# Enhancement of Video Streaming QoS with Active Buffer Management in Wireless Environments

Zigmund Orlov and Marc C. Necker
Institute of Communication Networks and Computer Engineering, University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart
{zigmund.orlov | marc.necker}@ikr.uni-stuttgart.de

***Abstract*— With the emergence of high-speed wireless cellular networks, such as High Speed Downlink Packet Access (HSDPA) or WiMAX, new services with a high bandwidth demand have been introduced. One example is non-interactive video streaming. In contrast to wireline networks, the characteristics of wireless links, such as a time-varying bandwidth and the trade-off between delay and reliability, impose problems to a streamed video. In this paper, we study the impact of delay and losses within a Radio Access Network (RAN) on the video quality at the example of a state-of-the art HSDPA network. The nature of transmission errors is discussed. We investigate how active buffer management strategies within the RAN can alleviate the impact of an unreliable and time-varying link on the video quality, and we discuss the advantages and disadvantages of several proactive and reactive approaches. We further propose a proactive buffer management scheme with data differentiation which significantly increases the video quality by taking into account MPEG frame dependencies. Finally, we show that the structure of the MPEG-4 video data affect the performance of proactive buffer management schemes with data differentiation.**

**Keywords: video streaming, active queue management, QoS, UMTS, HSDPA**

## I. Introduction

Mobile cellular telecommunication networks have been growing continuously. New generations of mobile networks, such as the recent High Speed Downlink Packet Access (HSDPA) for UMTS offer more bandwidth resources in radio access networks in comparison to previous technologies. Because of this technical progress, new services will be offered in mobile networks, that have only been available in fixed networks until recently due to bandwidth limitations. One such service is the non-interactive video streaming service.

Generally, video streaming applications in a best-effort environment have to deal with unknown transmission channel characteristics with respect to the available bandwidth and the experienced delay and packet loss. This is of particular importance in a mobile wireless environment, where a highly time-varying and error-prone radio channel with a limited bandwidth may lead to unpredictable delay and bandwidth variations. Despite the efforts taken on the physical and the MAC-layer to alleviate the time-varying nature of wireless links, such as Automatic Repeat Request (ARQ) mechanisms and scheduling, the disturbing impact of the wireless transmission may eventually lead to a degradation of the perceived quality for a video service with real-time requirements. It is therefore desirable to take additional measures on the network, transport, or application layer.

In the area of video communication, two approaches for this have been established. The first approach adapts the streamed content to the current network conditions at the end terminals and is called end-to-end QoS control [1]. It mainly consists of congestion control, error control and error-resilient coding [2][3][4]. The second approach offers network support for video streaming and is named network-centric [2]. Two major representatives of network-centric approaches are queue management mechanisms and transcoders. While the task of queue management is to alleviate or prevent a congestion situation by dropping packets, transcoding nodes create a new version of the streamed content adapted to current network conditions.

In this paper, we focus on queue management mechanisms within a RAN. They can be grouped into reactive and proactive buffer management strategies [5], where the latter are also referred to as Active Queue Management (AQM). In [6], different simple reactive mechanisms are compared in a UMTS/HSDPA environment. In [7], the proactive mechanism FDDT (Frame-Level Packet Discard With Dynamic Thresholds) is proposed and evaluated in a fixed-network one-link-scenario. FDDT treats arriving packets differently, depending on the frame type they carry. In [8], a simple video frame discard scheduling algorithm with data differentiation in a WLAN-environment was presented.

In this work, we compare several reactive and proactive mechanisms in a state-of-the art wireless HSDPA-environment taking into account deadline constraints of the video service. We first discuss the reasons for video transmission errors and explain the dependencies of different error types. It will be demonstrated, that the objective video quality of a streamed non-interactive video-on-demand-content in a wireless environment can significantly be improved by means of application-aware buffer management, which takes into account video frame priorities and video frame dependencies. Further, it is shown that the deadline of the play-out buffer must be considered in video streaming investigations because it has a significant impact on the quality of the online playback of the streamed content. Finally, we discuss how the structure of the video content affects the performance of AQM-approaches with data differentiation.

This paper is organized as follows: In Section II, we briefly introduce the basic structure of an MPEG-4 video content, discuss the reasons for video data losses and describe the metrics used for the video quality evaluation. The nature of network losses and delay in a RAN is discussed in Section III. Section IV introduces the investigated buffer management schemes. The considered system and scenario is presented in Section V. The performance of the buffer management schemes is evaluated in Section VI, and section VII concludes the paper.

