(k)$ are the MSEs of frames $k-1$ and k accordingly [13]. $M = M_1 \times M_2$ is the 2D array of pixels in each frame k. Then the distortion due to a burst of error frames is

$$\begin{aligned} & D(k - B + 1, \ldots, k) \\ & \quad = \sum_{i=k-B+1}^{k-1} \sigma^2(i) + A(B) \cdot \sigma^2(k), \\ & A(B) = A_0 + c \cdot (B - 2), \quad B \geq 2, \\ & \sigma^2(k) = \sum_{i=k-B+1}^{k} \sigma_s^2(i) + 2 \cdot \sum_{i=k-B+1}^{k} \sum_{j=i+1}^{k} \rho_{i,j} \cdot \sigma_s(i) \cdot \sigma_s(j). \end{aligned} \tag{6}$$

In (6), $\sigma^2(i)$ is the sum of the distortion over all i frames, B is the burst length and $B \geq 2$, and $\rho_{i,j}$ is the correlation coefficient of frames i and j. $A(B)$ represents the variation on the shape of the introduced error power relative to burst length B and A_0 is the ratio for $B = 2$. In [13] it was shown that as the burst length B varies, the shape of the initial error signals PSD also varies, which leads to a variation in the spatial filtering factor r in (3). The process of error power reduction by loop filtering can be modeled with a linear system, and r is the proportion of the power of the introduced error passing through the system. In [10], the loop filter is approximated by a Gaussian lowpass filter. Hence, as B increases, r increases as the PSD of the error is more concentrated in the lower band. The variation of r is relatively small and can be approximated as a linear function of B, that is, $A(B) = A_0 + c \cdot (B - 2)$, where A_0 is the ratio for $B = 2$ and c is the slope of the increase.

Extending the above analysis in order to predict the video distortion in the event of two losses separated by short lag l smaller than the intraframe period N, the distortion can be expressed by the following equation:

$$D(k-1,k) = \sigma_s^2(k-1) + \sum_{i=0}^{l-1} \sigma^2(i) + A \cdot \sigma^2(k), \quad l \in (l,N]. \tag{7}$$

The above distortion prediction model represented by equations (4), (6), and (7) is based on specific error patterns (isolated errors, burst errors, and errors with lag). In order to predict the distortion caused by more complex error patterns that may include different combinations of the above, it is required to provide a more generic formula. This formula will take into account both the type of error and the different error frame (multireference) dependencies that are present in H.264/AVC. Hence, in the general case of multiple combinations of erroneous frames the distortion D_n, where n is the error pattern size and $n \geq 1$, is modeled by recursive formula (8). In (8), $D_1, D_2, \ldots, D_n$ are the total video distortions due to $1, 2n$ error frames. The frame number of the nth erroneous frame is denoted by F_n. This recursive formula calculates the total distortion for the first error frame and depending on whether the next error frame is correlated or not with the previous error frame, it combines the above formulas and estimates the total distortion of the resulted error pattern:

$$\begin{aligned}
D_1 &= \sum_{\substack{k=F_1\\ i=0}}^{\substack{k=F_1+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_1) = \sum_{k=F_1,i=0}^{k=F_1+i,i=N-1+k} \Lambda_{(i)} \cdot \sigma_s^2(F_1),\\
D_2 &= \sigma_s^2(F_1)\\
&\quad + \begin{cases}
\displaystyle\sum_{\substack{k=F_2\\ i=0}}^{\substack{k=F_2+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma_s^2(F_2), & \text{uncorrelated,}\\
\displaystyle\sum_{\substack{k=F_2\\ i=0}}^{\substack{k=F_2+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_2), & \text{burst,}\\
\displaystyle\sum_{\substack{k=F_1\\ i=0}}^{\substack{k=F_2-1\\ i=F_2-1-F_1}} \Lambda_{(i)} \cdot \sigma^2(F_2) + \sum_{\substack{k=F_2\\ i=0}}^{\substack{k=F_2+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_2), & \text{lag,}
\end{cases}\\
&\vdots\\
D_n &= D_{n-1} - \sum_{\substack{k=F_{n-1}\\ i=0}}^{\substack{k=F_{n-1}+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_{n-1}) + \sigma^2(F_{n-1})\\
&\quad + \begin{cases}
\displaystyle\sum_{\substack{k=F_n\\ i=0}}^{\substack{k=F_n+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma_s^2(F_n), & \text{uncorrelated,}\\
\displaystyle\sum_{\substack{k=F_n\\ i=0}}^{\substack{k=F_n+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_n), & \text{burst,}\\
\displaystyle\sum_{\substack{k=F_{n-1}\\ i=0}}^{\substack{k=F_n-1\\ i=F_n-1-F_{n-1}}} \Lambda_{(i)} \cdot \sigma^2(F_n) + \sum_{\substack{k=F_n\\ i=0}}^{\substack{k=F_n+i\\ i=N-1+k}} \Lambda_{(i)} \cdot \sigma^2(F_2), & \text{lag.}
\end{cases}
\end{aligned} \tag{8}$$

The proposed model presented in this section has been evaluated against real measurements. Extensive simulations have been performed for different combinations of error patterns. These include single isolated frame losses, burst of frame losses with burst length $B \geq 2$, and error frames separated by a small lag. Additionally, the proposed model was compared against the additive distortion model introduced in [10]. The simulations showed that our model can accurately capture the distortion effect for reference-based H.264 coding. Appendices at the end of the paper present the simulation results for the *Foreman* testing sequence.

