DTU Library

Congestion control in wireless links based on selective delivery of erroneous packets

Korhonen, Jari; Perkis, Andrew; Reiter, Ulrich

Congestion Control in Wireless Links based on Selective Delivery of Erroneous Packets

Jari Korhonen[*]

Technical University of Denmark (DTU)

Department of Photonics Engineering

Oersteds Plads, bldg 343, 2800 Kgs. Lyngby, Denmark

Tel: +45 4525 6594, Fax: +45 4525 6581

E-mail: jark@fotonik.dtu.dk

Andrew Perkis, and Ulrich Reiter

Norwegian University of Science and Technology

Centre for Quantifiable Quality of Service in Communication Systems (Q2S)[†]

O.S. Bragstads plass 2E, 7491 Trondheim, Norway

Tel: +47 735 {92383,92771}, Fax: +47 7359 2790

E-mail: andrew@iet.ntnu.no, reiter@q2s.ntnu.no

ABSTRACT

Traditionally, congestion control in packet networks is performed by reducing the transmission rate when congestion is detected, in order to cut down the traffic that overwhelms the capacity of the network. However, if the bottleneck is a wireless link, congestion is often accumulated because of retransmissions derived from bit errors. In this case, it might be beneficial to allow delivery of partly corrupted packets up to the application layer instead of reducing the transmission rate. This would decrease the number of retransmissions in the link layer and therefore relieve congestion, but at the cost of bit errors appearing in the packet payload. In this paper, we study a congestion control

---

[*] Corresponding author.

mechanism for streaming applications that combines traditional congestion control with selective link layer partial checksumming allowing bit errors in the less sensitive parts of data. We have compared the performance of the proposed mechanism against traditional congestion control with a simulation study. The results show that the proposed approach can improve the overall performance both by increasing the throughput over the wireless and improving the video quality in terms of *peak signal-to-noise ratio* (PSNR) by up to 8 dB, depending on the error conditions and the content.

Keywords: Multimedia communication, Video streaming, Rate control, Cross-layer optimization, UDP Lite

1. INTRODUCTION

Traditionally, congestion control in packet-switched networks is performed by reducing the transmission rate when congestion is detected. This kind of congestion control mechanism is included in *transport control protocol* (TCP), the most commonly used transport protocol carrying data reliably over *internet protocol* (IP) networks. In order to avoid buffer overflows and underflows in multimedia streaming applications, the transmission rate must match with the bitrate of the encoded media. This is why media coding rate and transmission rate should be adjusted jointly [1]. When prerecorded content is streamed, the most straightforward method is to store several versions of the content at the streaming server, each encoded with different rates and quality levels. Then, the server can choose the version that fits the channel bandwidth restrictions. This technique is referred to as *bitstream switching* [2]. A more advanced approach is to use scalable coding. In this scheme, the encoded bitstream can be split into several layers, one base layer and one or more enhancement

layers. The base layer alone is sufficient to produce a basic quality and the enhancement layers can be inserted in the transport stream if the bandwidth constraints allow so [1].

Rate control is the only option to combat congestion, if the amount of traffic is exceeding the absolute capacity of the network. However, wireless technologies, such as *wireless local area networks* (WLANs), are gaining more and more relevance in multimedia communications, and in wireless networks congestion can also be related to physical transmission errors. This is because a radio link is typically much more prone to physical transmission errors than fixed cables, and corrupted packets are discarded and retransmitted. For this reason, the actual throughput of a wireless link is in many scenarios much lower than the theoretical maximum throughput [3,4]. Nevertheless, it is sometimes better to receive partly corrupted data packets than lose them entirely. This was the motivation for developing a lightweight version of the *user datagram protocol* (UDP), denominated as UDP Lite [5]. UDP Lite uses a partial checksum covering only the protocol headers and the most vulnerable parts of the packet payload. When bit errors occur in the unprotected part of the packet, it is delivered to the application in spite of errors [5,6]. Evidently, the throughput observed at the application layer (goodput) can be substantially increased using UDP LITE in the presence of bit errors [7,8,9]. However, when partial checksums are used, the application must be capable of handling bit errors in the media stream. Several different schemes and mechanisms have been proposed in the literature to increase bit error resilience in multimedia coding and compression [10,11,12]. There are also different proposals for combining UDP Lite with *forward error correction* (FEC) to cope with errors [7,13].

In practice, all standards for wireless networking with commercial products employ some kind of bit error detection and correction mechanisms at the link or *medium access control* (MAC) layer

already. Typically, some level of FEC is employed to correct individual bit errors and short error bursts, whereas more extensive clusters of errors not correctable by FEC are recovered by retransmitting the damaged packets. Usually, retransmissions at the lower layers hide efficiently all the errors in the physical transmission channel from the upper layers, even without introducing very large observable retransmission delay [14]. This is one of the main reasons why UDP Lite has gained little popularity in practical systems [15].

Even though MAC layer retransmission works well when the channel utilization is reasonable, it may be beneficial to use partial checksums to complement traditional rate control when the radio link is highly congested, since retransmissions consume link capacity and reduce the goodput. With partial checksums, higher goodput can be obtained, but at the cost of corrupted bits in the received data. Basically, the choice between conventional rate control and delivery of corrupted packets is therefore a choice between high source distortion (i.e. distortion derived from lossy compression with low bitrate) and low source distortion with additional channel distortion (i.e. the distortion derived from the bit errors). By intuition, it is not obvious which of these two approaches results in the lowest possible total distortion level. The related research has mostly focused on studying different combinations of FEC and UDP Lite [7,13], as well as comparing the perceptual impact of packet losses (traditional UDP) and bit errors (UDP Lite) [8,9]. To our knowledge, studies comparing the quality distortions derived from conventional rate control and bit errors due to partial checksumming have been largely omitted in related research, and our goal in this study is to improve the general understanding of this aspect.

In our earlier work, we have compared traditional rate control and rate control performed jointly with partial checksums with a network simulation. The results showed that the proposed system

using partial checksums improves both the throughput and the measured video quality in terms of PSNR, when compared against traditional rate control [16]. In this paper, we present further simulation results with more realistic video bitstreams and larger network configurations. We also discuss the limitations and scope of validity of our results with more details.

The paper is organized as follows. In Section 2, an overview of congestion control for streaming applications is given. In Section 3, we review the potential benefits and problems with delivering corrupted datagrams to the application. In Section 4, we present and discuss simulation results, comparing the performance of conventional congestion control and the proposed technique based on delivery of corrupted datagrams. A more general discussion is included in Section 5. Finally, the concluding remarks are given in Section 6.

2. TRADITIONAL CONGESTION CONTROL

The dominating transport protocols used in today's Internet are strongly based on the idea of dumb routers and intelligent end systems. Therefore, congestion control is typically performed on an end-to-end basis. In the traditional TCP, packet losses are used as a criterion for rate adjustment, since a packet loss is assumed to indicate congestion. TCP reduces its transmission rate aggressively when a packet loss is observed. This kind of behavior is not acceptable when smoother variations in transport rate are desired, which is typically the case with streaming applications. This is why equation-based *TCP-friendly rate control* (TFRC) has been adopted as a commonly accepted alternative to TCP for real-time applications. With TFRC, the receiver estimates the available bandwidth continuously from the packet loss rate and adjusts the transmission rate accordingly [17].

It is worth noting that the range of different possible transmission rates for streaming applications is restricted by the used media format and it is usually much more limited than in traditional TCP applications. Most of the practical media coding and compression standards do not support *fine grain scalability* (FGS). When the number of available levels of transmission rates is limited, smooth rate adjustment is not viable as such and the full benefit of the TFRC scheme cannot be obtained. However, the basic principles of traditional congestion control would still be valid: whenever congestion is detected, a lower transmission rate must be used.

