INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2004; **17**:421–435 (DOI: 10.1002/dac.654)

Wireless video streaming with TCP and simultaneous MAC packet transmission (SMPT)[‖]

Frank H. P. Fitzek[1,§,¶] and Martin Reisslein[2,*,†,‡]

[1] *Acticom GmbH, Berlin, Germany*
[2] *Department of Electrical Engineering, Arizona State University, Goldwater Center MC 5706, Tempe, AZ 85287-5706, U.S.A.*

SUMMARY

Video streaming is expected to account for a large portion of the traffic in future networks, including wireless networks. It is widely accepted that the user datagram protocol (UDP) is the preferred transport protocol for video streaming and that the transmission control protocol (TCP) is unsuitable for streaming. The widespread use of UDP, however, has a number of drawbacks, such as unfairness and possible congestion collapse, which are avoided by TCP. In this paper we investigate the use of TCP as the transport layer protocol for streaming video in a multi-code CDMA cellular wireless system. Our approach is to stabilize the TCP throughput over the wireless links by employing a recently developed simultaneous MAC packet transmission (SMPT) approach at the link layer. We study the capacity, i.e. the number of customers per cell, and the quality of service for streaming video in the uplink direction. Our extensive simulations indicate that streaming over TCP in conjunction with SMPT gives good performance for video encoded in a closed loop, i.e. with rate control. We have also found that TCP is unsuitable (even in conjunction with SMPT) for streaming the more variable open-loop encoded video. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: multi-code CDMA; rate-controlled video; TCP; video streaming; wireless communication

1. INTRODUCTION

Market research finds that mobile commerce for 3G wireless systems and beyond will be dominated by basic human communication such as messaging, voice, and video communication

*Correspondence to: Martin Reisslein, Department of Electrical Engineering, Arizona State University, Goldwater Center MC 5706, Tempe, AZ 85287-5706, U.S.A.
† E-mail: reisslein@asu.edu
‡ web: http://www.fulton.asu.edu/~mre
§ E-mail: ff@kom.auc.dk
¶ He is now with Aalborg University, Aalborg, Denmark.
‖ A shorter preliminary version of this paper has appeared in the Proceedings of the International Conference on Third Generation Wireless and Beyond (3GWireless), San Francisco, CA, May 2002, 755–760.

Contract/grant sponsor: National Science Foundation; contract/grant number: career ANI-0133252
Contract/grant sponsor: National Science Foundation; contract/grant number: ANI-0136774

Received June 2003
Revised February 2004
Accepted March 2004

[1]. Because of its typically large bandwidth requirements, video communication (as opposed to the lower rate voice and the elastic e-mail traffic) is expected to emerge as the dominant type of traffic in 3G/4G wireless systems [2]. Video services are typically divided into real-time services (e.g. video conferencing) and streaming (e.g. web-based streaming of a news clip or the video feed from a surveillance camera). Our focus in this paper is on video streaming where the client may tolerate a small start-up delay before the playout commences. Video streaming schemes typically rely on the user datagram protocol (UDP) as the transport protocol. As streamed video applications become more popular, the Internet may become dominated by UDP streams. UDP streams, however, can lead to instability in the Internet. This is because UDP streams are not responsive to network congestion, as opposed to transmission control protocol (TCP) streams [3, 4]. With the explosive growth of multimedia applications, UDP-based multimedia streams have the potential to cause two major problems in the Internet: (i) congestion collapse, and (ii) unfair allocation of bandwidth among competing traffic flows [5]. Therefore, TCP friendly video streaming schemes are desirable.

There have been efforts to develop streaming protocols that emulate the 'friendliness' of TCP, see for instance References [6–8]. There have also been efforts to adapt the TCP mechanisms and parameters to accommodate video, see for instance references [9, 10]. In contrast, in this paper we investigate the direct use of TCP as the transport protocol for video streaming. Besides ensuring the stability of the Internet and achieving fair bandwidth allocation, TCP has a number of important advantages. First, TCP is reliable and ensures the lossless transport of the video stream. This is important for video streams that do not tolerate errors or error propagation, such as video for surveillance, tele-medicine, and some distance learning applications. Another advantage of TCP is that it ensures the in-order delivery of the video frames. We also note that there are a number of drawbacks to using TCP as the transport protocol for video streaming. First, TCP does not support multicast. Secondly, TCP's slow start mechanism and its Automatic Repeat reQuest (ARQ) based recovery from packet losses may interfere with the timely delivery of the video frames. Essentially, TCP trades off increased delay for lossless transport service.

As is generally accepted, using TCP as the transport protocol in wireless environments leads to the well documented performance problems of TCP over wireless links. Our approach is to employ multi-code CDMA (MC-CDMA) in combination with a recently developed simultaneous MAC packet transmission (SMPT) scheme [11–13] to stabilize the data link throughput by reducing losses and delay variations. We demonstrate in this paper that by using SMPT at the link layer we can mitigate the interference of TCP's slow start and ARQ mechanisms with the timely delivery of the video frames. Our focus in this paper is on the streaming of (closed-loop) rate controlled encoded video in the uplink direction in a cellular wireless system, i.e. from wireless clients to a central base station. (In a more extensive study [14] we have also considered open-loop encoded video and summarize our findings in Section 5 of this paper.) We focus on a reliable video streaming service that does not skip frames but instead suspends the playout at the receiver when the video consumption (temporarily) exceeds the video delivery. We provide extensive simulation results that demonstrate that our approach of combining SMPT at the data link layer and TCP at the transport layer supports video streaming in an efficient manner for closed-loop encoded video. In a typical wireless streaming scenario, the studied streaming approach causes playout suspensions that are typically shorter than 0.5 s and are over eight minutes spaced apart. We believe that this performance makes the studied approach attractive even for video applications that do not require lossless transmission, e.g. entertainment videos, music video clips, and news clips.

