A Router-Based Congestion Control Scheme For Real-Time Continuous Media*

Kevin Jeffay Mark Parris F. Donelson Smith Terry M. Talley

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

Abstract: Many distributed multimedia applications require real-time network communication services to be effective. Although architectural models for real-time communication on the Internet are evolving (*e.g.*, [2]), none appear to consider the problem of realizing real-time performance on existing local-area networks. This is important as the "first" and "last mile" of a network connection — the portion from a sending workstation to an Internet-bound router on the originating campus, and the portion from an inbound router to a receiving workstation on the destination campus — are likely to employ conventional LAN technologies that cannot reliably provide real-time service. We are investigating the problem of congestion control in a campus-area network. We advocate the placement of a set of simple, generic media adaptations into campus routers for the purpose of ameliorating the effects of congestion along the "first/last mile" of a connection. The adaptations and their implementation are designed to complement proposed schemes such as RSVP for real-time communication through the wide-area network. We demonstrate empirically that our router-based congestion control scheme is capable of supporting an effective, best-effort, real-time transmission service across congested campus networks.

1. Introduction

To be effective, distributed applications that manipulate real-time continuous media require continuous, low-latency delivery of media units from sender to receiver(s). One of the major obstacles to realizing these requirements on local-area networks (LANs) is the fact that the vast majority of today's LANs (*e.g.*, ethernet, token ring, and FDDI networks) provide little support for real-time communication. Without such support, competition for network resources can lead to network congestion that adversely affects the playout of media streams and, in the case of highly interactive applications such as desktop videoconferencing, can render an application useless.

Network congestion manifests itself in two ways depending on the origin and nature of the congestion. First, as congestion in a LAN increases, a workstation attached to that network encounters delays when attempting to access the shared medium to transmit data (*e.g.*, waiting for idle carrier on an ethernet or a token on a token ring). Second, as congestion increases on intermediate networks or at interconnection points such as bridges and routers, packets encounter queuing delays. In cases of severe congestion, packets may be lost at routers or bridges because buffer space is exhausted.

Packet delays and losses have a severe impact on the playout of continuous media such as audio and video. For example, in a desktop videoconferencing system, packet delays increase end-to-end latency and can seriously impair and impede interaction. Similarly, packet loss, and variable packet delays, can lead to gaps in the playout of media streams — intervals in which no audio/video data is played — that disrupt communication between conference participants. Thus, a fundamental problem in the transmission and management of real-time continuous media data on current LANs is ameliorating the effects of congestion to provide adequate throughput and latency.

For real-time multimedia data, there are two dominant approaches to dealing with congestion in LANs. The first is to proactively protect audio/video streams from the effects of congestion by reserving resources (*e.g.*, buffers and CPU cycles at a router) on behalf of these streams to guarantee predictable levels of service. Examples of reservation-based approaches include the Internet protocols ST-II [14] and RSVP [16], buffer management schemes like Stop-and-Go queuing [6], and the Tenet protocol suite [5]. However, reservation approaches fundamentally assume the local-area network itself is capable of assisting in reserving capacity for the transmission of media streams. In contrast, the second approach, the so-called best-effort approach, assumes no direct support for resource reservation in the network and attempts to adapt the media streams to current network conditions. Best-effort transmission schemes adaptively scale (reduce or increase) the bandwidth requirements of audio/video streams to approximate a connection that is currently sustainable in the network. Examples of this approach

* Supported by grants from the IBM and Intel Corporations and the National Science Foundation (number IRIS 95-08514).

In: Proceedings of the Sixth International Workshop on Network and Operating System Support for Digital Audio and Video, Zushi, Japan, April 1996, *to appear*.

include the spatial and temporal scaling mechanisms in the HeiTS system (scaling the video image resolution and video frame rate respectively) [4], the simple temporal scaling in MTP [8], spatial scaling in Sun Media Transport Services [7], scaling of a packetized H.261 stream in IVS [1], and scaling using quadtree compression in the Tenet scaling system [3].

