An Architectural Comparison of ST-II and RSVP

Danny J. Mitzel[1 2] Deborah Estrin[1 3] Scott Shenker[4] Lixia Zhang[4]

{mitzel, estrin}@usc.edu {shenker, lixia}@parc.xerox.com

Abstract

This paper presents a comparison of two reservation protocols, ST-II [5] and RSVP [6], in terms of their ability to support applications anticipated to be in common use in future Integrated Services Networks. Our main objective is in comparing the scaling behavior of the protocols; this affects the size and number of groups that can be supported. We use simulations to determine the network-wide resource requirements for each protocol to support a number of application communication styles, across a range of group sizes and membership distributions. We also present a comparative analysis of the protocol features to accommodate network and group membership dynamics.

1 Introduction

There has been considerable research effort recently in developing an integrated services network architecture to support new applications, such as remote video, multimedia conferencing, scientific visualization, and virtual reality. Two requirements of many of these new applications are their need for Quality of Service (QoS) guarantees from the network, and support for multipoint-to-multipoint communications. Traditional data networks, such as those based on the DARPA TCP/IP protocol suite, maximize network utilization by multiplexing multiple data streams, can provide multipoint communication, and provide robustness by adapting to network dynamics; however, they typically provide only a datagram, best-effort delivery service. Current circuit switched telecommunication and ISDN networks provide service guarantees; however, the circuit model leads to inefficient use of network resources when sending bursty data, it does not adapt to link and router failures, and it lacks support for multipoint communications. The goal of an Integrated Services Packet Network (ISPN) is to merge these two paradigms; combining the multiplexing, multipoint communication, and robustness of packet switched networks, with the service guarantees of the circuit switched model.

Development of this new ISPN network architecture requires several distinct components, including: (1) a flow specification defining the source traffic stream, and receiver service requirements; (2) a routing protocol supporting QoS and multicast paths; (3) a reservation protocol to create and maintain resource reservations; (4) an admission control algorithm to maintain network load at a proper level; and (5) a packet service algorithm to schedule packet transmissions in an order that maintains service guarantees for individual data streams.

The reservation protocol is responsible for requesting allocation and release of network resources along the data distribution path to ensure QoS requirements are met. The resulting network utilization and efficiency depends to a great extent on the reservation protocol's service model and dynamic response. Service models can be characterized by the set of communication styles (point-to-point, multipoint) and reservation styles (to control aggregation of reservations at intermediate switches) supported, and by the ability to support heterogeneous group members. The dynamic response of the reservation protocol can be characterized by the support for dynamic group memberships, and the response to link and router failures.

[1] Computer Science Department, University of Southern California, Los Angeles, CA 90089.
[2] Hughes Aircraft Company, P.O. Box 92919, Los Angeles, CA 90009.
[3] USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292.
[4] Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304.

Initial work in supporting end-to-end guaranteed service, within the Internet protocol suite, resulted in the development of the ST stream protocol [3], and the later development of a second version of the protocol, ST-II [5]. A more recent proposal, targeted at supporting the resource reservation requirements of an ISPN, is the RSVP protocol [7]. In this paper we compare the operation of the ST-II and RSVP protocols in support of applications typical of an ISPN. In Section 2 we present an overview of the two protocols. We divide our comparison of the protocols into two distinct topics, static resource requirements, and dynamic behavior. The static resource investigation looks at network resource requirements to support a fixed set of communicating applications over a range of communication styles; these results are presented in Section 3. In Section 4 we describe the protocol mechanisms for supporting network dynamics (link and router failure/recovery), and look at the protocol overhead associated with accommodating group membership dynamics. Section 5 concludes with a summary and a few words on future work to be performed.

2 Protocol Overview

A simplistic reservation service could be implemented on top of a point-to-point service model by establishing a separate reservation between each pair of communicating applications. This service model might be sufficient if the only goal of an ISPN was to extend the current IP point-to-point service model with QoS support. However, the goal of the ISPN architecture described is to provide efficient support for new applications requiring QoS support *and* multipoint communications. As we shall see, the simplistic point-to-point reservation mechanism is very inefficient in terms of network resource allocation required to support multipoint communications.