3. SYSTEM MODEL

3.1. IEEE 802.11e EDCF

The IEEE 802.11e, recently established working group of IEEE 802.11e, provides applications with differential services and QoS support by priority-based contention service, that is, the enhanced DCF (EDCF). A concept of traffic category is introduced. Each traffic category is associated with the predetermined contention parameters, arbitration IFS (AIFS), $CW_{\min}$ and $CW_{\max}$, and the backoff persistence factor (PF) [21]. The lower AIFS/$CW_{\min}$/$CW_{\max}$ results in the higher probability of winning the channel contention. In EDCF, the contention window is expanded by the PF after collision. In the original DCF, the contention window is always doubled after collision (PF = 2), while in EDCF, PF may be of a different value. For each station, multiple traffic categories with different contention parameters can exist in parallel, thus leading to internal contention in each station. The collisions among internal contention are avoided by letting the highest priority traffic category win the contention window.

The values of AIFS[AC], $CW_{\min}[AC]$, and $CW_{\max}[AC]$, which are referred to as the EDCF parameters, are announced by the AP via beacon frames. The AP can adapt these parameters dynamically depending on network conditions. Basically, the smaller AIFS[AC] and $CW_{\min}[AC]$, the shorter the channel access delay for the corresponding priority, and hence the more capacity share for a given traffic condition. However, the probability of collisions increases when operating with smaller $CW_{\min}[AC]$. These parameters can be used in order to differentiate the channel access among different priority traffic.

3.2. H.264/AVC standard overview

The Moving Picture Experts Group and the Video Coding Experts Group (MPEG and VCEG) have developed a new standard that promises to outperform the earlier MPEG-4 and H.263 standards, providing better compression of video images. The new standard "advanced video coding" (AVC) is published jointly as Part 10 of MPEG-4 and ITU-T Recommendation H.264 [22]. Some of the important terminology adopted in the H.264 standard are as follows.

(1) A field or a frame (of progressive or interlaced video) is encoded to produce a coded picture. A coded frame has a frame number, which is not necessarily related to decoding order and each coded field of a progressive or interlaced frame has an associated picture order count, which defines the decoding order of fields.

(2) Previously coded pictures (reference pictures) may be used for interprediction of further coded pictures. Reference pictures are organized into one or two lists, refered to as list 0 and list 1.

(3) A coded picture consists of a number of macroblocks. Within each picture, macroblocks are arranged in slices, where a slice is a set of macroblocks in raster scan order. An I slice may contain only I macroblock types, a P slice may contain P and I macroblock types, and a B slice may contain B and I macroblock types.

H.264/AVC defines a set of three profiles, each supporting a particular set of coding functions and each specifying what is required of an encoder or decoder that complies with the profile. The baseline profile supports intra- and intercoding (using I-slices and P-slices) and entropy coding with context-adaptive variable-length codes (CAVLC). The main profile includes support for interlaced video, intercoding using B-slices, intercoding using weighted prediction and entropy coding using context-based arithmetic coding (CABAC). The extended profile does not support interlaced video or CABAC but adds modes to enable efficient switching between coded bitstreams (SP-and SI-slices) and improved error resilience (data partitioning).

Potential applications of the baseline profile include videotelephony, videoconferencing, and wireless communications; potential applications of the main profile include television broadcasting and video storage; and the extended profile may be particularly useful for streaming media applications. However, each profile has sufficient flexibility to support a wide range of applications and so these examples of applications should not be considered definitive.

The innovative feature on the development of H.264 encoder is the ability to use one or two of a number of previously encoded pictures (or frames) as a reference for motion-compensated prediction of each intercoded macroblock or macroblock partition. This enables the encoder to search for the best match (the most correlated picture) for the current macroblock partition from a wider set of pictures than just the previously encoded picture. The encoder and decoder each maintains one or two lists of reference pictures, containing pictures that have previously been encoded and decoded (occurring before and/or after the current picture in display order). Intercoded macroblocks and macroblock partitions in P slices are predicted from pictures in a single list, *list 0*. Intercoded macroblocks and macroblock partitions in a B slice may be predicted from two lists, *list 0* and *list 1*. The configuration of the H.264/AVC encoder used for the following simulations includes a reference *list 0* that contains $M_{\mathrm{REF}} = 5$, number of previously encoded pictures for motion compensated prediction, although in the appendices there are simulation results for a reference list of size $M_{\mathrm{REF}} = 10$ that prove that the proposed recursive model of (8) is valid for any *list 0* as long as $M_{\mathrm{REF}} \geq 2$.

