

# Controlling Quality of Session in Adaptive Multimedia Multicast Systems

Alaa Youssef\*, Hussein Abdel-Wahab, and Kurt Maly  
Department of Computer Science  
Old Dominion University  
Norfolk, VA 23529  
{youssef, wahab, maly}@cs.odu.edu

## Abstract

*Controlling the quality of collaborative multimedia sessions, that deploy multiple media streams, is a challenging problem. In this paper, we present a framework for achieving quality of session (QoS) control, focusing on two main components of the QoS control layer. The first component is a scalable and robust feedback mechanism which allows for determining the worst case state among a group of receivers of a stream. This mechanism is used for controlling the transmission rate of multimedia sources in the cases of layered and single-rate streams. The second component is the inter-stream bandwidth adaptation mechanism that dynamically controls the bandwidth shares of the streams belonging to a session. We compare the performance of several adaptation algorithms. Additionally, in order to ensure stability and responsiveness in the inter-stream adaptation process, several measures are taken, including devising a domain rate control protocol. The performance of four mechanisms is analyzed and their advantages are demonstrated by simulation and experimental results.*

## 1. Introduction

Explosive growth in the deployment of new network technology has created many opportunities for *Interactive Multimedia Collaborative (IMC)* applications. IMC applications are being developed for distance learning/training, scientific and engineering cooperative efforts, Internet games and tele-meetings, to mention just a few areas. However, the real time requirements of the multimedia streams, deployed by IMC applications, demand special treatment. The two basic approaches for handling the requirements of multimedia streams are the proactive approach and the reactive approach. The proactive approach relies mainly on

the existence of a resource reservation protocol [5, 13], and underlying scheduling mechanisms, to reserve and guarantee end-to-end resources. On the other hand, the reactive approach relies mainly on the ability of the application (senders and receivers) to adapt itself to the level of available resources in the face of network and computer load imbalances [2, 3, 6, 8].

A key issue that characterizes most of the proactive and reactive approaches, taken for handling multimedia streams, is the management of the *Quality of Service (QoS)* offered to individual connections in isolation of others. We believe, however, that the QoS offered by the system should be dynamically controlled across the set of connections belonging to the IMC application, in order to avoid any potential competition for resources among the session's streams. This control should be based on the application semantics, with the objective of maintaining the best overall quality of session, at every instant in time. The QoS control layer constantly monitors the observed behavior of the streams, takes inter-stream adaptation decisions, and sets the new operating level for each stream from within its range of permissible operating points. Over a wide area network, in the presence of a resource reservation protocol such as RSVP [13], the QoS control layer manages the resources that are collectively reserved for the streams of a distributed application.

The QoS layer is composed of two main components: monitoring agents, and inter-stream adaptation (**ISA**) agents. An overview of the design of the two agents is given, together with explanation of the underlying design principles. Much emphasis is directed in this paper towards two building blocks of the QoS control layer: scalable and robust state feedback, and inter-stream bandwidth adaptation mechanisms.

The state feedback mechanism is embedded in the monitoring agents and serves to provide the source of a multimedia stream with deterministic information regarding the state of its receivers. The state of a receiver is defined as the layers which it is interested in receiving from that

---

\*Supported in part by IBM. Author current address: IBM T.J. Watson Research Center, 30 Sawmill River Road, Hawthorne, NY 10532.

source. Given this knowledge, the sender can suppress or start sending the needed layers. The feedback mechanism is not only important for saving the sender's host and LAN resources but for saving WAN resources as well in situations where the IMC application's addressing scheme for the layers does not permit the intermediate routers to suppress unwanted layers, or where the IMC session is conducted over an Intranet whose subnets are inter-connected via low level switches that do not implement the IGMP protocol [4] for suppressing multicast packets for which no receivers exist on the subnet.

Soliciting information from receivers in a multicast group might create a *reply implosion* problem, in which a potentially large number of receivers send almost simultaneous redundant feedback messages. Typical solutions to address this problem include probabilistic reply, expanding scope search, statistical probing, and randomly delayed replies [2]. We present a scalable and robust state feedback mechanism, which extends the concept of randomly delayed replies with suppression of redundant replies. The proposed mechanism does not rely on periodic session messages.

The inter-stream bandwidth adaptation mechanism is deployed by the ISA agent, in order to allocate the bandwidth available to a receiver among the different streams in a way that enhances the overall session quality, from that receiver's perspective. We have devised a new inter-stream adaptation algorithm, which does not require knowledge about the session bandwidth, and which accounts for the fact that typical multimedia sources are able to vary their transmission rates in discrete steps only [11]. In this paper, we show the advantages of our new algorithm by comparing its performance to other known inter-stream bandwidth adaptation algorithms [1, 12].