An enabling technology for supporting multipoint communications, incorporated into both ST-II and RSVP, is multicast routing. Deering [2] describes how multicast distribution can be incorporated into a datagram network, to improve network resource utilization. However, ST-II and RSVP make different assumptions about the level of multicast support provided by the network. ST-II builds a multicast distribution tree based upon unicast routing tables, and performs the replication and forwarding of data packets. RSVP is decoupled from the multicast routing and data forwarding functions, it is assumed they are provided by the underlying network. The only direct reference to multicast identifiers within RSVP is to associate data source and receivers with a group. This difference in assumptions about the level of multicast support provided by the network is largely historical; at the time ST-II was developed, there was no internetwork multicast routing. While incorporating multicast forwarding in the ST-II protocol adds some processing overhead, it does not affect the resource reservation or message overhead, and thus does not affect our comparative analysis.

The multipoint communication capabilities of ST-II and RSVP provide improved network resource utilization when compared to the simplistic point-to-point reservation model. Additional gains, in terms of improved resource utilization, are possible by incorporating application-level communication requirements in the reservation service model. In the following subsections an overview of the ST-II and RSVP reservation protocol and service model are presented; these act as the basis for our comparisons throughout the remainder of the paper. It should be stressed that these descriptions provide only a summary of the protocol functions relevant to the discussion; for a complete protocol description, the appropriate protocol documents should be consulted.

2.1 ST-II Protocol

ST-II [5] models a resource reservation as a simplex data stream rooted at the source and extending to all receivers via a multicast distribution tree. Stream setup is initiated when a source ST agent sends a *Connect* message, listing the flow specification and initial set of participants. Connect processing at each intermediate ST agent involves determining the set of next hop subnets required to reach all downstream receivers, installing multicast forwarding state, and reserving network level resources along each next hop subnet. If the actual resource allocation obtained along a subnet is less than the amount requested, then this is noted in the Connect packet by updating the flow specification. Upon receiving a Connect indication, a receiver must determine whether it wishes to join the group, and return either an *Accept* or a *Refuse* message to the stream source. In the case of an Accept, the receiver may further reduce the resource request by updating the returned flow specification.

During connection setup, the stream source must wait for an Accept/Refuse reply from each initial receiver before beginning data transmission. ST-II treats the entire stream as a homogeneous distribution path. Whenever the source receives an Accept with a reduced flow specification, it must either adapt to the lower QoS for the entire stream, or reject group participation for the specific receiver by sending it a *Disconnect* message.

Group membership dynamics are accommodated by allowing stream receivers to be added and deleted after initial stream setup. Each addition of a receiver requires an interaction with the stream source to trigger the sending of a Connect message; this interaction is not included in the protocol and the specification suggests that it be done out-of-band using IP. As in the initial setup, the stream source must examine the flow specification in a returned Accept, and either reduce its QoS or reject the new receiver if the resources allocated are less than those currently allocated for the stream. Deletion of receivers may be done asynchronously, by a receiver sending a Refuse message, or the source sending a Disconnect message; the Disconnect message can either list individual receivers to remove, or set the Global-Disconnect flag to tear down the entire stream.

Reliability and robustness is incorporated into the ST-II protocol via two separate mechanisms. First, all control messages used to create and manage a stream are transmitted reliably using hop-by-hop acknowledgements with retransmission. Second, a *Hello* protocol is used to query the status of neighboring ST agents sharing active streams. When a change in reachability between neighboring ST agents is detected, automatic stream recovery may be possible.

The only service model directly supported by ST-II is that of a point-to-multipoint reservation, over a homogeneous distribution tree. The ST-II protocol specification defines the concept of a group of streams, which may be useful in developing more sophisticated service models. The group mechanism provides a method to express relationships among individual streams, and to perform operations on the group as a whole. However, this is a purely experimental feature, and no stream relations have been defined at this time. We do not consider the group mechanism in any of our analysis.

2.2 RSVP Protocol

The RSVP [6] reservation model is similar to ST-II, in that a data stream is modeled as a simplex distribution tree rooted at the source and extending to all receivers. However, the mechanisms for group sources and receivers to establish resource reservations, and the reservation styles supported, differ substantially from the ST-II model.