In streaming applications, timely delivery is more critical than reliability. This is why streaming applications typically employ unreliable UDP as a transport protocol instead of reliable TCP, and TFRC is implemented on top of UDP on the application layer. An interesting alternative for UDP is *datagram congestion control protocol* (DCCP). DCCP is an unreliable transport protocol that offers congestion control and many other useful features for streaming applications, including partial checksums [18]. Unfortunately, DCCP has not gained wide popularity yet, and it is not supported by most of the currently available IP protocol stack implementations. Nevertheless, even though we focus on UDP Lite in this paper, the cross-layer approach we have taken is not strictly bound to UDP Lite. The same ideas could be implemented using DCCP as transport protocol as well.

The assumption of congestion derived losses is valid in wireline networks, in which the number of physical transmission errors is very low. However, in wireless networks this assumption is challenged, since physical errors play a bigger role and may also cause packet losses. Optimally, the rate control mechanism would be able to distinguish between packet losses caused by bit errors and those caused by congestion in order to choose the optimal behavior accordingly. This is why several end-to-end packet loss differentiation techniques have been developed to be employed together with

rate control. Performance comparisons for such techniques are available in [19,20]. According to the performance studies, the use of packet loss differentiation generally improves the performance of TFRC, but the reliability of the algorithms depends on the network topology, competing media flows, and even the used queuing schemes. More reliable information about bit errors conditions in the wireless link can be obtained by detecting the errors in the link layer and conveying this information up to the congestion control protocol using some kind of cross-layer notification mechanism. However, real life experiments have shown little success with cross-layer notifications to enhance performance of TCP over IEEE 802.11 WLAN [21]. Pack et al. have proposed an enhanced TFRC with awareness of link layer retransmissions based on cross-layer notifications [22], but this proposal has been validated by simulations only. Apparently, the practical advantage of cross-layer notifications for multimedia streaming is still an issue to be explored more comprehensively.

As discussed in Section 1, most of the practical wireless systems use MAC layer retransmissions. This is why packet loss differentiation has little relevance in practice. When MAC layer retransmissions are employed, bit errors do not appear as packet losses but reduced goodput, since part of the total link capacity is used for retransmissions, which can easily lead to congestion in the wireless *access point* (AP). Congestion caused by bit errors is not easily differentiated from "normal" congestion, since they are, in fact, conceptually the same thing. This is why most packet loss differentiation mechanisms described in literature do not consider wireless systems using link layer retransmissions. Link layer retransmissions have been taken into consideration in [22], but the major difference to our work is that the use of erroneous packets at the application layer has not been addressed. In this paper, the question of packet loss differentiation is largely omitted since we

operate in a simulation environment, where the wireless channel is known to be the performance

bottleneck. In the real life, this assumption is not necessarily valid, and packet loss differentiation

could be useful to complement the rate control approach studied in this paper.

3. DELIVERY OF CORRUPTED DATAGRAMS

Several independent studies show that in the presence of bit errors, data goodput can be significantly

increased in different kinds of network configurations and scenarios by using UDP Lite [7,8,9].

However, the improved link capacity usage comes at the cost of bit errors occurring in the media

stream. This is why the advantage of using UDP Lite depends highly on how the bit errors are

handled in the application layer. Regarding the video quality, different studies concerning UDP Lite

have led to different conclusions, depending on the error resilience of the video codec and the bit

error characteristics in the studied scenario. For example, Khayam et al. [7] concluded that in spite

of improved throughput, video quality does not improve with UDP Lite, unless additional FEC is

employed. On the other hand, according to Singh et al. [8], as well as Vehkaperä et al. [9], some

video quality improvement can be gained with UDP Lite together with an error resilient codec, but

the results are highly dependent on the bit error characteristics.

Obviously, UDP Lite requires either a bit error resilient media codec or a relatively strong FEC. The

use of FEC leads to less efficient utilization of the link capacity, due to required redundancy

overhead. On the other hand, bit error resilience comes practically always more or less as a trade-off

for compression efficiency. This is why bit-error resilience is omitted in many state-of-the-art media

compression standards. In this paper, we focus on the H.264/AVC video codec, which does not

support bit error resilience on the syntax level. It is, however, possible to achieve some degree of

error robustness by detecting syntax violations during decoding and using traditional error concealment to recover the remaining macroblocks in a slice [23,24].

In practice, the proportion of impacted frames depends on both bit error rate and media coding rate, as illustrated in Figure 1. When the bitrate of the encoded video gets higher, more bits are required for each frame and the likelihood that an encoded frame is hit by bit errors is higher. Self-contained intra frames (I-frames) usually require more bits than predicted inter frames (P- and B-frames), and they are therefore more likely to be impacted. Unfortunately, errors in I-frames are propagated to the predicted frames. Nevertheless, spatial error propagation can be restricted by splitting each frame in smaller slices and interleaving macroblocks among slices using tools such as *flexible macroblock ordering* (FMO) in H.264/AVC [24,25,26]. In our earlier work, we have achieved a reasonable level of bit error resilience for H.264/AVC video by using small slices and a modified reference decoder capable of detecting decoded data elements with illegal values [16,24]. In some cases, mutated values are legal but corrupted, causing annoying visual artifacts. Some of these artifacts can be detected and concealed using appropriate post processing algorithms [16].
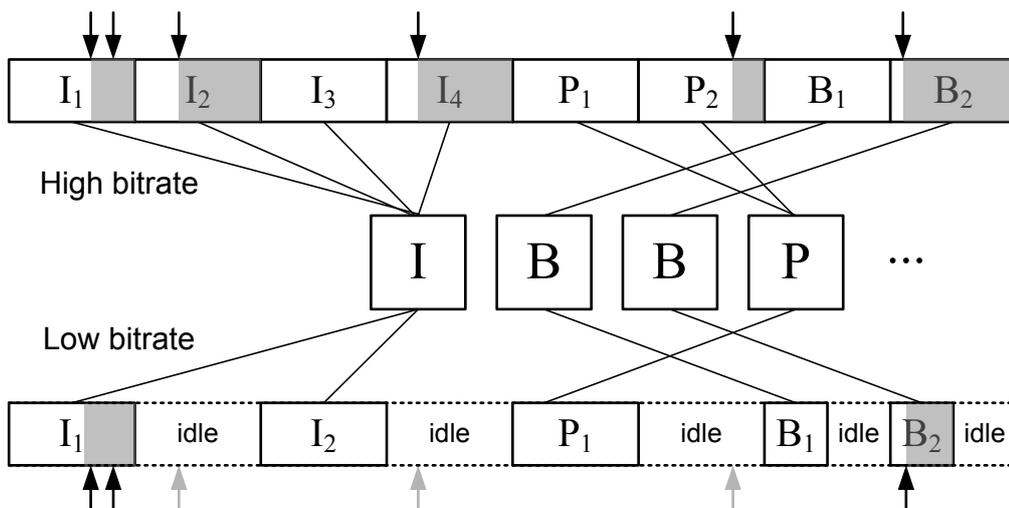
Figure 1. Impact of bit errors on two video streams encoded with different bitrates. Arrows show the positions of bit errors in the transmission channels and grey color denotes impacted frames.

Obviously, the benefits of UDP Lite cannot be achieved if link layer retransmissions are hiding the bit errors from upper layers. When UDP Lite was originally proposed in [6], the authors suggested two alternative solutions for this problem. First, link layer protocols could be modified so that they peek at the partial checksum coverage field of UDP Lite packets and perform partial checksum computation at the link layer already. Second, link layer protocols could detect UDP Lite packets and omit error detection for these packets altogether. Both of the proposed approaches require support from the link layer implementations. In fact, UDP Lite is today often considered irrelevant, since there seems to be little motivation for developers of low level protocols to allow delivery of erroneous packets [15].

