Packet Loss Effects on MPEG Video Sent Over the Public Internet

Jill M. Boyce Robert D. Gaglianello

Bell Laboratories, Lucent Technologies
101 Crawfords Corner Rd
Holmel, NJ 07733
{jillb, rdg}@lucent.com

1. ABSTRACT

This paper presents results from a study of streaming MPEG compressed video over the public Internet, using the RTP and UDP transport protocols. Two five minute video clips were MPEG coded at rates of 384 Kbps and 1 Mbps. The resultant coded streams were transmitted at their respective data rates between four sites in the United States and Europe. Measurements were taken between sites during all hours of the day for several weeks at a time to generate a clear picture of the time varying nature of Internet errors. The paper concentrates on network loss/error characteristics that specifically affect the quality of the received MPEG compressed streams. Due to the nature of MPEG data streams, losses in certain parts of the data stream are more disturbing when viewed than losses in other parts of the data stream. For instance, since MPEG video is inter-frame coded, artifacts due to network loss/errors, can persist for many frames. Thus, a meaningful measure of received video quality requires a more thorough analysis of network errors than average error rates. Our results include patterns of packet loss over time, conditional packet loss probabilities, packet loss with respect to packet size, out of order packet reception, and persistence of packet errors in MPEG video. By examining the error characteristics exhibited in real heterogeneous networks such as the public Internet, better error recovery and concealment techniques can be developed.

ACM Multimedia'98, Bristol, UK
© 1998 ACM 1-58113-036-8/98/0008 $5.00

1.1 Keywords

MPEG, Streaming Video, Internet packet loss

2. INTRODUCTION

New Internet connection technologies such as ADSL modems, cable modems, and T1 lines, coupled with the ubiquity of the public Internet have increased the interest in developing high-quality Internet-based multimedia applications such as multipoint video teleconferencing, distance learning, and telemedicine. Key to the success of such applications is the quality of the received video and audio. Network errors require the use of concealment techniques to provide the video and audio quality necessary for the above applications to be widely accepted and used. Effective concealment techniques must be driven by the types of network errors and by the artifacts these errors produce in the reconstructed multimedia data streams. The goal of this paper is to study the effects of network errors encountered when using the public Internet for the transmission of MPEG coded video streams. We concentrate on video because it places the greatest demand on the network in terms of bandwidth, requirements for low delay, and susceptibility to errors. We present experimental results of transmission of MPEG compressed video over the public Internet, at 384 Kbps and 1 Mbps, with over 1000 samples from four geographically distributed sites.

Several international video compression standards exist, MPEG-1/2 [7], H.261/3, Motion-JPEG (MJPEG) [5], as well as several proprietary streaming video solutions, such as Real Video, Vivo Active and VDO. We have chosen to study MPEG-1 compressed streams because MPEG-1 is an open standard and real-time hardware encoders and decoders can be purchased at a reasonable price. MPEG-1 also offers the best video quality for the bandwidth range (384 Kbps – 3.0 Mbps). This range is important because the new Internet connection technologies mentioned above function in this range.

Two types of packet transport protocols can be used for sending the compressed video over the Internet, the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP, a reliable protocol, guarantees delivery of all packets and in order, while UDP does not guarantee delivery of packets or the ordering of received packets [4]. UDP is the protocol studied in this paper because it is more suitable for real-time applications such as interactive video conferencing since it has lower overhead and lower delay than TCP transport. Also, the retransmission nature of TCP is unsuitable for real-time data streams, such as MPEG video. It is the low-delay characteristic of UDP that makes it suitable for real-time applications. However its unreliable nature requires that care must be taken to conceal the errors that are introduced due to packet losses.

Concealment techniques must be designed to take into account both network error characteristics and the type of coding scheme. Some video compression standards, such as MJPEG, use only intra-frame coding, so errors in one frame do not propagate to other frames and can be concealed with little noticeable effect to a viewer. Other standards, such as MPEG and H.263 also use inter-frame coding, which can improve the compression efficiency. However, if a packet loss occurs in an MPEG video, the effect is much more noticeable, because it can persist for many frames. For example, consider an MPEG video sequence with I frames every 15 frames. If an error occurs while transmitting the I frame, the effect persists for 15 frames, or 500 msec, which is quite noticeable to a viewer. Therefore, the error characteristics of the network and transport protocol over which the video is sent has a large effect on the quality of the received video.

This paper gathers information in an attempt to gain an understanding of how the public Internet differs from an ideal network, and what effect those differences have on the streaming of an MPEG video stream. Packet loss is studied in several ways – absolute packet loss, conditional packet loss, and packet loss rate over time. Out of order packet delivery, and the persistence of packet loss effects in MPEG video are also discussed. Video error concealment techniques for displaying the lossy video bitstream with the best possible video quality are beyond the scope of this paper, but a thorough overview can be found in [13]. It is hoped, however, that the results of this paper will aid in developing ways to improve the reconstructed video both through error concealment techniques that are applied at the decoder and by error resilience techniques that are applied at the encoder.

Section 3 describes the data gathering methods we employed and the characteristics of the data streams used in this study. Section 4 presents the results and accompanying analysis. In section 5 we present conclusions and ideas for future work.

