A General Framework for Continuous Media Transmission Control*

Terry Talley Kevin Jeffay

University of North Carolina at Chapel Hill
Department of Computer Science
Chapel Hill, NC 27599-3175
{talley, jeffay}@cs.unc.edu

Abstract: *One of the major problems experienced with LAN-based video conferencing systems is the degradation of conference latency and fidelity due to network congestion. In this paper, we present a comprehensive transmission control framework describing the relationship of latency and fidelity to properties of human perception and network congestion. We discuss how network factors such as physical link capacity and size of maximum transmission units may exacerbate the effects of congestion on the conference and how the impact of congestion can be ameliorated through judicious adaptation of the media streams' bit and message rates. Finally, we demonstrate the ability of an adaptive transmission control algorithm based upon the framework to produce low-latency, high-fidelity conferences over congested internetworks.*

1. INTRODUCTION

We are interested in supporting desktop video conferencing across existing local area networks (LANs). Although the media streams in video conferencing systems have real-time constraints on delay, jitter, and loss, existing LANs typically do not support real-time transmission and most current internetworks of LANs do not provide mechanisms, such as resource reservation, to support guaranteed Quality of Service (QoS). Thus, during periods of high network congestion, packets from the media streams may experience long queueing delays or be lost. This results in high stream latency and reduced throughput that dramatically affects the perceived quality of the video conference [4, 7].

The primary technique used to address congestion on current internetworks is to *scale* the media steams by changing the transmitted bit rate of the stream. The dominant scaling strategy is to reduce the transmitted bit rate when congestion is high and increase the bit rate when congestion is low [3]. Scaling is primarily applied to the video stream since it typically has much higher bit rates than audio and has proven to be an effective technique for dealing with certain types of congestion [1, 2]. There are, however, situations where media scaling is a less effective response to network congestion than adapting the *packaging* of the media data units into network messages [5]. The fundamental packaging strategy is to decrease the number of messages (for example by increasing the number of media units carried by a message) when congestion is high and increasing the message rate (for example by decreasing the number of media data units carried in a message) when network congestion is low. Clearly, packaging is related to scaling in the sense that large increases or decreases in the stream bit rate may indirectly force changes in the stream packaging. Indeed, we believe the changes to packaging resulting from large bit rate changes explains some of the success of video scaling techniques; however, we believe that packaging and scaling are orthogonal, complementary techniques. We have previously developed a transmission control framework which integrates media scaling and media packaging and have shown that the framework is effective at controlling the effects of congestion in environments where the application message rate is the same as the resulting network packet rate [5, 6].

Unfortunately, the requirement that the network packet rate equal the application message rate is overly simplistic. In many current internetworks it is necessary to fragment messages into multiple network packets as the size of media data units are greater than the maximum transmission units available on some networks. This raises the possibility that any packaging decisions made by the application may be undone or rendered inconsequential by network fragmentation. In this paper, we expand the transmission control framework to deal with the effects of fragmentation and show the empirical results of transmission over a congested network where fragmentation occurs. We show that even in the face of fragmentation, the application may successfully adapt to network congestion via packaging at the message level and that this adaptation can have a significant effect on the resulting conference quality. Furthermore, we show that in current internetworks, adaptation of both the bit and message rates produces conferences with lower latency and higher fidelity than those produced from non-adaptive transmission algorithms or by those produced by scaling the video bit rate alone. Algorithms based upon the transmission control framework can provide acceptable quality conferences across a broader range of network loads and types of congestion than algorithms which do not manipulate both message and bit rates. This shows that our transmission control framework is general enough to be successfully applied in a wide range of common internetworks employing packet-switching technology.

Section 2 of this paper provides an overview of the general transmission control framework. Section 3 discusses the effect of fragmentation in congested networks. Section 4 presents the results of experiments illustrating the effect of fragmentation on a congested token ring network and shows an adaptive algorithm can amelio-

*This work supported by grants from the Intel and IBM Corporations and the National Science Foundation (CCR-9510156 & IRIS-9508514).

In: *Proceedings of the 21st IEEE Conference on Local Computer Networks*, Minneapolis, MN, October 1996, pages 374-383.

rate these effects. Section 5 provides similar experimental results for an internetwork of token ring and ethernet networks. Section 6 presents our conclusions.

2. The Transmission Control Framework

Abstractly, a video conferencing application is a program that generates and receives time-ordered sequences of audio and video samples called *frames*. Let the bit-rates at which audio and video are generated be b^a and b^v respectively and let f^a and f^v be the corresponding frame-rates. We assume frames are generated periodically: one audio frame is generated every $1/f^a$ time units and one video frame is generated every $1/f^v$ time units. We further assume that it is possible for the conference application to exercise some control over both the size (number of bits) and the rate (inter-frame generation time) at which frames are produced. Let m^s be the application message rate for stream s. Messages may be fragmented into multiple packets as they cross the internetwork. Let p^s_k be the resulting packet rate on the k^{th} hop on the path from source to destination resulting from an initial message rate of m^s.

At well-defined points in time, the sender chooses a bit and message transmission rate for each media stream. In principle, the bit-rate may be changed over time by either changing the frame size (*e.g.*, by changing the coding or compression scheme) or by changing the frame rate (*e.g.*, by temporal scaling [3]). The message-rate may be changed by varying the number of frames from the same stream packed into a single message.

We characterize each media stream s in a conference by the set OP^s of *operating points* in a message-rate × bit-rate space. For stream s, $(m^s, b^s) \in OP^s$ if and only if the conference is capable of generating b^s bits/s and partitioning s into m^s messages/s. Figure 2.1 shows the set of operating points for the audio and video streams of one of the conferencing systems used in this work. For video, this system has a choice of three coding schemes; a *high* quality scheme (approximately 8,000 bytes/frame), a *medium* quality scheme (approximately 4,000 bytes/frame), and a *low* quality scheme (approximately 2,000 bytes/frame). Each scheme generates between 1 and 30 frames per second. Coding is frame-independent, so the system may transmit video at any message (frame) rate from 1-30 messages per second. For example, transmitting 30 messages per second yields full-motion video; 15 messages per second yields half-motion video, *etc.* Thus the set of video operating points contains 90 points, 30 per coding scheme, represented as squares in Figure 2.1. For audio, the system is capable of generating only 1 bit-rate. The system generates sixty 250 byte audio frames each second and can transmit 60, 30, 20, 15, 12, 10, or 6 messages per second (corresponding to 1, 2, 3, 4, 5, 6, or 10 audio frames per message). The circles in Figure 2.1 represent the audio operating points. The set of operating points specify all possible operations of the audio/video devices and all possible uses of the network interface. The transmission control problem is that of determining when to change the operating point and which new operating point to select.

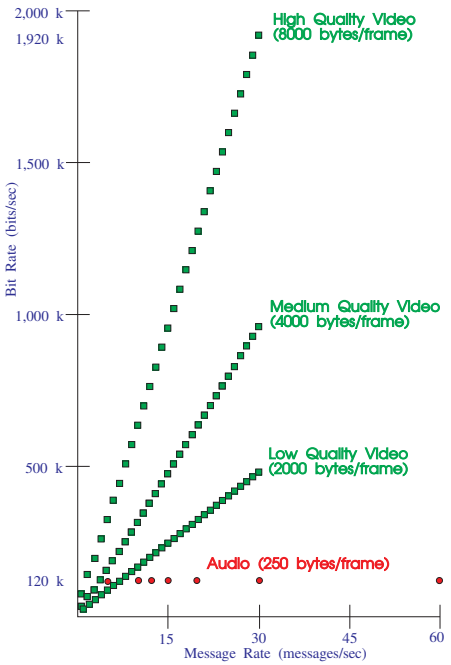


Figure 2.1: Operating points (video as squares; audio as circles).