4. PROPOSED SCHEDULING ALGORITHMS

4.1. Even-loss distribution

In this scenario, each video sender has precalculated the distortion effect of each packet and the total video distrortion of all the combination of dropped packets. The error combinations may include individual packet losses, burst of dropped packets with burst length $B \geq 2$, or dropped packets separated by a short lag and any combination of the above. These calculations are repeated in each transmission window. In the following analysis, each video sender can transmit packets during a sliding window 36 frames long. The packets in each transmission window, hence the resulted losses, are considered correlated. Each video sender is informed by the shared network about the available transmission rate. According to the precalculated video distortion, each sender may select the optimum pattern of packets to drop in order to optimize its transmission rate. The "even-loss distribution" algorithm ensures that each video stream will be evenly affected by the channel restrictions. Therefore, each video sender will be required to drop on average an equal number of packets. Although which packets will select to drop depends on the distortion prediction model.

Each of the video streams is characterized by the average distortion due to the packet scheduling. The 802.11e network has the ability to differentiate among different priority traffic depending on several QoS criteria. According to the "even-loss distribution" scenario, the 802.11e model that is applied has been configured to differentiate among the multiple video streams based on their average distortion. As it was said before, the number of dropped packets may be equal on the average, among the video streams, however the resulting distortion is different for each video stream. Therefore, the channel access differentiation that is applied in this scenario indicates that the video stream with the highest distortion will be considered as high-priority traffic. The tradeoff between the high distortion due the "even-loss distrtibution" scheduling and the low-loss probability due to prioritization ensures that the resulting average distortion of all video streams will be kept minimum.

Table 1: 802.11e MAC parameters.

Access category	AIFS	$CW_{\min}$	$CW_{\max}$	Queue length	Max retry limit
AC3	50	7	15	50	8
AC2	50	15	31	50	8
AC1	50	31	1023	50	4
AC0	70	31	1023	50	4

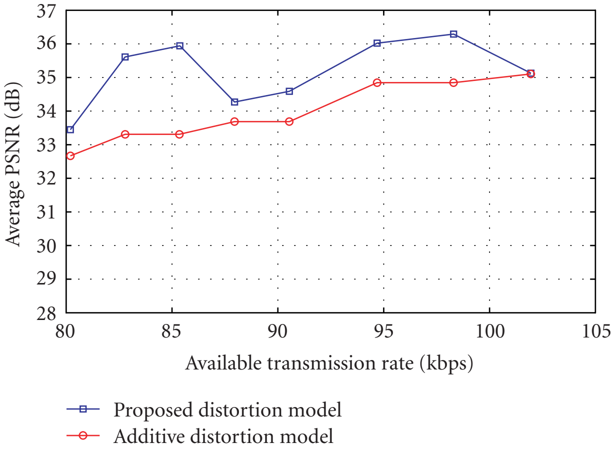


Figure 1: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "even-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *Foreman*.

4.2. Greedy-loss distribution

On the other hand, "greedy-loss distribution" scheduling ensures that the most significant video stream will be less affected by packet scheduling. The algorithm considers packet scheduling jointly for all video streams. Thus, the limited channel capacity available for video transmission will cause a larger number of packets to be dropped by these video senders that their video streams are least significant in terms of context, coding, and QoS characteristics. In our analysis, a video sequence that can be classified as significant for the "greedy-loss distribution" algorithm is a video sequence that the frames are highly correlated with each other, and a loss of one will result to a large distortion effect over all correlated frames. This phenomenon is even more distinct in the case of multireference H.264/AVC coding. A lost reference frame will result in a distortion effect that will not fade out quickly and will affect all the forthcoming frames.

The "greedy-loss distribution" algorithm will cause uneven losses among the multiple video streams. However, this will be balanced by the priority categorization of the 802.11e configuration. Similar to the previous algorithm, the prioritization scheme applied by the system will differentiate the video streams in terms of average distortion due to packet dropping. The tradeoff among the video distortion and the priority categorization has different effects on the video streams. The following section presents the simulation results and includes a discussion over the interesting findings.

5. SIMULATION SETUP AND RESULTS

This section evaluates the performance of the proposed framework through a set of simulations. An NS-2-based simulation environment with the appropriate extensions for simulating 802.11e WLANs is adopted [23]. Three YUV *Quarter Common Intermediate Format* (QCIF) 4:2:0 color video sequences consisting of 300 and coded at 30 frames per second are used as video sources (*Foreman*, *News*, and *Salesman*). Each group of pictures (GOP) is structured as *IPPPPPPP*. and contains 36 frames, and the maximum UDP packet size is at 1024 bytes (payload only). The H.264/MPEG-4 AVC encoder provided in [24] is used for encoding YUV sequences. We simulate a unicast H.264 video transmission (one video server and one video client) utilizing a *basic service set* (BSS) at 1024 Mbps. The video frames are then encapsulated into RTP packets using a simple packetization scheme [25] (by one-frame one-packet policy). The size of each RTP packet is maximally bounded to 1024 bytes. Additionally, the server station generates background traffic (300 Kbps) using constant bit rate (CBR) over UDP, in order to increase the virtual collisions at the server's MAC layer. Moreover, we include four wireless stations where each station generates 300 kbps of data using CBR of data, overloading the simulated 802.11e wireless network. The simulated 802.11e draft defines four access categories; AC3 corresponds to the highest access priority, and AC0 to the lowest. Based on this traffic specification, it is possible to differentiate the H.264 frames at the MAC layer according to distortion.

