Video Multicast over the Internet *

Xue Li†	*Mostafa H. Ammar*	*Sanjoy Paul*
Bell Laboratories	College of Computing	Bell Laboratories
Lucent Technologies	Georgia Institute of Technology	Lucent Technologies
Murray Hill, NJ 07974	Atlanta, GA 30332	Holmdel, NJ 07733
lixue@research.bell-labs.com	ammar@cc.gatech.edu	sanjoy@dnrc.bell-labs.com

Abstract

Multicast (multipoint) distribution of video is an important component of many existing and future networked services. Today's Internet lacks support for quality of service (QoS) assurance which makes the transmission of real-time traffic (such as video) challenging. In addition, the heterogeneity of the Internet's transmission resources and end-systems makes it extremely difficult, if not impossible, to agree on acceptable traffic characteristics among multiple receivers of the same video stream. In this paper we survey techniques that have been proposed for transmitting video in this environment. These techniques generally involve adaptation of the video traffic carried over the network to match receiver requirements and network conditions. In addition to their applicability to the near-term capabilities of the Internet, they also are of relevance to a future, QoS-aware Internet environment because of the inevitable inaccuracies in traffic and resource reservation specifications. We first consider source-based techniques in which the source adjusts the video stream traffic to match some consensus among the receivers about its desired characteristics. These techniques can result in an unfair treatment for receivers especially those whose capability is significantly above or below the group consensus. We then consider techniques that aim to improve the fairness among the receivers by sending the video in multiple (layered or replicated) streams. We also discuss several error control mechanisms, using timely retransmission of missing data to further improve the quality of the received video. Finally, we discuss some of the issues that remain to be resolved in the development of Internet video multicast protocols.

1 Introduction

The multicast (multipoint) distribution of real-time video is an important component of many current and emerging Internet applications, such as video conferencing, distance learning, remote presentation and media-on-demand. Improvements in network delivery infrastructure and increases in end-system processing power have made these applications feasible. Multicast video distribution over the Internet requires 1) support mechanisms for multicast data delivery and 2) the ability to accommodate the real-time requirements of digital video.

The provision of multicast data delivery within the Internet has been the subject of significant research and deployment efforts recently [1, 2]. We later summarize the features of this support

*This work has been supported in part by research grants from Lucent, Intel, IBM, and the National Science Foundation grant NCR-96283793.

†Most of Xue Li's work on this paper was conducted while she was a PhD student in Georgia Institute of Technology.

relevant to our discussion here. Our focus in this paper is on how the real-time requirements of multicast digital video can be accommodated over the Internet. There are two general approaches that can be used:

- incorporating quality-of-service support within the Internet to allow video distribution applications to be able to reserve resources and establish bounds on data delivery delay, delay jitter and data loss [3, 4, 5, 6, 7].
- using adaptive rate control techniques to adjust the video traffic characteristics according to the available Internet resources [8, 9, 10, 11, 12].

We are concerned, in this paper, with the protocols that implement the second approach. Such protocols are of significance because they are compatible with the near-term capabilities of the Internet. Further, even when resource reservation and quality of service guarantees may become widely deployed, application adaptation will continue to be required to allow for tolerance to reservation inaccuracy because of the difficulties inherent in making accurate traffic specification.

The paper is organized as follows. In Section 2 we first describe the delivery context we need to work within and elaborate on the challenges of multicast video transmission over best-effort networks (such as the Internet). In Section 3 we discuss the basic rate control/adaptation and error control ideas and techniques that can be used to address the challenges described in Section 2. Sections 4, 5 and 6 discuss three basic techniques that have been proposed for providing rate adaptation: single-stream video multicast, replicated-stream video multicast and layered video multicast. Section 7 discusses error control techniques. Finally, Section 8 concludes the paper and provides a discussion of some of the open issues that need to be resolved before deployment of these protocols can proceed.

2 Context and Challenges

In this section, we first summarize the networking context within which we present our survey of techniques for video multicast delivery. Next we identify the challenges of transmitting real-time video to multiple receivers within this context.

2.1 Multicast Infrastructure within the Internet

The efficient provision of multipoint services within the Internet hinges on adequate support for the delivery of multicast packets. The exact architecture of how large-scale deployment of multicast support is still evolving within the Internet community. However, the basic features of the architecture that are relevant to this paper have been mostly worked out.

1. The Internet supports *group addressing* of multicast packets. Group addressing eliminates the need for the source to know the identity of all the receivers and provides for a scalable multicast infrastructure.
2. Packets are delivered to receivers that have declared their membership in the group over a *tree* that spans all such receivers.
3. When a new receiver joins the multicast group, a path reaching this new receiver is *grafted* onto the tree if necessary. When a receiver leaves the group, any paths within the tree that are no longer needed are *pruned*. There are different proposed multicast routing protocols

that can be used to construct the multicast tree[13, 14, 15]. The details of these protocols differ but they all share the use of grafting and pruning as a means to provide for efficient multicast routing.

2.2 Issues in Real-Time Video Multicast

The current best-effort Internet provides a challenging environment over which to transport real-time compressed digital video. First, real-time video is generated at the source in a periodic fashion (e.g., 1 frame every 1/24 second) but at a variable bit rate (i.e., the number of bits/frame varies from frame to frame). The frame periodicity needs to be preserved for playout at the receiver. Data not available at the receiver at the required playout time is considered lost. To accommodate these playout requirements, the network delay jitter (difference in the delay of packets) needs to be small. Buffering at the receiver can help absorb some delay jitter up to a limit imposed by maximum buffer availability and, in the case of interactive applications, the need to reduce end-to-end delay. Second, real-time video has a limited tolerance for random loss within the compressed digital video stream. Excessive losses resulting from network congestion (or late arrival at the receiver) can cause significant degradation of the perceived quality of the decoded video at the receiver. In cases where these losses and delays are caused by network congestion, a decrease in the data rate of the video reaching the receiver can help improve the quality of the received video. If this decrease is accomplished through changing of the compression parameters at the source, one can view such decrease as the controlled loss of data through "heavier" compression, which is always favored over random and uncontrolled losses within the network. This observation naturally leads to data rate adaptation or *rate control* as the means to overcome the lack of service guarantees within a best-effort network such as the Internet.

The effect of losses can also be overcome through the use of *error control* techniques. These techniques resemble ones used in the reliable multicasting of files or web-pages (see for example, [16, 17]). Reliable multicasting of continuous media, such as real-time audio/video is significantly different from reliable multicasting of files. Reliability for file distribution means an absolute reliable delivery of files. However, reliability for continuous media means an *improvement* over best-effort delivery. While reliable file distribution has relaxed end-to-end latency requirements, the end-to-end latency requirements for continuous media are quite stringent. Once again, there is a difference in the latency requirements for interactive real-time applications, such as conferencing, and non-interactive real-time applications, such as streaming. The former cannot tolerate a latency higher than 200 ms while the latter can have a delay budget of a few seconds.

The multicast nature of the video distribution application adds another interesting dimension, *heterogeneity*. Different receivers of the same multicast video may have different processing capabilities (important when software decoding of compressed video is used) and may have different bandwidth available in the paths leading to them. This makes the issue of adapting the video transmission to accommodate this heterogeneous collection of receivers problematic. Should one let the receiver with the least capacity dictate the adaptation? This is clearly unacceptable in most circumstances. Ignoring such a receiver can also lead to problems. As a guiding principle in the design of multicast video distribution protocols we adopt a fairness ideal that states [18]:

> Ideally, each receiver should receive video that is commensurate with its own capability and the capability of the path leading to it from the source, regardless of the capabilities of the other receivers.

We will describe later some protocols that have been proposed to approach this fairness ideal.

In addition to the *inter-receiver fairness* issue described above, one also needs to consider *inter-session fairness*, which results from the interaction of a particular multicast video session with other flows being carried over the Internet. This is of particular concern in this instance because the video multicast packets are typically carried using UDP which does not have any flow or congestion control capabilities. To make sure that network resources are being shared fairly, one would need to implement such sharing mechanisms within the video distribution protocol itself. We do not discuss this issue in great detail in this paper but point to it as a ripe area for future research in Section 8.

3 The Basic Ideas

3.1 Digital Video Bit-Rate Adaptation

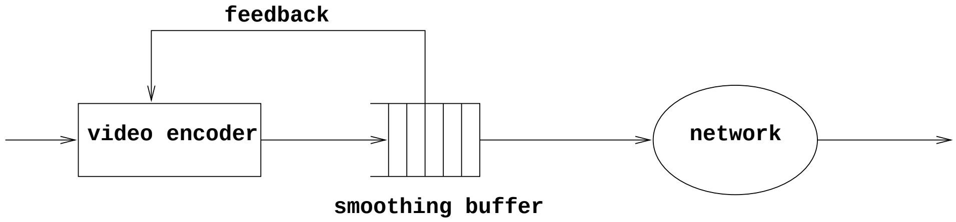

Figure 1: Video rate adaptation with constant bit rate network

The rate of a video sequence varies with the compression mode, scene complexity and motion level of the video. So, in general, the output of a video encoder is at a variable bit rate. When transmitted over a traditional constant bit rate network, e.g. telephone or cable TV networks, the video rate is adaptively controlled at the source encoder so as to maintain a constant transmission rate. This is shown in Figure 1, where a raw video stream is fed into an encoder, and then the output is sent into a smoothing buffer that is drained at a constant rate. In order to keep the constant drain rate at the buffer without overflowing or underflowing the buffer, the encoder's output should be adaptively controlled. Typically, the buffer's data level is used as a feedback signal to control the video output rate from the encoder, and the rate adaptation is achieved by adjusting certain encoding parameters, e.g., the quantization level, the frame rate or the pixel resolution. The adjustment of the video data rate through adjustment of compression parameters is sometime called *media scaling*. The paper by Delgrossi *et al* [19], reporting on video multicast within the Heidelberg Transport System HeiTS, is perhaps one of the first instances of the use of media scaling in the context of multicast video transmission. The paper contains a good description of the different scaling options.

Similar feedback control schemes are possible for the transmission of packet video over variable bit rate networks, as shown in Figure 2. Feedback in this case is derived from the network, (e.g. queuing delay information from the switches), or from the end system(s) (e.g. packet loss rate at the receiver(s)). The video rate then adapts according to the changing conditions in the network. It should be emphasized that for the current Internet, there is typically no feedback from the network and only feedback from the end systems is utilized in adaptive video transmission systems.

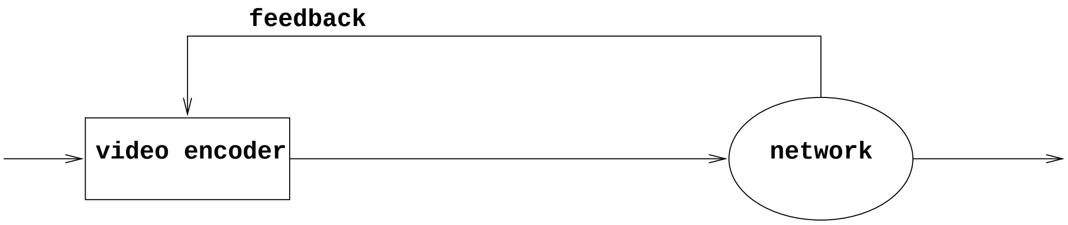

Figure 2: Video rate adaptation with variable bit rate network

3.2 Adaptive Bit-Rate Video Multicast Options

With this basic adaptation framework in mind and given the network context and challenges described earlier, one can envision three different approaches to the multicast transmission of digital video.