Our focus here is on best-effort transmission across existing LANs. Our overall perspective on the continuous media congestion control problem is that we assume a reservation-based service will (ultimately) provide bounded loss, low-latency, high-throughput wide-area delivery but that this wide-area service will originate and terminate at the edge of a local campus which, for the foreseeable future, will continue to use the existing network infrastructure. We know that even if streams are delivered perfectly to the edge of a destination campus, congestion along the "last mile" — the connection from the destination campus's Internet-inbound/outbound router to the user's workstation — can severely degrade the quality of the stream actually delivered to the destination node [8, 11]. Similarly, if a sending workstation reserves resources in the wide-area network and is capable of generating media streams exactly according to its desired flow-specification, congestion along the "first mile" — the connection from the sending workstation to the campus's Internet-bound router — can similarly degrade the quality of the streams delivered to the real-time-communication-capable network and hence to the destination node.

Proactive transmission techniques such as resource reservation are not likely to work well across the "first/last mile" without the cooperation of all stations attached to the local-area network. This is because the physical links employed today along the first/last mile to the desktop allow end-stations to transmit (or attempt to transmit) at will. Thus at a minimum, applications must, at present, be prepared to adapt the media streams they generate to ameliorate the effects of congestion within the campus.

We are investigating a scheme for adapting audio and video streams within the network, specifically, at the entrance and exit points of the campus network. The approach is based on IP-in-IP tunneling and IP flows. Our thesis is that one can use existing media flow specifications (an RSVP flow spec [16]), existing feedback protocols (RTCP [9]), and simple congestion control mechanisms such as bit-rate and packet-rate scaling [13], to dramatically improve the performance of distributed multimedia application that require real-time media transmission.

The following sections briefly describe our congestion control scheme and its impact on the performance of a desktop videoconferencing system.

2. A Network-Based Congestion Control Scheme

Historically an IP internetwork was viewed as a pure store and forward, connectionless network. New Internet protocols such as ST-II and RSVP represent a fundamental point of departure from this view as they require that state information on sender-receiver pairs (or receiver-sender pairs) be cached in the network, specifically in network elements such as routers and gateways. Moreover, during the course of processing packets, such networks elements will be required to run protocols to maintain this state information as well as execute programs that will use this state information to make resource allocation decisions within the element.

We seek to extend the proposed frameworks for connection-oriented real-time communications to better accommodate multi-access, legacy LANs. We consider a wide-area real-time connection to have three components as shown in Figure 1: (1) the first mile from a sender to the campus router responsible for wide-area connections, (2) the wide-area connection from the originating campus to the destination campus, and (3) the last mile from the destination campus's router to the final receiver. We assume that the wide-area connection is capable of supporting real-time transmission and that each campus router in Figure 1 is capable of managing a real-time connection across the wide-area network.

We are advocating the placement of executable functions in network elements at campus entrance and exit points to (1) adapt media streams to minimize the effects of

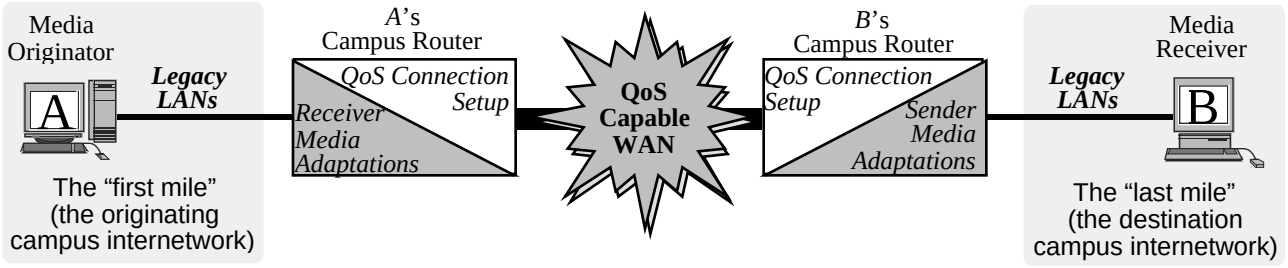

Figure 1: Router-based congestion control concept.

congestion along the first/last mile on these streams, and (2) to provide feedback to (suitably enabled) applications so that they may employ application level adaptations to further help minimize the effects of congestion.