Even if UDP Lite is used and link layer error detection omitted, bit errors located in the protected area would still cause packet drops. Some kind of mechanism would be useful to recover the lost packets in this case. Of course, FEC or retransmissions can be implemented at the transport or application layer, but link layer error control is much more efficient in terms of delay and overall network capacity utilization than end-to-end FEC or retransmissions. This is why the use of UDP Lite in parallel with link layer retransmissions is an option that deserves serious consideration. This is also the approach we have taken in our study.

4. SIMULATIONS

4.1 Setting

In our earlier work, we have proposed a concept system that combines conventional rate control with MAC layer partial checksums. In our system, UDP Lite datagrams with partial checksum coverage are identified by the MAC layer, and the actual checksum computation and bit error detection are performed at the MAC layer already. This is why we refer to the partial checksum mode as *MAC Lite mode*. In brief, the system works as follows: if congestion is detected in the AP of the *wireless local area network* (WLAN), the sender first switches into partial checksum mode with minimum coverage (only the protocol and media headers will be covered). We have adopted an intelligent packetization strategy to suit partial checksumming: several small H.264/AVC *network adaptation layer units* (NALUs) are allocated in each packet, but the most sensitive part in the beginning of each NALU is extracted and packet separately in the beginning of the payload, in the area that is protected by the partial checksum. More details of the packetization scheme can be found in [24,25]. On top of MAC Lite, conventional rate control is performed: if congestion is not sufficiently relieved by delivering partly corrupted datagrams, the transmission rate will be reduced. If the congestion is relieved later, the transmission rate is increased, and when the full rate is achieved again, the system starts to increase the proportion of the protected part of the media packets, until full protection is regained. The flowchart in Figure 2 illustrates the functionality of the joint MAC Lite and rate control mechanism.

Start

Set minimum rate and maximum checksum coverage

Feedback from network

Decrease transmission rate

Switch to minimum checksum coverage

Bottleneck link utilization

High (congestion)

Low (underutilization)

Increase transmission rate

Increase checksum coverage

No

No

Normal

No

No

Yes

Minimum rate achieved?

Yes

Minimum checksum coverage?

Maximum rate achieved?

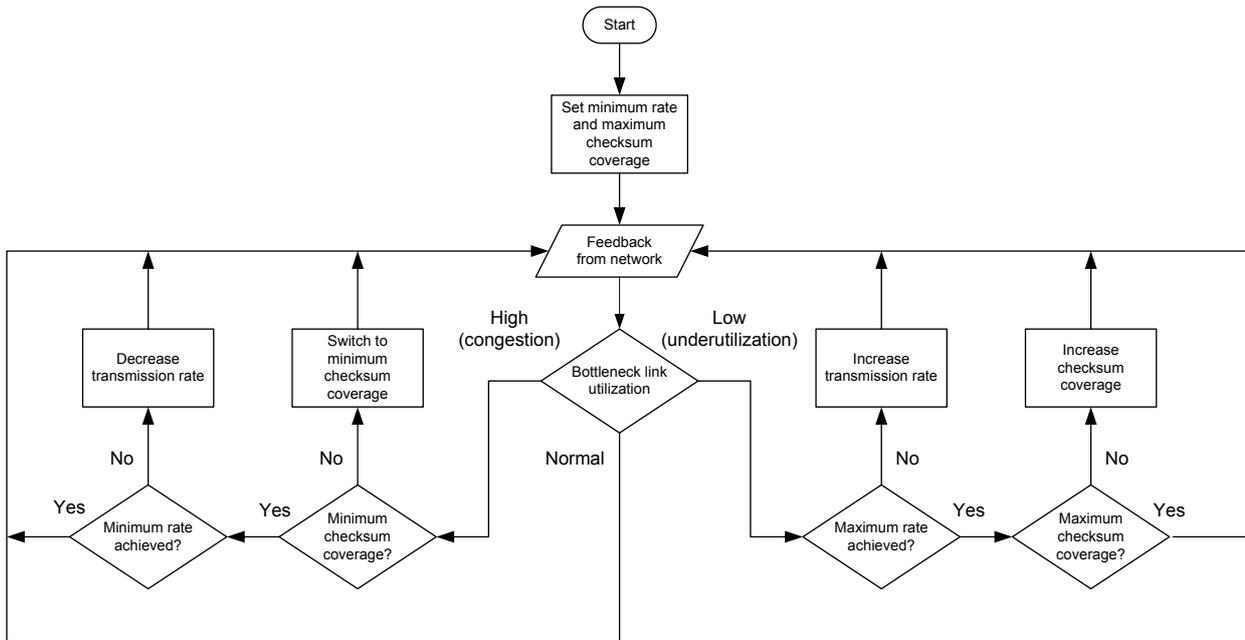Yes

Maximum checksum coverage?

Yes

Figure 2. Flowchart for the proposed congestion control mechanism.

It is worth noting that since the number of transmission rates is limited by the number of bitstreams to switch between, our test configuration does not even allow much flexibility to experiment different rate control mechanisms. The used thresholds for bottleneck link utilization (buffer fullness) as well as the steps for increasing UDP Lite partial checksum coverage have been defined specifically for the used simulation environment, and optimization of the protocol for real-life environments would require more work. However, the concept of using UDP Lite in conjunction with congestion control is not linked to any specific rate control mechanism, and it could be replaced by any other relevant scheme, such as TFRC. In practice, this would require larger number of bitstreams or the use of scalable coding with fine grain scalability.

To study the proposed system, we have implemented a test application in the ns-2 network simulator environment [27]. Rate adjustment is performed via bitstream switching: the test server application

uses ten different content files, each corresponding to the same content with different quality and bitrate. In this study, we do not want packet losses to occur, since losses would also contribute to quality distortion and our intention is to compare the distortions caused by low coding rate and bit errors only. This is why buffer fullness in the wireless AP is used as a congestion indicator, instead of packet losses. For this purpose, we have defined a specific field in the media packet header that carries the number of packets enqueued at the AP. If the number of packets exceeds a predefined threshold, it is taken as an indicator of high utilization (congestion). This kind of cross-layer communication mechanism may not be easily implemented in real-life network devices, but it is considered appropriate for the concept study, since the rate control mechanism is not strictly bound to a certain type of congestion indicator.

In our earlier work [16], we have studied the proposed scheme with a minimalistic network configuration with two senders and receivers only, both sending H.264/AVC streams encoded in intra mode. In this paper, we have extended our study by using a larger number of senders and receivers, as well as more realistic video streams with temporal prediction (i.e. P- and B-frames are also enabled). The network topology is depicted in Figure 3. There are n independent streaming servers (S_1-S_n) in the wired part of the network, sending data to the respective mobile receivers (R_1-R_n) located in the same WLAN cell. In our study, the main focus is on the performance of the wireless radio link. This is why such a simple topology in the wired part was considered appropriate for our purposes.
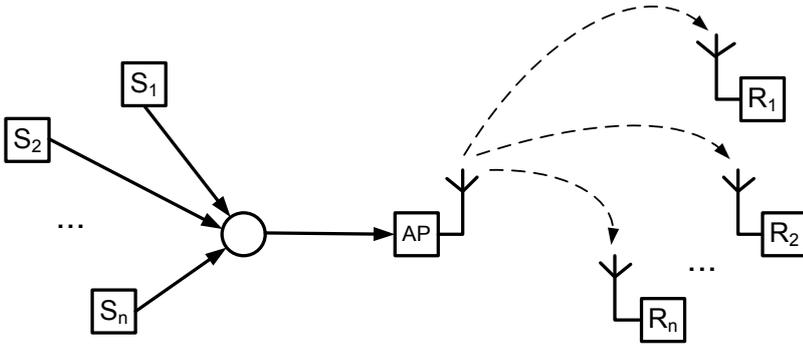
Fig. 3. The network configuration used for the simulations. Arrows indicate the direction of the streamed media flow.

