

# The Adaptive Load Service (ALS): An ABR-Like Service for the Internet\*

Dorgham Sisalem  
GMD-Fokus, Berlin  
sisalem@fokus.gmd.de

Henning Schulzrinne  
Columbia University, New York  
schulzrinne@cs.columbia.edu

## Abstract

In this paper, we present a novel approach for realizing adaptive QoS control in the Internet called the adaptive load service (ALS). ALS was designed in a similar fashion to the available bit rate service (ABR) proposed for ATM. That is, the senders transmit control messages indicating their requirements, intermediate routers adjust the indicated values in accordance with their available resources and the receivers send the contents of the control messages back to the senders. The transmission behavior of the senders is then adjusted in accordance with the indicated values in the returned control messages. Unlike ABR, ALS is IP-based and was designed to scale for large multicast groups and accommodate the needs of heterogeneous receivers through the integration of sender-based and receiver-based adaptation mechanisms in addition to relying on the network feedback for adjusting the transmission behavior.

Performance tests of ALS under different simulation topologies show its efficiency in terms of bandwidth utilization and loss reduction. The tests also show ALS to be fair towards competing TCP connections. Additionally, even though ALS uses a simple algorithm for determining the fair bandwidth share a connection should be using, ALS manages to fairly distribute available resources in accordance with the max-min fairness criterion.

## 1 Introduction

In addition to providing Email and web services, the Internet is increasingly being used to support multimedia services such as media-on-demand and Internet telephony. However, these multimedia services are usually based on the UDP protocol which supports no means of QoS or congestion control. Sending non-congestion controlled traffic in the Internet without regard to the actual network capacity might easily lead to high losses and consequently to the starvation of competing TCP traffic which still constitutes more than 95% [30] of the Internet traffic and reacts to losses by reducing its transmission rate.

In this paper, we present a novel approach for improving the quality of service of multimedia communication in the Internet and reducing congestion and losses in the network. The scheme

---

\*This work was funded in part by the BMBF (German Ministry of Education and Research) and the DFN (German Research Network).

called the adaptive load service (ALS) is basically similar to the available bit rate (ABR) service proposed for ATM [23]. With ALS, the sender transmits control messages including its desired transmission rate to use. The intermediate routers adjust this value in accordance with their available resources and forward the control messages to the next network node until they reach the receiver. The receiver in its turn transmits the updated information back to the sender who needs to adjust its transmission behavior in accordance with the received information.

While ALS is rather similar to the ABR service, it is much simpler in its specification and is based on the IP protocol. ALS utilizes the RTP/RTCP [22] protocol for carrying the control information. This avoids the need for introducing a new protocol and allows ALS to benefit from the scalable design of RTP/RTCP and thus be used in large multicast groups. Additionally, ALS was designed to accommodate the needs of heterogeneous receivers by supporting the notion of layered data transmission. That is, a data stream can be divided into different layers and sent to different receiver groups. In this case, ALS provides the receivers with exact information about the layers to join in order to receive the QoS level that corresponds with their capacities. ABR, on the other hand, was mainly designed for point-to-point communication. While there have been some proposals for supporting multicast communication [9] they were not scalable for large groups and could not accommodate the needs of heterogeneous receivers. Another major difference between ALS and ABR, is that while ABR uses rather complicated algorithms for determining the fair bandwidth shares competing connections should be using, ALS is based on a simple algorithm that only calculates the minimal fair share and allows the end systems to probe the network for a higher share.

In Sec. 2, we will look at some related work to the subject of QoS control in the Internet and discuss some of the advantages and disadvantages of some of the proposed approaches for QoS control and congestion avoidance. ALS is then described in Sec. 3. In Sec. 4, we will investigate the performance of ALS and its behavior under different network topologies. A look at some of the yet unsolved problems and future work will be given in Sec. 5.

## **2 Motivation and Related Work**

A large part of the multimedia conferencing applications used currently in the Internet, such as VIC [19] and VAT [18] are based on the UDP transport protocol. However, UDP offers no quality of service control mechanisms and can therefore not guarantee any level of QoS. Fluctuations of network conditions combined with the inability of those protocols to support QoS

control often render multimedia applications useless.

In addition, deploying non-congestion-controlled UDP traffic results in extreme unfairness towards competing TCP traffic. Sending best-effort traffic without any consideration of the network congestion state can easily result in packet losses. In response to that, TCP connections sharing the same bottleneck would reduce their transmission rates. However, without any rate reduction on behalf of the non-congestion-controlled traffic, the TCP connections which constitute around 95% of the Internet traffic today [30] would starve and receive much less than their fair bandwidth shares. Therefore, UDP sources need to be enhanced with quality-of-service control mechanism that not only aims to reduce loss ratios and improves bandwidth utilization but also is fair towards competing TCP connections, i.e. be *TCP-friendly*.

Currently, different approaches are being discussed for solving the QoS and congestion control problems such as: resource reservation [6], priority mechanisms [4] and application control [8], i.e., to instruct the applications at the end systems to adapt the bandwidth share they are utilizing to the network congestion state.

Reserving enough resources for supporting a certain QoS level in advance guarantees this quality level and would be the most straightforward approach for handling the problem. However, as it is usually rather difficult to determine the exact characteristics of a stream in advance, one would tend to over-allocate resources to guarantee the requested QoS level, leading to network underutilization. In addition to the complexity and scalability problems of reservation based QoS control schemes, these schemes usually do not easily allow the use of extra bandwidth for improved quality during network underload states. In order to benefit from idle capacities or change the amount of the reserved resources during the life time of a connections renegotiation protocols are needed [13].

With priority mechanisms, individual packets or flows are labeled with different priorities and are hence treated differently at the network routers. This approach is simpler than the reservation approach as it requires no signaling and less complicated control mechanisms at the routers. However, the exact mechanisms for setting the priority levels, the router mechanisms for realizing the desired QoS levels and the actual gain achieved with such an approach are still under discussion [4].

While reservation based schemes guarantee the required QoS level, mechanisms using application control aim at improving the quality by reducing losses on the one side and increasing utilization on the other and usually offer no guarantees. However, it is usually the case that

due to the loss of content or the need to use a considerable overhead for forward error correction a low bandwidth video stream, for example, with no or only low losses can have a higher perceived quality than a high bandwidth yet lossy stream [31]. With application control based mechanisms the sender adapts its transmission rate in accordance with the network congestion state. That is, based on feedback information from the receivers, the sender increases its transmission rate during underload situations and reduces it otherwise. This is especially beneficial as the bandwidth availability may change during a session, particularly during long-lived sessions typical for multimedia communications.