## II. Video Data

### A. Basic MPEG-4 Video Structure

An MPEG-4 video stream comprises three basic frame types, namely I-frames, P-frames and B-frames. I-frames (intra-coded frames) are encoded and decoded independently of any other frames. In contrast, P-frames (predictive coded frames) depend on the previous I- or P-frame. B-frames (bidirectionally coded frame) depend on the previous I- or P-frame as well as on the following I- or P-frame.

Predictively and bidirectionally coded frames (inter-coded frames) improve the compression efficiency. On the other hand, they decrease the error resilience since the loss of an intra-coded frame results in the loss of depending P- and B-frames. This leads to a trade-off, especially in an unreliable and time-variant mobile environment. In order to investigate this trade-off, we consider two basic coding schemes, referred to as 2B-coding and 8B-coding. With 2B-coding, every third frame is an I- or P-frame, where the intermediate two frames are B-frames (e.g., IBBIBBIBBPBBPBBI...). With 8B-coding, only every ninth frame is an I- or P-frame (e.g., IBBBBBBBBIBBBBBBBBPBBBBBBBBI...).

### B. Impairment of Video Quality during Video Streaming

Generally, three artifact sources exist in a digital video [17]:

- Distortions due to transformation process between analog and digital formats,
- Distortions caused by digital compression,
- Distortions due to error-prone transmission.

In this paper, we will focus on the third item and study the impact of the transmission process on the video content played back on the client side.

Figure 1 shows a simplified transmission path across a network. IP-packets arrive from the video server and are forwarded through the network. The network introduces delay due to buffering, processing delay and transmission delay. Additionally, packets may be lost due to buffer overflows or link-layer errors. On the client side, the packets are stored in the play-out buffer and played back at their play-back time.

Figure 2 illustrates the impact of data transmission on the video quality. In the left part of the figure, network components introduce network losses and delay. If the application has certain delay constraints, this network delay can lead to so-called deadline losses, where the video data arrives at the client after its anticipated play-back time. Network losses and deadline losses both directly lead to video data losses, which can cause further losses due to MPEG frame dependencies. This effect is known as MPEG loss propagation. Note that additional delays may be introduced above the network layer, for example

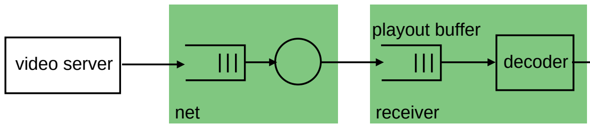


Fig. 1. Simplified transmission path

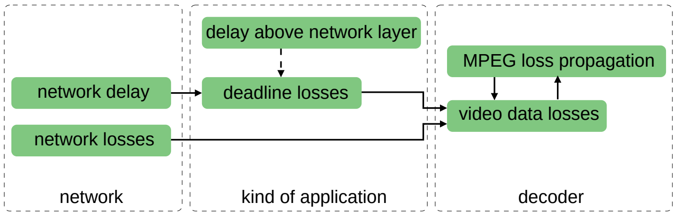


Fig. 2. Impact of network effects on video quality

during the encoding or decoding process in the application itself. These delays will be neglected in the following.

### C. Video Quality Evaluation

The quality of a video is highly subjective. This is in contrast to many other applications, where the performance can easily be captured with objective quality measures, such as for example the throughput for an FTP download. In order to evaluate the perceived quality of a video, large number of test persons are needed, who are asked to judge the video by using a given score range. One such score range is the well-known Mean Opinion Score (MOS), which is defined by ITU-T in [16]. The result of a test viewing is a *subjective* video quality statement. However, this procedure is very costly and also not very flexible. It requires a lot of different and also continuously changing test persons, which makes the evaluation of different algorithms with different parameter sets very expensive.

As an alternative, it is desirable to have an *objective* video quality metric, which can be obtained directly from the received video data without the need for test persons. Ideally, this objective metric can then be correlated with a subjective quality metric. A common objective metric is the Peak Signal to Noise Ratio (PSNR), which is also standardized by ITU-T in [17]. However, the unweighted PSNR is only loosely correlated to the human visual system [17]. Nevertheless, there are mapping tables of PSNR to MOS and the PSNR is widely used to determine the quality loss due to the MPEG encoding process. The distortion pattern of the encoding process is different from the distortions which occur due to packet losses in a network. Thus, the mapping tables determined for the encoding are not very suitable for other error sources.

In our investigations, the error concealment technique used at the client is the replacement of the lost frames by the last correctly received frame (frame-freezing). It is obvious that this technique works better with slow video scenes than with video scenes with fast motion. As the PSNR metric compares the decoded video to the originally transmitted video (i.e., it is reference-based), the video content impacts the evaluation results if the PSNR is used as a metric.

