Mixed Reliability Video Streaming in Wireless Networks

Justice Daka and Dwight Makaroff
Department of Computer Science
University of Saskatchewan
Saskatoon, SK S7N 5C9 Canada
jud013,makaroff@cs.usask.ca

Abstract

Video has become an important application for wireless networks as they have become popular for many users and uses. There are challenges delivering video content over a wireless link, both due to the volume of traffic and the high data loss rates experienced. Data loss in compressed media results in errors in the decoded video, and noticeable visual artifacts. These losses can have long-term effects in video display.

This paper presents the concept of mixed reliability video streaming (Mixed Streaming), which reduces the impact of video propagation errors in error prone wireless networks. Mixed Streaming delivers a video file using both reliable and best-effort connections. Simulation results show that Mixed Streaming reduces the impact of errors by making sure that errors on reference frames are corrected. Also, the delay cost associated with Mixed Sreaming is reasonable even for fairly high packet loss rates.

1 INTRODUCTION

The Internet is increasingly being used for streaming of video. There are currently three popular techniques for delivering streaming media: downloading, progressive download, and streaming. True streaming has tended to use UDP as an underlying protocol for data delivery while the download and progressive download techniques use TCP [10].

True streaming over UDP has been the delivery method of choice in most video research, as total reliability of data is not necessary for video decoding and the extra delay induced by a reliable protocol, such as TCP, has been viewed as unacceptable. Unfortunately, some networks block or restrict the volume of UDP traffic, such that true streaming is not practical. To address this, some video server products such as Windows Media Server use TCP for streaming [10]. In wired networks, this reliable protocol seems to provide acceptable performance under varied network conditions.

In a wireless network, channel loss is a more significant problem than in the wired network. TCP cannot distinguish between channel loss and congestion loss, however, and this leads to inappropriate response to loss [3]. Packet loss causes severe problems in video streaming applications. For instance, a packet loss rate of 3% could lead to loss of 30% of the frames [6]. The reason for the difference in data loss at the packet level and at the video frame level is due to temporal encoding schemes, such as MPEG.

The propagation error problem can be solved with two extreme approaches: proactive and reactive. For a proactive approach, the system takes action before the error occurs, while in a reactive approach, action is only taken when the error occurs. One proactive solution is to use Forward Error Control (FEC) [5] and transmit the video together with redundant data used for correcting errors, consuming additional bandwidth irrespective of packet loss. Packet dropping can also be used to reduce congestion [14]. Reactive approaches employ error recovery at the receiver after data loss. One option is retransmission, but this introduces undesirable delay. Post-processing error concealment is also a possibility, but assumes that neighboring frames experience little or no data loss [20]. These techniques are less likely to be effective for high loss rates as in wireless networks.

We develop the concept of *mixed reliability video streaming (Mixed Streaming)* over wireless networks and conduct a simulation performance study. Our approach is primarily proactive. We use multiple streams (reliable and best-effort transmission) to simultaneously deliver a video. Important (reference) data is sent reliably while the less important (non-reference) data is transmitted unreliably. To view a video, a decoder must be used which reconstructs the stream from time-stamped packets.

The rest of this paper is arranged as follows. Section 2 presents the related work. Section 3 gives the specification of Mixed Streaming. In Section 4, the experimental environment is described. In Sections 5 and 6 we discuss initial results. Finally, Section 7 presents conclusions and future work.

978-1-4244-2100-8/08/$25.00 ©2008 IEEE

2 RELATED WORK

Packet loss in streaming video has been addressed using various approaches. Methods used in prior work address issues at the transport layer [7, 12], MAC layer [8, 18], and application layer [19].

Tripathi and Claypool [19] reduces the amount of video transmitted by performing content aware scaling under congestion, by either dropping less important frames (temporal), or by re-quantizing the video (spatial). Chen and Zakhor [7] showed that using one TCP or TCP-Friendly Rate Control (TFRC) connection for video transmission underutilizes the available bandwidth on a wireless network and propose using multiple TFRC connections. Their results showed that using more than one connection efficiently utilizes the bandwidth, but is susceptible to fluctuations in video quality.

Huang *et al.* [12] studied the transmission of streaming media over the wired Internet to wireless networks. They proposed a proxy-based approach that uses rate control on the wired network and no congestion control on the wireless network. Using TCP only on the wired network shields TCP from cutting its sending rate due to non-congestion losses on the wireless network. While this reduces the buffering delay, it is unclear how video quality is affected with high loss rates on the wireless network.

Fitzek and Reisslein [8] examined the use of unmodified TCP over wireless. In order to hide the errors that occur on the wireless channel from TCP, they used multiple wireless channels to reduce MAC layer retransmission delays, significantly increasing the performance of video over wireless networks. Singh *et al.* [18] proposed a link layer method that works together with a modified transport layer protocol. Specifically, they used a modified UDP (called UDP-lite), which ignores the packet checksum if some data loss occurs in the packet payload, allowing delivery of partially corrupted packets to the decoder. Shorter end to end delays compared to traditional UDP were observed, as well as slightly better video quality.