Ergonomic studies and the experience gained from the Internet demonstrate that people can use audio and video data as long as the information content is above a minimum level [33]. This level depends on media content and the task at hand. For instance, a foreign language is more difficult to understand than a native language when audio quality is reduced. So, at the expense of slight degradation in user satisfaction, it is possible to adjust the transmission rates of end systems in accordance with the network congestion state. Several proposed mechanisms in the literature, see [5, 7, 27] already show that deploying application control results in a better overall bandwidth utilization, as it avoids network congestion.

Application control works best for live transmission for unicast and small multicast groups; in large multicast groups in a heterogeneous environment, a “race to the bottom” can occur so that one poorly connected receiver determines the quality of a possibly much larger number of well-connected receivers. To avoid these problems, various proposals have been made for receiver based adaptation schemes using layered data distribution [32, 20, 34]. Those proposals are based on partitioning a data stream into a base layer, comprising the information needed to achieve the lowest quality representation and a number of enhancement layers. The different layers are then sent to different multicast sessions and the receivers determine how many sessions to join and thereby adjust their QoS in respect to their own requirements and capacities.

While layered data transmission solves the heterogeneity problems, it might cause additional delays at the receivers. As the different layers might use different paths and hence have different round trip times, the receivers need to resynchronize the arriving data. Additionally, data partitioning might lead to a drift problem caused by motion compensated coding if only the lower bit rate layer is received [14]. Finally, as the receivers need to find out how many layers to join, they have to *probe* the network capacity by joining a higher layer for a while and deciding to keep receiving this layer or to leave it based on the observed losses. Note that join-

ing a new multicast session without a priori knowledge of the available network resources and the size of the joined layer would in the case of a highly utilized network part most likely lead to a congestion situation. This does not only lead to increased losses at receivers of connections traversing the same network part but might also distort the view of a member of the same session trying to join a lower layer. That is, the receiver cannot determine for sure if the losses noticed after joining a higher layer are because joining this layer causes congestion or simply a result of the congestion caused by another receiver trying to join an even higher layer. To avoid this situation, some synchronization between all receivers of the same session is needed to regulate when the probing procedure might take place. Note also, that for the case of members of two or more different groups trying at the same time to join a higher layer provided for their group all join attempts might fail even though there might have been enough capacity for the members of one session to successfully join a higher layer. To avoid this situation some global synchronization between the competing sessions is required which additionally increases the complexity and reduces the scalability of this approach.

An additional disadvantage of the proposed layered transmission mechanisms is the static setting of the sizes of the different layers. Currently, data is divided among different layers based on encoding characteristics or possible network topologies. For example to support users connected through ISDN, T1 and ATM links to a multicast session, a sender might divide its data stream into three layers with the base layer having a data rate of 64 kb/s, a first enhancement layer with a few hundreds kb/s and the rest of the data to be sent is then placed on the second enhancement layer. However, with no a priori knowledge of the traversed links, such a static distribution might be disadvantageous as it does not consider the actual resources of the receivers. For example, consider the distribution presented above for accommodating receivers connected over ISDN, T1 and ATM links. For the case that no members are connected over ISDN links, and some members have enough resources to receive a data stream larger than the base layer but smaller than the first enhancement layer, they would still be restricted to the lower layer. In [2] the authors describe a mechanism called SCUBA that enhances layered data transmission schemes by assignment of signal layers to network channels based on user preferences computed by the SCUBA scheme. Given a user preference, each source determines its signal layer mapping. While this approach accommodates receivers preferences in accordance with the sent data, the QoS of the participating receivers is still only changed with the granularity of a layer and the scheme provides no assistance in dynamically determining the appropriate

transmission rate of each layer.

The adaptive load service integrates both sender based application control with the receiver based schemes. Based on the information collected at the receivers and in the network the sender can adjust its transmission behavior in accordance with the available capacity. To account for heterogeneous receivers, the sender transmits its data in separate layers. By providing the receivers with information about the layers they can join, the receivers need not to probe the network and the scalability and complexity problems of the receiver based approaches are, thus, considerably reduced. Additionally, based on the feedback information ALS senders adjust the number and size of layers in accordance with the actual heterogeneity of the network.

### 3 Specification of the Adaptive Load Service (ALS)

When designing an adaptation mechanism for QoS control we usually need to consider the following goals:

- achieve an overall high resource utilization,
- reduce losses,
- maintain a stable quality-of-service,
- distribute the available resources in a fair way,
- scale to large and heterogeneous multicast groups.

Note, that while the issue of fair resource distribution is of utmost importance it is a subjective measurement. In this paper, we will consider a system to be fair if similar connections receive similar shares. Additionally, for the case that different connections have the same characteristics and requirements but differ in the number of traversed network hops, our goal will be to achieve the max-min fairness criterion often used for fairness measurement of ABR mechanisms and described in the ATM Forum traffic management specification 4.0 [23]. By this definition an intermediate router sets a connection's fair bandwidth share minimally to

$$\frac{\text{Bandwidth of link}}{\text{Number of connections}}$$

and maximally to

$$\frac{\text{Bandwidth of link} - \sum \text{Bandwidth of connections bottlenecked elsewhere}}{\text{Number of connections} - \sum \text{Number of connections bottlenecked elsewhere}}$$

A bottlenecked connection indicates that the fair share calculated for this connection at some other link is smaller than its minimum fair share at this link. A connection's fair bandwidth share is then set to the minimum fair share calculated at all traversed hops.

To achieve high utilization, low losses and fair distribution the adaptive load service was designed in a similar fashion to the ABR service in ATM. That is, the senders transmit control packets containing information about their desired amount of resources and the network nodes calculate the bandwidth shares the connections should be using and include them in the control messages. The receivers, finally, send the adjusted information back to the senders which need to adjust their transmission behavior based on the received control information. To avoid using a new protocol, ALS is based on the Real Time Transport Protocol (RTP) [22] designed within the Internet Engineering Task Force (IETF) and widely used for multimedia communication in the Internet. RTP sessions consist of two lower-layer data streams, namely a data stream for, say, audio or video and a stream of control packets (using the sub-protocol called RTCP). Each session member periodically sends RTCP control reports to all other session members. The RTCP messages consist of a source specification part for identification of the sender, receiver reports containing loss and jitter information for each incoming data stream and a sender report about the sent data, if any. Additionally, the end systems might include an application specific part (APP) intended for experimental use. The desire for up-to-date control information has to be balanced against the desire to limit control traffic to a small percentage of data traffic even with sessions consisting of several hundreds of members. Therefore, the control traffic is scaled with the data traffic so that it makes up a certain percentage of the data rate (usually 5%) with the sender reports making up at least a 25% of the bandwidth dedicated for the control packets. To stay within this limit the interval between the sending of two consecutive RTCP messages is adjusted as a function of the number of participating members, with a minimum interval of 5 seconds.