In order to evaluate the performance of the system independently of the streamed video content, we use the Frame Error Rate (FER) [7], which is reference-free. The FER describes the

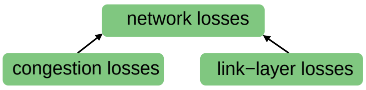


Fig. 3. Reasons for network losses in a RAN

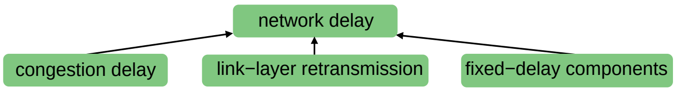


Fig. 4. Reasons for packet delay in a RAN

fraction of frames in error. If one IP-packet in a frame is lost, this frame and all other frames depending on this frame are considered to be frames in error. High or low FER values then stand for a bad or good perceived video quality, respectively.

## III. Network Losses and Delay in a RAN

The reasons for network losses and delay in a RAN differ from those in a fixed network. In addition to congestion losses, packets may be lost due to link-layer failures (Fig. 3). Congestion on a radio link is likely to occur, if the available bandwidth of the link decreases, for example when the terminal moves from the cell center to the cell border. Additionally, sharing of the wireless channel resources in a UMTS/HSDPA network also contributes to the congestion effect. The link-layer losses in a UMTS/HSDPA network are neglectable because of the powerful Hybrid ARQ (HARQ) mechanism which is active even in the case of a connection in unacknowledged mode.

Generally, the network delay (Fig. 4) comprises three components. The congestion delay is the main factor influencing the quality of non-interactive video streaming. Link-layer retransmissions eliminating the physical-layer errors on the air additionally contribute to the overall delay. Finally, the impact of fixed-delay components is not important for the consideration of the delay jitter in the context of non-interactive video streaming.

## IV. Buffer Management Schemes

### A. Classification

The transmission of video data over IP-based networks requires the segmentation of video data into IP-packets. The maximum size of an IP-packet is a constant parameter of the networks along the routing path. Generally, it is smaller than the average size of a video frame. Therefore, a video frame is segmented into several IP-packets during the transmission process. Consequently, we can distinguish between two buffer management strategies. The first one is a packet-based strategie, where each incoming IP-packet is treated individually. The second one is a frame-based approach, where a video frame is an atomic (undividable) unit during the buffer management decisions and actions.

Further, the buffer management schemes can be with or without data differentiation. With data differentiation, the buffer management decisions additionally depend on the received data (i.e. frame) type and its priority.

Another dimension in the classification room of the buffer management schemes is the distinction between reactive and proactive approaches. The reactive approaches become active when a congestion situation has already occurred, while the proactive approaches attempt avoiding the congestion in the first place.

### B. Buffer Management Without Data Differentiation

Most of todays network elements apply a simple drop-tail FIFO buffer management strategy, where newly arriving data are dropped if the queue is full. This reactive approach usually drops individual IP-packets. In the case of video streaming, this strategy can be extended to a frame-based drop-tail buffer, which drops all IP-packets belonging to the same video frame if one of its IP-packets was dropped.

Another reactive approach is the drop-head strategy, where those data units are dropped that reside longest in the queue. With this approach, the transmission of video data, that may arrive too late at the client is suppressed in favor of newly arriving data. Generally, the size of an arriving IP-packet may be different from the size of the packet waiting at the head of the queue. Our drop-head implementation removes as many packets from the head of the queue so that the dropped data amount is at least equal to the size of the newly arrived IP-packet.

In order to avoid the transmission of obsolete data, the data in the queue can be watched by a discard timer. This timer will remove all packets from the queue which have been waiting for a certain time period $T_D$. In our paper, we investigate this proactive approach in combination with a packet-based dropping strategy.

### C. Buffer Management With Data Differentiation

As we have seen, not all video data has the same importance. If an I- or a P-frame is lost, other video frames (P or B) depending on the lost frame can hardly be decoded properly. We investigate a proactive approach which drops newly arriving data units belonging to a B-frame (B-packets), for the benefit of data belonging to I- or P-frames (I- or P-packets), if a congestion situation is imminent. This will be referred to as *proactive B-dropping*. In order to determine whether a congestion situation is about to occur, a threshold $\delta$ is introduced. If the buffer occupancy exceeds $\delta$, the buffer will proactively drop arriving packets containing data of B-frames. This can be done either on a packet-basis or on a frame-basis.

*1) Packet-based proactive B-dropping:* The packet-based scheme manages the queue according to the already described drop-tail or drop-head strategies. In case of a drop-tail queue, a newly arriving IP-packet belonging to a B-frame (B-packet) is dropped, if the buffer occupancy exceeds the threshold value $\delta$. In case of a drop-head queue, the B-packet is stored at the tail of the queue after one or several B-packets are removed at the head of the queue if the threshold $\delta$ is exceeded.