Our method is similar to Chen and Zakhor [7]. Their approach aims to increase the throughput over a wireless network, while we try to improve the quality of the video in the midst of wireless packet losses. Our approach also uses multiple levels of reliability. A similar approach is used to deliver 3-D models by Al-Regib and Altunbasak [2].

3 MIXED RELIABILITY PROTOCOL

3.1 Common Streaming Techniques

TCP-based streaming is attractive because it delivers high quality video, especially in low data loss environments. It provides reliability features, such as retransmissions and congestion control. When packets are lost, TCP retransmits the lost packet and may cut its sending rate. This results in a big delay for high loss environments and low bandwidth utilization. The delay emanates from TCP's ordered delivery of packets to the application, which also causes less important frames to block more important I-frames.

There are 3 types of frames in MPEG video: Intracoded frames (I) that are self-contained, Predictive frames (P) that encode the difference from the previous I-frame, and Bidirectional frames (B) that use information from the nearest I-frame or P-frame in either direction for their reference. Figure 1 shows the transmission of an MPEG-encoded video file with an associated transmission schedule (TS). Figure 2 shows how TCP's transmission schedule is affected after losing data. Losing frames P_2, B_4 and B_{10} results in the overall transmission schedule being extended by three time slots, assuming it takes one time slot to retransmit a lost frame. For instance, B_{13} is transmitted at time slot 16 instead of time slot 13. B_3 is transmitted on schedule even after losing P_2 as TCP requires 3 duplicate acks to detect a loss. Losing less important frames results in frame I_{11} being delayed by two time slots. In addition to extending the transmission schedule, TCP will assume the packet loss is due to congestion even when it is from bad channel conditions. As a result, TCP cuts the transmission rate to half and invokes the slow start algorithm after a timeout [16].

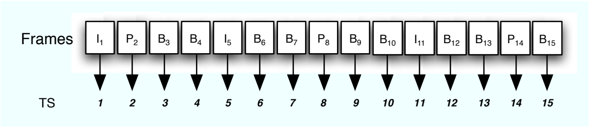

Frames
I_1 P_2 B_3 B_4 I_5 B_6 B_7 P_8 B_9 B_{10} I_{11} B_{12} B_{13} P_{14} B_{15}
TS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Figure 1. Video transmission: No data loss

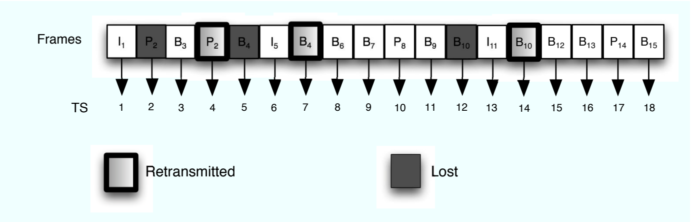

Frames
I_1 P_2 B_3 P_2 B_4 I_5 B_4 B_6 B_7 P_8 B_9 B_{10} I_{11} B_{10} B_{12} B_{13} P_{14} B_{15}
TS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Retransmitted
Lost

Figure 2. Video transmission: Data loss

To eliminate TCP's download delay, UDP is often used for streaming. UDP transmits the video according to the TS, because it does not restore lost data, but can be too aggressive if there is no rate control, leading to congestion. The loss of data in a reference frame damages any subsequent dependent frame. In order to overcome these limitations, *Mixed Streaming* separates the reference and dependent data using two connections.

3.2 Specification of Mixed Streaming

To stream a video clip using Mixed Streaming, the video file is split and stored into two parts. The first file contains independent video data (I-frames) and the second file has the dependent video data (P and B-frames). The independent data is progressively downloaded using a reliable TCP connection and the dependent data is streamed unreliably using UDP. I-frames are streamed back to back while the P and B-frames are streamed when they are needed. Figure 3 illustrates Mixed Streaming with packet loss. The unlabeled boxes represent free time slots.

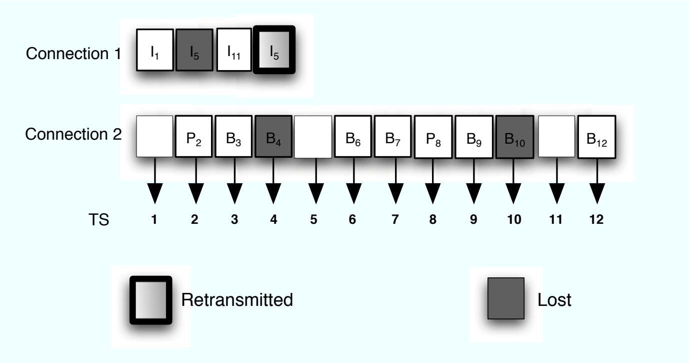


Figure 3. Video transmission: Mixed

If we assume each frame will fit into a single packet, Figure 3 shows that I-frames are not blocked by P and B-frames and that lost I-frame packets can be retransmitted before they are required for decoding/play-out . For instance, frame I_5 is initially sent at time slot 2, and then resent at time slot 4 but is not needed until time slot 7. I_5 can thus be retransmitted in time for playback. Many TCP variants attempt to provide in-order delivery and selective retransmission, further reducing the impact on the play-out schedule.