3. DESCRIPTION OF DATA GATHERING METHODS

For this study we chose three sites, the Gertrude Stein Repertory Theater in New York City, New York, the University of Texas in Austin, Texas, and the Queen Mary & Westfield College in London, England. Each site was connected to the public Internet and streamed data to Bell Labs in Holmdel, New Jersey where the error statistics were gathered. All sites had connections to the Internet with at least a 1.5 Mbps bandwidth, and used PC's running Windows NT 4.0. Two five-minute long MPEG-1 elementary video streams with data rates of 384 kbps (14.4 Mbytes) and 1 Mbps (37.6 Mbytes) were sent, using the Real-Time Protocol (RTP) [11] on top of the User Datagram Protocol (UDP). Other studies such as Paxson [9] contain more sites (35), but Paxson's results were for bulk file transfers of 100 Kbytes using TCP transport. The higher data rates of video transmission studied in this paper differentiate this work from the analysis of audio transmission over the Internet by Bolot [2].

For this study, the 384 kbps and 1 Mbps streams were each sent sequentially once per hour over the test period, with at least 100 samples for each data rate and path combination. Over 1000 total samples were sent in all, in December 1997, and January and February 1998. The large size and long time duration of the data streams allows us to observe effects within the time period as well as between time periods. As the state of the network changes often, average readings over time are not as useful as are the "instantaneous" readings gathered in these experiments. Both MPEG-1 video streams were approximately 5 minutes long, were coded at 30 frames per second, and were coded from live broadcasts of CNN. The two clips were of different material, but each contained a mix of news broadcasts and commercials, with many scene changes, and periods of slow moving and fast moving content. An Optibase MovieMaker board was used for the encoding. The MPEG parameters were set for I frames every 15 frames (N=15), and 2 B frames between anchor frames, or an anchor frame every 3 frames (M=3), for a GOP structure of IBBPBBPBBPBBPBB. The 384 kbps sequence used QSIF (176 x 112 pixels), and the 1 Mbps sequence used SIF (352 x 240 pixels).

For the SIF sequence, scene change detection with I frame insertion was used. Unfortunately the Optibase

Table 1. Statistics for 1 Mbps Video

	Frames		Bytes		Packets	
	number	percent	number	percent	number	percent
I	628	6.96%	10142196	26.23%	9420	18.42%
P	2381	26.38%	16364412	42.32%	35715	69.82%
B	6018	66.67%	12164820	31.46%	6018	11.76%
Total	9027		38671428		51153	

Table 2. Statistics for 384 kbps Video

	Frames		Bytes		Packets	
	number	percent	number	percent	number	percent
I	603	6.68%	4163464	27.75%	4221	15.59%
P	2405	26.64%	5941948	39.60%	16835	62.19%
B	6014	66.62%	4900632	32.66%	6014	22.22%
Total	9022		15006044		27070	

MovieMaker board did not offer this feature for QSIF. Therefore, in the SIF sequence, the actual distance between I frames varied based on the content, but never exceeded 15 and the QSIF sequence distance was fixed at 15. In the SIF sequence there were 628 I frames in the 9027 frame sequence, for an average value of N = 14.37.

The MPEG data was packed into variable sized RTP packets, sent over UDP. The recommended header from the Internet Engineering Task Force RFC "RTP Payload Format for MPEG1/MPEG2 Video" [6] was used. Each packet contained a 20 byte header, which was made up of 16 bytes for the general RTP header, and a 4 byte MPEG specific header. I and P frames were packed into RTP packets such that each slice was in its own packet. The slice size in the video encoder was set to equal one entire row of macroblocks. There were 22 macroblocks per slice for the SIF sequence and 11 macroblocks per slice for the QSIF sequence. Sequence headers and GOP headers were included in the same packet as the first slice of the I frame which they preceded. Picture headers were included in the same packet as the first slice of the frame they preceded. Each entire B frame was packed into a single RTP packet. The rationale for this choice is discussed in Section 4.3. For more information on the statistics of the MPEG files, see Tables 1 and 2.

In trying to meet the target bit rate for transmission of the streaming video, 384 kbps or 1Mbps, after each packet was sent the program evaluated the total number of bytes that had been sent, and the total time that had elapsed. For measurement purposes, we only counted the bitrate of the MPEG-1 video and did not include the overhead for RTP and IP. If the data rate was ahead of schedule, the program waited for an appropriate period of time before sending the next packet. The program used a Windows NT time function that was accurate to approximately 15 milliseconds.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Average Packet Loss

The simplest measure of network performance is average packet loss rate. The average loss rates for all paths and data rates are shown in Table 3 below. Because the packets were of varying sizes rather than of fixed size, the data byte loss rate differs slightly from the packet loss rate. The effect of packet size on the packet loss rate is discussed in Section C.

