

Optimized H.264/AVC-Based Bit Stream Switching for Mobile Video Streaming

Thomas Stockhammer,¹ Günther Liebl,² and Michael Walter²

¹Nomor Research GmbH, Tannenweg 25, 83346 Bergen, Germany

²Institute for Communications Engineering (LNT), Munich University of Technology (TUM), 80290 Munich, Germany

Received 12 August 2005; Revised 17 February 2006; Accepted 30 April 2006

In this work we show the suitability of H.264/MPEG-4 AVC extended profile for wireless video streaming applications. In particular, we exploit the advanced bit stream switching capabilities using SP/SI pictures defined in the H.264/MPEG-4 AVC standard. For both types of switching pictures, optimized encoders are developed. We introduce a framework for dynamic switching and frame scheduling. For this purpose we define an appropriate abstract representation for media encoded for video streaming, as well as for the characteristics of wireless variable bit rate channels. The achievable performance gains over H.264/MPEG-4 AVC with constant bit rate (CBR) encoding are shown for wireless video streaming over enhanced GPRS (EGPRS).

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

1. INTRODUCTION

High-quality video streaming is becoming a killer application in wireless systems. For this type of systems, compression efficiency, as well as adaptivity, are the most important features when selecting appropriate video codecs. The recently standardized H.264/MPEG-4 AVC codec (denoted as H.264/AVC in the following) provides both features, but especially the latter has not been discussed in too much detail up to now. Adaptivity allows reacting to dynamics in the system resulting from bursty traffic patterns, variable receiving conditions, as well as handovers and random user activity. Due to the commonly used error control features on wireless links, these variations mainly result in varying bit rates. However, it is important to understand that the variability cannot be attributed to a single effect and also underlies different time scales: typical variations are within a few milliseconds due to short-term fading and interference, within a few hundred seconds due to shadowing effects, within a few seconds due to changes in the receiver position, as well as within larger scales due to handover and changes in the overall system load. In case of online encoding, if the encoder has sufficient feedback, control strategies for variable bit rate (VBR) channels can be applied [1]. Hence, the encoder rate control dynamically adapts to changing bit rates [2].

For preencoded sequences, however, other means are necessary: in case of short-term channel bit rate variations, play out buffering at the receiver can compensate for bit rate fluctuations such that the display timeline is maintained.

For example, in [3] it has been shown that for UMTS-like channels the bit rate variations due to link layer retransmissions can be well compensated by receiver buffering without adding significant additional delay. In addition, in case of anticipated buffer underrun, techniques such as adaptive media play out [4] enable a streaming media client, without the involvement of the server, to control the rate at which data is consumed by the play out process.

Nevertheless, in many cases, play out buffering and adaptive media play out might not be sufficient to compensate for bit rate variations in wireless channels. Hence, rate adaptation of preencoded streams has to be performed by modifying the encoded bit stream. This adaptation can be carried out at different instances in the network: at the streaming server, in intermediate routers, or at the entry gateway to the wireless access network. Different methods are, for example, discussed in [5, 6]. Usually, one can assume that backbone networks are over provisioned such that the primary bottleneck is the wireless link. On the one hand, it is thus more likely that closer to the air interface, there exists more up-to-date channel state information about the expected transmission conditions which would allow making better decisions. On the other hand, a streaming server usually includes much more intelligence to react to variable bit rates than intermediate routers or gateways: the latter usually only drop packets in case of congestion without taking into account their individual importance, which results in error propagation. In this case bit rate adaptivity is equivalent to packet loss

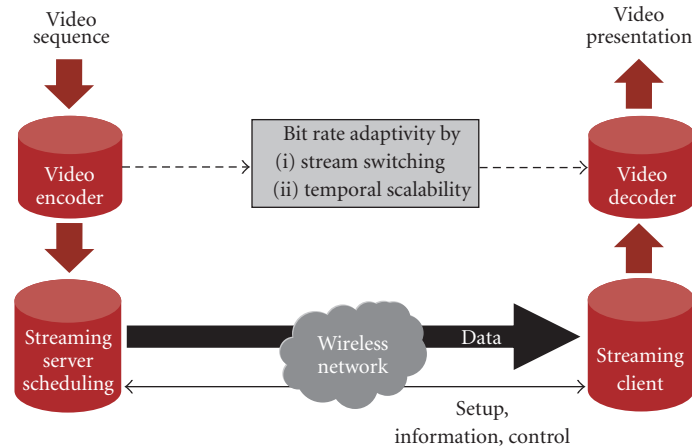


FIGURE 1: System overview.

resilience—features included in H.264/AVC for this purpose are discussed, for example, in [7].

In this work we assume that our rate adaptation entity—referred to as *scheduler*—has sufficient information and intelligence to be able to drop packets with respect to their relative importance. A formalized framework under the acronym rate-distortion optimized packet scheduling has been introduced [8] and serves as the basis for several subsequent publications. Obviously, this strategy requires a regular syntax, that is, by defining more and less important packets in a stream. Hence, if bit rate variations on the transmission path are expected, it is wise to preencode media streams with appropriate packet dependencies, such that the importance of the packets in the stream can be easily differentiated by the network components. The H.264/AVC standard already offers some options to support packets with different importance for bit rate adaptivity. However, a scalable extension, which will also include classical SNR-scalability, is still under discussion [9] and will not be considered here. Our proposed streaming system will thus rely on three different means for bit rate adaptivity, namely, (i) play out buffering, (ii) temporal scalability, and (iii) advanced bit stream switching.

The remainder of this paper is structured as follows: we will start with a brief overview of an end-to-end wireless video streaming system in Section 2. Next, we will introduce the various features available in H.264/AVC to support temporal scalability and bit stream switching in Section 3. We will present suitable encoding solutions for these features and develop an abstract framework for describing video streaming over arbitrary VBR channels. Section 4 then deals with a specific class of VBR channels, which result from including a wireless link in the end-to-end transmission chain. We will discuss several mathematically tractable models of different complexity to describe the influence of wireless links on packet transmission. For the system considered in this work, namely EGPRS, we will propose a relatively simple, yet sufficiently accurate description of the channel characteristics. In Section 5, we will integrate the previously developed concepts into an optimized decision making strategy for the se-

lection of frames and versions in a wireless streaming scenario. Experimental results for H.264/AVC video streaming over EGPRS links will demonstrate the applicability of our strategy in Section 6. The paper concludes with some general remarks and a summary of future work topics.

2. SYSTEM OVERVIEW

Figure 1 shows a simplified wireless streaming system, which usually consists of an end-to-end connection between a media streaming server and a client. The latter requests pre-encoded data stored at the server to be streamed to the end user. The client buffers the incoming data and starts with decoding and presentation of the reconstructed video sequence after some initial delay. Once playback has started, a continuous presentation of the sequence should be guaranteed. For CBR channels with constant delay successful play out can be guaranteed by encoding and streaming of the video sequence such that the resulting bit stream contains a leaky bucket [10].

However, in our investigated system neither the bit rate nor the delay is constant, and some data units are not even available at the decoder. Therefore, the media streams stored at the server have to be not only compression efficient, it should also be possible to flexibly adapt their bit rate to varying conditions on the wireless link.

H.264/AVC, in addition to its compression efficiency, also provides means for bit rate adaptivity: the flexible reference frame concept in combination with generalized B-pictures allows a huge flexibility on frame dependencies, which can be exploited for temporal scalability and rate shaping of pre-encoded video. For example, the rate can easily be adapted by dropping nonreference frames, which does not result in error propagation. This H.264/AVC operation mode is equivalent to temporal scalability. Furthermore, sequences could be encoded such that, for example, less important background is dropped in favor of a more important foreground scene [11]. However, very often it is still necessary to further adapt the bit rate in the application, usually in larger bit rate scales, as well as in time scales larger than the initial play out delay. In

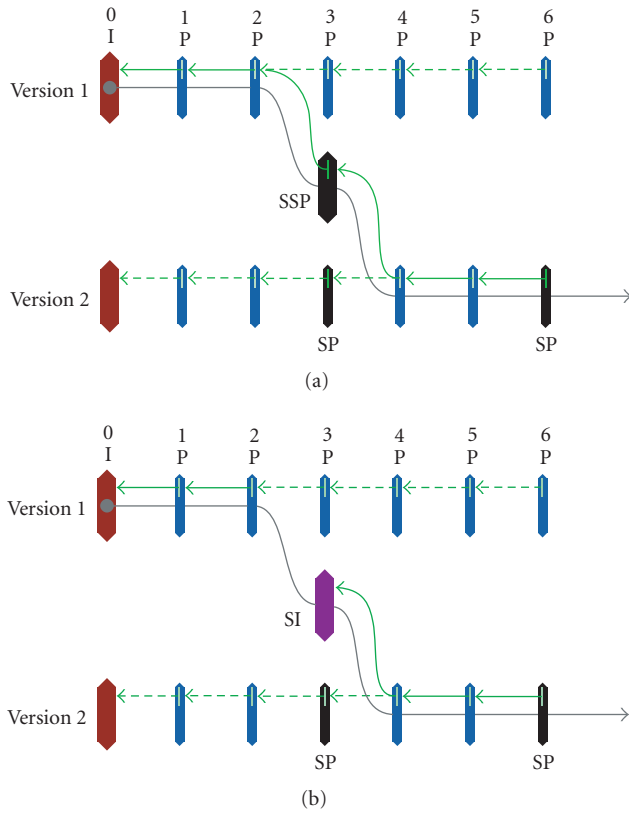


FIGURE 2: Bit stream switching with SP- and SI-pictures in H.264.

this respect, it has been recognized that the bit rate on wireless links is a precious resource, especially when compared to storage on servers. Finally, most applications provide sufficient buffer feedback, as well as channel state information, such that the streaming server has at least an estimate of the supported bit rate. Under these common premises bit stream switching provides a simple, yet powerful, means to support bit rate adaptivity in wireless streaming environments. In this case the streaming server stores the same content encoded with different versions in terms of rate and quality. Each of these versions must include means to randomly switch into it. Instantaneous decoder refresh (IDR) pictures provide this feature, but they are also costly in terms of compression efficiency (for an analysis of bit stream switching for streaming, see [12]).

The switching predictive (SP) picture concept in H.264/AVC [13], however, is more adequate for this purpose: in this case the streaming server not only stores different versions of the same content, but also secondary SP-pictures, as well as SI-pictures. As long as the bit rate does not change, efficient primary SP-pictures are transmitted at the pre-selected possible switching points. If switching becomes necessary, one can rely on secondary SP- or SI-pictures. Some preliminary work on bit stream switching using the SP-picture concept for congested links has been presented in [14].

In Figure 2, a simplified switching scenario is depicted with only two preencoded versions 1 and 2. An extension to more than two versions is straightforward, but is omitted

here for the sake of clarity. These two versions result from encoding of the same original video sequence with two different quantization parameters. Primary SP-pictures have been used periodically at identical positions in both sequences. Thus, at every “SP-position” either the primary is transmitted, if no switching happens, or the secondary (either SSP or SI) is transmitted in case of switching.

In this work we will consider a wireless video streaming environment which employs a central unit at the transmitter, referred to as *scheduler*. The latter has access to information about all source data to be transmitted next, as well as to information on current expected transmission conditions. The scheduler attempts to optimize its decision which packets, as well as which versions, are to be transmitted next. The accessible source and channel information will be specified in more detail in the following two sections, and the proposed scheduler is presented in Section 5.