*2) Frame-based proactive B-dropping:* Frame-based proactive B-dropping extends the just described packet-based scheme to a frame-based scheme (cmp. section IV-A). In the case of a drop-tail queue, a newly arriving IP-packet is dropped, if previous IP-packets belonging to the same frame were dropped. Also, if a newly arriving IP-packet is dropped,

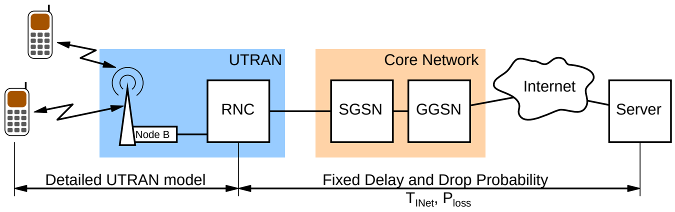


Fig. 5. Architecture of the considered 3G network

all IP-packets that belong to the same frame and which are already stored in the queue, are removed. This scheme works very similar to FDDT [7]. However, unlike FDDT, we remove already stored packets from the queue and do not perform preventive dropping of P-frames.

As a final extension of this scheme, we will consider the dependencies in-between different video frames. As the loss of a frame will strongly complicate or even prevent the decoding of the frames which are interdependent with the lost frames, we propose an approach that removes all involved interdependent frames in the queue. We will refer to this strategy as frame-based *with inter-frame dependencies*.

## V. System Model

### A. System overview

The basic scenario is shown in Fig. 5. We consider a single-cell environment, where several User Equipments (UEs) connect to the Node B via a High Speed Downlink Shared Channel (HS-DSCH) in the downlink and a dedicated channel (DCH) in the uplink direction. The Node B is connected to the RNC, which itself is connected to the Internet via the 3G-SGSN and 3G-GGSN of the cellular system's core network. The UEs establish a video streaming connection with a host in the Internet. The Internet and core network were assumed to introduce a constant delay $T_{\mathrm{INet}} = 20\,\mathrm{ms}$ in each direction and not lose any IP-packets.

The simplified queuing model in the downlink direction is shown in Fig. 6. The main buffer for each connection is the RNC input buffer located in the RNC. The Node B-buffer holds only a small portion of the data which is about to be transmitted on the air interface. Consequently, all buffer management strategies are assumed to be implemented in the RNC.

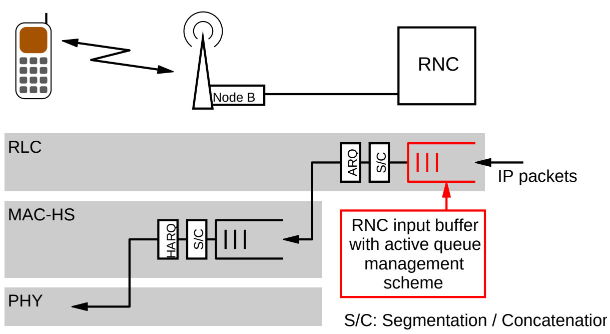


Fig. 6. Downlink queuing model of the HSDPA system

### B. Emulation Model

For our performance evaluation, we used a detailed model of a single-cell HSDPA-system. The HSDPA network was modeled with all its relevant RLC, MAC-d and MAC-hs protocols. The physical layer was modeled based on BLER-curves obtained from physical layer simulations including HARQ. Transport formats (TF) on the MAC-hs layer were selected based on the channel quality such that the BLER is 10%. We assumed ideal conditions for the reporting of Channel Quality Indicators (CQI) from the mobile terminals to the Node B, i.e. zero delay, in order to avoid side effects. The Iub flow control was operating with a short update period and no deadtime, since higher values can introduce unpredictable delays and delay spikes [20], which would make it difficult to isolate the performance influence of the investigated buffer management schemes. The maximum number of MAC-hs retransmissions was limited to $R_{\mathrm{max,hs}} = 4$, and the RLC layer was operated in unacknowledged mode. We neglect the convergence layer, as it only introduces a very small overhead in a single-cell environment. For a more detailed description of the model, please refer to [21].

The model was implemented with the event-driven simulation library IKR SimLib [18]. This model was then extended as an emulation based on the IKR EmuLib emulation environment [19].

### C. Video Server and Video Client

For these investigations, a software was developed that streams an MPEG-4 video content over UDP through the UMTS/HSDPA-emulator. Each IP-packet contains the necessary information as an IP-option in order to make content-dependent buffer decisions in all here discussed buffer management mechanisms. On the client side, IP-packet capturing software received the packets. The evaluation of the video quality is carried out frame-based: if any packet belonging to a frame is lost, the whole frame is considered as lost. Further, all frames depending on a lost frames, are also considered as lost. This approach is widely used [7][9][10][11]. The computation of the video data losses due too late frame arrivals at the client is performed with a cumulative jitter concept [12][13].

