The Performance of Two-Dimensional Media Scaling for Internet Videoconferencing*

Peter Nee Kevin Jeffay
University of North Carolina at Chapel Hill
Department of Computer Science
Chapel Hill, NC 27599-3175
{nee,jeffay}@cs.unc.edu

Gunner Danneels
Intel Corporation
Intel Architecture Labs
Hillsboro, OR
Gunner_Danneels@ccm.jf.intel.com

http://www.cs.unc.edu/Research/dirt

Abstract: Until mechanisms for true real-time network communications are deployed and pervasive, one must rely on adaptive, best-effort congestion control methods to provide acceptable levels of service for interactive, real-time multimedia applications. Here we report on our experiences with a novel media-scaling congestion control scheme that was implemented in an experimental version of the Intel ProShare™ videoconferencing system and tested over the Internet. The media scaling scheme is unique in that it employs two-dimensional media scaling — the bit-rate and packet-rate of media streams are independently scaled. The goals of our study were (1) to empirically assess the performance improvement of two-dimensional media scaling over the simpler, and more commonly employed, one-dimensional scaling approaches and (2) to determine if it was possible to sustain ProShare conferences for a significant enough fraction of the time that two-dimensional scaling could be considered effective. We observed that systems using one-dimensional and two-dimensional scaling were both able to sustain conferences and that the two-dimensional scaling system always produced conferences with greater effective throughput. Our study provides empirical evidence that two-dimensional media scaling can be used effectively to ameliorate the effects of congestion in the Internet and can significantly extend the usability of an interactive multimedia application on the Internet.

1. Introduction

This paper presents the results of a set of experiments using an adaptive, best-effort videoconferencing system on the Internet. The system, a modified version of the Intel ProShare™ videoconferencing system, is novel in that it employs a two-dimensional media scaling algorithm to ameliorate the effects of congestion. Both the bit-rate of media streams and the partitioning of the streams into network packets (the packet-rate) are independently scaled [9]. Our goal in this work was not to attempt to construct a videoconferencing system that would provide acceptable quality conferences under all network conditions, but rather to build a system that could sustain conferences over a sizable Internet path under daytime traffic conditions and to use the system as a testbed for the evaluation and comparison of adaptive scaling schemes. In particular, we were interested in the comparison between two-dimensional scaling schemes and conventional one-dimensional scaling in the Internet environment.

Our experience with this system and the experiments we performed highlight several of the issues surrounding real-time, low latency communications on current internetworks, and especially on the Internet itself. These issues include the practicality of videoconferencing on today's Internet, the applicability of adaptive scaling over a lengthy Internet path, the utility of two-dimensional scaling on the Internet, and the proper criteria for assessing the congestion control mechanisms of adaptive multimedia applications.

Our experiments show at least some promise for high-quality, low latency videoconferencing on the Internet. Although peak-period traffic makes conferences with our system infeasible, there are certain times during an average weekday at which we were able to routinely sustain quality conferences. Our experiments also indicate that adaptive techniques are able to function over Internet paths of some complexity. In addition we are able to measure a qualitative benefit of two-dimensional scaling over one-dimensional scaling.

The following section reviews the congestion control problem for interactive multimedia applications and describes some approaches reported in the literature for dealing with congestion. Section 3 presents the videoconferencing system used in this work and describes our two-dimensional media scaling scheme. Sections 4 and 5, respectively, describe our experimental method and the re-

* Supported by grants from the IBM and Intel Corporations, the National Science Foundation (grants IRIS-9508514 & CCR-9510156), and the Advanced Research Projects Agency (grant number 96-06580).

In: Proceedings of the Seventh International Workshop on Network and Operating System Support for Digital Audio and Video, St. Louis, MO, May 1997, *to appear*.

sults of using our system over the Internet over the course of three months. These experiments raise a number of issues such as the interplay between adaptive media scaling and TCP's congestion control. We discuss these and other issues in Section 6 and make the case for a definition of "network-friendly" application behavior that is consistent with the congestion control requirements of real-time multimedia applications. In addition, based on experiences with our system, we offer some insight into desirable attributes of scaleable audio and video codecs.

2. Background and Related Work

Current internetworks are built primarily from best-effort components. Paths between end systems are composed of elements such as shared media LANs, wide-area telecommunication links, and routers and switches that simply propagate packets as quickly as possible with no attempt to allocate resources or manage congestion. These best-effort components are cost effective and widely deployed. Therefore, even as more sophisticated components become available, best-effort networking will remain dominant for some time. The large investment in best-effort components and the availability of (presumably lower cost) best-effort service classes in proposed integrated network services for the Internet provide powerful economic incentives for the continued use of best effort services. Thus, the continued study of applications in a best-effort environment is important.

Videoconferencing is an application that is especially sensitive to network congestion. The goal of videoconferencing is to present elements of multimedia streams in a regular, periodic fashion with minimal latency to human participants. Network congestion can cause packet loss, which reduces the quality and regularity of the streams by introducing gaps in the playout of the streams. Congestion can also cause queuing at network elements such as routers, which introduces delays that impede interactivity. Finally, congestion can cause high variation in end-to-end delay (*delay-jitter*) which interferes with the smooth playout of a stream, even to the point where some portions of the stream, though successfully transmitted, arrive too late to be played and must be discarded.

Two primary methods of dealing with congestion for real-time multimedia applications have been proposed: *resource reservation*, and *adaptive media scaling*. Resource reservation schemes allocate network resources to a specific application or class of applications [2, 6, 11, 12]. Adaptive schemes monitor end-to-end performance and attempt to deal with congestion through adjustments to the attributes of media streams. The goal is to find a suitable set of attributes that results in media streams that are deliverable in real-time in the current network conditions [1, 4, 5, 8, 9]. The primary advantage of reservation schemes is that with a suitable admission control policy, they can provide an *a priori* assessment of feasibility, and guarantee success if the conference is feasible. However, reservation schemes have two disadvantages. One is that the reservation of network resources must be done at every point in the path between the end systems. Otherwise, the feasibility of the transmission can be compromised at nodes supporting only best-effort service. This is a serious limitation as a broad deployment of reservation-capable network components remains some years away. Secondly, the static model of a reservation for a multimedia stream is overly restrictive. Multimedia applications are inherently scaleable, capable of conveying useful information over a wide range of quality/resource consumption tradeoffs. Making a reservation requires an arbitrary choice of a set of application operation parameters. Once this choice is committed by a reservation, it cannot easily be adjusted over time.