3. A FRAMEWORK FOR VBR STREAMING OF H.264/AVC VIDEO

3.1. Preliminaries of the SP-picture concept

The SP-picture concept allows applying predictive coding even in case of different reference signals by performing the motion-compensated prediction (MCP) process in the transform domain rather than in the spatial domain. The reference frame is quantized—usually with a finer quantizer than that used for the original frame—before it is forwarded to the reference frame buffer. The resulting so-called primary SP-pictures are placed in the encoded bit stream at the pre-selected possible switching points. In general, they are slightly less compression-efficient than regular P-pictures, but significantly more efficient than regular IDR-pictures. The major benefit results from the fact that the quantized reference signal can be generated mismatch-free using any other prediction signal. In case that this reference signal is generated by predictive coding, the picture is referred to as secondary SP (SSP) picture. They are usually significantly less efficient than P-pictures, as an exact reconstruction is necessary. To generate the reference signal without any previous dependencies, the so-called switching-intra (SI) pictures can also be used, which are only slightly less inefficient than common I-pictures, but can also be used for adaptive error resilience purposes. For more details on this unique feature within H.264/AVC the interested reader is referred to [13].

3.2. An optimized encoder for SSP/SI-pictures

An encoder realization for generating primary SP-pictures is already included in the H.264/AVC test model software. In addition, we have developed an optimized encoder for SSP-pictures, as well as for SI-pictures. The respective encoder structure for SSP-pictures is shown in Figure 3. Here, lower-case letters (e.g., l) denote quantized signals, while capital letters (e.g., L) denote nonquantized signals. Furthermore, signals in the transform domain are indicated by the letter “ l ,” while signals in the pixel domain are indicated by the letter

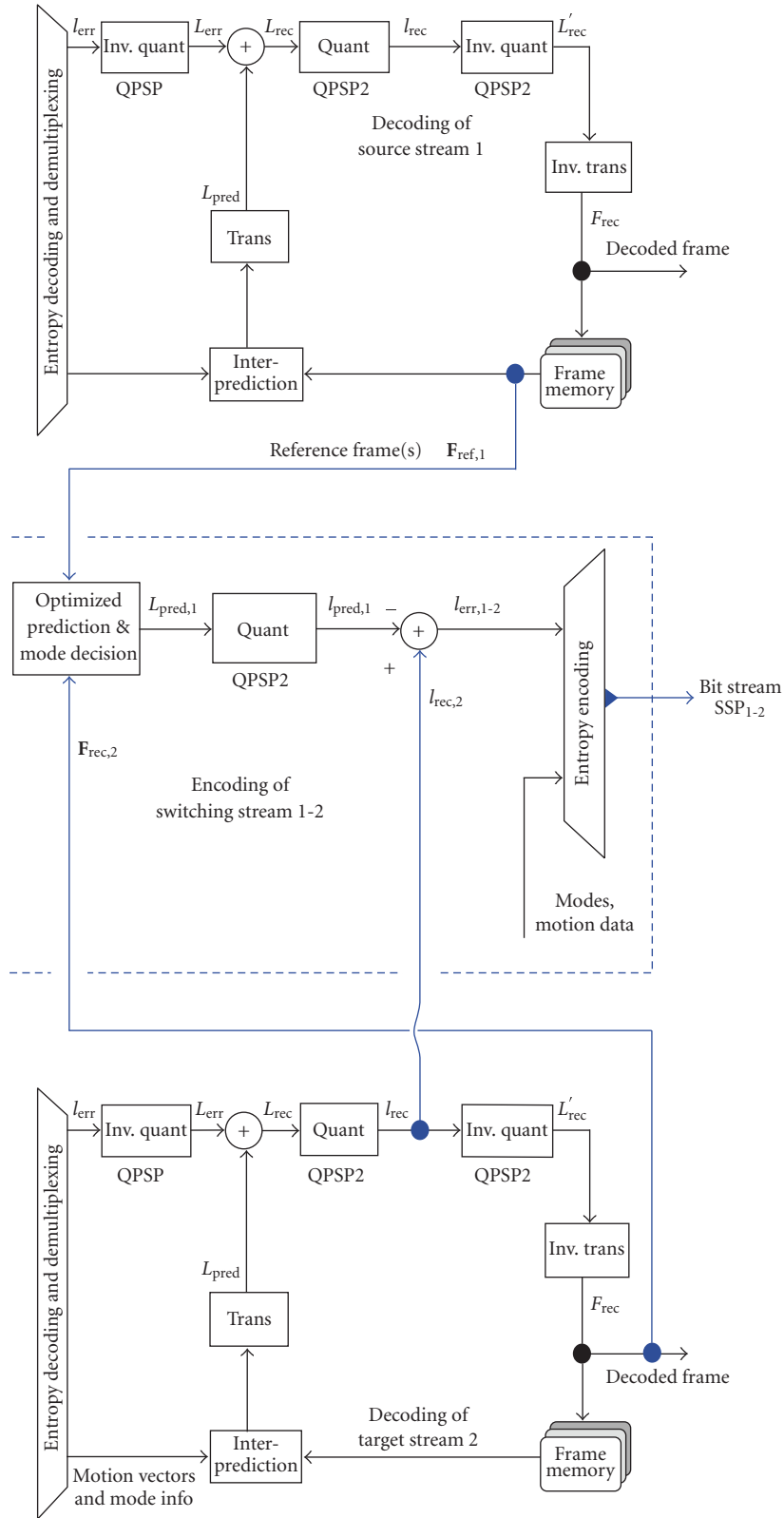


FIGURE 3: Optimized secondary SP-picture encoder.

“ f .” The individual meaning of a signal (e.g., pred for “predicted”) can be derived from its index.

According to Figure 3 we obtain the SSP-picture for switching from source stream 1 to target stream 2 by extracting and combining information from both runs. The encoding process for the secondary representations depends on the signal $l_{\text{rec},2}$ that is generated in the encoding and decoding process of the primary target SP-picture. We decided to use the decoding process of target stream 2 for exporting $l_{\text{rec},2}$ as shown in Figure 3. SSP-encoding also requires the prediction signal $L_{\text{pred},1}$. In our implementation, $L_{\text{pred},1}$ is generated using all reference frames $F_{\text{ref},1}$, which are available by decoding source stream 1. For SI-pictures the same concept applies with the only difference that the prediction signal can be computed without any signals exported from stream 1.

It is also worth mentioning that the straightforward approach to simply use the prediction signal, motion vectors, and modes from encoding/decoding the primary source stream 1 is not efficient: the partition modes and the motion vectors chosen for encoding the source primary SP-picture do not necessarily fit well for encoding the SSP and result in a suboptimal prediction signal with a large prediction error $l_{\text{err},1 \rightarrow 2}$. This implies that coding efficiency is low, as the residual has to be encoded without any further quantization. Hence, a prediction signal $L_{\text{pred},1}$ is required which minimizes the residual. Since no restrictions apply on $L_{\text{pred},1}$, we can optimize it by using all available reference frames $F_{\text{ref},1}$. Classical rate-distortion optimization [15], as used in the JM test model, is applied. However, the encoded SSP will be identical to the primary SP-reconstruction of the target stream. The goal of the motion estimation and compensation must therefore be to match the reconstructed primary target frame $F_{\text{rec},2}$, rather than the original frame F_{orig} . With this modified mode selection we save up to 10% in bits for SSP-picture coding compared to the case when we use the prediction signal optimized to F_{orig} . The gains compared to the nonoptimized approach using the prediction signal $L_{\text{pred},1}$, for which the frame sizes often exceed or equal those for SI-pictures, are in the order of 100–400%. For details on encoding results, the exact encoder implementation, as well as on guidelines for the selection of quantization parameters for primary and secondary representations, we refer to [14, 16].

3.3. General abstraction of the encoding, transmission, and decoding processes

Efficient streaming media algorithms require a formalized description of the encoded multimedia data to be able to make good decisions during the transmission process [8]. Assume that *source* units f_n , $n = 1, \dots, N$ (i.e., video frames), are encoded and mapped one-to-one onto *data units* \mathcal{P}_n (i.e., packets). Any advanced packetization modes, such as flexible macroblock ordering, slice structured coding, or packet interleaving schemes, are not considered here. Note, however, that our framework is general enough to include such concepts. In addition, we assume that for each source unit f_n we generate several versions $v = 1, \dots, V$, which are represented by individual data units $\mathcal{P}_{n,v}$. The reconstructed version of

each source unit is denoted as $\hat{f}_{n,v}$. Furthermore, we define a quality measure $Q(f, \hat{f})$ reflecting the rewards/costs when representing f by \hat{f} .

Each source unit (and hence each data unit) has assigned a decoding time stamp (DTS) T_n representing the latest time instant the data unit \mathcal{P}_n must be decoded to be useful. The decoding time is relative to T_1 , which is assumed to be 0 without loss of generality. Data unit indices are ordered with increasing DTS T_n . According to [8], video encoding and packetization can then be represented as a directed acyclic graph. However, note that this only holds for the data units within one version. An extended framework for different versions is not addressed in [8]. We restrict ourselves in the following to the practical case where the graph for each version is of identical structure. Again, generalization to different structures for each version is straightforward, but the benefit in terms of encoding efficiency needs to be carefully considered. To specify decoding dependencies among data units, we write $n' < n$ if $\mathcal{P}_{n'}$ is necessary to decode \mathcal{P}_n .

When transmitting a stream to a client, a server may select an appropriate version vector $\mathbf{v} = \{v_n\}_{n=1}^N$, with v_n the version chosen for each f_n . Hence, with this definition any arbitrary stream-switching strategy is possible, since different versions may be transmitted for each successive data unit. However, for our strategy we apply restrictions on version vector elements to avoid the problem of reference frame mismatches: since switching is only allowed at I- or SP-picture positions, versions can only change at these positions as well.

Assume now that we operate in an environment where not necessarily all data units are received at the media decoder. In this case, concealment has to be done for any representation of a missing data unit. In the remainder we apply the common “freeze-picture” concealment, that is, missing data units are represented by the timely nearest available source unit. Note that while the encoder only considers this type of error concealment in the optimization process, our decoder does actually apply this strategy. The index of the first candidate to conceal source unit f_n is denoted by the concealment index $c(n)$. If there is no preceding source unit, for example, I-pictures, we assume that the lost source unit is concealed with a standard representation, for example, a grey image (denoted as $c(n) = 0$).

In case of consecutive data unit loss, concealment is applied recursively. Assume that $c(n) = i$. If data unit \mathcal{P}_i is also lost, the algorithm uses source unit f_j to conceal f_i , that is, $c(i) = j$. To avoid any lengthy recursive notation we simply use $j \vdash n$ to express the fact that source unit f_n is eventually concealed with unit f_j . The resulting concealment dependencies can also be expressed by a directed graph. Figure 4 shows an example of possible frame dependencies and the corresponding concealment graph.

3.4. Importance definition

To allow prioritization of different data units and also of different versions over others, the importance of a single data unit for the overall reconstruction quality needs to be quantified. The previous definitions and the abstraction of

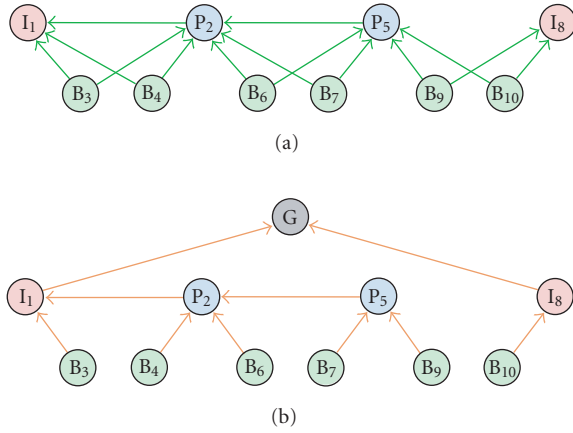


FIGURE 4: Frame dependencies and concealment graph.

the encoding, transmission, and decoding processes lead to the definition of the so-called *importance* of each data unit $\mathcal{P}_{n,v}$: the latter reflects the amount by which the quality at the receiver increases if the data unit is correctly decoded and can be written as

$$I_{n,v} \triangleq \frac{1}{N} \left(Q(f_n, \hat{f}_{n,v}) - Q(f_n, \hat{f}_{c(n),v}) + \sum_{\substack{i=n+1 \\ n-i}}^N [Q(f_i, \hat{f}_{n,v}) - Q(f_i, \hat{f}_{c(n),v})] \right). \quad (1)$$

The importance definition takes into consideration the quality of data unit $\mathcal{P}_{n,v}$, the chosen concealment strategy, as well as the dependency and concealment graph. In other words, the importance quantifies the improvement in quality if the source unit contained in $\mathcal{P}_{n,v}$ is displayed instead of the concealment source unit $f_{c(n),v}$ for this unit, as well as for all other source units for which f_n is eventually used for concealment.

