An Examination of MBone Performance

Mark Handley
UCL and ISI

January 10, 1997

Abstract

To design tools that will utilise the Internet's Multicast Backbone effectively requires an understanding of the processes that affect multicast traffic. In this paper we examine what can be discovered about mbone performance from the point of view of a single end-system. In so doing we derive a valuable insight into how packet loss in the mbone affects traffic flows. Based on this insight, we make suggestions as to how various classes of multicast application much be better designed and to the limits of some mechanisms that have been suggested.

1 Introduction

In order to design tools that will perform a task effectively, it is necessary to have a good understanding of the environment in which those tools will function. In this paper, we monitor and analyze the performance of the Internet's Multicast Backbone (Mbone), and attempt to loosely characterize the environment in which Mbone tools have to operate.

This characterization leads us to make suggestions about the design of multcast applications. In particular, it leads us to conclude the the designers of reliable multicast algorithms should think about the use of packet level forward-error-correction.

These results also have implications for the design of rate-adaptive multicast applications, and lead us to conclude that such applications will not perform well with current FIFO queuing schemes as the loss feedback they obtain is not useful for such applications to use in a rate-adaptive feedback loop. Thus we believe that queuing algorithms such as RED and Fair Queuing are *necessary* to allow such applications to obtain more meaningful feedback from the network about appropriate transmission bandwidths.

Two other studies [5][14] of this area have been performed. The work of Bolot et al [5] studies temporally correlated loss in fixed bit-rate audio streams, and finds results which are in general agreement with this study. Yajnik et al [14] study both spacial and temporal loss correlation by using dedicated monitoring software at 17 sites. Thus they get more detailed correlation information than we are able to manage, but because these sites are members of the networking research community, their results are not necesarily representative of the actual distribution of sites involved in large sessions. When monitoring MBone sessions at real receiver sites we discover higher loss rates than they encounter, although the pattern of their results is in general agreement with ours. Both these studies use fixed bit-rate audio data as their test traffic source. By using variable-rate video as our test traffic sources, we are additionally able to draw conclusions about congestion control mechanisms.

1.1 MBone Monitoring

The ways we can monitor the state of the Mbone from an end-system are necessarily limited, but in the last two years the options have increased significantly. Multicast routes and (multicast) link loss statistics can be collected with *mtrace*, and with the wide-scale deployment of RTP version 2 based conferencing tools, the tools themselves report limited loss statistics using RTCP. Both mtrace and RTCP only provide loss averages, rather than information about individual packet losses.

Prior to the development of mtrace and RTPv2 based tools, Stuart Clayman and Atanu Ghosh at UCL had written the tools *mr*, *mwatch* and *rtpmon* which performed similar tasks. However, mr is unreliable due to it's reliance on being able to contact an mrouted using IGMP and also due to the fact that DVMRP multicast routing in older versions of mrouted is non-deterministic in its choice of route when two equal metric paths are available. Rtpmon was reliable, but required that sites involved in a session run the monitor daemon, which is difficult at achieve in sessions of any significant size.

In addition, we can examine the routing tables from *mrouted* directly to monitor route stability, and we can observe traffic as it arrives at sites to which we have access to look at individual packet losses.

Of primary importance for audio, video, reliable multicast and conference control applications is information about packet loss. In the design of audio and video tools, information about delay and jitter is also of significance, but currently there is no way to monitor this information as current RTPv2 based tools do not report jitter correctly.

2 Data Collection

Three programs were written to collect data on which the following examination is based:

- An RTP logger, which logs data packet arrivals at a receiver.
- An RTCP logger, which logs RTCP reception reports from all receivers in a session.
- An mtrace daemon, which attempts to trace the multicast routes from a sender to all the receivers in a session.

The RTP logger

For every RTP data packet that arrives at a receiving site on the specified address, this logs the synchronisation source, packet Length, sender timestamp and arrival time.

The RTCP logger

For every RTCP control packet that arrives at a receiving site on the specified address, this logs the synchronisation source of the reporting host, RTP SDES information of the reporting host including NAME, CNAME and TOOL, and the report arrival time. In addition it logs a list of reception reports for each sender heard at this reporting host, giving the synchronisation source of the sender and the fraction of expected packets lost in the last reporting interval.

An example of a logged RTCP report is:

```
TIME: Sun May 26 14:47:27 1996
RR: ssrc=296197894 flost=0
SDES: ssrc=295962065
CNAME: jim@199.45.1.66
NAME: Jim Martin (InteropNet)
TOOL: vic-2.7a38
EMAIL: jim@interop.net
```

The Mtrace Daemon

To monitor where loss is occurring in the Mbone, we wrote an mtrace daemon. This listens to the RTCP packets sent from receivers to a multicast group and builds sender and receiver lists for the session. It then attempts to *mtrace* the multicast route from the chosen sender to all the receivers.

At the time these statistics were gathered in April 1996, the *mwatch* daemon reported multicast router version information as:

version	hosts	percent			
3.8	763	47.0%	3.3	31	1.9%
11.0	194	11.9%	2.0	21	1.3%
10.3	168	10.3%	1.0	14	0.9%
11.1	140	8.6%	3.5	12	0.7%
3.6	88	5.4%	11.2	7	0.4%
2.2	70	4.3%	3.2	5	0.3%
10.2	68	4.2%	3.1	5	0.3%
3.4	34	2.1%	3.7	4	0.2%
			Total	1624	

Of these, only versions prior to 3.3 do not properly support mtrace, and so it should be possible to mtrace through approximately 93% of the multicast routers in the Mbone. For example, on May 21st 1996, there were 112 hosts listening to the NASA STS-27 video session, but of the 112 mtraces attempted, only 68 were successful. Of the successful traces, the mean path length was approximately 8. Assuming this to be typical of unsuccessful mtraces, we expect a path to contain one or more mrouters that do not support mtrace approximately 44% of the time, which is in line with these figures, and so implies that it is possible that routes where mtrace was unsuccessful are similar to those where it was successful.

However, an alternative explanation is that mrouters that do not support mtrace are located in less well supported locations (as these are older versions of mrouted), so are more likely to be located away from the "backbone" routes, and therefore less likely to be on any particular mtrace path. If so, then mtrace is also failing for some other reason. If this is due to excessive loss, then the statistics reported by successful mtraces will not be representative of typical conditions in the Mbone.

If this latter explanation holds true, then any prediction of typical global Mbone conditions based on successful mtraces will be an upper bound on loss rates likely to be seen. For this reason, we do not use loss statistics gathered from mtrace alone, but rather use them to illustrate where problems are occurring in the mbone.