The loss of P and B-frames does not add delay to the original transmission schedule. The frames on connection 2 are transmitted over UDP according to the time specified by the encoder. When a video clip is streamed via Mixed Streaming, it is likely to have a shorter start-up delay than downloading and progressive download approaches. One disadvantage of using two connections simultaneously when the bandwidth is limited is an increase in congestion. At high traffic and high loss rates, we expect very few dependent frames would be successfully transmitted.

4 EXPERIMENTAL ENVIRONMENT

4.1 Performance Metrics and Factors

The first performance metric that we use is *Peak Signal to Noise Ratio(PSNR)* [21]. PSNR is the most common method used to measure video quality. For the interested reader, Wolf and Pinson [21] provides a detailed discussion of other video measurement techniques. One common subjective technique is the Mean Opinion Score (MOS). These labour-intensive experiments are left to future work.

Another performance metric used to measure the video quality is the *frame rate*. To calculate the frame rate, all frames that fall below a set threshold are regarded as incorrectly received. The threshold is chosen by observing the PSNR value at which the video becomes unwatchable.

The performance of the schemes will also be studied in terms of the *delay* that is introduced during transmission and buffer space needed for smooth delivery. If a packet is delayed, it will cause undesirable jitter. Thus frames are buffered to permit the startup delay. We also quantify the *buffer space* needed for the startup delays.

4.2 Experimental Tools

We use three activity levels of video clips: high, alternating, and low. For the high action category, a music video (Santana) is used, which has scenes of a musical concert with people dancing and singing. For the alternating action category, a video clip from the movie “The Matrix” is used with a combination of fast paced chases and low action scenes of a person seated on a computer. For the low action category, the Bridge video clip is used, containing a scene of a bridge with water under it. The GOP size in these videos is 30 frames, which is rather long. Tests with shorter GOP sizes show similar results. The characteristics of the videos are shown in Table 1.

The video files are encoded to MPEG-4 using FFMPEG [17]. We use Evalvid [13], a video evaluation tool-set, to parse the files and produces a packet trace. Evalvid is also used after network simulations to analyze network traces and to generate the resulting video as shown in Figure 4.

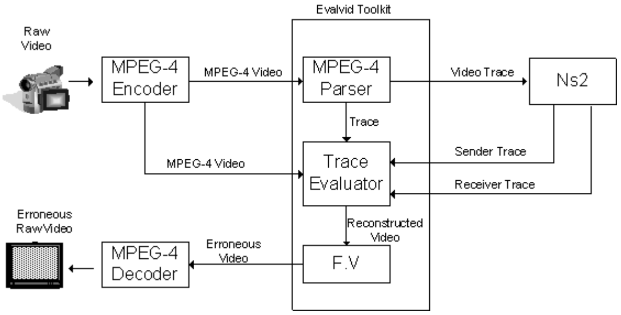


Figure 4. Experimental tools and setup

4.3 Simulation Environment

The *ns-2* simulator generates its traffic by reading a video trace file that is produced by the MPEG-4 parser of

Table 1. Video Sequence Characteristics

File	Santana	Matrix	Silent
Resolution	CIF(352 x 288)	CIF	CIF
Duration	15 min	12 min	12 min
Bitrate	200 Kbps	200 Kbps	200 Kbps
Frames	22500	18000	18000
Action	high	alternating	low
FPS	25	25	25

the Evalvid tool-set. The frame sizes from the trace file are used to create network packets with the appropriate sizes.

The simulation experiments use a dumbbell network topology as shown in Figure 5. This simple network topology is similar to that used by Balakrishnan *et al.* [3]. The sources are wired nodes while the sinks are wireless nodes. R1 is a router and R2 is a base station node.

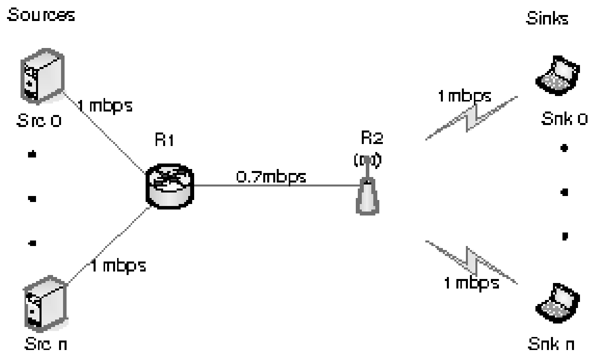


Figure 5. Topology

A bottleneck link connects R1 and R2. The bottleneck capacity is set to 0.7 Mbps unless specified differently. All other links are 1 Mbps. The routers use FIFO queuing with drop-tail queue management and have a 30 packet queue length (which exceeds the bandwidth delay product of the bottleneck link [1]). The one-way propagation delay is set to 2 ms between source 0 and R1, and 100 ms for the bottleneck link, a typical Internet propagation delay [4].