3.5. Received and expected quality

The end-to-end performance of a streaming media system strongly depends on the versions chosen (expressed by the version vector \mathbf{v}) and the amount and importance of packets not available at the decoder. To be more specific, we define the observed channel behavior at a streaming client for data unit $\mathcal{P}_{n,v}$ as $c_n \triangleq \mathbf{1}\{\text{data unit } \mathcal{P}_{n,v} \text{ available}\}$. Here, $\mathbf{1}\{A\}$ denotes the indicator function being 1, if A is true, and 0 otherwise. Hence, the combination of a certain observed channel sequence $\mathbf{c} = \{c_1, \dots, c_N\}$ with (1) and the concealment strategy as introduced above yields the following expression for the (actual) received quality:

$$Q(\mathbf{c}, \mathbf{v}) \triangleq Q_0 + \sum_{n=1}^N I_{n,v} c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m. \quad (2)$$

Here, $Q_0 \triangleq (1/N) \sum_{n=1}^N Q(f_n, \hat{f}_0)$ denotes the minimum quality, if instead of the original sequence all pictures are presented as grey. The latter is obviously quite hypothetical, but it is necessary to have a comprehensive framework. In summary, in order to benefit from data unit \mathcal{P}_n , it is necessary that all data units \mathcal{P}_m it depends on are also available at the receiver. For a proof that (2) actually corresponds to the received quality given the above assumptions, we refer to Appendix A.

The importance of each data unit and version is quite easily computed during the encoding process. As a consequence, (2) significantly simplifies the simulation of video streaming systems, as the achievable quality at the simulated media clients can be determined via linear combination of the channel vector and the importance of the selected versions of each data unit. Any decoding of erroneous video streams is thus not necessary.

The practical importance of (2) for system optimization, however, is rather limited, since in wireless transmission systems, the channel behavior is in general not deterministic. Nevertheless, the notion of importance can be used quite effectively at the transmitter for simple computation of the expected quality (at the receiver), as will be shown in the following: a certain data unit might be lost entirely or might arrive too late at the receiver such that the decoding of the data unit is no more useful due to expired deadlines (we assume here that the client does not use any advanced strategies, such as rebuffering). The channel behavior sequence $\mathbf{C} \triangleq \{C_1, \dots, C_N\}$ is in general random, with $C_n \in \{0, 1\}$ the random variable indicating whether data unit n is received successfully ($C_n = 1$) or lost ($C_n = 0$). Therefore, not only the channel is random, but also the received quality, denoted as $Q(\mathbf{C})$: for certain channel realizations we obtain a good quality, whereas for others the received quality is much worse.

In the following we are interested in a single measure to compare the different transmission strategies. The most obvious and suitable measure is the expected quality $E\{Q(\mathbf{C})\}$. The following equation provides a definition of the expected received quality, as well as a simplified method to derive it:

$$\begin{aligned} E\{Q(\mathbf{C})\} &\triangleq \sum_{\mathbf{c} \in \{0,1\}^N} Q(\mathbf{c}) \Pr\{\mathbf{C} = \mathbf{c}\} \\ &= Q_0 + \sum_{n=1}^N I_n \Pr\{C_n = 1 \mid \forall_{k < n} C_k = 1\} \\ &\quad \cdot \prod_{\substack{m=1 \\ m < n}}^{n-1} \Pr\{C_m = 1 \mid \forall_{k < m} C_k = 1\} \\ &= Q_0 + \sum_{n=1}^N I_n \Pr\{C_n = 1 \wedge \forall_{k < n} \Delta_k = 1\}. \end{aligned} \quad (3)$$

Note that the expectation in this case is only over the channel statistics \mathbf{C} . For a proof of the various equalities in (3), we refer to Appendix B.

3.6. Summary: media abstraction for video streaming over VBR channels

With these preliminaries we are able to develop an effective abstraction of streamed media data. For channels which exhibit data unit loss (as will be considered in the remainder of this work), it is sufficient to know the number of encoded source versions V , the initial quality Q_0 , and the following metrics for each data unit $n = 1, \dots, N$ and each version $v = 1, \dots, V$:

- (i) the importance $I_{n,v}$,
- (ii) the data unit size $R_{n,v}$ in bytes,
- (iii) the decoding time stamp T_n , and
- (iv) the dependencies expressed by the index of the directly preceding data unit(s) of \mathcal{P}_n .

Furthermore, for each SP-picture in each version v , the data unit size $R_{n,v \rightarrow v'}$ of the SSP-picture when switching to version v' and the SI-picture size are required [16]. As already mentioned, this abstract description can be used on the one hand to effectively simulate video streaming over lossy channels (via (2)). On the other hand, (3) or one of its variants provides a means to optimize the transmission schedule, as will be shown in Section 5.

4. A FRAMEWORK FOR THE DESCRIPTION OF WIRELESS LINKS

4.1. General characteristics and modeling aspects

Wireless channels are becoming increasingly important as a transport medium for various types of multimedia information. While the appeal of tetherless mobility is great, numerous issues need to be resolved in order for wireless transport of real-time multimedia data to become reality (including communications issues, low-power implementations, etc.). In this work we consider a scenario where due to the user's mobility the channel behavior will be inherently time-varying, with periods of higher data rates alternating with periods of lower rates.

In general, the available bandwidth and, therefore, the bit rate over the radio link are limited. In addition, the mobile environment is characterized by harsh transmission conditions in terms of attenuation, shadowing, fading, and multiuser interference, which result in time- and location-dependent channel conditions. New directions in the design of wireless systems do not necessarily attempt to minimize the error rates in the system, but to maximize the system throughput. This is especially attractive for services with relaxed delay constraints, such as file downloads and streaming applications. The nonergodic behavior of the channel is exploited such that in case of good channel states a significantly higher data rate is supported than in bad channel states. This behavior is typically achieved by rate adaptation via adaptive modulation and coding (AMC). In addition, reliable link layer protocols with persistent automatic repeat request (ARQ) are often used to guarantee error-free delivery. This concept is, for example, applied in EGPRS and further extended in high-speed downlink packet access (HSDPA). In

the following we will focus on EGPRS, since both appropriate descriptions and models are available. However, most concepts discussed and presented here are also applicable in other wireless systems with slight modifications and parameter adjustments.

In order to emulate time-varying EDGE-(enhanced data rates for GSM evolution) based radio channels in real time, a model has been developed and proposed in [17], which allows describing both short-term and long-term effects. This simulation model consists of three levels, which reflect typical physical layer and system properties [17].

- (i) The top level of the simulation model considers the overall cellular layout. Users are distinguished in two groups, one in good locations, and one with poorer receiving conditions.
- (ii) The second level characterizes system configurations, such as the applied power control, the velocity of the user, the interference conditions, and other system dynamics. This is reflected in the model by defining several states, which basically correspond to the coding schemes defined for EGPRS.
- (iii) Finally, the lowest level specifies the transmission conditions in a certain state. Throughout this work we assume a static resource allocation in terms of a constant number of assigned radio slots α . Independent of the current state, link layer packets are sent out periodically according to the fixed transmission time interval (TTI) τ_I . The payload size C_ξ of the packets differs for each state ξ , as different channel code rates and modulation schemes are applied to adapt to changing transmission conditions. Furthermore, since we assume operation in persistent acknowledged mode (i.e., lost link layer packets are retransmitted until they are received correctly), we extend the channel model to incorporate the transmission mode.

We summarize the description of the channel model including persistent acknowledged mode for a certain state ξ as $\mathcal{W}_\xi \triangleq \mathcal{W}(C_\xi, \tau_I, p_\xi, N_\tau)$, with p_ξ the loss probability, and N_τ the number of transmissions in state ξ . In case of multislot transmission and noise-limited scenarios, the payload is multiplied with α , such that $C_\xi \rightarrow \alpha C_\xi$. In interference-limited scenarios, the TTI can be divided by the number of slots, that is, $\tau_I \rightarrow \tau_I/\alpha$.

Figure 5 depicts the statistical EDGE radio link model specified by a two-group, five-state Markov chain according to [17]. The radio system is completely characterized by the payload size C_ξ for each state, the link layer packet error rate¹ $p_\xi = p$, the state transition probabilities λ , μ_1 , and μ_2 , and finally, the group probabilities $p_{G,1}$ and $p_{G,2}$. All of these parameters depend on the actual radio system configuration, such as frequency reuse pattern, power control option, number of users per sector, and so forth. An exemplary set of

¹ For the investigated EGPRS configuration the link layer packet error rate is independent of the state. In other words, the coding schemes and the power are adapted such that a constant error rate is maintained.

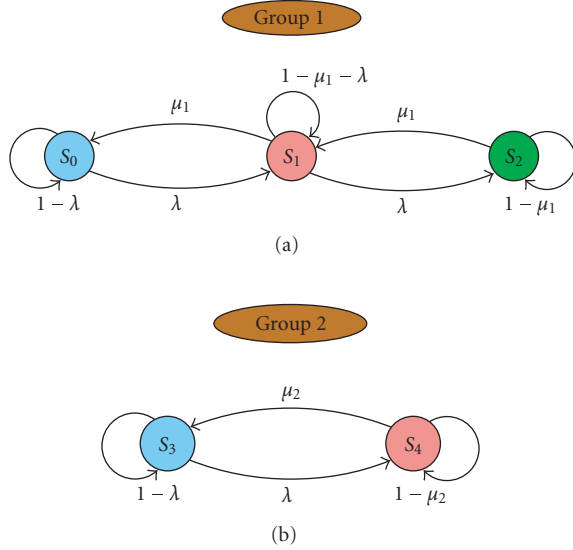


FIGURE 5: Two-group, five-state Markov channel model.

TABLE 1: Radio system parameters for EGPRS with frequency hopping, frequency reuse 1/3, and radio aware power control.

Users/sector	$p_{G,2}$	λ	μ_1	μ_2	p
2	0.93	0.3	0.055	0.05	0.11
8	0.64	0.3	0.094	0.3	0.20
15	0.28	0.3	0.27	0.59	0.27

parameters [17] for the EDGE radio system used in this work is presented in Table 1.

4.2. Abstract channel representation

An accurate model as presented in Section 4.1 is definitely helpful to obtain representative results. However, it is obvious that such a model is never comprehensive, nor can it be assumed that the parameters are known in advance. Nevertheless, it is always advantageous to include channel state information into decisions at the transmitter. Therefore, an abstraction of the previously introduced channel characteristics to some meaningful but also measurable and simple information at the sender unit is highly desired.

Sufficient information for our scheduling entity (specified in more detail in Section 5) is some a priori information on the probability that the channel supports a certain data rate over a certain time interval. More precisely, we ask how likely it is that a certain amount of data has left the sender buffer by some time measured as delta from the actual time τ_a . Note that in our case the sender and the receiver buffers are each other's complement and we assume the propagation delay to be negligible. Hence, without loss of generality, the time the data leaves the sender buffer is equivalent to the time it is available at the receiver. To formalize this notion, we define the event that the channel is able to support some rate r (in bits) within a time interval t as $\mathcal{R}(r, t)$. However, it is

not only sufficient to receive a certain rate by some time for the data to be useful at the receiver: due to the dependency graph it might be necessary that also some preceding data is sent out at some earlier time. Therefore, we generally require a joint probability distribution $\Pr\{\cup_i \mathcal{R}(r_i, t_i) \mid \xi\}$, which depends on the probability of the joint events, as well as on the current channel state ξ at time τ_a .