The congestion control scheme we propose consists of a set of media adaptations — a set of functions for manipulating the audio and video streams generated by an application — and an implementation of these adaptations within the network. These are briefly described next.

2.1. Congestion Control Mechanisms

Our starting point is previous work on application-level media adaptations for best-effort delivery of continuous media. Our approach, in essence, is to apply generic forms of proven end-system adaptations within the network. Specifically, we use two adaptations in the network: an elastic queuing mechanism to reduce latency after periods of high congestion [12], and an adaptive packet reformatting mechanism to increase throughput during periods of congestion [13]. The function of each mechanism is best described via an example.

Consider a router that connects a campus internetwork to an Internet service provider as shown in Figure 1 (router "*B*"). Consider a stream of audio and video frames for a videoconference that arrives at this router from a remote site, destined for a receiver on the campus. When the receiver's campus network is congested, packets are queued in the router waiting to enter the campus network. In particular, if the rate at which videoconference packets arrive at the router is greater than the current (instantaneous) rate at which they can be introduced into the campus network, a queue will form in router *B*.

It is possible to ameliorate the effects of this congestion by repackaging media packets so as to reduce the routers demand for network transmissions slots on the campus network. For example, consider a high-quality, high bandwidth videoconferencing system wherein the audio sub-system generates 256 byte samples every 16 ms. Depending on the size of the MTU (maximum transmission unit) of the campus network segment to which router *B* in our example is attached, it may be possible to repackage multiple incoming audio frames into a single new packet and thus reduce the videoconference's demand for transmission slots on the campus network. For example, if the campus backbone network is an ethernet (MTU ≈ 1450 bytes), up to 5 original audio packets may be recombined into a single packet. This reduces the routers demand for access to the campus network on behalf of the audio stream from once every 16 ms to once every 80 ms. If the campus backbone network is a ring, say a 16 Mb token ring (MTU ≈ 16K bytes), over 60 original audio packets may be recombined into a single packet, reducing the demand for access to the campus network to less than once every second. This reformatting technique will clearly increase the end-to-end latency of the media stream (a critical application performance parameter), however, the process can easily be controlled through application-specified formatting parameters that are part of the applications flow specification.

A second way in which the reformatting mechanism can be used to reduce the router's demand for campus network transmission slots is to "reassemble" application messages which have previously been fragmented. For example, for high-quality videoconferencing systems, the size of a video frame (*e.g.*, 2-8K bytes) is typically larger than the MTU of networks such as ethernets and hence will be fragmented if sent (whole) in a network message. If the MTU of the receiving campus backbone is larger than the fragment size of an application's messages arriving at the campus, then multiple fragments may be combined at the entrance to the receiving campus's network into new network messages. Note that this recombination is not envisioned as a complete IP reassembly operation as this would likely overburden the router. It is instead only a combining operation (*i.e.*, a concatenation) as described below.

The second adaptation is an elastic queuing mechanism based on the concept of *queue monitoring* [12]. There are two aspects to the mechanism. The first is a simple change in the way queues within a router are maintained. When a queue overflows, *queued packets* should be discarded rather than *arriving packets*. That is, newer, "fresher" data should be favored over older, "stale" data. Second, if the length of a queue in a router is always greater than k over an interval of time (a function of k), then the congestion on the campus network can be viewed as being at "equilibrium" with respect to the rate at which packets are arriving from the external world. That is, the rate at which packets are arriving is roughly equal to the rate at which they can be introduced into the campus network. (A queue of length k in the router will not overflow or underflow.) In this case, packets of application media can be discarded, thereby lowering the end-to-end latency for the stream. For example, if over a sufficiently long interval, there are always at least 3 audio packets queued in the router, deleting the oldest audio packet currently in the queue will lower the latency in the audio stream (by decreasing the queue length). Moreover, deleting a packet, in theory, will not effect the receiver's ability to playout media continuously. This is because a queue of audio packets still exists at the router and hence in the short term, the rate at which the router will introduce audio packets onto the campus network, and

hence the rate at which the receiver can playout media samples, is unaffected.

