1st International IEEE Workshop on Emerging Densely Connected Networks (EDCN)

Multi-Connection DCCP User-to-User Video Streaming over Mobile WiMAX

Salah Saleh Al-Majeed and Martin Fleury
University of Essex, Colchester, United Kingdom

***Abstract*—The need to respond to congestion and channel loss over not one but two wireless links is an impediment to video streaming between one mobile user device and another via an intervening IP network. The result can be poor wireless channel utilization and interruptions to the streaming process. This paper proposes that multi-connection streaming will avoid the need for application-specific cross-layer intervention to address this problem. In considering IEEE 802.16e (mobile WiMAX), the paper demonstrates that multi-connection streaming is certainly necessary but only sufficient if an appropriately sized Time Division Duplex frame is selected. Received video quality will also depend on correct determination of the number of connections and the compression ratio of the video itself. The quality of the received video must be traded off against the time to stream the complete video sequence, which in term is determined by the number of connections.**

Index Terms— DCCP, multi-connection, video streaming, WiMAX

I. INTRODUCTION

Streaming a single video over multiple Datagram Congestion Control Protocol (DCCP) [1] connections is a promising way of separately coping with both wireless channel losses and traffic congestion, without the need for cross-layer intervention or retransmission delay at the data-link layer. At the same time the wireless channel is properly utilized [2], as throughput improves with an increasing number of connections. However, a scenario in which one mobile user device exchanges a video with another via a wired IP network [3] is challenging and has scarcely been investigated. The penalty if single-connection DCCP streaming is used is that the total streaming time may far exceed the display period of the video sequence, as simulation in this paper of this scenario has demonstrated. According to the ITU-T's 2001 G.1010 recommendation, start-up delay should not exceed by more than 10s the duration of a video sequence at a particular frame rate. Therefore, if the period during which the video is streamed exceeds this there will be excessive startup-time and there may be interruptions to the continuous display of the video. When there is not one but two wireless links at either end of the network path that will misleadingly cause the DCCP congestion mechanism to misinterpret channel loss as traffic congestion. As a consequence the inter-packet gap is widened by single-connection DCCP, leading to under utilization of the capacity and a general slow-down in delivery.

This paper proposes that DCCP can be used for user-to-user video streaming but with multiple connections. Over IEEE 802.16e (mobile WiMAX) [4], one of the candidate technologies for mobile backhaul, received video quality will depend on correct determination of the number of connections and the compression ratio of the video itself, as this affects the number of packets sent and, hence, buffer queuing through self-congestion. Tuning is also needed to select the size of Time Division Duplex (TDD) frame, as otherwise the potential gains from multi-connection streaming will not result. Simulation results in this paper show how the proposed user-to-user multi-connection streaming can be tuned to achieve the desired aim. One paradoxical result is that it may not always be enough to achieve optimal video quality at the receiver if that is achieved at a cost in an excessively long sending period.

DCCP is, along with Stream Control Transmission Protocol (SCTP) [5], a Standards-based way of providing congestion control for video streaming without the disadvantages of TCP. Though SCTP mitigates other TCP shortcomings such as lack of message structuring and exposure to SYN flooding, it still essentially provides a TCP-like reliable service. DCCP supports delay-sensitive streaming by means of UDP transport, which allows the delays that can arise from TCP's retransmissions whenever a packet loss is detected. DCCP has two varieties: TCP-like congestion control and TCP-Friendly Rate Control (TFRC) [6], one of which is selected during initial handshaking. TFRC is employed in this study as it reduces the saw-tooth like rate changes associated with TCP congestion control mechanisms. However, DCCP lacks an interesting feature of SCTP, which is support for multi-streaming with optional out-of-order delivery to avoid TCP's potential head-of-line blockages. More generally multi-connection DCCP can be compared with peer-to-peer (P2P) streaming, which is a video chunk-oriented mechanism with multiple connections from different sources. Just as in P2P streaming, multi-connection DCCP requires a reordering buffer. However, it is a packet-based streaming protocol and does not send chunks or use multiple sources.