Whereas access to an estimate of the single event success probability $\Pr\{\mathcal{R}(r, t)\}$ is feasible, as will be shown later, estimation of the joint probability function is rather complex. However, if we only have access to the single event success probabilities, the joint event success probability can at least be bounded by the product of the single success probabilities and the minimum of the single success probabilities, that is,

$$\prod_i \Pr\{\mathcal{R}(r_i, t_i)\} \leq \Pr\left\{\bigcup_i \mathcal{R}(r_i, t_i)\right\} \leq \min_i \{\Pr\{\mathcal{R}(r_i, t_i)\}\}. \quad (4)$$

4.3. Simplified description for EGPRS

The exact derivation of the single event success probability distribution for complex channel models is still too complicated and likely without practical relevance, as discussed previously. Therefore, we attempt to obtain a simplified description for the single event success probability $\Pr\{\mathcal{R}(r, t)\}$ in case of an EGPRS channel. Despite being verified only for this specific system, it can be conjectured that the proposed model is relatively generic and can also be adapted for other wireless systems.

Recall that transmission within each single state is represented by $\mathcal{W}(C_\xi, \tau, p_\xi, N_\tau)$. Then, let X_ξ be a random variable which describes the amount of data transmitted with a single link layer packet in state ξ , with $X_\xi \in \{0; C_\xi\}$. Furthermore, let $1 - p$ be the probability of successful packet reception ($X_\xi = C_\xi$), and p the probability of a packet loss ($X_\xi = 0$). The mean and variance of this process are $m_\xi = C_\xi(1 - p_\xi)$ and $\sigma_\xi^2 = C_\xi^2(1 - p_\xi)p_\xi$, respectively.

As, in general, provision of feedback and retransmission at the link layer happen quite fast, the respective delay can be neglected. This is especially the case for scenarios where the channel propagation time of one packet is sufficiently smaller than the time interval between two consecutive higher-layer data units. Moreover, in delayed feedback systems packet labeling allows reordering of received packets. Therefore, we can assume that the lost packet will immediately be retransmitted at time instant $k + 1$. Then, for some channel state sequence $\xi_K = \{\xi_1, \dots, \xi_K\}$, the random sum rate $S(\xi_K)$ can be defined as

$$S(\xi_K) \triangleq \sum_{k=1}^K X_{\xi_k} = \sum_{\xi=1}^{N_\xi} \omega_\xi X_\xi, \quad (5)$$

with ω_ξ the frequency of state ξ in the sequence ξ_K . For sufficiently large K , it can be assumed that the sum rate $S(\xi_K)$ approaches a normal distribution due to the central limit theorem [18]. In addition, if the frequency ω_ξ for each state is also

sufficiently large, the distribution of the normalized sum rate can be characterized as a normal distribution,² that is,

$$\frac{S(\xi_K) - Km(\xi_K)}{\sigma(\xi_K)\sqrt{K}} \rightarrow \mathcal{N}(0, 1), \quad (6)$$

with normalized mean

$$m(\xi_K) = \frac{1}{K} \sum_{\xi=1}^{N_\xi} \omega_\xi m_\xi \quad (7)$$

and normalized variance

$$\sigma^2(\xi_K) = \frac{1}{K} \sum_{\xi=1}^{N_\xi} \omega_\xi \sigma_\xi^2 \quad (8)$$

due to the central limit theorem and some extensions [18].

However, in general the state sequence is also random and follows the underlying Markov model. Assuming that the actual state ξ is known, we are interested in the distribution of the sum rate $S_{K|\xi}$ after the transmission attempt of K link layer packets, that is,

$$S_{K|\xi} \triangleq \sum_{k=1}^K X_{\xi_k|\xi}. \quad (9)$$

For K sufficiently large, a normal distribution of the sum rate is still justified. However, the derivation of the mean and the variance is not straightforward. Therefore, it is recommended to estimate those parameters $m_{K|\xi}$ and $\sigma_{K|\xi}^2$ depending on the number of link layer packets K and the initial state ξ . If the channel state, however, is not accessible, we denote the mean as m_K and the variance as σ_K^2 . Figure 6 shows the normalized means $m_{K|\xi}/K$ and m_K/K , as well as the normalized variances $\sigma_{K|\xi}^2/K$ and σ_K^2/K for the EGPRS parameters given in Table 1. When comparing the different curves for the two parameters, it is obvious that additional simplifications and modeling might be performed. In a practical system, these parameters might be estimated in advance or are constantly updated during the transmission. In the following we will assume that the parameters $m_{K|\xi}$ and $\sigma_{K|\xi}^2$, or at least some estimates, are available to the transmitter.

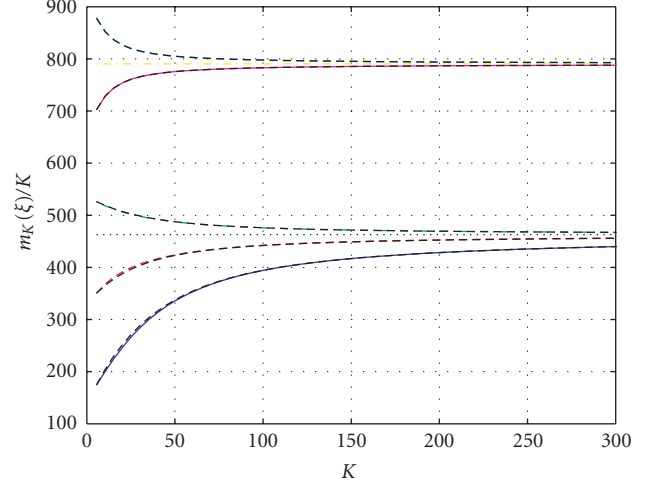
With knowledge of the mean and the variance for each K (and each initial state ξ), the probability of a certain sum rate is readily expressed as

$$\Pr\{S_K = s\} = \frac{1}{\sqrt{2\pi\sigma_K^2}} e^{-\frac{(s-m_K)^2}{2\sigma_K^2}}. \quad (10)$$

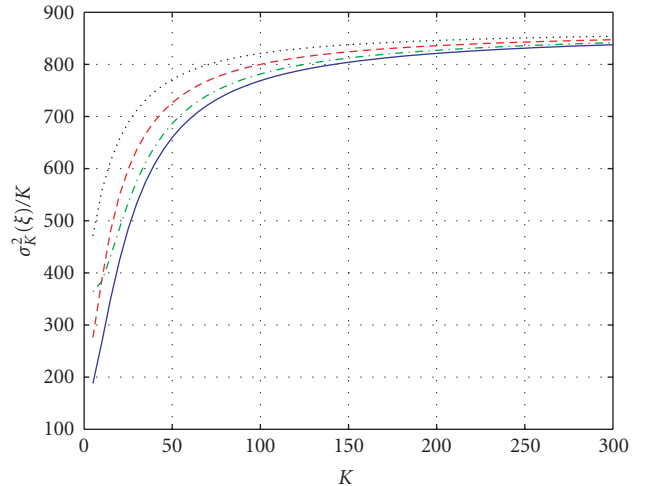
Hence, the single event success probability in case of knowledge of the channel state can be written as

$$\Pr\{\mathcal{R}(r, t)\} = \Pr\{S_{\lfloor t/\tau_3 \rfloor} \geq r\} = \frac{1}{2} \operatorname{erfc}\left(\frac{r - m_{\lfloor t/\tau_3 \rfloor}}{2\sigma_{\lfloor t/\tau_3 \rfloor}}\right). \quad (11)$$

For ease of exposition, we will in the following only present the case where the channel state is not known. The exten-



(a)



(b)

FIGURE 6: Normalized $m_{K|\xi}/K$ and m_K/K as well as normalized variances $\sigma_{K|\xi}^2/K$ and σ_K^2/K versus number of link layer packets K .

sion to the case when the channel state is known, however, is straightforward.

5. OPTIMIZED TRANSMISSION SCHEDULING AND BIT STREAM SWITCHING

5.1. Transmitter assumptions

We will consider a wireless video streaming system as introduced in Section 2, with a central scheduling unit in the transmitter. The latter should decide at each time instant

² Throughout this work, $\mathcal{N}(m, \sigma^2)$ will denote the normal distribution with mean m and variance σ^2 .

which data unit to transmit next out of the set of available ones $\{\mathcal{P}_{n,v}\}$, with $N = 1, \dots, N$ and $v = 1, \dots, V$, on the streaming server. To achieve good user experience, some obvious principles for the selection of data units are as follows.

- (1) The algorithm should be able to react to varying channel conditions by bit stream switching. Only if the channel conditions change too fast, additional reduction of the temporal resolution should be allowed.
- (2) Data units should be transmitted as close as possible to the time instant they are due at the receiver. Otherwise, bandwidth is wasted, which might result in expiration and consequently dropping of other earlier data units.
- (3) Nevertheless, it should be possible to transmit important data units earlier to guarantee their delivery even in bad channel conditions.
- (4) Version switching should preferably be accomplished with SP-frames rather than with SI-frames.

Previous work on this subject has for example been performed in [6], which is an extension to the well-known early deadline first (EDF) scheduling [5]. In [6] the EDF scheduling is extended taking into account frame dependencies. In this work we formalize the concept of frame dependencies and frame importance, extend it to stream switching, and introduce schedulers which try to optimize sending order. Before we present our proposed algorithm for optimized transmission scheduling and bit stream switching, we want to discuss some reasonable constraints. The latter will be helpful for significantly reducing the amount of possible data units to be considered in the optimization process.

- (i) Each data unit $\mathcal{P}_{n,v}$ is only transmitted once from end-to-end, since we assume that the lower link layer retransmission protocol clears out all errors. Hence, a loss in our system only happens due to late-arrival at the media client.
- (ii) If the transmission of data unit $\mathcal{P}_{n,v}$ in version v has been attempted, all data units at the same position n in the video sequence, which resemble different versions $v' \neq v$, are removed from the set of data units considered for future transmissions.
- (iii) It is also assumed that the information on the successful reception or loss of a single data unit is immediately available at the transmitter. As a consequence, a status³ γ_n can be assigned at the transmitter to each position n in the video sequence. If any data unit $\mathcal{P}_{n,v}$ ($v = 1, \dots, V$) has already been transmitted, the status takes on one of the following two (final) values:
 - $\gamma_n = \text{ACK}$, if the data unit is known to have been received correctly and
 - $\gamma_n = \text{NAK}$, if the data unit is known to be lost.
- (iv) Positions at which no data unit $\mathcal{P}_{n,v}$ (of any version) has been transmitted yet are assigned one of the remaining two (intermediate) status values:

- $\gamma_n = R$ (for READY), if transmission is possible in general, since all ancestors $\mathcal{P}_{n',v,n'}$ are available at the receiver (i.e., have $\gamma_{n'} = \text{ACK}$),
- $\gamma_n = P$ (for PENDING) if transmission is not recommended yet, since there are still some missing ancestors at the receiver (i.e., which have $\gamma_{n'} = P$).

- (v) As a consequence, only data units with status $\gamma_n = R$ are considered for transmission.
- (vi) Any data units $\mathcal{P}_{n,v}$ with expired deadline $T_n + \Delta > \tau_a$ (with Δ the initial play out delay and τ_a the actual time at the transmitter) are not transmitted and, together with all of their dependants, are assigned $\gamma_n = \text{NAK}$. Note that this procedure is already quite intelligent, as in this case the channel is not blocked with no more useful data.
- (vii) Switching positions in the video sequence are assigned two status values: one for SI-frames γ_n and one for SP-frames $\tilde{\gamma}_n$. For SP-frames to be decodable, it is assumed that it is necessary and sufficient that the previous P-frame of any version is available.