1. **Single Stream Adaptive Approach:** In this approach a single encoded video stream is transmitted by the source with feedback being returned from the receivers to the source. The source uses the feedback information to adapt its data rate. One of the potential problems with this approach is the problem of *feedback implosion* which can occur if there is a large number of receivers attempting to return feedback to the source. Practical video multicast protocols targeting a large number of receivers need to address this issue as we will discuss later. The single stream approach while being the most straightforward is unable to deal adequately with the heterogeneity problem described earlier.

2. **Replicated Adaptive Streams Approach:** This is a simple extension to the single-stream approach that addresses the heterogeneity issue. In this approach the source sends multiple streams carrying the same video with different quality and bit rate (obtained by encoding the different streams with different compression parameters). Each stream is multicast on a different multicast address with receivers being able to join the group that corresponds to the stream that is commensurate with their capability. Feedback from the receivers can be used to adjust the data rate of the stream they are receiving, within certain limits. One challenging aspect in the design of such a scheme is that, since receiver capabilities can be changing over time, the scheme needs to allow receivers to move among the different streams. We will summarize how this may be accomplished in an efficient manner in later discussion. While the simplicity of this scheme in addressing the heterogeneity issue is attractive, it has the problem of requiring the network to carry redundant information because the video streams replicate each other.

3. **Layered Video Streams Approach:** This scheme relies on the ability of many video compression schemes to divide their output bit stream into layers, a *base* layer and one or more *enhancement* layers. The base layer can be independently decoded and it provides a "basic" level of video quality. The enhancement layers can only be decoded together with the base layer and they provide improvements to video quality. Using this capability, a video multicast source could send each layer to a different multicast group. Receivers would then join at least the group over which the base layer is being transmitted and join as many enhancement layer groups as their capabilities allow. Here, again, we have to provide protocol mechanisms for the receivers to be able to change which enhancement layers they are receiving as their capabilities change over time. This approach provides, perhaps, the most elegant and

efficient way to deal with the heterogeneity issue. This, however, is achieved at the expense of protocol complexity as shall be seen later.

3.3 Error Control for Digital Video

Traditionally real-time video and audio have been transported over the Internet using a best-effort mechanism, because improving the quality of reception using retransmission via automatic repeat request (ARQ) protocols has been deemed impossible given the stringent latency requirements. In fact, several recent studies have shown that these notions are not well founded and it is indeed possible to improve the quality of video and audio transport by using retransmissions judiciously. The basic idea is that retransmissions should be used as long as it is possible to have the retransmitted data arrive at the receiver prior to its playout deadline.

Error control can also be done using forward error correction (FEC). In fact, using forward error correction for error control has been an acceptable solution except for the additional bandwidth requirement it may incur. PET [20] (Priority Encoding Transmission) applies pure FEC approach. It focuses on fault tolerant transmission of prioritized data over packet switched networks.The idea is that the prioritized message is encoded into packets in a such way that any destination can recover the message in priority order based solely on the number of packets received and processed.

The appropriate error control mechanisms to use can, in some cases, depend on which rate control approach is being employed. We will see some of this dependency in our discussion of the LVMR protocol later in this paper.

3.4 The Real-Time Transport Protocol (RTP)

Schulzrinne *et al* in the Audio-Video Transport Working Group in the Internet Engineering Task Force developed the Realtime Transport Protocol (RTP) [21]. RTP is the Internet standard protocol that provides end-to-end network transport functions for real-time data transmission over multicast or unicast network services. It consists of a data part and a control part. The data part of RTP is an application-layer framing [22] protocol that provides support for applications with real-time properties (e.g., timing reconstruction, loss detection, security and content identification). The control part, called Realtime Transport Control Protocol (RTCP) [21], monitors the data delivery in a manner scalable to large multicast networks, and provides minimal control and identification functionality.

RTP and RTCP are of particular interest in our discussion here since they can be used to provide the quality-of-service feedback from receivers typically required in adaptive video multicast protocols.

4 Single-stream Video Multicast: The IVS Approach

IVS (the INRIA Video-conference System) [23, 24] is an "all-software" video conference system for the Internet that employs a single stream adaptive approach.

The rate control mechanisms for IVS, implemented in the H.261 encoder, is described in [23]. To adjust the output rate of the video coder, three parameters are considered: the refresh rate, the quantizer and the movement detection threshold. The specific requirements of the video application determines which of these parameters will be adjusted when adapting the output rate of the encoder. Feedback information is based on packet loss measured at the receivers, and packet loss is detected

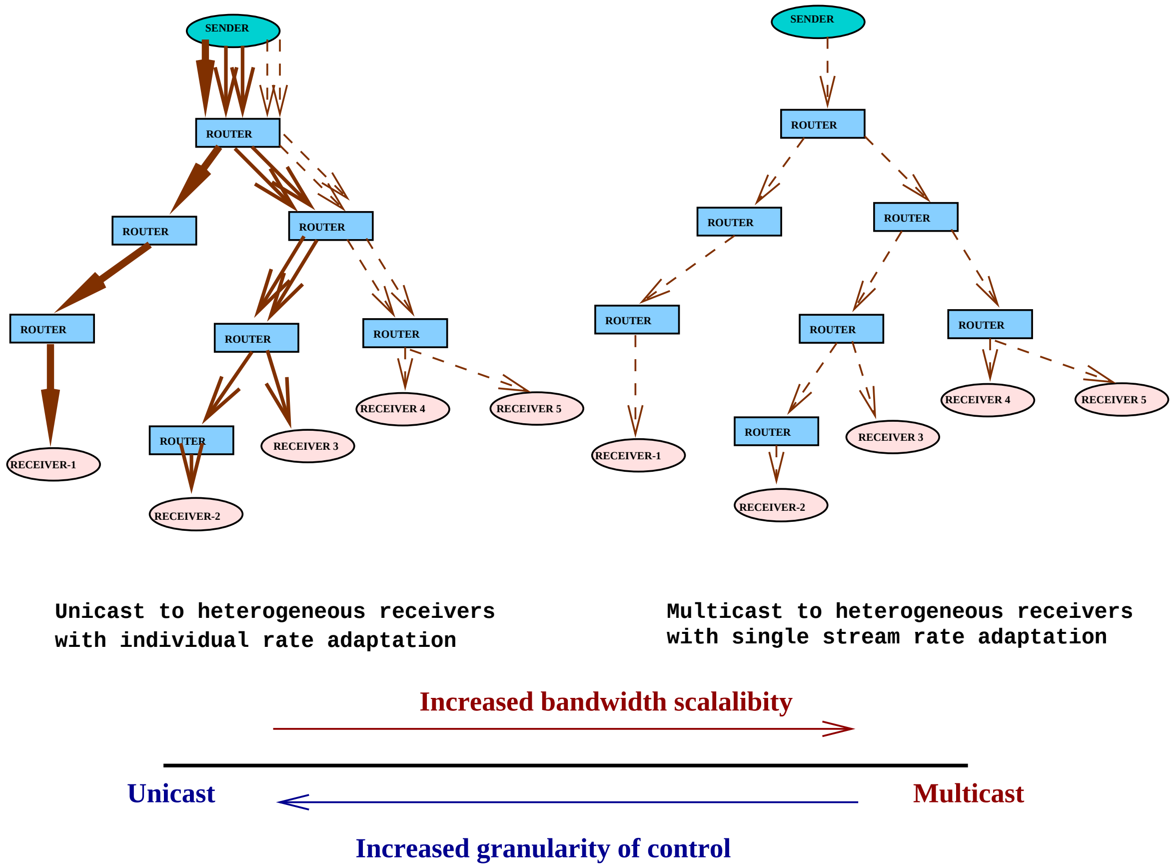

Figure 3: Tradeoff between video multicast and unicast

using the RTP sequence number. The control protocol for RTP, RTCP (Realtime Transport Control Protocol), is used to send *reception reports* that provide feedback information.

When a video source multicasts to a large group of receivers, if every receiver sends feedback to the source, it may cause the *feedback implosion* problem: the network gets congested and the sender gets overwhelmed. Ammar proposed in [25] the *probabilistic multicast* technique to address this problem. A *probabilistic multicast* message is only accepted by members in the multicast group with a certain probability, and only those members respond. Bolot, Turletti and Wakeman [24] developed a mechanism similar to probabilistic multicast in IVS : they proposed a *probing mechanism*, to solicit feedback information in a scalable manner.

As mentioned earlier, a potential problem of IVS and any single-stream video multicast is its inability to provide fair treatment to multiple receivers in a heterogeneous environment. To elaborate on this problem we observe that multicast communication makes a tradeoff between economy of bandwidth and *granularity of control*, as demonstrated in Figure 3. If multicast video were to be distributed using individual feedback-controlled point-to-point streams, the bandwidth economy is low but the granularity of control is very high because the sender can negotiate the setting of communication parameters individually with each receiver. In contrast, using a single multicast stream has good bandwidth economy, but very low granularity of control; the setting of communication parameters may not be optimal for many of the receivers. The replicated stream multicast approach described in the next section is one way to operate a video multicast protocol that spans the middle region of this Unicast/Multicast spectrum.

5 Replicated-Stream Video Multicast

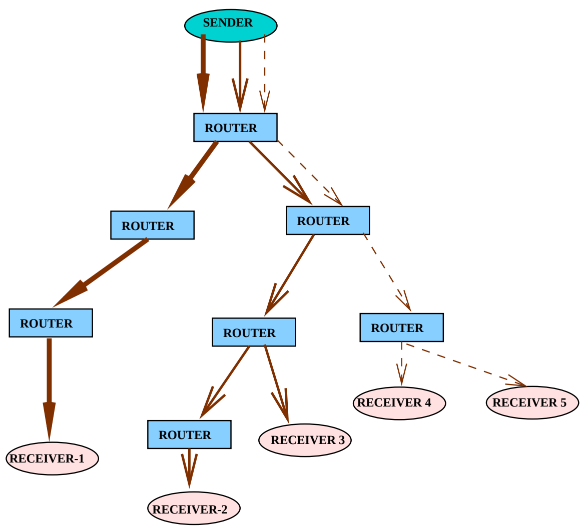

Figure 4: Replicated-Stream Video Multicast with DSG Protocol

In order to address the *fairness* issue in feedback-controlled multicast video distribution, Cheung, Ammar and Li designed and implemented a protocol called Destination Set Grouping (DSG) [18, 26]. In DSG, as shown in Figure 4, the source keeps a small number of video streams carrying the same video but each targeted at receivers with different capabilities. Each stream is feedback-controlled within prescribed limits by its group of receivers. Receivers may move among the streams as their capabilities or the network capabilities change.

5.1 Intra-stream and Inter-stream Protocols

The fundamental design goals of the DSG video multicast protocol are: improved fairness over a single-group feedback-controlled video multicast scheme and the ability to operate efficiently when the number of receivers is large (commonly referred to as *scalability*). Fairness is achieved in DSG by transmitting video of differing quality and differing data rates on different multicast channels and allowing receivers to select the most appropriate one. The DSG protocol is highly scalable because the stream change decisions are made by receivers. Receivers are provided with the necessary information to make the correct stream change decisions. The authors also use a slightly modified version of the probabilistic feedback technique [25, 24] to avoid feedback implosion.