In multi-connection DCCP video streaming, a *single* video source is multiplexed onto several connections across the wireless link in order to increase the throughput, thereby improving wireless channel utilization. In this way it is hoped that the impact of packet loss on one or more of these connections will be mitigated by the aggregate data-rate across the connections. Though cross-layer approaches to avoid misinterpretation of channel packet loss as congestion drops are possible [7], these approaches are complex to implement and inflexible. In fact, cross-layer approaches are most appropriate when a network has a fixed application, not one in which multimedia streaming is mixed in with other types of traffic.

In comparison with our work in [8] a straightforward simulation of DCCP, SCTP and UDP over a single fixed WiMAX link occurred. It appears that the channel may have been error free, though congestion occurs. The paper found that DCCP outperformed UDP and SCTP across a range of network metrics, though video quality was not given. [9] comes to much the same conclusion but for mobile WiMAX. Crucially these studies did not examine the total sending period, which with single-connection DCCP can far exceed the duration of the transmitted video clip.

II. Simulation model

The user-to-user scenario evaluated in this paper is shown in Fig. 1. The following describes the WiMAX parts and this description is followed by a description of the intervening IP network.

A. WiMAX components

In the Figure, once a WiMAX Base Station (BS) has allocated bandwidth to each Subscriber Station (SS), each SS must manage its queue according to the data arrival rate from user applications. WiMAX networks support multiple service classes to accommodate heterogeneous traffic with varying requirements. WiMAX's real-time polling service (rtPS) is most suitable for real-time video services, particularly for Variable Bitrate Video (VBR), which is employed herein to maintain delivered video quality but which may lead to 'bursty' arrival rates. Other traffic is assumed to enter the nonreal- time Polling Service (nrtPS) queue at the SS. In our experiments, for both queues a drop-tail queuing discipline was simulated, as it is the default for WiMAX. Queue sizes were all set to fifty packets. This value was selected as it seems appropriate to mobile, real-time applications for which larger buffer sizes might lead both to increased delay and larger memory energy consumption in mobile devices. Video is delivered from one of the SS's UL, passes through the intervening wired network until it is delivered over the downlink to another SS.

The PHY and other settings selected for the WiMAX simulations are given in Table I. The antenna is modelled for comparison purposes as a half-wavelength dipole. The Gilbert- Elliott 'bursty' channel model is further explained in Section II.D. The frame length is significant, as a longer frame reduces delay at the queue being serviced by permitting more data to be removed from queues at each polling event. The frame length values were varied according to available durations in the Standard [4].

B. WiMAX traffic characteristics

There were three SSs communicating to the BS over their uplinks, with one of the SS sending an H.264/AVC (Advanced Video Coding) [10] encoded VBR video sequence split between the multiple DCCP connections. The other SSs are introduced as sources of competing traffic across the wireless link and do not indicate the likely size of a WiMAX network, which obviously could be larger. In this WiMAX component (to the left of Fig. 1) there is also wireless congestion from BS sources occupying the downlink. Table II records the simulated traffic characteristics for the three SSs communicating with the BS. Network adaptation Layer Units (NALUs) output from the H.264/AVC encoder were encapsulated with Real Time Protocol (RTP) headers. After the addition of IP/UDP headers, these in turn formed a single WiMAX MAC Packet Data Unit (MPDU), which are variable sized WiMAX packets. For simplicity, a WiMAX MPDU is now referred to as a packet.

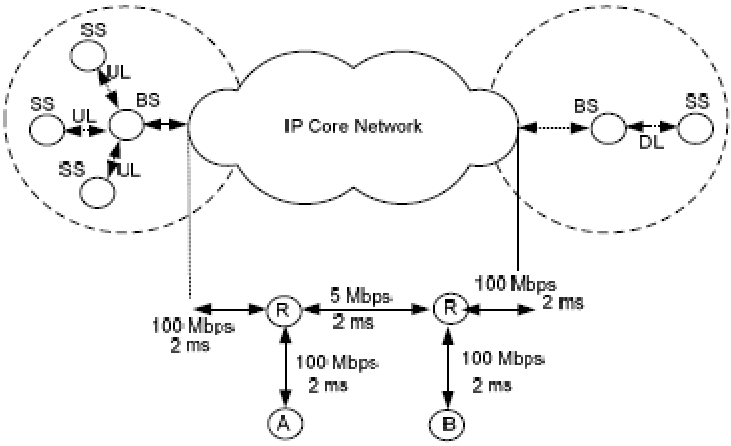

Figure 1. User-to-User device network with inset showing routing across the intervening network, A and B being sources and sinks, and R = router