5.2. Periodic update of side information at the transmitter

Any optimal scheduling strategy requires up-to-date side information on the state of the system in the decision process. Therefore, we will explain the various update steps next that are performed before each scheduling process starts. Upon initialization, the first position $n = 1$ in the video sequence, as well as all other switching positions which have an SI-frame available, are assigned $\gamma_n = R$. All other positions are initialized with $\gamma_n = P$. After each successful or unsuccessful completion of the transmission of a data unit \mathcal{P}_{n,v_n} at actual time τ_a , the status values at other data unit positions in the transmitter are updated as follows.

- (1) All data unit positions n' for which the deadline has expired, that is, where $T_{n'} + \Delta > \tau_a$, are assigned $\gamma_{n'} = \text{NAK}$.
- (2) If the previous transmission of data unit \mathcal{P}_{n,v_n} was successful, the corresponding status value is changed to $\gamma_n = \text{ACK}$.
- (3) If the previous transmission, however, was not successful, the corresponding status value is changed to $\gamma_n = \text{NAK}$.
- (4) All data unit positions \tilde{n} for which at least one ancestor $\tilde{n} < n'$ has status $\gamma_{n'} = \text{NAK}$ are also assigned status $\gamma_{\tilde{n}} = \text{NAK}$.
- (5) All data unit positions with status $\gamma_{n'} = P$ for which all ancestors have $\gamma_n = \text{ACK}$ are switched to status $\gamma_{n'} = R$.
- (6) At switching positions for which all ancestors of the SP-frame are now available at the receiver, the status is changed to $\tilde{\gamma}_n = \text{ACK}$. In this case, either the SI-frame or the SP-frame (depending on the rate) for each version can be selected as a possible candidate for transmission.

³ Note that the status is only indexed with the position n , but not with the version v .

After this update procedure has been performed, a new data unit with $\gamma_n = R$ must be selected for transmission by the scheduler as described in the next section.

5.3. The scheduling process

The task of the scheduler is to determine an optimal transmission order and version of the sent data units, which maximizes the expected overall system performance. In particular, the scheduler decides at each transmission opportunity

- (i) which data unit to transmit next, possibly out of decoding order,
- (ii) and in case of an SP-, SI-, or I-picture, which version to transmit next.

During this decision process the scheduler should take into account as much side information as possible: the currently expected channel behavior, the actual time τ_a , as well as the deadlines, the importance, the different versions, and the updated status of different data units.

As already mentioned, the scheduler might decide to transmit more “important” data units earlier to guarantee their timely delivery with high probability, whereas other data units with very low importance might not be transmitted at all. Hence, we express the actual delivery order by the transmission schedule $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots)$, where π_k holds the index of the data unit (i.e., the position in the video sequence) to be transmitted at (temporal) position k . Furthermore, for each element in the transmission schedule, a version v_{π_k} is also selected.⁴

We propose to select the next data unit for transmission based on some utility function, for which we use the expected quality at the receiver $\mathcal{Q}(\boldsymbol{\pi}, \mathbf{v})$. This metric generally depends on all relevant source and channel information, such as rates, deadlines, importance vector, and so forth. More specifically, the optimal schedule $\boldsymbol{\pi}_{\text{opt}}$ and the optimal version vector \mathbf{v}_{opt} satisfy

$$(\boldsymbol{\pi}_{\text{opt}}, \mathbf{v}_{\text{opt}}) = \arg \max_{(\boldsymbol{\pi}, \mathbf{v})} \tilde{\mathcal{Q}}(\boldsymbol{\pi}, \mathbf{v}), \quad (12)$$

with

$$\begin{aligned} \tilde{\mathcal{Q}}(\boldsymbol{\pi}, \mathbf{v}) = & Q_0 + \sum_{\substack{n=1 \\ \gamma_n = \text{ACK}}}^N I_{n, v_n} + \sum_{\substack{n=1 \\ \gamma_n = \text{NAK}}}^N 0 \\ & + \sum_{\substack{n=1 \\ \gamma_n \in \{R, P\}}}^N I_{n, v_n} \tilde{P}_n(\boldsymbol{\pi}, \mathbf{v}) \prod_{\substack{m=1 \\ m < n}}^{n-1} \tilde{P}_m(\boldsymbol{\pi}, \mathbf{v}). \end{aligned} \quad (13)$$

Here, $\tilde{P}_n(\boldsymbol{\pi}, \mathbf{v})$ expresses the probability that data unit \mathcal{P}_{n, v_n} will be received in time for this selection of $\boldsymbol{\pi}$ and \mathbf{v} . Note that for data units with $\gamma_n = \text{ACK}$, this probability is equal to 1, while for $\gamma_n = \text{NAK}$ it is equal to 0. In case

$\gamma_n \in \{R, P\}$, the probability depends on the sum rate of the scheduled data units, the delivery deadline $T_n + \Delta$, the actual time τ_a , and the channel statistics $\mathcal{R}(r, t)$, and can be written as

$$\begin{aligned} \tilde{P}_n(\boldsymbol{\pi}, \mathbf{v}) = & \Pr \left\{ \mathcal{R} \left(\sum_{k=1}^{\rho_n} R_{\pi_k, v_{\pi_k}} T_n + \Delta - \tau_a \right) \mid \bigcup_{\substack{m < n \\ \gamma_m \in \{R, P\}}} \right. \\ & \left. \times \mathcal{R} \left(\sum_{k=1}^{\rho_m} R_{\pi_k, v_{\pi_k}} T_m + \Delta - \tau_a \right), \boldsymbol{\pi}, \mathbf{v} \right\}, \end{aligned} \quad (14)$$

where ρ_n defines the (temporal) position of data unit \mathcal{P}_{n, v_n} in the schedule $\boldsymbol{\pi}$.

Hence, when determining the expected quality according to (14), we acknowledge the fact that due to the dependencies in the video sequence not only the actual data unit must have been received, but also all of its predecessors. The above notation can be simplified by using the joint probability $P_n(\boldsymbol{\pi}, \mathbf{v})$ instead of the conditional probability $\tilde{P}_n(\boldsymbol{\pi}, \mathbf{v})$, that is,

$$P_n(\boldsymbol{\pi}, \mathbf{v}) = \Pr \left\{ \bigcup_{\substack{m < n \\ \gamma_m \in \{R, P\}}} \mathcal{R} \left(\sum_{k=1}^{\rho_m} R_{\pi_k, v_{\pi_k}} T_m + \Delta - \tau_a \right) \mid \boldsymbol{\pi}, \mathbf{v} \right\}. \quad (15)$$

The optimal transmission schedule and version vector now have to satisfy

$$(\boldsymbol{\pi}_{\text{opt}}, \mathbf{v}_{\text{opt}}) = \arg \max_{(\boldsymbol{\pi}, \mathbf{v})} \mathcal{Q}(\boldsymbol{\pi}, \mathbf{v}), \quad (16)$$

with

$$\mathcal{Q}(\boldsymbol{\pi}, \mathbf{v}) \triangleq \sum_{\substack{n=1 \\ \gamma_n \in \{R, P\}}}^N I_{n, v_n} P_n(\boldsymbol{\pi}, \mathbf{v}). \quad (17)$$

Note that in (17), we have already considered the fact that only data units with status $\gamma_n = R$ or $\gamma_n = P$ must be part of the schedule $\boldsymbol{\pi}$. The version vector \mathbf{v} remains identical to the previous definition.

With these preliminaries, the scheduler repeats the following operations, until there are no more ready or pending data units at the transmitter.

- (1) After successful or unsuccessful completion of the transmission of a data unit, the status of the data units in the transmission set is updated according to Section 5.2.
- (2) Then, by combining this updated status information with (possibly new) channel state information, the scheduler determines $(\boldsymbol{\pi}_{\text{opt}}, \mathbf{v}_{\text{opt}})$ according to (16).
- (3) Finally, transmission of data unit $\mathcal{P}_{\pi_{\text{opt},1}, v_{\text{opt},1}, \pi_{\text{opt},1}}$ is initiated.

5.4. Implementation aspects and complexity reduction

The number of possible combinations the scheduler has to compare in (16) is huge, since exchange of a single element

⁴ Note that the version vector $\mathbf{v} = \{v_n\}_{n=1}^N$ is ordered with respect to the position of the data units in the video sequence.

in π even at some later position generally influences the expected quality and thus the selection of the next data unit. To find the optimal schedule and version vector, in principle, a brute-force search is necessary. Since this is far from being practically feasible, complexity reduction is essential. In the following we will discuss some simplifications first before we present our optimized scheduling algorithm.

5.4.1. Weighted cumulative importance

A major problem in the computation of the expected quality for a certain combination $\{\pi, \mathbf{v}\}$ originates from the fact that all later parts in the transmission schedule can be reordered in many ways. However, only the (typically few) data units with status $\gamma_n = R$ are candidates for transmission, and we want to find the best transmission order and version for them at the actual scheduling opportunity. While the aforementioned later parts are not used, any modifications there usually affect the decision of the data unit to transmit next (i.e., the one we are actually interested in). Hence, we still have to consider the influence of the transmission of current data units on later frames: it is definitely not sufficient to replace the entire set of possible transmission schedules and version vectors in (16) by a set which only includes data units with $\gamma_n = R$.

We take these dependencies into account by introducing the weighted cumulative importance of a certain data unit \mathcal{P}_{n,v_n} . For a given transmission schedule π and version vector \mathbf{v} the weighted cumulative importance $\tilde{I}_n(\pi, \mathbf{v})$ is recursively defined as

$$\tilde{I}_n(\pi, \mathbf{v}) \triangleq I_{n,v_n} + \sum_{\{k|\gamma_k=P \wedge n < k\}} w_k \tilde{I}_k(\pi, \mathbf{v}) P_k(\pi, \mathbf{v}). \quad (18)$$

Here, $n < k$ means direct dependency (i.e., n is a direct ancestor of k), and w_n denotes the weight of a data unit at position n in the video sequence. In case of P-frames and B-frames we define for the sake of simplicity

$$w_n \triangleq \begin{cases} 1 & \text{if } n \text{ is a P-frame or SP-frame,} \\ \frac{1}{2} & \text{if } n \text{ is a B-frame.} \end{cases} \quad (19)$$

This weighted cumulative importance has to be recomputed at each scheduling opportunity for all data units in the transmission set with $\gamma_n \in \{R, P\}$.

Assume now that the version vector \mathbf{v} is fixed. Let $\mathcal{S} = \{n \mid \gamma_n = R\}$ be the set containing all data unit positions with status $\gamma_n = R$. Correspondingly, $\bar{\mathcal{S}} = \{n \mid \gamma_n = P\}$ contains all remaining data unit positions with $\gamma_n = P$, which are still waiting for transmission. In the following we will only investigate schedules, which only have elements from \mathcal{S} in the first positions, before any remaining ones from $\bar{\mathcal{S}}$ are added. Such a schedule is denoted as $\{\pi(\mathcal{S}), \pi(\bar{\mathcal{S}})\}$. Furthermore, let us define the sum rate of all data units in \mathcal{S} as

$$R_{\mathcal{S}} \triangleq \sum_{n \in \mathcal{S}} R_{n,v_n}. \quad (20)$$

Thus, the weighted cumulative importance according to (18) can be approximated by

$$\begin{aligned} \tilde{I}_n(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) &= I_{n,v_n} + \sum_{\substack{k \in \bar{\mathcal{S}} \\ n < k}} w_k I_{k,v_k} \\ &\times \Pr \left\{ \bigcup_{\substack{j \in \bar{\mathcal{S}} \\ j < k}} \mathcal{R} \left(R_{\mathcal{S}} + \sum_{i=1}^{\rho_j(\bar{\mathcal{S}})} R_{\pi_i}, T_j + \Delta - \tau_a \right) \right\}, \end{aligned} \quad (21)$$

with $\rho_j(\bar{\mathcal{S}})$ the (temporal) position of data unit \mathcal{P}_{j,v_j} in $\pi(\bar{\mathcal{S}})$.