Table 3. Average Packet and Data Loss Rates

Path	Average Packet Loss	Average Data Loss
New York 1 Mbps	7.123%	6.625%
New York 384 kbps	3.031%	3.308%
Texas 1 Mbps	12.654%	13.505%
Texas 384 kbps	9.364%	9.543%
London 1 Mbps	5.694%	5.581%
London 384 kbps	5.236%	5.144%

It was also observed, as expected, that the average packet loss rates were higher for the 1 Mbps case than for the 384 kbps case. The experiments were done using UDP, an unreliable transport method, which does not reduce the transmitted data rate in the presence of network congestion, as is done for TCP. It would be helpful in an actual system to provide a means for backing off the data rate based on presence of network congestion such as the Streaming Control Protocol (SCP) described in [3], or by using information available in the Real Time Control Protocol [12].

4.2 Packet Loss versus Time

Another simple network performance measure is packet loss rate versus time. The packet loss rates of the various connections change with time of day and with the day of the week. Unsurprisingly, higher loss rates were observed during business hours and lower loss rates in the middle of the night. As an illustrative example, Figure 1 shows one week's worth of data for the SIF 1 Mbps transmission from New York City to Holmdel. Figure 2 enlarges a single day's worth of data from Figure 1, to better illustrate the effect of time of day. The packet loss rates ranged from 0.018% to 100% for the 5 minute samples, a result that is hidden if only the average packet loss rate of 7.123% is considered. Each sample point on the graphs is the average packet loss rate for the transmission of each five minute long MPEG stream.

Figures 3 and 4 show the cumulative distribution of the packet loss rates for all paths and data rates. There are many sample points with close to zero packet loss rates, but as Figure 1 indicates, many of those occur in the middle of the night, when there is virtually no other network traffic. More interesting are the packet loss rates up to about 25%, which accounts for 98.8% of the New York 384 kbps samples and for 86.6% of the Texas 1 Mbps samples. These are the packet loss rates where the use of error resilience and error concealment techniques is especially necessary.

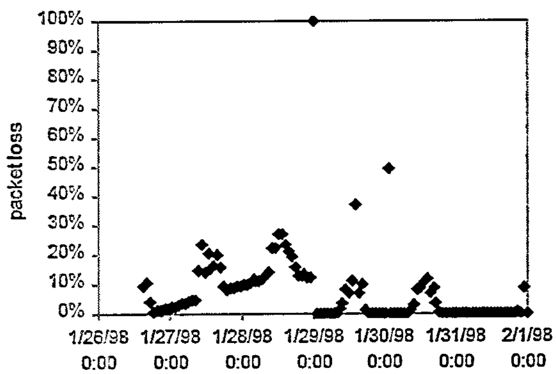

Figure 1. New York 1 Mbps Packet loss vs. time

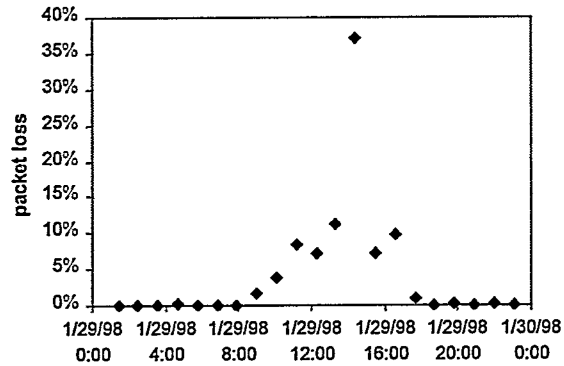

Figure 2. Expansion of Figure 1

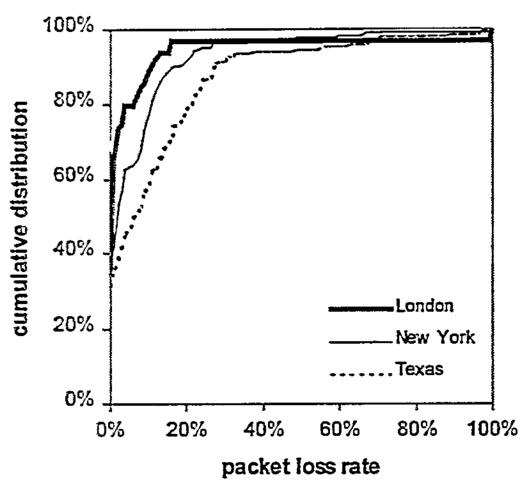

Figure 3. Cumulative packet loss distribution for 1 Mbps

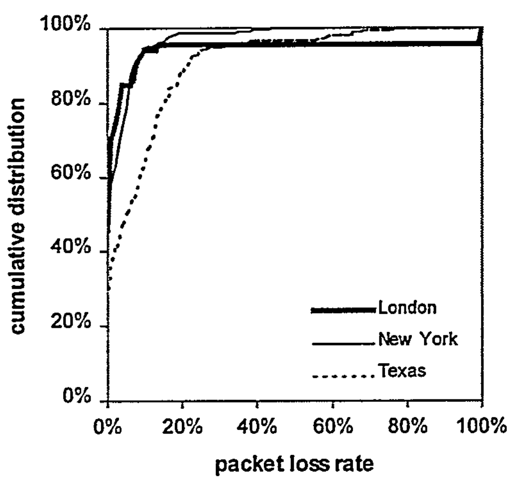

Figure 4. Cumulative packet loss distribution for 384 kbps

4.3 Packet Loss versus Packet Size and Algorithms for Filling IP Packets with MPEG Data

The Internet Engineering Task Force's *RFC RTP Payload Format for MPEG1/MPEG2 Video*[6] provides information about how MPEG data can be arranged into IP packets. It allows for the possibilities that multiple slices may be placed in a single packet, a single slice may fill a single packet, or a single slice may be split into multiple packets. How MPEG slices are arranged into IP packets affects the packet sizes, and packet sizes have an effect on packet loss rates, as seen in Table 3 where the average packet loss rates and data byte loss rates differed slightly because variable sized packets were used in the experiments. This section discusses the experimental results of the relationship between packet size and packet loss rates, and suggests a method for filling IP packets with MPEG data.