TABLE I. SIMULATED WIMAX SETTINGS

Parameter	*Value*
PHY	OFDMA
Frequency band	5 GHz
Duplexing mode	TDD
Frame length	5–20 ms
Max. packet length	1024 B
Raw data rate	10.67 Mbps
IFFT size	1024
Modulation	16-QAM 1/2
Guard band ratio	1/8
UL/DL ratio	1:1
Channel model	Gilbert-Elliott
SS transmit power	250 mW
BS transmit power	20 W
Approx. range to MS	0.7 km
Antenna type	Omni-directional
Antenna gains	0 dBD
SS antenna height	1.5 m
BS antenna height	32 m

OFDMA = Orthogonal Frequency Division Multiple Access, QAM = Quadrature Amplitude Modulation, TDD = Time Division Duplex

Co-existing rtPS queue CBR sources were all sent at 1500 kbps, i.e. at a similar rate to the video source. The inter packet gap was 0.03 s for the CBR traffic. The FTP applications, which continuously supplied data according to available bandwidth, were set up out of convenience as a way of occupying the nrtPS queues; otherwise a Best-Effort queue might be more appropriate. Likewise, the DL traffic is simply selected to fully occupy the DL link capacity. For ease of interpretation, we did not introduce congestion at the wireless link at the receiver-side. Though traffic from other SSs and/or in a different service class does not directly affect waiting time or packet drops in WiMAX rtPS queues, there is an indirect effect, due to the service time consumed which could be otherwise spent serving the video-bearing queue.

For DCCP, the inter-packet sending time gap was varied according to the TFRC equation [6], not the simplified version reported in [2]. As described in [6], TFRC is a receiver-based system in which the packet loss rate is found at the receiver and fed-back to the sender in acknowledgment messages (through TCP in the simulations). The sender calculates the round-trip time from the acknowledgment messages and updates the packet sending rate. A throughput equation models TCP New Reno to find the sending rate:

$$TFRC\,(t_{rtt}, t_{rto}, s, p) = \frac{s}{t_{rtt}\sqrt{\dfrac{2bp}{3}} + t_{rto}\,\min\left(1, 3\sqrt{\dfrac{3bp}{8}}\right) p(1+32p^2)} \tag{1}$$

where t_{rtt} is the round-trip time, t_{rto} is TCP's retransmission timeout, s is the segment size (TCP's unit of output) (herein set to the packet size), p is the normalized packet loss rate, w_m is the maximum window size, and b is the number of packets acknowledged by each ACK. b is normally set to one and $t_{rto} = 4t_{rtt}$. It is important to notice that t_{rto} comes to dominate TFRC's behaviour in high packet loss regimes, which is why it is unwise to use a simplified form of (1). Clearing packet loss and round-trip time cause the throughput to decrease in (1), whereas other terms are dependent on these two variables in the denominator.

C. IP network traffic characteristics

In Fig. 1, all links except a bottleneck link within the IP network are set to 100 Mbps to easily accommodate the traffic flows entering and leaving the network. The link delays are minimal (2 ms) to avoid confusing propagation delay with reordering delay in the results. A bottleneck link with capacity not meant to physically correspond to a network layout but to represent the type of bottleneck that commonly lies at the network edge.

In order to introduce congestion within the wired network, Node A sources to node B a CBR stream at 1.5 Mbps with packet size 1 kB and sinks a continuous TCP FTP flow sourced at node B. Node B also sources an FTP flow to the BS and a CBR stream at 1.5 Mbps with packet size 1 kB (see Table II downlink). The effect of traffic sourced from B is to cause some congestion to the returning TFRC acknowledgement. Other SS sources apart from the video connections do not pass over the IP network shown but are assumed to be routed elsewhere after passing the WiMAX BS.

D. Management of connections

To systematically test the effect of multiple DCCP connections the number of DCCP connections was incrementally stepped up in successive experiments. In our experiments, video slices were multiplexed onto the multiple connections, with the slice size being set at the codec to 900 B. The advantage of forming slices in this way is that no segmentation takes place at the network level, thus avoiding breaks in synchronization at the decoder from splitting slices. Each slice contains resynchronization markers allowing the decoder's entropic encoder to restart processing. Each connection was statically allocated its slices, which are taken in interleaved manner from the video sequence. As previously mentioned, this assumes that a re-ordering buffer is available at the receiver. The multiple connections then enter the SS's rtPS queue before transmission over the WiMAX link to the BS.