The network link between R2 and sink 0 is an 802.11 wireless network with a propagation delay of 2 ms, characterized by lossy periods and loss-free periods. The losses are caused by link impairments, such as multi-path fades and impulsive noise. The error on the wireless channel is modeled using the Gilbert model [9].

The time spent in the good/bad state of the Gilbert model is set dynamically for each packet loss rate. The time spent in the good state is between 0.75 and 1 ms and the time spent in the bad state is between 0 and 0.25 ms. Three different packet loss levels are considered in detail: 5%, 15% and 25%. The choice of packet loss rates is guided by measurement studies [6, 15] that showed wireless networks have a wider range of packet loss rates than wired networks.

5 VIDEO QUALITY EXPERIMENTS

5.1 Video Quality with Packet Loss

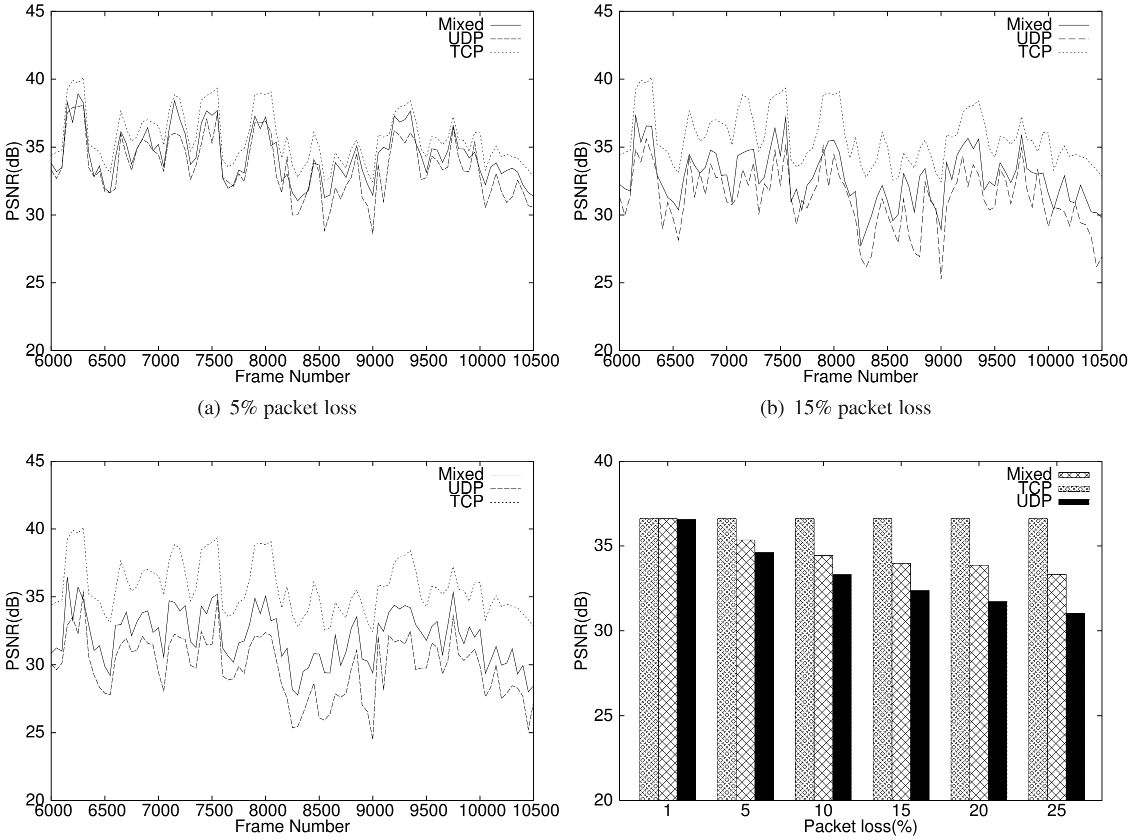
Data loss leads to impairments, such as pixelation effects and black spots in the video. Errors that occur on B-frames are isolated to that frame, while errors on P-frames will last for a few frames (intervening B frames). Errors on independent frames are propagated to the entire GOP.