In general, a transmitted IP packet is either received entirely or lost entirely. However, packets larger than the Ethernet Maximum Transfer Unit (MTU) size of 1500 bytes may be divided into fragments [4]. If any of the fragments is lost during the network transmission then the entire packet is lost. It is therefore expected that packets larger than the MTU size will have a higher packet loss rate than packets smaller than the MTU size.

At a given bitrate, smaller packet sizes translate into higher packet transmission rates. This in turn affects the packet loss rate. As shown in [8] and [12], Internet congestion is affected not only by bit rate capacity constraints, but also by access constraints. An access constrained network node is a node whose performance is more sensitive to the number of packets it must handle than to the number of bits, i.e., a router queue.

In MPEG, predictive coding is used both within frames and between frames. Within I frames, the DC coefficients in a slice are predictively coded. Within P and B frames, motion vectors within a slice are predictively coded. The loss of a single macroblock makes the rest of the data in the slice unusable. Also, because MPEG video is predictively coded between frames, the loss of data in I and P frames will propagate and cause errors in later frames until a new I frame arrives.

For our experiments, we chose to place entire B frames into a single packet, because B frame slices are small, and it would be inefficient to pack B frame slices individually. We also placed individual slices from I and P frames into packets. Figures 5 and 6 show distributions of the packet sizes, within 64 byte ranges, for the two MPEG files. The packet sizes include the 20 byte RTP header, and are shown separately for I, P and B frames.

These choices were made, in part, to provide a range of slice sizes over which to gather data, in order to explore the relationship between packet size and packet loss rate. The packet loss rate versus packet size relationship for the various paths and the two data rates are shown in Figures 7 and 8. Each point on the graphs is for a 128 byte range of packet sizes.

The magnitude of the loss rates differed for the different paths and data rates. However, for all of the paths and data rates, there was a jump in the loss rate at the MTU size of 1500, and again at size 3000. The increase in the packet loss rate at the MTU size is expected because the packet will be divided into two fragments, either of which may be lost, and if either fragment is lost the entire packet is lost. The loss rate increase for packets greater than one MTU size was less than a doubling of the loss rate in all cases.