## 2. Architecture Overview

Figure 1 illustrates the architecture of the QoSSess control layer. In the figure, the IMC application is modeled as a set of sender and receiver units and a session manager (SM) unit, for clarity. In general, no specific requirements are imposed on the architecture of the application, and it may be composed of one or more processes. The abstract SM unit is responsible for providing the ISA agent with information regarding the relative priorities of the streams, and whether each stream is active or passive. The QoSSess layer is composed of several independent agents that cooperate together to provide the QoSSess control framework. Two types of agents constitute the QoSSess layer: monitoring agents, and inter-stream adaptation (ISA) agents.

The monitoring agent is implemented as a library which is linked to the client (sender or receiver) code at compilation time. The client provides the agent with information

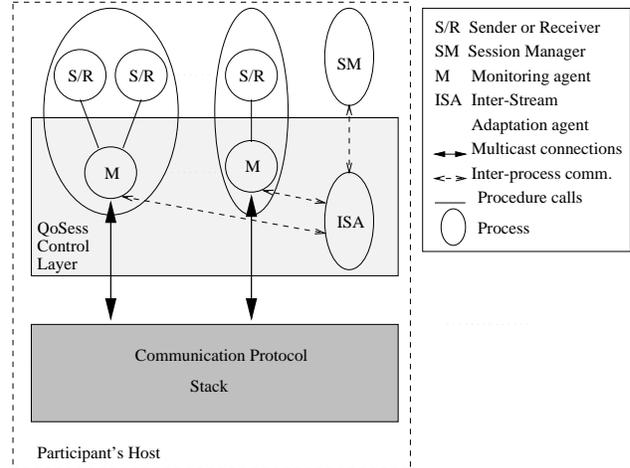


Figure 1. QoSess layer architecture

regarding the characteristics of the stream, through a **control interface**, at initialization time. The agent communicates this information to the local ISA agent, which uses it in taking rate adaptation decisions.

The client sends/receives the data stream through a **data interface**, which permits a **QoS measurement unit** to add/extract header information thus allowing for estimation of the perceived quality of the stream by checking monitored parameters, such as losses and jitter, against the client specified QoS requirements. Any change in the perceived quality of the stream is reported to the ISA agent, running on the local host.

On receiving an ISA decision regarding the operating point of the stream, the agent triggers a callback function to inform its client about the new operating point. In addition, these decisions are fed to a **state feedback protocol** machine, which engages the agents associated with the sender and receivers of a stream in a scalable and robust feedback protocol which enables the sender to set its transmission rate appropriately according to the requirements of the receivers and its ability. This feedback protocol is detailed in Section 3.

An ISA agent runs, as an independent process, on each host participating in the IMC session. The ISA agent is responsible for dynamically determining the operating points of the streams belonging to a session. These dynamic changes should be handled carefully in order to avoid instabilities and oscillations in operating points. The ISA agent must not react immediately to every notification received from monitoring agents to avoid over-reactions that may lead to instabilities. In the mean time, excessive delays in reaction time affect the responsiveness of the system and are not desired. The **ISA state machine** controls the state transitions of the ISA agent, in a way that ensures stability and responsiveness. This is discussed further in Section 5.

The **ISA decision making unit** implements the mechanisms necessary to select dynamically the operating point for each stream. The ISA decision making unit is triggered to recompute the operating points of the active streams by the ISA protocol state machine, when the latter detects either an overload or an under-load overall receiver state. Also, the decision making unit is triggered by external events such as the activation/deactivation of a stream or the change of the relative priorities of some of the streams. The inter-stream adaptation algorithms implemented by the decision making unit are discussed in Section 4.

### 3. A Scalable Feedback Mechanism

The objective of the feedback mechanism is to find out the worst case state among a group of receivers. The definition of the worst case state is dependent upon the context in which the feedback mechanism is applied. It can be the network congestion state as seen by the receivers, which may be useful for single-rate streams. Or, it can be the highest layer any receiver is expecting to receive from the source of a hierarchically encoded stream. This allows the sender to adjust its transmission rate in order not to waste resources on unneeded layers. This is particularly important in the context of managing multimedia streams in IMC sessions where the assumption that the sender has abundant resources is typically invalid.

In what follows, we assume that at every instant in time each receiver is in one state  $s$ , where  $s = 1, 2, \dots, H$ .  $H$  is the highest or worst case state, and the state of a receiver may change over time.