The key problem is to choose a *feasible operating point*. For stream s, an operating point is *feasible* if and only if it (1) provides acceptable latency and throughput for stream s and (2) it is sustainable given the current level of congestion in the network. The first requirement is an issue of human perception. For example, it is possible that the video hardware is capable of generating a video bit-rate that results in images that are unacceptable for conferencing (*e.g.*, the resolution and/or the frame rate may be too low). Similarly, latency may make a set of operating points infeasible. Let the *network latency*, $L^s(t)$, for stream s at time t be the difference between the time a message is delivered to the packaging function at the receiver and the time the message is available to the network interface on the sending machine. Let the *transmission latency* for a particular frame f in stream s be the difference between the time f is delivered to the packaging function at the receiver and the time f was available to the packaging function at the sender. The difference between network latency and transmission latency is the amount of time a frame spends buffered on the sending side or the *induced latency* for frame f.

Let L^s_{MAX} be the maximum tolerable transmission latency for any frame in stream s. To ensure acceptable latency, the sender cannot buffer individual s frames for more than $MAX^s_L(t) = L^s_{MAX} - L^s(t)$ seconds before they must be transmitted. Therefore, at time $t + 1$, each stream s frame must be transmitted within $MAX^s_L(t)$ seconds and messages generated no slower than the rate of 1 every $\lceil f^s MAX^s_L(t) \rceil / f^s$ secs. We call the fidelity or latency constraints *perceptual constraints*.

The second requirement for feasibility is that the operating point be sustainable under the current network conditions. The network limits our choice of an operating point in two ways. First, for stream s operating point (m^s, b^s), in the absence of congestion, it may be the case that there exists a link in the network with insufficient capacity to process (*i.e.*, transmit or for-

ward) b^s bits/s. We call such constraints *structural capacity constraints*. Second, in the presence of congestion, there may be a hop k with insufficient capacity to sustain a b^s bit/s stream that is partitioned into $p^s{}_k$ packets/s. We call these constraints *congestion constraints*.

To describe the structural capacity constraints, let r^{link}_{min} be the slowest data transmission bit-rate of any network link in the conference path. In order for operating point (m^s, b^s) to be feasible, clearly $b^s \le r^{link}_{min}$, or $1 - b^s/r^{link}_{min} \ge 0$. If $b^s > r^{link}_{min}$ then we say a *structural capacity constraint* exists. Operating points with $b^s > r^{link}_{min}$ are inherently infeasible and should never be considered.

The second type of constraint imposed by the network are congestion constraints. Congestion constraints are caused by network load. The primary effect of congestion at a network element is the development of a queue of waiting packets. Expanding queues increase waiting times and eventually cause loss. To describe the operating points which are not sustainable at particular time t, we use the simple queueing model in Figure 2.2. We define a network *node* as the source or destination computer for a conference or a network interconnection point such as a router. We define a *hop* as a network node and the outbound transmission link of the node. An *internetwork* is composed of a series of hops. In Figure 2.2, a single hop is modeled as two service centers. The first center represents the node; the second represents the outbound network adapter and the associated outbound network link. We call the first center the *node server* and the second the *link server*. The time required to process a packet (*e.g.*, examine the packet and determine the outbound route) at the node server for hop k is represented by PP_k. This processing time is essentially constant for all packets regardless of size. The data transfer rate within a node server for hop k is represented by r^{node}_k The actual data transfer time for a particular packet varies according to the size of the packet. The *medium access* time at the link server at hop k is represented by $MA_k(t)$. This is the time required to obtain control of the outbound link (*e.g.*, the time to get a free token on a token ring) and may vary over time depending upon competition from other nodes for control of the link. Like PP_k, $MA_k(t)$ does not depend upon the size of the packet. The transmission rate on the outbound link at hop k is represented by r^{link}_k Like the internal data transfer time at the node, the transmission time on the outbound link depends upon the size of the packet being transmitted. $U^{node}_k(t)$ is the utilization of the router CPU at time t. $U^{link}_k(t)$ is the utilization of the CPU on the out-bound network adapter at time t and is distinct from the utilization of the outbound network itself, which is only indirectly reflected in $MA_k(t)$.

An operating point (m^s, b^s) is not sustainable at the node server on hop k if:

$$1 \le \frac{b^s / r^{node}_k + PP_k \cdot p^s_k}{1 - U^{node}_k(t)} \qquad (2.1)$$

An operating point (m^s, b^s) is not sustainable at the link server on hop k if:

$$1 \le \frac{b^s / r^{link}_k + MA_k(t) \cdot p^s_k}{1 - U^{link}_k(t)} \qquad (2.2)$$

If relation (2.1) or (2.2) holds for any hop in the path from conference source to destination, the conference is introducing work faster than it is handled and the current operating point is not sustainable.

Depending upon which term in the numerator of (2.1) or (2.2) dominates, hops may be constrained because either the number of bits or the number of packets to be processed exceeds the hop's capacity. If bit rate is the limiting factor at a hop, we say the hop is *capacity constrained*. If the packet rate is the limiting factor at a hop, we say the hop is *access constrained*. Typically, capacity constraints are caused by either (1) internal data movement time at forwarding nodes, or (2) limited network bandwidth on the outbound link. Access constraints are typically caused by (1) packet processing time (*e.g.*, routing decisions), or (2) medium access times when transmitting across shared-medium networks (*e.g.*, the waiting time for a free token on token ring networks).

Recall that the key problem with transmission control is to choose a *feasible operating point*, which is a sustainable operating point with adequate fidelity and latency. For stream s at time t, the set of feasible operating points is what remains after we remove all points from the set OP^s excluded because of perceptual, structural capacity, and congestion constraints. The video conference application could select any operating point in this set and deliver an acceptable quality conference (although some points in the set are clearly more desirable than others).

Ideally, we would compute the set of feasible operating points using (2.1) and (2.2). Unfortunately, to do this we would need to evaluate (2.1) and (2.2) at each hop in the path. On most current internetworks, this is not possible since there is too little information available at the conference endpoints to compute the results.

We can, however, estimate the set of feasible points. We have developed a algorithm, called *Recent Success*, which uses a heuristic to search across the operating points when adapting to congestion [5, 6]. The heuristics assumes that $p^s{}_k$ is directly related to the flow of messages into the network m^s and that by monitoring message performance, the application can successfully adapt to net-

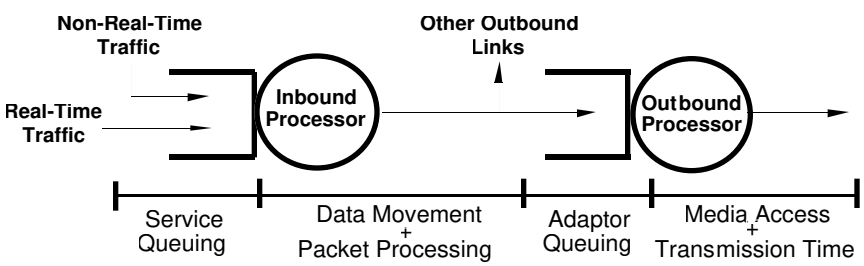


Figure 2.2: Simple model of a network hop

work conditions by adapting only the message and bit rates at the source. The next section discusses why we believe this is a valid assumption and experiments in sections 4 and 5 validate this assumption.