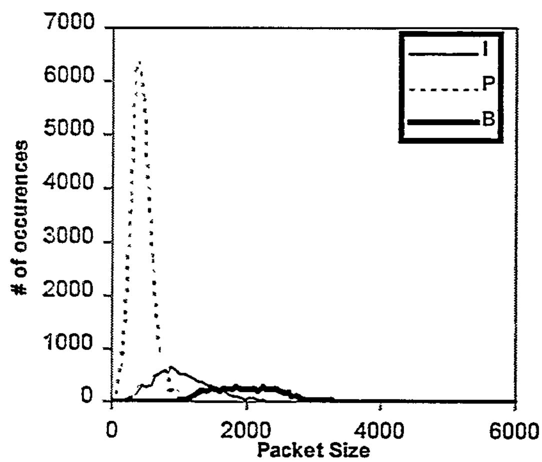

Figure 5. 1Mbps Packet size distribution

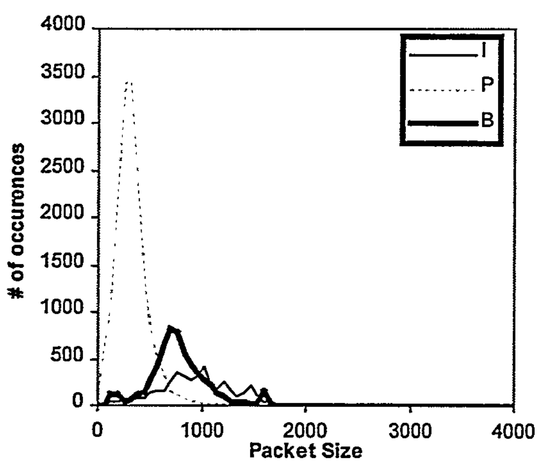

Figure 6. 384 kbps Packet size distribution

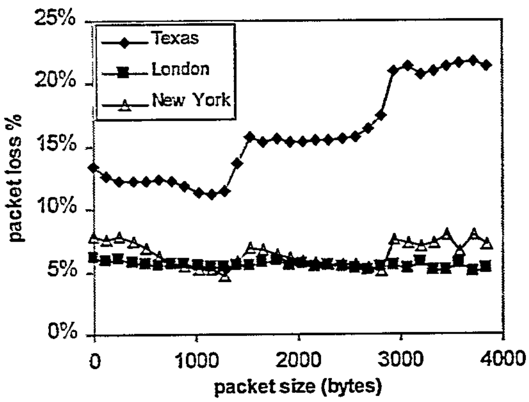

Figure 7. 1 Mbps Packet size vs. loss

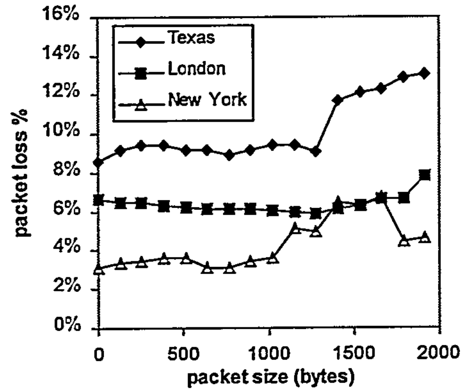

Figure 8. 384 kbps Packet size vs. loss

For packets that ranged in size up to 1500 bytes however, there were different characteristics for the two data rates. For the 384 kbps case, the packet loss rate was fairly flat over the size range below 1500 bytes. It is difficult to comment on the shape of the loss rate curve over 1500 bytes because, as seen in Figure 5, there are relatively few packets of that size in the 384 kbps case, so the results are not statistically significant.

For the 1 Mbps case, however, the loss rate curve shows the packet loss rate to be higher for smaller packets than for larger packets in the range below 1500 bytes. A possible explanation for this is that the test programs maintained transmission at a constant data rate, regardless of packet size, so if several small packets were sent in a row, more packets were sent within a given time period, increasing the packet rate. Because of the high packet rate, a router queue in the path may have become access constrained and had to drop packets. This condition had less of an effect on the 384 kbps QSIF case because the packet rate was always less than for the SIF case.

These experimental results led to the development of a method for filling the MPEG data into IP packets. Because the beginning of an MPEG slice must be available to decode the end of a slice, dividing a slice into multiple packets (or multiple fragments) increases the risk that the slice will be lost. The RTP and UDP/IP overhead of 20 bytes per packet makes sending small packets inefficient. Because of these reasons, it is better to place an entire slice in a packet instead of dividing a slice into multiple packets. Even for the case when a slice is larger than an MTU, because the loss rate increase is less than doubled, it is appropriate to fit the entire slice into a single packet, and allow the network to fragment it if necessary.

Fitting multiple slices into a single packet will be appropriate in some conditions and not in others. When the packet loss rate is higher for smaller packets than larger packets in the 0-1500 byte range, as in the 1 Mbps cases, packets should be filled with as many slices as would fit into an MTU. However, when the packet loss rate is the same for all packet sizes in the 0-1500 byte range, each slice should be placed into its own packet. In this case, the slice loss rate will be identical if two slices were in separate packets or in the same packet, but separating the slices into separate packets increases the chances that the missing slices will not be neighbors of one another. This would improve the ability of spatial error concealment techniques to conceal the slice errors.

An alternative to this method is for a video encoder to start a new slice whenever the MTU fills, as MPEG syntax allows. However, we are unaware of any commercially available MPEG encoders that allow this option.

4.4 Persistence of packet loss effect in MPEG frames

In MPEG compressed video, there are three frame types: Intra-coded (I) frames, Predictively coded (P) frames, and Bi-directionally coded (B) frames. Packet losses that cause errors in I frames and P frames will be propagated until the next I frame. Errors in B frames do not propagate because B frames are not used in predictions of other frames.

Much of the literature in error concealment in the presence of packet loss is for ATM cell losses. ATM cells are small compared with IP packets – if ATM cell are lost, only one or a few macroblocks are lost [1]. This leads to different types of concealment techniques than the losses experienced for video over IP as in this experiment, where an entire slice or an entire frame is lost. For the lost B frames, the simplest method of concealment is to replay a neighboring frame. For lost slices in I and P frames, the type of concealment to use is less obvious.

It is useful to have a measurement of how the IP packet loss affects the quality of the viewed video. SNR is an obvious measure of video quality, but to measure SNR accurately an error concealment method must be used. Developing new error concealment techniques or comparing the quality of existing schemes is beyond the scope of this paper.

In this paper, instead of using SNR we define a frame error state measure. For each received frame, it is determined whether or not a lost packet affects that frame. The lost packet could have been from the current frame, or from a frame that the current frame is predicted from. The frame error state measure has two levels – in error or not in error – no additional weight is given to multiple errors in a frame.

The experimental results for error state measurement may be slightly overstated since the measure was based only on the frame types. All P frames after an I frame in error would also be considered in error, until the arrival of the next I frame, rather than attempting to confirm through use of motion vectors and macroblock coding types that the earlier frame error actually did affect the following frame. We believe the effect of this simplification on the accuracy of the results to be small as entire slices are lost when an IP packet is lost, rather than individual macroblocks, and it seems uncommon that an entire slice would not be used in the next frame's prediction. Also, scene change detection with I frame insertion was used in the encoding, so it is not expected that a P frame would have large numbers of intra coded blocks, which would limit the persistance of an error.