1.1. Related work on streaming video over wireless links

The problem of efficient video streaming over wireless links has attracted a great deal of attention recently. Several works, see for instance References [15–17], attempt to improve the video quality by employing adaptive video coding schemes. The basic assumptions shared by all mentioned works are that the traffic source is based on H.263 encoded video, real time services are applied, and the wireless link is modelled with a two-state Markov chain. To bound the time delay within an acceptable range for real-time video services the allowed maximum number of retransmission attempts are limited. In Reference [15] it is further assumed that CDMA is applied as the air interface technology. Other works employ hybrid error correction, see for instance Reference [18], to make the video transmission more robust. We have recently developed a prefetching protocol for video streaming, which schedules the transmissions according to a join-the-shortest-queue (JSQ) policy and the current channel conditions [19].

While the approaches pursued in this literature have made significant progress towards improving the efficiency of video streaming over wireless links, the issue of TCP friendliness has received very little attention. In fact, the proposed approaches rely largely on UDP as the transport protocol and typically have no mechanism to ensure TCP friendliness. Generally, continuous media streaming over TCP has been investigated in only a few studies [20–22] which are primarily focused on wired networks. In this paper, we investigate wireless video streaming with TCP as the transport protocol which is by default TCP friendly.

We note that this paper is a companion paper to Reference [23] where video streaming with SMPT using UDP as the transport layer protocol is investigated.

1.2. SMPT

Due to the variations on the wireless link, the throughput becomes unstable, e.g. varies over time. With *Send and Wait*, the simplest ARQ mechanism, as it is discussed in Reference [24] and illustrated in Figure 1, each erroneous MAC packet is retransmitted. The subsequent packets in the transmission queue have to wait until the corrupted packet has been transmitted successfully. In our example packet number four can be transmitted at time slot 10 for the first time. Even the subsequent packets are influenced by the retransmission process of packet number three. Due to retransmissions, the delay-jitter for a single MAC packet as well as for segments of higher protocol layers (e.g. TCP segments) increases.

The SMPT approach uses multiple channels to overcome the variations on the wireless link. One SMPT transmission mode called *Slow Healing* [13] is presented in Figure 2. After the error

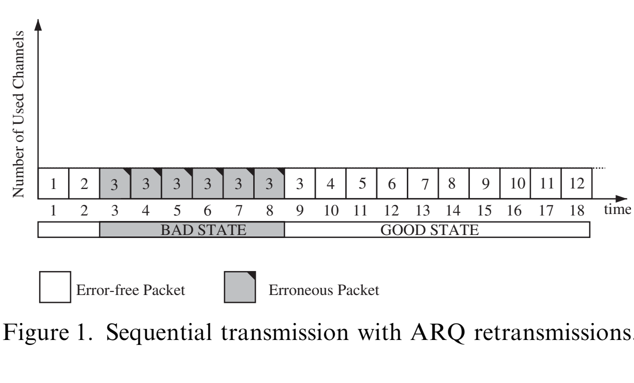

Figure 1. Sequential transmission with ARQ retransmissions.

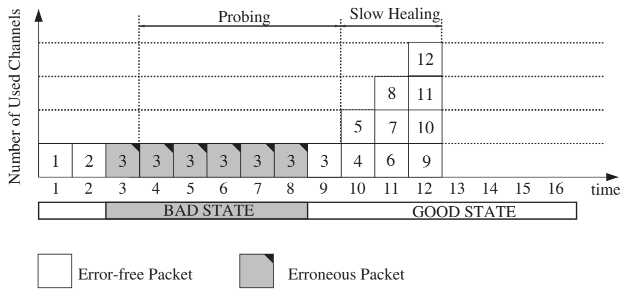

Figure 2. SMPT transmission with ARQ retransmissions.

burst, packets are transmitted via multiple channels. After each successful usage of parallel channels one more channel is added up to a maximum of R parallel channels. These parallel channels are used until the influenced jitter is healed (i.e. the backlogged MAC packets have been successfully transmitted). In comparison with Figure 1 we see that a higher protocol segment consisting of 12 MAC packets is transmitted within 18 time slots using the sequential transmission and 12 time slots using the SMPT mechanism. The sequential approach would have also used 12 time slots if no error had occurred on the wireless link. Therefore, with SMPT the variations in the wireless link errors are hidden from the higher protocol layers such as TCP.

On the other hand, we have to mention that multiple channels are used by the wireless terminal (WT) at it's own discretion. To avoid overwhelming signalling among the base station and the WTs we use pseudo-noise spreading sequences (codes). This means that by using additional channels we have an impact on the performance of all other WTs which are active in the cell. The question is how this will affect the capacity of the cell.

2. TCP OVER WIRELESS LINK

Typically, TCP is used in *wired* communication systems with very small errors probabilities. The error characteristics of wireless channels, however, differ significantly from that of wired channels. Therefore, TCP gives very poor performance if it is directly applied to a *wireless* communication system. Wired channels are characterized by minuscule packet loss probabilities and randomly spaced errors. In contrast, wireless channels are characterized by time-varying packet loss probabilities that are generally much larger than for wired channels. Also, the errors are typically bursty on wireless channels [25–29]. Moreover, wireless channels are distinct and time-varying between the WTs, that is, the wireless link errors are location-dependent. The variability of the wireless channel quality is due to the mobility of the WTs, fading effects, interference from other WTs, and shadowing. All of these effects degrade the channel performance significantly and have a significant impact on higher protocol layers. Numerous studies have found that TCP supports wireless Internet access only very inefficiently, e.g. see References [27, 28]. The key problems is that wireless channel errors lead to frequent expirations of the TCP retransmission timer, which are interpreted as congestion by TCP.

