Capacity and QoS for Streaming Video Applications over TCP in CDMA based Networks

F. Fitzek R. Supatrio A. Wolisz
Technical University Berlin
Telecommunication Network Groups
Technical University Berlin
[fitzek|supatrio|wolisz]@ee.tu-berlin.de

M. Reisslein
Arizona State University
Department of Electrical Engineering
Arizona State University
reisslein@asu.edu

We study the capacity, i.e., the number of customers per cell, and the quality of service for streaming video in the uplink direction of a cellular **C**ode **D**ivision **M**ultiple **A**ccess (CDMA)–based wireless system. We advocate the use of TCP as the transport layer protocol for streaming video in a **M**ulti–**C**ode CDMA (*MC–CDMA*) wireless system. Our approach is to stabilize the TCP–throughput for video transmissions over the wireless links by employing a recently developed **S**imultaneous **M**AC **P**acket **T**ransmission (*SMPT*) approach. Our extensive simulations indicate that employing SMPT significantly improves both the video quality and the capacity when streaming video over TCP.

Keywords: Multi–Code CDMA, 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 [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 **U**ser **D**atagram **P**rotocol (*UDP*) as the transport protocol. As streamed video applications become more popular, the Internet will 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 **T**ransport **C**ontrol **P**rotocol (*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: 1.) congestion collapse and 2.) unfair allocation of bandwidth among competing traffic flows [5]. Therefore, TCP friendly video streaming schemes are desirable. Jacobs and Eleftheriadis [6] introduced a TCP friendly approach for streamed video. In contrast, in this paper we advocate to use directly 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 surveillance video. 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 **A**utomatic **R**epeat Re**Q**uest (*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 well known, 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 **M**ulti–**C**ode CDMA (*MC–CDMA*) in combination with a recently developed **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) scheme [7, 8, 9] 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 rate controlled encoded video in the uplink direction in a cellular wireless system, i.e., from wireless clients to a central base station. 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.

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 [10, 11, 12] 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 modeled 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 [10] it is further assumed that CDMA is applied as the air interface technology. Other works employ hybrid error correction, see for instance [13], 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 [14]. 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. In this paper we propose a video streaming scheme that uses TCP as the transport protocol and is therefore by default TCP friendly. We also demonstrate that our scheme has favorable performance characteristics.

Simultaneous MAC Packet Transmission

Due to the variation 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 [15] 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 (e.g. TCP segment) of higher protocol layers increases.

The SMPT approach uses multiple channels to overcome the variations on the wireless link. One SMPT transmission mode called *Slow Healing* [9] is presented in Figure 2. After the error burst packets are transmitted via multiple channels. After each successful usage of parallel channels one more channel is added until the influenced jitter is healed. In comparison with Figure 1 we see that a higher protocol segment consisting out of 12 MAC packets is transmitted within 18 time slot using the sequential transmission and 12 time slots using the SMPT mechanism. The sequential approach would have used also 12 time slots if no error would have been occurred on the wireless link. Therefore for the higher protocol layers such as TCP the variation on the wireless link is not noticeable anymore if SMPT is applied.

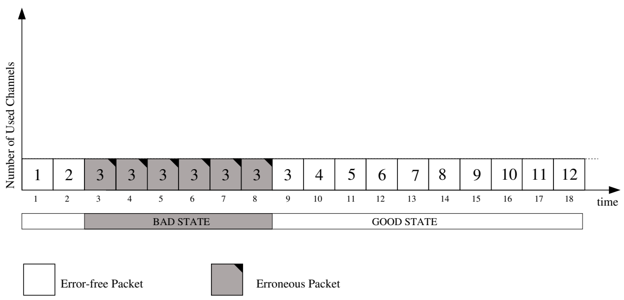


Figure 1: Sequential transmission mode with ARQ retransmissions in the time domain.

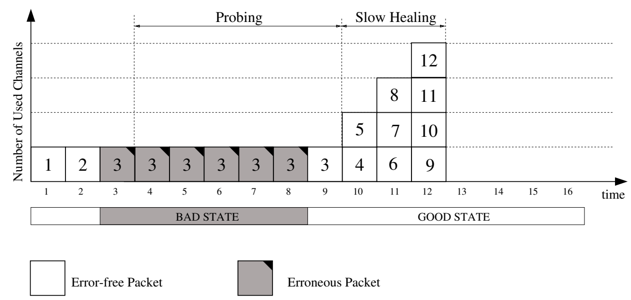


Figure 2: SMPT transmission mode with ARQ retransmissions in the code domain.