Adaptive methods are capable of operation over networks that are partially or completely made up of best-effort components. They exploit at least some of the scaleable attributes of multimedia streams, to make best use of the network under rapidly evolving conditions.

However, adaptive methods may exhibit decreasing effectiveness as the complexity of the network increases. Adaptive methods depend on feedback from the receiver to sample network conditions and attempt to operate at a stable point near the optimal for the current conditions. In a complex best-effort network, bottlenecks may appear and disappear at arbitrary locations, competing connections may come and go, and the availability of resources may fluctuate rapidly. If conditions evolve too rapidly for accurate sampling, the sender's adaptations will be ineffective. Indeed, there may exist significant periods of time during which there is no stable, sustainable mode of operation of the system. Experiments with our system indicate that such congestion is frequently experienced on a moderate length Internet path.

We characterize a videoconferencing system as a set of *operating points* in a bit-rate $\times$ packet-rate plane. Each point in this plane represents both a bit-rate that the system is capable of generating and a partitioning of the resulting media stream into network packets. The operating points for a videoconferencing system succinctly describe the cross-product of all the possible ways of operating the acquisition and compression hardware and all the possible ways of packaging data into packets and passing them to the network interface. The media scaling problem is that of choosing an operating point that is sustainable given the current network conditions.

To date, most adaptation schemes have been *one-dimensional*, that is, in times of congestion they primarily adjust the bit-rate of media streams. *Two-dimensional* scaling is a more sophisticated scheme that independently adjusts the bit-rates and the packet-rates of media streams [9]. Bit-rate scaling primarily ameliorates *capacity constraints* — a shortage of resources consumed by the transmission of bits. As an example, consider a router that is unable to transmit a packet from a stream in real-time. If the residence time for the packet at the router is dominated by the cost of physically transmitting the packet on an outbound link, or by the CPU time required to move data between buffers on router interfaces, then the stream is capacity constrained at that router. The benefit of two-dimensional scaling is that in addition to capacity constraints, it explicitly addresses constraints on the *access* to network resources. If the residence time for the packet at the router is dominated by the cost of per packet processing overhead (*e.g.*, route selection), or the media access time experienced when transmitting a packet on a shared media LAN, then the stream is access constrained at that router. Capacity constraints are alleviated only through a reduction in the bit-rate of a stream. Access constraints are alleviated only through a reduction in the packet-rate of a stream. In the latter case, this can be done by changing the packaging of a stream by simple techniques such as inter- or intra-stream aggregation of media units. In particular, one can often change the packaging of the stream with no reduction in throughput (bit-rate).

Previous work on two-dimensional scaling has shown that both types of constraints can arise on best-effort networks, and that independently adapting to both capacity and access constraints can greatly improve the quality of the delivered multimedia stream. Under certain network conditions two-dimensional scaling can even render a previously infeasible execution of a videoconferencing system feasible [9, 10].

Some adaptive schemes, such as temporal video scaling (*i.e.*, changing the video frame rate) adjust both packet-rate and bit-rate. However, these schemes are still one-dimensional in the sense that they do not independently adapt to both types of constraints, instead they adapt to the most severe and limiting of the two.

3. System Description

To experiment with the suitability of adaptive scaling schemes for an Internet application, we modified Intel's ProShare™ videoconferencing system (version 1.8) to support one- and two-dimensional media scaling. The ProShare™ system has many attributes well suited to our purposes. It generates low bit-rate, two-way audio and video streams. Although it was designed to operate at three preset, fixed data-rates (one data-rate for ISDN or LAN operation and two others for LAN operation only), we took advantage of an internal interface to the video codec to implement on-the-fly adjustments to the video bit-rate and frame-rate. The video frame rate was adjustable over the full range of frame-rates (from one to thirty frames per second). In addition, some flexibility in bits per frame was available. We exploited this bits per frame scalability to set three levels of compression, corresponding to 1,200 bytes per frame ("small size"), 2,100 bytes per frame ("medium size") and 2,800 bytes per frame ("large size"). In each case the compression scheme changes the number of bits per picture element in the image. These frame sizes correspond roughly to one Ethernet packet per video frame, one and one half Ethernet packets per video frame, and two Ethernet packets per video frame.

ProShare uses an interframe encoding scheme based on motion estimation. As a result of this, not all video frame rates are generated for each frame size. At frame rates below ten frames per second, the frames are generated so infrequently that the motion between frames is too great to sustain the small frame size. Conversely, at frame rates of twenty or above, frames are so frequent that the motion estimation typically works extremely well and the codec never generates more than 2,100 bytes per frame. Therefore the large frame size is not available above 19 frames per second.

ProShare generates audio at a single, aggressively compressed rate of ten 200 byte audio-frames per second. We are unable to adapt the bit-rate of the audio stream, and the low audio frame rate makes aggregation of two consecutive audio frames untenable; the introduced latency of 200 ms is just too great.