### **3.1 Router Behavior**

The senders transmit control packets containing the sender reports with two additional fields indicating the desired rate ( $R_d$ ), i. e., the rate the sender wishes to use, and a network utilization field ( $U$ ). Arriving at the intermediate network nodes the values of the desired rate ( $R_d$ ) of the connections are compared to a locally calculated fair bandwidth share this connection should be using ( $R_f$ ). For the case of  $R_d \geq R_f$ ,  $R_d$  is set to  $R_f$  and the control packet is forwarded with

the new value to the next hop. Additionally, to provide the end systems with a more complete view of the network congestion state the router includes its utilization percentage in the  $U$  field for the case of  $R_d \geq R_f$ .

A large part of the work done in the context the ABR service was dedicated to the issue of estimating the appropriate fair bandwidth share to advertise at the switches. This is basically done using the contents of the control cells sent by the senders, which indicate the desired rate the sender would like to use and the current rate the sender is using. Based on these information, the switches can estimate the number of connections congested at another network node and the available bandwidth share to distribute among the competing ABR connections. Even with efficient algorithms such as [16, 15, 3] various state information have to be saved and several multiplication operations and data lookups are needed. Depending on the complexity of the scheme and the amount of saved state information about the connections, several estimation cycles are needed to calculate an appropriate and stable fair bandwidth share. For the case of ABR, each estimation cycle consists of the round trip delay and the time between sending two control cells which is usually set to the time needed for sending 32 ATM cells -for an ATM transmission rate of 155 Mb/s this would be less than 100  $\mu$ s. For the case of RTP, such an estimation cycle would last around 7.5 seconds with each of the sender and receiver sending an RTCP message on the average of 5 seconds. So with around 10 estimation cycles required for determining the fair share, which is an appropriate value for a moderate complexity algorithm [24], ALS would need more than a minute until determining the fair share. Additionally, as the available bandwidth and number of connections will most properly change during the time needed for determining the fair share, the estimation time will even increase further. Hence, using a distribution algorithm similar to those proposed for ABR would not be of much benefit and would only lead to more complex routers. Therefore, ALS routers determine the fair rate share ( $R_f$ ) as follows:

$$R_f = \frac{b \times U_l}{n_{\text{conn}}}$$

with the  $b$  indicating the bandwidth dedicated for the ALS connections and  $n_{\text{conn}}$  as the number of connections seen in the last observation interval. Throughout this paper, we set the observation interval arbitrarily to one second. A connection is recognized through the IP-header with the source and destination addresses.  $U_l$  represents the utilization level we would like to achieve. Note, that setting  $U_l$  too high would lead to high utilization but also to high losses for the case of sudden changes in the number of connections or available bandwidth. In this paper, we use a

utilization level ( $U_l$ ) of 90% which is equivalent to the value used in similar switch mechanisms for ABR [3].

### 3.2 Receiver Behavior

After traversing all the intermediate routers the RTCP messages contain in the  $R_d$  field the smallest possible fair share calculated at all traversed network nodes and the utilization of the router where this fair share was determined. The receiver includes these values in its control messages and sends the information back to the sender. In addition to the fair share and utilization values, the receiver reports the usual RTCP statistics such as delay information and the losses in the sender's data stream.

### 3.3 Source Behavior

As already mentioned, the source needs to inform the network about its desired transmission rate in an additional  $R_d$  field in the RTCP messages. Additionally, the sender includes a network utilization field ( $U$ ) that is initialized to 0. In case the desired rate is not known,  $R_d$  can be set to the maximum possible rate, i.e., the rate of the output link of the system.

After receiving the feedback information about the fair rate share and network utilization in the receiver reports the sender needs to adjust its transmission rate in accordance with the received values.

Note, however, that the calculation of the  $R_f$  value at the routers does not correspond to the max-min fairness criterion described in Sec. 2. As an example, consider the topology depicted in Fig. 1. The first router has a capacity of 2 Mb/s and is shared between 2 connections, hence the router would calculate  $R_f$  to be 1 Mb/s. However, the second router traversed by the connection from sender 1 to receiver 1 has only a capacity of 1 Mb/s and is also shared between two connections, and thus the fair share value that will finally be advertised to sender 1 is only 0.5 Mb/s. Now, if sender 2 restricted its transmission rate to the fair share value included in the control messages, i.e., 1 Mb/s, 0.5 Mb/s of the bandwidth available at router 1 will be wasted. Using the max-min fairness criterion the fair share of sender 2 should actually be 1.5 Mb/s.

To realize the max-min fairness criterion and still use the simple router algorithm described in Sec. 3.1, we enhance the source's behavior to estimate the appropriate fair share by itself in addition to the network feedback.

An ALS sender does not set its transmission rate exactly to the values suggested in the re-

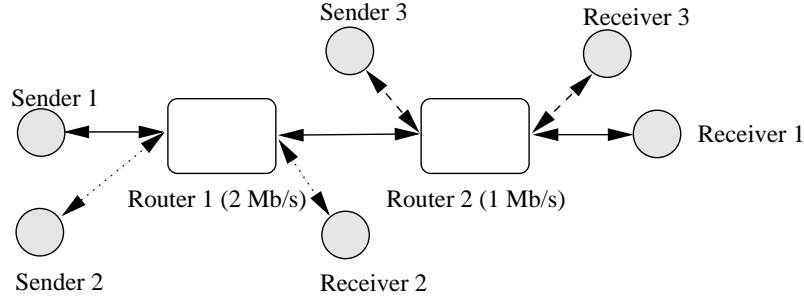


Figure 1: An example for max-min distribution

ceived control messages but actually uses an adaptation approach similar to a scheme we presented earlier [27]. For each incoming RTCP message from receiver  $i$  the sender calculates a transmission rate  $r_i$  based on the  $R_{d,i}$ ,  $U_i$  and losses indicated in receiver  $i$  reports and saves this rate in a data base. Here,  $r_i$  indicates the transmission rate the sender should be using if receiver  $i$  was the only receiving member and  $R_{d,i}$  is the  $R_d$  value included in the control messages of receiver  $i$ . For the case of multicast sessions with lots of receivers, ALS does not react to each received RTCP packet as some adaptation schemes suggest [7]. Instead, ALS is based on so called adaptation points, i.e., the adaptation process takes place only at specific time points. At each adaptation point, the sender searches the transmission rates data base for the minimum value ( $r_{\min} = \min(r_i)$ ) and sets the sender's transmission rate ( $r$ ) to  $r_{\min}$ .