### D. Simulation Scenario

The scenario which we used for our performance evaluation comprised one real UDP video source and 4 additionally simulated cross-traffic CBR video streams with the same data rate. The real video source streams a single-layer MPEG-4 video encoded with the free available ffmpeg software [14]. The video material is a multiple rerun of a short CIF-sequence of an American football [15] match with fast scenes. The transmitted video consists of 33120 video frames and was encoded for an average data bit rate of $308\,\mathrm{kbps}$.

Terminal mobility was modeled taking into account both slow and fast fading. All mobile terminals move at a speed of $v = 30\,\mathrm{km/h}$. They periodically experience the same slow fading profile, where each mobile starts at a different position

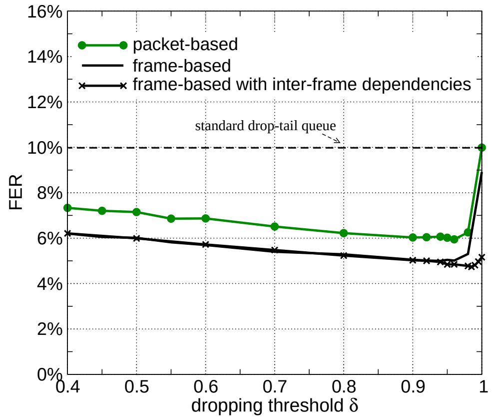


Fig. 7. Impact of proactive B-dropping in the case of an infinite play-out buffer

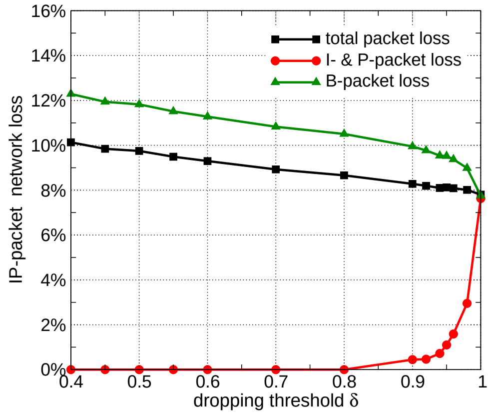


Fig. 8. Impact of proactive B-dropping on IP-packet network loss

of the profile in order to obtain independent channel conditions in-between the mobiles. A Proportional Fair (PF) [22] scheduler was used at the MAC-hs layer to assign resources to the different data streams.

## VI. Performance Evaluation

In this section, the comparison of the different buffer management schemes is presented and discussed. Furthermore, the impact of the coding scheme on the performance of the proactive approaches is shown and analyzed.

### *A. Comparison of different Buffer Management Schemes*

As we have seen in Section II-B, network losses and delay lead to video data losses. While timer mechanisms in the input queue and drop-head-approaches are designed to avoid deadline losses in the video client, the approaches with proactive B-dropping try to reduce the impact of network losses on the actual video data losses in the first place. However, these approaches also prevent deadline losses indirectly because of their data dropping strategy.

First, the case of an infinite play-out buffer at the video client is considered in order to evaluate the approaches with proactive B-dropping. This can be considered as the best case, since frames may be delivered arbitrarily late. In a further step, a more realistic case with a finite play-out buffer at the client is investigated.

For all investigations, the RNC-buffer has a finite size of $212\,$kByte. Thus, the data amount storeable in the queue corresponds to $6\,$s of the streaming video at $308\,$kbps.

*1) Infinite play-out buffer:* Fig. 7 compares packet-based and frame-based schemes with proactive B-dropping. All implementations are drop-tail based. Shown is the FER depending on the dropping threshold $\delta$. Note that a threshold of $\delta = 1$ corresponds to a classic FIFO queue without any proactive measures. We will use this simple packet-based drop-tail FIFO queue as a reference.

It can be seen, that the proactive B-dropping decreases the video data losses for threshold values $\delta$ smaller than $1$. We also observe, that frame-based approaches are better than packet-based approaches, since they avoid the transmission of data which is not usable at the receiver anyway. The results can even slightly be improved when considering inter-frame dependencies. This approach achieves the best FER value which is 50% smaller than the FER of the reference FIFO queue. According to [23], video data loss should be no more than 5%, which can be reached by both frame-based approaches in this scenario.