The DSG protocol has two main components, as illustrated in Figure 5:

1. An *intra-stream* protocol used by receivers listening to the same stream to adjust the data rate of the stream within its prescribed limits. For this the authors use a variation on the protocol described in [24]. An independent feedback control mechanism used within each stream. Each receiver estimates its video reception quality using its packet loss rate.

2. An *inter-stream* or *stream change* protocol used by receivers to change to a higher or lower quality stream as their needs change. The inter-stream protocol allows a receiver to change to a different stream in situations where they cannot adjust the rate of the stream they are currently receiving to their satisfaction.

5.2 A DSG Experiment

The DSG protocol was implemented and tested over the Mbone. To illustrate the protocol's behavior consider the following three experiments that were performed over the Mbone [27], with the sender in Georgia Tech (in Atlanta) and receivers in Georgia Tech, Emory University (also in Atlanta) and the Oregon Graduate Institute (OGI):

1. A set of unicast experiments, one to each receiver, allowing dynamic transmission rates corresponding to the full range of MPEG 1 quantization scale factor levels [8..31] [1]. Each unicast experiment used the DSG protocol multicasting to a group of size 1. These experiments served as the "baseline" with which the performance of the other schemes can be compared.

2. Single-stream multicast with dynamic transmission rates corresponding to the full range of quantization scale factor levels [8..31]. This essentially mimicked the behavior of the single-stream multicast approach described earlier.

3. Replicated-Stream (three) multicast: G_1 using quantization scale factors in the range [8..15], G_2 in [16..23] and G_3 in [24..31]. This was an experiment using the DSG protocol running in a general three-stream setting.

The results of the above experiments are illustrated in Figures 6, 7, and 8 respectively, where the received data rate is plotted as a function of time. Comparing Figure 6 to Figure 7, it is observed that multicasting single-stream video to heterogeneous receivers results in everyone receiving at very low data rate and thus poor video quality. Comparing with what they are capable of handling in a unicast experiment, as shown in Figure 6 (around 150kbps), the fast machines (Sun Sparc 20s) in Georgia Tech and Emory all receive a much lower data rate (around 50kbps) in Figure 7, because the video stream is adapted to match the capacity of the receivers in OGI. While with the DSG protocol, as shown in Figure 8, each receiver initially receives the lowest quality stream, and then more capable receivers move to streams with higher quality. In the steady state, (1) the slow machines (Sun Sparc LX) in Georgia Tech and those in OGI receive the lowest quality video stream; (2) the Sun IPC in Emory receives the higher video quality stream with the same data rate as what it receives in the unicast experiment; and (3) the other faster machines receive the highest quality stream, and compared with in unicast experiment (Figure 6), the data rate on the Sparc 20's is only 13% lower. Fairness among receivers is improved significantly over a single-group approach while incurring only a small additional bandwidth overhead.

5.3 Bandwidth Control for the DSG Protocol

Carrying replicated streams on network links in the DSG protocol can result in congestion on those links. Li and Ammar proposed bandwidth control mechanisms [26] for DSG, in order to limit the overloading effects caused by network links carrying multiple streams. Two bandwidth control schemes are applied at each receiver: (1) *Congestion History Checking* is conducted before stream

[1] A larger value corresponds to a lower data rate for the encoded stream.

switch to avoid oscillatory receiver join behavior, and (2) *Local Area Bandwidth Limiting* provides heuristics to limit the number of streams received within a certain locality. *Overall Bandwidth Limiting* mechanisms provide macro control at the video source to attempt to limit the overall bandwidth of the multicast video session.

Despite the techniques described in [26] for controlling the bandwidth used in the DSG protocol, this issue may remain somewhat problematic in some environments. The layered techniques we describe next provide an approach that achieves better bandwidth efficiency at the price of more complexity.

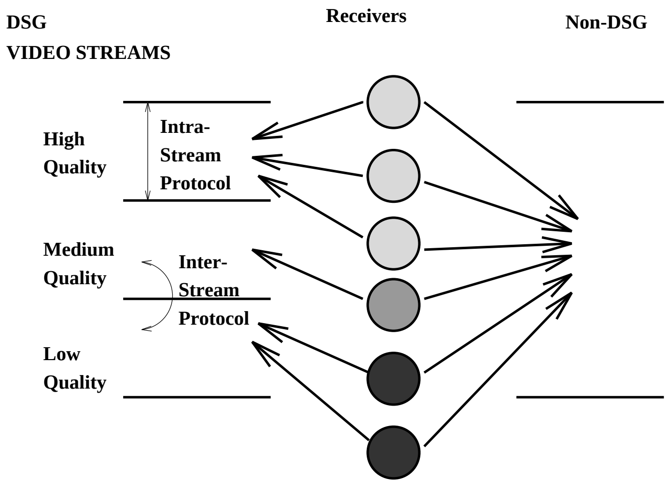

Figure 5: Intra-stream and inter-stream protocols in DSG

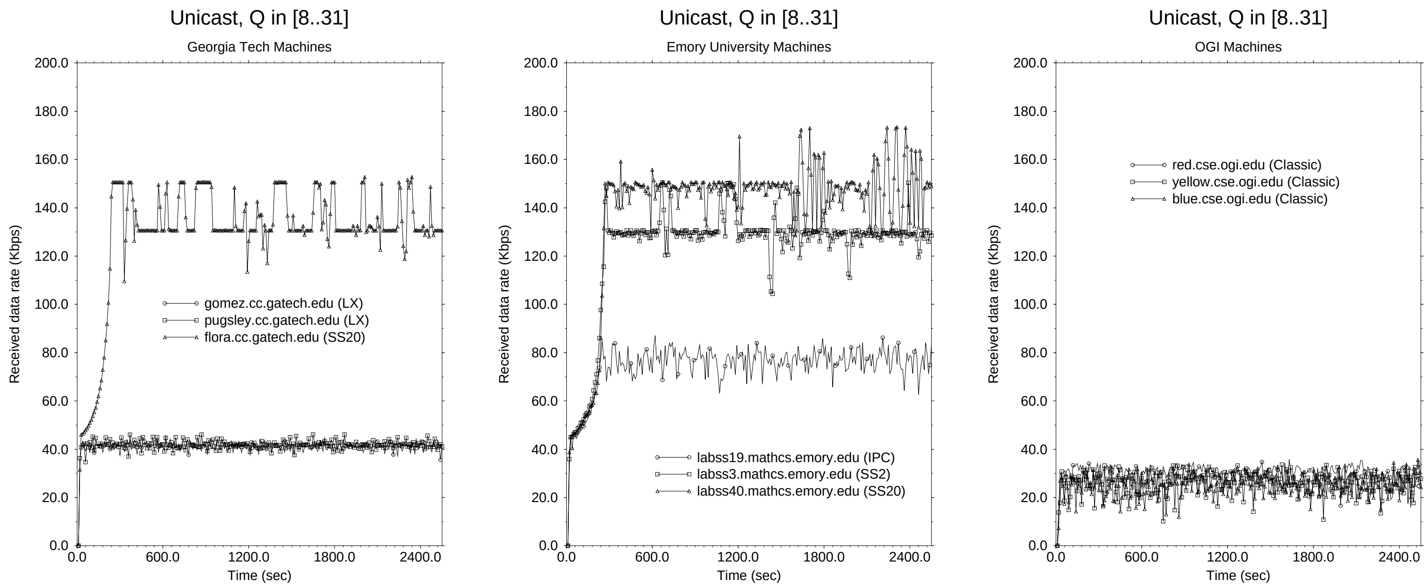

Figure 6: DSG experiment results: from the unicast experiment

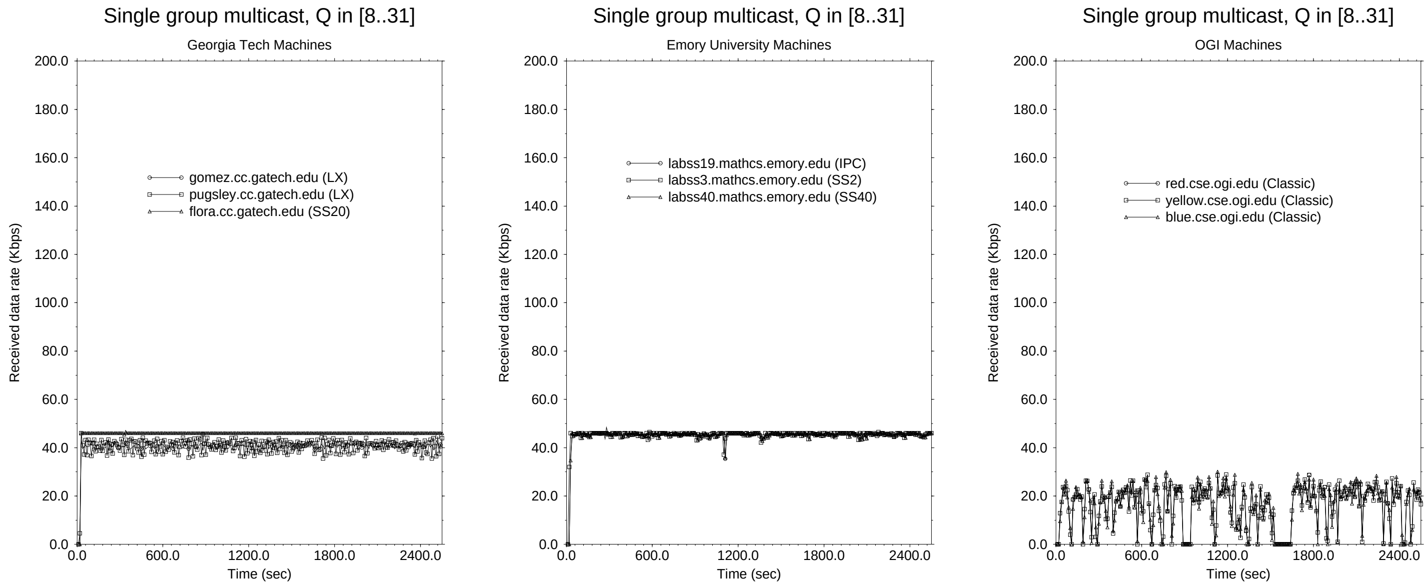

Figure 7: DSG experiment results:single group multicast experiment with full range quantization scale factors