3 Analysis of network logs

3.1 NASA Shuttle Video, May and Sept 1996

RTP and RTCP packets for the NASA Shuttle Video session were logged from Sunday 26th May 1995 to Tuesday 28th May inclusive. The RTP logfile totaled 279MBytes and logged 2.7 million packets. The RTCP logfile totaled 125MBytes and logged approximately 840,000 RTCP packets from a total of 306 receivers. A similar number of RTP and RTCP packets were also captured between the Monday 16th and Thursday 19th September.

Although it is not claimed that these traces are repre-

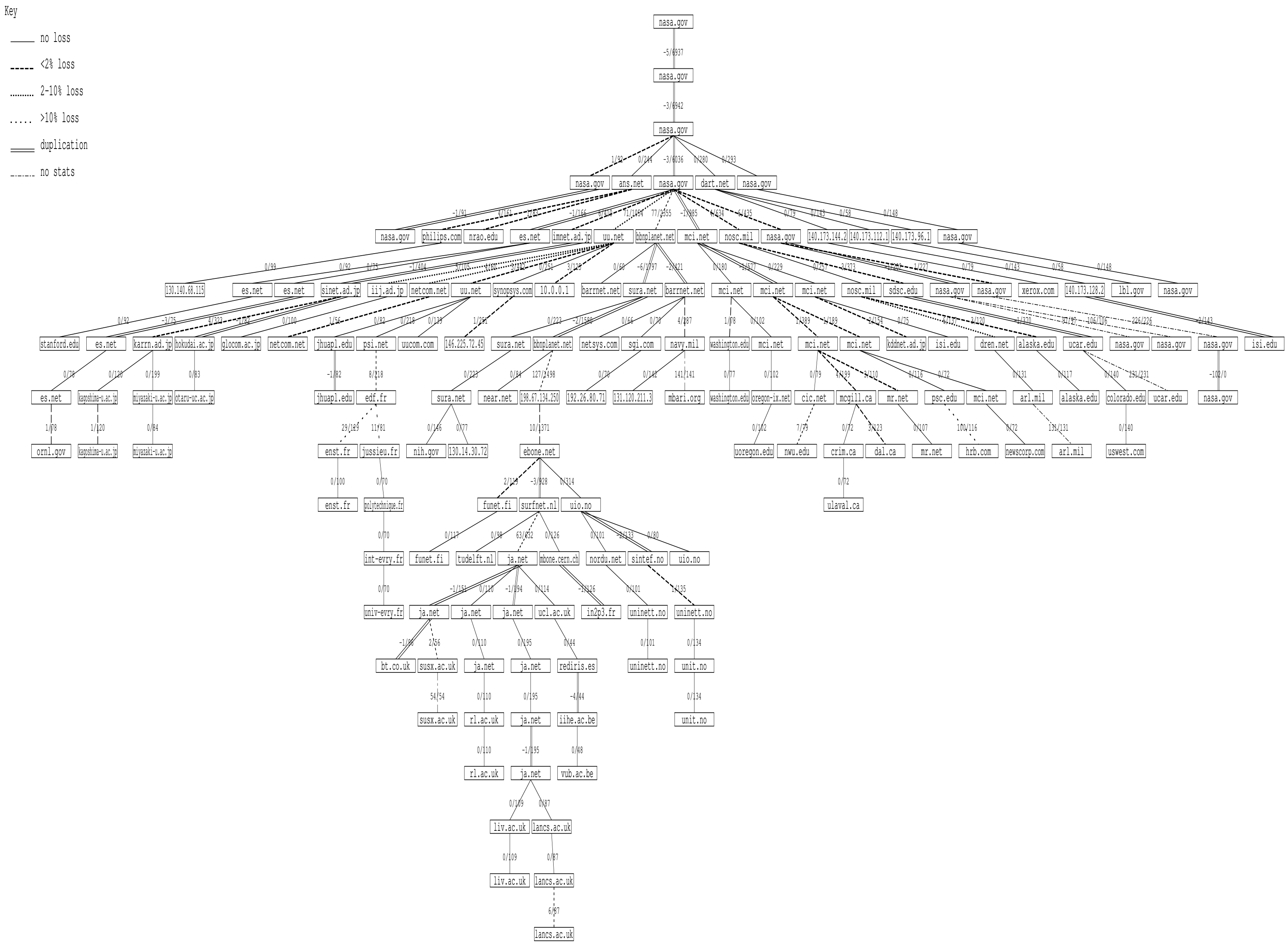


Figure 1: The Loss Distribution Tree for NASA Shuttle Video, 21/5/96, 19:57 BST

sentative of all Mbone sessions, the May traces cover two quiet days (a Sunday and a public holiday in Europe and the US) and a normal working day, and routes covered by the traffic cover most of the Mbone backbone links. The September traces cover several weekdays, although the network may be lighter loaded than normal because this is still during the period when UK university students are on holiday.

RTCP statistics from the vic video tool which was used by all participants give the loss rate calculated by dividing the packets received by the packets expected in that interval. This fraction is sent as an integer in the range 0-255, which is sufficient granularity for these purposes. It is possible that some of the RTCP receiver reports are reporting loss which occurs in the end-system rather than the network as vic suffers from processor starvation and overflows its UDP receive buffer. However, given that the traffic measured constitutes a relatively low frame-rate video stream (typically only 3-4 fps when the video is in motion, higher when there's no motion, but very little decoding needs to be done then) it is unlikely that this is a significant source of loss. On the author's 60MHz SparcStation 20, decoding and displaying this video stream requires between 4% and 10% of the CPU time. Even low performance workstations are unlikely to be CPU starved by this video stream alone. Thus although we cannot be sure that the statistics have not been affected to some degree, it is expected that the influence of this effect is small.