3.1 One-dimensional scaling implementation

There are numerous one-dimensional scaling schemes (see [5] for an excellent survey). Our one-dimensional scaling system was based on temporal video scaling (*i.e.*, scaling the video frame rate) using the medium level of compression. The frame rate was adjusted over a range from 6 frames per second to 30 frames per second. Because a medium video frame exceeds the size of an Ethernet MTU, each additional video frame has a cost of two packets. To minimize latency, no aggregation is performed. The fixed rate audio provides an additional ten packets and 16,000 bits per second. Since audio operates at a fixed point, we use the sum of the video operating point and the fixed audio overhead as the operating points of this system. A

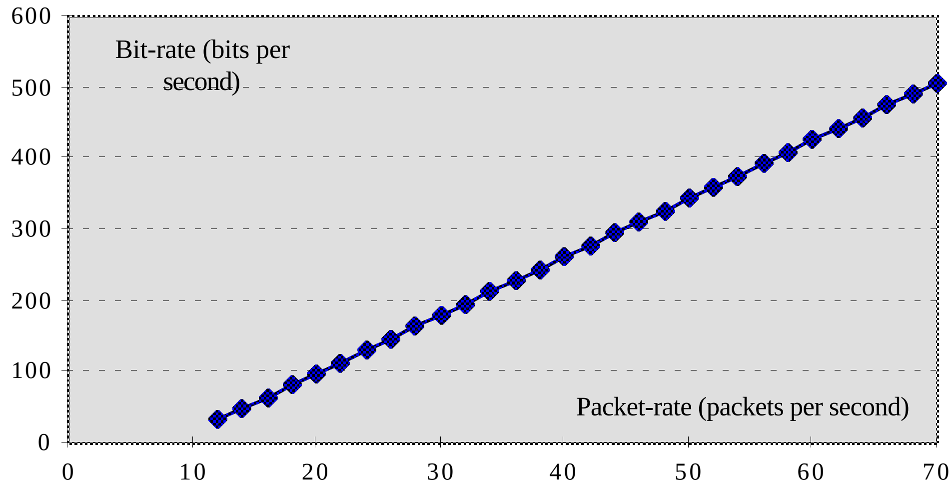

Figure 1: ProShare operating points (1-Dimensional scaling system).

plot of these combined operating points is provided in Figure 1. Note that while the operating points are plotted in the bit-rate × packet-rate plane, they all lie in a straight line in the plane. This is consistent with the thinking behind one-dimensional scaling schemes. The operating point set is geared toward scaling in the bit-rate dimension via frame rate adjustment, with the packet-rate at a fixed proportion to the bit-rate. No aggregation is attempted because that would only reduce the packet-rate, at the cost of increased latency. In a system that views bit-rate as the dominant constraint, there is no incentive to reduce the packet-rate at the cost of increasing latency.

We chose temporal video scaling for our one-dimensional system for several reasons. From a pragmatic standpoint, it is easy to implement and many of the other one-dimensional scaling schemes such as color-space scaling (reducing the color depth of images) require a more intimate knowledge of the internal workings of the codec than we possess. Second, we claim that all bit-rate scaling schemes yield sets of operating points that are similar to that shown in Figure 1. The slope of the line may change or the line may be translated vertically or horizontally in the plane, however, the result is essentially a line. As we will see shortly, our one-dimensional operating point set spans much of the same region in the plane as our two-dimensional point set. Therefore, the one-dimensional scaling scheme based on the point set in Figure 1 provides the best “competition” for the two-dimensional scheme.

All adaptive schemes use feedback from the receiver to detect congestion and adjust the stream appropriately. The adaptation process in one-dimensional schemes is straightforward as there is only one variable that is manipulated (*i.e.*, bit-rate). When an adaptation is required, our one-dimensional system always attempts to adjust the bit-rate of the video stream by a constant fraction of the current frame rate. It does this by increasing or decreasing the frame rate by:

- one frame per second when the previous frame rate was in the range of one to nine frames per second,
- two frames per second when the previous frame rate was in the range of ten to 19 frames per second, and
- three frames per second when the previous frame rate was in the range of 20 to 29 frames per second.

If the conference is operating at 30 frames per second, a decrement of four frames per second is used when congestion is present. In addition to these rules, frame rates below six frames per second are not used due to excessive video latency and lack of motion in the playout.

Feedback is given by the receiver at one second intervals. Feedback messages include measurements of network packet loss and estimated round trip latency. Loss of more than two packets, or a latency increase of fifty percent over a moving average of the previous five measurements is taken to indicate congestion. Each time feedback indicates congestion the video frame rate is reduced. Thus, repeated reports of congestion lead to a rapid reduction in video frame rate. If four successive feedback messages indicate no sign of congestion, this is taken as an indication of congestion relief, and the video frame rate is increased as described above.

3.2 Two-dimensional scaling implementation

A two-dimensional scaling system values both packet-rate and bit-rate adjustments. For this type of system, an oper-

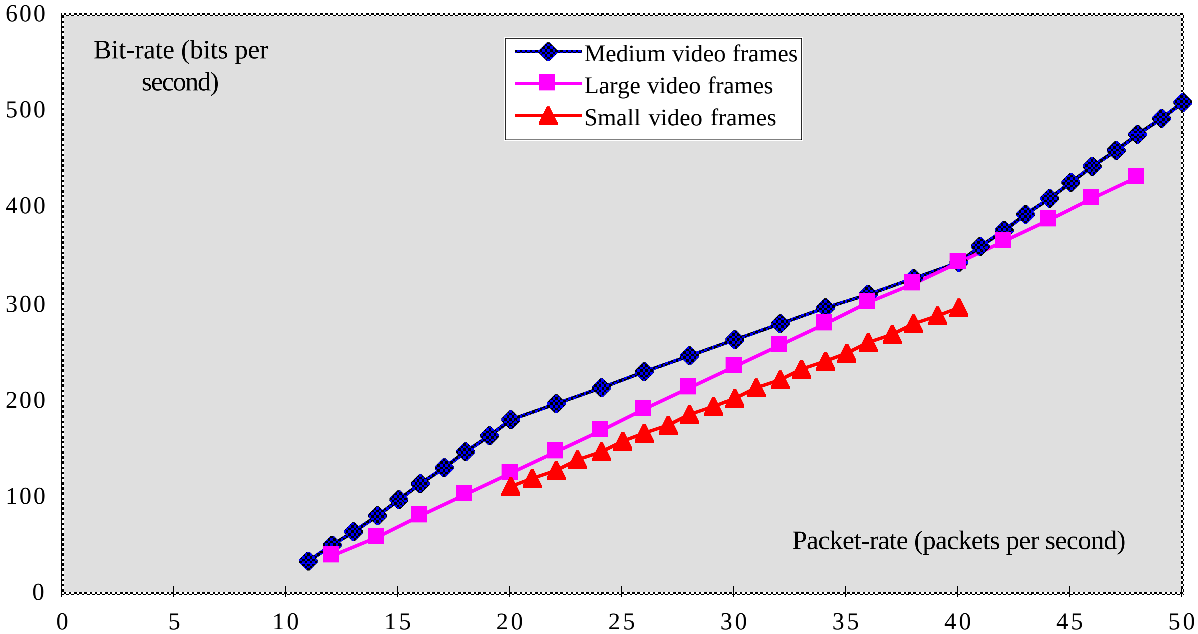

Figure 2: ProShare operating points (2-Dimensional scaling system).

ating point set that covers a large area in the bit-rate × packet-rate plane is desirable. However, with the video frame size as large or larger than the network MTU, and no possibility of audio adaptation, covering a sufficient area of the operating point plane is difficult.