On the other hand we have to mention that multiple channels are used by the **W**ireless **T**erminal (WT) at it's own discretion. To avoid overwhelming signaling among the **B**ase **S**tation (BS) and the **W**ireless **T**erminals (WTs) we use **P**seudo–**N**oise (PN) sequences. This means that by using additional channels we have an impact on the performance of all other WT which are active in the cell. The question is how this will influence the capacity of the cell.

2 TCP over wireless

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 channel are characterized by miniscule 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 [16, 17, 18, 19, 20, 21]. Moreover, wireless channels are distinct and time–varying between the **W**ireless **T**erminals (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 [17, 19, 20]. The key problems is that wireless channel errors lead to frequent expirations of the TCP retransmission timer, which are interpreted as congestion by TCP. In [7, 9] a SMPT scheme for stabilizing the data link layer throughput over the wireless channel was introduced and studied for elastic data traffic. The SMPT approach is based on MC–CDMA. We now give a brief discussion of the impact of wireless link error on the TCP performance for elastic traffic. We show the performance gain that can be achieved with SMPT as compared to a sequential transmission, we refer the interested reader to [7, 9] for a detailed study. For illustration we consider the TCP performance over a wireless link for **S**ingle–**C**ode CDMA (*SC–CDMA*) and MC–CDMA systems. For the SC–CDMA system we employ sequential transmission, while SMPT is employed in the **M**ulti–**C**ode CDMA (*MC–CDMA*) system. We consider one WT that is transmitting data to the base station using a **F**ile **T**ransfer **P**rotocol (*FTP*) application (note that in order to show the impact on TCP, FTP is the appropriate traffic source) without **M**ultiple **A**ccess **I**nterference (*MAI*) interference, considering only a static channel error. The most important parameter for FTP applications is the TCP throughput. In Figure 3 we plot the sequence number over the time [1] for the SMPT and the sequential transmission approaches for three different **P**acket **E**rror **P**robability (*PEP*). While the sequential approach gives smaller TCP throughput as the PEP increases, the SMPT approach is stable over a wide range of the PEP (10^{-6} up 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* retransmission of TCP segments can be avoided. These spurious retransmission takes place every time the **C**ongestion **W**indow (*CW*) shrinks down. We observe further that for the SMPT approach the sequence number increases steadily. For the sequential case we notice some collapse 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 important 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).

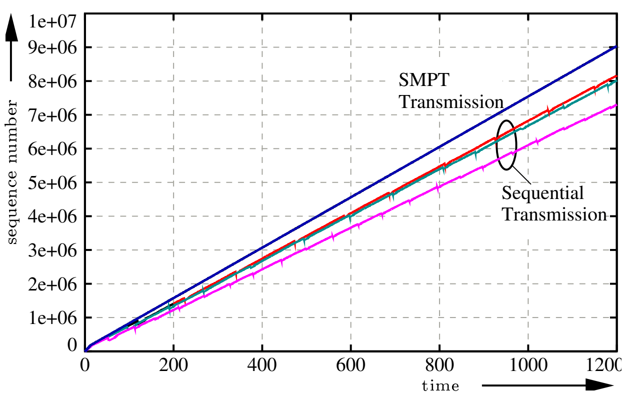


Figure 3: TCP sequence number versus time for different error probabilities

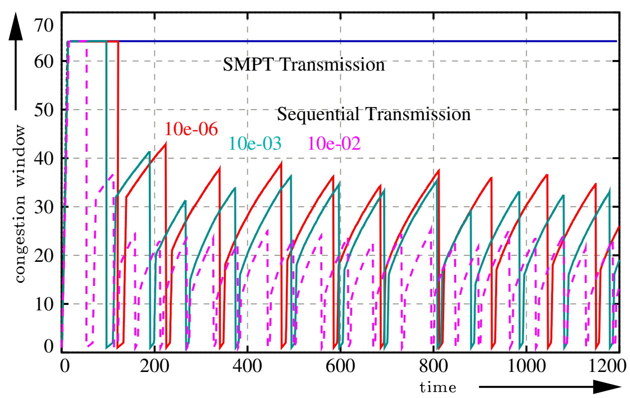


Figure 4: TCP congestion window size versus time for different packet error probabilities