3. Fragmentation, Congestion, and the Transmission Control Framework

Here we discuss the impact of fragmentation on network congestion. In an internetwork where all the maximum transmission units (MTUs) are greater than the largest application message size, $m^s = p^s_k$ for all hops k. However, for internetworks that have one or more hops with MTUs smaller than the largest message size, p^s_k must be calculated for each hop k for a given m^s. Since fragmentation is deterministic, it is possible to derive equations describing the effects of fragmentation and thus calculate the packet rate for any hop k, for a particular operating point. For concreteness, we describe this calculation using the Internet Protocol (IP). Similar calculations can be made for any fragmenting protocol by determining the logical MTU at each hop. For simplicity, we assume all packets follow a single, static path from source to destination and that no packets are duplicated.

We call the set of triples (m^s, b^s, p^s_k) resulting from the points in OP^s the *realization* of the operating points at hop k along the path. Although the set of resulting triples may be different at every hop, the application still may manipulate only the bit and message rates at the source. However, the effect of changes to the operating point at the source may impact the resulting packet rates differently at different hops. For example, consider the network in Figure 3.1, consisting of two token ring networks connected by an ethernet. Since the MTU of the first 16 Mbit token ring hop (17,800 bytes) is large enough to carry any message corresponding to an operating point in Figure 2.1 in a single packet, the message and packet rates on the first hop are the same. However, the ethernet MTU (1,500 bytes) is too small to carry many of the application messages in a single packet, hence the message and packet rates are different on the second hop. Since IP does not reassemble at intermediate nodes, the relationship between messages and packets is the same on the second token ring as on the ethernet. Figure 3.3 shows the three-dimensional representation of the realization of the operating points for the ethernet and second token ring hops. Here, operating points are shown as columns rather than points in a three-dimensional space (*i.e.*, the point (x, y, z) is represented as a column from $(x, y, 0)$ to (x, y, z)). We do not show the realization for the first token ring segment, since $m^s = p^s_1$.

Figure 3.3 shows the large increase in packets on the ethernet and second token ring resulting from fragmentation of the video messages. On the first token ring segment, the maximum message rates for audio and video yield 60 audio packets per second and 30 video packets per second, but on the ethernet and the second token ring, video has a maximum packet rate of 180 packets per second since 6 ethernet packets are needed to carry each 8,000 byte video frame and there are 30 frames per second. The maximum audio rate remains 60 packets per second since the maximum message rate of 60 messages per second has one audio frame per message and is small enough to fit into one ethernet packet. If an access constraint is experienced on the first hop, audio is the dominant potential contributor of packets (a maximum of 60 versus 30 for video) and it is possible that packaging more audio frames per message (*i.e.*, lowering the audio packet rate) may relieve the constraint without any significant decrease in video bit or packet rate. This phenomenon has been demonstrated in [5]. On the other hand, if an access constraint is experienced on the ethernet or second token ring segment, it is likely that the bit rate of video will have to be scaled to address the constraint since video generates many more packets than audio after the messages are fragmented. Because of the large size of video frames relative to the packet size, reducing the packet rate to address the access constraint fundamentally means the video bit rate must be reduced. We believe this phenomenon is part of the reason video scaling techniques have often proven successful at adapting to congestion on ethernet networks, but we also believe that a combination of packaging and scaling is more effective, as is demonstrated later.

A second observation from Figure 3.3 is that the packet rates associated with audio are not monotonically increasing with message rate. This is shown more clearly for audio in Figure 3.4. The reason for this non-monotonic behavior is that packing successive frames into a message eventually reaches the point where the n^{th} frame is successfully packed into the packet, but frame $n + 1$ causes the message size to exceed the packet's capacity and another packet is required. If $n + 1$ frames are packaged into each message, the effect is compounded. Depending upon the conferencing system and the network technologies involved, video may also exhibit non-monotonic behavior (although not with the system shown here). In general, as message size increases, the packaging of frames into packets becomes increasingly efficient and the packet rate decreases with decreasing message rate. This relationship leads us to the hypothesis that manipulating the message rate at the conference source can effectively control the generation of packets in the network and allow the conference source to adapt to

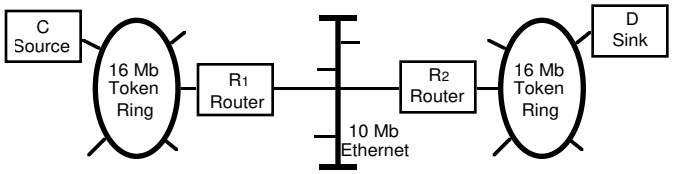


Figure 3.1: Token ring-ethernet internetwork.

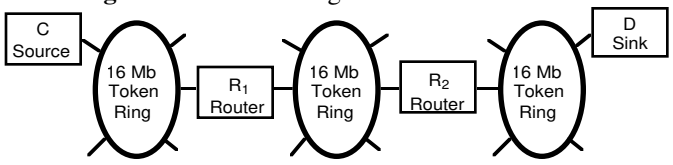


Figure 3.2: Three hop token ring internetwork.

access constraints along the network path, even if the messages are fragmented.

4. Using the Transmission Control Framework on Token Ring Networks

We now present a set of experiments designed to illustrate the effects of fragmentation on message delivery in a congested network and show how these effects can be ameliorated by a transmission control algorithm based upon independently manipulating bit- and message-rates at the source. In our first set of experiments, we show that, for a given network topology, traffic load, and conferencing system, the degree of fragmentation in the network can severely impact conference quality. We also show that an adaptive transmission control algorithm can produce a high quality conference in this environment. In our second set of experiments (in Section 5), we show how the combined effects of fragmentation and non-deterministic algorithms for media access can make loaded ethernet networks a difficult environment for video conferencing, but that with flexibility in packaging and scaling, an adaptive algorithm can produce quality conferences even on heavily loaded internetworks.

Our first set of experiments use the token ring network in Figure 3.2. Here we will show the potentially detrimental effects of fragmentation on a congested network. Note that fragmentation has little effect on capacity constrained servers since the difference between the number of bits transferred in the unfragmented and fragmented cases is small. However, fragmentation can significantly increase the number of packets which must be handled over the non-fragmented case and thus may adversely impact access-constrained servers downstream from the point(s) of fragmentation. We will demonstrate this by using a set of traffic generators to introduce load on the middle hop in Figure 3.2. The competition for tokens on the middle token ring introduces an access constraint at router R_1. In the first experiment, the middle segment uses a large MTU which results in no fragmentation of messages. In the second experiment, the middle segment uses a small MTU which forces fragmentation.