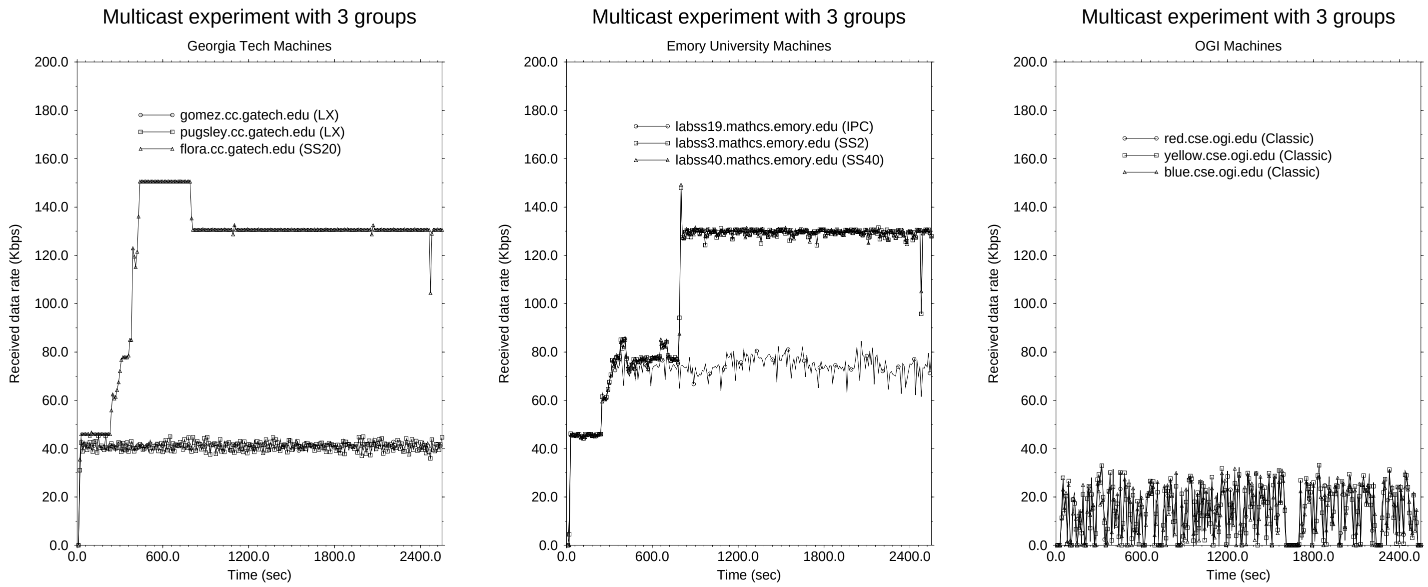

Figure 8: DSG experiment results:multicast experiment with 3 groups

6 Layered Video Multicast

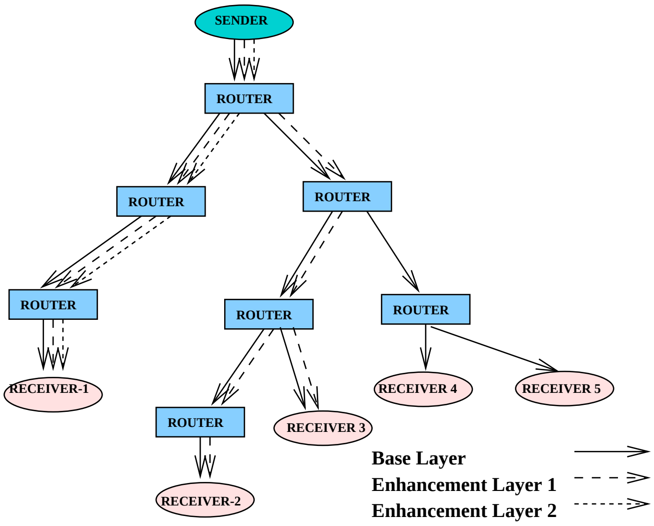

Figure 9: Layered Video Multicast

The layered video multicast approach, is illustrated in Figure 9. As discussed earlier, the sender sends out video in multiple layers, and each receiver receives a subset of the video layers commensurate with its processing power and network bandwidth availability. Video layering can be supported by many video compression techniques. The MPEG-2 International Standard [28] supports layered encoding by defining four *scalable modes*. There are also other techniques for providing video layers (see for example the PVH (Progressive Video with Hybrid transform) technique in [29], the DCT (Discrete Cosine Transform) co-efficient split technique in [30]) and multirate 3-D subband video coding [31].

6.1 Layered Video Multicast Protocols: an Overview

Layered multicasts provide a finer granularity of control compared to using a single video stream, because a receiver may subscribe to one, two, or more layers depending on its capabilities and, unlike in the scheme in [18], does not increase the bandwidth requirement because the layers are disjoint.

In [32], a paradigm was proposed for multicasting video in a heterogeneous environment, where each destination receives a subset of the source's signal that corresponds to that destination's processing and access bandwidth constraint. Shacham reviewed layered coding schemes, presented methods for finding maximum bandwidth available to the destinations, developed schemes to optimally assign bandwidth to the layers to maximize overall reception quality, and proposed error control procedures. This work has the assumption of admission control and resource reservation.

The HeiTS media scaling [19] schemes include *discrete scaling* for multicast. The scheme is built on the same philosophy that receivers should receive a certain level of the multimedia layers commensurate to their bandwidth constraint. The rate control is receiver-initiated and each receiver opens or closes ST-II multicast connections to receive certain layers of the multicast stream. But the authors did not present how the receivers decide their appropriate reception layers.

Based on IP multicast, Deering suggested a new approach for layered video multicast [33]: different video layers are sent in different multicast groups and receivers adapt to the appropriate set of reception layers by joining or leaving the multicast group. In the rest of this section, we will survey two rate control protocols in layered video multicast over IP networks.

6.2 Receiver-driven Layered Multicast (RLM)

McCanne, Jacobson and Vetterli proposed the Receiver-driven Layered Multicast protocol (RLM) [8], to support rate adaptation for layered video transmission over IP multicast.

6.2.1 Basic Rate Adaptation Schemes

In RLM, the sender sends each video layer to a separate IP multicast group and takes no active role in rate adaptation. Each receiver subscribes to a certain set of video layers by joining the corresponding IP multicast group. The advantage of receiver-based control over sender-based control is that the burden of adaptation is moved from the sender to the receivers, resulting in enhanced system scalability.

Each receiver tries to achieve the optimal level of subscription of video layers. The basic adaptive control is: when a receiver detects congestion, it drops a layer, and when there is spare bandwidth available, it adds a layer. Deciding whether the current reception level is optimal is a crucial step in the control loop. The video level is too high if congestion is detected, and in RLM, congestion is detected with packet loss rate. On the other hand, there is no explicit feedback on whether the current level is too low, so in RLM, *join- experiments* are carried out to find if the receiver is capable of handling the next video layer. If congestion is detected after the experiment, the receiver drops the newly added layer, otherwise if no congestion is observed, the experiment is successful.

A failed join-experiment can bring congestion to the network, resulting in degraded video quality to both the receiver who initiated the experiment, and possibly other receivers that share the congested link. Therefore, a learning algorithm is proposed so as to minimize the frequency and duration of join-experiments without impacting video layer convergence rate. To decrease the frequency of failed join-experiments, a *join-timer* is set for each video layer and exponential backoff is applied on the layers that would possibly bring congestion.

6.2.2 Shared Learning

A fully distributed approach is advocated in the receiver-driven rate adaptation, but if each receiver conducts its own rate adaptation control, the system will have poor scalability. Instead of each receiver adapting its rate independently, RLM proposes *shared learning* so that all receivers can learn from other receivers' failed join-experiments. Before a receiver starts a join-experiment, it informs the whole multicast group about this experiment by multicasting a message to the group address stating the specific layer it is going to try to join.

The idea of shared learning, although an improvement to adding and dropping layers indiscriminately, requires each receiver to maintain a variety of state information which it may or may not require. In addition, the use of multicasting to exchange control information may decrease usable bandwidth on low speed links and lead to lower quality for receivers on these links. So this scheme helps to improve scalability, but has a potential drawback of unnecessary bandwidth and message processing overhead to a lot of receivers. Adding some intelligence in the *shared learning* scheme can make it more efficient, and we will discuss a related scheme in Subsection 6.3.

6.2.3 Simulation Results

Simulation work has been conducted for RLM to explore the scalability in some simple network topologies. Two evaluation metrics are considered: (1) the worst-case packet loss rate over varying time windows: short-term loss rate represents the extent of transient congestion while long term loss rate reflects frequency of congestion occurrence in the steady state; (2) the time it takes each receiver in an RLM system to converge to the optimal reception of video layers. Some of the results presented in [8] are summarized below.

- Latency Scalability.

  When the link delay increases, the duration of join-experiments also grows as it takes longer to detect congestion.

- Session Scalability.

  The size of the receiver group is varied in the experiments, and the results show that the worst-case loss rate is essentially independent of the session size. When the number of receivers increases, however, longer convergence time to the optimum layer reception is expected since high layer join-experiments will be suppressed.

- Bandwidth Heterogeneity.

  The experiments show that the algorithm works well with a large number of receivers with heterogeneous bandwidth constraints, since the worst-case loss rate in the heterogeneous case is comparable with the homogeneous bandwidth experiments.

- Superposition.

  Experiments are conducted to explore the performance of independent video sessions sharing a common bottleneck link. The aggregated utilization on the shared link converges to close to one, but bandwidth allocation to each session is often unfair.

6.2.4 A Related Protocol: SCUBA

Amir, McCanne and Katz proposed the SCUBA (Scalable ConsenUs Bandwidth Allocation) protocol [34] that enables a video source to intelligently account for *receiver interest* in their rate-adaptation algorithms.

The basic idea is to reflect receiver interest back to the sources in a multicast session using a scalable control mechanism. Two variants of SCUBA are developed: a "flat delivery" variant to complement sender-based adaptation and a "layered delivery" variant to complement receiver-based layer adaptation. For layered delivery, SCUBA distinguishes more important sources from less important ones by assigning layers from different sources to network channels with different priorities. SCUBA computes source weights and then maps each source's layers to corresponding network channels.

6.3 Hierarchical Rate Control in LVMR

Layered Video Multicast with Retransmissions (LVMR) is another system for distributing video using layered coding over the Internet. The two key contributions of the system are: (1) improving the quality of reception within *each* layer by retransmitting lost packets given an upper bound on recovery time and applying an adaptive playback point scheme to help achieve more successful retransmission [35], and (2) adapting to network congestion and heterogeneity using Hierarchical Rate Control (HRC) mechanisms [9]. The first contribution will be discussed in the next section, while here we concentrate on the HRC schemes.

6.3.1 Hierarchical Rate Control and Comparison with RLM

In contrast to the existing sender-based and receiver-based [8] rate control in which the entire information about network congestion is either available at the sender (in the sender-based approach) or replicated at the receivers (in the receiver-based approach), the hierarchical rate control mechanism *distributes* the information between the sender, receivers, and some *agents* in the network in such a way that each entity maintains only the information relevant to itself. In addition to that, the hierarchical approach enables intelligent decisions to be made in terms of conducting concurrent experiments and choosing one of several possible experiments at any instant of time based on minimal state information at the agents in the network.

Comparing with RLM, HRC takes a hierarchical approach in the receivers' dynamic rate control schemes, so as to allow receivers to maintain minimal state information and decrease control traffic on the multicast session. It also provides more functionality compared to the simple receiver-driven schemes, as in RLM. In particular, it allows multiple experiments to be conducted simultaneously, and also helps drop the correct layer(s) during congestion in most cases; both are not possible in RLM because of its completely distributed approach.