As illustrated in the previous section, SMPT stabilizes the throughput over the wireless link at the data link layer. The effect of this link layer stabilization on the TCP performance for elastic data traffic has been investigated in References [11, 13]. We now briefly review the effects of

SMPT on the TCP sequence number and congestion window from References [11, 13] as these effects have important implications for video streaming and are vital for explaining the performance for the video transmission studied in this paper. We consider the TCP performance over a wireless link for single-code CDMA (SC-CDMA) and MC-CDMA systems. For the SC-CDMA system we employ sequential transmission, while SMPT is employed in the MC-CDMA system. We consider one WT that is transmitting data to the base station using an file transfer protocol (FTP) application without multi-access interference, considering only a static channel error. The most important parameter for elastic data applications, such as FTP, is the TCP throughput.

In Figure 3, we plot the sequence number over the time for the SMPT and the sequential transmission approaches for three different packet error probabilities; the TCP throughput is defined as the time derivative of the sequence numbers. We observe that the sequential approach gives smaller TCP throughput for increasing error probabilities, whereas the SMPT approach is stable for the entire considered range of packet error probabilities from 10^{-6} to 10^{-2}. There are two reasons for the increase in TCP throughput when using SMPT. Primarily, the SMPT approach offers more bandwidth than the sequential case. The second reason is that *spurious* retransmissions of TCP segments can be avoided. These spurious retransmissions take place every time the congestion window shrinks down. We observe further that for the SMPT approach the sequence number increases steadily. For the sequential case we notice some collapses of the sequence number. This indicates that the SMPT approach has a stabilizing effect on the wireless link. SMPT stabilizes the throughput by overcoming the fast-time scale variations (typically on the order of tens of milliseconds) of the wireless channel. This is very important for video streaming since video encoders can only react to the available channel bandwidth on a longer time-scale (typically on the order of hundreds of milliseconds or seconds).

Figure 4 shows the congestion window size over time for the SMPT and sequential transmission approaches for three different packet error probabilities. In this example we observe that for all considered packet error probabilities the SMPT approach—in contrast to the sequential transmission approach—never shrinks its congestion window. With a small error

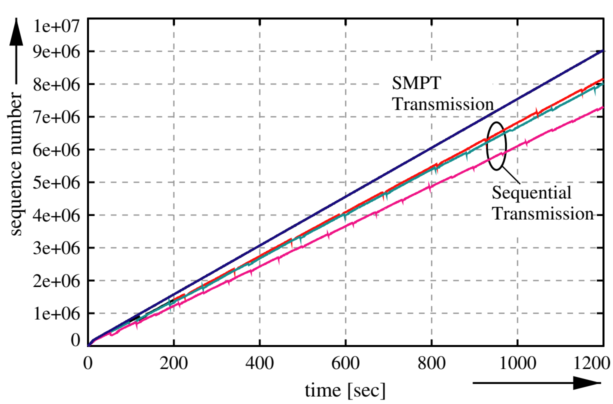

Figure 3. TCP sequence number versus time for different error probabilities for sequential and SMPT transmission of elastic data traffic.

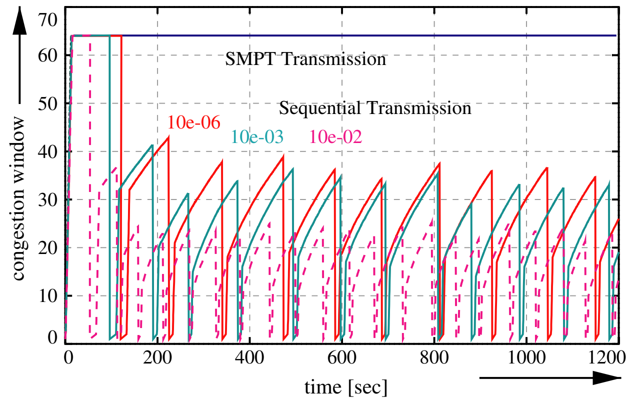

Figure 4. TCP congestion window size versus time for different packet error probabilities for sequential and SMPT transmission of elastic data traffic.

probability the congestion window is quite large for the sequential transmission. However, an increased error probability causes the congestion window to shrink more often in the case of sequential transmission. The congestion window behaviour of SMPT once more illustrates the stabilization of the throughput on the wireless link. The TCP segments are always transmitted within the pre-calculated re-transmission timeout when SMPT is employed. For this illustrative example we assumed a *non-responsive* environment, i.e. that the background noise and therefore the bit error probability are constant, irrespective of how many channels are used. Using SMPT in the uplink forces the usage of pseudo-noise spreading sequences [11–13]. Due to the correlated pseudo-noise sequences the multiple access interference increases if more channels are used within the cell. Henceforth, we consider an *all-responsive* environment, i.e. all active channels have an impact on each other. Also, the stabilizing effect of SMPT on TCP was illustrated for elastic data traffic in this section. Next, we conduct a detailed quantitative study for video streaming traffic.