Figure 9 shows the relationship between the packet loss rate and the frame error rate for the New York City 1 Mbps case. All of the experimental paths and data rates had very similar relationships. Each sample point on the graph is the average for a 5 minute MPEG file transmission.

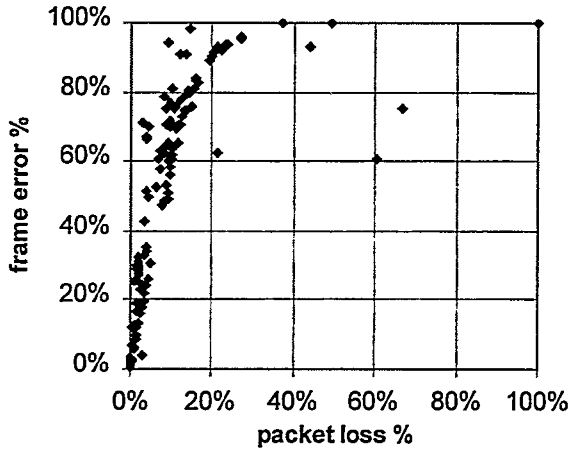

Figure 9. Packet loss effect on frame error rate

This frame error state measure indicates the difficulty of sending MPEG over a lossy network. Small packet loss rates translate into much higher frame error rates, for example a 3% packet loss percentage could translate into a 30% frame error. There are a few outlying points on the graph where the frame error rate is not dramatically higher than the packet loss rate, for example the point where a 60% packet loss causes a 61% frame error rate. These outlying cases occur when the packet losses were not distributed over the 5 minute sample period but instead were concentrated within a portion of the 5 minute period.

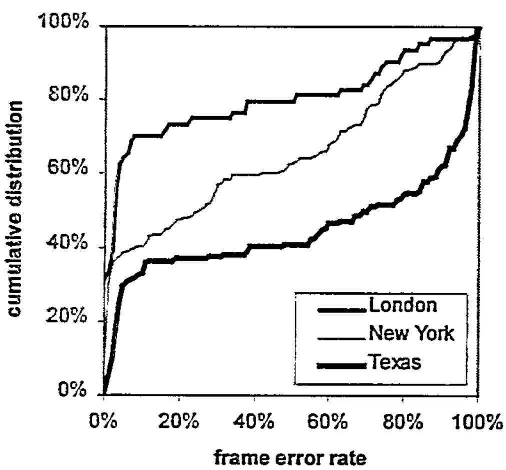

Figure 10. Cumulative frame error rate distribution for 1 Mpbs

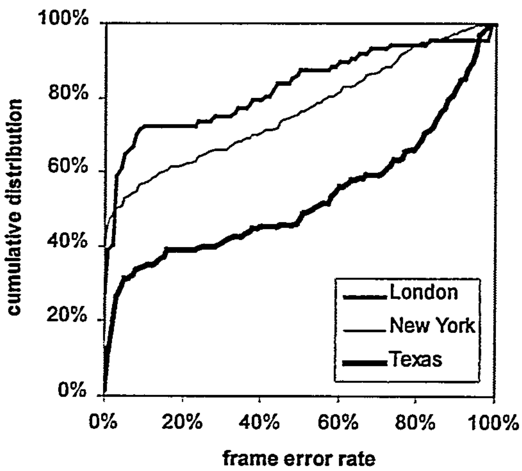

Figure 11. Cumulative frame error rate distribution for 384 kbps

Figures 10 and 11 show the cumulative distributions of the frame error rate for all paths and data rates, analogous to the packet loss distributions in Figures 3 and 4. Frame error rates of less than 10% occurred in only 33% of the Texas 1 Mbps samples, and in 72% of the London 384 kbps samples.

4.5 Conditional Probability of packet loss

Packet loss is not independently distributed; instead, packet losses tend to occur in bursts. One way to measure burst packet losses is to count the number of consecutive packet losses. However, a situation where every other packet is lost has high packet loss correlation but will only have burst lengths of one. So instead of burst length we compute a measure of conditional probability, which is the probability that packet n+k is lost, given than packet n was lost, as in [10].

Figure 12 shows conditional probability curves for several samples with overall packet loss rates ranging from 1.5% to 21.1%. Each curve is for data within a 5-minute transmission sample, rather than between the samples, because of the one-hour gap between samples.

The conditional probability curve shapes were related to the overall packet loss rates of that sample. The curves tended to be most steep for low overall loss rates and relatively flat for higher overall loss rates. This means that packet losses were more correlated for low overall packet loss rates than for high packet loss rates. All of the curves had the highest conditional probability correlation at one packet distance, and the conditional probabilities generally gradually decreased towards rates similar to their overall average loss rates at greater packet distances. Even in the steepest curves, however, the conditional probability at a distance of one was less than 50%, indicating that a majority of packet losses had a burst length of one, as seen in [2].

4.6 Out of Order Packets

Because of changes in routing paths, UDP packets may arrive in a different order than that in which they were sent. Table 4 indicates the average percentage of all sent packets that either arrived out of order, never arrived, or arrived in order. Consider an example where packets arrive in the order: 1 2 4 3 5. Packet number 3 is considered to have been received out of order.