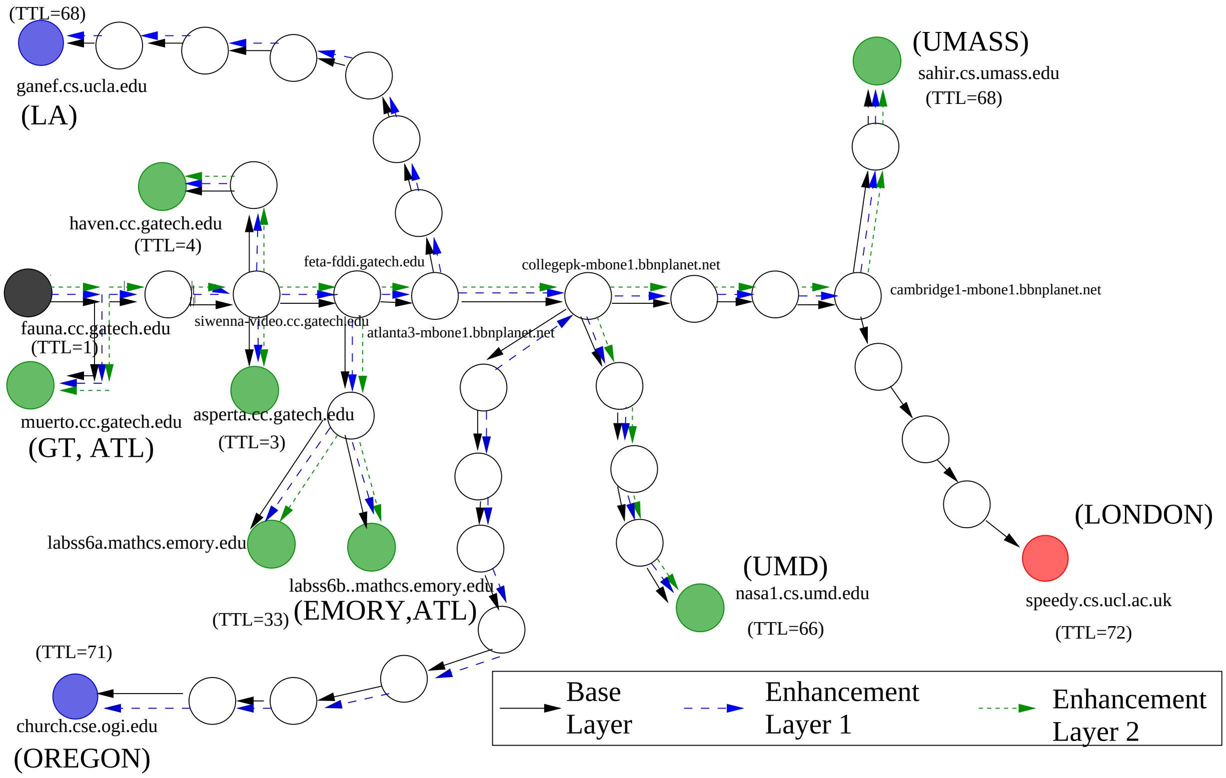

Figure 10: Topology of the Mbone Test and a Sample Video Layer Distribution

The key to scalability in layered multicast is for the receivers to make a decision on their own regarding adding or dropping a layer. However, if these decisions are made independent of

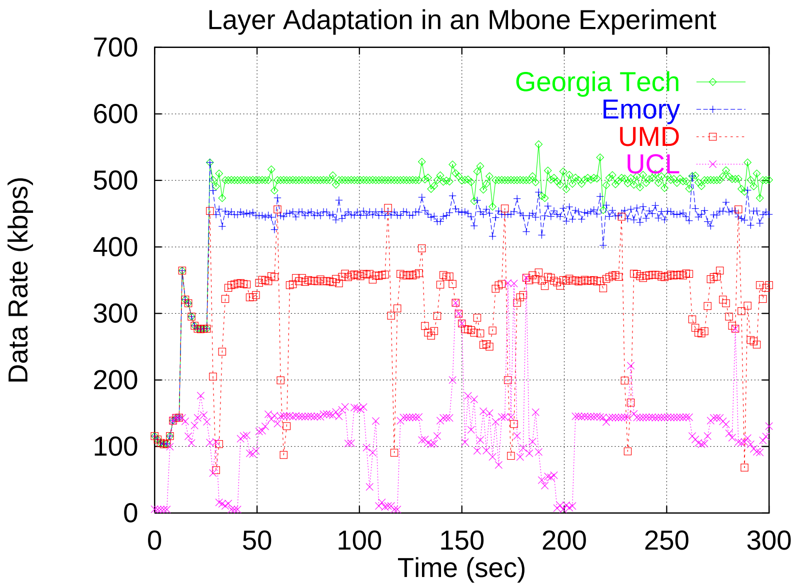

Figure 11: Video Layers in One Mbone Experiment

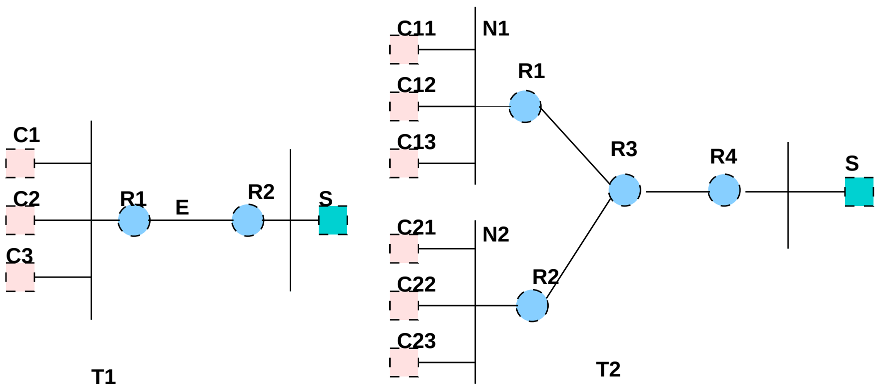

Figure 12: Topologies of Simulation

the results of join/leave experiments done by others, the results can be disastrous. Thus, it is fundamental for each receiver to know about the experiments and their results. The mechanism used in the RLM to achieve *shared learning* has the following drawbacks:

1. Every receiver need not know about every experiment and/or its result. That is just too much state information.

2. Using multicast to distribute control information, such as, beginning of experiments and their results, beyond a certain scope is inefficient, because it consumes additional bandwidth, particularly if every receiver need not know about every experiment and/or its results.

The solution in LVMR tries to avoid the above drawbacks of RLM by using *intelligent partitioning* of the knowledge base and distributing relevant information to the members in an efficient way. Note that there are several receivers in a multicast group, potentially distributed over a large heterogeneous network, running several experiments, some with success and some with failure. If all

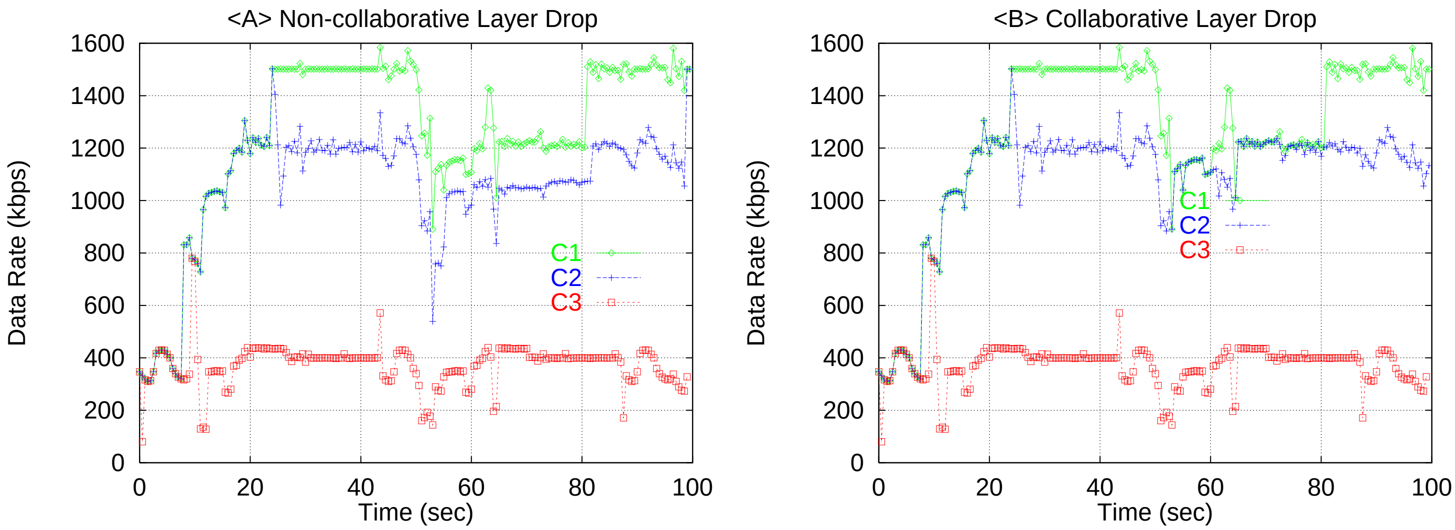

Figure 13: Layer Drop: Non-collaborative and Collaborative

these experiments along with their results are compiled into a knowledge base, that would represent the *comprehensive group knowledge base* . LVMR provides schemes to partition the comprehensive group knowledge base in an intelligent way [9].

In Hierarchical Rate Control schemes, LVMR proposes a hierarchy of *affected regions* to show the domain of receivers that can get affected when an add-layer experiment is performed. Within an affected region, LVMR provides heuristics to find which receivers can be congested by a failed add-layer experiment. LVMR also incorporates a *Collaborative Layer Drop* feature that is used to achieve more efficient layer adaption in time of congestion.

6.3.2 Experimental Results

A prototype system for LVMR with IP multicast was developed over the MBone, distributing MPEG II streams. The topology for the experiments is shown in Figure 10

Figure 11 shows the data rates and video layers some of the receivers received during one Mbone experiment. The Mbone video was sent out from Georgia Tech with frame rate of 8 f/s. All the receivers started to join only the base layer and after a certain time, all receivers except the one in UCL joined the first enhancement layer, and later the second enhancement layer. The Georgia Tech and Emory receivers both ended up receiving all three layers, but the node in Georgia Tech observed almost no loss while the receiver in Emory had some constant loss around 10%. The machine in UMD received 2 layers most of the time although it launched add-layer experiment from time to time. The receiver in UCL suffered a loss of around 20% and it could only join the base layer.

The authors also simulated the hierarchical rate control protocol with a modified *ns* [36] version 1.4. Figure 12 lists some of the topologies used in the simulation tests.

- Collaborative layer drop.

  Figure 13 shows how the receivers adapt their layers when the shared link between R_2 and R_1 is congested. The congestion is caused by constant bit rate background traffic (1-kbyte packet every 0.03 sec) on E between time 50 sec and 80 sec.