3. SIMULATION MODEL AND SCENARIO

In this paper, we focus on the streaming of video over the wireless links in a single cell of a cellular wireless system. We consider the uplink streaming of rate controlled H.263 encoded video from several WTs to the base station. We note that the streaming in the uplink direction is a particular challenge, as the WTs act in an independent, unco-ordinated fashion; unlike the case of downlink streaming where the base station can co-ordinate the transmissions. The base station acts as a receiver. We note that the bandwidths of the backbone networks are typically that high, that the wireless link is the bottleneck. We can therefore without loss of generality consider the base station as receiver. At the receiver side (i.e. base station) we assume a play-out buffer. In the simulations each WT randomly selects one out of 25 video sequences, which are obtained from Reference [30] including sports, movie, and news video sequences. Also, each WT selects an independent random starting phase into the selected trace to ensure the statistical independence of the transported video streams. The WT commences the video streaming by filling the receiver-side play-out buffer to a pre-specified *offset* value $\tau_{\rm off}$ in units of seconds. The receiver side application starts to play out the video once the play-out buffer reaches the offset value. Note that the start-up latency is the time it takes to transmit $\tau_{\rm off}$ seconds worth of video over the wireless link. (The transmission of the $\tau_{\rm off}$ seconds worth of video takes typically slightly less than $\tau_{\rm off}$ seconds since in the typical streaming scenario the wireless transmission resources are allocated such that the channel provides a long run transmission rate that is slightly larger than the average bit rate of the video.) Also note that the buffer at the receiver side must be sufficiently large to hold the $\tau_{\rm off}$ seconds of video. Under normal circumstances, for every frame period (which is typically an integer multiple of 40 ms for H.263 encoded video [30]) the receiver removes a frame from the play-out buffer, decodes it, and displays it. If at any of these epochs there is no complete video frame in the play-out buffer, the receiver experiences playback starvation, which we refer to as *buffer underrun*. When a buffer underrun occurs the receiver temporarily suspends the play-out of the video. The receiver waits until the play-out buffer is filled to the offset value, and then resumes the play out of the video. (The duration of the interruption is the time it takes to transmit $\tau_{\rm off}$ seconds worth of video, which is equivalent to the start-up latency.) Note that no video frames are skipped when a buffer underrun occurs. This makes this reliable video streaming scheme well suited for applications that can tolerate short pauses in the video playback, but do not tolerate any loss of video frames, such as the video feed from a wireless surveillance camera. The duration of the buffer underrun, i.e. the suspension of the video playback depends on the bit rate of the video and the throughput of the TCP transport protocol at that particular instant.

The wireless channel is modelled as a two-state Markov chain with a *good* and a *bad* state, where no communication is possible in the *bad* state. The bit error probability in the *good* state depends on the total number of active channels. We use the *improved-Gaussian–approximation* (derived by Holtzman [31]) to calculate the bit error probability. As in Reference [15] a Bose–Chaudhuri–Hocquenghem (BCH) FEC code is used and gives the packet error probability as a function of the number of active channels. All parameters used in the simulation are summarized in Table I, for a detailed discussion of these parameters we refer the interested reader to Reference [14]. At the data link layer we study two transmission approaches: (1) sequential transmission based on a SC-CDMA system with single and doubled bit rate, and (2) SMPT which is based on a MC-CDMA system. In our performance evaluation we study primarily the buffer underrun rate Ω (in buffer underruns per second) and the average time between buffer underruns T_Ω (in seconds). For higher values of T_Ω and lower values of Ω the systems performs better. We also investigate the inter-cell and intra-cell interference.

4. SIMULATION RESULTS FOR CLOSED-LOOP ENCODED VIDEO

In this section, we discuss the system behaviour for 64 kbps rate controlled H.263 encoded video. To accommodate the overhead of the upper protocol layers and some retransmissions, we chose a bit rate of 72.8 kbps for the physical layer. (Note that the overprovisioning of the wireless channels allows even the sequential transmission scheme to perform retransmissions.) In Figure 5, we plot the buffer underrun rate as a function of the number of WTs for the SMPT approach (with at most $R = 3$ parallel CDMA channels) and the sequential transmission approaches with single and double bit rate. The offset value $\tau_{\rm off}$ is set to 0.5 s (on the left side) and 1.0 s (on the right side). For the first set of simulations the spreading gain is set to 16.

Table I. Simulation Parameters for TCP-based video streaming.

Employment	Parameter	Value
Scenario	Number of wireless terminals (WTs)	1–26
Application	Type	Video
	Encoder	H.263 rate control
	Bit rate (kbps)	64
	Peak/mean frame size	5.48
Transport layer	Segment size L_{TCP} (bytes)	1400
	TCP header (bytes)	20
Network Layer	L_{buffer} (segments)	10 000
	IP header (bytes)	20
Data link layer	Packet size L_{MAC} (bytes)	91
	Backlog limit N_{backlog}	10 000
Physical layer	Slot length $\tau_{\mathrm{frame}} = \frac{L_{\mathrm{MAC}}}{C}$ (ms)	10
	Number of available channels R	1, 3
	Bit rate C (kbps)	72.8
	Spreading gain G	8, 16, 32
	Frame size (bits)	1023
	Payload (bits)	728
FEC code	Redundant (bits)	295
	Correctable errors	30
Wireless link	Bad state duration τ_{bad} (ms)	100
	Good state duration τ_{good} (ms)	1000
	Bad state p_{err}	1.0
	Good state p_{err}	Improved Gaussian approximation
Simulation	Confidence level CV	99%
	Measure interval τ_{tic} (s)	10

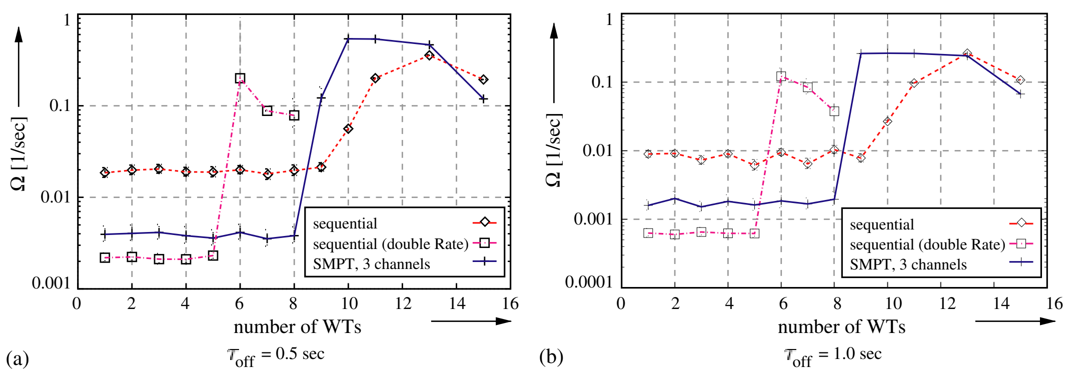

Figure 5. Buffer underrun rate Ω versus number of WTs for the SMPT approach ($R = 3$) and the sequential transmission approaches with single bit rate ($G = 16$) and double bit rate ($G = 8$).