The randomly delayed replies approach augmented with receiver suppression of redundant replies was successfully deployed in IGMP (Internet Group Management Protocol) [4]. In our case, the group of receivers may be distributed over a wide area network (WAN), thus a reply sent by one receiver may not be heard by another before the other one emits its own reply which may be redundant. This implies the need for careful selection of the delay randomization function to avoid the reply implosion problem, while maintaining a low response time.

In the proposed feedback algorithm, the sender sends a probe message on a special multicast group which the sender and all the receivers join. The probe message contains a  $RTT$  field, which contains the smoothed average round-trip time from the sender to the group members. Upon receiving the probe, a receiver sets a timer to expire after a random delay period which is drawn from the interval:

$$\left[ C_1 f(s) \frac{RTT}{2}, (C_1 f(s) + C_2 g(s)) \frac{RTT}{2} \right],$$

where  $f(s)$  and  $g(s)$  are two non-increasing functions of

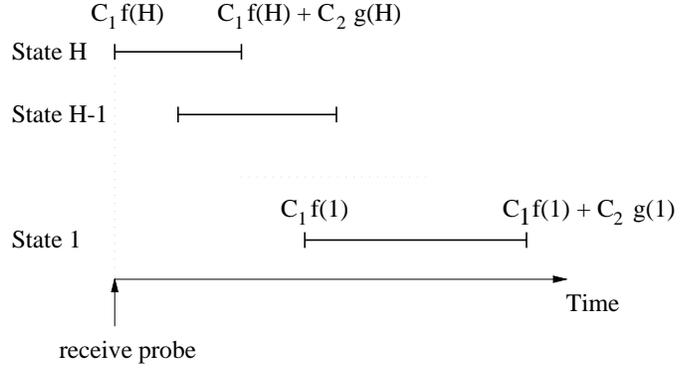


Figure 2. Distribution of timeout periods

the state  $s$ ,  $C_1$  and  $C_2$  are two parameters whose values are discussed later in detail. The receiver then keeps listening to the multicast group. If the timer expires, the receiver multicasts its state to the whole group. On the other hand, if the receiver receives another reply before its timer expires and that reply contains either the same or higher (worse) state, then the receiver suppresses its own reply.

The objective of setting the timeout periods as a function of  $f(s)$  and  $g(s)$  is to distribute the timeouts as in Figure 2. Receivers in higher states randomize their timeouts over periods that start earlier than receivers in lower states, thus allowing for higher state responses to suppress lower state responses. In addition, the lower state receivers randomize their timeouts over longer periods relative to higher state receivers. This is because as time elapses and no responses are generated, it can be implicitly deduced that the distribution of receivers over states is biased and more receivers belong to the lower states. Thus it is desired to randomize these condensed replies over longer periods.

In order to meet these objectives,  $f(s)$  and  $g(s)$  must be non-increasing functions of  $s$ . We chose to make  $f(s)$  and  $g(s)$  linear functions in  $s$  in order to avoid excessive delays in response time, where  $f(s) = H - s$ , and  $g(s) = f(s) + k = H - s + k$ .

$C_1$  controls the aggressiveness of the algorithm in eliminating replies from lower state receivers, while  $C_2$  controls the level of suppression of redundant replies from receivers in the same state. The values of these two parameters are explored in depth in the following sections. Given a certain value for  $k$ , the value of  $C_2$  can be tuned to optimize the performance of the mechanism. Thus, the value of  $k$  is set to one.

#### 3.1. Exploring the parameter space

Low values for  $C_1$  and  $C_2$  are desired in order to reduce the response time. On the other hand, excessive reduction in the value of either of the two parameters may lead to in-

efficiency in terms of the number of produced replies. In order to bound the values of  $C_1$  and  $C_2$ , we analyze an extreme network topology, namely: the star topology. Given a certain distribution of receiver distances from the sender, the feedback mechanism exhibits worst case performance when the receivers are connected in a star topology with the sender at its center. This is because connecting those receivers in a star topology maximizes the distance between any pair of receivers, hence, minimizing the likelihood of suppression of redundant replies. On the contrary, connecting those receivers in a chain topology minimizes the distance between any pair, hence, maximizing the likelihood of suppression of redundant replies.

In the star topology, the sender is connected to each receiver by a separate link. Any message sent from one receiver to another passes through the sender's node. Let all the receivers be at a distance  $r = \frac{RTT}{2}$  from the sender. Thus the distance between any two receivers is equal to  $2r$ .

Let  $G_s$  be the number of receivers in state  $s$ , and let  $T_s$  be the first timer to expire for receivers in state  $s$ . The expected value of  $T_s$  is  $(C_1 f(s) + \frac{C_2 g(s)}{G_s})r$ , since  $G_s$  timers are uniformly distributed over a period of length  $C_2 g(s)r$ .