In this section, the quantitative impact of packet loss on video clips transmitted by UDP and Mixed Streaming is evaluated via simulation. Before the results of the experiments are discussed, the characteristics of the experiment are outlined. The bottleneck capacity, round trip time and the topology are as described in the previous section. TCP streaming sends video packets as fast as possible (we use TCP-NewReno). For UDP streaming, video packets are delivered when they are needed (i.e according to the time specified by the encoder), as some limited form of rate control. As previously mentioned, Mixed Streaming sends TCP packets as fast as possible and sends UDP packets when they are needed. Sending UDP data only when needed reduces the chances of congestion because the UDP component by itself is less likely to overwhelm the receiver.

Video clips are decoded at the receiver after being transmitted with default MPEG-4 error resilience methods enabled. Measurements start after a warm-up time of 20 seconds. TCP measurement experiments have shown that it takes about 20 seconds for TCP to become stable [1].

For Santana, Figure 6 shows the video quality delivered with UDP and Mixed Streaming at various packet loss rates and no congestion. The y-axis shows the PSNR of each frame. Since the PSNR is logarithmic and all the values are higher than 20 dB, the y-axis scale starts at 20 dB. For the purpose of clarity, only 4500 frames are shown for each plot and 50 frames (2 seconds) were averaged for each plot point. Each plot line represents the average of 10 experimental replications. Similar results are obtained for other frame ranges. The standard deviation for each frame is quite large, because losing a packet may greatly reduce the frame quality. The deviation for each data point is also large, as it groups 1.6 GOPs. If an I-frame is corrupted in UDP, this will affect the quality of the entire GOP.

The results in Figure 6(a) shows that even 5% packet loss reduces the video quality delivered by Mixed Streaming and UDP. Despite MPEG-4 error control methods being enabled, frame corruption occurs. Similar results for other video clips were recorded, but are not shown, due to space



(a) 5% packet loss

(b) 15% packet loss

(c) 25% packet loss

(d) Avg. frame quality (No congestion)

Figure 6. Delivered video quality of the Santana clip for various loss rates

restrictions. Notably, the difference was proportional to the amount of action.

TCP delivers the highest video quality because it restores all the lost data. As expected, Mixed Streaming delivers the second highest video quality and UDP delivers the lowest quality. It can be observed from Figure 6(a) that there are instances where UDP delivers better video quality than Mixed Streaming for some frames. At first, this seems impossible, but due to compression and the error model used, such behavior is not unreasonable to expect that there will be some statistical variation in simulation runs.

To see why UDP delivers better quality than Mixed Streaming for some data points, traces from the *ns-2* simulator and the Evalvid evaluation tool were examined in more detail. Each individual run has a different loss profile. As a result, different frames are lost in each run. Therefore, UDP could have a higher PSNR average for the plot point than Mixed Streaming for that particular frame range.

When the packet loss rate is raised to 15%, the video shows a more significant drop in quality for UDP and Mixed due to packet loss. A difference of more than 5 dB in some cases is recorded in Figure 6(b). A PSNR difference of 1 dB may be visible, and consequently the MPEG committee set an improvement threshold of 0.5 dB to determine whether a coding feature was required in the standard [11]. Thus, 5 dB is very substantial. At 25% packet loss, Mixed delivers a much higher quality video than UDP.

Most of Figure 6 shows a limited number of frames for each video clip, giving only a snapshot of the overall video quality. The average PSNR (from 10 runs) for the entire clip is shown in Figure 6(d). The standard deviation is below 0.2 dB. The difference in quality between UDP and Mixed increases as packet loss rate increases. Mixed Streaming shows an improvement of more than 1.5 dB over UDP at loss rates above 10%.

Figure 7 shows the percentage of frames that experience impairments and the magnitude of the impairments experienced for the Santana video (averaged over 10 runs) at 15% loss. The values on the x-axis are obtained by subtracting frame PSNR values of the original encoded video from those of a video with impairments. Figure 7 shows that Mixed Streaming delivers a higher number of video frames without any impairments than UDP (47% of frames to 21% of frames). UDP has a higher occurrence of frames with

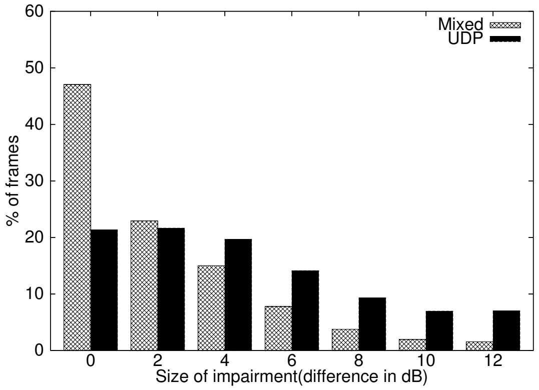


Figure 7. Impairment distribution (15% loss)