Each experiment uses three different transmission control algorithms. In the *Baseline* (BL) transmission scheme, video is generated in 30 frames per second and each video frame is sent in a separate message. This corresponds to operating point (30, 1920k) in Figure 2.1. Since each video frame is approximately 8,000 bytes and the MTU is 17,800 bytes at each hop, only one packet is needed per video message on any hop in the network. Audio is generated at 60 frames (samples) per second with an approximate size of 250 bytes per frame. In the BL algorithm, each audio frame is also sent as a separate message hence the audio operating point is (60, 120k). The *Video Scaling Only* (VSO) transmission scheme uses the same operating point for audio, but one of three video operating points, (30, 1920k), (30, 960k), or (30, 480k), depending upon the perceived network load. These operating points are obtained by changing the video coding scheme from high, to medium, to low quality. The *Recent Success* (RS) transmission scheme uses the transmission framework and thus may use any audio or video operating point in Figure 2.1. Both the VSO and RS schemes are adaptive and use periodic feedback from the conference destination to the conference source to estimate network performance and as the basis for changes to the selected stream operating points [5, 6].

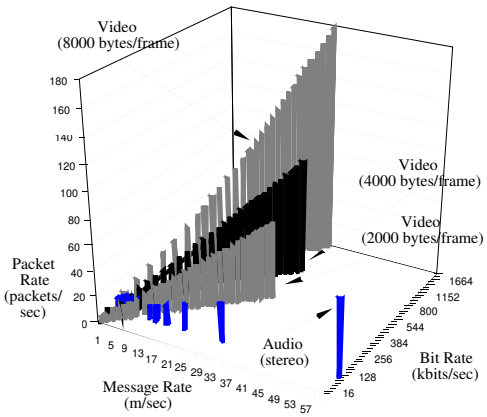


Figure 3.3: Realization of operating points on ethernet and second token ring.

Figure 4.1 shows the performance of the BL algorithm when a moderate load is applied to the middle token ring. The load generates a mild access constraint on the second hop in the network. The traffic generators used for all experiments use the same set of traffic distributions so the generated traffic is comparable, though not identical, for all runs. Figure 4.1 shows that the BL algorithm in the unfragmented case, provides good conference quality. Audio is delivered at the transmitted rate of 60 frames per second and video at the transmitted rate of 30 frames per second (Figure 4.1a). There is no packet loss (Figure 4.1b). Both streams have latencies well under our guideline of 250 ms for maximum acceptable delivered latency (Figure 4.1c&d).

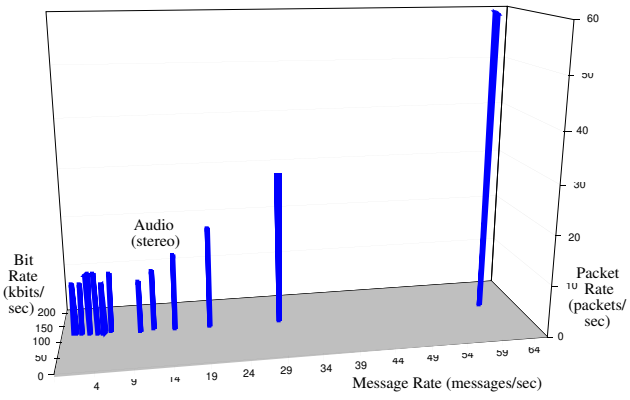


Figure 3.4: Audio packet × message rates, MTU = 1,500 bytes.

The point of this experiment is to demonstrate that under the offered load with no fragmentation, even a simplistic transmission control algorithm can produce a quality conference. We next force fragmentation to occur by requiring R_1 to use an MTU of 1,500 bytes on the second token ring hop. The result of this change is that R_1 will fragment video messages and place many more packets on the middle segment.

Figure 4.2 shows the performance of the BL algorithm with the logical MTU of the middle token ring set to 1,500 bytes. Since the traffic generator load is the same as the first the experiment, R_1 is still access constrained (*i.e.*, free tokens are scarce) and thus the increase in packets due to fragmentation causes a severe degradation of performance under the BL algorithm. Packet loss is high and hence audio and video frame rates are low. This inevitably leads to gaps in the playout of the streams.

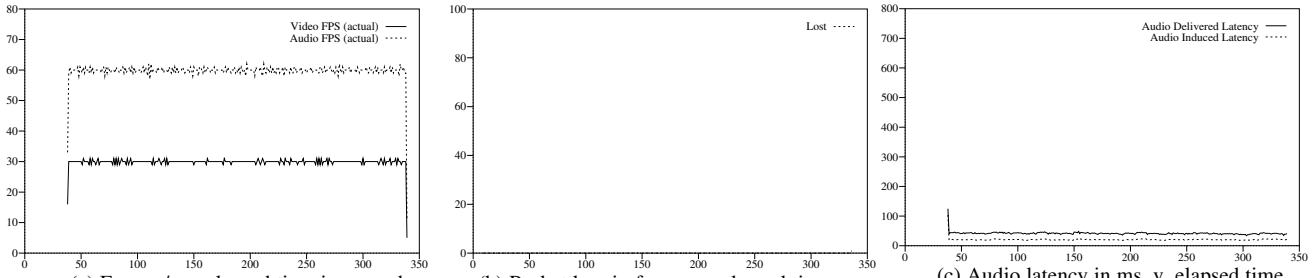
Figure 4.3 shows the performance of the VSO algorithm with the same load and network configuration as in Figure 4.2. The adaptation of the VSO algorithm can be seen in Figure 4.3e&f. In part (e), the *y*-axis is the number of audio frames packaged per message. The line labeled *Audio Frames/Message* plots the number of frames packaged per message over time. The lines labeled *Stereo* and *Mono* show if the conference was sending stereo or monaural audio during a particular interval. The heights of the stereo and monaural lines have no significance (they are boolean measures). The *y*-axis on part (f) is the number of video frames transmitted per second. The line labeled *Video frames/sec* plots the transmitted video frame rate over time. The lines labeled *High Quality*, *Medium Quality*, and *Low Quality* show the coding scheme in effect during each point in the conference. The *y* values of the coding scheme lines also have no significance.

Since the VSO algorithm uses a single operating point for audio, audio is always transmitted in stereo with one audio frame per message. The video stream, however, has three potential operating points and part (f) shows that although the video frame rate remains at 30 frames per second, the algorithm frequently changes the coding quality of the image in an attempt to adapt to the current network conditions. Note that since video scaling changes only the bit-rate of the video stream, there is only an indirect decrease in the number of packets which must be transmitted across the network. The VSO algorithm performs better than the BL algorithm, but overall conference quality is poor. In particular, both the audio and video streams have many gaps on playout. From a network perspective, the video scaling scheme still introduced much more work than the network was able to accommodate, as shown by the high packet losses in Figure 4.3b.

Figure 4.4 shows the performance of the recent success algorithm. RS uses the transmission framework to adapt both the audio and video operating points, resulting in significantly fewer packets transmitted across the network. Figure 4.4e shows that audio is always transmitted in stereo, but the number of audio frames per message changes based upon observed network conditions. Often as many as 10 audio frames are packaged into a single message. This results in higher induced latency in the audio stream (the dashed line in part (c)) compared to the BL or VSO algorithms (which do no buffering), but the delivered latency (the solid line in (c)) remains low. Similarly, part (f) shows the algorithm chose a variety of video frame rates and used all three coding schemes during the course of the conference. The result of this adaptation is a significantly better playout of audio and video with low packet loss after the initial reactive adaptations around time 50.

