Two-Dimensional Scaling Techniques For Adaptive, Rate-Based Transmission Control of Live Audio and Video Streams*

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 obstacles facing designers of video conferencing systems is the problem of ameliorating the effects of congestion on interconnected packet-switched networks that do not support real-time communication. We present a framework for transmission control that describes the current network environment as a set of sustainable bit and packet transmission-rate combinations and show that adaptively scaling both the bit and packet-rate of the audio and video streams can reduce the impact of congestion. We empirically demonstrate the validity of adapting *both* packet and bit-rate using a simple feedback mechanism and simple adaptation heuristics to deliver audio and video streams suitable for low-latency, high-fidelity playout.

1. INTRODUCTION

There is currently great interest in the problem of transmitting digital audio and video in real-time across local-area networks (LANs). For example, to be effective, systems that support real-time collaborative work, such as desktop video conferencing systems, require continuous, low-latency delivery of audio/video data. Realization of these requirements in a distributed system is complicated by 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. Thus, a fundamental problem in the transmission and management of live audio and video data on these LANs is that of ameliorating the effects of congestion.

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 case of severe congestion, packets may be lost at routers or bridges because buffer space is exhausted or the CPU is saturated. In either case, packet delays and losses induced by congestion have a severe impact on the playout of audio/video data. For example, packet delays increase end-to-end latency and can seriously impair and impede interaction in a video conference. Similarly, variable packet delays and losses can lead to gaps in the playout of media streams (*i.e.,* intervals in which no audio/video data is played) that are frequent enough to again hinder interaction in a conference.

*This work supported in parts by the National Science Foundation (grant number CCR-9110938), and the IBM and Intel Corporations.

With respect to real-time multimedia traffic, 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. Examples of such reservation based approaches include the Internet protocols ST-II [6] and RSVP [8]. The second is to adaptively scale (reduce or increase) the bandwidth requirements of audio/video streams to match that currently sustainable in the network. Examples of this approach include the spatial and temporal scaling mechanisms in the HeiTS system [1], and the simple temporal scaling in MTP [3].

Our thesis is that on campus-sized internetworks, one can effectively manage streams of live audio and video data by adaptively scaling the streams in both the bit-rate and packet-rate dimensions. For video conferencing systems, we present a framework for controlling the transmission of media streams via the manipulation of these two rates. For each stream this framework describes the capabilities of a video conferencing system as a set of bit-rate × packet transmission-rate pairs called *operating points.* Each operating point for a stream specifies a bit and packet-rate that the conferencing system is capable of generating. The framework also describes how human perception, physical network characteristics, and network congestion, limit the set of operating points which may actually be used by the video conference at any point in time. The framework characterizes the perceived network environment as a set of *feasible* operating points which is a subset of the full set of operating points. The elements of the feasible set are operating points which are sustainable under the current network conditions and result in the delivery of audio/video streams with adequate throughput and latency for video conferencing. The conference source uses feedback from the destination to periodically estimate the set of feasible operating points. The current operating point is then selected for each stream.

We demonstrate empirically that for interconnected token ring networks, one can effectively characterize network congestion with our framework and ameliorate its effects by judicious selection of operating points. Moreover, we show that scaling along both bit and packet-rate axis provides higher performance conferences than those obtained without any scaling or by simply scaling bit-rate. While not a panacea, we believe multi-dimensional scaling to be a promising technique for managing live audio/video data across the last mile to the desktop.

The remainder of the paper is organized as follows. Section 2 presents our framework for transmission control of media streams in greater detail. Section 3 demonstrates empirically that there exist cases of congestion wherein conferences with acceptable performance can be realized by scaling solely along the bit-rate dimension, and cases of congestion wherein acceptable conferences can be realized by scaling solely along the packet transmission-rate-dimension. Section 4 illustrates the use of the transmission framework in an experimental conferencing system.

In: Proc. of the Second ACM International Conference on Multimedia, San Francisco, CA, October 1994, ACM Press, pp. 247-254.

We present techniques for estimating the set of feasible operating points and heuristics for selecting a bit and packet-rate given this estimate. Some preliminary performance results for this system are reported. We summarize our results in Section 5.

2. A FRAMEWORK FOR AUDIO/VIDEO TRANSMISSION CONTROL

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.

To simplify the discussion we consider media transmission in only a single direction and assume audio and video frames are transmitted in separate network packets. Multiple frames from the same stream may be transmitted in a single packet in which case we assume each network packet initially contains an integral number of frames. The number of frames that may be transmitted in a packet is therefore a function of the frame size and the MTU (maximum transmission unit) of the network to which the sending machine is directly attached. If f_s is the current frame rate for stream s then the packet-rate p_s for the stream satisfies $p_s = f_s/k$ and packets are delivered to the network interface every k/f_s time units, for some integer $k \geq 1$. Each packet contains $kb_s/f_s = b_s/p_s$ bits.

At well-defined points in time, the sender chooses a bit and packet 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 the frame rate (*e.g.*, by temporal scaling). The transmission-rate may be changed by changing the number of frames in a single packet.

We characterize each media stream s in a conference by the set OP^s of *operating points* in a bit-rate × packet transmission-rate space. For stream s, $(b_s, p_s) \in OP^s$ if and only if the conference is capable of generating b_s bits/s and partitioning s into p_s packets/s. For example, for the conferencing system used in this work (described in Section 3.1), Figure 1a shows the set of operating points for the audio and video streams. For video, this system has a choice of three coding schemes. Each generates 30 frames per second. Each video frame is always sent in a separate network packet, however, since each scheme codes each frame separately (*i.e.*, coding is frame-independent), we can transmit video at any packet (frame) rate from 1-30 packets per second. For example, transmitting 30 packets per second yields full-motion video; 15 packets per second yields half-motion video, *etc.* Thus the set of video operating points contains 90 points, 30 per coding scheme. For audio, the system is capable of generating only 1 bit-rate. Sixty 250 byte audio frames are generated each second and the system can transmit 60, 30, 20, 15, 12, 10, or 6 packets per second (corresponding to 1, 2, 3, 4, 5, 6, or 10, audio frames per packet). These operating points assume the system is attached to a 16 Mbit token ring (which has an MTU of approximately 17,800 bytes).