A bitstream switching strategy has been adopted to implement rate control. Ten different encoded sequences were generated from two video content files ('Akiyo' and 'Football' both in CIF format, i.e. 352x288 pixels), each with different *quantization parameter* (QP). *Group of picture* (GOP) structure IBBPBBPBBP (10 frames per GOP) was used. 'Akiyo' represents content with very little temporal activity, and it can be compressed efficiently with low bitrate using temporal prediction. In contrast, 'Football' sequence contains high levels of spatial and temporal activity. Due to the different characteristics of the test sequences, the resulting ranges for bitrates and PSNR values are different for both sequences, as summarized in Table 1.

Table 1. Quantization parameters, bitrates, and PSNR values for the test video sequences.

QP	PSNR (Akiyo)	Bitrate (Akiyo)	PSNR (Football)	Bitrate (Football)
51	30.69	76 kbit/s	24.37	82 kbit/s
48	31.90	85 kbit/s	25.89	118 kbit/s
45	33.38	96 kbit/s	27.53	179 kbit/s
43	34.38	109 kbit/s	28.74	242 kbit/s
41	35.40	122 kbit/s	29.86	312 kbit/s

39	36.55	138 kbit/s	31.10	416 kbit/s
37	37.57	154 kbit/s	32.35	548 kbit/s
35	38.41	173 kbit/s	33.43	689 kbit/s
33	39.32	194 kbit/s	34.63	883 kbit/s
30	41.16	240 kbit/s	36.70	1.23 Mbit/s

For simulation purposes, the offset and length of each NALU within each video file has been stored in a configuration file so that there is no need to handle the actual video content during the simulation, but only dummy data is transmitted. Bitstream switching is implemented in the test server application by using the bitstream indices in configuration files. The test client collects the NALU offsets and bitstream indices and saves them in an output file. The information in the output file is used to generate the resulting mixed video bitstream offline after the network simulation. Bit errors were generated using a two-state Gilbert-Elliot (GE) model and the resulting bit errors were collected in an error pattern file. Then, bit errors are applied to the constructed video streams offline after network simulation in parallel of generating the resulting mixed video stream. Practical measurements have shown that in spite of some statistical inaccuracies of the GE model, it is anyway a useful tool to simulate bit error characteristics in a real wireless link [28], and considered appropriate for this study. The components of the simulation environment are shown in Figure 4.
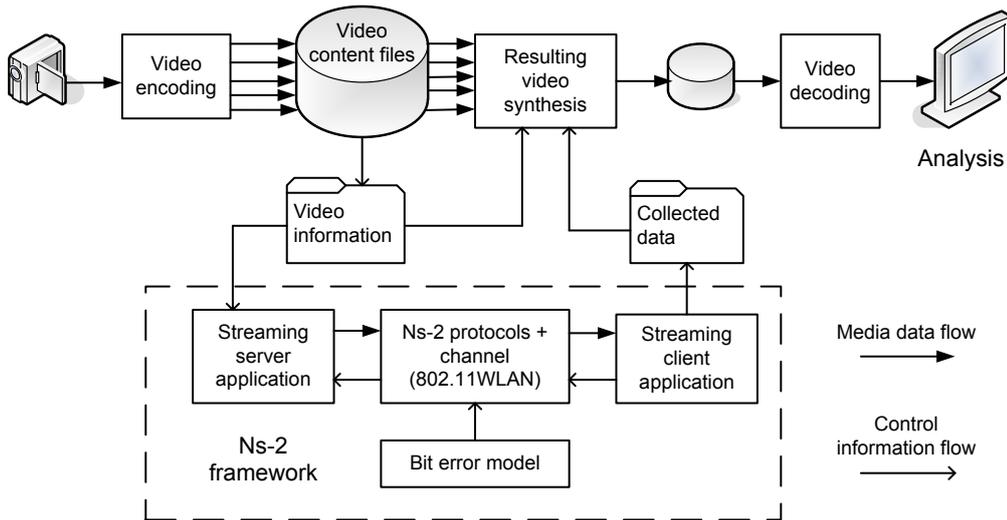
Figure 4. The main components and data flows of the simulation environment depicted.

4.2 Results

In the first set of experiments, we have used a fixed number of streaming servers and clients. To avoid unrealistic traffic burst, servers do not start transmission concurrently, but with short intervals. Different GE model parameters have been used to produce different bit error rates. In the model, there are two states: 'Good' and 'Bad'. In 'Good' state, all the bits are transmitted correctly, whereas in 'Bad' state, a bit is inversed with the probability P_{Err}. In the 'Good' state, the transition probability to 'Bad' state is $P_{Good->Bad}$, and in the 'Bad' state, the transition probability to 'Good' state is $P_{Bad->Good}$. With a complex bitstream syntax with flags and variable length codes, one single bit error can cause error propagation that can be as harmful as a longer bit error burst [29]. This is why the length of the error burst is less relevant a parameter as the frequency of bursts, and we have therefore varied only $P_{Good->Bad}$. $P_{Bad->Good}$ is kept constant (0.05), as well as P_{Err} (0.5). With these parameters, the average burst length is 20 bits, and each burst contains on average 10 erroneous bits. Parameters for

the model are given in Table 2. Different practical measurement studies show a relatively wide range

of BERs in real wireless channels from as low as 10^{-4} up to 10^{-1}, depending on the used technical

characteristics of the system and environmental factors [30,31]. The parameters used in our study fit

well to the appropriate parameter space.

Table 2. Gilbert-Elliot model parameters.

$P_{\text{Good->Bad}}$	Bit error rate
$5.0 \cdot 10^{-5}$	$5.0 \cdot 10^{-4}$
$1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-3}$
$1.5 \cdot 10^{-4}$	$1.5 \cdot 10^{-3}$
$2.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-3}$
$2.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-3}$

In the first experiment, we have studied the dynamics of the proposed scheme. Figure 5 shows how

the goodputs measured at three receiver nodes evolve in an example scenario, where two of the

streams are transported using MAC Lite mode and one is using the plain bitstream switching without

partial checksums. The content is 'Akiyo' and BER=$2 \cdot 10^{-3}$. The results show that after the

convergence phase goodput is very similar for the two streams using MAC Lite. Traditional

congestion control gains somewhat lower goodput, since it reduces transmission rate in the first