The generated video packets are delivered through the 802.11e at the form of UDP/IP protocol stack. In our simulation, Table 1 depicts the MAC parameters for the simulations.

5.1. Even-loss distribution scenario

In this scenario, we examine the performance of the proposed distortion model (PDM) and the "*additive*" model of (1) when they are applied in the "even-loss distribution" scheduling algorithm. The available transmission bandwidth is insufficient for the transmission of all three video streams hence the scheduling algorithm is applied to distribute the losses evenly among the streams. The video streams will adapt their transmission rate by selecting to drop the optimum combination of packets. The number of packets to drop is indicated by the "even-loss distribution" algorithm, however which packets will be selected depends on each video source and the distortion model that is used to

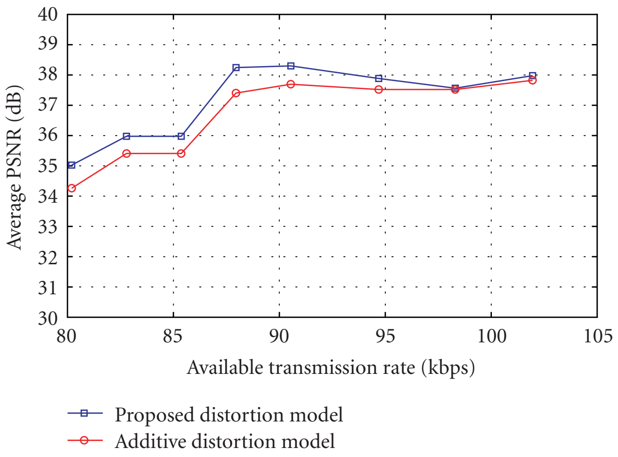


Figure 2: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "even-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *News*.

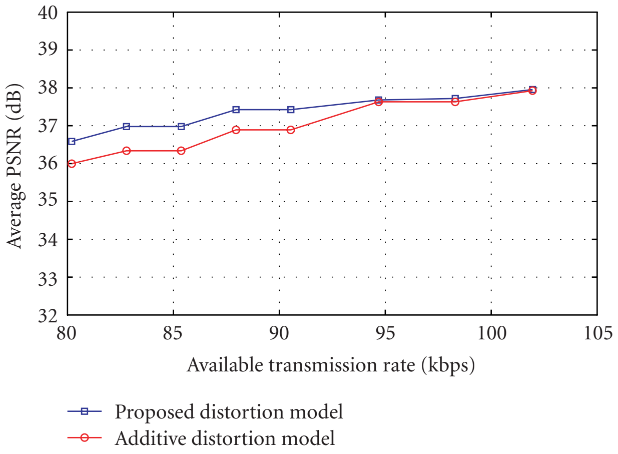


Figure 3: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "even-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *Salesman*.

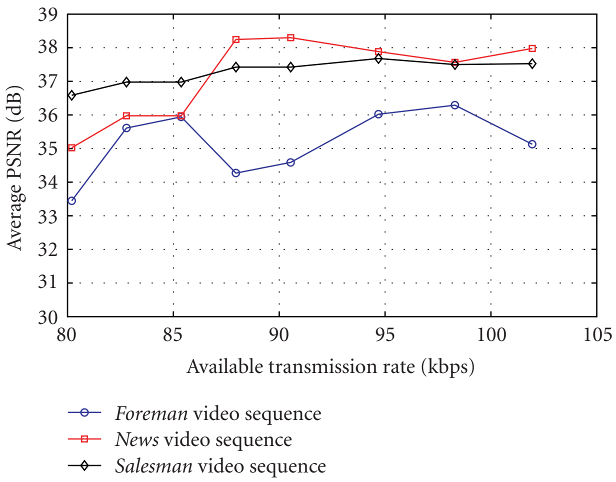


Figure 4: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "even-loss distribution" algorithm. Comparison between the proposed distortion model for all testing video sequences.