The key problem is to chose a *feasible operating point*. For stream s, an operating point is said to be *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 a gross quality issue. Even in the absence of congestion, not every operating point may be desirable. 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, the conference application may be capable of transmitting audio at a such a low packet-rate (*e.g.*, one packet every second) that the latency induced by buffering audio frames until they are ready to be transmitted makes it impossible for users to effectively interact with each other. A general discussion of conference quality issues is beyond the scope of this paper. For our purposes we simply characterize quality in terms of latency and throughput constraints. The effect of these constraints on the choice of a feasible operating point is discussed in Section 2.1.

The second requirement concerns the fundamental problem of transmission control. The network limits our choice of an operating point in two ways. First, for stream s operating point (b_s, p_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 forward) b_s bits/s. Second, in the presence of congestion, it may be the case that there exists insufficient capacity to sustain a b_s bit/s stream that is partitioned into p_s packets/s. The effect of these constraints on the choice of a feasible operating point is discussed in Section 2.2.

2.1 Constraints Imposed by Human Perception

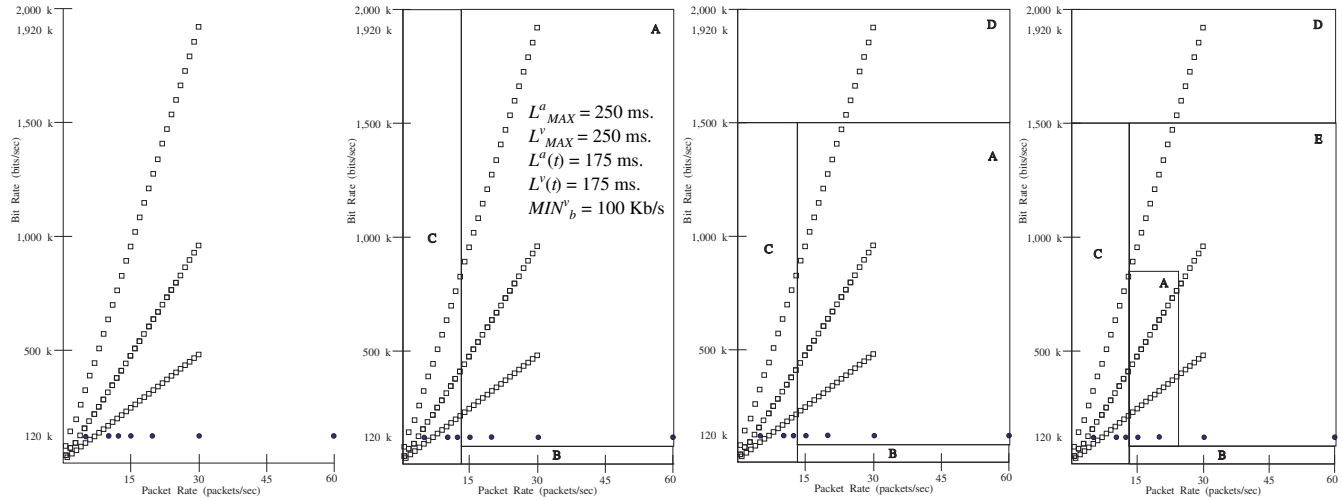
To quantify the constraints of human perception on the set of feasible operating points, let $L^s(t)$ be the observed end-to-end latency of a packet for stream s at time t. End-to-end latency is defined as the difference between the time a packet arrives at the receiver and the time the packet was delivered to the network interface on the sending machine. Let L^s_{MAX} be the maximum end-to-end latency tolerable for stream s. To ensure acceptable latency, individual s frames may be buffered at the sender for at most $MAX^s_L(t) = L^s_{MAX} - L^s(t)$ time units before they must be transmitted. Therefore, at time $t + 1$, each stream s frame must be transmitted within $MAX^s_L(t)$ seconds and hence at most $\lceil f_s MAX^s_L(t) \rceil$ frames may be transmitted in a packet and packets must be generated no slower than the rate of 1 every $\lceil f_s MAX^s_L(t) \rceil / f_s$ seconds.

Let MIN^s_b be the minimum bit-rate that is required for acceptable fidelity playout of stream s. Let $COP^s(L^s(t)) = \{(b_s, p_s) \mid (b_s, p_s) \in OP^s, 1/p_s \leq MAX^s_L(t) \wedge b_s \geq MIN^s_b\}$ be the set of *candidate operating points*. Operating points not in COP^s cannot be used at the time t as they will inherently lead to either unacceptable latency or fidelity in stream s. The sets of candidate operating points for a set of latencies is illustrated graphically as region A in Figure 1b.

2.2 Constraints Imposed by the Network

The set of feasible operating points is constrained by the bottleneck element(s) in the network. A bottleneck element can be either a transmission link or an interconnection point between two links. A network element can be a bottleneck for two reasons. First, the element may not have the physical capacity to process a data stream generated at a particular operating point. Second, because of congestion, the element may not have sufficient capacity at the present time to process a data stream.

Let r_{min} be the bit-rate at which data is transmitted on the slowest network that carries conference traffic. In order for stream operating point (b_s, p_s) to be feasible, clearly $b_s \leq r_{min}$, or $1 - b_s/r_{min} \geq 0$. If $b_s > r_{min}$ then we say that a physical capacity constraint exists. For example, for the network configuration shown in Figure 2, operating points in region D in Figure 1c are excluded from consideration because of a physical capacity constraint.