(A spreading gain of $G = 8$ is used for the double bit rate experiments, which should be ignored for now and are discussed shortly.) Even though the offset value has a significant impact on the buffer underrun rate, the two figures reflect the same overall behaviour.

First, we discuss the differences between the sequential transmission and the SMPT approach. The buffer underrun rate for the sequential case is nearly one order of magnitude larger than for SMPT for a certain range of the number of WTs (from 1 to 9). Within this range the buffer underrun rate is almost constant. We refer to this range as *operational phase*. For more WTs, i.e. to the right of the operational phase the buffer underrun rate increases dramatically. Note, that the operational phase of the sequential case contains one more WT (i.e. 9 WTs) than the operational phase of the SMPT approach (which is 1–8 WTs), but the user has to accept lower quality for the sequential case. The dramatic increase of the buffer underrun rate for SMPT with 9–13 WTs is the result of an increased usage of CDMA channels of all WTs, which results in higher bit error probabilities. Increasing the number of WTs further leads to a small decrease in the buffer underrun rate. This is caused by TCP mechanisms, which try to adapt to the channel behaviour. Henceforth, we concentrate on the operational phase, where both approaches give stable results. For the given scenario with 0.5 s (1.0 s) offset we observe for the operational phase that the average time between buffer underruns is $T_{\Omega,\mathrm{seq}} = 50$ s ($T_{\Omega,\mathrm{seq}} = 100$ s) for the sequential and $T_{\Omega,\mathrm{SMPT}} = 250$ s ($T_{\Omega,\mathrm{SMPT}} = 500$ s) for the SMPT approach.

The reason for lower buffer underrun rates is illustrated in Figure 6 where the buffer content versus time is depicted for sequential and SMPT transmission. The buffer size is measured at one dedicated WT. The figure reflects the stabilizing effect of SMPT. Within the investigated time interval no buffer underrun takes place for SMPT. On the other hand, the buffer content of the sequential transmission mode is highly variable and two buffer underruns occur. We note that for illustration the offset value (which corresponds to the buffer capacity at the receiver) is set to 3 s in this sample path plot. (The excursions up to 3.5 s are due to the granularity of the TCP segments.) For all the following figures the offset value is set to either 0.5 or 1 s.

To demonstrate that the SMPT gain is not only due to the higher bit rates (bundling of CDMA channels), we compare the SMPT approach with a sequential transmission mode using double bit rate. To double the bit rate in the SC-CDMA system we reduce the spreading gain to $G = 8$. (We assume that the intersymbol interference is not changed for the higher bit rates, which gives optimistic results for the higher bit rate scenario.) With the higher bit rate, we can send two packets within one time slot. For each packet we use the same coding approach as

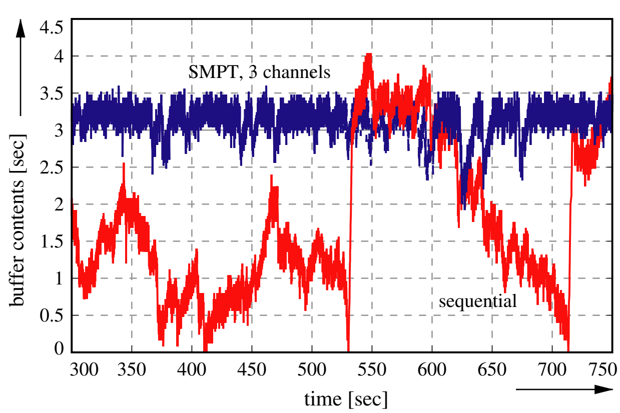

Figure 6. Buffer contents for sequential and SMPT transmission for $\tau_{\mathrm{off}} = 3$ s.

before. We observe from Figure 5 that for this transmission mode with doubled bit rate, the average time between buffer underruns is $T_{\Omega,\mathrm{seq}^2} = 500$ s for $\tau_{\mathrm{off}} = 0.5$ s; thus the improvement in the buffer underrun rate is slightly higher than for SMPT. However, the operational phase (capacity of the cell) is much smaller than for the SMPT approach (five video streams for doubled bit rate versus eight video streams with SMPT).

To decrease the buffer underrun rate further we could increase the offset value τ_{off}. The impact of the offset value τ_{off} is given in Figure 7. We note, that higher offset values require a larger buffer and introduce a larger time shift between play-out time and reality. We observe that for the entire range of studied offset values τ_{off}, the buffer underrun rate of SMPT is roughly one order of magnitude smaller than that of the sequential transmission mode.

Next, we investigate the capacity (i.e. the maximum number of supported WTs in the operational phase) and the impact of the spreading gain. With a spreading gain of 32 (instead of 16, as used before) the buffer-underrun rate versus the number of WTs is given for two different offset values in Figure 8. On the left side the offset value is set to $\tau_{\mathrm{off}} = 0.5$ s and on the right side to $\tau_{\mathrm{off}} = 1.0$ s. In this higher spreading gain scenario, SMPT achieves both a longer mean time between buffer underruns ($T_{\Omega,\mathrm{SMPT}} = 500$ s for $\tau_{\mathrm{off}} = 0.5$ s, compared to 250 s for a spreading gain of $G = 16$) and a higher capacity (21 video streams with SMPT versus 19 with sequential mode, noting that the sequential mode has a buffer underrun rate that is one order of magnitude larger). The reason for this behaviour is a larger multiplexing effect on the wireless link. With a spreading gain of $G = 16$ the bit error probability (obtained from the Holtzman approximation [31]) changes from 10^{-3} to 10^{-1} in the range of three active channels, while for $G = 32$ this range increases to five WTs. With this larger range, a higher multiplexing gain can be achieved.