TABLE II. SIMULATED WIMAX TRAFFIC CHARACTERISTICS

SS-UL	*Service type*	*Traffic type*	*Protocol*	*Packet Size (B)*
1	rtPS	VBR (video)	Multiple TFRC	Variable 1000
	nrtPS	CBR	UDP	
		FTP	TCP	
2	rtPS	CBR	UDP	1000
	nrtPS	FTP	TCP	
3	rtPS	CBR	UDP	1000
	nrtPS	FTP	TCP	
SS-DL				
1,2	rtPS	CBR	UDP	1000
3	nrtPS	FTP	TCP	

E. Channel model

A Gilbert-Elliott two-state, discrete-time, ergodic Markov chain [11] modelled the wireless channel error characteristics. The probability of remaining in the good state was set to 0.95 and of remaining in the bad state was 0.94, with both states modelled by a Uniform distribution. The packet loss probability in the good state was fixed at 0.01 and the bad state default was 0.05. These values were established by prior tests to be not so severe that all video quality was poor and not so benign that little impact was felt from the channel conditions. Space does not permit a fuller report on the effect of varying the channel conditions.

III. EVALUATION

A. Video input

The well-known network simulator ns2 (v. 2.29 employed) was employed to evaluate the behaviour of the multi-connection scheme in the scenario of Section II. Data points are the arithmetic mean of at least ten runs. A video trace file was input to ns-2 and packet losses recorded in the output. The output serves to calculate the PSNR. Video quality comparisons were made under the EvalVid environment (version 2) [12]. As a test sequence, we used the *Paris* clip at 30 Hz (frame/s) at Common Intermediate Format (CIF) (352 x 288 pixel/frame) with quantization parameter (QP) set to 26 (from a range 0 to 51), i.e. medium-quality video by default. At this QP, after comparison with input raw YUV video the PSNR was found to be 38 dB PSNR. Other tests varied the QP value to judge the effect of increasing or decreasing the quality, which of course changes the compression ratio and, hence, the packet sizes. Intra-refresh rate was every 15 frames with IPBB…I structure (sending order), i.e. allowing a user to swap channels every 30 s, as is normal. 1065 frames were transmitted, the total length of the *Paris* sequence. At 30 Hz, this number of frames corresponds to a duration of 35.5 s. Simple Previous Frame Replacement (PFR) was set for error concealment at the decoder for comparison with others' results.

Paris consists of two figures seated around a table in a TV studio setting, with high spatial coding complexity. News clips are reported as the most resulting in a good quality-of experience (QoE) compared to other video genre [13]. Another issue of interest is whether high quality is appropriate to mobile devices. Though the viewing distance is reduced, the screen resolution may not be increased. The knowledge that the device is mobile can also cause a reduction in expectations to 'fair' quality, between 25 and 31 dB PSNR.

B. Performance with constant QP

The behaviour with video quality at QP =26 was initially examined. Table III captures the basic problem with streaming with just one connection for video streaming. The period over which the test clip is sent is far longer than the time to display the clip at a rate of 30 Hz. To achieve sufficient wireless channel utilization a combination of at least six connections and a TDD frame size of 12.5 ms is required. Another interesting feature of Table III is that the sending period (the time duration between when streaming commences to when it ends) is dependent on TDD frame size. Too large a frame size (for an acceptable sending period) benefits uplink streaming but not downlink streaming, and too small a frame size, benefits downlink streaming but not uplink streaming. Therefore, setting an appropriate frame size to achieve an acceptable sending period is a compromise between uplink and downlink requirements. Finally notice from Table III, that increasing the number of connections above eight would increase the throughput still further but for a frame size of 12.5 ms this gain would be counterproductive because it would introduce a need to buffer the arriving video before it could be played out, as the sending period is less than the duration of the sequence. Thus we did not continue the experiments beyond eight connections.