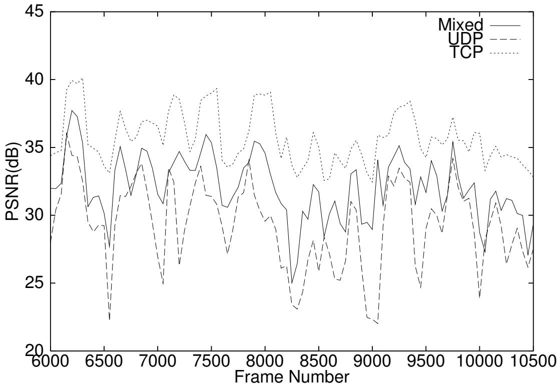


Figure 8. 15% loss w/ congestion

impairments greater than 4 dB. When the loss rate is increased, a similar pattern can be noticed, although the percentage of frames delivered without error is reduced. This complements our previous results, suggesting that recovery of only independent frames is beneficial for video data.

Further experiments were performed where we added background UDP and TCP traffic to the wired portion of the network to model congestion in the network. Figure 8 shows qualitatively similar results to those without congestion. The differences between Mixed Streaming and UDP are even more significant, showing that Mixed Streaming can improve the quality of the video even more if problems exist on the wired part of the network.

6 DELAY AND BUFFER SPACE

6.1 Required Start-Up Delay

The next experiments investigate the average start-up delay that is required to stream a video clip smoothly with TCP and Mixed Streaming respectively. UDP introduces negligible delay during transmission.

Figure 9 shows the start-up delay required for smooth display of the Santana video clip (average of 10 runs), as a fraction of clip length.[1] The network end-to-end propagation delay is 104 ms. To calculate the required start up delay, a video clip is transmitted over a network link with a specified packet error rate. An average delay of all the frames that miss their playback deadlines is obtained. The maximum delay amongst all packets is used as the start-up delay required. The y-axis is plotted using a log scale in order to cover the wide range of the required start-up delay.

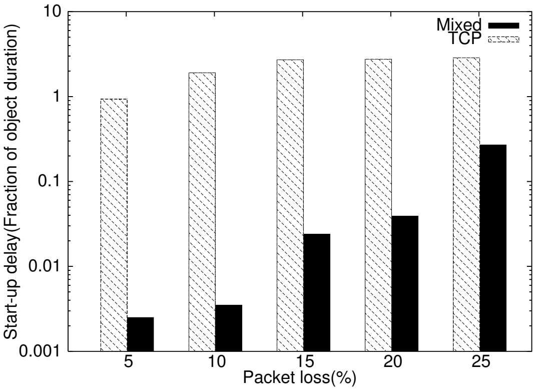


Figure 9. Required start-up delay

When packet loss is present, TCP requires a very large start-up delay. TCP requires a start-up delay that is more than the duration of the clip when the packet loss rate is greater than 5%. This behavior is similar for all the three video clips with different activity levels. The reason for this is because of TCP's error recovery algorithm that retransmits lost packets and attempts to deliver packets in order.

Successive packets will be delayed and will result in high start-up delay. In Mixed Streaming, the effect of TCP's error control algorithm is reduced because it is only limited to I-frames (about 39% of video file size with a 30 frame GOP). The remaining UDP frames are not affected at all.

Mixed Streaming has an obvious advantage over TCP streaming in terms of the start-up delay. It is able to reduce the start-up delay between 1 and 2 orders of magnitude in high loss environments. For instance, in Figure 9, a client streaming a 10 minute clip with TCP at 5% packet loss will have to wait for more than 10 minutes to view a smooth video. For Mixed Streaming, they will have to only wait for about 30 seconds. This suggests that Mixed Streaming could be useful in a wireless environment with high losses.

Krasic *et al.* [14] proposed reducing the data transmitted by dropping packets when the network network is congested. Sending less data reduces or even stops the congestion thereby reducing the number of lost packets and the required start-up delay. This approach is unlikely to re-

[1] Users tolerance of delay is somewhat proportional to clip length.

duce the start-up delay on a wireless network, where packet losses are mostly caused by channel conditions. A remaining question is how Mixed Streaming will perform in regard to the delivered quality when the start-up delay is limited.

Another important issue associated with the start-up delay is the buffer space, since packets have to be stored before playback. Table 2 shows an estimate of the buffer size needed to transmit a video file using TCP and Mixed Streaming respectively for Santana. This estimate is obtained by multiplying the required start-up delay by streaming rate. The number of frames is calculated by dividing the estimated required buffer size by the average frame size.

We see that as the packet loss rate increases, TCP needs a much larger buffer size than Mixed Streaming to accommodate delayed packets. For instance, when the packet loss rate is over 15%, TCP needs to buffer the entire file. Mixed Streaming would be useful for streaming video to small devices that have low storage capabilities.