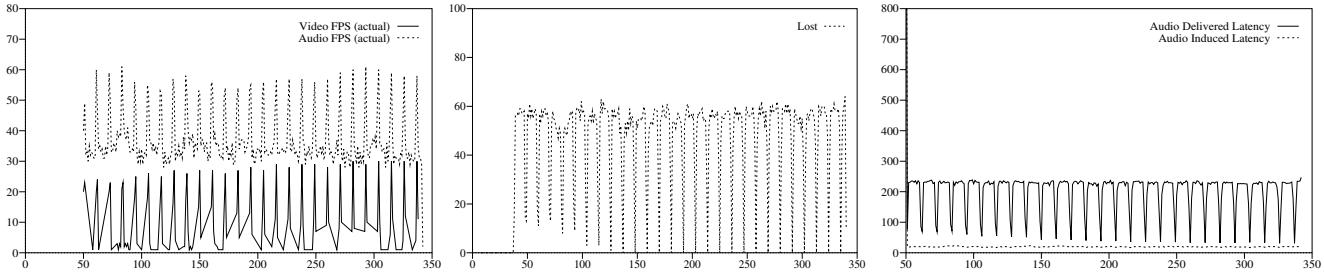
Note that although the RS algorithm uses the heuristic of monitoring message (rather than packet) latencies and interarrival times, and manipulates only the message and bit rates, it can still significantly affect the quality of the resulting conference and can successfully adapt to changes in the traffic load on downstream links even when messages are fragmented.

A second implication of these experiments is that selecting the minimum MTU in the path as the unit of transmission at the source, although it removes the need to



(a) Frames/s v. elapsed time in seconds. (b) Packet loss in frames v. elapsed time. (c) Audio latency in ms. v. elapsed time.

Figure 4.1: Logical token ring MTU = 17,800 — *Baseline* (video latency not shown; similar to audio)



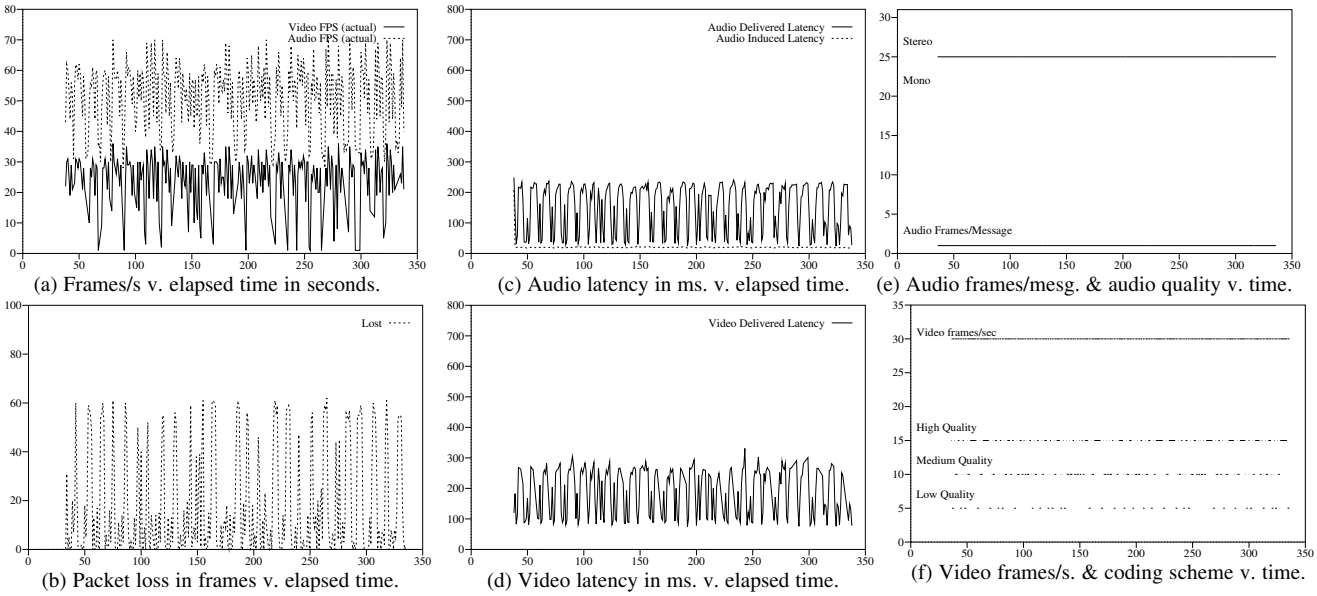
(a) Frames/s v. elapsed time in seconds. (b) Packet loss in frames v. elapsed time. (c) Audio latency in ms. v. elapsed time.

Figure 4.2: Logical token ring MTU = 1,500 — *Baseline* (video latency not shown; similar to audio)

fragment, is not necessarily a good idea. For example, consider the network in Figure 3.2, but with the third token ring segment replaced with an ethernet. Assume the token ring segments use an MTU of 17,800 bytes and the ethernet uses an MTU of 1,500 and that the first token ring and the ethernet are lightly loaded, but the middle network is loaded as in our previous experiments. Should the source elect to transmit audio and video in 1,500 byte messages to match the minimum MTU in the path we expect performance similar to that illustrated in Figure 4.2 (assuming no adaptation) since the access-constrained R_1 would have to process the same number of packets as in Figure 4.2 and thus require the same number of free tokens on the second token ring. On the other hand, if the source transmitted using larger messages and the messages are not fragmented until required at R_2, the effects of the access constraint at R_1 are avoided since R_1 requires the same number of tokens as in Figure 4.1, so performance similar to that shown in Figure 4.1 is likely.

5. Using the Transmission Framework on Ethernet

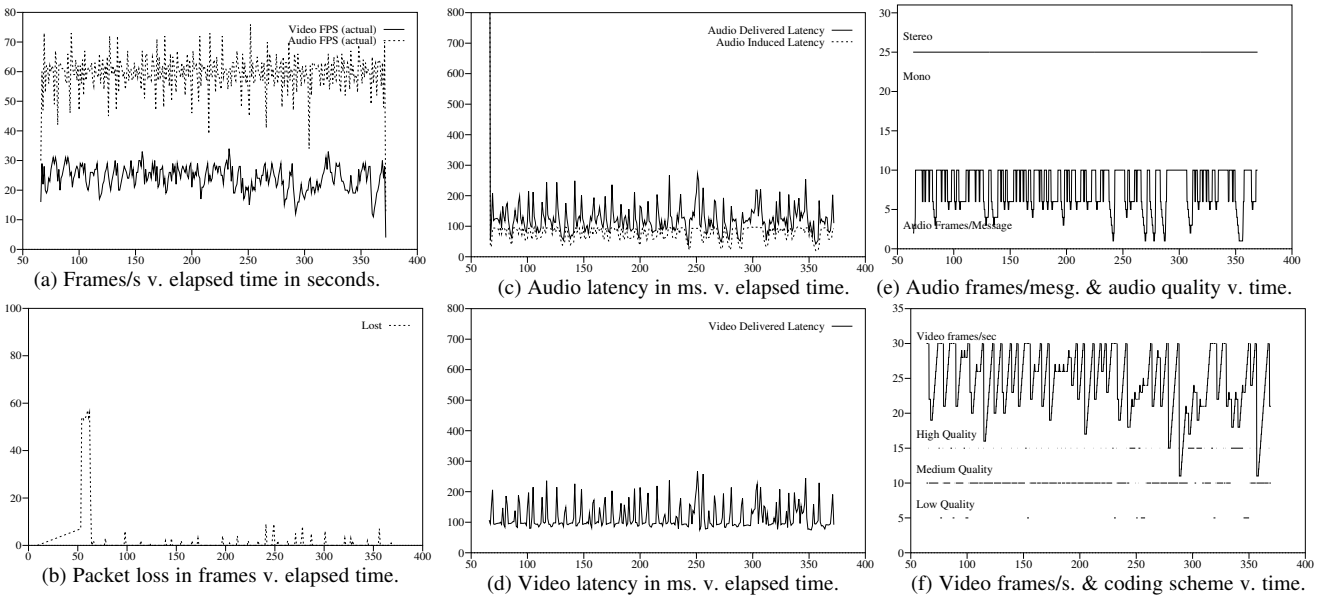
The experiments in Section 4 are useful for showing the relative performance of unfragmented versus fragmented streams, but are somewhat artificial in that R_1 would typically use an MTU higher than 1,500 bytes for the middle token ring network. Consider Figure 3.1 for a more realistic example of fragmentation. In this network, two