The preceding example was primarily concerned with ameliorating the effects of congestion on the last mile of the real-time connection. The adaptations described above, however, have similarly significant utility when deployed along the first mile. In particular, the first and last mile problems are duals of one another. In the first mile problem, the sending workstation plays the role of router in the last mile problem as it is the entity that requires access to the campus network to transmit data. Thus the sending workstation is responsible for adaptively packaging data when congestion exists along the first mile. The first mile router (router *A* in Figure 1) plays the role of the final receiver in the last mile problem. It provides feedback to the sender to report on the success of the current adaptation strategy and control its operation.

2.2. Implementation Issues

There are several key design issues in realizing our congestion control scheme. These include:

- where precisely adaptations will be placed in the network and how they will interoperate with existing protocols,
- how the data required to activate and control the adaptations is delivered to the implementation site,
- what specific mechanisms will be used to realize the adaptations and how they will interoperate with existing mechanisms in network elements.

Although we have described our congestion control scheme in terms of a router-based implementation, the most straightforward placement of the adaptation function we propose would be in an application-specific "gateway" located in the network at points where adaptations are expected to be needed. In the cases considered in this paper, this location would be at the egress points from source end-nodes (*e.g.*, in the IP layer in user's workstations) or at interconnection points in the network where congestion is expected on the next-hop network. Since protocols specific to the application already flow through such a gateway and the processing applied to each packet is well defined, it is relatively easy to extend the protocols and processing for these adaptations

As an example, consider the real-time transport protocol RTP [9]. RTP explicitly includes the notion of "translator" elements which can be considered as "intermediate systems" at the RTP level and located in the path followed by all packets in a flow from source to receiver. The translator elements also participate in the flows from receiver to source (RTCP) that report on how well the flows are received. It would be straightforward to include the adaptations described here in properly located RTP translator nodes and use the RTCP sender/receiver reports (possibly extended) to control the adaptations. The key advantage of locating adaptations in an application-specific gateway is that it already implements a defined protocol and is positioned in the path taken by packets. The obvious disadvantage is that it is a static (manual) placement and requires preplanning to determine the network points at which a gateway may be required. There may also be a significant cost for introducing a gateway which is needed only to deal with occasional congestion.

A more interesting possibility is to introduce (or activate) adaptations dynamically in the path followed by packets from a source to a receiver through conventional IP routers. In this case we can avoid introducing an application-specific node such as the RTP translator described above, but still be able to alter the handling of packets for certain flows as needed. In a typical IP router, the handling for each packet is highly optimized (often implemented with hardware assists) to achieve minimal delay/maximal throughput. Performing adaptations in this "fast-path" processing for each packet has to be considered with great care. Specifically, the introduction of adaptations should not require creation of "hard-state" that would make the network operation less robust in case of failures, and should not add appreciably (*i.e.*, only a few instructions) to the processing for each packet.

One plausible design we are evaluating is a generalization of the implementation reference model for routers underlying the Internet integrated services architecture [2]. This model, illustrated in Figure 2, cleanly separates components along the router's data forwarding path from background processes that are protocol processing agents. The agent processes participate in various protocols and create state information in "databases" that are shared with the components on the data forwarding path. For example, reservation and admission control agents participate in the RSVP protocol to create traffic-control state used by the classifier and scheduler components on the data forwarding path. The classifier component maps each incoming packet into some class which usually determines the queue it joins for scheduling onto an outbound link.

We can implement the adaptations considered here by including in this framework an agent that participates in a protocol (perhaps a generalization of RSVP) for marking the path taken by a "flow" of packets from source to receiver and introducing "soft-state" — state that times out if not refreshed — at nodes along the path. This agent can also participate in protocol flows from the receiver that provide feed-back necessary to control the adaptations. To couple adaptations into the data forwarding path, we could