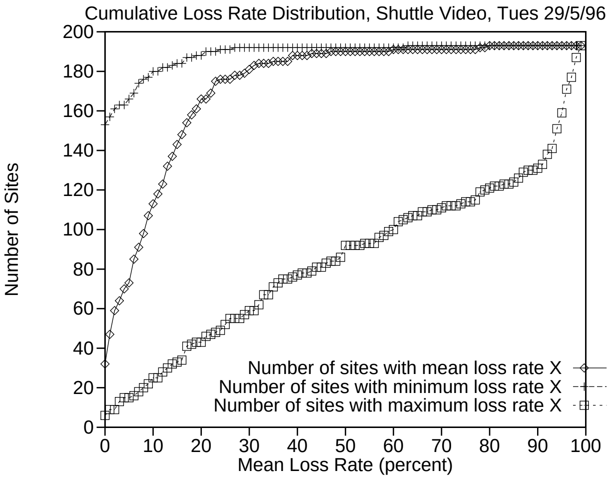


Figure 2: Loss Rates by Site

Loss Rates by Site

Figure 2 shows the breakdown of loss by site as a cumulative graph. Each RTCP loss rate field gives the mean loss rate seen by that receiver for the NASA source in that RTCP reporting interval. Reporting intervals vary depending on the number of receivers, but for this session they are typically between 10 and 60 seconds. For each receiving site, the minimum reported loss rate, the mean reported loss rate, and the maximum reported loss rate during the day are calculated.

This graph shows that 50% of receivers have a mean loss rate of about 10% or lower, and that around 80% of receivers has some interval during the day when no loss was observed. However, 80% of sites reported some interval during the day when the loss rate was greater than 20%, which is generally regarded as being the threshold above which audio without redundancy becomes unintelligible, and about 30% of sites reported at least one interval where the loss rate was above 95% at some time during the day.

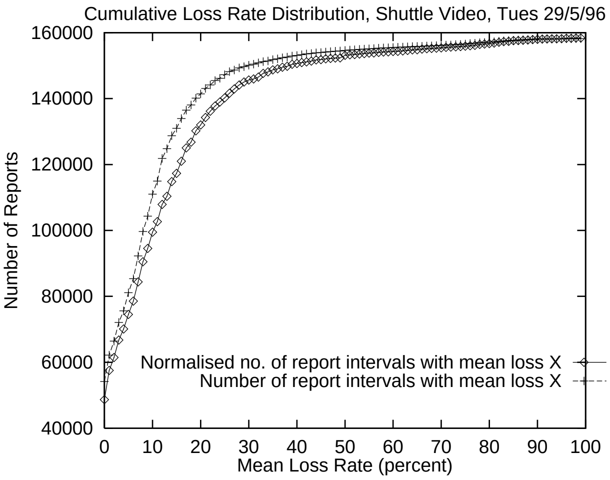
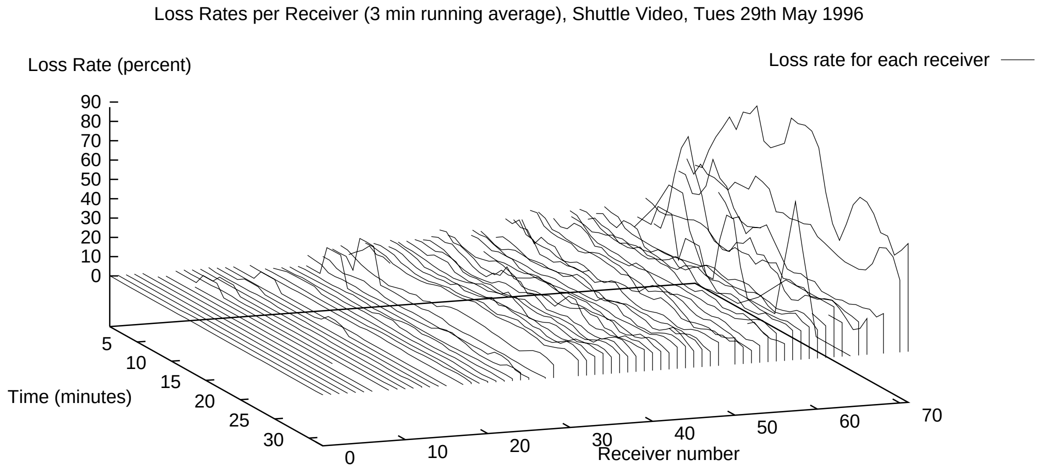
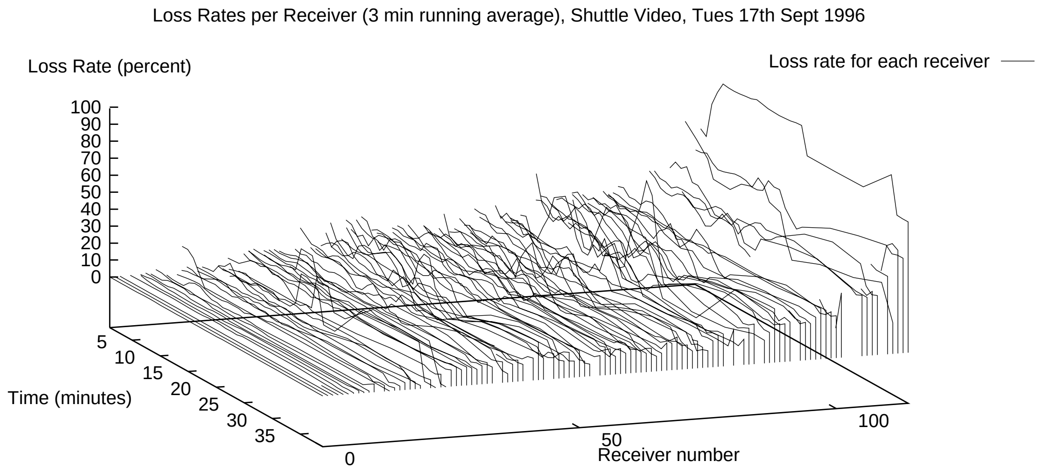


Figure 3: Loss Rate Report Distribution

Loss Rates Report Distribution

Figure 3 shows the number of loss rate intervals reporting each loss rate during the day as a cumulative graph. This includes reports from all receivers. As this graph may have been affected by the loss of RTCP reports, we show the raw results and the results by normalizing the reports from each individual receiver, such that a report from receiver i is given weight $w_i = n_i/\overline{n}$ where n_i is the number of reports received from i and $\overline{n}$ is the mean number of reports received by a receiver. The normalized curve is intended to weight receivers whose RTCP reports get lost on the way to the monitor as if their reports had arrived. Whether this is in fact the case is not clear, as it may overweight such receivers as some receivers duration in the session may also be shorter because the loss rate is high. However, as the true situation is likely to lie between the two curves and as the two curves are sufficiently close to each other this does not matter significantly.