Figure 4 shows the CW size over time for the SMPT and sequential transmission approaches for three different packet error probabilities. In this example we observe that the SMPT approach — in contrast to the sequential transmission approach — never shrinks its CW. With a small PEP the congestion window is quite large for the sequential transmission. However, an increased PEP causes

[1]The TCP throughput is defined as the time derivative of the sequence numbers.

the CW to shrink more often in the case of sequential transmission. The CW behavior of SMPT once more illustrates the stabilization of the throughput on the wireless link. The TCP segments are always transmitted within the pre–calculated **R**etransmission **T**ime **O**ut (RTO) when SMPT is employed. For this illustrative example we assumed a *non–responsive* environment, *i.e.* that the background noise and therefore the BEP are constant, irrespective of how many channels are used. Using SMPT in the uplink forces the usage of **P**seudo–**N**oise (PN) spreading sequences [7, 8, 9]. Due to the PN sequences the MAI increases if more channels are used within the cell. Henceforth, we assume an *all–responsive* environment, *i.e.* all active channels have an impact on each other.

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 J 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, uncoordinated fashion; unlike the case of downlink streaming where the base station can coordinate the transmissions.) The base station acts as a receiver[2]. 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 [22] including sport, 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 τ_{off} (in units of time). The receiver side application starts to play out the video once the play–out buffer reaches the offset value. Under normal circumstances, for every frame period (which is typically an integer multiple of 40 msec for H.263 encoded video [22]) 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 a *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 average *interruption time* as well as the *buffer underrun rate* Ω depend on the *offset* value. While the interruption time can be approximately calculated with

$$InterruptionTime = \frac{offset \cdot MeanFrameLength}{PhyLinkRate \cdot FrameSpacing}, \tag{1}$$

the buffer underrun rate Ω has to be evaluated by simulations. 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 modeled as a two state Markov chain with a *good* and a *bad* state, where no communication is possible in the *bad* state. The **B**it **E**rror **P**robability (BEP) in the *good* state depends on the total number of active channels. We use the *Improved–Gaussian–Approximation*

[2]As long as the wireless hop is the critical path we assume that the video streams are consumed within the base station. We note that the bandwidth of the back bones are normally that high, that all videos can be delivered to other terminals within the wired network in a timely fashion.

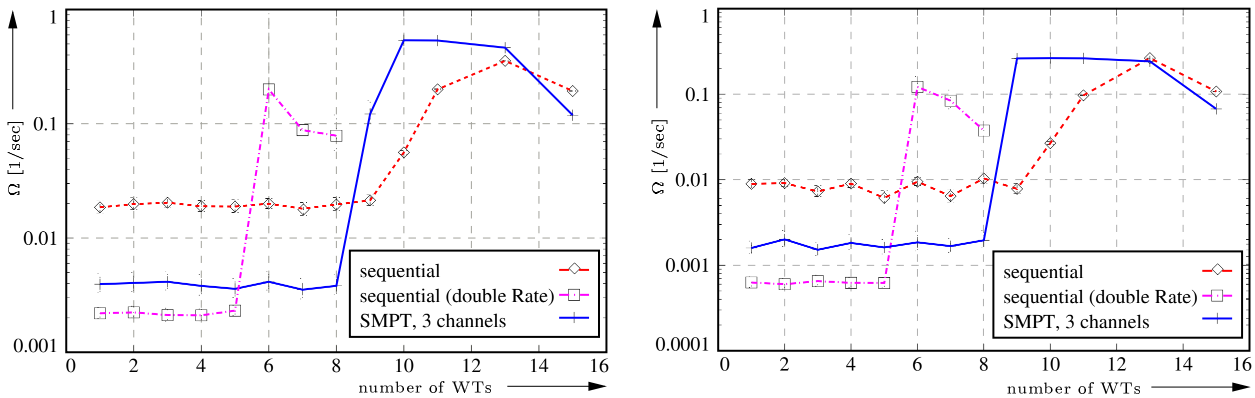
(derived by Holtzman [23] and applied in [24]) to calculate the BEP. As in [10] a **B**ose–**C**haudhuri–**H**ocquenghem (*BCH*) FEC code is used and gives the **P**acket **E**rror **P**robability (*PEP*) as a function of the number of active channels. All parameters used in the simulation are summarized in Table 1. 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.