The sending period does not reveal the order of arrival of packets, as it is entirely possible that packets bearing compressed slices from later frames in the display order arrive before those of earlier frames. As an indication, in [14] allowance was made for 10s start-up buffering before beginning decode, to avoid subsequent buffer underflow. However, that work [14] used data from MPEG-4 at only 10 Hz to test buffer occupancy. Another important point is that throughput is not distributed evenly over the period of streaming. In fact, the behaviour of DCCP in a multi-connection configuration is that the available bandwidth is probed initially but then there is a large increase in throughput towards the end of the sending period. This point is returned to in Section II.C. However, we surmise that the reasons for this behaviour are a combination of an over-reaction to loss or delay of acknowledgment packets at an early stage when DCCP is probing the available bandwidth and a growing dominance of a few connections as time passes by.

Packet end-to-end delay between the two user mobile devices is not just a result of queuing delay caused by congestion from coexisting traffic. DCCP reduces or staunches its sending rate in response to packet loss. By doing so it reduces self-congestion within the transmitter queue. In the scenario tested, Table IV, with just one connection jitter is relatively high and end-to-end delay rises above 100 ms if the TDD size is not chosen appropriately. High jitter arises because of the need because of the adverse reaction to packet loss by single connection DCCP. By introducing just one more connection, the packet losses are split in two between the connections, allowing DCCP to regulate its packet flow more evenly. Similar and small delays arise for other numbers of connections. However, the possibility of interactive streaming with end-to-end delays at this low level is precluded by the need to reorder packets.

As the total number of fixed size packets (900 B) with QP = 26 was 4739, from Table V no more than 7% of the slices were on average lost, whereas QoE subjective testing [11] suggests that broadly a round figure of 10% losses is needed on mobile devices to cause significant deterioration in delivered video quality, below 25 dB PSNR. Turning to objective video quality, Table VI, though PSNR with one connection is generally better, poor channel utilization renders this quality unusable for streaming, as it also does in this scenario for a TDD frame size of 20 ms. This is because the result of poor channel utilization is that the video sequence simply takes too long to transmit.

C. Performance with constant TDD frame

We also considered the consequences of a decision to either improve the VBR video quality at the sender (QP = 20) or to reduce it (QP = 35). The TDD frame size was fixed to 12.5 ms, as that setting gave superior overall performance in Section III.B. Table VII demonstrates that again the sending period for one connection is too long. It also shows that the increase in the number of slices (packets) to 8839 causes an unacceptable increase in the sending period if the QP is set to 20. Conversely, setting the QP to 35 reduces the number of packets to 2077, causing the sending period to be too rapid. From Fig. 2, there is a rising trend of throughput for the different QP settings (with a 20 s settling time for the simulator before streaming starts), giving rise to interruptions, causing 'freeze frames' at the display device unless addressed. Space is not sufficient to report our investigation of the size of a reordering buffer needed though we found it to be about 6 s. This implies a 6 s start-up delay in streaming. Comparing with [14], an allowance there was made for 10 s start-up buffering before beginning decode, also to avoid buffer underflow.

From Table VIII, the packet loss is no more than 5% for QP = 20, while for QP = 35 it is no more than 7%. A combination of lower initial quality and a higher ratio of packet loss results in PSNR, Table IX, which passes below the 25 dB level. Unfortunately also, the sending period for transmitting the higher quality video is too long. End-to-end delay and jitter did not noticeably depart from the general levels of Table IV, with mean packet delay in the multi-connection configurations being no more 75 ms for QP = 20 and no more than 58 ms for QP = 35, reflecting the differing numbers of packets contributing to congestion.

IV. CONCLUSION

Operating user-to-user streaming between mobile devices can be accomplished by means of multi-connections. The simulated scenario in this paper shows that the sending period may simply be too long if a single connection is used. This study has suggested that low numbers of connections, four or perhaps six will normally be sufficient and increasing the number of connections results in too short a sending period. For WiMAX, the TDD frame size is likely to be critical, which may be a problem if smaller frame sizes to support other data services are needed. However, the findings in this paper can be transferred to other broadband wireless technologies.