Table 2. Estimated buffer requirements

	TCP		
	Buffer	**% of File**	**Frames**
5%	10.8 Mb	54	11335
10%	14.2 Mb	71	14904
15%	18.9 Mb	100	19858
20%	18.9 Mb	100	19858
25%	18.9 Mb	100	19858

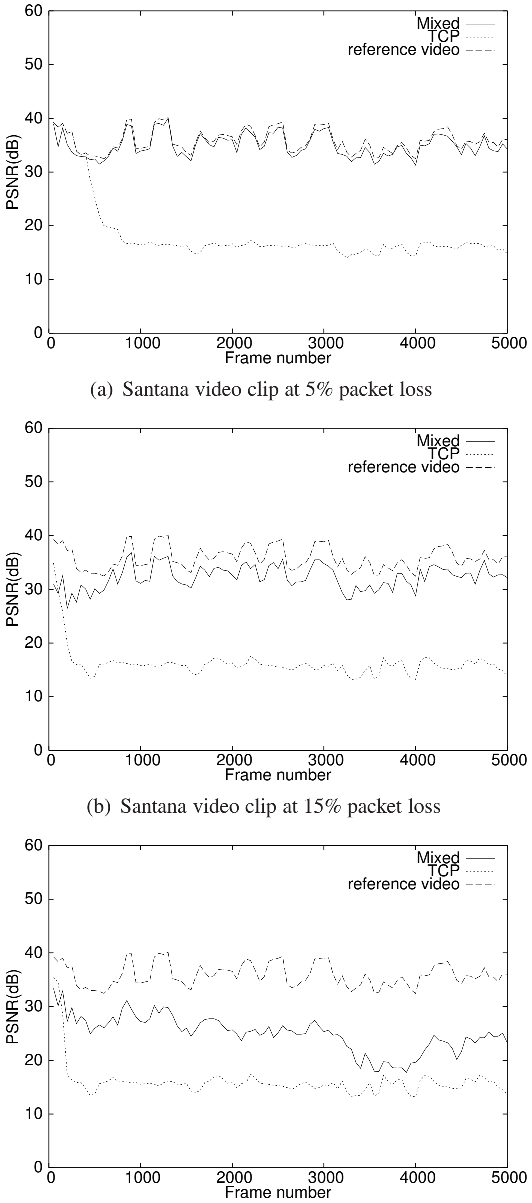
	Mixed		
	Buffer	**% of File**	**Frames**
5%	0.169 Mb	0.85	177
10%	0.460 Mb	2	428
15%	0.508 Mb	2.56	533
20%	0.702 Mb	3.54	736
25%	4.638 Mb	23.4	4859

6.2 Limited Start-Up Delay

The next set of experiments examine the video quality delivered by TCP streaming and Mixed Streaming when the start-up delay (and therefore, buffer size) is limited. The buffering delay is set to 30 seconds, since this value is used in popular media players, such as the RealNetworks Media Player[2] and Windows Media Player[3] during congestion.

Figure 10 shows the video quality delivered by Mixed Streaming and TCP, respectively, with limited start-up delay, no congestion, and the same display configurations as in the previous section. The reference video represents a sequence transmitted without loss or delay.

TCP streaming delivers the highest quality at the start of playback. As streaming continues, however, the video quality delivered by TCP degrades by more than 10 dB on average. No frames make their play-out deadline, so the difference between the expected frame and the frame available is huge. For the Mixed Streaming approach, at the very beginning of the clip, the quality is less than that delivered by TCP, but remains stable thereafter at low loss rates.



(a) Santana video clip at 5% packet loss

(b) Santana video clip at 15% packet loss

(c) Santana video clip at 25% packet loss

Figure 10. 30 sec start-up delay Video Quality

TCP streaming is unable to keep up with retransmissions during the latter parts of the clip. The 30 second start-up

[2] http://home.real.com/product/help/rp10v8_ts/en/Pref_Playback_Settings.htm

[3] http://support.microsoft.com/kb/843509#16

delay is not enough to mitigate these delays. Experimenting with a 50 second start-up delay showed the same pattern, but it took a bit longer before TCP's performance degraded.

Figure 10(c) shows the video quality received at 25% loss. Mixed Streaming shows a rapid decline in quality. The data loss rate is too high for I-frames to be delivered successfully. We see that TCP still has a higher PSNR during the initial parts of the video. This means that TCP is attractive for streaming very short video clips because only a very small amount of data needs to be retransmitted, but the relative delay to the length of the video is still quite large. A 30 second start-up delay is not sufficient in either scheme when the packet loss rate is 25%.

7 CONCLUSIONS AND FUTURE WORK

This paper presents the concept of Mixed Streaming and its performance over wireless networks. To measure the performance, Mixed Streaming was compared with common streaming techniques, namely UDP streaming (*true streaming*) and TCP streaming (*progressive download*).

In general, Mixed Streaming delivers video quality that is between TCP and UDP and degrades more gracefully than UDP in poor channel conditions with high packet loss. With a limited start-up delay, Mixed Streaming delivered better quality than TCP in the presence of wireless errors. At very high error rates ($> 25\%$) the performance gain offered by Mixed Streaming begins to reduce. Results also showed that Mixed Streaming requires a small startup delay, while TCP requires a much larger start up delay.