$r_i$  is calculated as follows:

- If the control messages received from receiver  $i$  indicate no losses the sender can try to increase its bandwidth share. This is based on the assumption that the reported  $R_{d,i}$  value is less than the fair bandwidth share that would have been calculated with an algorithm using the max-min fairness criterion. Similar to other adaptation schemes [7] the rate is increased additively with an additive increase rate (AIR). To be able to adjust AIR in accordance with the available network resources and network congestion state we do not set it to a static value but calculate it dynamically based on the current utilization of the network.

$$\text{AIR}_i = \text{AIR}_i^* + (\text{AIR}_i^* \times (1 - U_i))$$

with  $U_i$  as the utilization value indicated in receiver  $i$  reports and  $\text{AIR}_i^*$  as the increase rate calculated after receiving the last control message. The transmission rate ( $r_i$ ) is then set to

$$r_i = R_{d,i} + \text{AIR}_i$$

- For the case that the report from receiver  $i$  indicates losses the sender might assume that the  $R_{d,i}$  value indicated in the report is outdated due to a change in the network conditions after setting the  $R_{d,i}$  value in the senders RTCP messages. In this case, the sender should reduce its transmission rate ( $r_i$ )

$$r_i = R_{d,i} \times (1 - l_i)$$

where  $l_i$  is the loss indication from receiver  $i$  indicated as a fraction of 1. Additionally,  $AIR_i$  is set to an initial small value that we arbitrarily chose to be 5 kb/s.

We have used a period of 5 seconds between two adaptation points which is also the average value between generating two RTCP packets at the same source. As the time between two RTCP packets might actually be larger than 5 seconds, choosing this fixed value has the disadvantage that the adaptation decision might be taken based on the reports of only a subset of the session members and not all of them. However, the other alternative would be to wait for the reports from all members before adapting. While this might work for small groups, for larger groups the interval between two sent RTCP packets increases and thereby the adaptation decision will be taken less frequently and, thus, be less effective. The advantages and disadvantages of both alternatives still need to be investigated more closely.

### 3.4 Supporting Heterogeneous Receivers

As already described in Sec. 2, reacting to the feedback of receivers of varying capacities results in a race to the bottom situation and adjusting the transmission rate of the sender down to the worst receiver level. To avoid this situation, ALS supports layered transmission of data. That is, the data stream is divided into different layers which are then sent to separate multicast groups. To receive layer  $L_n$  the receiver must first join all the layers from  $L_0$  up to  $L_{n-1}$ . The need to probe the number of appropriate layers to join might lead to network congestion and is the major source of complexity with layered transmission schemes. To avoid this need for probing we enhanced ALS as follows:

- At each adaptation point the sender determines the smallest fair share indication seen in the last adaptation interval ( $R_{d,\min} = \min(R_{d,i})$ ) and the largest one ( $R_{d,\max} = \max(R_{d,i})$ ) in addition to the minimum calculated transmission rate ( $r_{\min} = \min(r_i)$ ) and the maximum transmission rate ( $r_{\max} = \max(r_i)$ ). Depending on the compression algorithm used, number of receivers and variations in the capabilities of the receivers the sender can deter-

mine the appropriate number of layers to use and the amount of data to send on each layer. In this paper, we arbitrarily set the number of layers to three, and set the transmission rate on the first layer to the minimum calculated rate ( $r_{\min}$ ). On each of the upper layers the sender transmits data with a rate of ( $\frac{r_{\max}-r_{\min}}{2}$ ). Note, that this distribution was chosen arbitrarily and for actual implementations another mechanism might be used. The sender includes in the control packets sent on the basic layer, which is received by all members of the session, additional information indicating the number of layers used, the addresses on which the different layers are sent on and an indication of the size of each layer.

- The receivers start by joining the basic layer. After receiving a control message with information about the number of available layers and their sizes the receiver compares the fair rate share ( $R_d$ ) indicated in the received control message and the size of the different layers. With  $R_d$  larger than the transmission rate of the basic layer the receiver can join up to  $n$  layers while:

$$R_d \geq \sum_{i=0}^n R_{L,i}$$

with  $R_{L,i}$  as the rate used for layer  $i$ . Note that the receiver makes the decision about joining or leaving a layer based on the  $R_d$  value set in the network. Therefore, the sender needs to indicate the sizes of the different layers in the control messages as  $R_{d,\min}$  for the basic layer and  $\frac{R_{d,\max}-R_{d,\min}}{2}$  for the upper two layers. This stems from adaptation procedure at the sender that leads to the actual transmission rates being higher than the values advertised by the network. Using the actual transmission rates as size indication in the sender reports might actually lead to network underutilization problems as the following example shows: consider a sender multicasts data to two receivers and gets the values  $R_{d,1}$  and  $R_{d,2}$  as the fair bandwidth shares to use on the paths leading to receiver 1 and 2, correspondingly. With the adaptation procedure, the sender might send data in two layers with rates  $R_{L,1}$  and  $R_{L,2}$  with ( $R_{L,1} \geq R_{d,1}$ ) and ( $R_{L,1} + R_{L,2} \geq R_{d,2}$ ). If the sender included in its reports  $R_{L,1}$  and  $R_{L,2}$  as size indication, receiver 2 would only be able to join layer 1 as ( $R_{d,2} < R_{L,1} + R_{L,2}$ ).

### 3.5 Implementation Issues

At the sender side, the additional fields for the fair share indication and utilization as well as the information about the number and sizes of the used layers are sent as an application specific

part (APP) [22] of the RTCP control messages. Likewise, the receiver returns the fair share and utilization values as an additional application specific part of its RTCP messages. This approach ensures the compatibility of senders using ALS with RTP-capable receivers with no extensions for ALS. If the receivers or the network do not support ALS the APP parts can be ignored at the receivers and the sender can refer to sender based adaptation schemes such as [27, 7].

To alert transit routers to more closely examine the contents of the control messages, the senders' RTCP messages are carried in IP packets with the IP router alert option [17]. This is the same approach as was proposed for RSVP or YESSIR [21].

After setting the appropriate  $R_d$  and  $U$  values in the RTCP messages the router can forward the packet to the next hop. To avoid the need to recalculate the UDP checksum the sender should set it to 0.

## 4 Performance of the Adaptive Load Service

In this section, we investigate the performance of the ALS scheme under different simulated topologies. In particular, we investigate the performance of ALS in terms of achieved utilization, avoiding losses and its fairness towards other ALS connections as well as competing TCP traffic. Additionally, we investigate the behavior of ALS in heterogeneous environments and for the case of multiple congested hops.

### 4.1 Competing TCP and ALS Traffic

For any adaptation scheme to be effective in the Internet environment it needs not only to ensure high utilization, avoid losses but also be fair towards competing TCP traffic. In this section, we investigate the performance of ALS for the case of several ALS connections sharing a bottleneck router with some TCP connections. Fig. 2 shows a simple network topology with  $n$  TCP connections sharing a link of  $b$  kb/s with  $m$  ALS connections. The TCP connections are based on the Reno TCP specifications with fast retransmission and fast recovery algorithms [28]. Both the ALS and TCP sources are persistent sources. That is, they always have data to send at the highest allowed transmission rate. The router used in the simulations as well in the simulations in the next sections uses random early detection (RED) [11] for buffer management. A RED router detects incipient congestion by computing the average queue size. When the average queue size exceeds a preset minimum threshold the router drops each incoming packet with some probability. Exceeding a second maximum threshold leads to dropping all arriving

packets. This approach not only keeps the average queue length low but ensures fairness and avoids synchronization effects. Based on results achieved in [12] the minimum drop threshold was set to half of the router’s buffer and the maximum one to 0.95 of the router’s buffer.