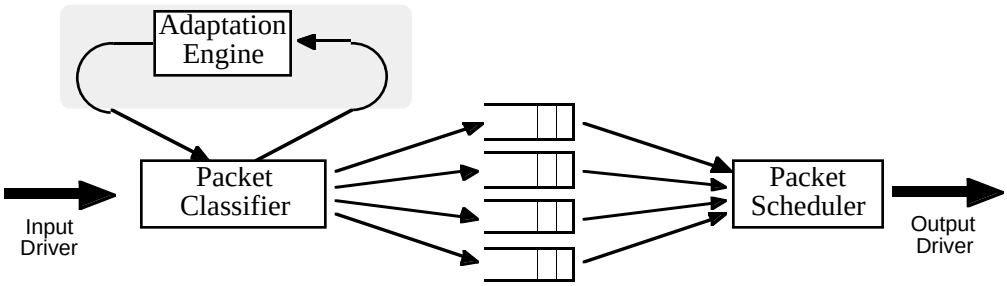

Figure 2: Modified Integrated Services router reference architecture (adapted from [2]).

make a minor addition to the classifier/enqueueing mechanism to allow it to divert packets to the adaptation agent only when the state information indicates certain flows require adaptation. The adaptation agent can return the packets (possibly repackaged for IP-in-IP tunneling [11]) to their normal destination queue (see Figure 2).

3. Preliminary Experiments

We have implemented the congestion control adaptations described in Section 2 in an IP router of our own construction and have begun a preliminary evaluation of both the overhead of hosting the adaptations in a router and the impact of the adaptations on the performance of audio and video streams that pass through the router. Here we only report on the latter set of experiments.

To evaluate the implementation, we constructed a network in our laboratory to simulate the wide-area connection illustrated in Figure 1. A source machine executes a videoconferencing application [8] that generates from 0.5 to 2 Mb/s of audio and video traffic. The output of the application is transmitted across token-ring and ethernet networks to a router. This collection of elements is collectively referred to as "campus *A*." Campus *B* is similar except that only token-ring networks are used. A sink machine on a campus *B* network receives and displays the audio and video sent by *A*. The campus *A* and campus *B* routers are connected by a private (uncongested) ethernet to simulate a real-time wide-area interconnection. Additional workstations on the campus *A* and campus *B* internetworks generate traffic to induce congestion within each campus. By controlling the traffic generators we are, in principle, able to create reproducible traffic patterns and thus run controlled experiments.

For each experiment, during one second intervals we measure the rate at which audio and video samples ("frames") arrive at *B*, and the end-to-end (sender to receiver) latency of each frame. In addition to these measures, the sender and the campus *B* router are instrumented to report how they adapt the audio and video streams over time. In all the experiments reported here, background traffic is generated such that the primary bottleneck in the overall system is the token-ring backbone on campus *B*. In each experiment we begin with idle networks and generate fixed length periods of "low," "medium," and "high" levels of background traffic.

Figure 3 shows the performance of the videoconference without any media adaptation or congestion control. For this execution of the videoconferencing system, 60 audio samples and 30 video frames are generated and transmitted by the sender each second. The results are considerable packet loss, high latency, and extreme delay-jitter. Qualitatively, the media that is played-out at the receiver is of extremely low quality. Video is nearly motionless and audio is incomprehensible. The jumps in audio latency correspond to the points where the level of background traffic is changed. (Note that with respect to latency, video performance appears to be superior that of audio. This is merely because virtually all of the video data is being lost.)

Figure 4 shows the results of two additional experiments. One in which end-system only adaptations are applied and one in which router-based adaptations are applied. The first of these experiments attempts to demonstrate the extent to which applications can adapt to congestion on their own. In this case, based on feedback messages from the receiver, the sender is dynamically adapting both the video bit-rate it generates and the manner in which it packages audio samples into network messages. (See [13] for additional details on these techniques.) For example, congestion is perceived by the receiver to be high enough that the sender transmits 10 audio samples at a time (thus inducing 160 ms worth of latency into the audio stream as can be seen on the audio latency plot), and frequently scales back the video stream to 5 or 10 frames per second with the quality of each frame (image resolution) reduced to further reduce the bit-rate.

Experiment 2 shows that the sender and receiver alone are capable of ameliorating much of the congestion present in the network. In particular, conference throughput is significantly better as packet loss is nearly eliminated.