(a) Frames/s v. elapsed time in seconds. (c) Audio latency in ms. v. elapsed time. (e) Audio frames/mesg. & audio quality v. time.

(b) Packet loss in frames v. elapsed time. (d) Video latency in ms. v. elapsed time. (f) Video frames/s. & coding scheme v. time.

Figure 4.3: Logical token ring MTU = 1,500 bytes — *Video Scaling Only.*



(a) Frames/s v. elapsed time in seconds. (c) Audio latency in ms. v. elapsed time. (e) Audio frames/mesg. & audio quality v. time.

(b) Packet loss in frames v. elapsed time. (d) Video latency in ms. v. elapsed time. (f) Video frames/s. & coding scheme v. time.

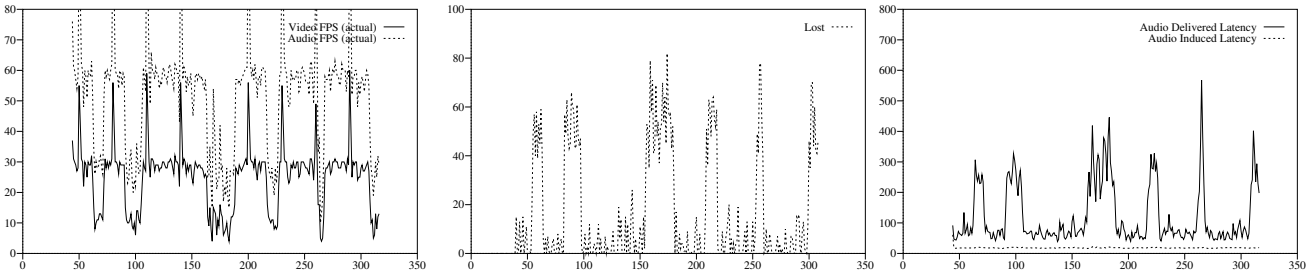
Figure 4.4: Logical token ring MTU = 1,500 bytes — *Recent Success.*

token ring segments are connected with an ethernet. Each token ring segment has a logical MTU of 17,800 bytes, while the ethernet uses a logical MTU of 1,500 bytes.

Using a set of traffic generators, we generated load on the ethernet and measured the performance of the three transmission algorithms. Figures 5.1, 5.2, and 5.3 show the performance of the BL, VSO, and RS schemes, respectively. Both VSO and RS outperform the BL algorithm, which performs poorly. Compared with VSO, RS delivers better audio performance at the expense of video performance and does a better job matching its transmission rate to that supported by the network.

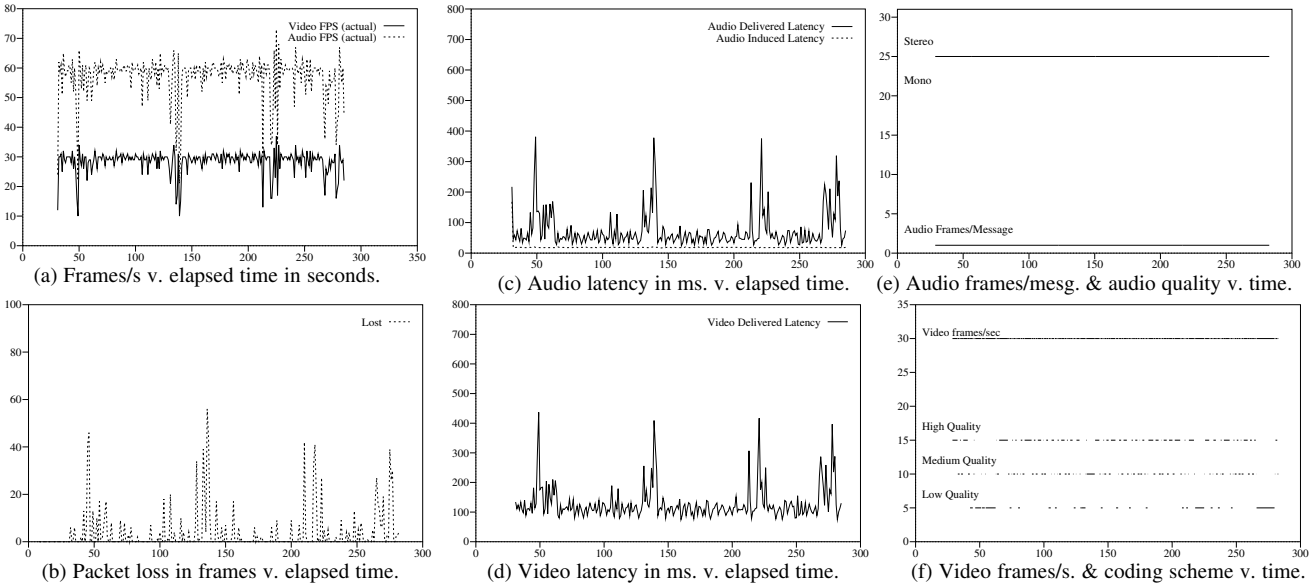
We hypothesized that with smaller video frames (and consequently fewer video packets) and very heavy load on the ethernet, the degree to which RS outperforms both the BL and VSO algorithms should dramatically increase since RS can further adapt to the access constraint on the ethernet (by packing more frames into messages) while the other algorithms cannot. To test this hypothesis, we performed another set of tests with a conferencing system that produces much lower data rates than the system used in the previous experiments. This *Low Bit-Rate* system produces video at up to 30 frames per second at bit-rates of either 384k or 128k bps. Audio generates 60 audio samples per second either in stereo or in monaural, for bit rates of 128k bps and 64 kbps, respectively. Figures 5.4 and 5.5 shows the operating points for the system and their realization for the ethernet hop and the second token ring hop. In particular, Figure 5.4 shows that the potential packet rates of audio and video are equivalent on the ethernet and second token ring hops. Therefore, audio packaging in this system has a greater potential impact on access constraints compared to the system described in Figure 3.4. In addition, in the low bit-rate system, RS can now scale audio bit-rate by 50% by changing from stereo to monaural.