pre-calculate packet distortion. In this case, there are two distortion models that are compared, the PDM presented in Section 2 and the "*additive*" model that omits the correlation among the video pictures. Figures 1, 2, and 3 show the overall Y-PSNR (dB) performances of PDM and "*additive*" overall three sequences as a function of the available data rate (kbps) on the shared 802.11e channel. It can be seen that PDM outperforms "*additive*" over the whole range of values considered for the available transmission rate. This is due to the fact that PDM exploits the knowledge about the correlation of referenced frames and can therefore predict accurately the optimum combination of dropping packets that will result in minimum distortion on the reconstructed video. Thus, by applying the PDM, users will drop packets from their transmission windows that will have the least impact on the overall quality of the reconstructed videos. As can be seen from the figures, the performance gain of PDM over "*additive*" is 1 dB on average for the *Foreman* sequence and reaches 2.5 dB when the transmission rate is 85 Kbps. The same applies for the *News* sequence of Figure 2, where on average the distortion due to packet dropping is improved with PDM by 0.5 dB compared with the "*additive*" model. Figure 3 shows that 0.5 dB on the average is the PDM gain against "*additive*" for the *Salesman* sequence as well. Additionally, in Figures 2 and 3, it is evident that as the available transmission increases the two distortion models are converging since the dropped packets are less and the resulting video quality degradation is minimum. This phenomenon is not clear in the case of *Foreman* since the "even-loss distribution" algorithm allocates the number of packets to drop evenly to all video streams. This affects *Foreman* sequence significantly as the context of this particular video contains fast moving pictures and motion scenes with low correlation among video frames. It is evident in Figure 1 that there is an occasion during the simulations that the average PSNR of *Foreman* drops roughly by 2 dB between 85 kbps and 90 kbps. This means that the algorithm chooses to drop packets from this video stream and at the same time increase the quality of the others. This is shown clearly in Figure 4 where all PDM-based streams are plotted together. This figure illustrates the tradeoff between the distortion increase due to packet dropping imposed by the "even-loss distribution" algorithm and the prioritization mechanism of 802.11e that tends to minimize further quality degradation and in some cases even increases the video quality. The *Salesman* sequence has on average a constant PSNR of 37.5 dB over all available rates. On the other hand, the *Foreman* and *News* sequences are competing with each other. When the PSNR of *Foreman* is high (38 dB at 88 kbps) the corresponding PSNR of the *News* is minimum (34 dB at 88 kbps). This is due to the fact that these two sequences

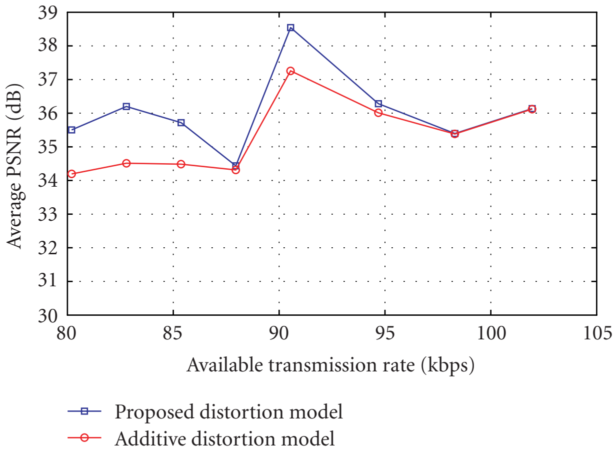


Figure 5: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "greedy-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *Foreman*.

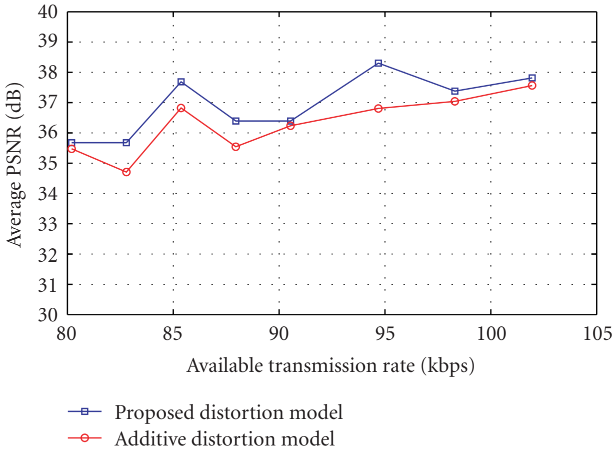


Figure 6: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "greedy-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *News*.

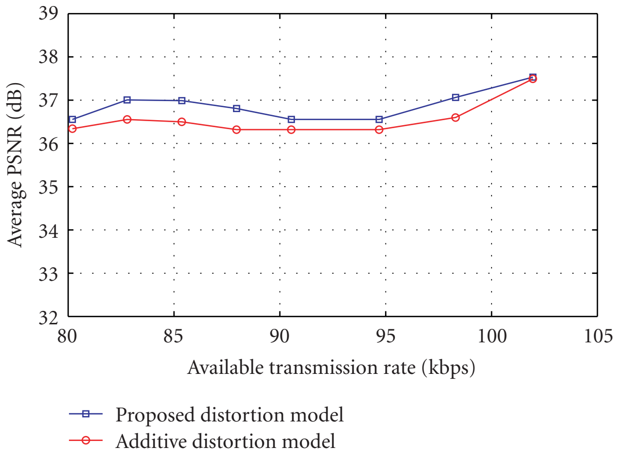


Figure 7: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "greedy-loss distribution" algorithm. Comparison between the proposed and the simplified "additive" distortion model for the video sequence *Salesman*.

have scenes with motion or changing backgrounds; hence the correlation among the reference frames varies accordingly. Therefore, there may be cases where a dropped packet may cause a dramatic effect on the distortion of these two sequences and cases that a dropped packet may not affect the overall distortion significantly. Finally, as it was expected the combination of "even-loss distribution" algorithm and priority categorization provides on the average better quality for *News* and *Salesman* and worse quality for *Foreman* due to its randomly distrtibuted correlations among its frames.