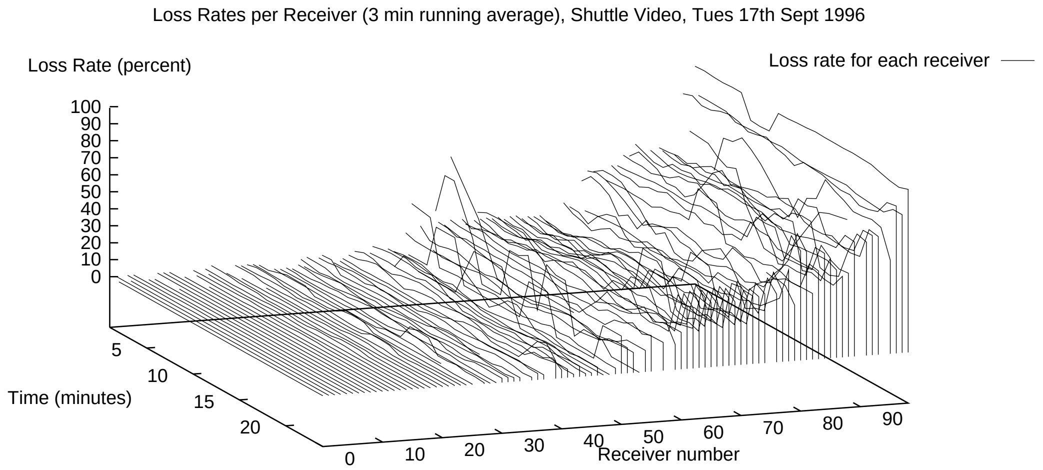


Figure 4: Loss rate against time for each receiver

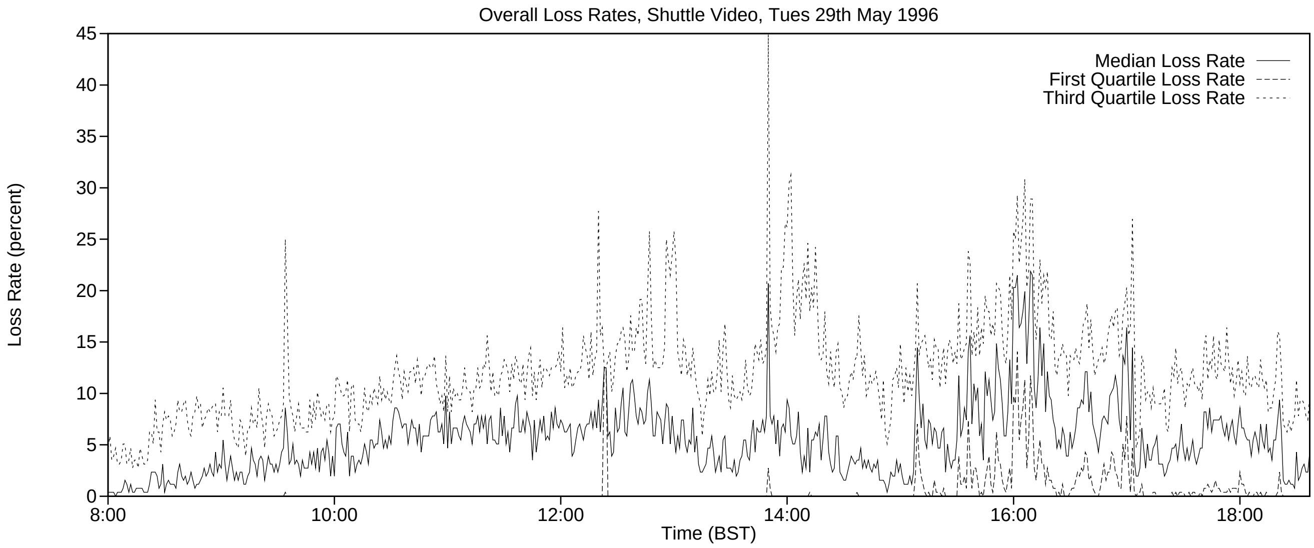


Figure 5: Loss Rate Distribution against Time

These graphs show the loss distribution to be very long tailed, with large amounts of relatively low loss rates observed, and although very high loss rates are seen, they are rare. 80% of all reports give loss rates of less than 20%.

Loss Rate Distribution with Time

Figure 5 shows the loss rate distribution against time in minutes for the shuttle video session on Tuesday 29th May. Three lines are shown, giving the median loss rate reported from all receivers reporting in one minute interval, and the first and third quartiles. Note that although 25% of the receivers are suffering no loss for much of the day, the median loss rate is between 5 and 10% for the majority of the day. Also 25% of receivers are suffering loss rates of greater than 15% for all the afternoon (between 12:00 and 4:30 UTC) when both the European and US communities are active.

Figure 4 shows the loss rate against time for each receiver during a 33 minute interval of the shuttle video session from 11:30am BST on Tuesday 29th May and for two intervals on Tuesday 17th Sept 1996. These graphs were obtained from the RTCP receiver reports for each receiver by taking a 3 minute windowed average to minimize boundary effects caused by the de-synchronized nature of RTCP receiver reports. Receivers are shown in order of mean loss rate. Although the precise time intervals are arbitrary, they were deliberately chosen from a period when fairly high loss rates are visible in figure 5 and the equivalent for the 17th Sept. Several graphs were taken from these data sets collected on 29th May and 17th Sept, and show similar degrees of correlation of loss between receivers.

The graphs shown are representative of the range seen, and vary from reasonably good correlation (on May 29th) to less good correlation (on the first graph from Sept 17th). They show some correlation of loss between receivers, as would be expected, but they also show a large amount of additional low amplitude noise, which would only be expected if many different points in the distribution tree are causing usually small amounts of loss independently. In addition, there are always a small number of receivers suffering very high uncorrelated loss. This agrees acceptably well with the loss distribution shown in figure 1, which shows a small number of high loss links which are responsible for the larger scale correlation in 4 and a large number of links.with low loss rates which add noise to the correlation.

Figures 6 and 7 show loss rate against time measured from the RTP traffic received at UCL, London from NASA. Unlike 4 these give very fine grain loss statistics, but only does so for one site. The first graph in figure 6 was taken early in the morning (UK time) on Tuesday 28th May 1996, when the networks were lightly loaded. The second graph was taken at the time of maximum loss as shown on figure 5. Both graphs show a clear 31 second periodic loss pattern, although this is less obvious in the second graph because of other loss. The loss autocorrelation graphs taken from the same RTP loss data in figure 8 do show that this effect is still significant when the network loss levels are high. Yatnik et al [14] also reported similar periodic losses.