For receivers in the same state, if the first timer expires at time  $t$ , then all the timers that are set to expire in the period from  $t$  to  $t+2r$  will not be suppressed, and all those that are set to expire after  $t+2r$  will be suppressed. Therefore, the expected number of timers to expire is equal to 1 plus the expected number of timers to expire in a period of length  $2r$ , which is equal to  $1 + \frac{2G_s}{C_2 g(s)}$ . Considering the case of  $s = H$ , since  $g(H) = 1$ , then setting  $C_2$  to any value less than 2 does not allow for suppression of any of the redundant replies from receivers in state  $H$ .

In order to suppress all replies from receivers in state  $s - 1$ , we must have:

$$\begin{aligned} T_s + 2r &\leq T_{s-1}, \\ \text{or } \frac{g(s)}{G_s} - \frac{g(s-1)}{G_{s-1}} &\leq \frac{C_1 - 2}{C_2}. \end{aligned}$$

For values of  $G_s$  and  $G_{s-1}$  which are relatively larger than  $g(s)$  and  $g(s - 1)$ , we find that  $C_1$  must be at least 2. In Section 6.1, we explore the performance of the feedback mechanism using simulation experiments.

## 4. Inter-Stream Bandwidth Adaptation

The ISA agent, at each participant's host, allocates the bandwidth available to the session streams based on their dynamically changing priorities. Typically, these priorities are identical for all the session participants (see e.g., [7]). However if for some application the receiver interests may vary, a feedback protocol, such as SCUBA [1], can be used for determining the average priorities of the streams.

In [10] and [12], we presented two inter-stream adaptation algorithms: RISA (Rate based Inter-Stream Adaptation), and I-WFS (Iterative Weighted Fair Share). The two algorithms assume the existence of a known amount of bandwidth reserved for the session, and that the transmission rate of each stream,  $i$ , can be set to any point in a continuous range of operating points in the interval  $[Rmin_i, Rmax_i]$ . RISA takes a greedy optimization approach in which the bandwidth is allocated to the streams with the objective of maximizing the overall gain to the session from this allocation. I-WFS, on the other hand, attempts to achieve fairness in bandwidth allocation among the streams of a session.

In a best-effort environment, it is desired to approximate the behavior of *I-WFS* and *RISA* under two additional constraints: the available bandwidth/capacity of a receiver is not known or fixed by a reservation protocol; and each source can change its transmission rate in discrete steps only.

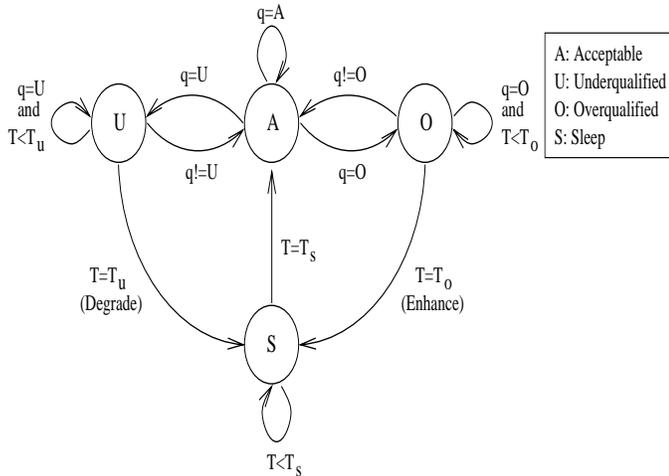
It can be easily shown that *RISA*, without any modifications, can support the above two constraints. However, devising an algorithm which approximates *I-WFS* under these constraints is more involved. In [11], we presented one such algorithm, namely: *A-IWFS*. The *A-IWFS* algorithm produces a linear order of layers from all sources in the session. Each receiver follows that linear order of layers. It cannot subscribe to a layer of higher order unless it has already subscribed to all lower order layers, and vice versa.

Another inter-stream bandwidth adaptation algorithm, which also requires knowledge about the session bandwidth, but accounts for the fact that sources can change their rates in discrete steps only, was presented, by Amir et al., in [1]. This algorithm maps the layers of the streams to artificial channels with fixed capacities. The channel packing effect is an obvious drawback in that approach, which may lead to inefficiencies in utilizing the available bandwidth. Another drawback to the concept of channels is that the receiver may have to join (or leave) multiple layers assigned to the same channel, simultaneously, in the adaptation process, which may introduce strong fluctuations that may lead to instability.

In Section 6.2, we compare the performance of these inter-stream adaptation algorithms.