To simplify the computation of the weighted cumulative importance, we apply the upper and lower bounds according to (4). Let us define

$$P_j(r, \pi(\mathcal{S})) \triangleq \Pr \left\{ \mathcal{R} \left(r + \sum_{i=1}^{\rho_j(\mathcal{S})} R_{\pi_i}, T_j + \Delta - \tau_a \right) \right\}. \quad (22)$$

Hence, a lower bound on the weighted cumulative importance in (18) is obtained as

$$L_n(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) = I_{n,v_n} + \sum_{\substack{k \in \bar{\mathcal{S}} \\ n < k}} w_k L_k(\mathcal{S}, \pi(\bar{\mathcal{S}})) P_k(R_{\mathcal{S}}, \pi(\bar{\mathcal{S}})), \quad (23)$$

which results in the following lower bound on the expected quality in (17):

$$\underline{\mathcal{Q}}(\{\pi(\mathcal{S}), \pi(\bar{\mathcal{S}})\}, \mathbf{v}) = \sum_{n \in \mathcal{S}} L_n(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) P_n(0, \pi(\mathcal{S})). \quad (24)$$

To obtain an upper bound on the expected quality in (17) let us recursively define a local success probability for all $k \in \bar{\mathcal{S}}$ as

$$\hat{P}_k \triangleq \min \{P_k(R_{\mathcal{S}}, \pi(\bar{\mathcal{S}})) \mid \forall_{m \in \bar{\mathcal{S}} \mid m < k} \hat{P}_m\}. \quad (25)$$

Hence, an upper bound on the cumulative weighted importance in (18) can be given for any $k \in \bar{\mathcal{S}}$

$$\bar{I}_k(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) = I_{k,v_k} \hat{P}_k + \sum_{\substack{m \in \bar{\mathcal{S}} \\ k < m}} \bar{I}_m(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}), \quad (26)$$

which results in the following upper bound on the expected quality in (17):

$$\begin{aligned} \bar{\mathcal{Q}}(\pi(\mathcal{S}), \pi(\bar{\mathcal{S}}), \mathbf{v}) &= \sum_{n \in \mathcal{S}} \left[I_{n,v_n} P_n(0, \pi(\mathcal{S})) + \sum_{\substack{k \in \bar{\mathcal{S}} \\ n < k}} \bar{I}_k(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) \right]. \end{aligned} \quad (27)$$

A third method to at least estimate the quality using the weighted cumulative importance would be ignoring the joint

channel events. The corresponding estimate of the weighted cumulative importance results for all $n \in \mathcal{S}$ in

$$\hat{I}_n(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}) \triangleq \sum_{\substack{k \in \bar{\mathcal{S}} \\ n < k}} I_{k,v_k} P_k(R_{\mathcal{S}}, \pi(\bar{\mathcal{S}})), \quad (28)$$

which yields the following estimate on the expected quality in (17):

$$\hat{Q}(\pi(\mathcal{S}), \pi(\bar{\mathcal{S}}), \mathbf{v}) = \sum_{n \in \mathcal{S}} I_{n,v_n} P_n(0, \pi(\mathcal{S})) + \hat{I}_n(\mathcal{S}, \pi(\bar{\mathcal{S}}), \mathbf{v}). \quad (29)$$

The advantage of using a fixed later schedule $\pi(\bar{\mathcal{S}})$ and the notion of cumulative weighted importance is that the latter can be precomputed before different transmission schedules $\pi(\mathcal{S})$ are tested. The specific implementation of this computation is not discussed in further detail here, but basically in most cases the computation starts in the leaves of the dependency graph and moves backwards to the data unit positions with status $\gamma_n = R$. Although none of the proposed modifications is optimal anymore due to the fixing of the later data unit positions in $\pi(\bar{\mathcal{S}})$, they significantly reduce the complexity, especially, if the set \mathcal{S} is kept small.

The number of selected data units for the set \mathcal{S} is referred to as *look-ahead units* and is denoted by $N_{\mathcal{P}}$ in the following. For the selection procedure, we have used two different modes: in the first mode, called *ready selection*, we choose the first $N_{\mathcal{P}}$ data unit positions with status $\gamma_n = R$. In the second mode, called *sequential selection*, we choose the next $N_{\mathcal{P}}$ data unit positions, regardless of whether their status is $\gamma_n = R$ or $\gamma_n = P$. In this case, the presented algorithm has to be modified slightly, as the status of data units changes due to scheduling. It is necessary to introduce an internal variable which keeps track of the status during the scheduling process. Nevertheless, for both selection modes, the fixed part of the schedule $\pi(\bar{\mathcal{S}})$ is sequentially filled with the remaining data units in the order of their index n .

These definitions and simplifications can be combined in order to develop a multistage scheduling algorithm for video streaming over VBR links. The respective flow diagram is depicted in Figure 7, and the various elements will be explained in the next sections.

5.4.2. Separation of transmission order and version selection

Based on one of the quality metrics in (24), (27), and (29) we select the best transmission order $\pi(\mathcal{S})$ for each scheduling opportunity. During this process we always operate within the same version, that is, if a switching point is included in $\pi(\mathcal{S})$, the same version as used for the previous group-of-pictures (GoP) is selected after the point. The case of version switching is discussed separately below.

Note that despite fixing the schedule $\pi(\bar{\mathcal{S}})$ and only using a single version vector, all data unit positions up to the end N would need to be considered for the computation of the weighted cumulative importance. However, for IDR-frame switching it is easily shown that it is sufficient to only con-

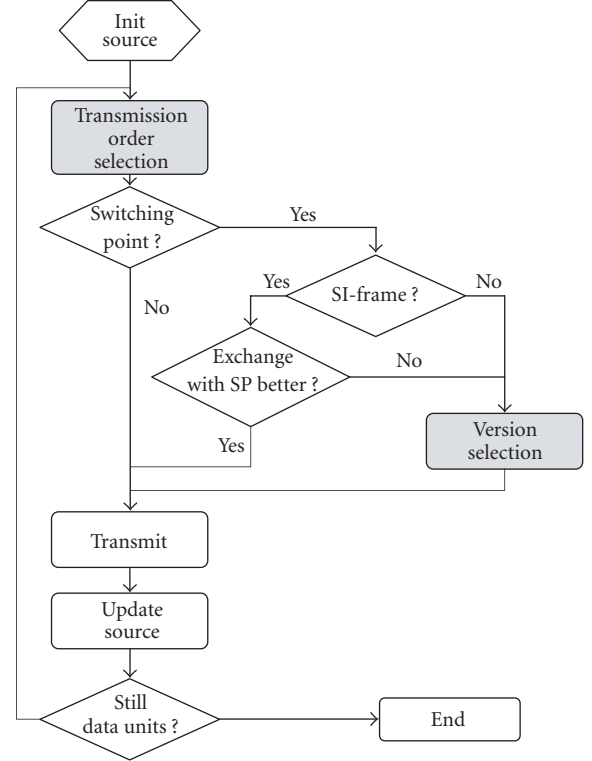


FIGURE 7: Flow diagram of the scheduling algorithm.

sider all data unit positions up to the next nontransmitted IDR-frame position. Though this is not completely accurate in case of SP/SI-frames for switching, we neglect this and use the same strategy for transmission order selection.

5.4.3. Data unit selection

In case a nonswitching position n is selected, the version v_n of the data unit is unambiguous, as exactly for one version the predecessors that are available at the decoder. The corresponding data unit \mathcal{P}_{n,v_n} is then transmitted. In case a switching position is selected, the corresponding data unit, however, is not immediately transmitted. If the switching position requires an SP-frame or an IDR-frame, the version selection procedure as presented below is immediately invoked. If the switching position requires an SI-frame, an alternative proposal is made: the selected SI-frame is replaced with the corresponding SP-frame. If this proposal yields a better metric than the SI-frame, the alternative is selected. If this is not the case, the version selection below is invoked.

5.4.4. Version selection

In addition to the local decisions on the transmission order, whenever a switching data unit position is due for transmission, the (possibly new) version is selected based on the following principles: only a single data unit position, namely the switching position, is included in \mathcal{S} . However, instead of

fixing the version vector \mathbf{v} , we evaluate the quality for all possible versions at this position. Our proposed algorithm allows taking into account not only the version selection of the next switching point, but a total of N_s switching points. Note that between two switching points the version is fixed for all data units to the version of the preceding switching point. Recursive computation of the quality in (24), (27), or (29) is then applied using the weighted cumulative importance. The number of considered switching points N_s is also referred to as *look-ahead switching points*.

Obviously, the more switching points N_s we take into consideration, the more complex the algorithm gets, but the performance also increases as more future data is taken into account for the decisions. In the next section, we will evaluate the performance of different system parameters and scheduling options for our proposed scenario and optimization strategy.

6. EXPERIMENTAL RESULTS

6.1. Simulation parameters

An exemplary set of simulations has been carried out using the following parameters: we have encoded $V = 4$ versions of a QCIF sequence of length $N = 2698$ with alternating speakers and sport scenes using H.264/AVC test model software JM8.2. The sequence has a length of about 90 seconds and contains sufficiently diverse content to yield representative results. We used a single QP for each version, namely $QP = 28, 32, 36, 40$, and a common frame rate of 30 fps, without any additional rate control algorithm. A group-of-picture (GoP) structure with IBBPBBP...SP has been applied with SP-picture distance of 1 second. The SP-pictures have "IDR" property in the sense that referencing over SP-pictures is not permitted. In addition, both SSP- and SI-pictures are possible. The initial play out delay at the streaming client is $\Delta = 1.5$ seconds.

The wireless link is modeled using the EGPRS channel model according to [17]. We restrict ourselves to the more challenging scenarios with 8 and 15 users per cell according to Table 1. Changes among the two groups of channel states may happen statistically independent every 20 seconds. Each simulation point represents the algebraic mean over 200 independent channel realizations.

As reference system, we also encoded the same video sequence with the rate control provided in JM8.2 to obtain two single-rate bit streams, one with 69 kbps and one with 96 kbps. The same GoP structure was used, but I-pictures have been applied instead of SP-pictures. We also investigate the case where we only apply SI-frames, such that the performance is actually similar to IDR-picture switching.

In the following we will compare the achievable performance for different scheduler settings (i.e., in terms of different number of look ahead units and look ahead GoPs), different selection modes (i.e., sequential and ready selection mode), and different methods to compute the weighted importance (i.e., lower bound, upper bound, and linear combination).

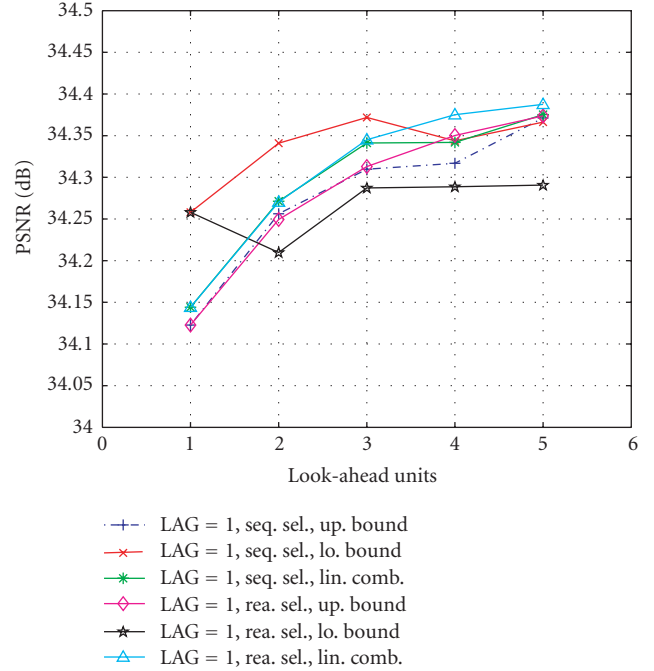


FIGURE 8: PSNR versus look-ahead units N_p for one look-ahead GoP and all selection modes and combining methods.