Our two-dimensional system uses two strategies to increase coverage of the bit-rate × packet-rate plane. First, all three levels of video compression are used. Second, we use aggregation of audio frames with medium size video frames to "spread out" the operating points in the plane and thus provide more flexibility for two-dimensional adaptation of the video. To see this, it is useful to view each medium size frame as a one packet "body" and a half packet "tail." At frame rates of ten or below, each tail can be aggregated with an audio frame. Thus, each additional medium size video frame in this range raises the overall packet-rate by one, instead of by two as would be required without aggregation. (In theory this technique could also be applied to small size frames, however, in practice when application headers are factored in, the combination of a small video frame and an audio sample is often larger than a single Ethernet packet.)

In the range of 11 to 20 frames per second, there are too many video tails to aggregate with the audio frames, so each additional frame beyond 10 frames per second requires two additional packets. Above 20 frames per second, the video frames are generated close enough together in time that we can aggregate the tail of each additional frame with the currently unaggregated tail of an adjacent frame with only a minimal impact on latency. Thus, in this range, each additional increase in video frame rate requires only one additional packet per second.

As always, there is some additional latency introduced by aggregation, but this rather complex arrangement ensures that the additional latency never exceeds 50 ms. In fact, the arrangement is implemented with a 50 ms timer for holding small packets that are candidates for aggregation.

The resulting operating points are plotted in Figure 2. The slope changes in the upper line (the medium frame rate) are the result of the changing number of packets per additional frame as we move from 10 and fewer frames per second (the region wherein we require one additional packet per additional frame) to the 10 to 20 frames per second range (two additional packets per additional frame) to the above 20 frames per second range (one additional packet per additional frame).

Note that the size of the range of bit-rates available varies at each packet rate. The bit-rate ranges are the largest in the twenty to forty packet range. This is the range in which this particular system is the most "two-dimensional," and, as we shall see, is the range of operation where two-dimensional scaling is the most beneficial for this system.

Our two dimensional system adapts the outgoing streams via two types of adjustments. We perform pure bit-rate scaling by moving between small, medium and large oper-

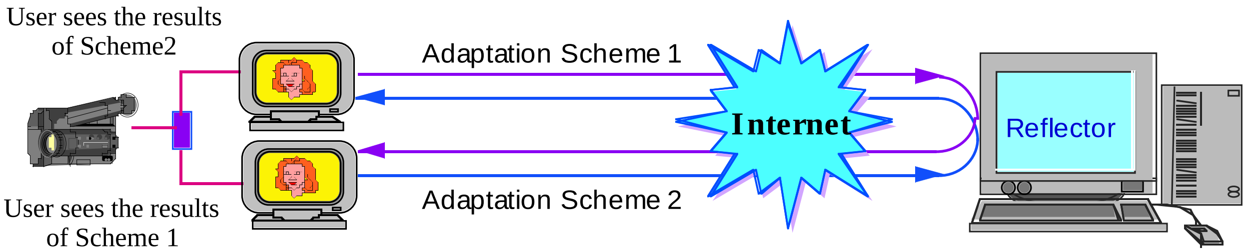

Figure 3: Experimental setup.

ating points while keeping the packet-rate constant. We also perform an adjustment in both bit- and packet-rate (but primarily in packet-rate) by temporal video scaling. We make the frame rate adjustments proportional to the current level of operation (*i.e.*, to change the bit-rate by a constant amount on each adaptation) as in the one-dimensional scaling scheme.

The mechanics of the adaptation process are more complicated in two-dimensional scaling than in one-dimensional scaling. With two-dimensional scaling there are two independent variables. However, both access and capacity constraints exhibit the same symptoms to end systems: increases in packet loss, packet latency, or packet delay-jitter. We use a "recent success" heuristic, which makes adjustments in one dimension in the bit-rate × packet-rate plane for as long as they seem effective. If adjustments in that dimension do not appear to alleviate congestion, recent success switches to adjustments in the other dimension. The contents of the feedback, as well as the criteria used to detect congestion and the relief of congestion, are identical to those used in the one-dimensional system described above.

4. Experimental Method

4.1 Setup

The goal of our experiments was to evaluate the performance of the modified ProShare system under real Internet traffic conditions. As each conference was competing with live Internet traffic, each experiment was non-repeatable and the results of different experiments were more or less incomparable. Therefore, to compare the relative performance of one- and two-dimensional scaling under comparable network congestion conditions, both scaling algorithms were run simultaneously using the configuration shown in Figure 3.

Each experiment consisted of a point-to-point, bi-directional conference between two machines located on a common Ethernet. One direction of the conference used a one-dimensional media scaling scheme to ameliorate the effects of congestion. A two-dimensional scheme was used in the other direction. Each conference end-point provided the appropriate feedback to drive the adaptation process at the other end as described in Section 3. A single camera and audio source provided the same input to each system. Thus an observer could view the displays simultaneously and qualitatively compare the effects of each scaling scheme on the same inputs.

Although the two ProShare systems are attached to the same LAN, they exchange data streams via a remote reflector located at the University of Virginia (10 hops away). Therefore, each media stream traverses the path from UNC to UVA twice, traveling across 20 network hops.

Experiments consisted of 10 minute conferences run on weekdays during predetermined time slots over a three month period. Multiple experiments on a day were permitted, but experiments were always separated in time by at least two hours.

In addition to running experiments comparing the performance of one- and two-dimensional scaling, we also ran a small number of experiments comparing the two-dimensional adaptive system with the original version of the ProShare system that used fixed points of operation.