TABLE VII. MEAN THROUGHPUTS AND SENDING PERIODS FROM USER-TO-USER DEVICE FOR DIFFERENT ORIGINAL VIDEO QUALITIES

	QP=20		QP=35	
	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)
1 conn.	119	196	105	78
4 conn.	575	96	335	26
6 conn.	790	69	432	21
8 conn.	930	59	553	15

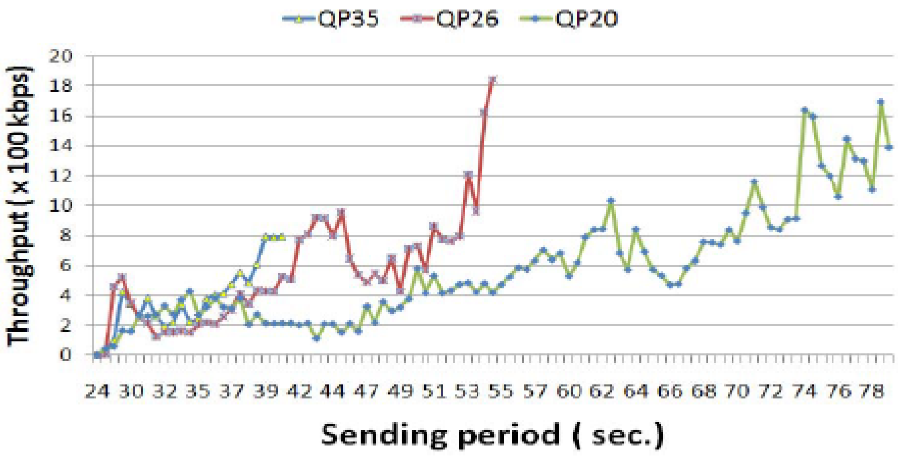

Figure 2. Throughput from sending the Paris video sequence at different QP encodings over eight connections using a 12.5 ms TDD frame size.

TABLE VIII. MEAN PACKET LOSSES FOR TDD FRAME SIZE = 12.5 MS

	QP = 20		QP=35	
1 conn.	133	1.50%	78	3.76%
4 conn.	400	4.53%	135	6.50%
6 conn.	440	4.98%	140	6.74%
8 conn.	430	4.56%	143	6.88%

TABLE IX. MEAN PER FRAME PSNR AND VARIATION

	QP=20		QP=35	
	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)
1 conn.	31.08	2.91	27.15	2.51
4 conn.	29.75	4.20	24.79	2.75
6 conn.	29.32	4.02	24.79	3.11
8 conn.	28.20	3.50	25.07	3.16

REFERENCES

[1] E. Kohler, M. Handley, and S. Floyd, “Datagram Congestion Control Protocol,” *Internet Engineering Task Force*, March 2006, RFC 4340.

[2] M. Chen, and A. Zakhor, “Multiple TFRC connection based rate control for wireless networks,” *IEEE Trans. Multimedia*, vol. 8, no. 5, pp.1045–1062, Oct. 2006.

[3] B. Görkemli, M. O. Sunay, and A. M. Tekalp, “Video streaming over wireless DCCP,” in *IEEE Int’l Conf. on Image Processing*, Oct. 2008, pp. 2028–2031.

[4] IEEE, 802.16e-2005, “IEEE Standard for Local and Metropolitan Area Networks. Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems,” 2005.

[5] R. Stewart, ed. “Stream Control Transmission Protocol,” *Internet Engineering Task Force*, Sept. 2007, RFC 4960.

[6] M Handley, S. Floyd, J. Padhye, and S. Widmer, “TCP-Friendly Rate Control (TFRC): Protocol Specification,” *IETF*, Jan. 2003, RFC 3448.

[7] H. Balkrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Trans. on Networking,* vol. 5, no. 6, pp. 756–769, 2007.

[8] H.M.O. Chughtai, S.A. Malik, and M. Yousaf, “Performance evaluation of transport layer protocols for video traffic over WiMax,” IEEE 13th International Multitopic Conference, 2009, 6 pages.

[9] M.N. Khalid, "Simulation based comparison of SCTP, DCCP, and UDP using MPEG-4 traffic over mobile WiMAX/IEEE 802.16e"', MSc report, Blekinge Inst. of Tech., Sweden, 2010.

[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 13, no. 7, pp. 560–576, July 2003.