We have attempted to verify that this effect is not a measurement artifact by performing the same dump on different workstations running different operating systems. The main dumps are all taken on a SparcStation 20 run-

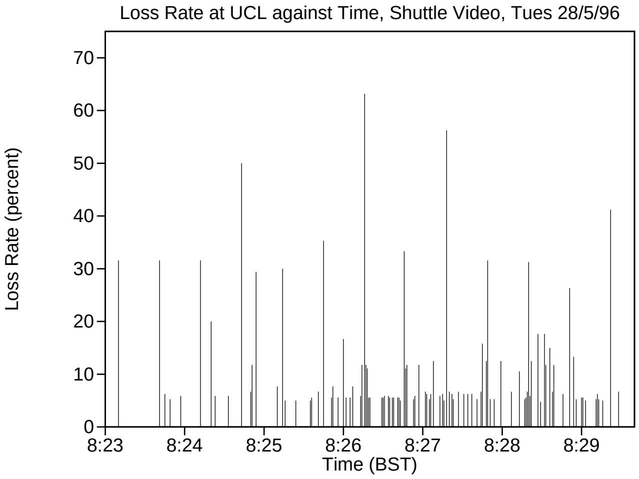


Figure 6: Loss Against Time

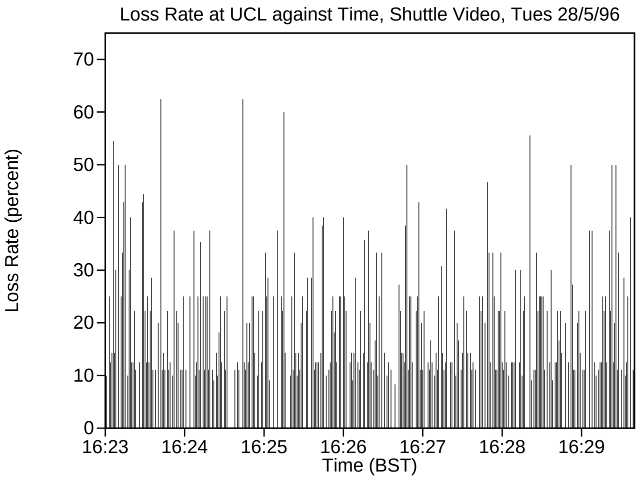


Figure 7: Loss Against Time

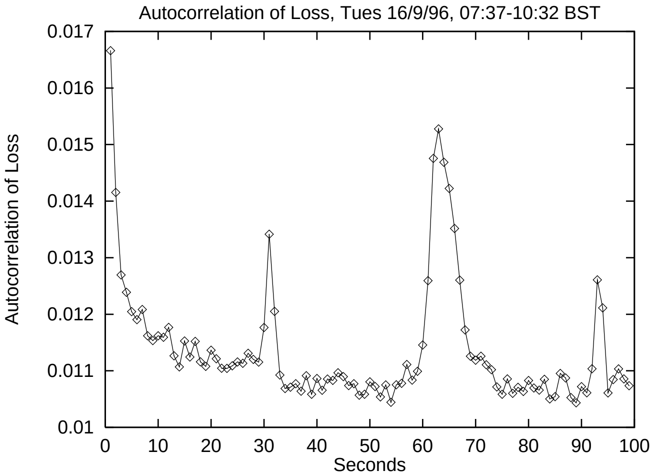


Figure 8: Autocorrelation of Packet Loss

ning Solaris 2.5. Short dumps were also taken on an HP 9000/725 running HPUX on the same ethernet as the SS20, and a SparcStation IPX running SunOS 4.1.3 on a different ethernet at UCL. The effect is still visible on these traces. Test traffic sent through the UCL multicast router monitored on the same machines also does not show this effect.

This effect cannot be a periodic effect in the sending workstation at NASA because it would be visible in the RTCP receiver reports. At UCL, this effect is approximately 50% of packets being lost in one second, repeating every 30 seconds. This corresponds to approximately 1.5% loss, which is large enough to be reported by RTCP. We see significant numbers of receivers which report no loss, which rules out an artifact in the sending machine at NASA.

Such period loss patterns are likely to result from a periodic effect in the network itself. In the past, Van Jacobson has attributed similar effects to a bug in certain routers, where they invalidate their interface routing cache while processing routing updates, and to synchronisation of routing updates [9]. Although these seem likely there is not enough data in these RTP or RTCP figures to verify either hypothesis.

Both local tests and the absence of such effects for significant numbers of receivers tend to imply that this is not likely to be a ubiquitous artifact of DVMRP routing itself in the mrouted implementation. However, an interaction between multicast routing and some other periodic process on a tiny minority of workstations running mrouted cannot be ruled out from this data.

Loss Rate Distribution with Senders Data Rate

The graphs in figure 9 show the mean loss rate for each one minute interval calculated from the RTCP receiver reports plotted against an estimate of the sent data rate. This estimate is calculated by assuming that RTP packets missing when they arrive at UCL were of mean size. As vic tends to send close to constant size packets, this assumption appears justified. The data plotted in both cases represents a fourteen hour UK daytime section of video reception.

The lines plotted over this data are least-squares fits for each 50 minute interval in the source data. They attempt to illustrate whether any effects may have only been present for part of the sampled data, but they should be read as illustrative only, as there may be coincidental boundary effects such as a short period of low loss rate due to reduced cross traffic coinciding with a period of increased sender bandwidth.

In particular, the data from the Tuesday graph illustrates the difficulty of building a congestion control mechanism for multicast traffic that must adapt on significantly longer time-scales than TCP does. In particular, in the data for Tuesday 29th May, there are long periods of time where there is no discernible correlation between sent data rate and mean receiver loss rate. If even a fairly small proportion of receivers were showing such a correlation, then such a trend should be visible in this graph.

If this data is extrapolated, it shows that multicast applications that attempt to adjust so that no receiver is suffering significant loss will almost inevitably adjust so that they send no data (or whatever their minimum data rate is configured to). Thus schemes such as [2] are unlikely to be useful in the wide area mbone as we see it today.

Consecutive Losses

The graphs in figure 10 (A-C) show the frequency of RTP sequence number increments as observed at UCL on each of several days. We know from figures 6 and 7 that at least two processes combine to produce loss, and this is reflected in figure 10. The base losses appear to obey a simple rule where the probability p of a sequence number increase of $n, n >= 1$, is approximated by $p(n) = an^b$ where for the graphs shown, b is in the range -8.7 to -3.5. In addition there is a peak in all the graphs in figure 10 at around 5 to 15 consecutive losses which does not follow the simple rule. These peaks are consistent with the thirty-second periodic loss bursts observed in figures 6 and 7.