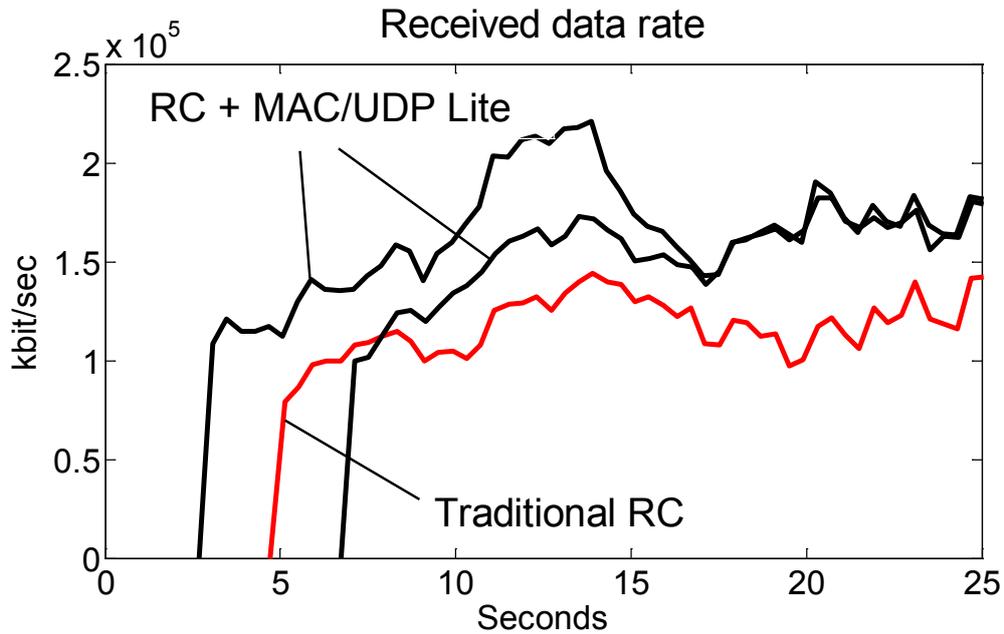place without trying partial checksumming first.

Figure 5. Evolution of goodput measured at the receivers in a scenario with three competing streams.

However, the issue of fairness is more complex in practice, since the utility of the streams transported with MAC Lite may be impacted by bit errors and it is therefore not directly comparable to fully protected streams with equal transmission rate. In Figure 6, the respective curves for wireless link utilization are shown, and the situation looks very different. We can notice a drastic decrease in the channel utilization curve for the first stream at around 5 seconds, as the sender switches from conventional mode to MAC Lite mode. In fact, the MAC Lite streams consume significantly smaller proportion of the bottleneck channel capacity than the conventional stream after the convergence phase, since they can avoid most of the MAC layer retransmissions. Readers interested in more detailed analysis of the relationship between checksum coverage and the amount of derived retransmission traffic may refer to studies about packet size optimization [32,33].
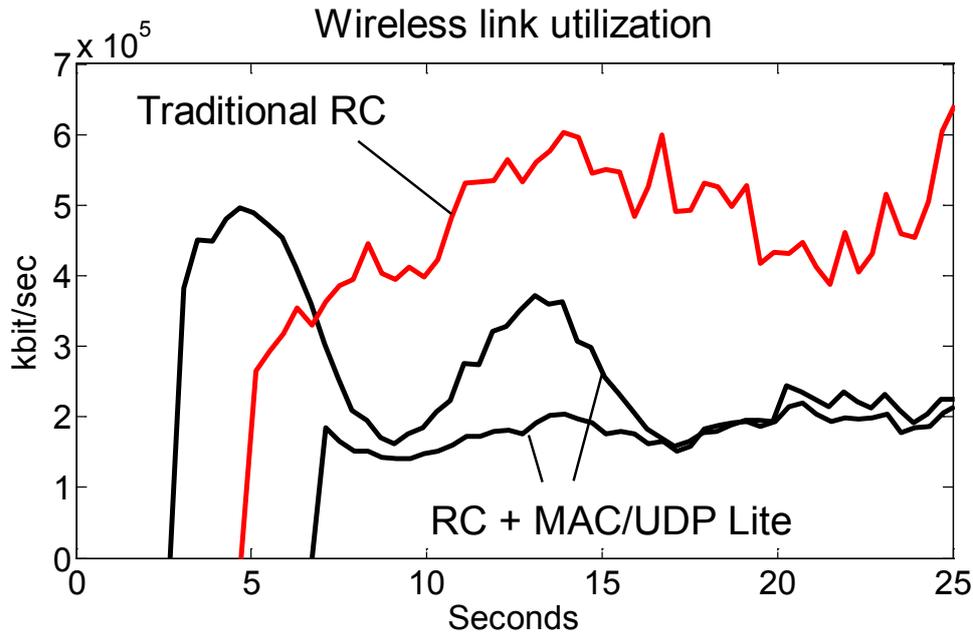
Figure 6. Evolution of wireless link utilization in a scenario with three competing streams.

In Figure 7, the average goodput is shown in different cases, when n=5. As BER increases, at some point packet losses cannot be avoided any more. We have omitted the results influenced by packet losses, and this is why some curves are terminated earlier than others. The results for goodput and video quality have been measured in the area where all the servers are transmitting and the transmission rates have converged. When MAC Lite is used for all streams, goodput remains approximately the same, even though the bit error rate (BER) increases. In this example scenario, the maximum bitrate for 'Akiyo' sequence is 240 kbit/s, and for 'Football' sequence 1.23 Mbit/s (see Table 1). The 'Akiyo' sequence reaches the maximum bitrate when BER is low (below $0.5 \cdot 10^{-3}$), and in this case there is no difference between traditional rate control and MAC Lite. When BER increases, traditional rate control starts to decrease the goodput, whereas with MAC Lite it remains approximately the same. For the 'Football' sequence the situation is different, since the radio link capacity is not sufficient to reach the maximum bitrate even in error free conditions. When BER=0,

the goodput with traditional rate control is actually slightly higher than with MAC Lite. This is because the attempt to adjust the partial checksum coverage before traditional rate control causes additional delay before the transmission rate is adjusted, leading to larger rate fluctuation and lower average goodput when congestion occurs without bit errors. This problem could be alleviated by providing the system with more intelligence to omit partial checksumming entirely when there are no bit errors present. If there were less active nodes, the shape of the curves for 'Football' would resemble that for 'Akiyo', except that the maximum goodput would be higher. When traditional rate control is used and BER is high, the resulting goodput is nearly the same for both sequences in any case.
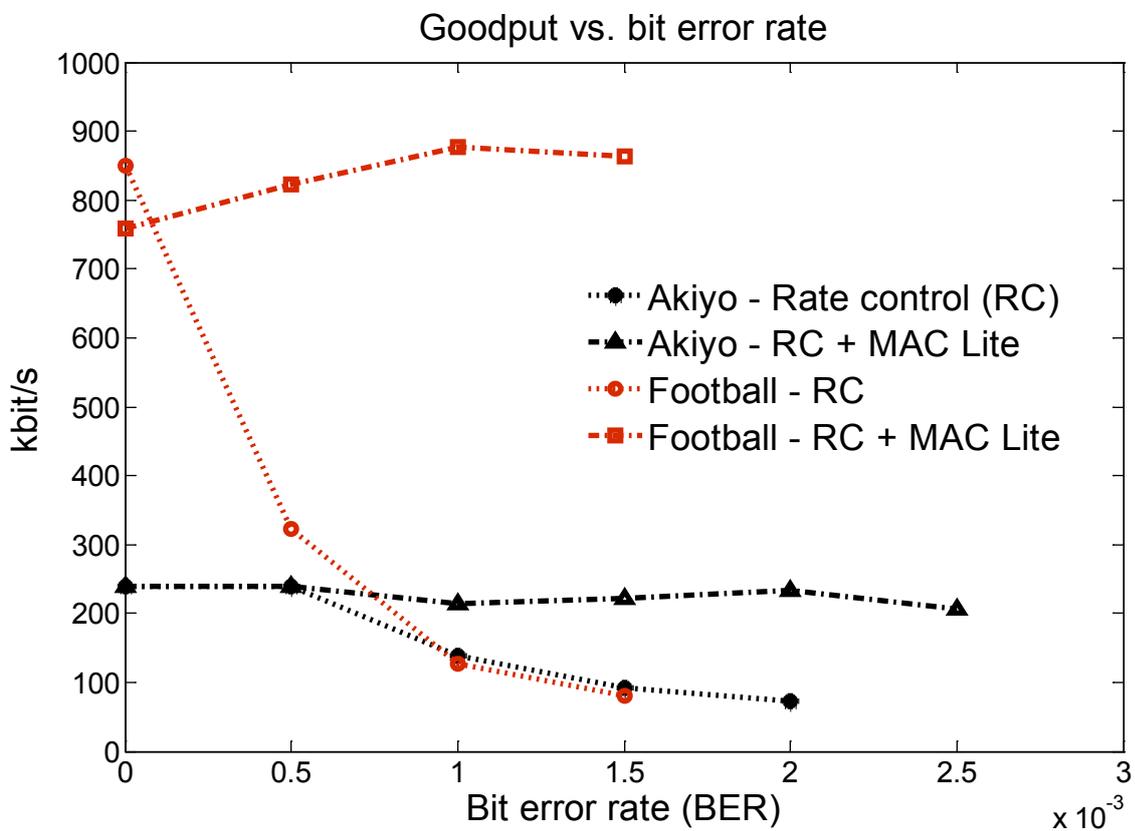