Figures 5.6 and 5.7 show the results of the experiments for the three transmission schemes using the low bit-rate conference. As expected, the RS algorithm produced significantly better overall results than either of the other algorithms. Both the BL and the VSO algorithms have long periods with low frame delivery (and resulting gaps in playout). In addition, both of these algorithms experience periods of extremely high stream latencies (far in excess of the 250 millisecond guideline) and high packet loss. Recent Success does a good job of matching the transmitted packet rates to what the network can support, which leads to low loss and controlled latency in the streams. Audio delivery is good. Video quality fluctuates based on traffic load in the ethernet, but generally varies from adequate to good.



(a) Frames/s v. elapsed time in seconds. (b) Packet loss in frames v. elapsed time. (c) Audio latency in ms. v. elapsed time.

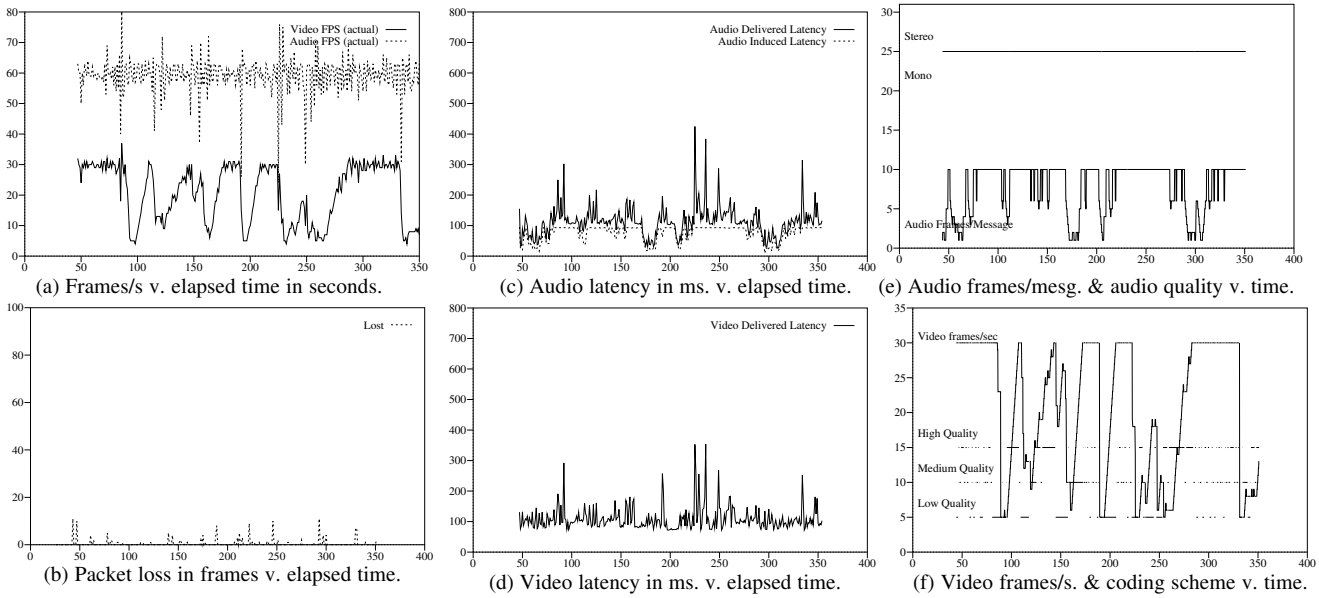
Figure 5.1: Token ring-ethernet-token ring — *Baseline* (video latency not shown; similar to audio)



(a) Frames/s v. elapsed time in seconds. (c) Audio latency in ms. v. elapsed time. (e) Audio frames/mesg. & audio quality v. time.

(b) Packet loss in frames v. elapsed time. (d) Video latency in ms. v. elapsed time. (f) Video frames/s. & coding scheme v. time.

Figure 5.2: Token ring-ethernet-token ring — *Video Scaling Only.*



(a) Frames/s v. elapsed time in seconds.
(c) Audio latency in ms. v. elapsed time.
(e) Audio frames/mesg. & audio quality v. time.
(b) Packet loss in frames v. elapsed time.
(d) Video latency in ms. v. elapsed time.
(f) Video frames/s. & coding scheme v. time.

Figure 5.3: Token ring-ethernet-token ring — *Recent Success.*

6. CONCLUSIONS

One of the major problems experienced with LAN-based video conferencing systems is the degradation of conference latency and fidelity in the face of network congestion. We have developed a transmission control framework for manipulating stream bit- and message-rates in response to network congestion. Although messages may be fragmented as they traverse the internetwork, and fragmentation dilutes the effect of the framework since the rate at which packets arrive at the receiver can at best be only indirectly controlled by the sender, we have demonstrated empirically that an adaptive transmission control algorithm based upon the framework can produce low-latency, high-fidelity conferences over congested internetworks.

In our experiments, the Recent Success transmission scheme, which is based upon our transmission control framework, outperformed the Baseline and Video Scaling Only transmission schemes. The heuristic of monitoring message interarrival times and latency to estimate actual packet throughput and latency appears justified. We conclude that end-to-end adaptive transmission techniques can successfully adapt to many traffic conditions on internetworks of token ring and ethernet LANs, even when application messages are fragmented. Moreover, adapting the stream transmission in both the bit and message rate dimensions gives better results than no adaptation or scaling of bit-rate alone. The performance differences are more pronounced in environments where there is flexibility in the packaging of frames into packets.

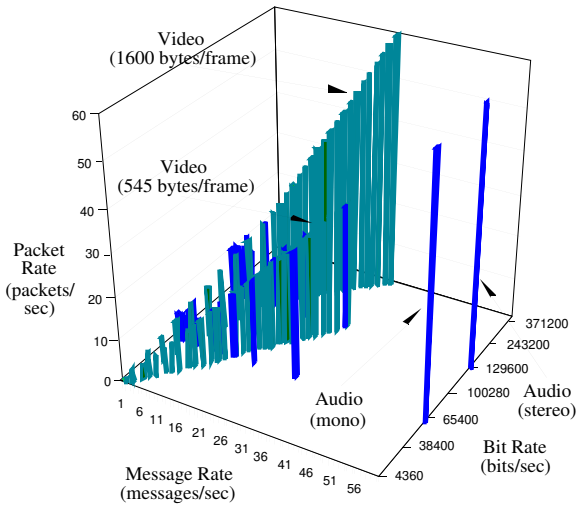


Figure 5.4: Realization of operating points on the ethernet and second token ring for low bit-rate system.