5.2. *Greedy-loss distribution scenario*

In this scenario, we evaluate the performance of the PDM and the "*additive*" model when they are applied to the "greedy-loss distribution" scheduling algorithm and the 802.11e channel access differentiation scheme. The difference with the previous scenario is that in this case the "greedy-loss distribution" algorithm allocates the number of dropped packets unevenly to the three video streams. Since a video sequence, demanding in terms of quality context, is greater affected even by small losses, the logarithm will tend to allocate more transmission bandwidth to this video stream. However, this uneven loss distribuion will be balanced by the prioritization scheme that will improve the quality of the most distorted video stream. The simulation results are shown in Figures 5, 6, and 7. In all cases, the PDM outperforms the "*additive*" model for up to 1.5 dB for *Foreman* at 82 Kbps, 1.5 dB for *News* at 95 Kbps, and 0.5 dB for *Salesman* at 85 Kbps. In addition to that, by comparing Figures 1 and 5, the *Foreman* PSNR for the "even-loss distribution" and the "greedy-loss distribution" algorithms, it is evident that the latter improves the total distortion of the video. However, the same applies less to *News* and does not apply to *Salesman*. These two sequences have been positively affected by the "even-loss distribution" algorithm compared to *Foreman*, but the "greedy-loss distribution" scheduling has improved the distortion effect of *Foreman* sequence as it is the most demanding in terms of context of the three. Figure 8 shows the PDM-based "greedy-loss distribution" algorithm and prioritization plots for all three video streams. The improvement on *Foreman* PSNR compared to Figure 4 is evident.

6. CONCLUSIONS

This paper proposes a novel framework based on two scheduling algorithms named "even-loss distribution" and "greedy-loss distribution" for effective packet scheduling and optimized differentiated channel access that minimizes the overall distortion of the video streams. The framework is based on a new distortion prediction model which takes into consideration the high correlation among reference frames of H.264/AVC. The proposed distortion model has been evaluated against real measurements and is compared with the

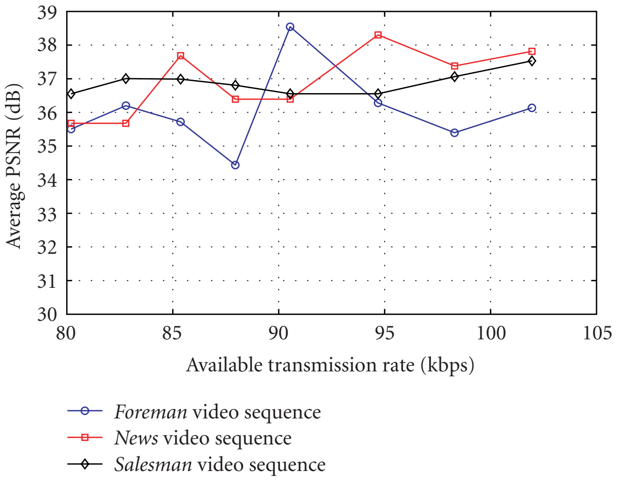


Figure 8: Average PSNR (in dB) as a function of available transmission rate (in kbps) for the "greedy-loss distribution" algorithm. Comparison between the proposed distortion model for all testing video sequences.

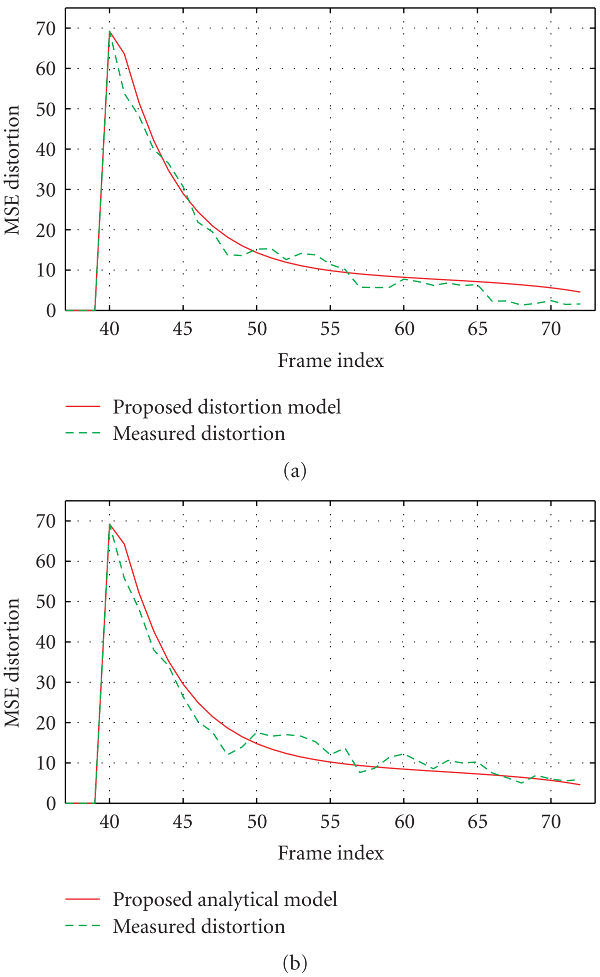


Figure 9: Single frame loss and error propagation. (a) Reference frame list size $M_{\text{REF}} = 5$. (b) Reference frame list size $M_{\text{REF}} = 10$.