Table 4. Out of Order Packet Rates

Path	Out of Order Packet Rate	Lost Packet Rate	In Order Arrivals Packet Rate
NYC SIF	6.095%	7.123%	85.93%
NYC QSIF	2.215%	3.031%	94.75%
Texas SIF	15.416%	12.654%	71.93%
Texas QSIF	10.005%	9.364%	80.63%
London SIF	3.252%	5.692%	91.06%
London QSIF	1.746%	5.726%	93.02%

Packets that arrive out of order can reduce the quality of the received video, if means are not taken to re-order the packets before decoding. Re-ordering the packets requires buffering the data prior to decoding. The packet loss rates discussed in Sections A-D were calculated assuming an infinite packet buffer – a packet was counted as received if it ever arrived. But for a data packet to be useful, it has to arrive in time for it to be decoded and displayed. A decoder buffer can be used to allow time for late packet arrivals, but using a buffer adds to the delay in viewing the video, which is a problem for certain applications. The choice of the size of the buffer has a great effect on the received video quality. If a buffer size

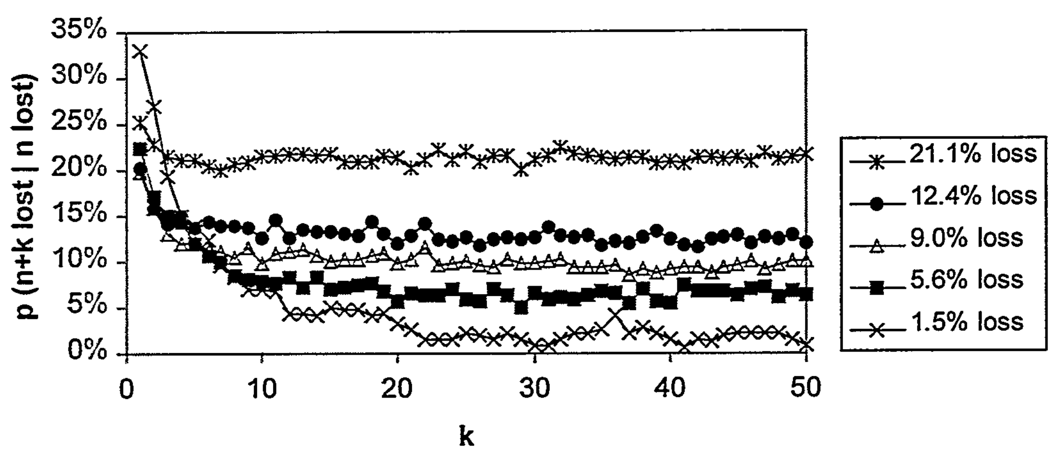

Figure 12. Conditional Probabilities of Packet Loss

is chosen that is too small to adequately deal with late packets, a late packet effectively becomes a lost packet. Effectively losing packets in this manner has the same dramatic effect on video frame error rates as seen for lost packets in section D.

Figures 13 and 14 show the percentage of all sent packets that were received out of order, with various packet number delays. These indicate how large a packet buffer is needed to avoid losing that percentage of packets. As shown in the figures, a majority of the out of order delays are only by a single packet. A single packet buffer could reduce the number of late packets due to out of order arrival in the Texas 384 kbps case from 10.0% of all transmitted packets to 4.7%. A five packet buffer could reduce it to 0.32%. Network jitter, or varying delays in packet arrival times, even if they do not arrive out of order, would also effect the buffer sizes and delays, but is not addressed in this paper.

Like the lost packet rate, the out of order packet rate varied with time of day. During the middle of the night, there were almost no out of order packets. There was no direct correlation between loss packet rates and out of order packet rates however. They both tended to be near zero at the same time, but they did not have peaks at the same times. When the packet loss rate was 100%, there were no out of order packets received, as no packets were received. One would expect lost packets when the network was overloaded with data. One would expect packet route changes, and hence out of order packets, when there is a variable level of other activity on the network that would affect the calculation of shortest routing path. The trace route path from New York to Holmdel, NJ had 13 hops, from Texas it varied between 21 and 22 hops, and from London it had 18 hops.

5. CONCLUSIONS

We have presented experimental results of transmission of MPEG compressed video over the public Internet, at 384 Kbps and 1 Mbps, with over 1000 samples from four geographically distributed sites. We studied how IP transmission differs from an ideal network, and how those differences affect received video quality. In future work we will use the experimental results to design systems with error concealment and improved error resilience techniques for better reconstructed MPEG video quality.

Packet loss rates were presented in several forms. Average packet loss rates varied from 3.0% to 13.5% for the three paths and two data rates. However packet loss rates over the 5-minute samples varied greatly from the average rates over time, with loss rates from 0% to 100%, with greater loss rates during periods of high network activity. Knowledge of the typical range of packet loss rate allows one designing error concealment techniques to target the appropriate error range of interest.

Because MPEG compression has a great deal of temporal and spatial dependency, packet loss effects persisted for many frames, and packet loss rates as low as 3% translated into frame error rates as high as 30%. This underscores the necessity of using error concealment and/or error resiliency techniques at the encoder when sending MPEG video over the public Internet.