## 5. Discussion of Rate Control and Stability

Figure 3 depicts the state machine that controls the operation of the ISA agent. The monitoring reports are assessed and aggregated into one input,  $q$ , which can take one of three values: *Acceptable*, *Underqualified*, or *Overqualified*. Transitions between states occur based on  $q$  and/or timeouts triggering the decision making unit to make add/drop decisions, as necessary. The hysteresis



**Figure 3. ISA state machine**

timeouts,  $T_u$  and  $T_o$ , as well as the sleep timeout  $T_s$ , play an important role in the stability and responsiveness of the agent. This is discussed below.

### 5.1. Multi-modal timers

Simultaneously satisfying the stability and responsiveness requirements of the protocol may lead to conflicting setups of the hysteresis timeouts. The key to satisfying these two objectives is to be able to accurately capture, at all times, the tendency of the system whether it is towards enhancement if the conditions are favorable, or towards degradation if the network is congested, or towards stabilizing at a certain subscription level that reflects the available resources. We refer to this tendency in the ISA agent as its *mode*. While this mode is reflected on the timers behavior dynamically, it does not imply the addition of any new states or transitions to the protocol state machine. The mode of the ISA agent is heuristically inferred based on its most recent actions. The agent can assume one of three modes at any instant in time:

**1. Enhance mode.** Being in this mode means that the overall conditions have been favorable, recently, and the tendency of the agent is towards adding more layers, thus, the add hysteresis timeout is relaxed (decreased). Also, any intermittent variations of the monitored QoS parameters are more likely to be transient, thus, the drop hysteresis timeout is backed off.

**2. Degrade mode.** This is the opposite of the previous mode.

**3. Probe mode.** In this mode, the agent is stabilizing around an operating point. The agent keeps probing periodically to check for the availability of more resources. Probing is done by adding a layer, and examining the effect of joining this layer on performance. If any deterioration in

the measured QoS parameters is noticed, the layer is immediately dropped. Here, it is important to back off the add hysteresis timeout aggressively over time in order to minimize the number of transient disturbances introduced by the probes.

### 5.2. Learning network delay

After adding or dropping a layer, the ISA agent must not take any further actions until it is sure that the impact of its previous action is fully established and can be sensed by itself, and hence the currently seen conditions are correct. Therefore, the sleep timeout must be a good indicator of the network reaction time. This is done by measuring the time that elapses from adding a layer until congestion is first detected by the agent, in a failed join experiment. The value of the sleep timeout is smoothly updated over time by these measurements.

### 5.3. Domain rate control protocol

The objective of this protocol is to help in maintaining the stability of the system while scaling to large groups of participants in a session. There is no doubt that the receiver oriented approach taken, where each participant decides for himself which layers of which streams to receive, is the key for scalability. However, the co-existence of several ISA agents in the same session opens further avenues for coordination and cooperation among those agents to enhance the stability of the system.

Our approach is to group the ISA agents of a session into domains. A domain is defined by the scope of the exchanged protocol control messages as determined by the time-to-live (TTL) field specified in those messages.

One way for enhancing the stability of the system by cooperation among ISA agents in the same local domain is to prevent unnecessary oscillations in the subscription levels of low rate receivers. In the case of network overload conditions, higher level subscribers are made to drop their upper layers first which may be sufficient to reduce congestion in the domain.

Another way for enhancing the stability of the system is by coordinating the join/leave of the highest layer in the domain. This minimizes the number of probes for rates above the current stable rate in the domain, and speeds up reaction to congestion. Also, letting other ISA agents in the domain learn about failed join experiments allows these agents to update their estimators for network reaction time without actually probing. The fact that the scope of the domain is limited is what allows for safely assuming that all agents in the domain are facing similar network conditions.