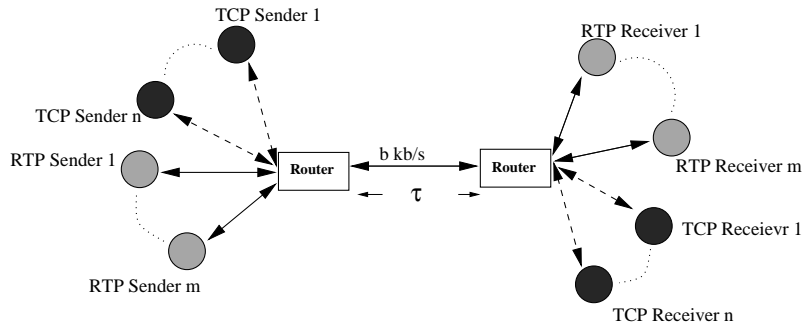


Figure 2: ALS performance testing topology

For the performance tests we consider two cases: in the first case, the number of ALS connections ( $m$ ) changes with time, and in the second case, the bottleneck bandwidth  $b$  is varied.

#### 4.1.1 Performance of ALS with changing number of connections

In this case, we investigate the effects of changing the number of ALS connections on the bandwidth distribution. We consider two scenarios: in the first scenario, the number of ALS connections increases with the time interval between the starting points of two connections set to less than the typical interval between two subsequent RTCP messages of 5 seconds. In the second scenario, the interval is set to a larger value. In both scenarios, the link bandwidth is set to 10 Mb/s and the round trip propagation delay to 0.01 seconds. The number of TCP connections is set to 3. Fig. 3 shows that even though the addition of connections in this topology causes at least a 10% increase in the number of connections at the routers the transmission rate of a sender starting at time 0 changes in accordance with the new network conditions for both scenarios without facing high losses, see Fig. 3(b). This test also shows the importance of choosing an appropriate value for the utilization factor at the routers. Choosing a high value would not account for sudden changes in the network conditions, i.e., the number of connections. As the ALS connections need a few seconds until receiving a new value of the updated fair bandwidth share to use, any increase in the number of connections would result in losses until the senders adjust their transmission rate.

Fig. 3(c) and Fig. 3(d) also show that the TCP connections receive a fair bandwidth share during the entire simulation time. Actually, they even receive a share that is around 10% higher

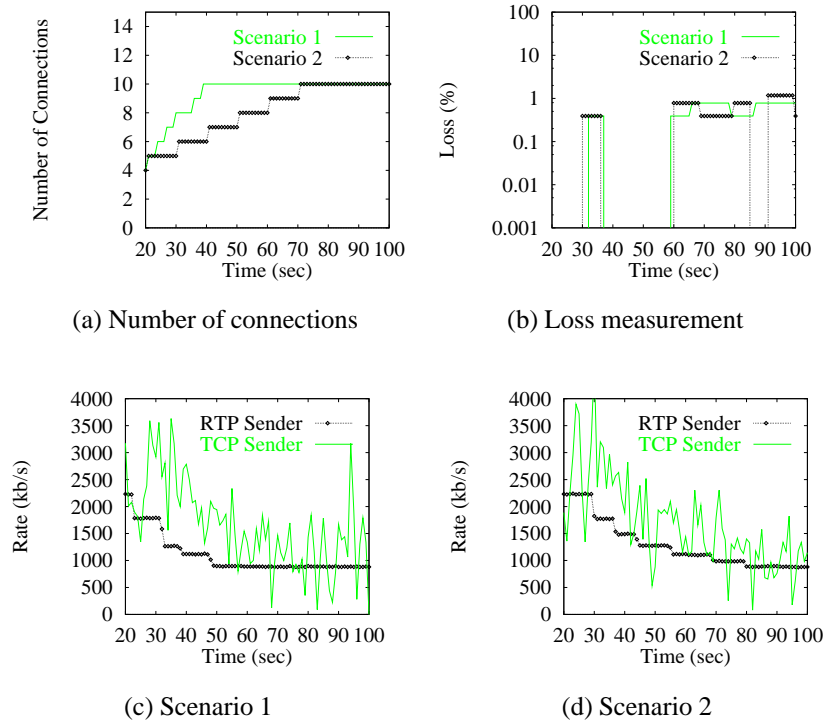


Figure 3: Performance of ALS with varying number of competing connections

than the share used by the ALS connections. This results from the shorter adaptation interval of TCP which allows TCP connections to increase their transmission rate each round trip time and thus faster utilize any available bandwidth.

#### 4.1.2 Performance of ALS with changing bottleneck bandwidth

In this part, the bottleneck bandwidth is varied and the number of connections is held constant. We use 3 TCP connections and 7 ALS connections. Again the variation is done in intervals smaller and larger than 5 seconds. Just as in Sec. 4.1.1, Fig. 4 shows that the transmission rate of an ALS sender is adjusted in accordance with the available bandwidth and the losses are below 2%.

## 4.2 Performance of ALS in Terms of Max-Min Fairness

Already, there has been different proposals for sender based adaptation schemes that achieve high utilization, low losses and are fair towards competing TCP traffic [27, 5]. However, just as it is the case for TCP [10, 26], all those algorithms show severe fairness problems when mixing long and short distance traffic and thus do not fulfill the requirements of the max-min fairness