simplified "*additive*"-based distortion model that omits the correlation among the frames. The "even-loss distribution" algorithm aims to allocate the packet losses evenly to all video streams. The scheduling algorithm indicated the number of packets that should be dropped by each video source in order to adapt its rate to the available transmission bandwidth. However, which packets will be selected depends on the precalculated packet distortion based on the PDM. The imposed distortion due to packet dropping is balanced by a modified 802.11e system that supports video stream categorization to different priority classes based on the average distortion. The simulations show that this scheme can improve the video distortion of the reconstructed video at the receiver end. However, it also shows that demanding video sequences in terms of context are affected more significantly by this approach. The "greedy-loss distribution"algorithm on the other hand, allocates more transmission rate to video streams with complex context and video scenes. Hence, the imposed distortion due to packet dropping is kept minimum, although the prioritization mechanism will treat these video streams as low priority traffic. Simulations show that on average the latter scenario is more effective for all streams, especially effective for demanding sequences with randomly distributed dependences among the frames of the sequence.

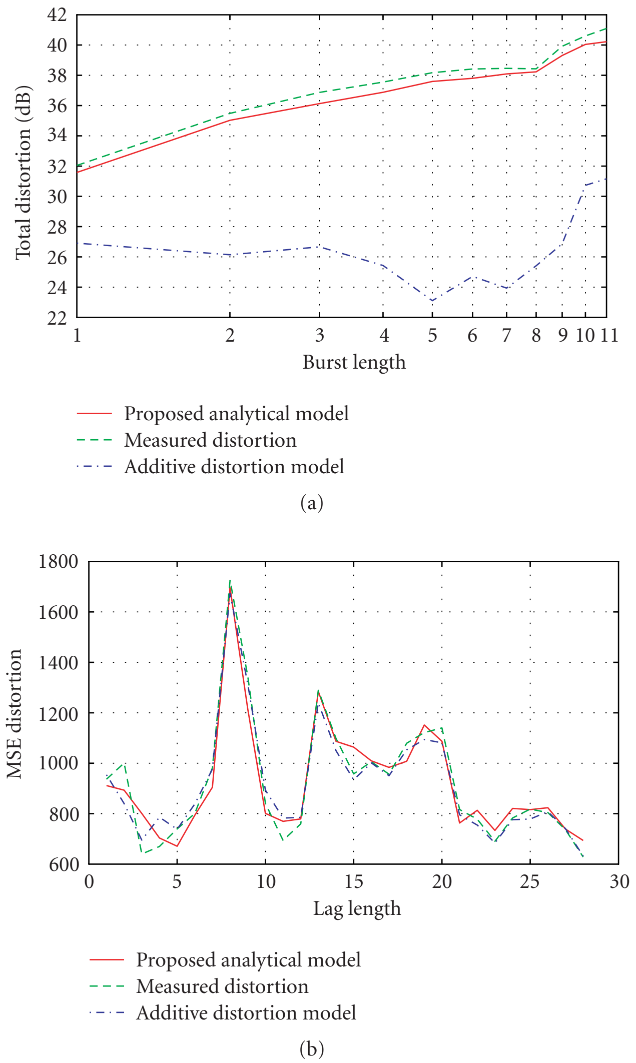
APPENDIX

A. EVALUATION OF DISTORTION MODEL

The proposed distortion model (8) is verified through extensive simulations of a standard video test sequence and is compared with actual measurement and the additive distortion model previously described. Coding and decoding is performed by using JM12 of the H.264/AVC video compression standard [24]. The selected test video sequence is *Foreman* at QCIF resolution with 300 video frames at 30 fps and a constant quantization step that results in an average PSNR of 36 dB. The size of the reference frames list varies from 5 to 10 frames. According to H.264 standard, the selected reference frames are frames that have been encoded and stored prior to the current frame. Every 4 frames, a slice is intra-updated, corresponding to an intraframe update period of $N = 4 \times 9$.

Figure 9 illustrates the distortion effect for the loss of a single frame in two cases, where 5 and 10 frames are used as reference frames, respectively. The comparison between the proposed model and the measured distortion clearly indicated that the proposed model accurately captures the distortion effect as long as there are reference frames stored in the decoder for motion vector prediction.

Figure 10(a) illustrates the MSE plotted for different burst error lengths. It clearly shows that the distortion due to varying burst error length is not equivalent to the sum of isolated losses (1), which is also consistent with [13]. Apparently, the proposed model is very accurate in the calculation of the distortion, allowing only adeviation from the actual data of ±0.4 dB. Moreover, Figure 10(b). plots the total distortion due to two frame errors separated by a lag, where the first erroneous frame is Frame 38. It can be shown that the proposed model accounts for the resulted distortion very ac-



(a)

(b)

FIGURE 10: (a) Total distortion versus burst error length. (b) Total distortion of two losses with lag (first error frame at 38 and the second 38 + lag.)