Both frame-based schemes have a very similar performance. On the one hand, the consideration of inter-frame dependencies makes the implementation in a real network node much more complex. On the other hand, such a scheme achieves a very good performance for $\delta = 1$, while it is not easy to find an optimal value. Note that the dropping threshold $\delta$ which achieves the best result depends on many factors, such as time-varying channel characteristics and the varying number of users over the shared wireless channel. Also, the sensitivity of the threshold value depends on the scenario and the buffer dimensioning. In our scenario, the threshold value with the best result is in the area of $\delta = 0.94 - 0.985$, compared to $0.9$ for FDDT in [7].

In Fig. 8, the operation of the proactive B-dropping is detailed at the example of a frame-based scheme. For $\delta = 1$, all packets are dropped with equal probability. As $\delta$ is decreased, more B-packets are dropped for the benefit of I- and P-packets. Although the total IP-packet loss increases, the I- and P-packet loss decreases. This results in the total elimination of the MPEG loss propagation effect.

This is detailed in Fig. 9, where the total FER, the direct FER (FER caused only by network losses), and the video data losses due to MPEG loss propagation are plotted over the dropping threshold $\delta$. There are not deadline losses because of the infinite play-out buffer. The total FER is the sum of the direct FER and MPEG-loss-propagation losses. Although the direct FER increases because of the proactive B-dropping, MPEG loss propagation effect disappears, which leads to a decrease of the total FER.

*2) Finite play-out buffer:* According to [4], a video streaming client can have a play-out buffer on the order of $5 - 15\,$s. For our investigations, we chose a play-out buffer of $6\,$s. A

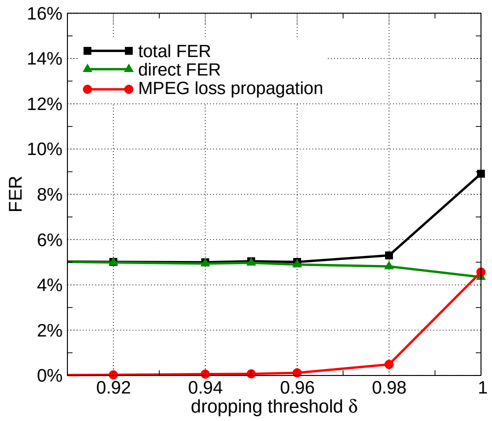


Fig. 9. Impact of proactive B-dropping in the case of a infinite play-out buffer

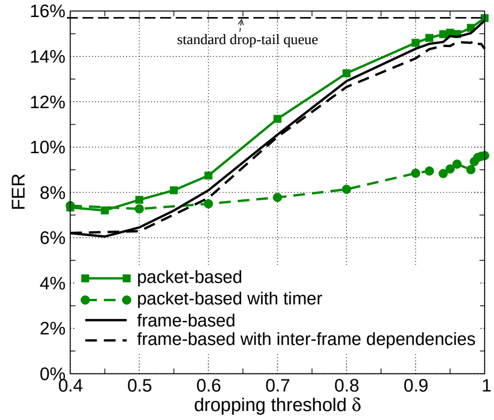


Fig. 10. Comparison of proactive B-dropping in the case of a finite play-out buffer (6 s)

finite play-out buffer makes it reasonable to discard packets in the RNC input buffer using a discard timer as described in section IV-B. Because of other delays within the RAN, such as retransmissions, the discard timer $T_D$ has to be chosen smaller than the play-out buffer. Here, we set $T_D$ to $5.35\,\text{s}$ if the discard timer is active.

Figure 10 compares packet-based and frame-based schemes depending on the dropping threshold $\delta$. All implementations are drop-tail based. Compared to the case of the infinite play-out buffer, the FER increases significantly, since deadline losses now account for a large portion of the video data losses in the considered scenario. The smaller the play-out buffer, the larger is the impact of the deadline losses.

Both frame-based schemes show a very similar performance, though the consideration of inter-frame dependencies gives a slight advantage in certain ranges of $\delta$. However, this comes at the cost of a much higher complexity. The minimum FER can be obtained for $\delta \approx 0.45$, where both schemes deliver about the same performance with an FER of 61% below the reference drop-tail queue.

In contrast, the proactive packet-based scheme shows a worse performance. However, as we add a timer mechanism to drop obsolete packets from the RNC input queue, the performance greatly increases. In particular, the performance of the packet-based scheme in combination with the discard timer is weakly dependent on the threshold $\delta$. This is also true for $\delta = 1$, where the proactive B-dropping scheme is disabled. This allows for a very easy implementation without any data differentiation and avoids the choice of $\delta$, even though the FER can be reduced by an additional 37% if a frame-based scheme with optimal choice of $\delta$ is used.