6.2. Local decisions: temporal scalability

First, we will investigate the performance of the scheduling algorithm when making local decisions on the actual transmission order, as well as on which frames are to be dropped (temporal scalability). Figure 8 shows the average PSNR versus number of look-ahead units N_p for one look-ahead GoP and all selection modes and combining methods. It can be observed in general that the achievable quality increases with larger scheduling sets. Note that the scheduling set size of one is identical to EDF [5, 19]. However, note that the PSNR variations are in a range of at most 0.3 dB and are therefore rather marginal.

The inconsistencies for the ready selection mode can be explained by the following additional observation of the authors: this method tends to use too often not the next ready data unit, but the second next one in the set, which has locally higher weighted cumulative importance. This locally optimum decision, however, leads to higher data unit drop rates in the long term.

Since larger values of N_p also increase the complexity of the algorithm, we can conclude from Figure 8 that $N_p = 3$ seems to be a good compromise for all scheduler options considered. We apply this setting in the remainder of this work unless mentioned otherwise.

6.3. Influence of selection mode and combining method on bit stream switching

Both the selection mode and the combining method used in the scheduler have an influence on the overall performance

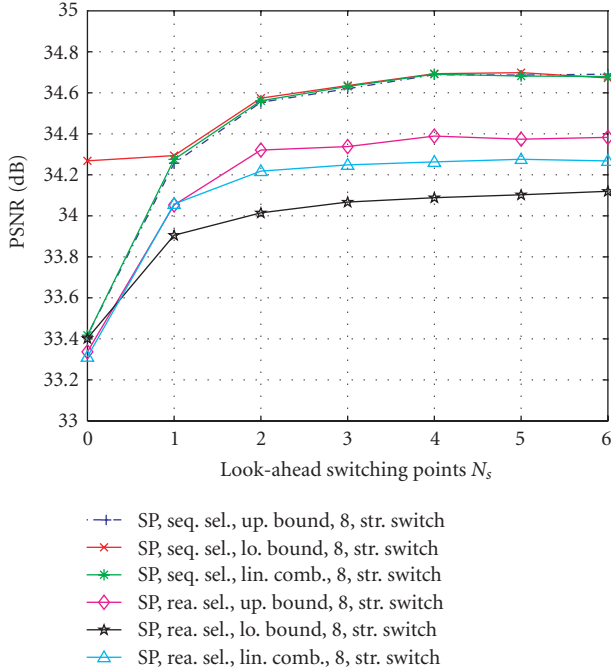


FIGURE 9: PSNR versus look-ahead switching points N_s and all selection modes and combining methods.

when stream switching is enabled. Figure 9 shows the PSNR versus number of look-ahead switching points N_s for different selection modes and combining methods and an EGPRS channel with 8 users. Obviously, if more switching points are taken into account, the performance increases for all strategies. Compared to not looking ahead (i.e., streaming the data as is and only allowing local decisions), a gain of up to 1.4 dB is now achievable.

While all strategies yield some gain, we have found out in a long series of tests that a scheduler using sequential selection with upper bound combining achieves the best and most consistent results.

6.4. System performance

Now we will compare the systems which use different streaming technologies, that is, constant bit rate (CBR) streaming (our reference system), streaming with optional bit stream switching using SI-pictures, and streaming with optional bit stream switching using SP-pictures. In all cases, we add smart dropping in the sense that the transmitter is aware of expired deadlines at the receiver and does not attempt to transmit this data. We believe that this setup is quite suitable to show the potential performance gains achievable with stream switching.

Figure 10 shows the PSNR versus look-ahead switching points N_s for different encoding strategies and an EGPRS channel with 8 users. As can be observed, SP-picture switching clearly outperforms SI-picture switching by about 0.8 dB. The gains are similar if IDR-pictures are used instead of SI-pictures (not shown here for the sake of conciseness). It is interesting to note that even in case of constant bit rate stream-

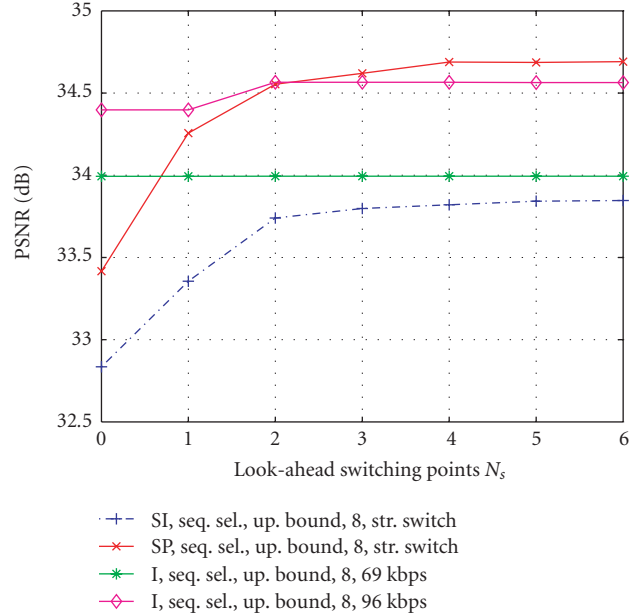


FIGURE 10: PSNR versus look-ahead switching points N_s for different encoding strategies and a channel with 8 users.

ing, the use of a scheduler at the transmitter provides some gains due to better local decisions, which result in temporal scalability. Furthermore, for the chosen scenario with only 8 users, bit stream switching only outperforms CBR streaming, if at the scheduler more than two look-ahead switching points are considered.

If we increase the system load at the transmitter by changing to a scenario with 15 users, the situation is different as shown in Figure 11: in this case the CBR stream with 96 kbps does not lead to an acceptable quality any more, and the 69 kbps stream seems more appropriate. The system which allows switching among 4 different streams, however, yields constant good quality and the overall degradation compared to the previous system with 8 users is only about 1 dB.

To better understand what happens exactly in the system, Figure 12 depicts the respective data unit drop rates. If we allow a reasonable amount of look-ahead switching points for transmission planning, the data unit drop rates significantly decrease in case of bit stream switching. As a consequence, the objective performance (in terms of PSNR) and also the subjective performance (in terms of viewer satisfaction) are largely enhanced.

7. CONCLUSIONS

In this work we have presented an optimized strategy for bit stream switching at the transmitter using the SP-frame concept in H.264/AVC. Provided that a sufficient amount of side information on the structure of the bit streams and the expected channel state is available at the scheduler, a significant performance gain over CBR streaming is achievable. In addition to this side information, it is also important to

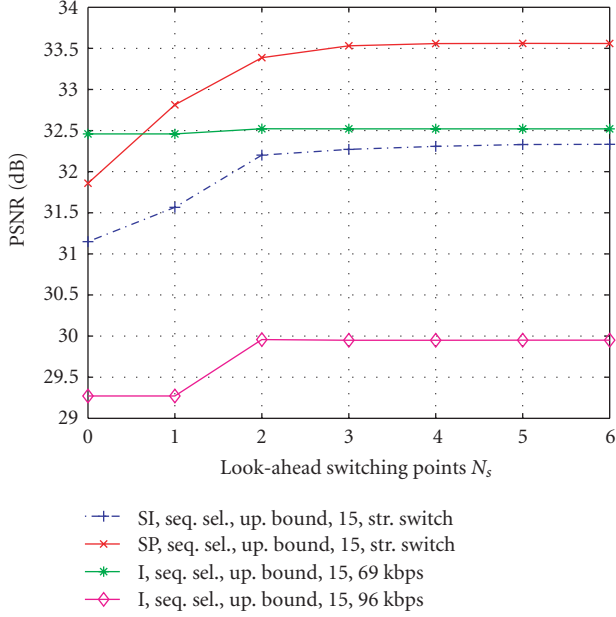


FIGURE 11: PSNR versus look-ahead switching points N_s for different encoding strategies and a channel with 15 users.

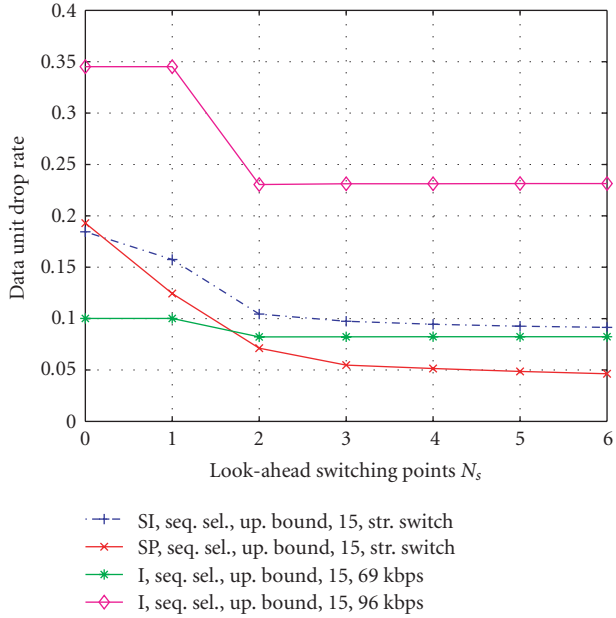


FIGURE 12: Data unit drop rate versus look-ahead switching points N_s for different encoding strategies and a channel with 15 users.

appropriately encode the various streams such that efficient switching is possible at all. Our proposed encoder solution for SSP- and SI-pictures meets this requirement.

Nevertheless, we note that complex optimization procedures inside the scheduler are required to fully exploit the potential of bit stream switching. By making use of some straightforward simplifications when considering the influ-

ence of the channel states in the scheduling decision, the complexity can be significantly lowered while still yielding good results.

Obviously, our proposed solution still leaves room for further optimization, which is the subject of ongoing work. For more details, we again refer to [16]. Future work items are, for example, to combine advanced bit stream switching with multiuser scheduling over wireless channels such that it can be included in the framework presented in [20].