Figure 10D shows the conditional loss probability; that is given that packets x to $x + n - 1$ were lost, the probability that packet n is also lost[1]. As these curves show, packet losses are not independent, and tend to occur in longer, bursts than would be the case if they were independent. However, with the exception of the thirty-second periodic losses, the excess of bursts of 2-5 packet losses compared with what would be expected from random loss, although statistically significant, is not sufficient to greatly influence the design of most applications, and single packet losses still dominate.

In the case of applications such as UCL's Robust-Audio Tool[11] and INRIA's FreePhone audio tool that provide redundancy by carrying a highly compressed redundant copy of data one or more packets behind the primary encoding, these curves show that there is some gain from adding the redundant encoding more than one packet behind the primary. For example, from the data shown in 10C (16:00-18:00), 77243 packets were sent and 18640 (24.1%) were lost. Adding a single redundant payload one packet behind the primary encoding allows the reconstruction of audio from 7082 missing packets resulting in an effective loss rate of 15.0%. Adding the same redundant payload two packets behind the primary allows the

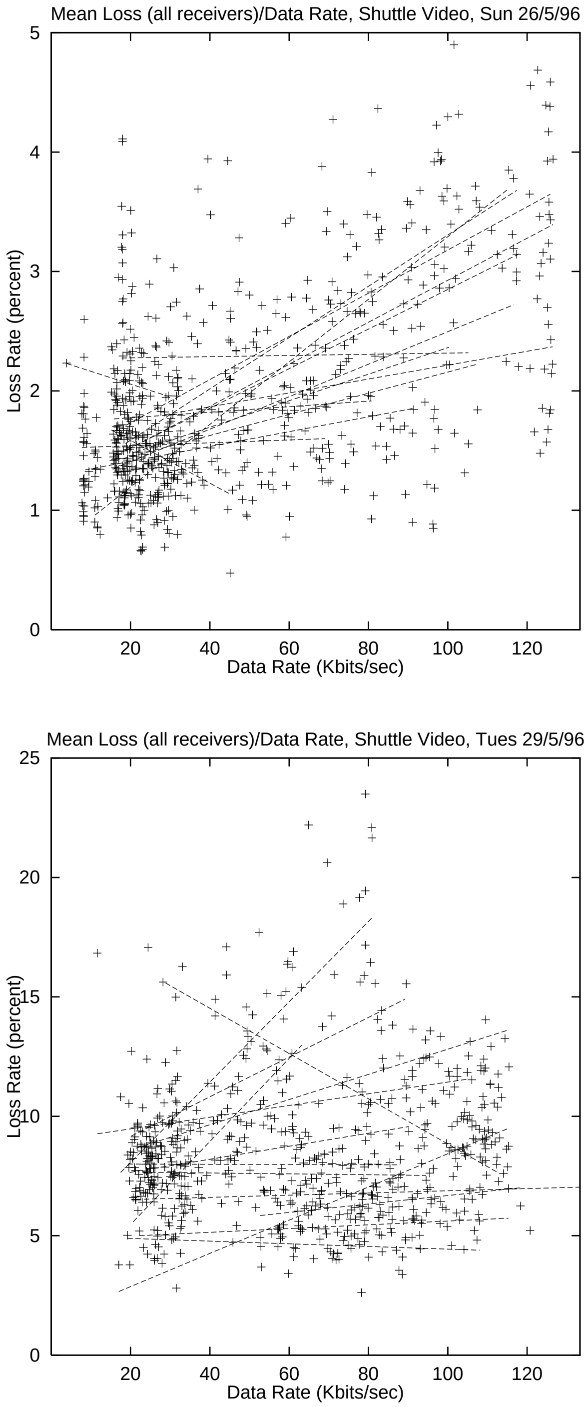


Figure 9: Loss Against Sent Data Rate

[1] n=0 gives the probability that a packet is lost if the previous packet was not lost