Finally, Fig. 11 compares the packet-based drop-tail with the packet-based drop-head scheme. Both curves show the same principal behavior, while the FER decreases much quicker for the drop-head scheme as $\delta$ is decreased. This makes the choice of an optimal threshold $\delta$ easier, since it offers a wider range with a near minimal FER. Note also that for $\delta = 1$ the FER of the drop-head scheme is below the FER of the drop-tail scheme because newly arriving packets benefit from the removal of the packets residing longest in the queue.

## B. Comparison of different Coding Schemes

As discussed in section II-A, the usage of inter-coded frames increases the coding efficiency and on the other hand decreases the error resilience. In the following, we investigate two coding schemes in order to evaluate their impact on the performance of the proactive B-dropping. We encoded the same raw video into two MPEG-4 videos, one with 2B-coding, the other one with 8B-coding (cmp. II-A). Both videos have the same average bit rate ($308\,\text{kbps}$).

The results for the frame-based proactive B-dropping are displayed in Fig. 12 (infinite size of the play-out buffer). First, in case of the disabled proactive B-dropping ($\delta = 1$), we can see the difference between two videos in terms of the error resilience explained in II-A. The difference is not large because we have to deal with the error sequences due to congestion which are relatively long in comparison to the number of B-frames between two reference frames. On the other hand, it can be observed, that the proactive B-dropping has a better performance with the 8B-coding video. This effect can be explained by Fig. 13, where the fraction of the video data in the different frame types is depicted for the both coding

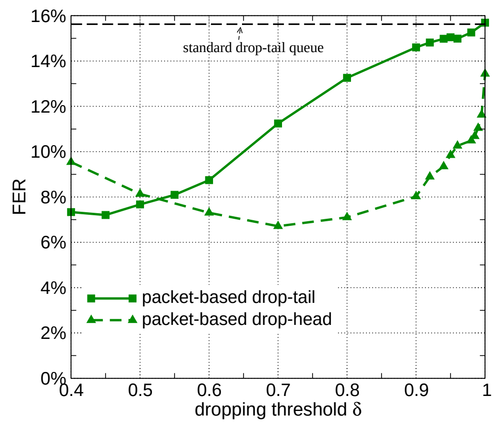


Fig. 11. Comparison of drop-tail and drop-head schemes

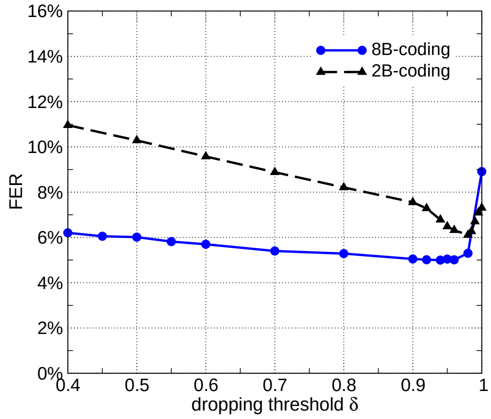


Fig. 12. Comparison of two coding schemes in the case of proactive B-dropping for the infinite play-out buffer case

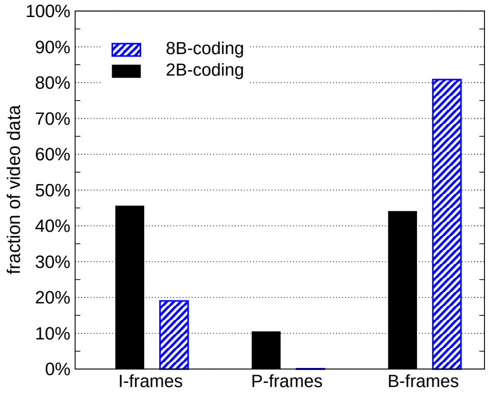


Fig. 13. Comparison of two coding schemes: fraction of video data carried by different frame types

schemes. In the case of the 2B-coding scheme, both I-frames and B-frames carry approximately equal data amount (45%). In the 8B-coding video, I-frames only carry approx. 20% of data and there are hardly any P-frames. Thus, the rest of the data (80%) are carried by B-frames. Therefore, the proactive B-dropping approach can drop more data (in the 8B-coding case) in order to avoid I-frame losses. However, a disadvantage of the 8B-coding scheme is a longer encoding duration because of the frame dependencies, which can have a negative impact on interactive applications.

## VII. Conclusion

In this work, different reactive and proactive queue management schemes were discussed and investigated with respect to enhancing the objective video quality of a streaming application in an HSDPA network. We showed that a simple timer-based queue management strategy significantly improves the objective video quality compared to a regular drop-tail queue. The video quality can further be improved by using proactive approaches with data differentiation, such as the discussed proactive B-dropping. The best results were achieved by the proposed frame-based scheme which considers MPEG-4 frame dependencies. In order to achieve good results with this proactive B-dropping scheme, the encoded video material should carry enough data in the B-frames. Consequently, a coding scheme with a larger number of B-frames achieves a better objective video quality, since it gives the proactive buffer management scheme more flexibility.