APPENDICES

A. RECEIVED QUALITY AS SUM OF IMPORTANCES

Outline of proof. To prove (2) let us first define the following abbreviations:

$$\epsilon_n \triangleq c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m, \quad (\text{A.1})$$

$$\hat{Q}_i(n) \triangleq \epsilon_n Q(f_i, \hat{f}_n) + (1 - \epsilon_n) \hat{Q}_i(c(n)), \quad (\text{A.2})$$

$$\hat{Q}_i(0) \triangleq Q(f_i, \hat{f}_0), \quad (\text{A.3})$$

$$Q_i \triangleq Q(f_i, \hat{f}_i). \quad (\text{A.4})$$

In addition, we need the following relationship:

$$\forall_{i,n+i} (1 - \epsilon_i) \hat{Q}_i(n) = (\epsilon_n - \epsilon_i) Q(f_i, \hat{f}_n) + (1 - \epsilon_n) \hat{Q}_i(c(n)), \quad (\text{A.5})$$

which is easily shown by using all $\prod_{i,n+i} \epsilon_i \epsilon_n = \epsilon_i$ and by inserting in (A.2). Hence,

$$\begin{aligned} & \sum_{\substack{i=n+1 \\ n-i}}^N \hat{Q}_i(i) \\ &= \sum_{\substack{i=n+1 \\ n-i}}^N \epsilon_i Q(f_i, \hat{f}_i) + (1 - \epsilon_i) \hat{Q}_i(c(i)) \\ &= \sum_{\substack{i=n+1 \\ n-i}}^N \epsilon_i Q(f_i, \hat{f}_i) + (\epsilon_{c(i)} - \epsilon_i) Q(f_i, \hat{f}_{c(i)}) \\ & \quad + (1 - \epsilon_{c(i)}) \hat{Q}_i(c(c(i))) \\ &= \sum_{\substack{i=n+1 \\ n-i}}^N \epsilon_i \left[(Q_i - Q(f_i, \hat{f}_{c(i)})) + \sum_{\substack{k=i+1 \\ i-k}}^N (Q(f_k, \hat{f}_i) - Q(f_k, \hat{f}_{c(i)})) \right] \\ & \quad + \epsilon_n \sum_{\substack{k=n+1 \\ n-k}}^N (Q(f_k, \hat{f}_n) - \hat{Q}_k(c(n))) + \sum_{\substack{k=n+1 \\ n-k}}^N \hat{Q}_k(c(n)). \end{aligned} \quad (\text{A.6})$$

The received quality according to (2) is then obtained as

$$\begin{aligned}
Q(\mathbf{c}) &\stackrel{(a)}{=} \frac{1}{N} \sum_{n=1}^N \hat{Q}_n(n) \stackrel{(b)}{=} \frac{1}{N} \sum_{\substack{n=1 \\ c(n)=0}}^N \left[\hat{Q}_n(n) + \sum_{\substack{i=n+1 \\ n-i}}^N \hat{Q}_i(i) \right] \\
&\stackrel{(c)}{=} \frac{1}{N} \sum_{\substack{n=1 \\ c(n)=0}}^N \left[\epsilon_n Q_n + (1 - \epsilon_n) \hat{Q}_n(c(n)) + \sum_{\substack{i=n+1 \\ n-i}}^N \hat{Q}_i(i) \right] \\
&\stackrel{(d)}{=} \frac{1}{N} \sum_{\substack{n=1 \\ c(n)=0}}^N \left[\epsilon_n \left(Q_n - Q(f_n, \hat{f}_0) + \sum_{\substack{k=n+1 \\ n-k}}^N (Q(f_k, \hat{f}_n) - \hat{Q}_k(0)) \right) \right. \\
&\quad \left. + Q(f_n, \hat{f}_0) \right. \\
&\quad \left. + \sum_{\substack{i=n+1 \\ n-i}}^N \epsilon_i \left((Q_i - Q(f_i, \hat{f}_{c(i)})) \right. \right. \\
&\quad \quad \left. \left. + \sum_{\substack{k=i+1 \\ i-k}}^N (Q(f_k, \hat{f}_i) - Q(f_k, \hat{f}_{c(i)})) \right) \right. \\
&\quad \left. + \sum_{\substack{i=n+1 \\ n-i}}^N \hat{Q}_i(c(n)) \right] \\
&\stackrel{(e)}{=} \frac{1}{N} \sum_{n=1}^N Q(f_n, \hat{f}_0) \\
&\quad + \frac{1}{N} \sum_{n=1}^N \epsilon_n \left(Q_n - Q(f_n, \hat{f}_{c(n)}) \right. \\
&\quad \quad \left. + \sum_{\substack{k=n+1 \\ n-k}}^N (Q(f_k, \hat{f}_n) - Q(f_k, \hat{f}_{c(n)})) \right) \\
&\stackrel{(f)}{=} Q_0 + \sum_{n=1}^N \epsilon_n I_n,
\end{aligned} \tag{A.7}$$

whereby

(a) holds applying the decoding and concealment strategy as introduced above,

(b) holds with the same argument as mentioned previously that each frame is either concealed with grey ($c(n) = 0$), or its concealment depends on a frame which is concealed by grey,

(c) can be shown by inserting (A.5),

(d) can be shown by inserting (A.6),

(e) results from simple reordering and the fact that the sets for the sums are mutually exclusive,

(f) is obvious with the definitions in (1) and (2). \square

B. EXPECTED QUALITY AS WEIGHTED SUM OF IMPORTANCE

Outline of proof.

$$\begin{aligned}
E\{Q(\mathbf{C})\} &= \sum_{\mathbf{c} \in \{0,1\}^N} Q(\mathbf{c}) \Pr\{\mathbf{C} = \mathbf{c}\} \\
&\stackrel{(a)}{=} \sum_{\mathbf{c} \in \{0,1\}^N} \left(Q_0 + \sum_{n=1}^N I_n c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m \right) \Pr\{\mathbf{C} = \mathbf{c}\} \\
&\stackrel{(b)}{=} \sum_{\mathbf{c} \in \{0,1\}^N} Q_0 \Pr\{\mathbf{C} = \mathbf{c}\} \\
&\quad + \sum_{n=1}^N \sum_{\mathbf{c} \in \{0,1\}^N} \left(I_n c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m \right) \Pr\{\mathbf{C} = \mathbf{c}\} \\
&\stackrel{(c)}{=} Q_0 + \sum_{n=1}^N I_n \sum_{\mathbf{c} \in \{0,1\}^N} \left(c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m \right) \\
&\quad \cdot \prod_{i=1}^N \Pr\{C_i = c_i \mid C_1 = c_1, \dots, C_{i-1} = c_{i-1}\} \\
&\stackrel{(d)}{=} Q_0 + \sum_{n=1}^N I_n \Pr\{C_n = 1 \mid \forall_{k < n} C_k = 1\} \\
&\quad \times \prod_{\substack{m=1 \\ m < n}}^{n-1} \Pr\{C_m = 1 \mid \forall_{k < m} C_k = 1\}.
\end{aligned} \tag{B.1}$$

Here,

(a) results from inserting (2),

(b) holds as the sums are obviously exchangeable,

(c) assumes that the loss of data units only depends on past, but not on future channel states,⁵

(d) can be shown by using the following relationship:

$$\begin{aligned}
&\sum_{\mathbf{c} \in \{0,1\}^N} \left(c_n \prod_{\substack{m=1 \\ m < n}}^{n-1} c_m \right) \prod_{i=1}^N \Pr\{C_i = c_i \mid C_1 = c_1, \dots, C_{i-1} = c_{i-1}\} \\
&= \sum_{\{\mathbf{c} \in \{0,1\}^N \mid c_n = 1 \wedge \forall_{m < n} c_m = 1\}} \\
&\quad \times \prod_{i=1}^N \Pr\{C_i = c_i \mid C_1 = c_1, \dots, C_{i-1} = c_{i-1}\} \\
&= \Pr\{C_n = 1 \mid \forall_{k < n} C_k = 1\} \prod_{\substack{m=1 \\ m < n}}^{n-1} \Pr\{C_m = 1 \mid \forall_{k < m} C_k = 1\}
\end{aligned} \tag{B.2}$$

and by replacing the left-hand side of this relationship by the right-hand side in (d). \square

⁵ Obviously, we cannot assume that loss occurs statistically independent, as we might have strong correlations between data units or link layer units.

ACKNOWLEDGMENT

The authors would like to thank Hrvoje Jenkac and Daniel Pfeifer on initial discussion of this work.

REFERENCES

- [1] A. Ortega and K. Ramchandran, "Rate-distortion methods in image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.
- [2] 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.
- [3] T. Stockhammer, H. Jenkač, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 268–277, 2004.
- [4] M. Kalman, E. Steinbach, and B. Girod, "Adaptive media play-out for low-delay video streaming over error-prone channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 841–851, 2004.
- [5] T. Hasegawa, T. Kato, and K. Suzuki, "A video retrieval protocol with video data prefetch and packet retransmission considering play-out dead line," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP '96)*, pp. 32–39, Columbus, Ohio, USA, October–November 1996.
- [6] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Proceedings of the International Packet Video Workshop (PV '02)*, Pittsburgh, Pa, USA, April 2002.
- [7] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.
- [8] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 390–404, 2006.
- [9] "Scalable video coding - working draft 4," in *Doc. JVT-Q201, Joint Video Team (JVT)*, J. Reichel, H. Schwarz, and M. Wien, Eds., Nice, France, October 2005.
- [10] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, 1998.
- [11] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Isolated regions in video coding," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 259–267, 2004.
- [12] B. Xie and W. Zeng, "On the rate-distortion performance of dynamic bitstream switching mechanisms," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05)*, Amsterdam, The Netherlands, July 2005.
- [13] M. Karczewicz and R. Kurceren, "The SP-and SI-frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, 2003.
- [14] E. Setton and B. Girod, "Video streaming with SP and SI frames," in *Proceedings of the Visual Communications and Image Processing (VCIP '05)*, vol. 5960 of *Proceedings of SPIE*, pp. 2204–2211, Beijing, China, July 2005.
- [15] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, 1996.
- [16] M. Walter, "Advanced bitstream switching for wireless video streaming," Diploma thesis, Munich University of Technology (TUM), Munich, Germany, December 2004, http://www.Int.ei.tum.de/mitarbeiter/liebl/students/MichaelWalter_Diplomarbeit.pdf.
- [17] J. Cai, L. F. Chang, K. Chawla, and X. Qiu, "Providing differentiated services in EGPRS through packet scheduling," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '00)*, vol. 3, pp. 1515–1521, San Francisco, Calif, USA, November–December 2000.
- [18] P. Billingsley, *Probability and Measure*, John Wiley & Sons, New York, NY, USA, 1995.
- [19] T. Stockhammer, T. Wiegand, and D. Kontopodis, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," in *Proceedings of the International Packet Video Workshop (PV '02)*, Pittsburgh, Pa, USA, April 2002.
- [20] G. Liebl, H. Jenkac, T. Stockhammer, and C. Buchner, "Radio link buffer management and scheduling for video streaming over wireless shared channels," in *Proceedings of the International Packet Video Workshop (PV '04)*, Irvine, Calif, USA, December 2004.

Thomas Stockhammer has been working at the Munich University of Technology, Germany, and was a Visiting Researcher at Rensselaer Polytechnic Institute (RPI), Troy, NY, and at the University of San Diego, California (UCSD). He has published more than 70 conference and journal papers, is a Member of different program committees, and holds several patents. He regularly participates and contributes to different standardization activities, for example, JVT, IETF, 3GPP, and DVB and has coauthored more than 100 technical contributions. He is an acting Chairman of the video adhoc group of 3GPP SA4. He is also the cofounder and CEO of Novel Mobile Radio (NoMoR) Research, a company developing simulation and emulation of future mobile networks such as HSxPA, WiMaX, MBMS, and LTE. The company also provides consultancy. Between 2004 and June 2006, he was working as a Research and Development Consultant for Siemens Mobile Devices, now BenQ mobile in Munich, Germany. He is also consulting for Digital Fountain, Inc. His research interests include video transmission, cross-layer and system design, forward error correction, content delivery protocols, rate-distortion optimization, information theory, and mobile communications.



Günther Liebl holds the position of a Research and Teaching Assistant at the Institute for Communications Engineering (LNT), Munich University of Technology (TUM). As Dr.-Ing. candidate, his research interests are in the area of effective multimedia transmission over wireless links. This includes unequal error protection of scalable video, multimedia conference and streaming systems, congestion control for video services in cellular base stations, and multiuser scheduling in wireless environments. In 2004, he was a Visiting Scholar at the Information Systems Laboratory, Stanford University, where he worked on deadline-aware scheduling for wireless video streaming. He has published more than 25 conference and journal papers, and has coauthored more than 20 technical contributions to standardization bodies and patent applications. At LNT, he was responsible for the development of the WiNe2 real-time demonstration platform for multimedia services over cellular links. This platform is now



distributed via Nomor Research GmbH, an LNT spinoff. Apart from his position at the university, he is currently affiliated with this company as part-time consultant.

Michael Walter has graduated with the degree of a Diplom-Ingenieur in electrical engineering and information technology from the Munich University of Technology in 2004. His diploma thesis was on advanced bit stream switching for mobile streaming applications. Since early 2005 he is working for Heidenhain Corporation, Traunreut, Germany, as a development engineer.