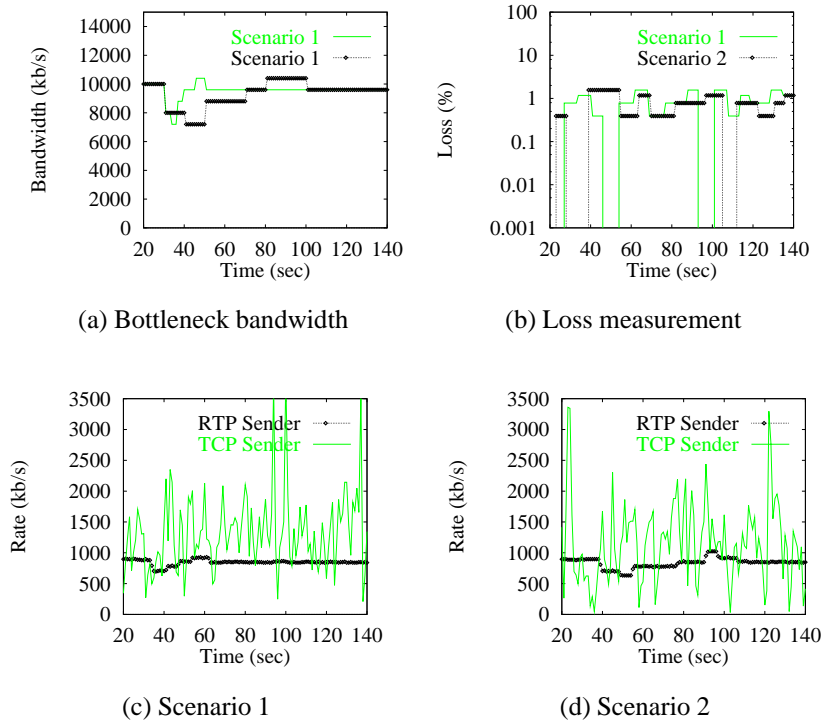


Figure 4: Performance of ALS with varying bottleneck bandwidth

criterion [25]. For testing the performance of ALS in terms of the max-min fairness criterion we use a chain topology with five routers and 7 connections, see Fig. 5. The distance between two neighboring routers is 1000 km and the distance from a source to a switch is 0.4 km. All the links have a bandwidth of 1 Mb/s. Depending on the number of routers they have to pass, three kinds of traffic can be distinguished:

- Long distance connections: The connection starting from source 0 traverses all four links. This connection will be denoted as  $C_0$ .
- Medium distance connections: The connections starting from sources 1 and 4 traverse two links. These connections will be denoted as  $C_1$  and  $C_4$ .
- Short distance connections: The connections starting from sources 2, 3, 5 and 6 traverse only one link. These connections will be denoted as  $C_2$ ,  $C_3$ ,  $C_5$  and  $C_6$ .

Connections	$C_0$	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$
fair share	250	250	500	250	250	500	750
achieved share	248	249	497	245	246	499	742

Table 1: Expected and achieved bandwidth share for the single connection's in kb/s

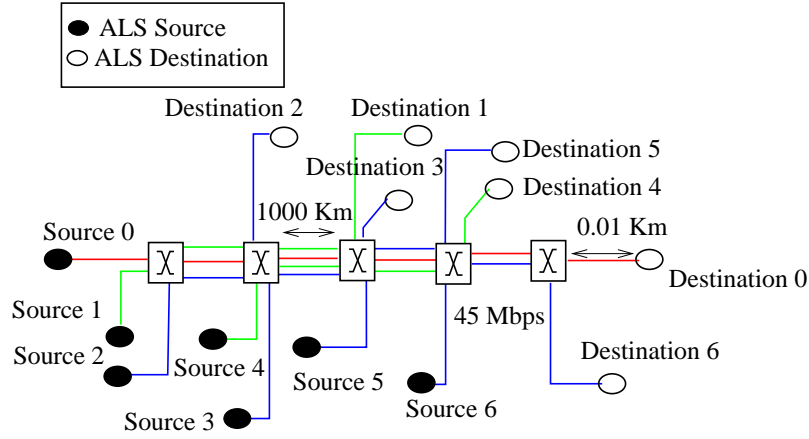


Figure 5: Chain topology for testing max-min fairness

The results of the transmission rates used by the different connections shown in Fig. 6 and Tab. 1 suggest that even though we use a simple distribution algorithm in the routers the connections manage to achieve their fair bandwidth share based on the adaptation mechanisms of the senders. The time required to achieve the fair bandwidth share is less than 100 seconds which corresponds roughly to 12 estimation cycles. While is is larger than what we would expect from an efficient ABR style algorithm [16] the gained router simplicity surely justifies this trade off.

### 4.3 Performance of ALS in Heterogeneous Environments

For testing the performance of ALS in a heterogeneous network environments we consider the topology depicted in Fig. 7.

A sender multicasts data over a 3 Mb/s link to router 1. Receiver 1 is connected over a 3 Mb/s to router 1, and receiver 2 is connected through router 2 over a 2 Mb/s link to router 1. Finally, router 3 is connected over a 1 Mb/s link to router 1. At the start of the simulation receivers 1 and 2 are in the multicast session. Receiver 0 joins the session at time 100 seconds. Each link is traversed additionally by 9 TCP connections so that the fair share for receivers 0, 1 and 2 are 90 kb/s, 270 kb/s and 180 kb/s on the basis of a utilization level of 90%. The model is based on IGMPv2. That is, joining and leaving a multicast session is done explicitly through join and leave messages. Additionally, we assume lossless control messages so that leaving and joining a multicast session is done instantously.

During the first 100 seconds the sender sets the transmission rate of the basic layer down to 170 kb/s, i.e., around the fair share we would expect for receiver 2. After receiver 0 joins the multicast session at time 100 seconds the sender reduces the transmission rate of the basic layer

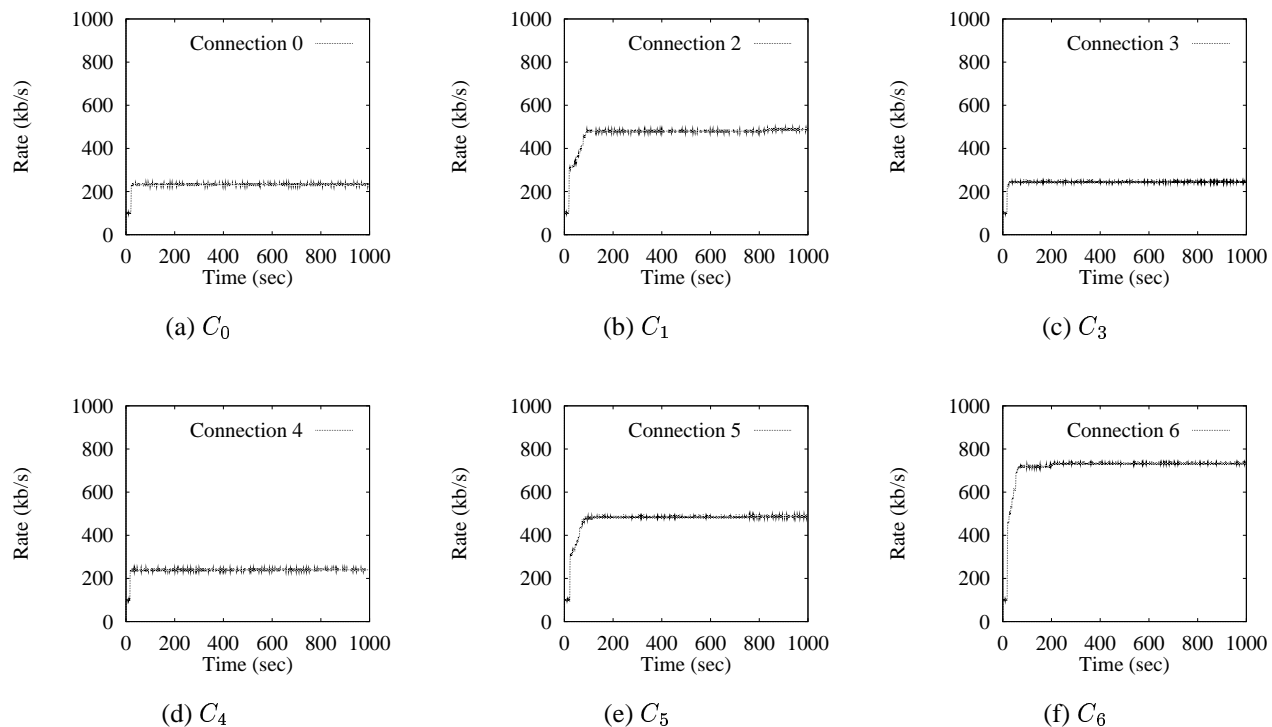


Figure 6: Bandwidth distribution for the case of a chain topology for connections  $C_0$  to  $C_6$