4.2 Evaluation Methodology

There were two aspects to the evaluation of each experiment. First, a human observer recorded a subjective impression of the utility of the conference based on direct observation of each system. The subjective evaluation was primarily based on an assessment of the quality of the audio playout and was used to judge the overall success of the conference. If the conference was judged too poor in quality for useful human-to-human communication it was judged a "failure." Second, detailed numerical measurements of conference quality were recorded by each system in an in-memory log to assess the relative performance of one-dimensional versus two-dimensional scaling. At the end of each experiment these logs were saved to disk for later analysis. The qualitative and quantitative analyses were combined as follows.

When an adaptive system operates over a wide range of conditions, we can evaluate it on three criteria:

- *Necessity of adaptation.* What network conditions constrain the system enough to require adaptation?
- *Feasibility of adaptation.* When adaptations are required, what network conditions permit the system to operate successfully with adaptation and what conditions preclude any adaptation from being successful?
- *Effectiveness of adaptation.* When adaptations are performed, what is the quality of the resulting conference?

For a given adaptive videoconferencing system, the first two criteria are determined by the extremal operating points. If there exist network conditions wherein the system's highest quality operating point (the operating point furthest from the origin — the operating point that generates the highest stream bit-rate and has the lowest latency due to packaging at the sender), can be routinely sustained, then adaptation is not necessary. Conversely, if there exist network conditions in which even the lowest quality operating point (the operating point closest to the origin — the operating point that generates the lowest stream bit-rate and has the greatest latency due to packaging at the sender), cannot be sustained, then operation of the system is not feasible.

Comparisons of adaptive videoconferencing systems based on these two criteria will hinge on the differences in the extremes of their operating point sets. For example, a system with operating points for highly compressed video will be feasible for a larger set of network conditions. A system with a maximum video frame rate of thirty frames per second will require adaptation of frame rate under more conditions than a system with a maximum video frame rate of fifteen.

For a given network, we can evaluate the utility of an adaptive videoconferencing system by the frequency with which conditions arise wherein adaptation is both necessary and feasible. Under these conditions, we can further compare videoconferencing systems by the effectiveness criterion. In this case (and only in this case), it is the richness of the interior space of the operating point set and the actual adaptation algorithm that are important. A rich operating point set gives a videoconferencing system the flexibility to closely adapt to network conditions. A good algorithm finds the current appropriate operating point quickly, and without excessive adjustment. For a given network, we can evaluate the effectiveness of an adaptive videoconferencing system by performance measurements on the delivered media streams, such as latency, audio loss, and video frame rate. These comparisons are complicated by the fact that individual systems often make implicit tradeoffs between these measures such as increasing the quality of the displayed images by increasing the time required for processing samples (and hence the increasing the playout latency). However, if care is taken, meaningful comparisons can be made.

5. Experimental Results

We now turn to the evaluation of our two adaptive ProShare systems operating on a 20 hop Internet path under weekday conditions. First, with respect to necessity, not surprisingly, adaptation was required in all runs. Second, with respect to feasibility, we can use the observer's subjective assessment to differentiate between feasible and infeasible conferences. No significant difference in success-rate was noted for the one-dimensional and two-dimensional systems. Table 1 summarizes the utility of adaptive scaling by time of day. The results are consistent with naive expectations. The traffic starts out in mid-morning low enough to permit adaptive videoconferencing, builds to a peak during the late afternoon during which time our system was unable to sustain a quality videoconference, and then gradually recedes. Our results show a peak-period consistently centered in the afternoon. Since much of experimental path passed through academic environments, this peak may be skewed later in the day than for other parts of the Internet.

Finally, for conditions wherein scaling is necessary and feasible, our comparison of one-dimensional and two-dimensional scaling showed subtle but noticeable benefits for two-dimensional scaling by the effectiveness criterion. Primarily, two-dimensional scaling delivered better quality conferences by allowing the transmission of more video frames, especially in moderate traffic conditions. In heavy traffic conditions, two-dimensional scaling was not significantly more

Table 1: Summary of qualitative results

Time Slot	Number of Successes	Number of Failures	Success Percentage
10:00-12:00	6	3	67%
12:00-14:00	4	4	50%
14:00-16:00	1	11	8%
16:00-18:00	3	9	25%
18:00-20:00	4	5	44%
Percentage	36%	64%	

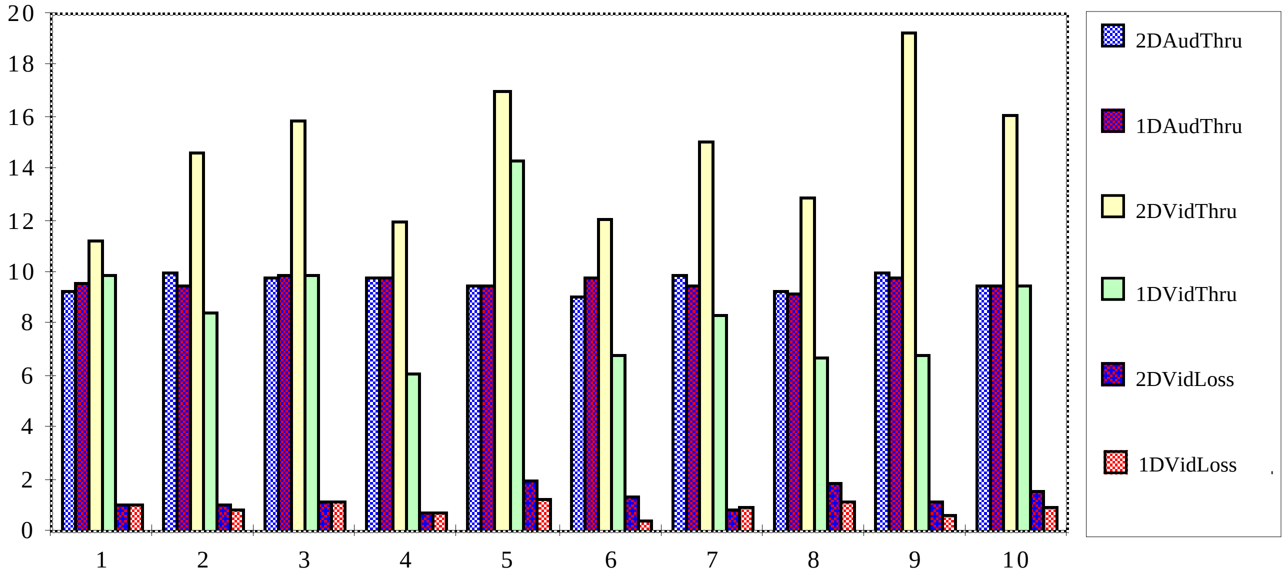

Figure 4: Video and audio throughput and video loss averaged over one minute intervals for a 10 minute conference.