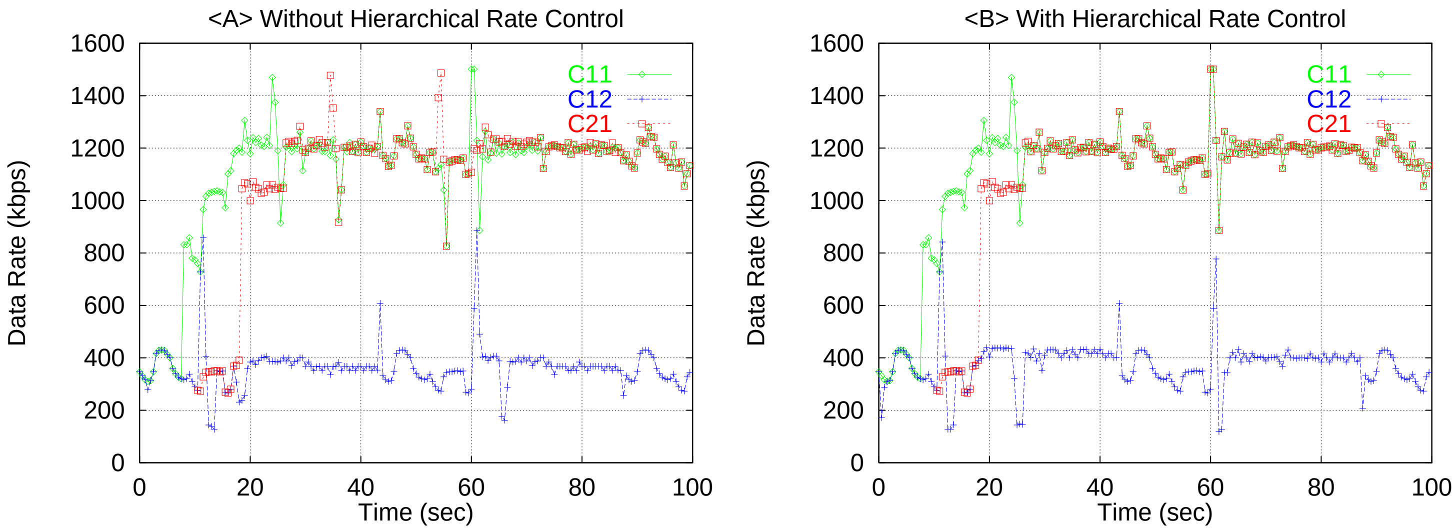

Figure 14: Layer Add: Without or With Hierarchical Rate Control

The results show that in time of congestion, collaborative layer drop (1)decreases layer oscillation, (2)achieves more effective rate adaptation, and (3)maintains better video reception quality.

- Add-layer experiment.

  Figure 14 shows the effect of Hierarchical Rate Control on the experiment on Topology T2. Topology T2 includes 2 subnets N_1 and N_2 each with 3 receivers, and the agent is set to be on receiver C_{22}. The video processing speed on C_{11} and C_{21} is 2.0Mbps and 400kbps on other receivers. C_{21} joins the video session late at time 10 sec, and it adds layers up to 3 at time 27 sec, while by then C_{11} receives 3 layers and has already conducted a failed add-layer experiment to join layer 4.

  The results in Figure 14 show that when adding layers, Hierarchical Rate Control (1)decreases unnecessary add-layer experiment, and (2) provides smoother video quality.

6.4 Other Approaches

Wu, Sharma and Smith proposed *Thin Streams* as a solution to the join/leave problem in layered video multicast [12]. Each video layer is divided into multiple thin streams of equal bandwidth, and each thin stream is multicast in a separate multicast group. Thin Streams help to decrease network oscillation caused by receivers joining and leaving multicast groups with high bandwidth corresponding to video layers, but there is the drawback of network overhead of supporting a large number of multicast groups and receiver processing overhead of synchronizing multiple thin streams within each layer.

Robinet, Au and Banerjea implemented an application level gateway for connecting adaptive multimedia applications using hierarchically encoded video across ATM and IP networks [11]. The gateway participates in RSVP signaling (in the IP network) and UNI signaling(in the ATM network). The receivers utilize such signaling information, together with local processing load information and packet loss information (in the absence of RSVP). Network capacity information is also sent back to the video source so as to feed-back control the bit rates of the video layers. The authors in [11] addressed the problem of dealing with the trade-off between layer granularity and overhead in their work. The basic idea is to group the thin streams into fat streams before transmitting, and dynamically adapt the grouping.

6.5 Multi-session Rate Control

A problem with RLM [8] and LVMR [9] is that the protocols do not provide fair bandwidth sharing between competing video sessions or between video sessions and TCP sessions. Li, Paul and Ammar proposed [37] an end-to-end [38] control scheme: layer-based congestion sensitivity rate control, to improve inter-session fairness when used to augment such video multicast protocols. The basic idea is to let higher video layers have higher sensitivity to congestion. This will cause receivers to drop higher layers more easily when competing for bandwidth with receivers receiving lower layers only. The result is that competing receivers end up with the same (or close to the same) number of layers. The paper also demonstrates that this scheme achieves TCP friendliness [39] and bounded TCP fairness.

Lorenzo, Rizzo and Crowcroft proposed TCP-like congestion control [40] for layered multicast data transfer and video multicast is listed as one of its applications. The bandwidth of a layer is set to be twice as much as the next lower layer. Because of this exponential relationship, when a receiver drops a layer during congestion, it behaves like TCP's window reduction algorithm and hence is TCP-friendly. Unfortunately, for video multicast, layering is determined more by the coding requirements than by the bandwidth requirements and hence the layering scheme proposed in [40] may bring some potential problems when applied to video multicast.

Multi-session rate control has also been addressed from a receiver's interest point-of-view in the SCUBA protocol [34]. SCUBA proposes to map the layers of different sessions into a "global" layering scheme based on receivers' overall feedback and thereby providing a receiver-driven rate control infrastructure. However, mapping layers of each session into a global network-wide layering scheme does not scale with the number of sessions. Also, the whole approach is based on RTCP-like multicast feedback which may generate unnecessary control traffic in the network because every member of the multicast group need not know about every other member's interest vector

6.6 Replicated-stream vs Layered Video: A Simple Comparison

The most significant differences between the two approaches are bandwidth economy and processing overhead.

- In general layered encoding gains bandwidth economy, and replicated streams usually result in multiple streams on some shared links. However when using replicated streams, if different receiver groups share very few links, the bandwidth overhead cost is not very high. Also for the same quality of video after decompression, sending it in one stream takes less bandwidth than sending it in layers. So in certain cases, for example, if the receiver groups of replicated streams are disjoint, then using the replicated streams scheme consumes less bandwidth than a layered encoding scheme.

- Layered encoding schemes require a more complex codec and there is overhead in the encoding and decoding time. At the receiver's side, it also requires more buffer size and overhead in synchronization of different layers before decoding.

Certain similarity can also be observed in the protocols applying the two approaches. Receivers listen to specific IP multicast addresses to receive certain streams or layers. Distributed control schemes are used both at the receivers and the sender. The total number of layers/streams is decreased when there is less heterogeneity and increased otherwise.

7 Error Control

In the context of video multicast there has been three approaches to error control:

1. Retransmission-based (ARQ) approach[35, 41, 42, 43]
2. Pure FEC (Forward Error Correction) approach[20]
3. Hybrid FEC-ARQ approach [44, 45]

In this section we will focus on the retransmission-based schemes since they have received the most attention in the literature.

7.1 Layered Video Multicast with Retransmission (LVMR)

The key ideas in LVMR [35] are:

1. Use a statically configured logical tree at the transport layer with Designated Receivers (DRs) at each level of the tree to help with retransmission of lost packets. This technique is similar to the one employed by RMTP [16].
2. Improve efficiency by not asking for retransmissions of those packets which cannot be recovered in time. This reduces control traffic and also retransmission traffic.
3. Improve response time by sending immediate NAKs and multicasting retransmissions.
4. Use buffers to not only absorb the jitter but also to increase the likelihood of getting retransmitted packets before the deadline.
5. Combine retransmission mechanisms with layered encoding and layered transmission and thereby increase the latency budget for high-priority layers when low-priority layers are dropped.

Some details are provided below.

7.1.1 Local Recovery

Given that there is a maximum budget for end-to-end latency, it is essential to recover lost packets as quickly as possible. Typically video frames are larger than transport-layer packets. Each frame is thus divided into multiple packets and when one or more packets belonging to a frame are lost, a partial frame is received.

Thus the problem is, when the round-trip time between the sender and the receivers is larger than the end-to-end delay budget, it is impossible to recover lost packets corresponding to a partial frame by relying on retransmissions from the sender. This is shown in Part(a) of Figure 15.

The key to solving this problem is to reduce the recovery time for the lost packets by using an entity closer to the receiver for retransmissions. This entity is the designated receiver (DR) [16] in a tree-based hierarchical framework (Figure 16). Part(b) of Figure 15 illustrates how retransmissions from a nearby entity, such as, a DR helps to meet the end-to-end latency requirements.

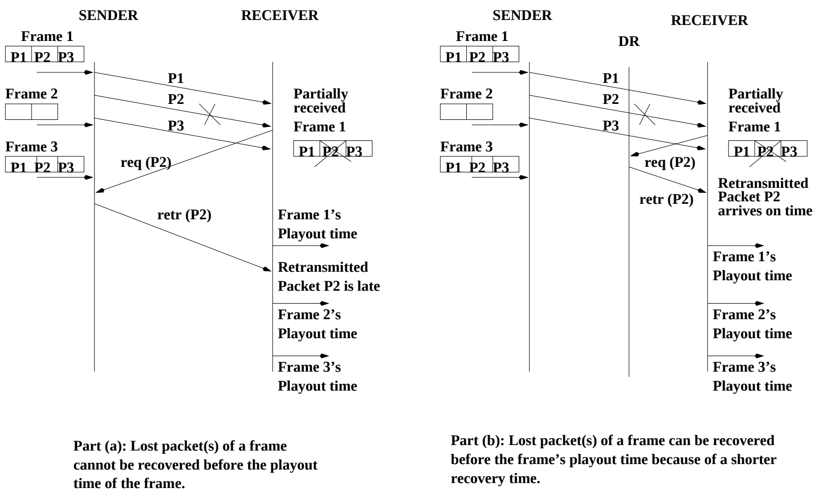

Figure 15: Local recovery helps to improve video quality by retransmissions

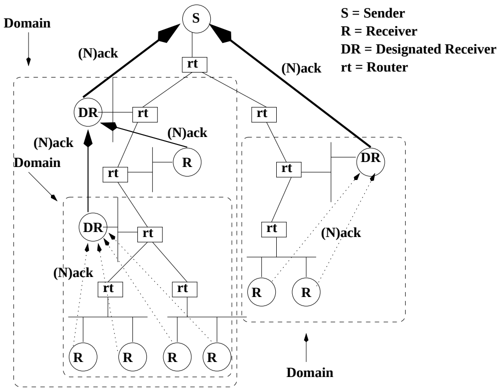

Figure 16: Designated Receiver (DR) for local recovery

7.1.2 Adaptive Playback Points with Extended Control Time

Since video frames need to be played back in a periodic manner at a receiver, the playback point of frame i is fixed once the playback point of frame 0 is determined and the frame rate is known. For example, if the playback point of frame 0 is p_0 and the frame rate is R frames/second, the playback point of frame i is $p_i = p_0 + i * 1/R$.

Typically p_0 is related to the arrival time t_0 of frame 0 as: $p_0 = t_0 + \Delta$ where Δ is the maximum jitter in the network. Typically Δ is *adapted* based on network delays. LVMR does not focus on the adaptation of Δ. There are schemes proposed by other researchers [46] to deal with this issue.

In LVMR, the playback point p_0 is extended by an additional amount δ which is referred to as *control time*. That is, $p_0 = t_0 + \Delta + \delta$. Since $p_i = p_0 + i * 1/R$, $\delta \leq (p_i - t_i) \leq \Delta + \delta$. That is, there is at least a budget of δ time units after the arrival of a frame (at time t_i) to recover any missing fragments. Thus *control time* is a parameter that provides an additional cushion of time which can be used for recovering video frames. This notion is referred to as using *extended control time* for *adapting* the playback point.