An effect that was not taken into consideration in our simulations so far is the inter-cell and intra-cell interference. The inter-cell interference is a measure for the energy that the cell *exports* to the neighbouring cells. We plot the mean number of used channels within a CDMA cell versus the number of WTs for the SMPT approach (with $R = 3$) and the sequential transmission approaches with single and double bit rate for an offset of $\tau_{\mathrm{off}} = 0.5$ s in Figure 9. We observe

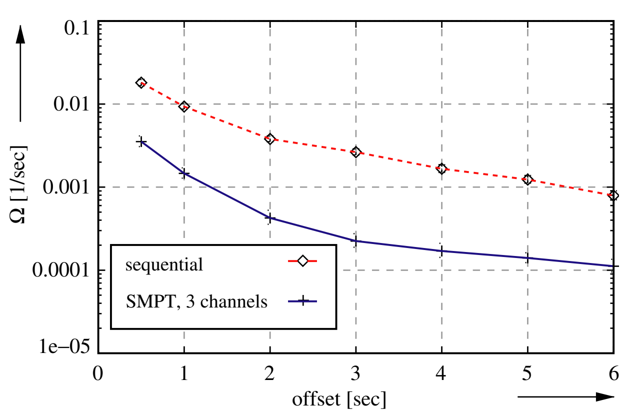

Figure 7. Buffer underrun rate Ω for sequential and SMPT transmission versus offset τ_{off}, number of WTs = 8.

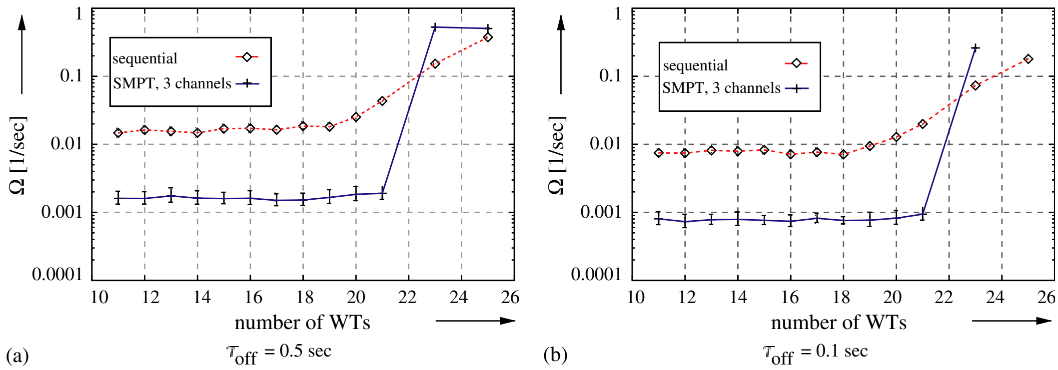

Figure 8. Buffer underrun rate Ω versus number of WTs for the SMPT approach ($R = 3$) and the sequential transmission approach ($G = 32$).

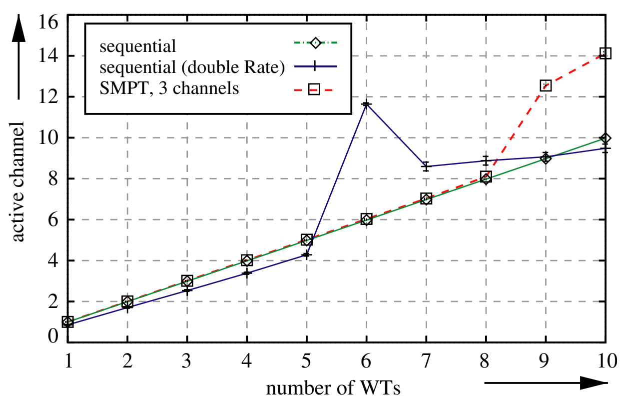

Figure 9. Mean number of used channels within a CDMA cell versus number of WTs for the SMPT approach ($R = 3$) and the sequential transmission approaches with single ($G = 16$) and double bit rate ($G = 8$) for an offset of $\tau_{off} = 0.5$ s.

that the sequential transmission mode with single bit rate and SMPT produce approximately the same amount of interference within the operational phase. Only the sequential transmission mode with double bit rate gives smaller inter-cell interference within its operational phase. This may be explained by the fact that transmitting with a higher data rate is helpful if all data can be send before the next bad channel state starts.

The intra-cell interference, on the other hand, reflects the variability of the number of used channels. The higher the variability the better has to be the power control entity. We depict the standard deviation of the number of used channels in a CDMA cell versus the number of WTs for the SMPT approach ($R = 3$) and the sequential transmission approaches with single and double bit rate for an offset of $\tau_{off} = 0.5$ s in Figure 10. We observe that the values of the standard deviation for SMPT are one order of magnitude larger than for the sequential transmission mode with single bit rate and one magnitude smaller than for the sequential transmission approach with double bit rate.