effective than one-dimensional scaling. Since heavy traffic requires operation at the low end operating points (those closest to the origin), this is consistent with the lack of a significant difference in feasibility between the two systems.

Examination of the operating point plot provides insight here. First, any two-dimensional video scaling system provides less flexibility at the low end, since all the quality lines converge at the origin. Secondly, for our system, due to the constraints built-in, this was even more the case. For example, the small video frame size was only available at video frame rates above ten frames per second. Thus at low frame rates, less packet-rate scaling is possible. Also, at moderate frame rates, there were enough video tails available to take full advantage of packaging with audio. Thus, where our system is most able to take advantage of two-dimensional scaling, the middle range of operating points, is where we see a benefit typical of two-dimensional scaling: the ability to push more bits (higher quality) through by more attention to frame size and packaging. Therefore, our comparison to one-dimensional scaling shows that two-dimensional scaling is beneficial, and that the more the codec allows operating points spread out over the bit-rate × packet-rate plane, the greater the benefit.

Figure 4 shows a minute by minute plot of received audio frame rate, received video frame rate, and video frame loss (video generated and sent but not received) for both two-dimensional and one-dimensional scaling during a successful videoconference under moderate conditions. We see that the two adaptive schemes have no significant difference in delivered audio or in video loss. However, the two-dimensional scheme delivers consistently more video frames. This pattern is typical for successful runs.

Figure 5 shows a more detailed second by second comparison of quality measurements for the two conferences in this same run. In the first row we again see the equivalent audio throughput and consistently higher video throughput for two-dimensional scaling. In the second row, we see equivalent patterns of audio latency for both schemes. (Since audio is synchronized with video, video latency is similar to audio and hence is omitted.) Finally, in that last row, we see that the two-dimensional scheme's higher video throughput does not come at the cost of higher packet loss. Both schemes exhibit an acceptable level of packet loss.

Finally, we briefly discuss runs of the original, unmodified ProShare system. As before, each direction of the conference used a different congestion control scheme. In this case one direction used two-dimensional scaling, and the other direction used a fixed operating point (*i.e.*, performed no adaptations). The audio and video throughput for one such conference is shown in Figure 6. We had hoped that in some cases the two-dimensional scheme would sustain a feasible conference while the fixed point operation did not, but this was not the typical case. Instead, we again saw the simultaneous success or failure of both sides of the conference. We hypothesize that the adaptations of the two-dimensional system were capable of sufficient adaptation to enable the success of the fixed operating point system. Figure 6 provides some support for this, since a second