## VIII. Acknowledgments

This research was done in cooperation with Alcatel Research and Innovation Department, Stuttgart.

## References

[1] X. Wang, H. Schulzrinne, *Comparison of adaptive Internet multimedia applications*, IEICE Trans. Commun., vol. E82-B, no. 6, pp. 806-818, June 1999.
[2] D. Wu, Y. T. Hou, Y. Zhang, *Transporting Real-Time Video over The Internet: Challenges and Approaches*, Proceedings of the IEEE, vol. 88, No. 12, December 2000.
[3] D. Wu, Y. T. Hou, W. Zhu, Y. Zhang, J. M. Peha, *Streaming Video over the Internet: Approaches and Directions*, IEEE Transactions on circuits and systems for video technology, vol. 11, no. 3, March 2001.
[4] J. G. Apostolopoulos, W. Tan, S. J. Wee, *Video Streaming: Concepts, Algorithms and Systems*, HP Laboratories Palo Alto, September 2002.
[5] S. Bodamer, *Verfahren zur relativen Diensteguetedifferenzierung in IP-Netzknoten*, Dissertation, Institute for Communication Networks and Computer Engineering, University of Stuttgart, 2004.
[6] G. Liebl, H. Jenkac, T. Stockhammer, C. Buchner, A. Klein, *Radio Link Buffer Management and Scheduling for Video Streaming over Wireless Shared Channels*, in Proc. Packet Video Workshop, 2004.
[7] Y. Bai, M.R. Ito, *Application-Aware Buffer Management: New Metrics and Techniques*, IEEE Transactions on Broadcasting, vol. 51, no. 1, pp. 114–121, March 2005.
[8] S. Kozlov, P. v. d. Stok, J. Lukkien, *Adaptive scheduling of MPEG video frames during real-time wireless video streaming*, in Proc. 6th Int. Symp. on a World of Wireless Mobile and Multimedia Networks, 2005.
[9] P. Seeling, M. Reisslein, B. Kulapala, *Network Performance Evaluation Using Frame Size and Quality Traces of Single-Layer and Two-Layer Video: A Tutorial*, IEEE Communications Surveys & Tutorials, Third Quarter 2004, vol. 6, no. 3.
[10] J. Klaue, B. Rathke, A. Wolisz, *EvalVid - A Framework for Video Transmission and Quality Evaluation*, Proc. 13th Conference on Modelling Techniques and Tools for Computer Performance Evaluation, USA, September 2003.
[11] F. Fitzek, P. Seeling, M. Reisslein, *VideoMeter tool for YUV bitstreams*, acticom mobile networks, October 2002.
[12] F. Hartanto, H. R. Sirisena, *Cumulative inter-ADU jitter concept and its applications*, in Proc. 10th Int. Conf. on Computer Communications and Networks, pp. 531-534, 2001.
[13] L. Tionardi, P. Hartanto, *The use of cumulative inter-frame jitter for adapting video transmission rate*, in Proc. IEEE TENCON 2003, vol. 1, pp. 364–368, October 2003.
[14] *ffmpeg*, http://ffmpeg.mplayerhq.hu/ .
[15] *football video*, http://www.tkn.tu-berlin.de/research/evalvid/cif/football_cif.yuv.bz2 .
[16] ITU-T, *Methods for subjective determination of transmission quality, ITU-T Recommendation P.800*, 1996.
[17] ITU-T, *Principles of a reference impairment system for video, ITU-T Recommendation P.930*, 08/1996.
[18] IKR, *IKR Simulation Library*, http://www.ikr.uni-stuttgart.de/IKRSimLib
[19] M. C. Necker, C. Gauger, S. Kiesel, U. Reiser, *IKREmuLib: A Library for Seamless Integration of Simulation & Emulation*, in Proc. MMB 2006, Nuremberg, Germany, March 2006.
[20] M. C. Necker, A. Weber, *Impact of Iub Flow Control on HSDPA System Performance*, in Proc. IEEE PIMRC 2005, Berlin, Germany, September 2005.
[21] M. C. Necker, A. Weber, *Protocol Interference between Up- and Downlink Channels in HSDPA*, in Proc. PIMRC 2006, Helsinki, Finland, September 2006.
[22] N. Challa, H. Cam, *Cost-Aware Downlink Scheduling of Shared Channels for Cellular Networks with Relays*, in Proc. IEEE IPCCC, Phoenix, AZ, USA, April 2004.
[23] T. Szigeti, C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs and VPNs*, Cisco Press, USA, 2005.