Table 1: Simulation Parameters for TCP-based video streaming

Employment	Parameter	Value
Scenario	number of Wireless Terminals J	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]	10000
	IP header [bytes]	20
Data Link Layer	Packet size L_{MAC} [bytes]	91
	Backlog limit N_{backlog}	10000
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	16, 32
	Frame size [bits]	1023
FEC code	Payload [bits]	728
	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

4 Performance Evaluation

In this section we discuss the system behavior 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 4 we plot the buffer underrun rate as a function of the number of WTs for the SMPT approach ($R_{max} = 3$) and the sequential transmission approaches with single and double bit rate. The offset value τ_{off} is set to 0.5 sec (on the left side) and 1.0 sec (on the right side). For the first set of simulations the spreading gain is set to 16. Even though the offset value has a significant impact on the buffer underrun rate, the two figures reflect the same overall behavior.



Offset 0.5 sec

Offset 1.0 sec

Figure 5: Buffer underrun rate Ω versus number of WTs for the SMPT approach ($R_{max} = 3$) and the sequential transmission approaches with single and double bit rate ($G_{Spreading} = 16$)

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 to 8 WTs), but the user has to accept lower quality for the sequential case. The dramatic increase of the buffer underrun rate is the result of an increased usage of CDMA channels of all WTs, which results in higher BEPs. 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 behavior. Henceforth, we concentrate on the operational phase, where both approaches give acceptable results. For the given scenario with 0.5 sec (1.0 sec) offset we observe for the operational phase that the average time between buffer underruns is $T_{\Omega,seq} = 50$ sec ($T_{\Omega,seq} = 100$ sec) for the sequential and $T_{\Omega,SMPT} = 250$ sec ($T_{\Omega,SMPT} = 500$ sec) 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 the sequential and the SMPT transmission mode. 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 sec in this sample path plot. (The excursions up to 3.5 sec are due to the granularity of the TCP segments.) For all the following figures the offset value is set to either 0.5 sec or 1 sec.

To show that the SMPT gain is not only due to the higher bit rates (bundling of CDMA

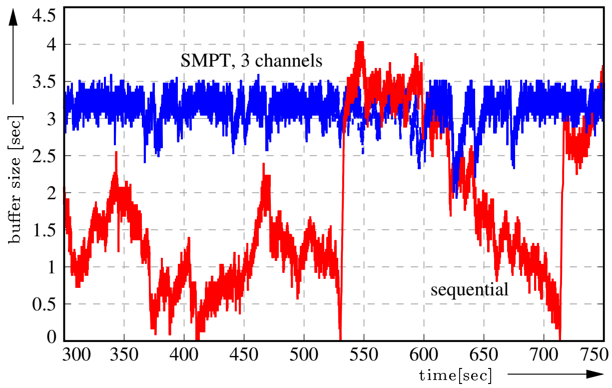


Figure 6: Buffer size for sequential and SMPT transmission approach over 450ms

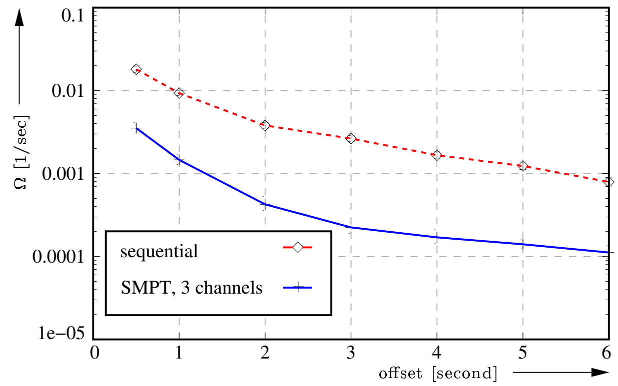


Figure 7: Buffer underrun rate Ω for the sequential and the SMPT transmission approach versus offset values of 0.5 sec up to 6 sec

channels), we compare the SMPT approach with a sequential transmission mode using double bit rate[3]. The doubled bit rate is achieved by reducing the spreading gain G. Note, that this is still a SC–CDMA system. With the higher bit rate, we can send even two packets within one time slot. For each packet we use the same coding approach as before. We observe from Figure 4 that for this transmission mode with doubled bit rate, the average time between buffer underruns is $T_{\Omega,seq^2} = 500$ sec; 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 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 0.5 sec and on the right side to 1.0 sec. In this higher spreading gain scenario, SMPT achieves both a lower buffer–underrun rate 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 behavior is a larger multiplexing effect on the wireless link. With a spreading gain of $G = 16$ the BEP changes from 10^{-3} to 10^{-1} in the range of three active channels, while for $G = 32$ the range increases to five WTs. With this larger range, a higher multiplexing gain can be achieved.