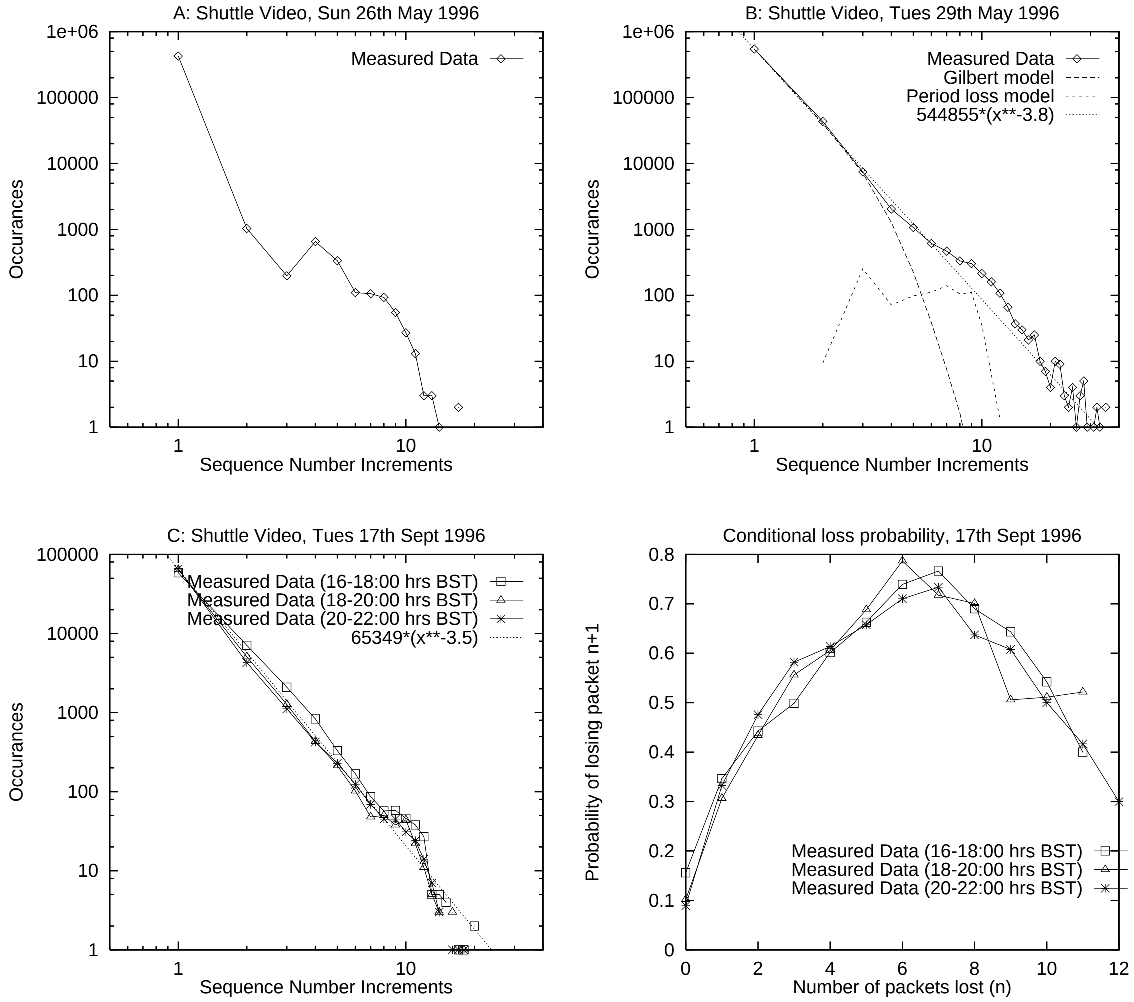


Figure 10: Frequency of Sequence Number Increments (Consecutive Losses - 1)

reconstruction 7885 packets resulting in a effective loss rate of 13.9%. Unfortunately this extra gain comes at the expense of extra delay in the case of this audio coding scheme so may or may not be worthwhile depending on the purpose of the stream itself.

Whilst these results agree with the conclusions of Bolot[3], the two-stage Gilbert model used only provides a first order approximation to these results. Figure 10B shows a two-state Gilbert model (figure 11, p=0.083, q=0.823).

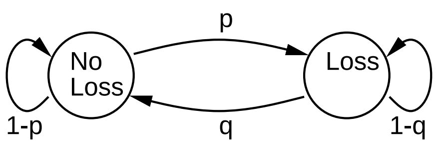


Figure 11: Gilbert Model

This is not entirely unexpected given that the loss process measured clearly consists of a congestive loss process which the Gilbert model tracks quite well and an additional periodic failure which the model does not take into account. Modeling this periodic failure using the traffic data rate distribution in figure 9B and assuming a 500ms failure every 30 seconds gives the distribution shown in 10B as Periodic model. This does go some way towards explaining the tail on these distributions, but does not do so entirely, so some other failure must be occurring fairly infrequently.

4 Conclusions

Drawing general conclusions from limited data is a process fraught with dangers. We believe that the RTCP data we have presented is reasonably representative of network conditions on the Mbone as a whole. The same general patterns show up in data gathered at different times and (to a lesser extent because the sessions are smaller) data sourced from different locations. We are less certain about how representative the mtrace data is because we do not know sufficient information about the sites to which the mtraces did not succeed. However the loss distribution expected from the mtrace data matches fairly well the loss distribution reported from the RTCP data, so we believe these results too are fairly accurate. The RTP data was only gathered at UCL, and we do not claim it to be representative of the Mbone as a whole. However we do believe that the path to UCL is not atypical. Although this path is longer than the mean (both physically and topologically) the loss rates in figure 7 correspond quite well with the median loss rate in figure 3. We do not know how prevalent period loss patterns such as those in figure 8 are, nor where in the network they arise, so we make no claims about how typical this feature is. The loss produced by this period failure corresponds to 1% - 1.5%, so is only a relatively small proportion of the loss seen at busy times.

Based on tentative confidence in how representative these figures are, we attempt to draw some guidelines for tool designers.

4.1 Reliable Multicast

Taking the link loss statistics from the mtrace daemon data in figure 1 as independent loss probabilities[2], the probability of a single packet reaching all destinations during the time the mtrace statistics were gathered is 0.03. Other sessions monitored produce similar probabilities; for example in the USENIX 97 conference the mtrace tree built from the 17 successful mtraces shows a probability of a single packet reaching all receivers of 0.06.

The accuracy of these probabilities is open to some debate. The mtrace statistics may contain some inaccuracies due to boundary effects of the sampling, but these are thought to be small. Some links have been assumed to have $p(loss) = 0$, either because the statistics showed some small duplication, or 100% loss indicating that the downstream mrouter didn't collect traffic statistics. In addition, between 60 and 70% of sites were not able to be mtraced, although it is likely that many of these sites would have joined the traced tree nearer to leaves than the root. The assumption of independence of congestion for small variable rate flows is not conclusively shown, but we have circumstantial evidence that it is a reasonable first approximation. We therefore believe these statistics overestimate the probability that a packet will reach all destinations, though we do not know how much of an overestimation this is.

Assuming that these probabilities are not unreasonable, they have great importance for the design of reliable multicast protocols (RMPs). Most current RMPs tend to rely on retransmission of missing packets, and the primary difference between such RMPs is in the mechanisms for scalably deciding who can report loss, and in who can perform the retransmission. For example, SRM[6] is based around both timer and probabilistic delay of NACKs, then multicasting NACKS which serve to suppress duplicate retransmission requests. Whilst this ensures that close to one receiver NACKs a particular packet loss, it is not geared well to environments in which a large proportion of packets are lost by at least one receiver. In such situations the majority of packets will be NACKed and retransmitted at least once. In the vanilla SRM described in [6] all NACKs and retransmissions are sent to all receivers, and this may result in very significant retransmission overhead. In [7] the possibility of local recovery mechanisms is discussed. These show promise to help this situation in the case of a small number of particularly congested links by dynamically discovering significant loss points and performing local retransmission across the lossy link to the downstream receivers. However, although our traces do show a small number of particularly lossy links, they also show a larger number of slightly lossy links which can combine to cause a significant additional retransmission load even if local retransmission deals with the significant loss points.

These observations lead us to conclude that packet-level or ADU-level forward error correction (FEC) techniques should be seriously considered by the designer of any reliable multicast protocol. Even when receivers suffering high loss rates are removed from the multicast groups we measured, significant retransmission rates would be required even though no single receiver is in a significant overload state. Packet-level FEC can result in greatly reduced reliability overheads for such multicast applications because the addition traffic for the FEC scheme servers to fix many *different* small losses at each different site. Such FEC schemes can be combined with local-retransmission mechanisms to avoid needed to provide excessive error-protection to all receivers, or varying degrees of protection could be provided on different multicast groups allowing receivers to select the level of FEC required by joining or leaving these groups.