further to 90 kb/s, or the fair share of receiver 0. The transmission rate on the 2 upper layers is set now to 70 kb/s, see Fig. 8(a). Receiver 2 joins now layer 2 and receives an overall rate of 160 kb/s.

During the entire simulation run, receiver 1 receives all three levels and its perceived rate does not alter with the new distribution after receiver 0 joins in. The values of the reception rates at the receivers, see Fig. 8(b), suggest that ALS successfully manages to dynamically distribute the data over three different levels in accordance with the actual heterogeneity of the system. Additionally, the receivers join the appropriate number of layers to their capacity without the need for probing the network. Note also, that irrespective of the joined number of layers the bandwidth is being shared fairly among the ALS receivers and the TCP connections at the routers, see Fig. 9.

## 5 Summary and Future Work

With the available load service (ALS) we presented an efficient yet simple mechanism for adapting the transmission behavior of senders in accordance with the network congestion state and

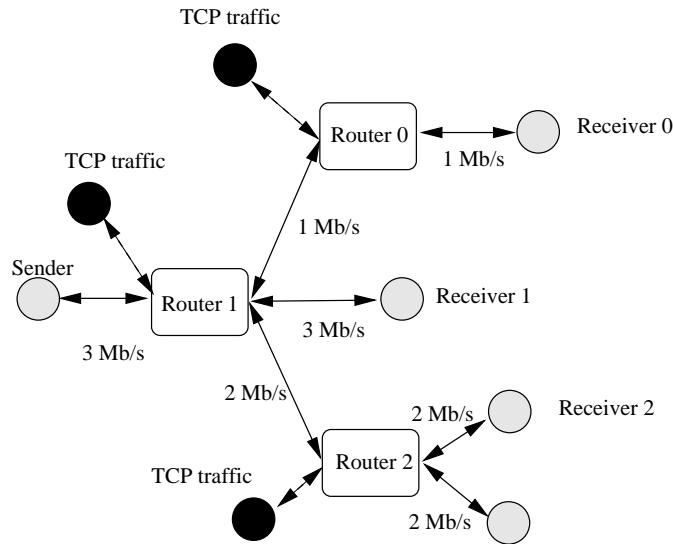


Figure 7: Test topology for ALS in heterogeneous environments

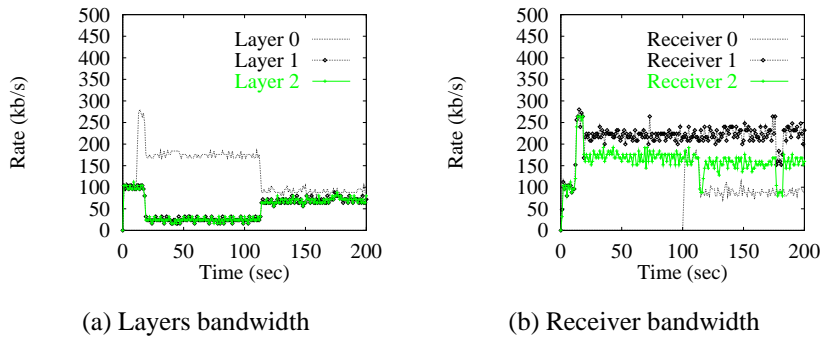


Figure 8: Bandwidth distribution of the layers and measured receiver rates

heterogeneity of the receivers. The simulations we made suggest the efficiency of ALS in terms of bandwidth utilization, loss reduction and fairness. However, the performance of ALS in environments with heavier loads and more members joining and leaving a multicast session still needs to be investigated. Also, the effects of choosing the length of the adaptation intervals on the achieved utilization and fairness is to be considered. Additionally, the performance of ALS under a real network situation has to be tested. In this context, investigating the processing overhead of the RTCP messages at the routers will be of utmost importance.

While ALS is simpler than ABR and supports multicast, ABR offers a minimal bandwidth grantee. This is of great advantage for multimedia communication that is rather flexible in nature but requires a minimal guaranteed QoS. Such a grantee can be achieved by integrating ALS with a reservation scheme such as YESSIR [21]. YESSIR is a sender based reservation proto-

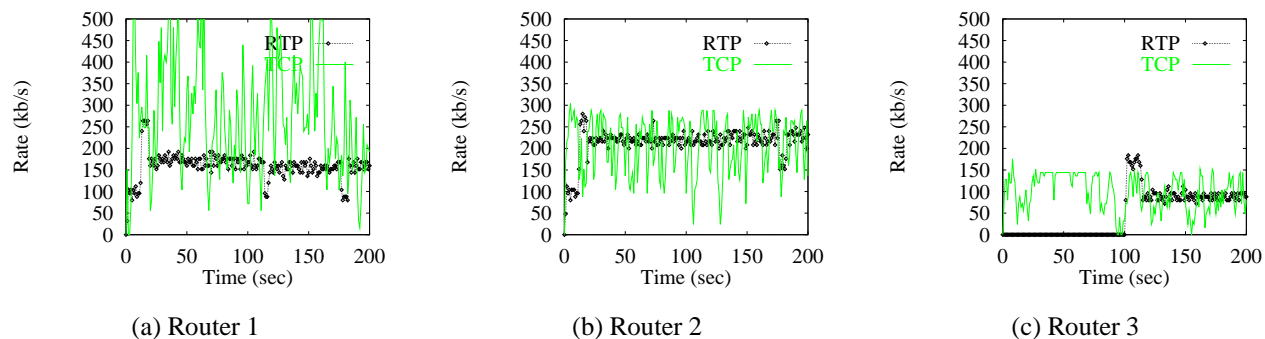


Figure 9: Bandwidth distribution between TCP and ALS at the different routers

col that uses the RTCP protocol for signaling reservations. Through the Integration with ALS, the receiver can reserve a minimal QoS using YESSIR and try to achieve a better QoS level through utilizing free network resources using ALS. Alternatively, an approach similar to the scalable resource reservation protocol (SRP) [1] or renegotiated constant bit rate (RCBR) [13] can be integrated with ALS. That is, the sender keeps informing the network about its desired transmission rate and the router indicate the actual reserved capacity. Additionally, the senders can still try to use a higher transmission rate using the adaptation scheme of the sender.