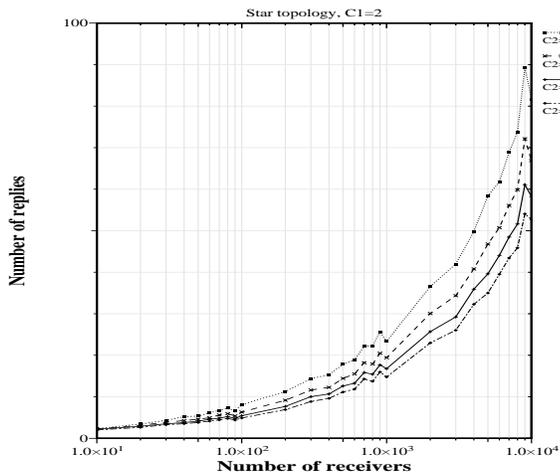


Figure 4. The effect of  $C_2$  on the number of replies

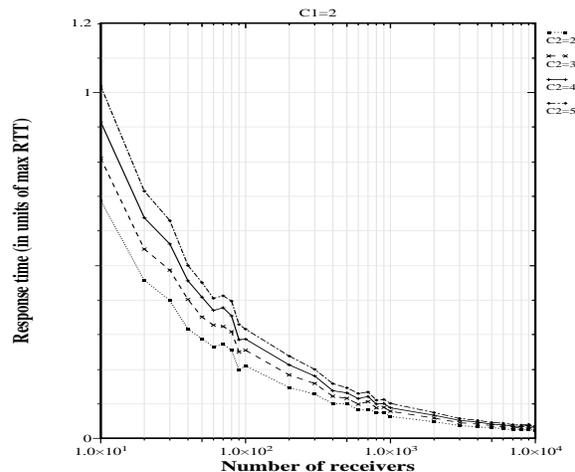


Figure 5. The effect of  $C_2$  on response time

## 6. Performance Study

### 6.1. Simulating the feedback mechanism

In this section, we examine the performance of the feedback mechanism, using simulation. We show the ability of the new mechanism to eliminate the reply implosion problem as we explore the effect of  $C_2$  on its performance. We ran several simulation experiments. In each experiment, the group size,  $G$ , and the maximum round-trip time,  $RTT_{max}$ , were selected. Round-trip times uniformly distributed in the interval  $[0, RTT_{max}]$  were assigned to all the receivers. The number of states,  $H$ , was set to 5, and each receiver was randomly assigned one of these states. The choice of 5 states (or layers) is reasonable as the state of the art hierarchical video encoders typically provide a number of layers in this range [8, 9]. Also, in applications where feedback information represents the perceived quality of service, typically 3 to 5 grades of quality are used [2, 3].

In Figure 4, the average number of replies is plotted for different values of  $C_2$ . The value of  $C_1$  was set to 2, according to the analysis of Section 3.1. It is clear from the figure that the reply implosion problem is totally eliminated. Moreover, over 95% of the redundant replies were correct replies (i.e., worst case state replies), which shows the robustness of the mechanism in facing network losses and its efficiency in eliminating non-worst case replies. This also means that, practically, the sender may safely react according to the first received reply. Figure 5 depicts the corresponding average response times. The response time is

measured at the sender, and represents the time from sending a probe until receiving the first correct reply.

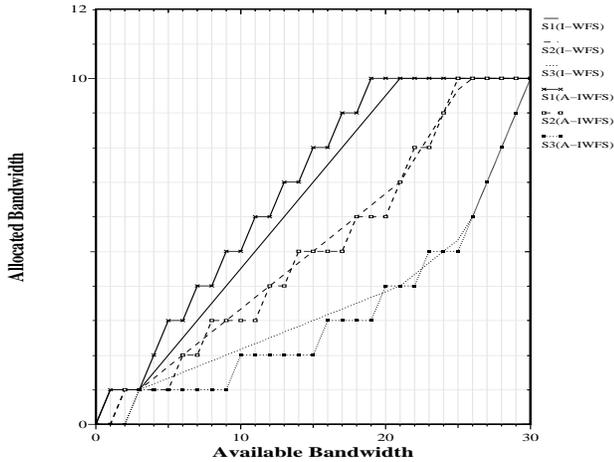
As can be seen from the figures, for  $C_2 = 4$  and a typical IMC session with up to 100 participants (e.g., IRI sessions [7]), less than 10% of the receivers reply to a probe, in the worst case, while for larger sessions of thousands of participants the reply ratio is below 1.5%.

### 6.2. Performance of the ISA mechanisms

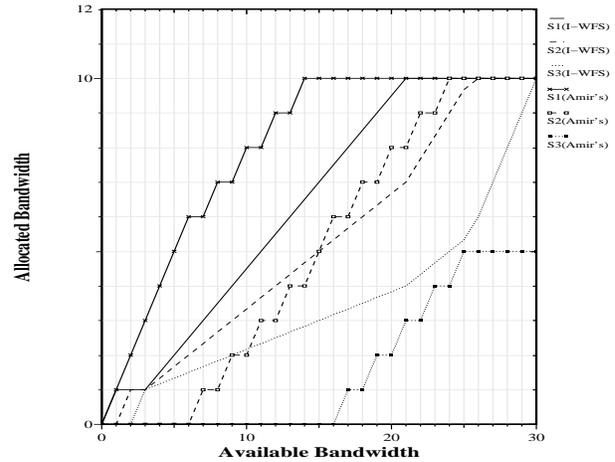
In this section, we compare our two algorithms, A-IWFS and RISA, to Amir's algorithm [1]. For Amir's algorithm, we used channels of unit bandwidth capacity in order to avoid penalizing the algorithm by the channel packing effect. The total number of channels was chosen such that all layers can be accommodated by the high-end receivers in the session. We simulated an IMC session composed of three streams S1, S2, and S3. Their priorities were set to 0.5, 0.333, and 0.167, respectively, with S1 being the most important. Each stream has 10 layers, each requiring 1 unit of bandwidth.