[3]Even if we consider higher bit rates, we assume that the **I**nter **S**ymbol **I**nterference (*ISI*) will not change. This will lead to better results for the higher bit rate scenario.

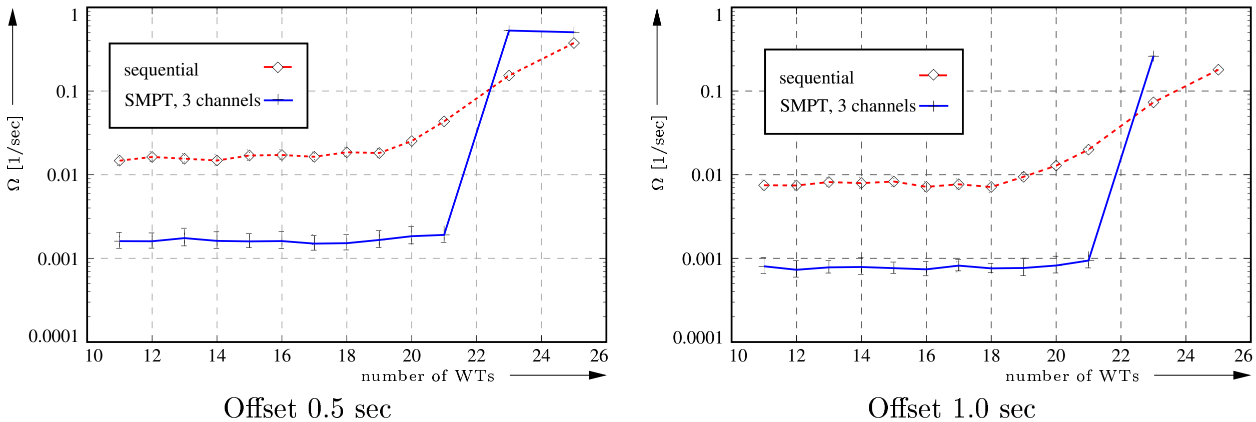


Figure 8: Buffer underrun rate Ω versus number of WTs for the SMPT approach (R_{max}=3) and the sequential transmission approach ($G_{Spreading} = 32$)

5 Conclusion and Outlook

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 proposed a scheme that uses TCP to stream video in a MC–CDMA system. Our scheme employs **S**imultaneous **MAC** **P**acket **T**ransmission (*SMPT*) to effectively stabilize the wireless links. This stabilizing effect significantly improves the performance of TCP over the wireless links. We found that our approach gives good results for rate controlled encoded video. For a given buffer underrun requirement (which corresponds to a pre–specified client playback starvation probability), our SMPT approach increases the number of supported streams in a cell by 60% in typical CDMA scenarios. We also observed that SMPT performs better with higher spreading gains. We note that our 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. On a final note, all numerical experiments in this paper were conducted for rate controlled encoded video, which is typically less variable than video encoded without any rate control (i.e., open–loop encoded video). The peak–to–mean frame size ratio is typically less than 4 – 5 for rate controlled video, while the peak–to–mean ratio of the frame sizes may be up to 14 or even higher for open–loop encoded video. Our numerical experiments reported in this paper 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) 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. We attribute this result to the limited ability of TCP to accommodate the highly variable open-loop encoded video streams.

References

[1] Dr. Bose. M–comerce for 3G. talk, September 2001. 11^{th} Time-Market, Sony–Center Berlin.

[2] L. Roberts and M. Tarsala. Inktomi goes wireless; forms alliances. In *CBS MarketWatch*, March 14th 2000.

[3] C. Oliveira D. Hong, C. Albuquerque and T. Suda. Evaluating the Impact of Emerging Streaming Media Applications on TCP/IP Performance. *IEEE Communications Magazine*, 39(4):76–82, April 2001.