In addition, during congestion, the more important video layers have a longer time cushion for *repair*. This is the second type of *adaptation* of the playback point used in LVMR. More details

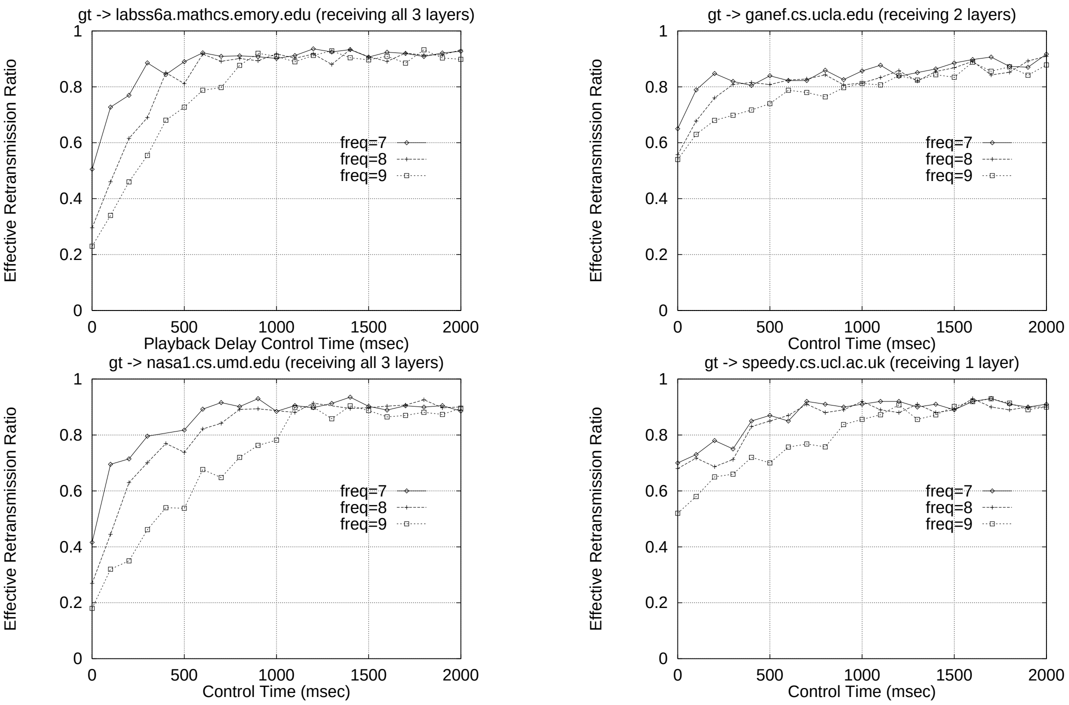

Figure 17: Multicast Experiments over Mbone: 4 Destinations

can be found in [35].

7.1.3 Experiments

Several experiments were performed for LVMR over the Mbone to test how effective the extended control time, adaptive playback point and retransmission requesting schemes are. The topology for the experiment was the same as shown in in Figure 10.

We summarize some results dealing with the proportion of lost packets effectively retransmitted. A packet is *effectively retransmitted* if the retransmission arrives before the playback point of the frame that this packet belongs to. Let ER denote the total number of effectively retransmitted

Receiver	Layers	Control Time				
		0sec	0.5sec	1.0sec	1.5sec	2.0sec
labss6a.mathcs.emory.edu	3	0.23	0.72	0.90	0.89	0.89
labss6a.mathcs.emory.edu	2	0.50	0.81	0.86	0.91	0.89
labss6a.mathcs.emory.edu	1	0.67	0.89	0.90	0.90	0.90
sahir.cs.umass.edu	3	0.16	0.39	0.65	0.85	0.83
sahir.cs.umass.edu	2	0.41	0.75	0.85	0.87	0.87
sahir.cs.umass.edu	1	0.52	0.78	0.84	0.85	0.86
nasa1.cs.umd.edu	3	0.18	0.53	0.78	0.85	0.89
nasa1.cs.umd.edu	2	0.42	0.80	0.85	0.84	0.88
nasa1.cs.umd.edu	1	0.55	0.79	0.86	0.86	0.89

Table 1: Effective Retransmission Ratio vs Control Time and Video Layer (Freq=9)

packets during a video session and L total number of packets lost (whether retransmitted or not). *Effective retransmission ratio* is defined as ER/L.

As shown in Figure 17 and Table 1 [2], the effective retransmission ratio increases with δ. When it is around 1.5 sec, a majority of the lost packets can be recovered for the receivers and this shows the feasibility of retransmission for real-time video distribution over the Internet. Note also that the frame rate and the number of video layers received by a receiver affected the effective retransmission ratio. More experimental results may be found in [35].

7.2 STORM

While LVMR uses a statically configured logical tree at the transport layer, meaning that the receivers always bank on the same DRs for retransmissions, STORM [41] (Structure-Oriented Resilient Multicast) provides a mechanism for each receiver to dynamically select the best possible DR at any instant of time. The key ideas in STORM are:

1. Use a *dynamically changing* logical tree at the transport layer to improve the probability of getting retransmissions in time.

2. Let the receivers decide on their own the trade-off between latency and reliability.

Some details are given below.

Building the recovery structure When a receiver first joins a multicast group, it uses an expanding ring search (ERS) to look for potential parent nodes (designated receivers). An ERS consists of sending queries to the multicast group with increasing values of TTL. Members which are already a part of the logical tree structure send unicast replies with an indication of their perceived loss as a function of the playback delay. When the new members collects enough responses, it stops the ERS.

Selection of Parent Nodes A member selects its parent based on the packet loss rate of the candidate parent nodes, its one-way delay from the candidate parent nodes and its own playout buffer size. For example, if the member has a playout buffer of 200 ms and one of the two potential parents received 85% of the packets within 20 ms and 90% of the packets within 100 ms, while the second potential parent received 75% of the packets within 20 ms and 95% of the packets within 140 ms, the member will choose the latter node as its parent. More details can be found in [41].

Adapting the structure The logical tree structure built by STORM changes over time as the network conditions change. For example, if some part of the network gets congested, a parent may fail to provide with timely recovery. Each receiver periodically computes the ratio of the number of successful repairs from a parent to the number of NACKs sent to the parent. If this ratio falls below a threshold, the receiver starts the ERS process once again.

7.2.1 Performance Results

Experiments were conducted over the Mbone by incorporating STORM and SRM [17] in the Mbone audio program *vat*. In addition STORM and SRM were compared using network simulation.

[2] Freq. in the figures refers to the number of video frames transmitted per second.

Site	Initial		Initial	
	STORM	SRM	STORM	SRM
Berkeley	3.71%	4.11%	0.01%	0.09%
Ga. Tech†	4.37%	4.02%	0.00%	0.29%
ISI	3.82%	3.97%	0.04%	0.11%
UCLA†	3.82%	3.97%	0.04%	0.11%
Kentucky	10.19%	6.88%	0.52%	0.62%
UMASS†	10.65%	14.46%	0.05%	0.68%
Virginia	42.95%	45.57%	0.17%	22.67%

Table 2: Loss rate without recovery and with recovery in one experiment. Hosts marked with † used 200 ms playout buffer while the rest used 500 ms playout buffer.

The Mbone experiment of STORM consisted of 8 sites, namely, Berkeley, UCLA, ISI, UMASS, CMU, Ga. Tech, University of Virginia and University of Kentucky. Only one result from the Mbone experiment of STORM is provided here (Table 2). More details can be found in [41].

The point to observe here is that the performance of STORM and SRM are comparable in most cases except for the sites with high initial loss rates (such as, University of Virginia), where STORM outperformed SRM by a large margin.

Simulation results show that the average overhead per host remains a small constant as the group size increases from 10 to 400. Thus STORM scales very well with group size. In addition, the dynamic choice of parent nodes in simulation experiments reduced the average loss rate from 1.3% to 0.28%, indicating that the adaptive nature of STORM can provide tangible benefits.

7.3 Client-Server Recovery Architecture

The key ideas in the Client-Server approach proposed in [42] are:

1. Separate the actual senders and receivers from a repair architecture consisting of a retransmit server Svr (located close to the sender) and several repair servers located in various locations of an intranet) as shown in Figure 18. The repair servers are called clients ($Clnt$) which ask for missing packets from the retransmit server and generate an improved version of the original video transmission by using repair buffers.

2. Receivers either subscribe to the original video stream from the server (indicated by solid lines in Figure 18 or to the repaired video stream from the repair server (indicated by dashed lines).

3. The retransmit server and the repair servers can be organized into a logical tree structure as in RMTP/LVMR.

Some details are given below.

Using RTP The client-server protocol is built on top of RTP and hence can take advantage of the sequence numbers used in RTP header. These sequence numbers are used for indicating missing packets in NACKs. NACK suppression is performed by causing receivers to delay sending of NACKs until either their timer expires or someone else sends the same NACK. Since repairs are sent by the repair servers, there is no repair implosion.

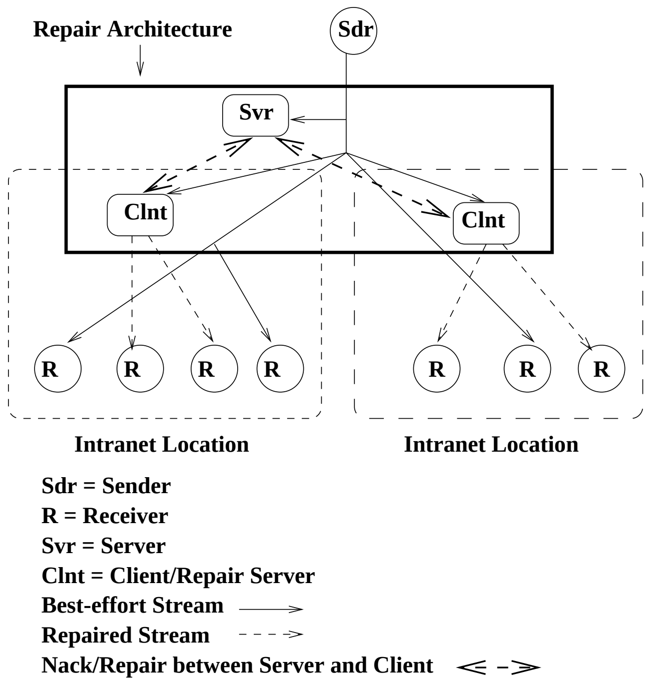

Figure 18: Client-Server Architecture

Repair Buffer Repair buffers in the repair servers are used exactly in the same way the playout buffer of LVMR is used. That is, repair buffers not only absorb the jitter in the network, but they also allow for a few retransmissions for recovering a lost packet. The prototype implementation used video streams with bandwidths between 128 kbps and 384 kbps. Thus a buffer of size 0.5 MB is enough to support a 10 second delay. In addition, less than 5 seconds of buffering is enough to allow more than two attempts at recovering a lost packet. This result agrees with the results of LVMR which showed that 2 second of buffering is enough to recover close to 90% of the lost packets on the MBone.

7.3.1 Experimental Results

Experiments were performed using the client-server architecture for multicasting both audio and video from a Lucent location in Indian Hill near Chicago to Murray Hill, New Jersey. Typical packet loss rate between these two locations is 6-15% on the average for a 60-minute session with short-term loss rates as high as 30-40%. The results showed that at most two retransmissions were sent while most of the lost packets were recovered using a single retransmission. More details can be found in [42].