(*a*) Audio and video operating points. Squares are video operating points, circles are audio operating points.

(*b*) Exclusion of operating points due to perceptual constraints. Points in *C* excluded because of latency constraints ($1/p_s > MAX^s_L(t)$); points in *B* because of throughput constraints ($b_s \le MIN^s_b$).

(*c*) Exclusion of operating points due to physical capacity constraints (region *D*). Here r_{min} = 1.544 Mb/s.

(*d*) Exclusion of operating points due to congestion constraints (region *E*).

Figure 1: The bit-rate × packet transmission-rate space.

To motivate our modeling of the effects of congestion on the selection of an operating point, consider the network segment to which the conference sender is directly attached. Let r_1 be the physical transmission-rate (in bits/s) of this link. If (b_s, p_s) is the current operating point for stream *s*, then each *s* packet requires at least $b_s/p_s r_1$ seconds for transmission across this link. When there is traffic on the network the time required to transmit a packet can be decomposed into a queueing component (time spent waiting to access the physical medium) and a transmission time component. Let $MA_1(t)$ represent the medium access time at time *t* for the first network. If $MA_1(t) + b_s/p_s r_1 > 1/p_s > b_s/p_s r_1$, then we say that the conference is *constrained by congestion* from operating stream *s* at point (b_s, p_s) at time *t*. That is, if there does not exist a physical capacity constraint ($1/p_s > b_s/p_s r_1$) but the medium access time plus the packet service time is greater than the interarrival time of packets ($MA_1(t) + b_s/p_s r_1 > 1/p_s$), then packets cannot be transmitted in real-time. A queue of packets will build up at the network interface and packets will eventually be dropped. For example, at time *t*, operating points in region *E* in Figure 1d are excluded from consideration because of (hypothetical) first-network congestion constraints. The congestion relation may be rewritten as

$$MA_1(t) > \frac{1-\frac{b_s}{r_1}}{p_s} > 0. \qquad (2.1)$$

This relation, bounds the queueing delay at the first network for infeasible operating points. If the packet generation period $1/p_s$ minus the transmission time of the packet on the slowest network link is less than the media access time on the first network then the operating point (b_s, p_s) will not be sustainable. Thus, eliminating points satisfying (2.1) from $COP^s(L^s(t))$ yields a superset of the set of feasible operating points. Figure 1d illustrates this superset (region *A*). For stream *s* at time *t*, the set of feasible operating points $FOP^s(t)$ is what remains after we remove all points from the set OP^s that have been excluded because of perceptual, physical capacity, and congestion constraints. To completely specify $FOP^s(t)$ would require that we develop expressions similar to (2.1) for each network element in the path from sender to receiver and thus completely characterize the congestion in the network at time *t*. To actually use these relations to compute $FOP^s(t)$ at run-time, however, would require information on MTUs, CPU speeds (at interconnection points), buffer usage, and medium access times for all network elements.

Our thesis is that one can bound $FOP^s(t)$ effectively by modeling the entire internetwork of interest as a single virtual network with a transmission-rate of r_{min} (the transmission-rate for the slowest physical network that carries conference traffic) and a medium access time at any point in time greater than or equal to $MA_1(t)$, and less than or equal to $L^s(t)$. Let

$$NB^s(t) = COP^s(L^s(t)) \cap \{(b_s, p_s) \mid \frac{1-\frac{b_s}{r_{min}}}{p_s} > MA_1(t)\}$$

$$nb^s(t) = COP^s(L^s(t)) \cap \{(b_s, p_s) \mid \frac{1-\frac{b_s}{r_{min}}}{p_s} > L^s(t)\}$$

$NB^s(t)$ is a superset of $FOP^s(t)$, *i.e.*, $FOP^s(t) \subseteq NB^s(t)$. $NB^s(t)$ is not exactly $FOP^s(t)$ because the first network may not be the bottleneck and hence the access time for the first network may not be the primary factor constraining the choice of an operating point. Set $nb^s(t)$ is a subset of $FOP^s(t)$, *i.e.*, $nb^s(t) \subseteq FOP^s(t) \subseteq NB^s(t)$. Set $nb^s(t)$ is not exactly $FOP^s(t)$ because the end-to-end packet latency may be considerably larger than the largest medium access time in the network. Together, NB^s and nb^s bound FOP^s. For example, Figure 3 illustrate the sets NB^s and nb^s for the audio and video streams for a point in a run of our conferencing system.

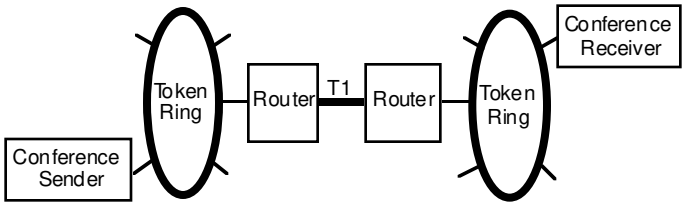


Figure 2

We refer to $NB^s(t)$ as the "network box": an area (not necessarily rectangular) in the bit-rate × packet transmission-rate space that contains the feasible operating points for stream *s* at time *t*. The position of the

"lower left hand corner" of the box is a function of human perceptual constraints and the current end-to-end latency and can be easily computed at run-time. The position of the "upper right hand corner" of the box is function of the congestion in the network and can also be easily computed.

Next we demonstrate that together, NB^s and nb^s are a useful bound on FOP^s and thus can be used for effective transmission control of media streams. When the network becomes congested, a conference application can easily compute the network box (using feedback from a receiver) and then apply simple but effective heuristics to find a feasible operating point within the box.