[4] M. Hassan and R. Jain. TCP Performance in Future Networking Environments. *Guest Editorial, IEEE Communications Magazine*, 39(4):51, April 2001.

[5] C. Albuquerque, B. J. Vickers, and T. Suda. Network Border Patrol. In *IEEE INFOCOM 2000*, March 2000.

[6] S. Jacobs and A. Eleftheriadis. Streaming video using dynamic rate shaping and TCP congestion control. *Journal of Visual Communication and Image Representation*, 9(3):211–222, 1998.

[7] F. Fitzek, B. Rathke, M. Schläger, and A. Wolisz. Simultaneous MAC-Packet Transmission in Integrated Broadband Mobile System for TCP. In *ACTS SUMMIT 1998*, pages 580–586. ACTS, June 1998.

[8] F. Fitzek, B. Rathke, M. Schläger, and A. Wolisz. Quality of Service Support for Real-Time Multimedia Apllications over Wireless Links using the Simultaneous MAC-Packet Transmission (SMPT) in a CDMA Environment. In *Proc. MoMuC 1998*, pages 367–378, October 1998.

[9] F. Fitzek, R. Morich, and A. Wolisz. Comparison of Multi-Code Link-Layer Transmission Strategies in 3Gwireless CDMA. *IEEE Communication Magazine*, pages 58–64, oct 2000. Technologies on Broadband Wireless Mobile: 3Gwireless and Beyond.

[10] A. S. Tosun and W.-C. Feng. On Improving Quality of Video for H.263 over Wireless CDMA Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 2000.

[11] W. Kumwilaisak, J. W. Kim, and C.-C. J. Kuo. Reliable Wireless Video Transmission via Fading Channel Estimation and Adaptation. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, September 2000.

[12] D. Qiao and Kang G. Shin. A Two–Step Adaptive Error Recovery Scheme for Video Transmission over Wireless Networks. In *INFOCOM*, pages 1698–1704, Tel Aviv, Israel, March 2000.

[13] H. Liu and M. Zarki. Performance of H.263 video transmission over wireless channels using hybrid ARQ. *IEEE JSAC*, 15(9):1775–1786, December 1997.

[14] F. Fitzek and M. Reisslein. A Prefetching Protocol for Continuous Media Streaming in Wireless Environments. *JSAC – Mobility and Resource Management in Next Generation Wireless Systems*, 19(6):2015–2028, October 2001.

[15] G. Bertsekas. *Data Networks*, volume 2. Prentice Hall, 1992.

[16] G. Bao. Performance evaluation of TCP/RLP protocol stack over CDMA wireless link. *Wireless Networks*, pages 229–237, 1996.

[17] B. Rathke, M. Schlaeger, and A. Wolisz. Systematic Measurement of TCP Performance over Wireless LANs. Technical report, TKN, December 1998. TKN-01BR98.

[18] A. DeSimone, M. C. Chuah, and O. Yue. Throughput Performancce of Transport-Layer Protocols over Wireless LANs. In *Proceedings IEEE GLOBECOM 93*, 1993.

[19] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R.H. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE ACM Transactions on Networking*, December 1997.

[20] H. Balakrishnan, S. Seshan, and R.H. Katz E. Amir. Improving TCP/IP Performance over Wireless Networks. In *Proceedings of Mobicom*, November 1995.

[21] M. Zorzi and R. R. Rao. The effect of correlated errors on the performance of TCP. *IEEE Communications Letters*, 1(September):127–129, 1997.

[22] F. Fitzek and M. Reisslein. MPEG–4 and H.263 Video Traces for Network Performance Evaluation. *IEEE Network*, 15(6):40–45, November/December 2001. video traces available at `http://www-tkn.ee.tu-berlin.de/research/trace/trace.html` or `http://www.eas.asu.edu/trace`.

[23] J. Holtzman. A Simple, Accurate Method to Calculate Spread Spectrum Multiple Access Error Probabilities. *IEEE Trans. Commun.*, 40(3):461–464, March 1992.

[24] R. Buehrer and B. Woerner. Analysis of adaptive multistage interference cancellation for CDMA using an improved Gaussian approximation. *IEEE Trans. Commun.*, 44:1308– 1321, Oct 1996.