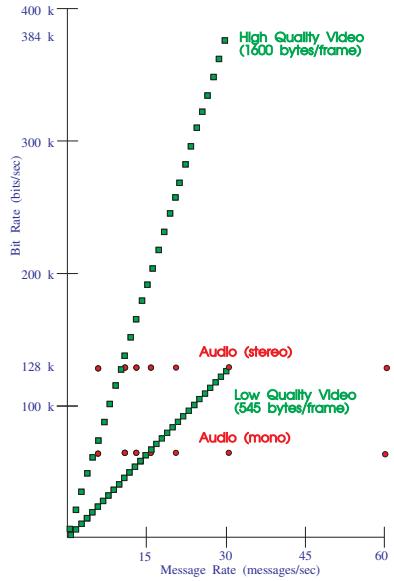
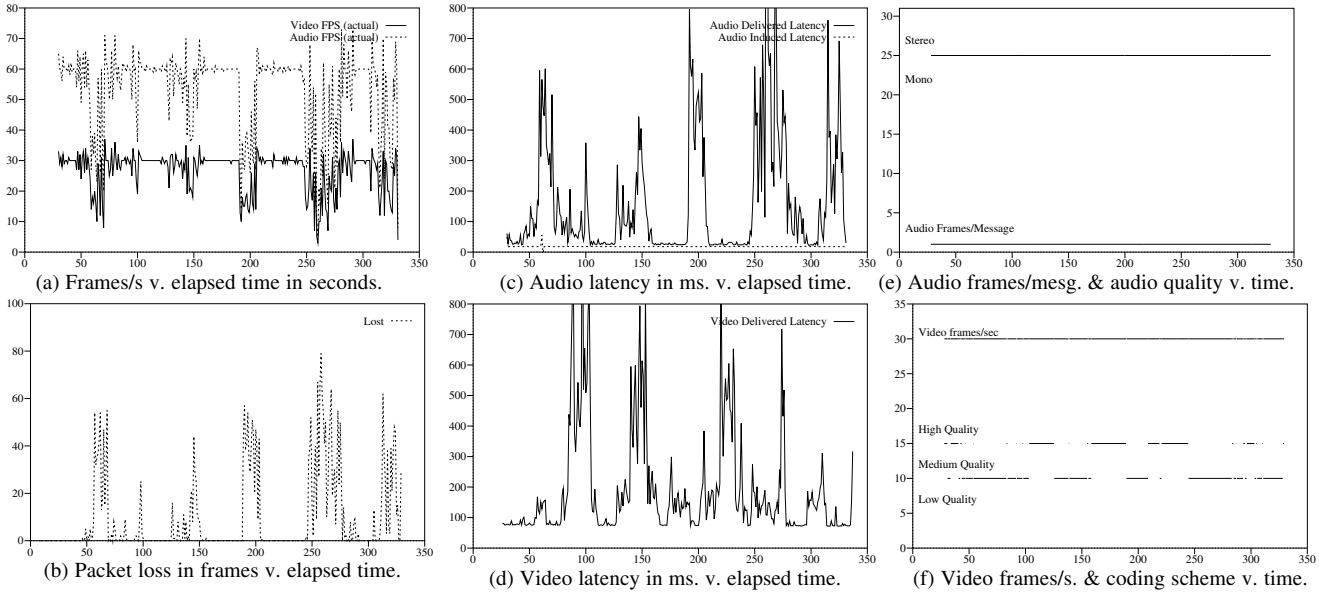


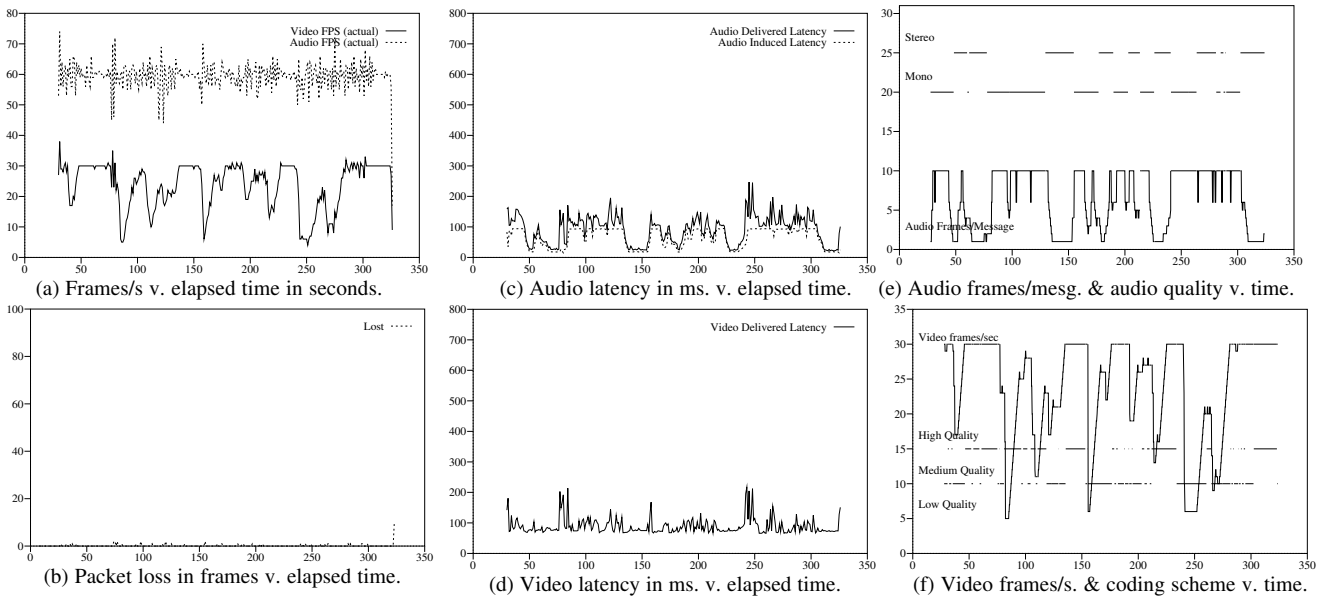
Figure 5.5: Low bit-rate system operating points (video as squares; audio as circles).



(a) Frames/s v. elapsed time in seconds. (c) Audio latency in ms. v. elapsed time. (e) Audio frames/mesg. & audio quality v. time.

(b) Packet loss in frames v. elapsed time. (d) Video latency in ms. v. elapsed time. (f) Video frames/s. & coding scheme v. time.

Figure 5.6: Token ring-ethernet-token ring, low bit-rate conferencing system — *Video Scaling Only.*



(a) Frames/s v. elapsed time in seconds. (c) Audio latency in ms. v. elapsed time. (e) Audio frames/mesg. & audio quality v. time.

(b) Packet loss in frames v. elapsed time. (d) Video latency in ms. v. elapsed time. (f) Video frames/s. & coding scheme v. time.

Figure 5.7: Token ring-ethernet-token ring, low bit-rate conferencing system — *Recent Success.*

7. REFERENCES

1. Bolot, J., Turletti, T., *A Rate Control Mechanism for Packet Video in the Internet*, Proc. of IEEE INFOCOMM '94, Toronto, Canada, June 1994, pp. 1216-1223.
2. Chakabarti, S., Wang, R., *Adaptive Control for Packet Video*, Proc. of IEEE International Conference on Multimedia Computing and Systems 1994, Boston, MA, May 1994, pp. 56-62.
3. Delgrossi, L., *et al.*, *Media Scaling for Audiovisual Communication with the Heidelberg Transport System*, ACM Multimedia '93, Anaheim, CA, August 1993, pp. 99-104.
4. Isaacs, E., Tang, J.C., *What Video Can and Can't Do for Collaboration: A Case Study*, Proc. ACM Multimedia '93, Anaheim, CA, August 1993, pp. 199-205.
5. Talley, T.M., Jeffay, K., *Two-Dimensional Scaling Techniques for Adaptive, Rate-Based Transmission Control of Live Audio and Video Streams*, Proc. of ACM Multimedia '94, San Francisco, CA, Oct 1994, pp. 247-254.
6. Talley, T.M., Jeffay, K., *Audio and Video Transmission Control via Media Scaling and Packaging*, UNC Technical Report, (in submission).
7. Wolf, C., *Video Conferencing: Delay and Transmission Considerations*, in Teleconferencing and Electronic Communications: Applications, Technologies, and Human Factors, L. Parker and C. Olgren (Eds.), 1982.