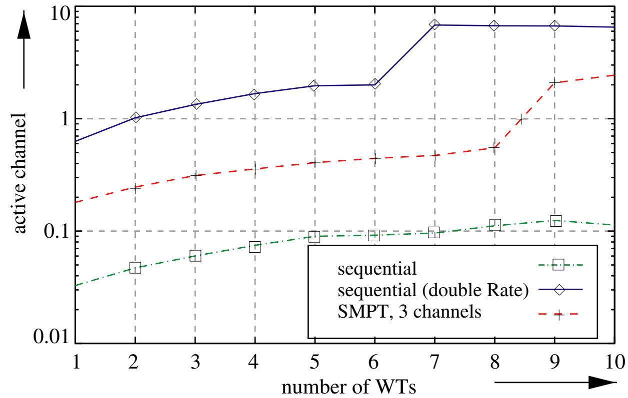

Figure 10. Standard deviation of the number of used channels within a CDMA cell versus number of WTs for the SMPT approach ($R = 3$) and the sequential transmission approaches with single ($G = 16$) and double bit rate ($G = 8$) for an offset of $\tau_{\text{off}} = 0.5$ s

5. SUMMARY OF SIMULATION RESULTS FOR OPEN-LOOP ENCODED VIDEO

All numerical experiments described in the preceding section were conducted for rate controlled encoded video. With rate control, the quantization scale employed for the quantization of the transform coding coefficients in the video coding is varied to achieve a desired target bit rate (in the long run average) and to keep the variations of the bit rate relatively small. These smaller variations of the video traffic come at the expense of variations in the video quality due to the varying coarseness of the quantization of the coding coefficients. For video coding without rate control (i.e. open-loop video coding), on the other hand, the quantization scale is kept fixed, resulting in close to constant video quality at the expense of highly variable video traffic. For rate controlled video the peak-to-mean frame size ratio is typically less than 4–5, while for open-loop encoded video the peak-to-mean ratio of the frame sizes may be up to 14 or even higher. Our numerical experiments reported in the preceding section indicate that the streaming of rate controlled encoded video over TCP gives good performance when the stabilizing SMPT scheme is employed. We have also conducted extensive numerical experiments (not given here because of space constraints, see Reference [14] for details) for the streaming of *open-loop* encoded video. We have found that the streaming of open-loop encoded video over TCP in wireless systems gives generally poor performance, both without and even with SMPT. The buffer underrun rates are typically on the order of 0.03–0.05 for an offset of $\tau_{\text{off}} = 1$ s for open-loop encoded video, compared to roughly 0.001 for closed-loop encoded video, see Figure 7. We attribute this result to the limited ability of TCP to accommodate the highly variable open-loop encoded video streams.

6. CONCLUSIONS

We have studied the streaming of video using TCP as the transport protocol in cellular CDMA-based wireless systems. TCP has a number of desirable properties, such as network stability and fair bandwidth allocation, for the future Internet (both wired and wireless). Video streaming over TCP, however, is generally known to give poor performance, especially in wireless environments. We have examined a streaming scheme that uses TCP to stream video in a MC-CDMA system. Our scheme employs SMPT to effectively stabilize the wireless links. We have found that the stabilizing effect of SMPT at the link layer significantly improves the performance of TCP video streaming over the wireless links. We found that our approach gives good results for closed-loop *rate controlled* encoded video, especially with higher spreading gains. We note that the investigated streaming scheme preserves the isolation of the protocol layers. Specifically, SMPT does not require knowledge of any TCP parameters. Independently of the TCP operation, SMPT stabilizes the throughput over the wireless link and thus significantly reduces the probability that the TCP round trip time out is exceeded.

ACKNOWLEDGEMENTS

The work of M. Reisslein is supported in part by the National Science Foundation under Grant No. Career ANI-0133252 and Grant No. ANI-0136774.

REFERENCES

1. Bose D. M-comerce for 3G. Presentation, 11th Time-Market, Sony-Center Berlin, Germany, September 2001.
2. Roberts L, Tarsala M. Inktomi goes wireless; forms alliances. In *CBS MarketWatch*, 14 March 2000.
3. Hong D, Albuquerque C, Oliveira C, Suda T. Evaluating the impact of emerging streaming media applications on TCP/IP performance. *IEEE Communications Magazine* 2001; **39**(4):76–82.
4. Hassan M, Jain R. TCP performance in future networking environments. *IEEE Communications Magazine* 2001; **39**(4):51. (*Guest editorial*).
5. Albuquerque C, Vickers BJ, Suda T. Network border patrol. In *Proceedings of IEEE Infocom*, Tel Aviv, Israel, March 2000.
6. Jacobs S, Eleftheriadis A. Streaming video using dynamic rate shaping and TCP congestion control. *Journal of Visual Communication and Image Representation* 1998; **9**(3):211–222.
7. Kim J, Kim Y-G, Song H, Kuo T-Y, Chung YJ, Kuo C-CJ. TCP-friendly internet video streaming employing variable frame-rate encoding and interpolation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2000; **10**(7):1164–1177.
8. Widmer J, Denda R, Mauve M. A survey on TCP-friendly congestion control. *IEEE Network* 2001; **15**(3):28–37.
9. Bansal D, Balakrishnan H. Binomial congestion control algorithms. In *Proceedings of IEEE Infocom*, Anchorage, Alaska, 2001; 631–640.
10. Loguinov D, Radha H. On retransmission schemes for real-time streaming in the internet. In *Proceedings of IEEE Infocom*, Anchorage, Alaska, 2001; 1310–1319.
11. Fitzek F, Rathke B, Schläger M, Wolisz A. Simultaneous MAC-packet transmission in integrated broadband mobile system for TCP. In *Proceedings of ACTS Summit 1998*, June 1998; 580–586.
12. Fitzek F, Rathke B, Schläger M, Wolisz A. Quality of service support for real-time multimedia applications over wireless links using the simultaneous MAC-packet transmission (SMPT) in a CDMA environment. In *Proceedings of MoMuC 1998*, October 1998; 367–378.
13. Fitzek F, Morich R, Wolisz A. Comparison of multi-code link-layer transmission strategies in 3Gwireless CDMA. *IEEE Communication Magazine* 2000; **38**(10):58–64. (Technologies on broadband wireless mobile: 3Gwireless and beyond).
14. Fitzek F. Simultaneous MAC packet transmission for QoS support in wireless CDMA systems. *Ph.D. Dissertation*, Technical University Berlin, Germany, June 2002, available at http://www.fitzek.net and http://www.eas.asu.edu/˜mre.
15. Tosun AS, Feng W-C. On improving quality of video for H.263 over wireless CDMA networks. In *Proceedings of IEEE Wireless Communications and Networking Conference* (*WCNC*), Chicago, USA, September 2000.
16. Kumwilaisak W, Kim JW, Kuo C-CJ. Reliable wireless video transmission via fading channel estimation and adaptation. In *Proceedings of IEEE Wireless Communications and Networking Conference* (*WCNC*), Chicago, USA, September 2000.