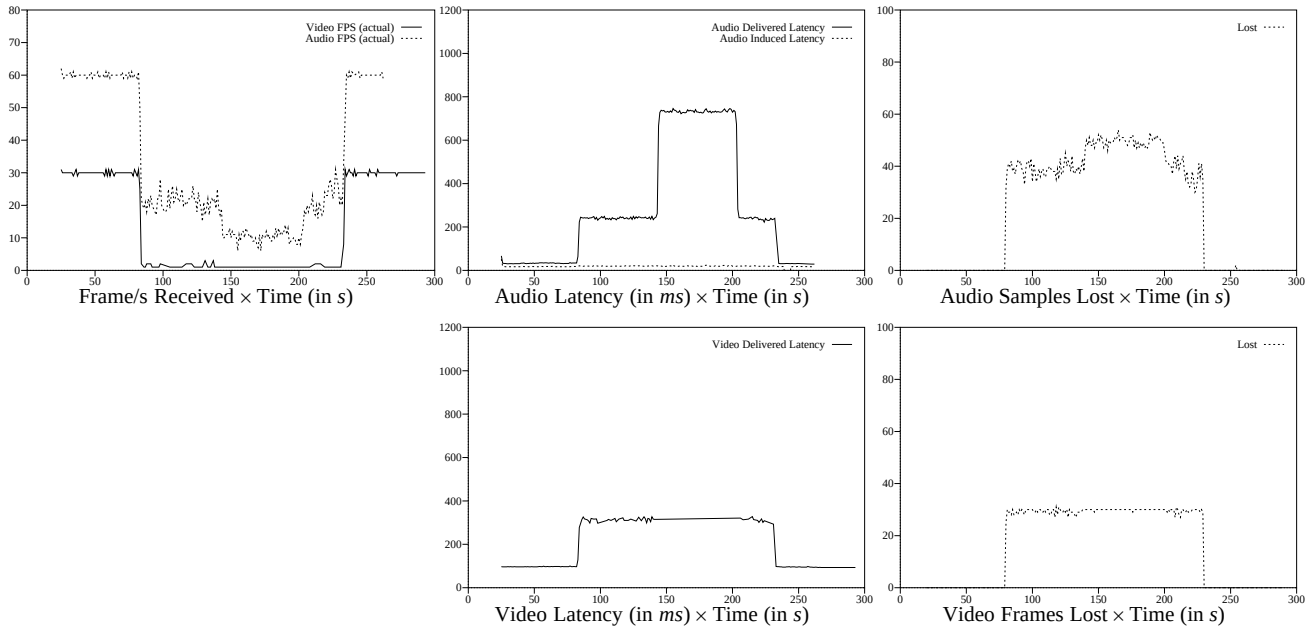

Figure 3: Experiment 1 — Conference performance with no media adaptations.

However, conference latency and delay-jitter are quite high. Qualitatively, the conference is acceptable and comparable with conferences run over the Internet today.

Experiment 3, compares the performance of our router-based congestion control scheme (with no end-system adaptations) to the baseline and the end-system only adaptation experiments. In this experiment a RTCP-like agent in the receiving machine feeds back information on current performance levels to the campus *B* router. The router uses this information to detect changes in latency. Changes in latency or evidence of packet loss are used as indicators of congestion. When congestion is suspected, the router adaptively delays audio and video packets incoming from the wide-area network for concatenation into larger packets which are then "tunneled" from the router to the receiver. The tunneled packets are unpacked and reassembled (if necessary) at the receiver and passed to the conference application.

The end result is that latency, and delay-jitter are dramatically lower and packet loss is eliminated. Moreover, *more* data is successfully being transmitted — 30 high-quality video frames and 60 high-quality audio samples are received each second. The result is a fully functional videoconference that resembles an analog system.

The router-based adaptations outperform the end-system only adaptations primarily because they are better able to directly influence the cause of congestion — the fact that router *B* can not forward packets onto the *B* campus as fast as it needs to. End-system adaptations can influence this problem (positively) by reducing the amount of data transmitted, however, this effect is diluted by the fact that packets, in particular video packets, are fragmented before they reach the *B* campus. Thus, if the bottleneck at *B* is access to the shared media (the *B* campus backbone), reducing the data transmitted at the sender is effective only if it results in video packets sent by *A* that are small enough so that after fragmentation on *A*'s campus, the number of packets seen at router *B* is reduced.