We compared the bandwidth allocation devised by A-IWFS to that devised by I-WFS. Figure 6 depicts the bandwidth shares of the 3 streams as allocated by A-IWFS and I-WFS. It is clear from the figure that A-IWFS tracks well the I-WFS allocation, in spite of its operation under more constraints.

Figure 7 depicts the bandwidth shares of the above 3 streams as obtained by Amir's algorithm, in contrast to the I-WFS case. As can be seen from the figure, Amir's allocation is far from that of I-WFS, i.e., the session bandwidth



**Figure 6. Comparing A-IWFS to I-WFS**



**Figure 7. Comparing Amir's algorithm to I-WFS**

is not shared fairly among the streams. The deviation from the I-WFS allocation exceeds 30% in some cases.

Figure 8 compares the allocation of Amir's algorithm to RISA. Although Amir's allocation is closer to RISA than to I-WFS (deviation does not exceed 20% before saturation), yet it generally has two major drawbacks relative to our two algorithms. First, for the high-end receivers, some of the streams may saturate in spite of the availability of bandwidth leading to under utilization of resources, as is the case with S3 in this experiment, which leaves over 16% of the available bandwidth non-utilized. Second, for the low-end or congested receivers, the number of active streams may be low and some streams may not be granted their initial base layer until after other streams are well enhanced, e.g., in this experiment, a receiver with 6 units of available bandwidth will not receive any layers from S2 or S3, and all the available bandwidth will be dedicated to S1.

### 6.3. Experimental results

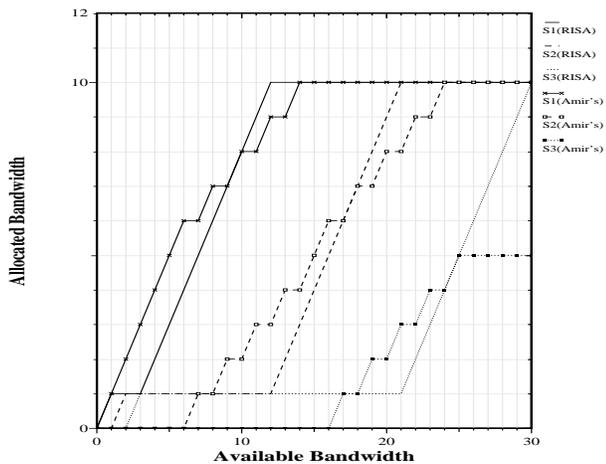
Figure 9 illustrates the result of deploying multi-modal timers and learning network delay by an experiment. In this setup, a distance learning session is composed of three video streams: a higher priority stream for the teacher video (TV), and two lower priority student video streams (SV1 and SV2). Each of the streams is hierarchically encoded by an encoder which produces three layers of constant bit rates of 45, 180, and 525 Kbps for 15 frames per sec video [9]. The figure plots, over time, the cumulative rate received by a receiver sitting behind a 1.5 Mbps bottleneck link with 15 KB (15 packets) router buffer size, which is well above double the bandwidth-delay product of the link. As shown

in the figure, the ISA agent switches to the *Enhance* mode soon after the session starts and quickly reaches the stable subscription level. Then the agent switches to the *Probe* mode where the time between two consecutive probes gets backed off over time. After 180 seconds from the beginning of the session, the bottleneck link is subjected to cross traffic load of 1.2 Mbps for 1 minute. As soon as the congestion is detected, the agent switches to the *Degrade* mode where drop hysteresis is reduced and thus it quickly drops layers and stabilizes at a low rate, where it switches to the *Probe* mode and then to the *Enhance* and finally *Probe* mode again. Similar results were obtained for emulated higher link delays, up to 10 seconds.

## 7. Conclusion

In this paper, we highlighted the main components of a complete framework for controlling the quality of collaborative multimedia sessions. The architecture of a QoS layer, that realizes the framework, was presented. The layer is composed of two types of agents: monitoring agents and inter-stream adaptation (ISA) agents. An overview of the functions of each of the agents was given.