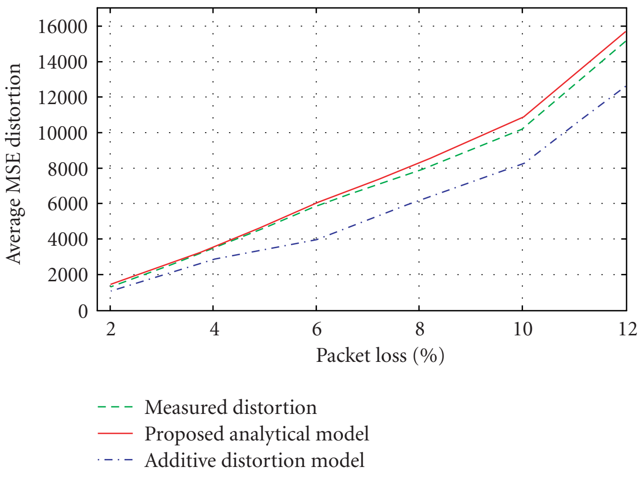


FIGURE 11: Average total distortion versus packet loss.

curately, as it considers the correlation between each pair of error frames as opposed to the additive model.

Finally, in Figure 11 the averaged distortion is plotted for different packet losses. Each packet loss is the result of different error frame patterns. In order for the model to be precise, up to 10 000 random combinations of the 100 video frames have been used for the calculation of the actual video distortion and the proposed recursive approach of (6). It is clear that the model follows the actual distortion very closely even at high packet losses. In particular, at packet loss beyond 9% the distortion model deviates from the actual measurements approximately by 1000 in MSE distortion.

ACKNOWLEDGMENT

This work was supported by the project PENED no. 03636, which is funded in 75% by the European Social Fund and in 25% by the Greek State-General Secretariat for Research and Technology.

REFERENCES

[1] J. Chakareski and P. Frossard, “Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources,” *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 207–218, 2006.

[2] IEEE 802.11g, “Part11: wireless LAN medium access control (MAC) and physical layer (PHY) specification band,” Supp. IEEE 802.11, 2003.

[3] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[4] J. Deng and R.-S. Chang, “A priority scheme for IEEE 802.11 DCF access method,” *IEICE Transactions on Communications*, vol. E82-B, no. 1, pp. 96–102, 1999.

[5] I. Aad and C. Castelluccia, “Differentiation mechanisms for IEEE 802.11,” in *Proceedings of the 20th Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM ’01)*, vol. 1, pp. 209–218, Anchorage, Alaska, USA, April 2001.

[6] J.-Y. Yeh and C. Chen, “Support of multimedia services with the IEEE 802.11 MAC protocol,” in *IEEE International Conference on Communications (ICC ’02)*, vol. 1, pp. 600–604, New York, NY, USA, April-May 2002.

[7] IEEE 802.11e, “Wireless LAN medium access control (MAC) enhancements for quality of service (QoS),” *802.11e draft 8.0*.

[8] F. Wu, G. Shen, K. Tan, F. Yang, and S. Li, “Next generation mobile multimedia communications: media codec and media transport perspectives,” *China Communications Magazine*, vol. 4, no. 5, pp. 30–44, 2006.

[9] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*, IEE Press, London, UK, 2003.

[10] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, 2000.

[11] I.-M. Kim and H.-M. Kim, “A new resource allocation scheme based on a PSNR criterion for wireless video transmission to stationary receivers over Gaussian channels,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 3, pp. 393–401, 2002.

[12] J. Apostolopoulos, W.-T. Tan, S. Wee, and G. Wornell, "Modeling path diversity for multiple description video communication," in *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP '02)*, vol. 3, pp. 2161–2164, Orlando, Fla, USA, May 2002.

[13] Y. Liang, J. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: does burst-length matter?" in *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP '03)*, vol. 5, pp. 684–687, Hong Kong, April 2003.

[14] F. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network*, vol. 15, no. 6, pp. 40–54, 2001.

[15] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 84–97, 2005.

[16] S. R. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 983–1001, 1997.

[17] P. Bucciol, G. Davini, E. Masala, E. Filippi, and J. C. De Martin, "Cross-layer perceptual ARQ for H.264 video streaming over 802.11 wireless networks," in *IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 5, pp. 3027–3031, Dallas, Tex, USA, November-December 2004.

[18] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

[19] W. Tu, W. Kellerer, and E. Steinbach, "Rate-distortion optimized video frame dropping on active network nodes," in *Proceedings of the International Packet Video Workshop*, Irvine, Calif, USA, Decemeber 2004.

[20] T. Wiegand, G. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[21] H. L. Truong and G. Vannuccini, "The IEEE 802.11e MAC for quality of service in wireless LANs," in *Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR '03)*, L'Aquila, Italy, January 2003.

[22] ISO/IEC 14496-10, ITU-T Rec., and H.264 standard, "Advanced video coding," 2003.

[23] "Network simulator v.2 (ns-2) DCF and EDCA extensions," http://www.tkn.tu-berlin.de/research/802.11e_ns2/.

[24] "H.264 reference software version," Image Processing Department, Fraunhofer-Institute, http://ip.hhi.de/.

[25] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "Rtp: a transport protocol real-time applications," *RFC 1889*, January 1996.