Figure 7. Goodput measured at one of the receivers with different bit error rates in the radio channel.

The throughput analysis curves clearly show that in the presence of bit errors, the received bitrate can be increased significantly by using the MAC Lite scheme. However, the more interesting question is, if the video quality is also improved. In order to study the resulting video quality, we have applied the collected bit error traces gained in simulations to the resulting video bitstreams and decoded them using the H.264/AVC reference codec, modified to support bit error resilience as reported in [16,24]. Since we have applied temporal prediction in the encoded video sequences, we have also replaced spatial error concealment used in [16] with temporal error concealment, i.e. missing or damaged macroblocks are copied from the preceding frame.

The quality of the resulting decoded video sequences has been evaluated by measuring PSNR for the received sequence in each test case. In spite of the limitations and criticism of PSNR, it is still the most commonly used objective metric to estimate the observed subjective quality of video streams, and the correlation between relative subjective quality and PSNR has been found to be high when the content and distortion type are fixed [34]. This is why we have used it as a quality metric in our study as well. The validity of the PSNR results for a subjective quality assessment is discussed in more details in Section 4.3.

The respective PSNR curves for the experiments explained above are shown in Figure 8. As a comparison to Figure 7 shows, the relationship between goodput and PSNR is not straightforward. However, the use of partial checksums seems to improve the performance of the system also in terms of video quality. The higher the BER, the more beneficial the MAC Lite mode gets. Partial checksumming also increases the range of bit error levels for which the system still works. Figure 9 shows the PSNR curves for the case that the BER is kept constant, but the number of active nodes transmitting video streams is altered. In this scenario, $BER=1.0 \cdot 10^{-3}$. The results show that when the

number of streams increases, traditional rate control performs worse compared to the partial

checksumming. For higher BER, the benefit of MAC Lite would be even larger than in this example.
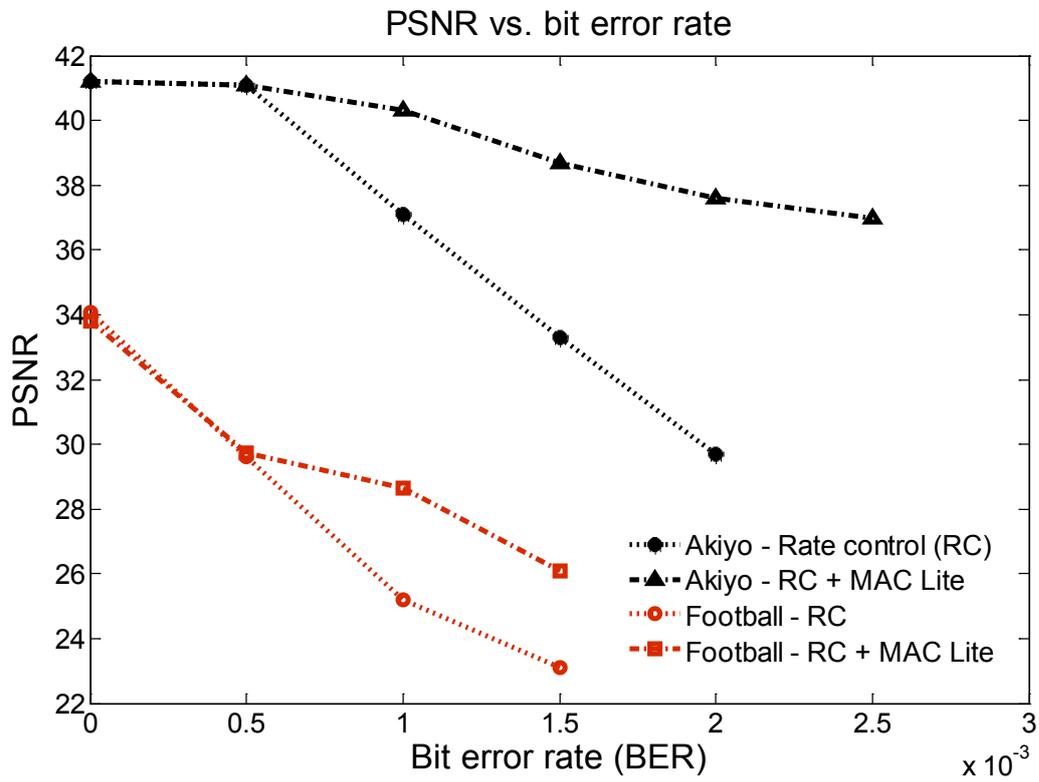

Figure 8. PSNR of the resulting video with different bit error rates in the radio channel.
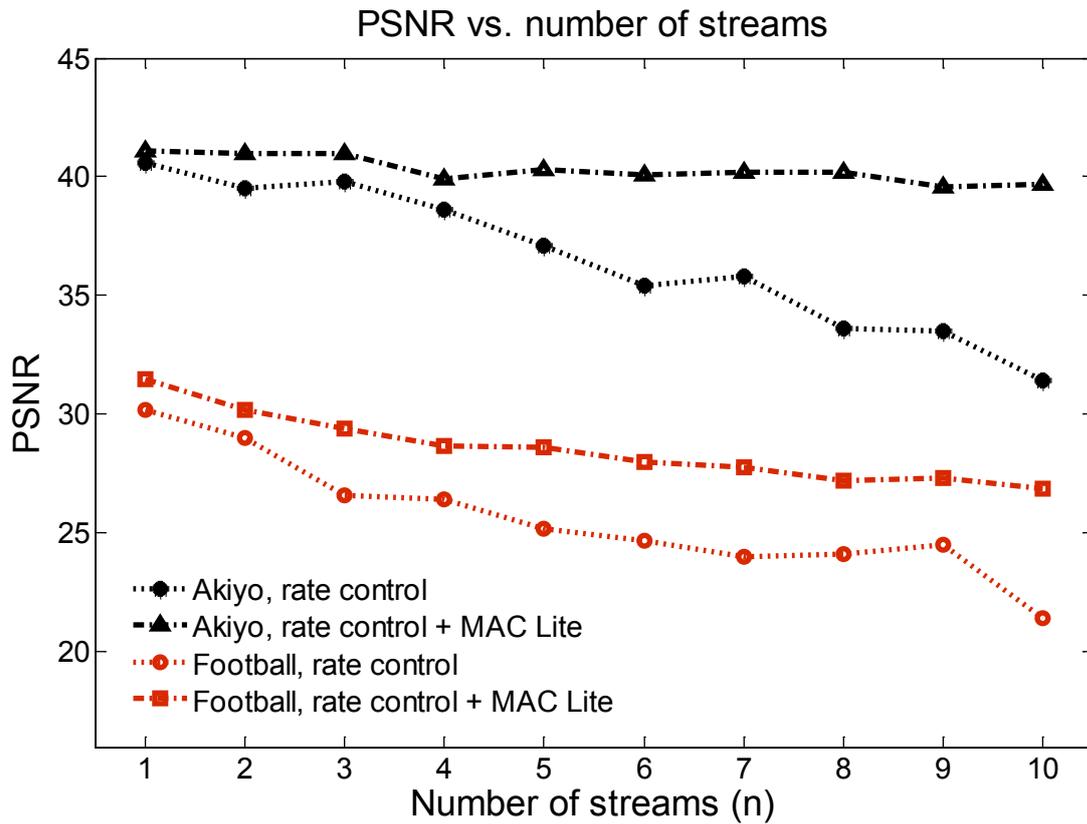
Figure 9. PSNR values compared against the number of active streams with fixed BER=$1.0 \cdot 10^{-3}$.