3. USING THE NETWORK BOX FOR TRANSMISSION CONTROL

The framework presented in the previous section can be used to exactly characterize the set of operating points that both are sustainable and that will lead to an acceptable quality conference. Although we cannot exactly compute this set at run-time, we can easily estimate it using $NB^s(t)$ and $nb^s(t)$ and use the framework of the bit-rate × packet transmission-rate space to adaptively find feasible operating points when congestion renders the current operating point infeasible. Below we present a heuristic for finding a feasible operating point given the current network box and information fed back from the receiver. To motivate the heuristic we begin with a simple taxonomy of the causes of congestion.

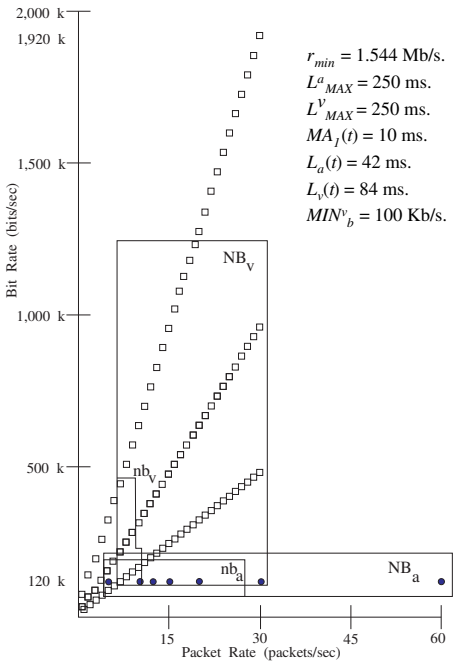


Figure 3: The audio and video network boxes.

The primary effect of congestion at a node in the network is the development of a queue of waiting packets. Expanding queues lead to long waiting times and eventually loss. It is possible that congestion may be eliminated by reducing either the bit-rate, b_s, or the packet transmission-rate, p_s, of one or more of the conference streams so that the congested node recovers. The type of congestion present in the network determines whether the bit-rate or the packet-rate has the most effect on the congestion.

A *capacity constrained network node* is a node whose performance is more sensitive to bit-rate than packet-rate. Typically, capacity constraints are caused by either limited network bandwidth on the outbound link of a source or forwarding node, or internal data movement time at forwarding nodes. Congestion at a capacity constrained node results in a decrease in the maximum supportable bit-rate. If the performance of the network is primarily determined by a capacity constrained node we say the network is capacity constrained. Decreasing b_s may relieve the node and result in reductions in latency and loss because of the reduced queuing delays at the node. Changes in p_s alone are unlikely to relieve a capacity constrained node since the number of bits handled by the node is unchanged.

An operating point (b_s, p_s) is *capacity constrained* at time t if $(b_s, p_s) \notin FOP^s(t)$ and there exists some other operating point, $(b'_s, p'_s) \in FOP^s(t)$ such that $b'_s < b_s$ and $p'_s \geq p_s$. In other words, point (b_s, p_s) is capacity constrained if it is not a feasible operating point and there is a feasible operating point with a packet-rate at least as great as p_s and a lower bit-rate. Area F in Figure 4 illustrates the capacity constrained operating points.

An *access constrained network node* is a node whose performance is more sensitive to the number of packets it must handle than to the number of bits. Access constraints are typically caused by (1) packet processing time at forwarding nodes (*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). Congestion at an access constrained node causes a decrease in the maximum supportable packet-rate. If the performance of the network is primarily determined by an access constrained node, we say the network is access constrained. Reductions in p_s may relieve the access constrained node by reducing its queue for medium access and hence reduce latency, loss, and access demands on shared-medium networks. Reductions in b_s alone are unlikely to relieve access constrained nodes since the number of packets handled by the node remains the same.

Operating point (b_s, p_s) is *access constrained* at time t if $(b_s, p_s) \notin FOP^s(t)$ and there exists some other operating point, $(b'_s, p'_s) \in FOP^s(t)$ such that $b'_s \geq b_s$ and $p'_s < p_s$. Point (b_s, p_s) is access constrained if it is not a feasible operating point and there is a feasible operating point with a bit-rate at least as great as b_s and a lower packet-rate. Area G in Figure 4 illustrates the access constrained operating points.

It is possible for the network to be both capacity and access constrained. In such environments, both b_s and p_s must be reduced in order to relieve the effects of network congestion. We say an operating point (b_s, p_s) is *capacity and access constrained* if $(b_s, p_s) \notin FOP^s(t)$ and there exists some other operating point, $(b'_s, p'_s) \in FOP^s(t)$ such that $b'_s < b_s$ and $p'_s < p_s$. Area E in Figure 4 illustrates these operating points.

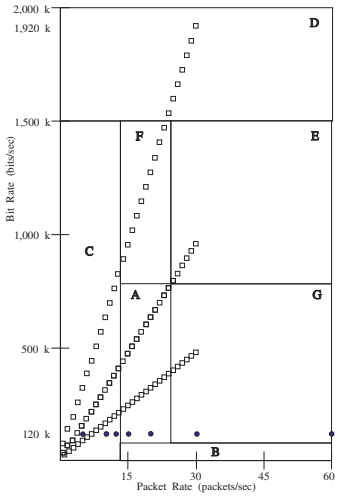


Figure 4: Capacity (region F) and access (region G) constraints.

3.1 Demonstrating Capacity and Access Constrained Networks

We demonstrate the difference between capacity and access constrained networks using the network, shown in Figure 5, and an experimental video conferencing system. The network topology is a common token ring configuration where "floor" rings in a building or a campus connect to a "backbone" ring spanning the campus. In the following experiments, node C is the source for a one-way conference with destination D. Machines C

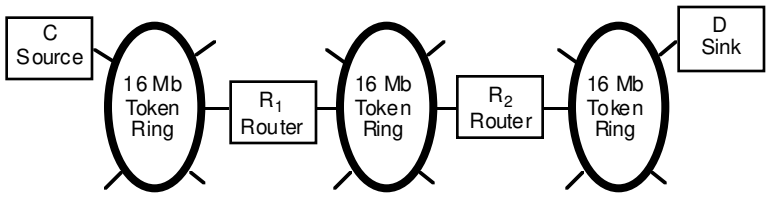


Figure 5: Network topology.

and *D* are IBM PS/2 486 66 Mhz personal computers using IBM/Intel Action-Media I audio/video adapters. The conferencing software was built at the UNC and is described elsewhere [4]. Unless otherwise stated, the conferencing system generates 60 audio frames and 30 video frames per second. Audio and video are transmitted separately as UDP messages with one frame per packet.