The experiments in this study were carried out using simulation in a limited number of network configurations and conditions. We showed that performance improves under moderately high and unpredictable loss behavior. The GOP used in the experiments reduced the relative amount of data sent reliably and a more comprehensive comparison of the encoding parameters could provide tighter bounds on when Mixed Streaming becomes the best alternative. The overhead required to reconstruct and display the stream at the wireless client has also not been quantified.

It would be interesting to see how Mixed Streaming behaves on a real network. While we introduced background traffic in some of the experiments, we did not determine how friendly Mixed Streaming is to other network flows. The discussion of these issues is left as future work.

References

[1] M. Allman and A. Falk. On the Effective Evaluation of TCP. *SIGCOMM Comp. Comm. Review*, 29(5):59–70, 1999.

[2] G. AlRegib and Y. Altunbasak. 3TP: 3-D Models Transport Protocol. In *9th Inter. Conf. on 3D Web Technology*, pages 155–163, Monterey, CA, Apr. 2004.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. Katz. A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. *IEEE/ACM Trans. on Networking*, 5(6):756–769, 1997.

[4] J. Bolot. End-to-End Packet Delay and Loss Behavior in the Internet. In *ACM SIGCOMM '93*, pages 289–298, San Francisco, CA, 1993.

[5] J. C. Bolot and T. Turletti. Adaptive Error Control for Packet Video in the Internet. In *ICIP 1996*, pages 25–33, Lausanne, Switzerland, 1996.

[6] J. M. Boyce and R. D. Gaglianello. Packet Loss Effects on MPEG Video Sent Over the Public Internet. In *ACM Multimedia 1998*, pages 181–190, Bristol, UK, 1998.

[7] M. Chen and A. Zakhor. Rate Control for Streaming Video over Wireless. In *Infocom 2004*, pages 1181–1190, Hong Kong, China, 2004.

[8] F. H. P. Fitzek and M. Reisslein. Wireless Video Streaming with TCP and Simultaneous MAC Packet Transmission (SMPT): Research Articles. *International Journal of Communication Systems*, 17(5):421–435, 2004.

[9] E. N. Gilbert. Capacity of a Burst Noise Channel. *Bell Systems Technical Journal*, 39(9):1253–1265, 1960.

[10] L. Guo, S. Chen, Z. Xiao, and X. Zhang. Analysis of Multimedia Workloads with Implications for Internet Streaming. In *WWW 2005*, pages 519–528, Chiba, Japan, 2005.

[11] M. Hairuo and M. E. Zarki. Broadcast/Multicast MPEG-2 Video over Broadband Fixed Wireless Access Networks. *IEEE Network*, 12(6):80–93, 1998.

[12] L. Huang, U. Horn, F. Hartung, and M. Kampmann. Proxy-based TCP-Friendly Streaming over Mobile Networks. In *WoWMoM 2002*, pages 17–24, Atlanta, GA, 2002.

[13] J. Klaue, B. Rathke, and A. Wolisz. Evalvid - A Framework for Video Transmission and Quality Evaluation. In *Comp. Perf. Eval.*, pages 255–272, Urbana, IL, 2003.

[14] C. Krasic, J. Walpole, and W. Feng. Quality-adaptive Media Streaming by Priority Drop. In *NOSSDAV 2003*, pages 112–121, Monterey, CA, 2003.

[15] T. Kuang and C. L. Williamson. Hierarchical Analysis of RealMedia Streaming Traffic on an IEEE 802.11b Wireless LAN. *Computer Communications*, 27(6):538–548, 2004.

[16] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, Boston, MA, 2001.

[17] R. Mantiuk, G. Krawczyk, K. Myszkowski, and H. Seidel. Perception-Motivated High Dynamic Range Video Encoding. *ACM Transactions on Graphics*, 23(3):733–741, 2004.

[18] A. Singh, A. Konrad, and A. D. Joseph. Performance Evaluation of UDP Lite for Cellular Video. In *NOSSDAV 2001*, pages 117–124, Port Jefferson, NY, 2001.

[19] A. Tripathi and M. Claypool. Improving Multimedia Streaming with Content-Aware Video Scaling. In *6th Joint Conference on Information Science*, pages 1021–1024, Research Triangle Park, NC, 2002.

[20] B. W. Wah, X. Su, and D. Lin. A Survey of Error Concealment Schemes for Realtime Audio and Video Transmission over the Internet. In *Intl. Symposium on Multimedia Software Engineering*, pages 17–24, Taipei, Taiwan, 2000.

[21] S. Wolf and M. Pinson. Video Quality Measurement Techniques. Technical Report TR-02-392, U.S. Department of Commerce, NTIA, 2002.