4.3 Subjective video quality

According to both goodput and PSNR results, the benefits of MAC Lite mode seem to be obvious.

However, it must be noted that PSNR is not a perfect metric for perceived video quality. For

example, there is evidence that a video with strongly fluctuating quality from frame to frame is

perceived as more annoying than a more stable video with lower average PSNR value [35]. The

quality impairment caused by a decreased coding rate is substantially different from the quality

distortion due to bit errors: a low coding rate impacts the general quality of the whole frame, whereas bit errors appear seemingly randomly in temporally and spatially limited areas. The difference between source distortion and channel distortion is illustrated with example frames in Figure 10. Due to the qualitative difference of these distortion types, PSNR cannot be considered as a fully reliable metric for comparing the perceptual impact of source and channel distortion.

a) Source distortion   b) Channel distortion

Figure 10. Example frames with source distortion caused by lossy compression (a) and channel distortion caused by bit errors (b).

In our other work, we have performed subjective tests to compare the subjective experience of source and channel distortion [29,36]. Our results suggest that in relatively static video sequences, such as 'Akiyo', PSNR tends to overestimate the quality, when channel errors are present and the source distortion level is low. On the other hand, for sequences with intensive spatial and temporal activity, such as 'Football', the difference is diminished or even turned into the opposite: PSNR may even underestimate the perceived quality level in the presence of channel errors. As the source

distortion level increases, the perceived impact of channel distortion becomes less dominating, and the bias in PSNR (towards favoring channel distortion) disappears. This applies to both static and intense video contents. These results suggest that the perceived subjective quality difference in the 'Akiyo' sequence favoring MAC Lite may not be as significant as the PSNR results show (see Figures 8 and 9). On the other hand, the subjective quality difference in the 'Football' sequence favoring MAC Lite may be even more significant than the results indicate. However, especially at high BER the difference in PSNR is so large (over 8 dB at largest) that there is no doubt about the general quality difference: partial checksum mode provides a favorable effect on subjective video quality, especially when BER is high and there is a lot of contending traffic (several video streams at the same time).

5. DISCUSSION

This study is continuation of our previous work published in [16]. Many of the open issues discussed in the earlier work have been addressed in this paper. In particular, we have used temporal prediction in our encoded video sequences (i.e. P- and B-frames have been enabled), temporal error concealment for missing or damaged macroblocks instead of spatial concealment, and we have examined network configurations with a larger number of active sending and receiving nodes. We have also included two different types of content to show that our conclusions are valid independent of content. The validity of PSNR in our work has been discussed and we referred to the subjective quality assessment experiments published in [29,36]. Whereas it is widely accepted that PSNR is not a good metric for comparing source and channel distortion, we have noted that PSNR can still be used for rough estimation of subjective quality. The bias favoring towards either channel or source distortion is not so large that it would compromise the general conclusions.

However, there are still some issues that remain open and provide opportunities for future research. In our study, we have used the standard error resilience and concealment mechanisms implemented in the H.264/AVC codec, such as *flexible macroblock ordering* (FMO), and copying the missing or damaged macroblocks from the preceding frame. Some further improvement could be achieved with more advanced error concealment mechanisms that take both spatial and temporal correlations into account [37]. Occasionally, there are bit errors that cannot be detected during the decoding process, since they do not violate the syntax, but they can still cause significant visible defects in the video. We have used a primitive post-processing algorithm to reveal damaged macroblocks by computing the gradients at the edges of each macroblock. This method improves the quality to some extent, but there are still opportunities to develop more efficient and reliable error detection and concealment schemes designed specifically for bit errors.

The results in [29] show that intra-only coding mode is more robust against bit errors than coding with temporal prediction. However, the shape of the curve for quality degradation versus bit error rate is similar in both coding modes, when scaled to different applicable ranges of bit error rates. The cost of higher robustness is lower compression rate, and we can assume that similar or even higher improvement in robustness could be gained by using FEC to decrease the amount of residual bit errors. Therefore, FEC and selective delivery of packets with bit errors are not actually rival technologies, but they could be used to complement each other. It is not trivial to define the optimal trade-off between channel efficiency and overhead for more robust coding or FEC, and this would be an interesting topic for future research.

The most significant open issue in the proposed system is its applicability to real life network devices and infrastructures. We have studied the proposed system concept in a network simulation

environment, where it is relatively easy to modify the networking protocols and implement *ad hoc* mechanisms for cross-layer signaling. In real networks, such modifications would be more difficult to implement, due to the standardized network protocols and devices that do not intrinsically support cross-layer information exchange. In addition, the proposed scheme is useful only in situations where a wireless link is the bottleneck. This is not always the case in realistic networking scenarios. Therefore, a real system would require sophisticated mechanisms for cross-layer signaling and for choosing between traditional rate control and MAC Lite mode dynamically. These considerations have been mainly left out of the scope of this paper. However, it is noteworthy that most of the proposals in the related literature have been validated with simulation studies only, and the few studies with results from practical experiments have some elements that are essentially different from our work, such as focus on reliable transport [21] or entirely disabled link layer retransmissions [38]. We acknowledge the importance of real world experimental results, and our intention is to perform practical studies with real wireless network devices in the future. We also hope that our results motivate developers of future networking devices and protocols to provide appropriate cross-layer control mechanisms that could be used to implement schemes such as the one proposed in this paper.

6. CONCLUSIONS

Traditionally, congestion control in multimedia streaming is performed by reducing the end-to-end transmission rate. This approach may not perform optimally if the bottleneck is a wireless link with high rate of physical transmission errors. In this paper, we have extended our previous study regarding a scheme that uses MAC layer partial checksums to allow delivery of partly damaged packets in order to reduce the number of link layer retransmissions when the radio link appears to

get congested due to bit errors. By simulations, we have shown that this method can improve the throughput significantly, compared to traditional rate control, especially when a high bit error rate is combined with a large amount of contending traffic. Naturally, bit errors in the media data cause channel distortion, but we have shown that by using appropriate mechanisms for bit error control, it is possible to achieve improved perceived quality, when compared against conventional rate control.

REFERENCES

[1]   D. Wu, Y.T. Hou, Y-Q. Zhang, and J.M. Peha, "Streaming Video over the Internet: Approaches and Directions," IEEE Trans. on Circuits and Syst. for Video Technol. 11 (2003) 282-300.

[2]   T. Stockhammer, and M. Hannuksela, "H.264/AVC Video for Wireless Transmission," IEEE Wireless Commun. 12 (2005) 6-13.

[3]   M. Arranz, R. Agüero, L. Muñoz, and P. Mähönen, "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks," Proc. of PIMRC'01, San Diego, CA, USA, Sep. 2001.