Packet loss rates varied with packet size, with a jump in packet loss rate in packets larger than the Ethernet MTU size of 1500 bytes. A method was suggested for filling IP packets with MPEG data based upon the experimental results.

Packet losses were not independently distributed, and instead showed an increased conditional probability of packet loss in the vicinity of other packet losses.

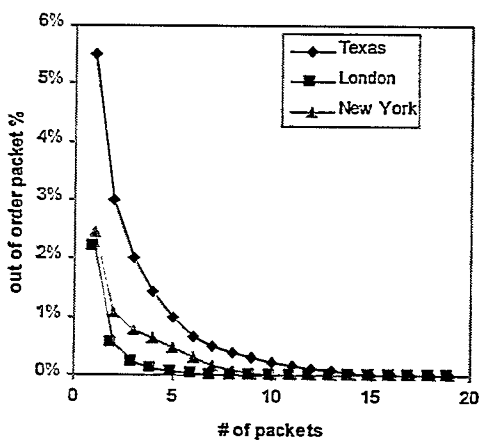

Figure 13. 1 Mbps Out of Order delay in packets

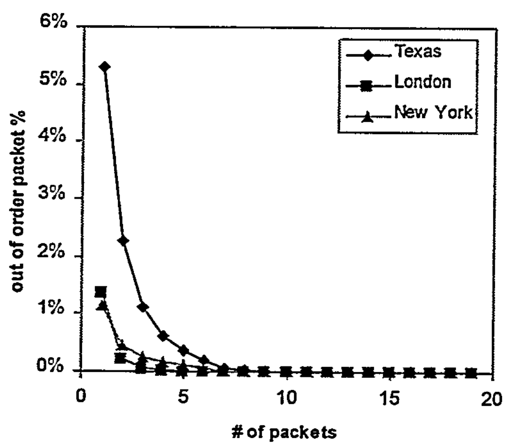

Figure 14. 384 kbps Out of Order delay in packets

UDP transmission resulted in significant numbers of out of order packet arrivals, with average rates ranging from 1.75% to 12.65% of the sent packets for the three paths and two data rates. A majority of the out of order arrivals were delayed by a single packet. The experimental out of order packet arrival rates aid in choosing a decoder buffer size.

6. ACKNOWLEDGMENTS

We wish to thank the Gertrude Stein Theater Company, Graeme Balfour and the Queen Mary & Westfield College, Prashant J. Shenoy and the University of Texas, and Marah Rosenberg for their assistance and use of their networks to run our experiments.

7. REFERENCES

[1] Aravind, R., Civanlar, M., and Reibman, A., *Packet Loss Resilience of MPEG-2 Scalable Video Coding Algorithms*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 5, October 1996.

[2] Bolot J. Crepin H, Vega Garcia A., *Analysis of Audio Packet Loss on the Internet*, Proceedings NOSSADV 95 (Network and Operating System Support for Digital Audio and Video), pp 163-174, Durham, NH, April 95.

[3] Cen, S. , Pu, C., and Walpole, J., *Flow and Congestion Control for Internet Streaming Applications*, Proceedings Multimedia Computing and Networking 1998 (MMCN98).

[4] Comer, D.E., *Internetworking with TCP/IP, Vol I: Principles, Protocols, and Architecture*, Third Edition, Prentice Hall, 1995.

[5] Gaglianello, R. and Cash, G., Montage: *Continuous Presence Teleconferencing Utilizing Compressed Domain Video Bridging*, IEEE ICC95, Seattle, Washington, June 1995.

[6] Hoffman, D., Fernando, G., Goyal, V., and Civanlar, M., *RTP Payload Format for MPEG1/MPEG2 Video*, Internet Engineering Task Force RFC 2250, January 1998.

[7] ISO/IEC JTC 1/ SC 29/ N 071, *Coding of moving pictures and associated audio – for digital storage media at up to about 1.5 Mbits/s – Part 1: Systems, Part 2: video*, CD 11172, 1992.

[8] Nee,P., Jeffay, K., Danneels, G., *The Performance of Two-Dimensional Media Scaling for Internet Videoconferencing*, Proc. 7th Intl. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), St. Louis, MO, May 1997, pp. 237-248.

[9] Paxson, V., *End-to-End Internet Packet Dynamics*, ACM SIGCOMM '97, September 1997, Cannes, France.

[10] Rosenberg, J., *Slides on Internet Loss Measurements*, presented to the Voice on the Net Conference, Boston MA, Sep. 25, 1997, http://www.cs.columbia.edu/jdrosen/papers/vonslides.pdf.

[11] Schulzrinne, H., Casner, S.,Frederick, R., Jacobson, V., *RTP: A Transport Protocol for Real-Time Applications*, Internet Engineering Task Force RFC 1889, January 1996.

[12] Talley, T., Jeffay, K., *Two-Dimensional Scaling Techniques For Adaptive, Rate-Based Transmission Control of Live Audio and Video Streams*, Proc. Second ACM International Conference on Multimedia, San Francisco, CA, ACM Press, October 1994, pp. 247-254.

[13] Wang,Y. and Zhu, Q.F., *Error Control and Concealment for Video Communication – A Review*, Proceedings of the IEEE, May 1998.