17. Qiao D, Shin KG. A two-step adaptive error recovery scheme for video transmission over wireless networks. In *Proceedings of IEEE Infocom*, Tel Aviv, Israel, March 2000; 1698–1704.
18. Liu H, Zarki M. Performance of H.263 video transmission over wireless channels using hybrid ARQ. *IEEE Journal on Selected Areas in Communications* 1997; **15**(9):1775–1786.
19. Fitzek F, Reisslein M. A prefetching protocol for continuous media streaming in wireless environments. *IEEE Journal on Selected Areas in Communications* 2001; **19**(6):2015–2028.
20. Krasic C, Li K, Walpole J. The case for streaming multimedia with TCP. In *Proceedings of 8th International Workshop on Interactive Distributed Multimedia Systems* (*IDMS*), Lancaster, U.K. Lecture Notes in Computer Science, vol. 2158. Springer: Berlin, 2001; 213–218.
21. de Cuetos P, Guillotel P, Ross K, Thoreau D. Implementation of adaptive streaming of stored MPEG-4 FGS video over TCP. In *Proceedings of IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, 2002; 405–408.
22. Noda Y, Sakai T, Shigeno H, Okada K, Matsushita Y. A survey of transport layer protocols suited for real-time data delivery over diffserv-capable networks. In *Proceedings of International Conference on Distributed Computing Systems*, Vienna, Austria, 2002; 305–310.
23. Fitzek F, Reisslein M, Wolisz A. Uncoordinated real-time video transmission in wireless multicode CDMA systems: an SMPT-based approach. *IEEE Wireless Communications* 2002; **9**(5):100–110.
24. Bertsekas G. *Data Networks*, vol. 2. Prentice-Hall: Englewood Cliffs, NJ, 1992.
25. Bao G. Performance evaluation of TCP/RLP protocol stack over CDMA wireless link. *Wireless Networks* 1996; **2**(3):229–237.
26. DeSimone A, Chuah MC, Yue O. Throughput performance of transport-layer protocols over wireless LANs. In *Proceedings of IEEE Globecom 93*, Houston, USA, 1993.
27. Balakrishnan H, Padmanabhan V, Seshan S, Katz R. A comparison of mechanisms for improving TCP performance over wireless links. *IEEE/ACM Transactions on Networking* 1997; **5**(6):756–769.
28. Balakrishnan H, Seshan S, Amir RKE. Improving TCP/IP performance over wireless networks. In *Proceedings of ACM Mobicom*, Berkeley, USA, November 1995.
29. Zorzi M, Rao RR. The effect of correlated errors on the performance of TCP. *IEEE Communications Letters* 1997; **1**(5):127–129.
30. Fitzek F, Reisslein M. MPEG–4 and H.263 video traces for network performance evaluation. *IEEE Network* 2001; **15**(6):40–45. (video traces available at http://trace.eas.asu.edu).
31. Holtzman J. A simple, accurate method to calculate spread spectrum multiple access error probabilities. *IEEE Transactions on Communications* 1992; **40**(3):461–464.

AUTHORS' BIOGRAPHIES

Frank H. P. Fitzek is an Associate Professor in the Department of Communication Technology, University of Aalborg, Denmark, heading the Future Vision group. He received his diploma (Dipl-Ing) degree in electrical engineering from the University of Technology—Rheinisch-Westfälische Technische Hochschule (RWTH)—Aachen, Germany, in 1997 and his PhD (Dr-Ing) in Electrical Engineering from the Technical University Berlin, Germany, in 2002. As a visiting student at Arizona State University, Tempe, he conducted research in the field of video services over wireless networks. He co-founded the start-up company acticom GmbH in Berlin in 1999. In 2002, he was Adjunct Professor at the University of Ferrara, Italy, giving lectures on wireless communications and conducting research on multi-hop networks. His current research interests are in the areas of QoS support for multimedia services over wireless networks, access techniques, security for wireless communication, and the integration of multi-hop networks in cellular systems. Dr Fitzek serves on the Editorial Board of the IEEE Communications Surveys & Tutorials. He is the program chair for the International Conference on Advances in Computer Entertainment Technology (ACE2004) and serves on the program committees for VTC2003, VTC2004, ACE2004, and IEEE MWN2004.

Martin Reisslein is an Assistant Professor in the Department of Electrical Engineering at Arizona State University, Tempe. He received the Dipl-Ing (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the MSE degree from the University of Pennsylvania, Philadelphia, in 1996, both in electrical engineering. He received his PhD in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994–1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 to October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin. While in Berlin he was teaching courses on performance evaluation and computer networking at the Technical University Berlin. He is editor-in-chief of the *IEEE Communications Surveys and Tutorials* and has served on the Technical Program Committees of *IEEE Infocom*, *IEEE Globecom*, and the *IEEE International Symposium on Computer and Communications*. He has organized sessions at the *IEEE Computer Communications Workshop* (*CCW*). He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG-4 and H.263 encoded video, at `http://trace.eas.asu.edu`. He is the co-recipient of the Best Paper Award of the *SPIE Photonics East 2000—Terabit Optical Networking* conference. His research interests are in the areas of Internet Quality of Service, video traffic characterization, wireless networking, and optical networking.