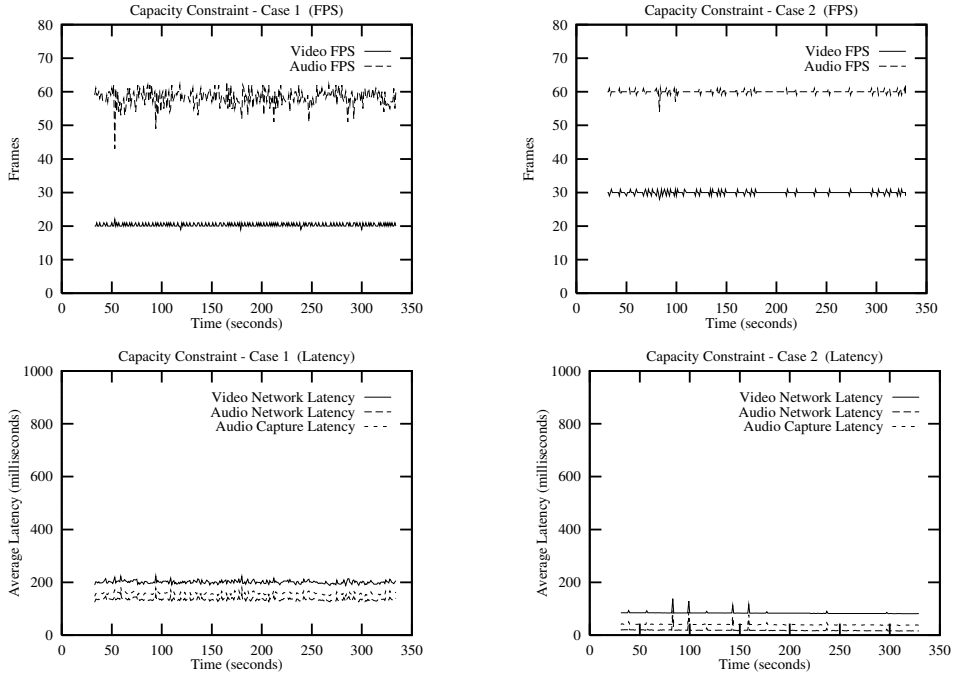
We demonstrate a capacity constrained network by modifying the routing software in router R_1 (Figure 5) to limit its throughput to approximately 1.5 Mbits/s. All token rings in this experiment are lightly loaded. The frames per second graph (FPS) in Figure 6a gives the observed conference frame rate for case 1 of the capacity constrained network. The *x*-axis is time in seconds as the conference proceeds. The *y*-axis is the number of frames received at the destination in 1 second intervals. In this case, *C* is transmitting 30 video frames (8000 bytes/frame) and 60 audio frames (250 bytes/frame) per second. The ideal delivery graph would show 60 frames of audio and 30 frames of video delivered every second.

The combined bit-rate of the audio and video streams is approximately 2 Mbits/s, which exceeds the capacity of R_1. Figure 6a shows that the delivered audio frame rate varies but generally all audio frames arrive. Video, on the other hand, is delivered at a rate of only about 20 frames per second, compared with a transmission-rate of 30 frames per second, which means the conference is wasting network resources.

The latency graph in Figure 6a shows the measured conference latency. The *x*-axis is as before; the *y*-axis is the average latency measured over 1 second intervals. Three latency values are reported: *video network latency* is the time between delivery of the video frame to the network adapter at *C* (Figure 5) to receipt by the video conference application at *D*. *Audio network latency* is similar. *Audio capture latency* is the audio network latency plus the time between capture of the audio frame and delivery of the frame to the network interface for transmission. Audio capture latency is a measure of the latency induced by choosing a packet-rate smaller than the audio frame rate. For case 1, the latency of both audio and video streams is acceptable.