[11] G. Haßlinger, and O. Hohlfeld, “The Gilbert-Elliott model for packet loss in real time services on the Internet,” in *14th GI/ITG Conf. on Measurement, Modelling, and Evaluation of Computer and Commun. Systs.,* 2008, pp. 269-283.

[12] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid — A framework for video transmission and quality evaluation,” in *Int’l Conf. on Modeling Techniques and Tools for Computer Performance*, 2003, pp. 255–272.

[13] F. Agboma, M. Smy, and A. Liotta, “QoE analysis of a peer-to-peer television system,” in *IADIS Int. Conf. on elecommunications, Networks and Systems*, 2008.

[14] M. Chen and A Zakhor, “Rate control for streaming video over wireless,” *IEEE Wireless Comms.*, vol. 12, no. 4, pp. 32–14, Aug. 2005.

TABLE III. MEAN THROUGHPUTS AND SENDING PERIODS FROM USER-TO-USER

	TDD frame size					
	8 ms		*12.5 ms*		*20 ms*	
	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)
1 conn.	187	148	187	152	100	179
4 conn.	658	41	745	37	466	60
6 conn.	630	40	797	34	591	46
8 conn.	653	38	910	29	706	39

TABLE IV. MEAN PACKET END-TO-END DELAY AND JITTER

	TDD frame size					
	8 ms		*12.5 ms*		*20 ms*	
	Delay (s)	Jitter (s)	Delay (s)	Jitter (s)	Delay (s)	Jitter (s)
1 conn.	0.041	0.034	0.063	0.033	0.139	0.063
4 conn.	0.048	0.009	0.057	0.008	0.081	0.012
6 conn.	0.047	0.007	0.058	0.006	0.077	0.008
8 conn.	0.046	0.006	0.061	0.005	0.076	0.007

TABLE V. MEAN PACKET LOSSES

	TDD frame size		
	8 ms	12.5 ms	20 ms
1 conn.	331	202	119
4 conn.	348	217	253
6 conn.	453	272	294
8 conn.	469	320	304

TABLE VI. MEAN PER FRAME PSNR AND VARIATION (STDV = STANDARD DEVIATION)

	TDD frame size					
	8 ms		12.5 ms		20 ms	
	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)
1 conn.	28.85	5.11	29.21	3.16	33.51	4.69
4 conn.	26.93	3.03	29.88	4.32	28.99	3.55
6 conn.	26.11	3.17	28.47	3.41	28.54	4.06
8 conn.	25.53	2.95	27.22	3.24	28.50	3.61

TABLE III. MEAN THROUGHPUTS AND SENDING PERIODS FROM USER-TO-USER

	TDD frame size					
	8 ms		*12.5 ms*		*20 ms*	
	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)
1 conn.	187	148	187	152	100	179
4 conn.	658	41	745	37	466	60
6 conn.	630	40	797	34	591	46
8 conn.	653	38	910	29	706	39

TABLE IV. MEAN PACKET END-TO-END DELAY AND JITTER

	TDD frame size					
	8 ms		*12.5 ms*		*20 ms*	
	Delay (s)	Jitter (s)	Delay (s)	Jitter (s)	Delay (s)	Jitter (s)
1 conn.	0.041	0.034	0.063	0.033	0.139	0.063
4 conn.	0.048	0.009	0.057	0.008	0.081	0.012
6 conn.	0.047	0.007	0.058	0.006	0.077	0.008
8 conn.	0.046	0.006	0.061	0.005	0.076	0.007

TABLE V. MEAN PACKET LOSSES

	TDD frame size		
	8 ms	12.5 ms	20 ms
1 conn.	331	202	119
4 conn.	348	217	253
6 conn.	453	272	294
8 conn.	469	320	304

TABLE VI. MEAN PER FRAME PSNR AND VARIATION (STDV = STANDARD DEVIATION)

	TDD frame size					
	8 ms		12.5 ms		20 ms	
	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)
1 conn.	28.85	5.11	29.21	3.16	33.51	4.69
4 conn.	26.93	3.03	29.88	4.32	28.99	3.55
6 conn.	26.11	3.17	28.47	3.41	28.54	4.06
8 conn.	25.53	2.95	27.22	3.24	28.50	3.61