Another important issue to consider is the aspect of flow protection. That is, protecting adaptive flows from unresponsive ones that do not reduce their transmission rate during congestion states and might thus lead to the starvation of adaptive traffic. ALS provides a minimal degree of protection by allowing the senders to transmit at their fair share regardless of the behavior of other connections. However, without protection schemes at the routers, a greedy connection will still cause ALS connections to face losses and can receive a bandwidth share higher than its fair one. Lakshman et al. [29] present an intelligent buffer management approach that supports the isolation of responsive from unresponsive flows in a simple way. In an environment with routers supporting such buffering mechanisms ALS would be most suitable for achieving high utilization, low losses and fair bandwidth distribution.

## 6 Acknowledgments

The RTP/RTCP simulation models were mainly implemented by Timur Friedman and improved by Frank Emanuel. The comments of Adam Wolisz are gratefully acknowledged and were the basis for various aspects of this work.

## References

- [1] W. Almesberger, T. Ferrari, and J. L. Boudec. Srp:scalable resource reservation for the internet. In *IWQoS'98*, Napa, California, May 1998.
- [2] E. Amir, S. McCanne, and R. Katz. Receiver-driven bandwidth adaptation for light-weight sessions. In *ACM Multimedia'97*, Seattle, Nov. 1997.
- [3] A. W. Barnhart. Explicit rate performance evaluations. Technical Report 94-0983, ATM Forum, Oct. 1994.
- [4] S. Blake. Some issues and applications of packet marking for differentiated services. Internet Draft, Internet Engineering Task Force, Dec. 1997. Work in progress.
- [5] J.-C. Bolot, T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the internet. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 58–67, London, England, Aug. 1994. ACM.
- [6] R. Braden, L. Zhang, and S. Berson. Resource reservation protocol (RSVP) – version 1 functional specification. Internet Draft, Internet Engineering Task Force, Nov. 1995. Work in progress.
- [7] I. Busse, B. Deffner, and H. Schulzrinne. Dynamic QoS control of multimedia applications based on RTP. *Computer Communications*, 19(1):49–58, Jan. 1996.
- [8] C. Diot, C. Huitema, and T. Turletti. Multimedia application should be adaptive. In *HPCS*, Mystic, Connecticut, aug 1995.
- [9] S. Fahmy, R. Jain, R. Goyal, B. Vandalore, S. Kalyanaraman, S. Kota, and P. Samudra. Feedback consolidation algorithms for ABR point-to-multipoint connections in ATM networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, California, Mar. 1998.
- [10] S. Floyd. Connections with multiple congested gateways in packet-switched networks, part 1—one-way traffic. *ACM Computer Communication Review*, 21(5), Oct. 1991.
- [11] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug. 1993.
- [12] S. Floyd and F. Kevin. Router mechanisms to support end-to-end congestion control. Technical report, LBL-Berkeley, Feb. 1997.
- [13] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A simple and efficient service for multiple time-scale traffic. *IEEE/ACM Transactions on Networking*, pages 741 – 755, Dec. 1997.
- [14] U. Horn and B. Girod. Scalable video coding for the internet,. In *8th Joint European Networking Conference*, Edinburgh, England, May 1997.
- [15] R. Jain, S. Fahmy, S. Kalyanaraman, and R. Goyal. ABR switch algorithm testing: A case study with ERICA. Technical Report 96-1267, ATM Forum, Oct. 1996.
- [16] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. An efficient rate allocation algorithm for ATM networks providing max-min fairness. In *HPN'95*, Palma de Mallorca, Spain, Sept. 1995. IFIP.
- [17] D. Katz. IP router alert option. Technical Report RFC 2113, Internet Engineering Task Force, Feb. 1997.
- [18] I. Kouvelas, V. Hardman, and A. Watson. Lip synchronization for use over the internet: Analysis and implementation. In *GLOBECOM'96*, London, UK, Nov. 1996.
- [19] S. McCanne and V. Jacobson. vic: A flexible framework for packet video. In *Proc. of ACM Multimedia '95*, Nov. 1995.
- [20] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *SIGCOMM Symposium on Communications Architectures and Protocols*, Palo Alto, California, Aug. 1996.
- [21] P. P. Pan and H. Schulzrinne. YESSIR: A simple reservation mechanism for the Internet. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998. also IBM Research Technical Report TC20967.

- [22] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications. Technical Report RFC 1889, Internet Engineering Task Force, Jan. 1996.
- [23] S. S. Shirish. ATM Forum traffic management specification version 4.0. Technical Report 94-0013R6, ATM Forum, June 1995.
- [24] D. Sisalem. Behavior of various switch mechanisms for the ABR service in the presence of persistent and dynamic traffic. In *The Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS'97)*, Chalkidiki, Greece, June 1997.
- [25] D. Sisalem. Fairness of adaptive multimedia applications. In *International Conference on Communications (ICC'98)*, Atlanta, USA, June 1998.
- [26] D. Sisalem and H. Schulzrinne. Congestion control in TCP: Performance of binary congestion notification enhanced TCP compared to Reno and Tahoe TCP. In *International Conference on Network Protocols (ICNP)*, pages 268–275, Columbus, Ohio, Oct. 1996.
- [27] D. Sisalem and H. Schulzrinne. The loss-delay based adjustment algorithm: A TCP-friendly adaptation scheme. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998.
- [28] W. R. Stevens. *TCP/IP illustrated: the protocols*, volume 1. Addison-Wesley, Reading, Massachusetts, 1994.
- [29] B. Suter, T. Lakshman, D. Stiliadis, and A. Choudhury. Design considerations for supporting TCP with per-flow queueing. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, California, Mar. 1998.
- [30] K. Thompson, G. J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, 11(6):–, November/December 1997.
- [31] O. Verscheure, P. Frossard, and M. Hamdi. MPEG-2 video services over packet networks: Joint effect of encoding and data loss on user-oriented QoS. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Cambridge, England, July 1998.
- [32] L. Vicisano, L. Rizzo, and J. Crowcroft. TCP-like congestion control for layered multicast data transfer. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, San Francisco, USA, Mar. 1998.
- [33] F. Wilson, I. Wakeman, and W. Smith. Quality of service parameters for commercial application of video telephony. In *Human Factors in Telecommunication Symposium*, Darmstadt, Germany, Mar. 1993.
- [34] L. Wu, R. Sharma, and B. Smith. Thin streams: An architecture for multicasting layered video. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, St. Louis, USA, May 1997.