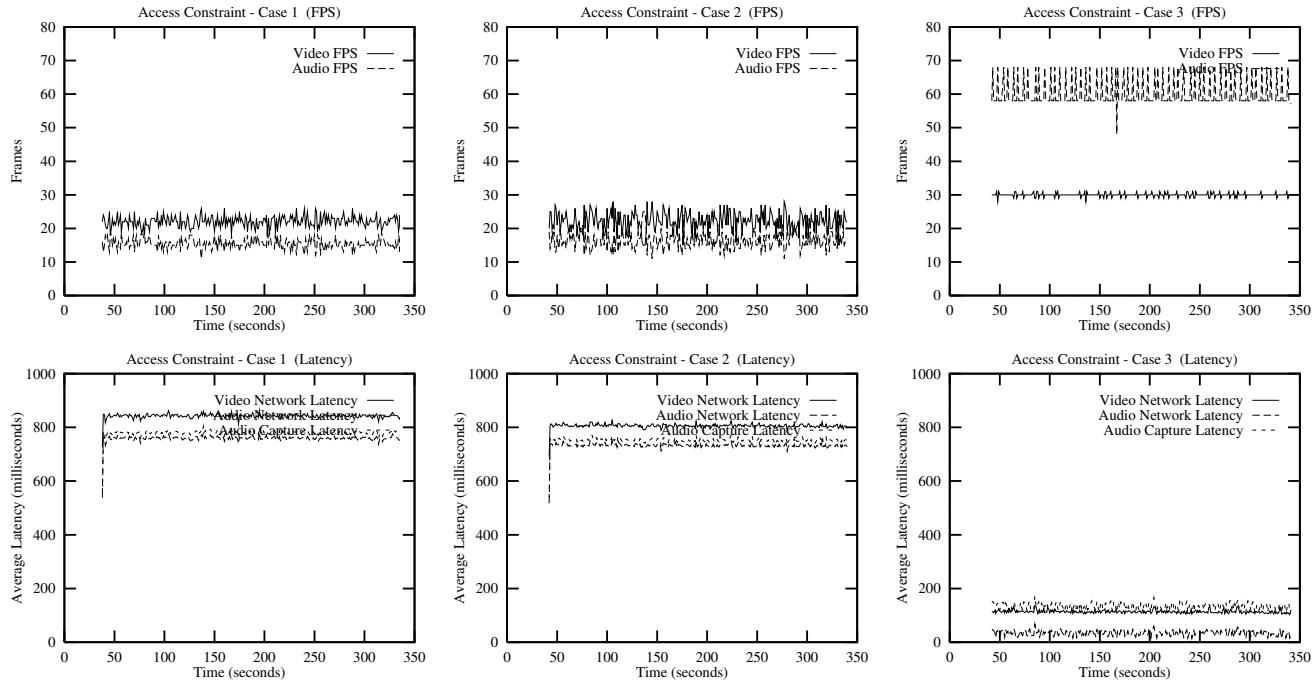
Figure 6b shows the results for case 2 of the capacity constrained network. In this case, *C* is transmitting 60 audio frames and 30 video frames per second, as in case 1, but video frames are now 4000 bytes. This corresponds to moving the video operating point from 30 packets per second on the top video rate line in Figure 1a to 30 packets per second on the middle video rate line. Even though the quality of each video frame is lower (the video bit-rate has been cut in half), the overall conference quality improves because the capacity constraint has been relieved. Audio jitter has been reduced and latency is lower than in case 1. Furthermore, all 30 frames of video are arriving at the destination. Note that although the transmitted bit-rate of the video has been reduced, the packet-rate of case 2 is identical to case 1.

To demonstrate an access constrained network we remove the artificial capacity constraint in R_1 and generate a heavy load on the backbone (the middle ring in Figure 5) using synthetic traffic generators. Figure 7 shows the conference performance for three pairs of audio/video operating points. In case 1, audio and video frames are packaged one frame per packet and transmitted as early as possible. Both conference fidelity, as measured in frames per second, and latency are extremely poor. The audio is unintelligible and the latency approaches a one-way delay of a second. In case



(*a*) Case 1: 8K video frames. (*b*) Case 2: 4K video frames.

Figure 6: Capacity constraint example.



(*a*) Case 1: 8K video frames. (*b*) Case 2: 4K video frames. (*c*) Case 3: 8K video frames, 10 audio frames/packet.

Figure 7: Access constraint example.

2, we again cut the video bit-rate in half as in case 2 of the previous example. Unfortunately, while this strategy was very successful for a capacity constrained network, it has little effect in an access constrained network. Conference fidelity and latency remain poor even with a significantly reduced bit-rate. In case 3, 10 audio frames are now packaged into a single network packet. Both conference fidelity and latency are radically improved. The reduction of audio packet-rate has relieved the access constrained router, R_1. Note that the transmitted bit-rate of the conference in case 3 is identical to case 1 and higher than case 2, but the change in packet-rate relieves the effects of congestion without any reduction in conference fidelity.

3.2 Capacity and Access Constraints as Heuristics for Finding $FOP^s(t)$

We conjecture that a transmission control algorithm that uses the concepts of access and capacity constraints can quickly find feasible operating points within the network box under a wide range of network conditions. Experiments described in the following section provide evidence for this assertion. Here we give an overview of one transmission control algorithm.

The algorithm uses feedback on packet loss and latency in a media stream as an indicator of the feasibility of the stream's current operating point. If feedback indicates that the current operating point is infeasible (*i.e.*, not currently sustainable) then a new operating point must be chosen. To do this we first compute the set nb^s. If $nb^s(t)$ is non-null then one can simply choose an element of $nb^s(t)$ (by definition an element of $FOP^s(t)$). Such an element may be a conservative operating point (*i.e.*, lead to relatively high latency and low throughput), however, one can iteratively move to more aggressive operating points over time. If $nb^s(t)$ is empty, then we compute $NB^s(t)$ and use the distinction between capacity and access constrained nodes as a heuristic for finding the points in $NB^s(t)$ that are also in $FOP^s(t)$.

The selection of an operating point in $NB^s(t)$ is based on a "recent success" heuristic. We record whether a bit-rate change or packet-rate change was the last adjustment which "improved" delivery of the streams. This is analogous to (grossly) characterizing the network as either capacity or access constrained. When the network is categorized as capacity constrained, the algorithm adjusts bit-rates. Specifically, if (b_v, p_v) is the current video operating point then the algorithm adjusts the video bit-rate if there exists a point $(b'_v, p'_v) \in NB^v(t)$ such that $b'_v < b_v$ and $p'_v \leq p_v$. The video bit-rate is adjusted first because it is typically much higher than that for the audio stream and because, for a video conference, the information contained in the video stream is typically less important than that in the audio stream. In extreme cases, no new operating point (b'_v, p'_v) exists and the algorithm will adjust the audio bit-rate (if there are points in $NB^a(t)$ with lower bit-rates than the current audio operating point).

When the network is categorized as access constrained, the algorithm adjusts the audio packet-rate. If (b_a, p_a) is the current audio operating point the rate is adjusted if there exists a point $(b'_a, p'_a) \in NB^a(t)$ such that $b'_a \leq b_a$ and $p'_a < p_a$. The audio packet-rate is adjusted first because audio frequently is available for transmission before its corresponding video (because of the relative time to compress video compared with audio) and because audio frames are typically much smaller than video frames, more frames can be carried in a single packet. When there are no feasible operating points with audio packet-rates lower than the current rate, the algorithm reduces video packet-rates using a similar criteria.