[4]   J. P. Singh, N. Bambos, B. Srinivasan, and D. Clawin, "Wireless LAN Performance under Varied Stress Conditions in Vehicular Traffic Scenarios," Proc. of VTC'02, pp. 743-747, 2002.

[5]   L. Larzon, M. Degermark, S. Pink, L. Jonsson, and G. Fairhurst, "The Leightweight User Datagram Protocol (UDP-Lite)," IETF RFC 3828, 2004.

[6]   L. Larzon, M. Degermark, and S. Pink, "UDP Lite for Real Time Multimedia Applications," HP Technical Report HPL-IRI-1999-001, Apr. 1999.

[7]   S. Khayam, S. Karande, H. Radha, and D. Loguinov, "Performance Analysis and Modeling of Bit Errors and Losses over 802.11b LANs for High Bit-Rate Real-Time Multimedia," Signal Processing: Image Communication 18 (2003) 575-595.

[8]   A. Singh, A. Konrad, and A. Joseph, "Performance Evaluation of UDP Lite for Cellular Video," Proc. of NOSSDAV '01, pp.117-124, Port Jefferson, New York, USA, Jun. 2001.

[9]   J. Vehkaperä, J. Peltola, J. Huusko, M. Myllyniemi, M., and M. Majanen, "Evaluation of Achieved Video Quality in Wireless Multimedia Transmission using UDP-Lite," IASTED IMSA '06, Honolulu, Hawaii, USA, Aug. 2006.

[10]   G. Cote, F. Kossentini, and S. Wenger, "Error Resilience Coding," in Compressed Video over Networks, M. Sun and A. Reibmann (Eds.), CRC Press, 2000, pp. 309-342.

[11]   Y. Takishima, M. Wada, M., and Hurakami, H., "Reversible Variable Length Codes," IEEE Trans. Commun. 43 (1995) 158-162.

[12]   R. Sperschneider, D. Homm, and L. Chambat, "Error Resilient Source Coding with Variable-Length Codes and its Application to MPEG Advanced Audio Coding," 109th AES Convention, Los Angeles, California, USA, Sep. 2000.

[13]   Z. Chen, "An Adaptive FEC to Protect RoHC and UDP-Lite H.264 Video Critical Data," J. Zhejiang Univ. SCIENCE A 7 (2006) 910-918.

[14]   J. Korhonen, and Y. Wang, "Effect of Packet Size on Loss Rate and Delay in Wireless Links," Proc. of WCNC '05, pp. 1608-1613, New Orleans, Louisiana, USA, Mar. 2005.

[15]   M. Welzl, "Passing Corrupt Data across Network Layers: An Overview of Recent Developments and Issues," EURASIP J. Appl. Signal Proc. (2005) 242-247.

[16]   J. Korhonen, and A. Perkis, "Wireless Congestion Control based on Delivery of Erroneous Packets," Proc. of VCIP '09, San Jose, California, USA, Jan. 2009.

[17]   S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based Congestion Control for Unicast Applications," Proc. of ACM SIGCOMM '00, pp. 43-56, Stockholm, Sweden, Aug. 2000.

[18]   E. Kohler, M. Handely, and S. Floyd, "Designing DCCP: Congestion Control without Reliability," Proc. of ACM SIGCOMM '06, pp. 27-38, Pisa, Italy, Sep. 2006.

[19]   S. Cen, P. Cosman, and G. Voelker, "End-to-End Differentiation of Congestion and Wireless Losses," IEEE/ACM Trans. on Networking 11 (2003) 703-717.

[20]   A. Boukerche, G. Jia, and R. Pazzi, "Performance Evaluation of Packet Loss Differentiation Algorithms for Wireless Networks," Proc. of PM2HW2N '07, pp. 50-52, Chania, Greece, Oct. 2007.

[21]   M. Welzl, M. Rossi, A. Fumagalli, M. Tacca, "TCP/IP over IEEE 802.11b WLAN: the Challenge of Harnessing Known-Corrupt Data," Proc. of IEEE ICC'08, Beijing, China, May 2008.

[22]   S. Pack, X. Shen, J. Mark,a and L. Cai, "A Two-Phase Loss Differentiation Algorithm for Improving TFRC Performace over IEEE 802.11 WLANs," IEEE Trans. on Wireless Commun. 6 (2007) 4164-4175.

[23]   L. Superiori, O. Nemethova, and M. Rupp, "Performance of a H.264/AVC Error Detection Algorithm Based on Syntax Analysis," Proc. of MoMM '06, Yogyakarta, Indonesia, Dec. 2006.

[24]   J. Korhonen, and P. Frossard, "Bit-error Resilient Packetization for Streaming H.264/AVC Video," Proc. of MV '07, pp. 25-30, Augsburg, Germany, Sep. 2007.

[25]   J. Korhonen, and P. Frossard, "Error Control for Video Streaming with Small Data Units," Proc. of MobiMedia '08, Oulu, Finland, Jul. 2008.

[26]   H.264/AVC Reference Software Archive. http://iphome.hhi.de/suehring/tml/ download/old_jm/, accessed 18. July 2007.

[27]   The Network Simulator - ns-2. http://www.isi.edu/ nsnam/ns, accessed 4. February 2010..

[28]   H. Wang, and N. Moayeri, "Finite State Markov Channel-A Useful Model for Radio Communication Channels," IEEE Trans. on Vehicular Technol. 44 (1995) 163-171.

[29]   J. Korhonen, and U. Reiter, "Analysis on the Perceptual Impact of Bit Errors in Practical Video Streaming Applications," Proc. of IMSAA '09, Bangalore, India, Dec. 2009.

[30]   A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a Wireless Link in an Industrial Environment Using an IEEE 80.11-Compliant Physical Layer," IEEE Trans. on Ind. Electronics 49 (2002) 1265-1282.

[31]   S. Khayam, and H. Radha, "Markov-based Modeling of Wireless Local Area Networks," Proc. of MSWiM'03, San Diego, CA, USA, pp. 100-107, Sep. 2003.

[32]   P. Lettieri, and M.B. Srivastava, "Adaptive Frame Length Control for Improving Wireless Link Throughput, Range, and Energy Efficiency," Proc. of IEEE INFOCOM'98, pp. 564-571, San Francisco, CA, USA, Mar. 1998.

[33]   J. Korhonen, and Y. Wang, "Effect of Packet Size on Loss Rate and Delay in Wireless Links," Proc. of WCNC'05, New Orleans, LA, USA, Mar. 2005.

[34]   Q. Hyunh-Thu, and M. Ghanbari, "Scope of Validity of PSNR in Image/Video Quality Assessment," Electronics Letters 44 (2008) 800-801.

[35]   M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz, "Subjective Impression of Variations in Layer Encoded Videos," Proc. of IWQoS '03 , pp. 137-154, Monterey, California, USA, Jun. 2003.

[36]   U. Reiter, and J. Korhonen, "Comparing Apples and Oranges: Subjective Quality Assessment of Streamed Video with Different Types of Distortion," Proc. of QoMEX '09, San Diego, California, USA, Jul. 2009.

[37]   A.S. Bopardikar, O.I. Hillestad, and A. Perkis, "Temporal Concealment of Packet-Loss Related Distortions in Video Based on Structural Alignment," Proc. of Eurescom Summit '04, Heidelberg, Germany, Apr. 2004.

[38]   J. Chesterfield, R. Chakravorty, S. Banerjee, P. Rodriguez, I. Pratt, and J. Crowcroft, "Transport Level Optimization for Streaming Media over Wide-area Wireless Networks," Proc. of WIOPT'04, Cambridge, UK, Mar. 2004.