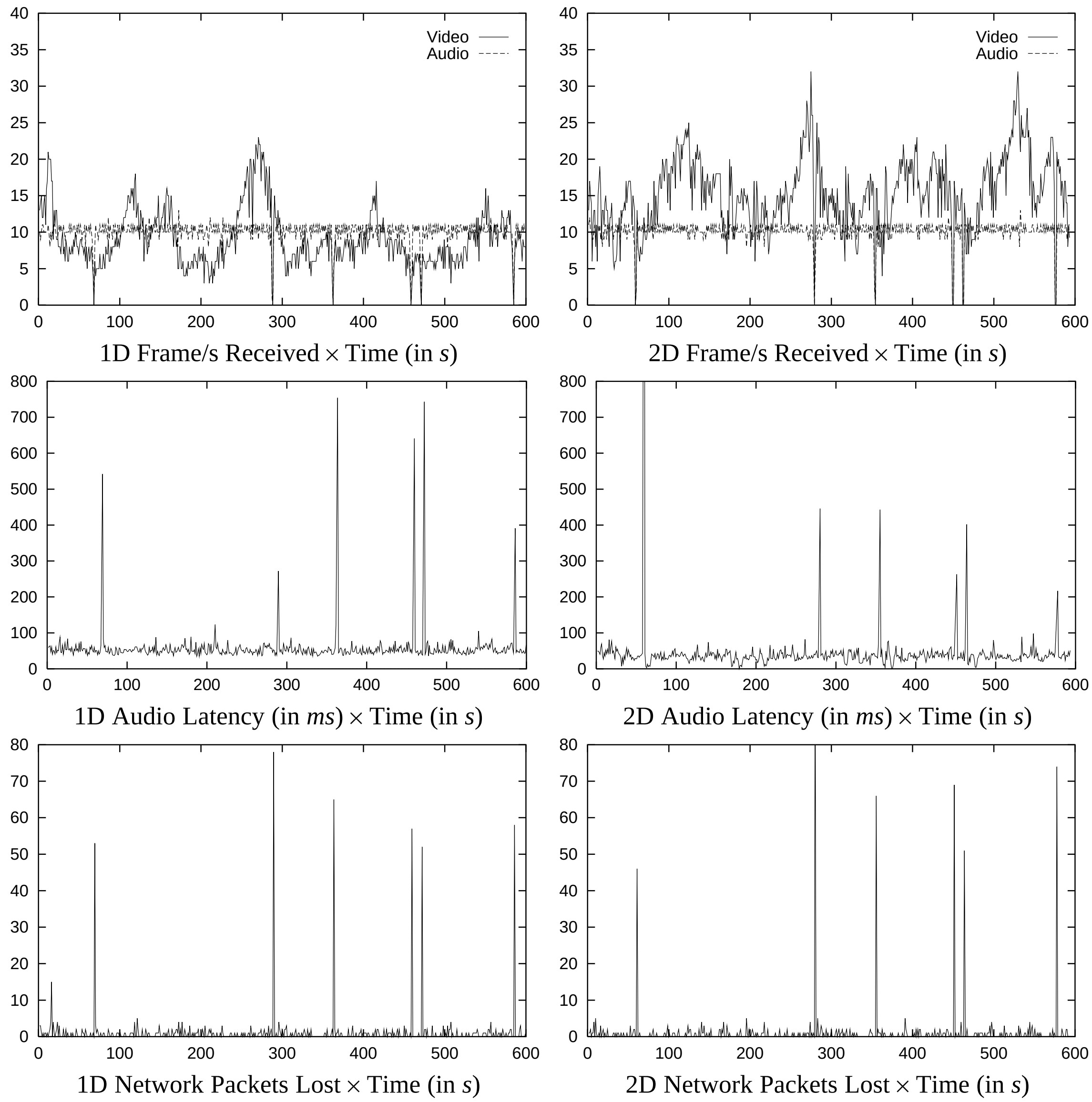

Figure 5: Sample conference performance with 1-dimensional and 2-dimensional media adaptations.

fixed operating point conferences would have exceeded the throughput that the adaptive method could sustain. Of course, there were also many runs where the level of network traffic exceeded the ability of the adaptive system to compensate for the fixed operating point system, and both sides of the conference were infeasible. The unfairness inherent in fixed point operation always severely degraded the performance of the adaptive side of the conference.

6. Discussion

Our results show that an adaptive approach to Internet videoconferencing is workable, and that receiver feedback is able to accurately characterize the level of congestion on a non-trivial Internet path. Further, at least in off-peak hours, adaptive videoconferencing is possible on today's Internet. The dynamic interplay of growth in the number of users and changes in usage and infrastructure makes the future availability of Internet resources purely a matter of speculation. However, our results provide support for a small measure of optimism for future videoconferencing and other best-effort real-time multimedia applications.

Our work further showed that it was possible to build a practical system incorporating two-dimensional scaling from an existing fixed-rate LAN videoconferencing system. This system benefited from two-dimensional scaling and benefited most in the area of operation with the greatest

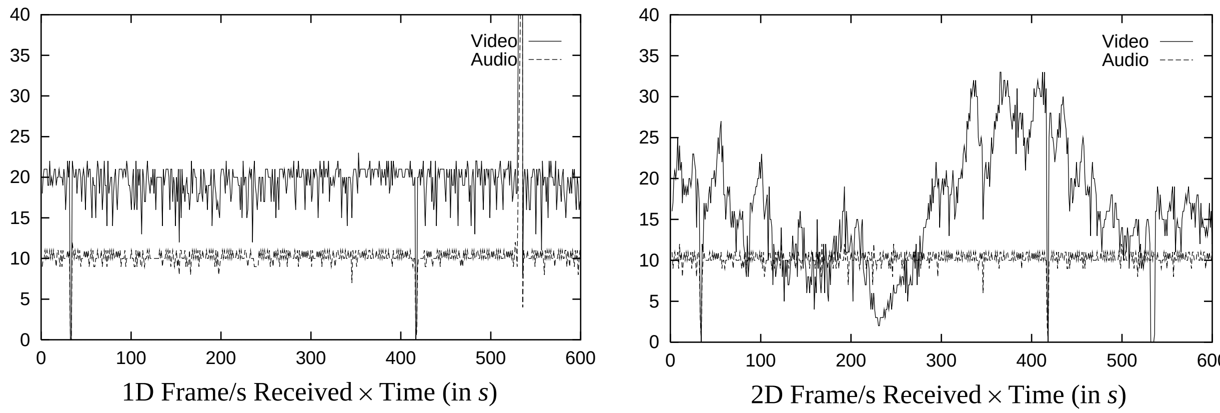

Figure 6: Audio and video throughput with fixed-rate transmission and 2-dimensional media adaptations.

range of bit-rate and packet-rate combinations. We surmise that systems with codecs expressly designed to provide even more combinations will reap even greater benefits from two-dimensional scaling.

An additional subjective observation was that occasionally the displayed video frame rate was lower than the number of video frames being sent through the network. This is attributable to synchronization or jitter problems that cause video frames to arrive too late to be useful for playout. Although infrequent, such video frames are essentially wasted transmissions, possibly as wasteful as a lost packet. (However, systems like ProShare that provide key frame data across all video frames may still benefit from the successful arrival and processing of unplayable frames.) In these cases, the detection of unplayable data in the video stream would be a useful additional factor in measuring congestion. Thus, we suggest that reports of frames usable for playout and frames unusable for playout would be a useful interface from media subsystems (*e.g.*, audio codec) to the transport layer performing congestion control.

We address our final remarks to a topic that must always be considered for high bandwidth applications that use best effort networks. This is the question of the impact of the application's congestion control mechanism on the network. An application's congestion control mechanism is most often evaluated by two criteria. First, its behavior must help protect the network from congestion collapse — a situation where the network is heavily utilized, but little or no useful throughput is achieved. Second, the high-bandwidth application's behavior must not negatively affect other applications.

We argue that the first criterion is appropriate, because an end-system application can readily assess network congestion through receiver feedback and can use this feedback to implement some form of congestion control. We note that if an application does not aggravate congestion to the point of collapse, the only harm it can do to other applications is to use more than its "fair share" of network resources. Thus, if an application's congestion control scheme adequately protects the network, protection of other applications is an issue of fairness. Since an end-system application cannot readily assess its fairness to other applications, this is not an appropriate criterion for assessing its congestion control mechanism. Working from this position, we now examine the congestion control behavior of our videoconferencing application.

Floyd and Fall [7] give three application behaviors that lead to congestion collapse. First, there is the classic case of excessive and unnecessary retransmission, generally avoided by the use of conformant TCP implementations. Second, is the successful transmission of data that is unusable because of timing or the loss of related packets (*e.g.* loss of fragments of a fragmented packet). Finally, there is the transmission of packets that make partial progress through the network and are then lost. All three behaviors contribute to congestion collapse because they waste network resources when the network is most congested.