After initially classifying the network as either capacity or access constrained, the algorithm selects a new operating point and evaluates the result of the change through feedback messages from the receiver. If the change improves the transmission of the conference streams to the point where all transmitted data is received and the conference latency is acceptable, no further adjustments are made to the operating points. If the change in operating points improves the transmission of the streams, but the

delivered streams are still unacceptable because of loss or increasing latency, the algorithm selects another operating point based upon the same classification of network constraints. In other words, "recent success" using a particular network classification continues the use of the associated strategy. If the change in operating point does not improve the transmission of the streams, the algorithm changes the classification of the network constraint and selects the next operating point based upon the new classification. This "stairstep" decline (*i.e.*, moving vertically downward then horizontally to the left in the operating point space) continues until the algorithm finds operating points for audio and video which provide an adequate quality conference and are sustainable in the current network environment. In other words, the algorithm selects operating points until the streams are either "in the network box" or all candidate operating points in $NB^s(t)$ are eliminated. In the latter case it will not be possible to continue an acceptable quality conference ($\forall s$, $FOP^s(t) = \varnothing$).

After a stable period at a particular set of operating points, the algorithm attempts to move to operating points with increased bit or packet-rates (to either improve fidelity or reduce latency). To select the new operating point, the algorithm considers the current classification of the network and attempts to increase the rate along the other dimension. For example, if the network is classified as access constrained, the algorithm attempts to increase the bit-rate of video. The algorithm monitors feedback to assess the effect of the increase. If the stream improves, the network classification is retained and after another stable period, the algorithm will attempt to increase along the same axis. If the stream does not improve, the algorithm retraces the last operating point change (*i.e.*, it "undoes" the most recent rate change) and changes the classification of the network. If the operating point proves stable, the next attempted rate increase will be along the other axis.

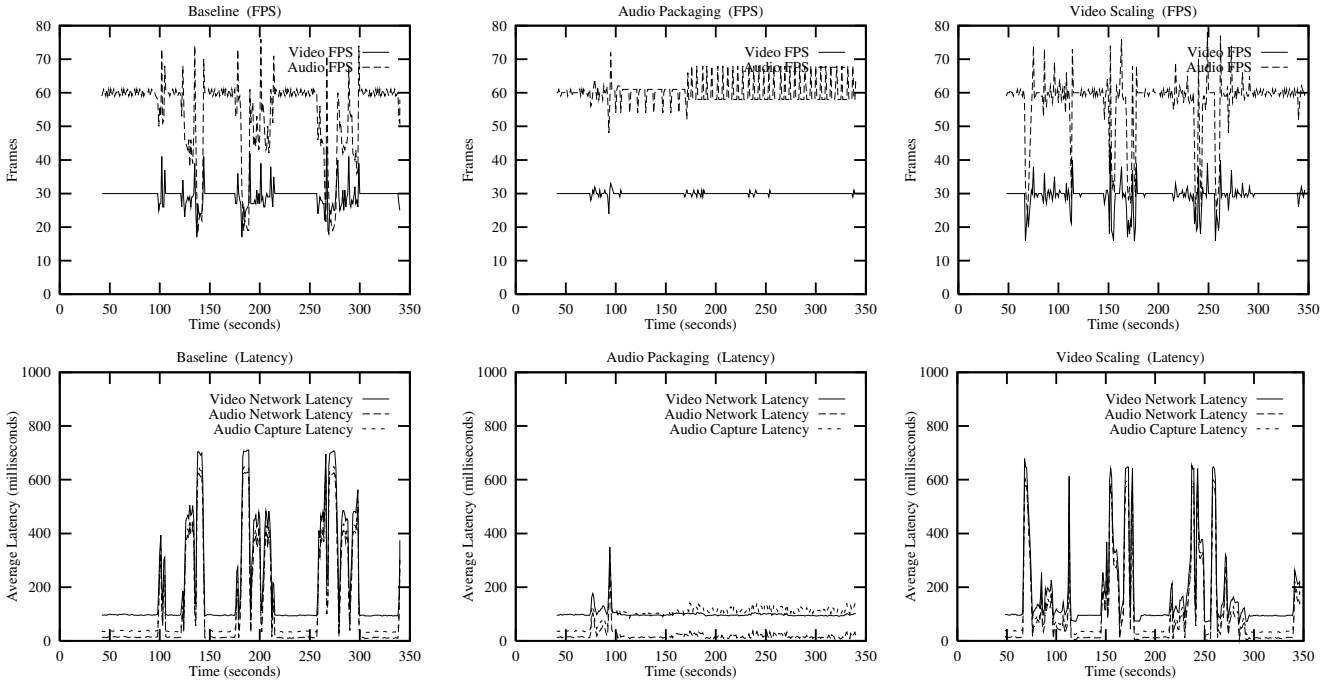
4. EXPERIMENTAL RESULTS

This section describes a set of experiments designed to illustrate the potential benefits of an adaptive transmission strategy based on manipulation of bit and packet-rates. In particular, these experiments demonstrate how the recent success algorithm can dynamically find and move with the network box.

The experiments were conducted on the network shown in Figure 5. In these experiments, *D* generates feedback messages once a second giving the number of audio and video frames received during the last feedback interval and the average network latency for packets (measured separately for audio and video packets). The floor rings to which *C* and *D* directly connect are lightly loaded; the backbone ring load varies between moderately to heavily loaded. The token wait time on the backbone, as measured by the IBM Trace and Performance token ring monitoring software, varies from 2 to 20 milliseconds depending upon the load. When the backbone token ring is heavily loaded, router R_1 becomes access constrained because of the long waiting time for free tokens on the backbone ring. Load is generated on the backbone by a set of synthetic traffic generators. The traffic generators represent load introduced onto the backbone ring from "floor" rings other than those used by *C* and *D* and are used to ensure repeatable experiments.