4. Summary and Conclusions

Ultimately some form of admission control and resource reservation is required in the wide-area network to realize effective real-time communications. However, current proposals for real-time transmission of continuous media have not addressed the problem of managing flows across legacy local-area networks. Without some form of coordinated, centralized control of end-stations, one cannot ensure that the effort expended in delivering media in real-time across the wide-area network will be rendered ineffective by congestion at the entry and exit points of the network.

We advocate a best-effort approach for congestion control on the confederations of local-area networks that are typically found on a campus along the "first mile" from the desktop and the "last mile" to the desktop. Specifically, we are experimenting with a simple packet concatenation and elastic queuing mechanism and have shown that they can dramatically improve the performance

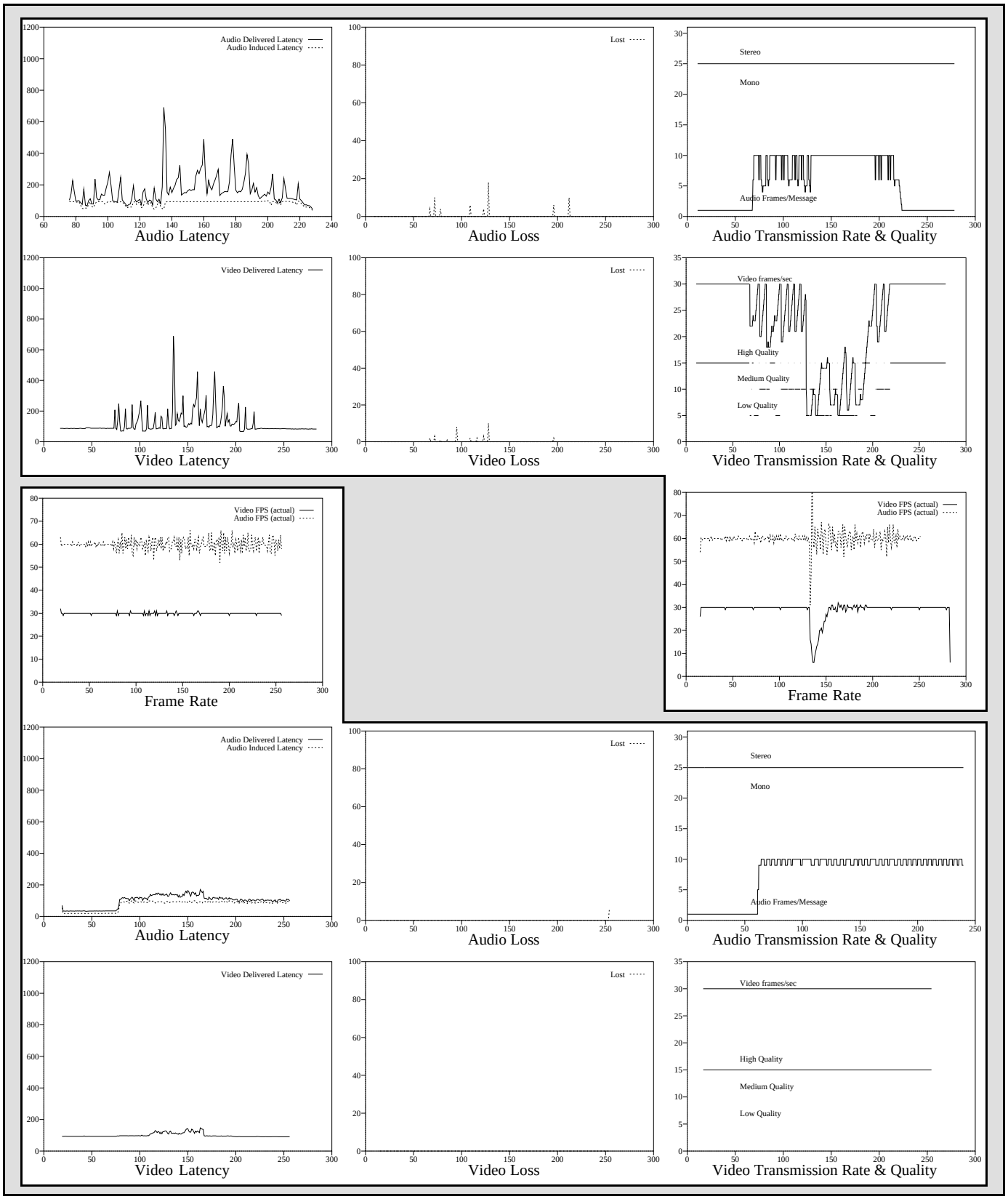

Figure 4: Experiment 2 (top) — conference performance with end-system only adaptation (media scaling and packaging). Experiment 3 (bottom) — conference performance with only router-based adaptation (packet aggregation).

of distributed multimedia applications. Because our adaptations can be applied independently at sender's and receiver's campus's and are potentially completely implementable within the IP layer of the protocol stack (and hence need not involve the sending or receiving application directly), the techniques scale well in multicast environments.

Our approach is simple and we believe can be made to work well with both proposed proactive schemes for real-time communications in the wide-area network, and application-level best-effort media scaling techniques.

5. References

[1] Bolot, J., Turletti, T., *A Rate Control Mechanism for Packet Video in the Internet*, Proc. IEEE INFOCOMM '94, Toronto, Canada, June 1994, pp. 1216-1223.

[2] Braden, R., D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," IETF RFC-1633, July 1994.

[3] Chakrabarti, S., Wang, R., *Adaptive Control for Packet Video*, Proc. IEEE International Conference on Multimedia Computing and Systems 1994, Boston, MA, May 1994, pp. 56-62.