8 Concluding Remarks

Open-loop, constant bit rate unicast streaming of video is already in use over the Internet today. The inability to adapt the video traffic to network conditions inherent in the streaming approach has two important implications:

- Network congestion can lead to random losses which can result in serious degradation of the video quality
- Non-adaptive streams of video data do not share resources well in a best-effort network.

Also, multicast video has the added complexity of a heterogeneous set of receivers and network paths to deal with.

In this paper we have surveyed the ideas and some protocols that have been proposed with the objective of providing adaptive and error-controlled video multicast over the Internet. These efforts, however, represent only initial attempts to provide a comprehensive practical solution to the problem. Many issues need to be resolved before deployment of such protocols on a large scale over the Internet can become a reality. These issues include:

1. Further investigation in building protocol mechanisms to provide for the co-existence and resource sharing of video multicast streams amongst each other and among TCP flows in large scale networks. We have already surveyed some work that can provide starting points for this direction of research. An unsolved problem is whether to deploy purely end-to-end solutions [38], or to develop schemes based on the infrastructure with gateways and agents, or a hybrid solution.
2. Formalization of the notion of fairness amongst receivers of a multicast video session. A more quantitative understanding of this performance measure will be important in the design and tuning of video multicast protocols. The work reported in [47] may be of interest to this problem.
3. Understanding the effect of pricing on the behavior of receivers in a multicast video environment. Most of the work so far makes the assumption that each receiver is always interested in getting the highest possible video quality stream they can get. If pricing is used to provide an incentive system, then receiver behavior may be quite different and this can be used to influence the design of video multicast protocols. There is very limited work in this area. The study reported in [48] begins to address this question.

The list above represents a set of very hard questions. However, the need to multicast video over the Internet is expected to intensify in the near future. Resolution of these and other issues (some discussed in the body of the paper) is, therefore, essential.

9 Acknowledgments

The authors would like to thank the reviewers for their detailed and very helpful comments and pointers to further references.

10 Acronym List

Here is the list of acronyms used in this paper:

DCT: Discrete Cosine Transform
DSG: Destination Set Grouping
DR: Designated Receiver
ERS: Expand Ring Search

FEC: Forward Error Correction
HeiTS: Heidelberg Transmission System
HRC: Hierarchical Rate Control
IVS: INRIA Video-conferencing System
LVMR: Layered Video Multicast with Retransmission
MPEG: Moving Picture Expert Group
PET: Priority Encoding Transmission
PVH: Progressive Video with Hybrid transform
QoS: Quality of Service
RLM: Receiver-driven Layered Multicast
RMTP: Reliable Multicast Transport Protocol
RTP: Realtime Transport Protocol
RTCP: Realtime Transport Control Protocol
SCUBA: Scalable ConsenUs Bandwidth Allocation
SRM: Scalable Reliable Multicast
STORM: Structure-Oriented Resilient Multicast
TTL: Time To Live

References

[1] M. H. Ammar, G. Polyzos, and S. Tripathi, "Special issue on network support for multipoint communications," *IEEE Journal on Selected Areas in Communication*, vol. 15, April 1997.

[2] J. C. Pasquale, G. C. Polyzos, and G. Xylomenos, "The Multimedia Multicasting Problem," *ACM Multimedia Systems*, vol. 6, no. 1, pp. 43 – 59, 1998.

[3] A. Banerjea, D. Ferrari, B.Mah, M. Moran, D. Verma, and H. Zhang, "The Tenet Real-time protocol Suite: Design, Implementation, and Experience," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 1 – 10, Febuary 1996.

[4] D. Ferrari and D. C. Verma, "Scheme for Real-Time Channel Establishment in Wide-Area Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 368 – 379, April 1990.

[5] R. Bettati, D. Ferrari, A. Gupta, W. Heffner, W. Howe, M. Moran, Q. Nguyen, and R. Yavatkar, "Connection Establishment for Multi-Party Real-Time Communication," in *Proceedings of ACM NOSSDAV'95*, pp. 255 – 266, 1995.

[6] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource reservation protocol," *IEEE Network Magazine*, vol. 7, pp. 8 – 18, September 1993.

[7] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview." RFC 1633, June 1994.

[8] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proceedings of ACM SIGCOMM '96*, (Stanford, CA), September 1996.

[9] X. Li, S. Paul, and M. H. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of hierarchical rate control," in *Proceedings of IEEE Infocom'98*, (San Francisco, CA), March 1998.

[10] X. Li, *Scalable and Adaptive Video Multicast over the Internet.* PhD thesis, Georgia Institute of Technology, 1998.

[11] J. Robinet, Y. Au, and A. Banerjea, "Implementation of a gateway for hierarchically encoded video across ATM and IP networks," in *Proceedings of High Performance Networking Workshop (HPN)'98*, (Vienna, Austria), September 1998.

[12] L. Wu, R. Sharma, and B. Smith, "Thin Stream: An architecture for multicasting layered video," in *Proceedings of NOSSDAV'97*, (St. Louis, MI), May 1997.

[13] S. E. Deering, "Multicast routing in inter networks and extended lans," *ACM Computer Communications Review*, vol. 19, no. 4, pp. 55 – 64, 1988.

[14] S. E. Deering, D. Estrin, D. Farinacci, V. Jacobson, L. Ching-Gung, and L. Wei, "An architecture for wide-area multicast routing," in *ACM SIGCOMM'94*, pp. 126 – 135, August 1994.

[15] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees: An architecture for scalable inter-domain multicast routing.," in *Proceedings of ACM SIGCOMM '93*, September 1993.

[16] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, 1997.

[17] S. Floyd, V. Jacobson, C. G. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," in *Proceedings of ACM SIGCOMM'95*, September 1995.

[18] S. Cheung, M. H. Ammar, and X. Li, "On the use of Destination Set Grouping to improve fairness in multicast video distribution," in *Proceedings of IEEE INFOCOM '96*, (San Francisco, CA), pp. 553 – 560, March 1996.

[19] L. Delgrossi, C. Halstrick, D. Hehmann, R. G. Herrtwich, O. Krone, J. Sandvoss, and C. Vogt, "Media scaling in a multimedia communication system," *ACM Multimedia Systems*, vol. 2, pp. 172–180, 1994.

[20] A. Alanese, J. Blomer, J. Edmonds, and M. Luby, "Priority Encoding Transmission," Tech. Rep. RT-94-039, Department of Computer Science, University of California, Berkeley, 1994.

[21] H. Schulzrinne, S. Casner, R. Fredrick, and V. Jacobson, "RTP: A transport protocol for real-time applications." Internet Engineering Task Force, Audio-Video Transport Working Group, RFC 1889., January 1996.

[22] D. Clark and D. Tennenhouse, "Architectural considerations of a new generation of protocols," in *Proceedings of ACM SIGCOMM'90*, pp. 201 – 208, September 1990.

[23] J.-C. Bolot and T. Turletti, "A rate control mechanism for packet video in the Internet," in *Proceedings of IEEE INFOCOM'94*, pp. 1216 – 1223, March 1994.

[24] J.-C. Bolot, T. Turtelli, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," in *Proceedings of ACM SIGCOMM'94*, pp. 58 – 67, September 1994.

[25] M. H. Ammar, "Probabilistic multicast: Generalizing the multicast paradigm to improve scalability," in *Proceedings of IEEE INFOCOM'94*, March 1994.

[26] X. Li and M. H. Ammar, "Bandwidth control for replicated-stream multicast video distribution," in *Proceedings of HPDC'96*, (Syracuse, NY), August 1996.

[27] H. Eriksson, "Mbone: The multicast backbone," *Communications of the ACM*, vol. 37, pp. 54 – 60, 1994.

[28] P. Tudor, "MPEG-2 video compression," *Electronics and Communications Journal*, December 1995. http://www.bbc.co.uk/rd/pubs/papers/paper_14/paper_14.htm.

[29] S. McCanne, *Scalable Compression and Transmission of Internet Multicast Video*. PhD thesis, UC Berkeley, 1996.

[30] P. Pancha and M. El-Zarki, "Prioritized transmission of Variable Bit Rate MPEG video," in *Proceedings of Globecom'92*, 1992.

[31] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 35, pp. 572 – 588, 1994.

[32] N. Shacham, "Multipoint communication by hierarchically encoded data," in *Proceedings of IEEE Infocom'92*, pp. 2107 – 2114, March 1992.

[33] S. E. Deering, "Internet multicast routing: State of the art and open research issues," in *Multimedia Integrated Conferencing for Europe (MICE) Seminar at the Swedish Institute of Computer Science*, October 1993.

[34] E. Amir, S. McCanne, and R. Katz, "Receiver-driven bandwidth adaptation for light-weight session," in *Proceedings of ACM Multimedia'97*, (Seattle, WA), November 1997.

[35] X. Li, S. Paul, P. Pancha, and M. H. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of error recovery," in *Proceedings of NOSSDAV'97*, (St. Louis, MI), May 1997.

[36] S. McCanne and S. Floyd, "The UCB/LBNL network simulator." software online. http://www-mash.cs.berkeley.edu/ns/.

[37] X. Li, S. Paul, and M. Ammar, "Multi-Session Rate Control for Layered Video Multicast ," in *Proceedings of Symposium on Multimedia Computing and Networking (MMCN99),*, (San Jose, CA), January 1999.

[38] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet." submitted to IEEE/ACM Transaction of Networking, February 1998.

[39] J. Mahdavi and S. Floyd, "TCP-friendly Unicast Rate-based Flow Control ." Technical note sent to the end2end-interest mailing list, available at http://www.psc.edu/networking/papers/tcp-friendly.htm l, September 1997.

[40] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *IEEE INFOCOM'98*, (San Francisco, CA), March 1998.

[41] X. R. Xu, A. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous-media applications," in *Proceedings of NOSSDAV'97*, (St. Louis, MI), May 1997.

[42] N. Maxemchuk, K. Padmanabhan, and S. Lo, "A cooperative packet recovery protocol for multicast video," in *Proceedings of 1997 International Conference on Network Protocols (ICNP'97)*, (Atlanta, GA), pp. 259 – 266, October 1997.

[43] B. Dempsey, J. Liebeherr, and A. C. Weaver, "A new error control scheme for packetized voice over high-speed local networks," in *Proceedings of IEEE 18th Local Computer Networks Conference*, (Minneapolis, MN), pp. 91 – 100, September 1993.

[44] D. Rubenstein, J. Kurose, and D. Towsley, "Real-time reliable multicast using proactive Forward Error Correction," Tech. Rep. 1998-19, Department of Computer Science, University of Massachusetts, Amherst, 1998.

[45] M. Podolsky, C. Romer, and S. McCanne, "Simulation of FEC-based error control for packet audio on the internet," in *IEEE INFOCOM'98*, (San Francisco, CA), March 1998.

[46] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive playout mechanism for packetized audio applications in wide-area networks," in *Proceedings of IEEE INFOCOM'94*, March 1994.

[47] T. Jiang, M. H. Ammar, and E. W. Zegura, "Inter-Receiver Fairness: A novel performance measure for multicast ABR sessions," in *Proceedings of ACM SIGMETRICS'98*, (Madison, Wisconsin), June 1998.

[48] A. Krishnamurthy, T. D. C. Little, and D. Castanon, "A pricing mechanism for scalable video delivery," *ACM Multimedia Systems*, no. 4, pp. 328 – 337, 1996.