Providing the source of a stream with feedback information about the used layers of the stream is crucial to the success of the QoS control layer in efficient utilization of the available resources. A scalable and robust feedback mechanism was devised for this purpose. It allows the sender to always send only layers for which interested receivers exist, and to suppress unused layers. Simulation results showed that the proposed feedback mechanism scales



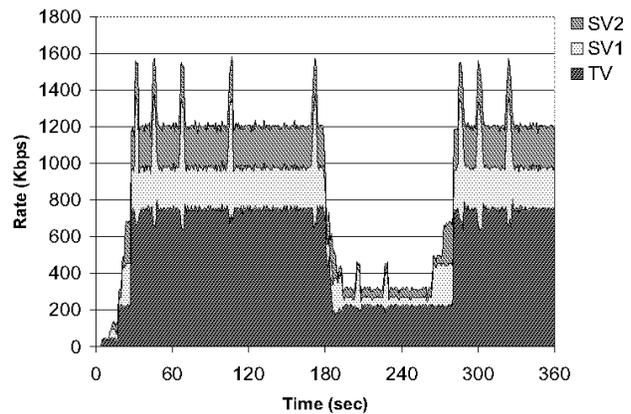
**Figure 8. Comparing Amir's algorithm to RISA**

well for groups of up to thousands of participants, and is robust in facing network losses. Currently, we are investigating the performance of the feedback mechanism under different distributions for round-trip times and receivers' states. Also, several adaptive enhancements to the delay-randomizing function are being investigated.

The quality of the session is controlled by means of inter-stream adaptation mechanisms, that base their decisions on information stemming from the application semantics and reflecting the instantaneous relative importance of the different streams to the session. We compared the performance of several inter-stream bandwidth adaptation algorithms, and showed the advantages of the A-IWFS algorithm over the others. The algorithm was shown to be more efficient and fair in utilizing the available bandwidth and in maximizing the number of admitted streams than other known inter-stream adaptation algorithms. Additionally, stability and preventing oscillations in layer subscription was illustrated experimentally.

## References

- [1] E. Amir, S. McCane, and R. Katz, "Receiver-driven Bandwidth Adaptation for Light-Weight Sessions", *ACM Multimedia*, Seattle, WA, November 1997.
- [2] J. Bolot, T. Turetli, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet", *ACM SIGCOMM*, 24(4):58-67, October 1994.
- [3] S. Cheung, M. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution", *Proc. of IEEE INFOCOM'96*, pp. 553-560, San Francisco, CA, March 1996.



**Figure 9. Bottleneck link throughput**

- [4] S. Deering, and D. Cheriton, "Multicast Routing in Internetworks and Extended LANs", *ACM Trans. on Computer Systems*, 8(2):85-110, May 1990.
- [5] A. Gupta, W. Howe, M. Moran, and Q. Nguyen, "Resource Sharing for Multi-Party Real-Time Communication", *Proc. of IEEE INFOCOM'95*.
- [6] X. Li, S. Paul, M. Ammar, "Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control", *Proc. of IEEE INFOCOM 98*, San Francisco, CA.
- [7] K. Maly, H. Abdel-Wahab, C.M. Overstreet, C. Wild, A. Gupta, A. Youssef, E. Stoica, and E. Al-Shaer, "Interactive Distance Learning over Intranets", *IEEE Internet Computing*, 1(1):60-71, January 1997.
- [8] S. McCane, and V. Jacobson, "Receiver-driven Layered Multicast", *ACM SIGCOMM*, Stanford, CA, August 1996.
- [9] S. Senbel and H. Abdel-Wahab, "A Quadtree-based Image Encoding Scheme for Real-Time Communication", *Proc. of the IEEE Intern'l Conf. on Multimedia Computing and Systems*, pp. 143-150, Ottawa, Canada, June 1997.
- [10] A. Youssef, H. Abdel-Wahab, and K. Maly, "Performance Evaluation of an Inter-Stream Adaptation Algorithm for Multimedia Communications", *Proc. of IFIP Intern'l Conf. on Performance of Information and Communication Systems (PICS'98)*, pp. 333-344, Lund, Sweden, May 1998.
- [11] A. Youssef, H. Abdel-Wahab, and K. Maly, "The Software Architecture of a Distributed Quality of Session Control Layer", *Proc. of the Seventh IEEE Intern'l Symposium on High Performance Distributed Computing (HPDC-7)*, Chicago, IL, July 1998.
- [12] A. Youssef, H. Abdel-Wahab, K. Maly, and C. Wild, "A Comparative Study of Two Generic Resource Allocation Models", *Proc. of the Third Intern'l Conf. on Computer Science & Informatics*, Durham, NC, March 1997.
- [13] L. Zhang, et al., "RSVP: A New Resource ReSerVation Protocol", *IEEE Network Magazine*, September 1993.