[4] Delgrossi, L., Halstrick, C., Hehmann, D., Herrtwich, R., Krone, O., Sandvoss, J., Vogt, C., *Media Scaling for Audiovisual Communication with the Heidelberg Transport System*, Proc. ACM Multimedia '93, Anaheim, CA, Aug 1993, pp. 99-104.

[5] Ferrari, D., Banjea, A., and Zhang, H., *Network Support for Multimedia: A Discussion of the Tenet Approach*, Computer Networks and ISDN Systems, Vol. 26, No. 10 (July 1994), pp. 1267-1280.

[6] Golestani, S.J., *A Stop-and-Go Queuing Framework for Congestion Management*, Proceedings of ACM SIGCOMM '90, Philadelphia, PA, September 1990, pp. 8-18.

[7] Hoffman, Don, Spear, M., Fernando, Gerard, *Network Support for Dynamically Scaled Multimedia Streams*, Network and Operating System Support for Digital Audio and Video, Proceedings, D. Shepard, *et al* (Ed.), Lecture Notes in Computer Science, Vol. 846, Springer-Verlag, Lancaster, UK, November 1993, pp. 240-251.

[8] Jeffay, K., Stone, D.L., Smith, F.D., *Transport and Display Mechanisms For Multimedia Conferencing Across Packet-Switched Networks*, *Computer Networks and ISDN Systems*, Vol. 26, No. 10, (July 1994) pp. 1281-1304.

[9] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V, *RTP: A Transport Application For Real-Time Applications*, Internet Engineering Task Force, Audio-Video Transport Working Group, RFC-1889, January 1996.

[10] Simpson, W., *IP in IP Tuneling*, Internet Engineering Task Force, Network Working Group, RFC 1853, October 1995.

[11] Stone, D.L., *Managing the Effect of Delay Jitter on the Display of Live Continuous Media*, Ph.D. dissertation, University of North Carolina at Chapel Hill, 1995.

[12] Stone D.L., Jeffay, K., *An Empirical Study of Delay Jitter Management Policies*, *ACM Multimedia Systems*, Vol. 2, No. 6, (January 1995) pages 267-279.

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

[14] Topolcic, C. (Ed.), *Experimental Internet Stream Protocol, Version 2 (ST-II)*. Network Working Group, RFC 1190, IEN-119, CIP Working Group, October 1990.

[15] 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.

[16] Zhang, L., Deering, S., Estrin, D., Shenker, S., Zappala, D., *RSVP: A New Resource ReSerVation Protocol*, *IEEE Network*, Vol. 5, No. 5 (September 1993), pp. 8-18.