In a "baseline" experiment, audio and video is transmitted one frame per packet without any scaling. Video frames are approximately 8000 bytes and audio frames are approximately 250 bytes (with a common header of approximately 200 bytes per packet). Figure 8a shows that in this case, the number of media frames which actually arrive in a given second is highly variable. Moreover, there are long intervals with low frame arrival rates. This causes gaps in the playout of both streams and results in a poor quality conference. The latency graph in Figure 8a shows that the baseline transmission strategy also gives long periods of high audio and video latency. Studies have found that latency of this magnitude is unacceptable for video conferencing [2, 7].

The second experiment exercises the recent success transmission control algorithm. Our implementation of this algorithm determines when to adapt the operating point based upon the



(*a*) No transmission control. (*b*) Recent success heuristic. (*c*) Video scaling only.

Figure 8

number of frames delivered per second. Since audio is more critical than video, a set of simple audio frame rate and ramp thresholds govern the decision to adapt and control the aggressiveness of the adaptation. Figure 8b gives the performance of the algorithm. The delivered frame rates are well within the ranges controllable by jitter control schemes such as those described in [5]. The latency of the conference is also improved. Note that network latency is now only a small component of the "capture" latency of the audio stream. Most of the latency in this stream is now due to the induced latency caused by buffering multiple audio frames at the sender in order to send large numbers of audio frames in a single network packet. For example, 200 to 350 seconds into the conference, 8 to 10 audio frames are being packaged in each network packet.

To illustrate the adaptive nature of the algorithm, consider the operating point choices made approximately 90-110 seconds into the conference. At 90 seconds into the conference the video stream operating point is (1920K, 30), which is the operating point generating 30 packets per second using the highest quality encoding. The audio stream operating point is (120K, 30), which means 2 audio frames were being packaged into each packet. At 93 seconds, the conference source gets feedback from the destination that shows packet loss in each stream. This implies that the current operating points are no longer feasible. The last successful choice of operating point had been to move from (120K, 60) to (120K, 30) for the audio stream. Therefore, the source now considers the network access constrained and selects a new operating point for audio of (120K, 10). This change improves the loss and latency in both streams. At 103 seconds, the audio operating point is moved further along the path of recent successful changes to (120K, 6). At 109 seconds, the source determines that the operating points have been stable and attempts to increase the bit-rate of the streams; however, since each stream is operating at its maximum bit-rate, there are no operating points with higher bit-rates. Since the source cannot improve service along the bit-rate axis, it seeks to improve service along the packet-rate axis and selects a new operating point for audio at (120K, 10).

For comparison, Figure 8c shows the performance of scaling only in the bit-rate dimension. The video conference packages frames as in the baseline case but attempts to scale the size of the video frames (*i.e.*, jumps between video rate lines in Figure 1a) when feedback indicates the network is having trouble with the offered load. The video scaling algorithm decides to adapt the video encoding scheme using the same metric to evaluate the state of the conference as the recent success algorithm. Figure 8c shows that scaling bit-rate alone did not significantly improve the conference.

5. SUMMARY AND CONCLUSIONS

The desire to support real-time communication on LANs that do not themselves support real-time communication will likely persist for the foreseeable future. One of the major obstacles facing designers of video conferencing and other distributed multimedia systems is the problem of ameliorating the effects of congestion to achieve real-time communication. We have presented a framework for scaling media streams along bit and packet transmission-rate axis to reduce the impact of congestion. A novel aspect of the framework is its recognition that adapting the packet transmission-rate can significantly effect the performance of a video conference. Preliminary use of the framework by a simple transmission control algorithm on token-ring networks demonstrates the effectiveness of the two-dimensional scaling concept and shows the benefit that adapting packet transmission-rate can provide over simply scaling bit-rates.

Much work remains. First, to date we have only been able to apply our framework to token ring networks. These networks work well for our system because they offer a large MTU and hence a large number of operating points are possible for our system since we can place multiple frames into a packet. While such would not be the case for other networks, most notably ethernets, we believe our work would apply to conferencing systems on these networks that generate lower bit-rates than ours (*e.g.*, more conventional systems that use PCM audio and 364 Kb/s video). Moreover, it is not clear how our work would apply in campus network environments that use ATM technology (*e.g.*, as the backbone). Although ATM specifies a tiny packet (cell) size, it is likely that the local-area network service abstractions (adaptations) exported by an ATM LAN will offer a range of larger (virtual) packet sizes (with the potential of real-time delivery) and hence we believe our work will be applicable.

A second issue concerns whether or not a discriminator exists that identifies the network as primarily capacity or access constrained. Given the amount of information available about the network at the endpoints, it seems unlikely that such a discriminator exists. We suspect only heuristic adaptation algorithms are viable and are investigating several additional algorithms. Most of these algorithms are end-to-end, but some preliminary work has been done on modifying router nodes to support "best-effort" transmission (*i.e.*, without resource reservation) using hop-by-hop strategies based upon the transmission framework.

Lastly, for any environment in which our framework is useful and for whatever discriminators we develop, we need to characterize how well the network box concept performs as a function of the size of the network and the number of interconnection points.

6. REFERENCES

[1] 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, August 1993, pp. 99-104.

[2] 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.

[3] Jeffay, K., Stone, D.L., and 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.

[4] Jeffay, K., Stone, D.L., and Smith, F.D., *Kernel Support for Live Digital Audio and Video*, Computer Communications, Vol. 16, No. 6 (July 1992), pp. 388-395.

[5] Stone, D.L., Jeffay, K., *An Empirical Study of Delay Jitter Management Policies*, ACM Multimedia Systems, *to appear*.

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

[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.

[8] 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.