Our system avoids or attempts to avoid all these behaviors. First, our system does not use retransmission. Second, while our system does occasionally transmit data that is not useful to the receiver, these transmissions will be associated with a symptom of congestion and lead to adaptation. One example is a video frame spanning two packets, one of which is lost. In this case, our system detects a packet loss. A second example is an audio frame that arrives too late to be used for playout. In this case, our system detects an increase in network latency, another trigger for a congestion avoiding adaptation. As mentioned above, this second example could be even better dealt with if the audio codec had some way to indicate to the transport layer that the late frame was unusable. Finally, since our system measures and adapts to packet loss, it attempts to avoid losing packets in the network.

The TCP protocol provides applications a proven congestion control mechanism that helps the network avoid congestion collapse. Clearly, any application that shapes its traffic to closely resemble the behavior of a TCP application (*i.e.*, mimics slow start, uses a congestion window, employs multiplicative decrease, *etc*.) will also meet the congestion collapse avoidance criterion for effective congestion control. However, this approach is not the best choice for every application. The quality of service goals of the application may differ greatly from those, for example, of a bulk transfer stream. In videoconferencing, occasional loss is tolerable, retransmission is unhelpful, and drastic adjustments in data rate are undesirable. An adaptive scheme that takes into account the quality of service requirements of the application it supports will clearly do a better job of minimizing the impact of congestion on that application. For example, it may retreat more slowly to avoid a jarring adjustment in an audio stream.

In addition, an adaptive scheme may benefit from techniques incompatible with TCP behavior. For example, our adaptive scheme uses more detailed feedback than TCP, avoiding overreaction due to isolated packet loss, and anticipating congestion by monitoring average latency. Our adaptive scheme also uses two-dimensional scaling, which uses aggregation to balance a packet-rate versus latency tradeoff that is suitable to the current network conditions. A typical bulk transfer application will use fixed size packets near the network MTU, thus already effectively minimizing packet rate. This is an appropriate choice for bulk transfer, because the latency of data in the packet is irrelevant. Thus, for real-time multimedia, where each media frame has an inherent latency constraint, two-dimensional scaling is a useful way to adapt a tradeoff that is not a concern for many existing TCP applications.

An adaptive scheme that behaves differently from TCP may or may not adequately protect the network from congestion collapse. For each new adaptive scheme this is an open question until demonstrated through extensive experimentation. However, the steps taken by our adaptive scheme to avoid the underlying causes of congestion collapse, and our experience with it so far, provide compelling evidence that it meets the network protection criterion for effective congestion control.

We now consider the appropriateness of the effect on other applications as a criterion for assessing an application's congestion control scheme. A specific example is the concern is that an application that adapts to congestion differently than TCP may, under the right conditions, steal resources from TCP connections. In essence, this criterion places the burden for the protection of the existing body of applications on any, and every, new application that is introduced.

We believe this criterion is inappropriate because an application has no information about the effect of its traffic on the traffic of other applications. An application can only be fair to TCP applications by behaving exactly as TCP does. Again, TCP's behavior is at best suboptimal for many types of applications, and such applications may have compelling reasons to use congestion control that differs in behavior from TCP. Further, TCP is known to be unfair in some circumstances. For example, connections with a long round trip time get a smaller share of congested links than those with a shorter round trip time.

However, the concern over fairness to TCP is legitimate; our algorithm may scale back more slowly than TCP under certain conditions, and this may result in a shift of resources from TCP traffic to videoconferencing traffic. However, the end system congestion control scheme is not the place to address this. Fairness is properly dealt with in the intermediate nodes (routers and switches) of the network. Implementations for fairness by flow, by application type, and other criteria have already been proposed [3]. Intermediate network nodes have sufficient information to deal with fairness issues accurately and efficiently. End system congestion control mechanisms do not.

An additional objection to the application fairness criterion is a recognition that new applications will inevitably be deployed, and congestion control schemes specialized for these applications will inevitably be introduced. As additional congestion control techniques are deployed, the problem of assessing each congestion control protocol's interaction with each other protocol, or with a "reasonable" population of other protocols quickly becomes untenable.

We submit that fairness to other applications is not properly a criterion for assessing an application's congestion control mechanism. Policing resource competition between applications is a job for the network itself. A congestion control scheme for an application should be concerned with meeting the quality of service requirements of the application in a way that does not waste network resources and helps avoid congestion collapse. This is the most practical, reasonable criterion for assessing application-level congestion control and initial indications are that our two-dimensional ProShare system meets this criterion.

7. Summary and Conclusions

We have presented results from experiments with an adaptive videoconferencing system on the Internet. These experiments showed that an adaptive system using inexpensive components can, with a significant rate of success, provide usable videoconferences between widely separated points on the Internet. Further, our results demonstrated that two-dimensional scaling, with its attention to appropriate choice of both bit-rate and packet-rate, provides measurable quality improvements over one-dimensional scaling on an equivalent system. Our work also pointed to the desirability of media codecs that can cooperate with adaptive transport layers to more accurately assess the value of delivered packets, and can provide a flexible array of operating points enabling a rich set of bit-rate and packet-rate combinations. Finally, we argued that the behavior of our system is consistent with reasonable criteria for application-level congestion control. Our system avoids wasting network resources through congestion control scheme based on adaptive media scaling that is appropriate for the real-time multimedia application domain.

8. Acknowledgments

We are indebted to Jorg Liebeherr at the University of Virginia for the remote use of his facilities to perform the experiments reported herein.

9. 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] Braden, R., *et al.*, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, IETF End-to-End Research Group, Internet draft (work in progress), March 1997.

[4] 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.

[5] 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.

[6] 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.

[7] Floyd, S., Fall, K., *Router Mechanisms to support End-to-End Congestion Control*, Technical Report, 1997. URL *ftp://ftp.ee.lbl.gov/papers/collapse.ps*

[8] 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.

[9] 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.

[10] Talley, T.M., Jeffay, K., *A General Framework for Continuous Media Transmission Control*, Proc. 21st IEEE Conference on Local Computer Networks, Minneapolis, MN, October 1996, pages 374-383.

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

[12] 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.