TABLE VII. MEAN THROUGHPUTS AND SENDING PERIODS FROM USER-TO-USER DEVICE FOR DIFFERENT ORIGINAL VIDEO QUALITIES

	QP=20		QP=35	
	Thru-put (kbps)	Period (s)	Thru-put (kbps)	Period (s)
1 conn.	119	196	105	78
4 conn.	575	96	335	26
6 conn.	790	69	432	21
8 conn.	930	59	553	15

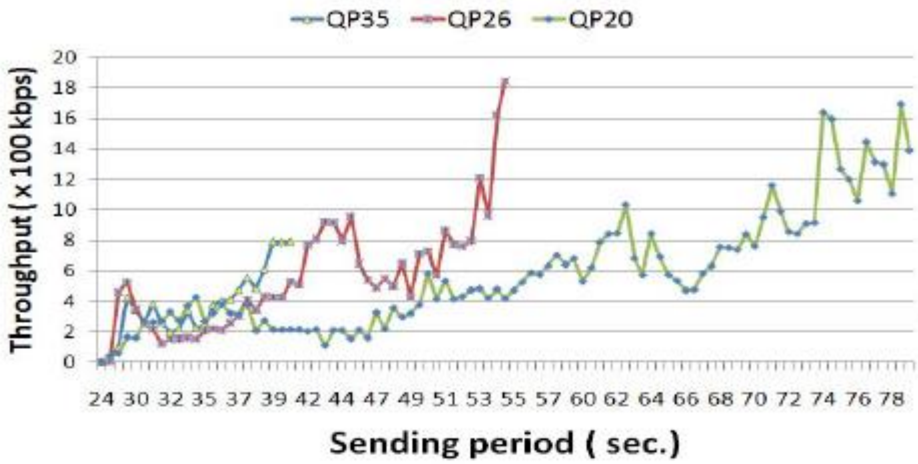

Figure 2. Throughput from sending the Paris video sequence at different QP encodings over eight connections using a 12.5 ms TDD frame size.

TABLE VIII. MEAN PACKET LOSSES FOR TDD FRAME SIZE = 12.5 MS

	QP = 20		QP=35	
1 conn.	133	1.50%	78	3.76%
4 conn.	400	4.53%	135	6.50%
6 conn.	440	4.98%	140	6.74%
8 conn.	430	4.56%	143	6.88%

TABLE IX. MEAN PER FRAME PSNR AND VARIATION

	QP=20		QP=35	
	PSNR (dB)	Stdv. (dB)	PSNR (dB)	Stdv. (dB)
1 conn.	31.08	2.91	27.15	2.51
4 conn.	29.75	4.20	24.79	2.75
6 conn.	29.32	4.02	24.79	3.11
8 conn.	28.20	3.50	25.07	3.16

[7] H. Balkrishnan, V. Padmanabhan, S. Seshan, and R. Katz, “A comparison of mechanisms for improving TCP performance over wireless links,” *IEEE/ACM Trans. on Networking,* vol. 5, no. 6, pp. 756–769, 2007.
[8] H.M.O. Chughtai, S.A. Malik, and M. Yousaf, “Performance evaluation of transport layer protocols for video traffic over WiMax,” IEEE 13th International Multitopic Conference, 2009, 6 pages.
[9] M.N. Khalid, "Simulation based comparison of SCTP, DCCP, and UDP

using MPEG-4 traffic over mobile WiMAX/IEEE 802.16e'", MSc report, Blekinge Inst. of Tech., Sweden, 2010.
[10] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 13, no. 7, pp. 560–576, July 2003.
[11] G. Haßlinger, and O. Hohlfeld, “The Gilbert-Elliott model for packet loss in real time services on the Internet,” in *14th GI/ITG Conf. on Measurement, Modelling, and Evaluation of Computer and Commun. Systs.,* 2008, pp. 269-283.
[12] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid — A framework for video transmission and quality evaluation,” in *Int'l Conf. on Modeling Techniques and Tools for Computer Performance*, 2003, pp. 255–272.
[13] F. Agboma, M. Smy, and A. Liotta, “QoE analysis of a peer-to-peer television system,” in *IADIS Int. Conf. on Telecommunications, Networks and Systems*, 2008.
[14] M. Chen and A Zakhor, “Rate control for streaming video over wireless,” *IEEE Wireless Comms.*, vol. 12, no. 4, pp. 32–14, Aug. 2005.