Note that whether or not to provide FEC is orthogonal to the issue of congestion control, which should be performed irrespective of the reliability mechanism chosen.

[2] We treat links with 100% loss to be mtrace statistic gathering errors and assume no loss occurs on them. Assuming losses on one link are independent of losses on another link is based in part on the lack of correlation between data rates and loss rates shown in figure 9B. In addition, experiments with mwatch on the MICE project implied that this assumption is not unreasonable as a first order approximation for flows that do not constitute a large proportion of the traffic on any individual link. We do not however have firm evidence that this is the case.

4.2 Bulk Data Reliable Multicast and Congestion Control

One of the major problems in congestion control for reliable multicast is that a sender cannot know on a timely enough basis whether congestion feedback is representative of a problem to only one receiver or to many receivers. If congestion control is performed from time-averaged results from many receivers such as those in figure 9, or if the sender performs congestion control based on per-packet NACKs (as described above) then the sent data rate is likely to decrease to zero.

Figure 4 shows that loss correlation is fairly readily observable from RTCP-style receiver reports. This suggests a mechanism by which congestion control may be performed for reliable-multicast bulk-data-transfer. Using this correlation data, receivers can be subdivided into separate multicast groups, each of which receive well correlated loss. It does not then matter which receiver sends a NACK because the sender can assume it is significant with reasonable confidence. Thus a representative may be elected to perform undelayed NACKing (although others can send delayed NACKs if required ala SRM) and the sender may perform TCP-style congestion control on the same time-scales that TCP does. This will allow non-zero throughput and achieve a fair-share[3] of the bandwidth in the presence of background TCP traffic. This does mean that the sender must send all the data multiple times, once to each different correlation groups, but this is still much more efficient than unicasting the traffic many times. There are likely to be a small number of receivers for which very high loss rates and a lack of correlation with other receivers mean that unicast is still the best option.

4.3 Audio and Video

For packet audio traffic without any form of loss protection or masking, the limits to comprehensibilty are reached at around 10% loss[11], although it may become annoying at lower loss rates. Simple error correction techniques[11][4] exist that increase this comprehensibility limit, and make loss much less noticable at lower loss rates, although the distribution of consecutive losses shown in figure 10 mean that noticable artifacts are likely to occur at lower loss rates than suggested.

However, it is clearly not desirable to keep adding forward error correction to protect an audio or video stream in the face of persistent high levels of congestion as the dropped packets unnecessarily use up bandwidth between the sender and the bottleneck, and more well behaved traffic sharing the bottleneck will suffer significantly.

[3] in as much as TCP receives a "fair share" of the bandwidth

Layered encodings[1] have been suggested as one solution to this problem of heterogeneity, and indeed show great promise as a mechanism to allow different data rates to be delivered to different receivers depending on available resources. However, as figure 9 indicates, determining whether the traffic being received is a significant *cause* of loss is not an easy problem in complex scenarios where that traffic represents only a very small proportion of the traffic flowing over the congested links. Thus it is very unclear how techniques such as RLM[12] would fair in such scenarios.

We are forced to conclude that one of the following must be the case:

- 15Kbps is too high a rate for the current Mbone to support, or;
- to multicast audio and video over the current best-effort internet, loss rates of around 10% must be tolerated and that congestion control mechanisms that operate in a background of TCP traffic on significantly longer timescales than TCP without assistance from router queue management will not work well because loss statistics do not provide useful feedback.

We do not believe the former to be the case although "too high a rate" is not a well defined concept. Typically a "reasonable" rate is regarded as being that which TCP would achieve over the same path. We cannot test TCP performance over the same routes because the DVRMP routing is often set up specifically to avoid congested links, but over similar paths TCP connections do normally achieve a *mean* throughput of greater than 15Kbps. However, whilst TCP adjusts its rate rapidly to lessen congestion when it observes it, adaptive layered mechanisms such as RLM cannot adapt so rapidly, and so must tolerate congestion transients.

Thus for such real-time adaptive streams, it appears that we need support from router forwarding engines such as would be provided by RED[8] and Fair Queuing[10][13] before we can design even receiver driven multicast congestion control mechanisms that will accurately sense appropriate network bandwidth.

References

[1] E. Amir, S. McCanne and M. Vetterli, "A Layered DCT Coder for Internet Video." IEEE International Conference on Image Processing. September, 1996. Lausanne, Switzerland.

[2] J.-C. Bolot, T. Turletti, I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the

Internet" Proc ACM Sigcomm '94, London, ACM, 1994.

[3] J-C. Bolot, T. Turletti, Adaptive error control for packet video in the Internet, Proc. ICIP '96, Lausanne, Sept. 1996.

[4] J-C. Bolot, A. Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet", Proc IEEE Infocom '96, San Francisco, CA, pp. 232-239, April 1996.

[5] J-C. Bolot, H. Crepin, A. Vega-Garcia, "Analysis of Audio Packet Loss in the Internet", Proc 1995 Workshop on Network and Operating System Support for Audio and Video, pp. 163-174, 1995.

[6] S. Floyd, V. Jacobson, S. McCanne, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", ACM SIGCOMM 95, August 1995, pp. 342-356.

[7] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing". To appear in IEEE/ACM Transactions on Networking.

[8] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance" IEEE/ACM Transactions on Networking, V.1 N.4, August 1993, p. 397-413.

[9] S. Floyd and V. Jacobson, "The Synchronisation of Periodic Routing Messages", ACM Transactions on Networking, Vol. 2, No. 2 (April 1994), pp. 122 - 136.

[10] A. G. Greenberg and N. Madras, "How fair is fair queuing?," Journal of the ACM, vol. 39, no. 3, pp. 568-598, 1992.

[11] V. Hardman, A. Sasse, M. Handley, A. Watson, "Reliable Audio for Use over the Internet" Proc INET '95, Hawaii, Internet Society, Reston, VA, 1995.

[12] S. McCanne, V. Jacobson and M. Vetterli, "Receiver-driven Layered Multicast". ACM SIGCOMM, August 1996, Stanford, CA, pp. 117-130.

[13] A. K. Parekh, and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks - the multiple node case," in Proc. IEEE Infocom '93, vol. 2, (San Francisco, California), pp. 521-530 (5a.1), IEEE, March/April 1993.

[14] M. Yajnik, J. Kurose, D. Towsley, "Packet Loss Correlation in the MBone Multicast Network" Proc IEEE Global Internet Conf. , London, Nov. 1996.