Under RSVP, a source application begins participation in a group by sending a *Path* message containing a flow specification, to the destination multicast address. The Path message serves two purposes, to distribute the flow specification to the receivers, and to establish Path state in intermediate RSVP agents to be used in propagating reservation requests toward specific sources. RSVP does not restrict a source from transmitting data, even when no receiver has installed a reservation to it; however, data service guarantees are not enforced.

Before establishing a reservation, each receiver must first join the associated multicast group to begin receiving Path messages; this multicast group join operation is a function of the multicast routing protocol, and is outside the scope of RSVP. Each receiver may use information from Path messages, and any local knowledge (computing resources available, application requirements, cost constraints) to determine its QoS requirements; it is then responsible for initiating its own *Reservation* request message. Intermediate RSVP agents reserve network resources along the subnet leading toward the receiver, then use the established Path state to propagate the Reservation request toward the group sender(s). Reservation message propagation ends as soon as the reservation "splices" into an existing distribution tree with sufficient resources allocated to meet the requested QoS requirements. This *receiver-initiated*[1] reservation style enables RSVP to

[1] The receiver-initiated approach was inspired by Deering's work on multicast routing [2], in which the receiver is responsible for initiating group membership requests.

accommodate heterogeneous receiver requirements.

RSVP models a reservation as two distinct components, a resource allocation and a packet filter. The resource allocation specifies *what amount* of resources is reserved, while the packet filter selects which packets can use the resources. This distinction between the reservation and filter, and the ability to change the filter without changing the resource allocation, enables RSVP to offer several different reservation styles.

RSVP maintains *soft state*[2] in intermediate switches, and uses periodic refreshes to provide reliability and robustness. Path refreshes automatically adapt to changes in the multicast distribution tree, and install Path state in any new branches of the tree. Reservation refreshes maintain established reservations, and incorporate new receiver reservations. This refresh based mechanism allows idle reservations and state to be automatically timed out and recovered.

The receiver-initiated reservation mechanism, and distinction between resource allocation and packet filter, provide the flexibility for RSVP to support several distinct service models. As mentioned, the receiver-initiated reservation incorporates heterogeneous receiver requirements in the reservation service. RSVP offers several reservation styles, to capture application-level service requirements; these dictate how reservation requests from individual receivers should be aggregated inside the network. RSVP defines three reservation styles, *wildcard*, *fixed-filter*, and *dynamic-filter*. A wildcard reservation indicates that a source specific filter is not required, any packets destined for the associated multicast group may use the reserved resources; this allows a single resource allocation to be made across all distribution paths for the group. The wildcard reservation is useful in support of an audio conference, where at most a small number of sources are active simultaneously, and may share the resource allocation. When a source specific filter is required, a receiver may indicate whether it desires to receive a fixed set of sources, or the ability to dynamically switch its reservation among the sources. A fixed-filter reservation cannot be changed during its lifetime, without re-invoking setup and admission control; this allows the reservation to be shared among multiple requests for the same source. Dynamic-filter reservations allow a receiver to modify its filter over time, this requires that sufficient resources be allocated to handle the worst case when all downstream receivers take input from different sources.

3 Static Analysis

The protocol descriptions in Section 2 noted that both ST-II and RSVP incorporate multipoint communication directly in their service models to improve network resource utilization, and model a data stream as a simplex multicast distribution tree; however, the RSVP model incorporates receiver initiated reservations and several reservation styles, providing additional opportunities to improve network-wide resource utilization. In this section we look at several applications typical of an ISPN, map the two service models of the two protocols to the application communication requirements, and compare the network-wide resource requirements for supporting the application.

3.1 Supporting Mixing Reservations

A number of multipoint-to-multipoint applications have application-level constraints that prohibit all data sources from transmitting simultaneously, a good example is a large audio conference. In an audio conference there is typically only one person talking at a time; when more than a few speakers are simultaneously active the result is usually unintelligible. Incorporating support for this service model presents an opportunity to reduce network resource requirements for the group; instead of reserving enough resources for every potential speaker to transmit simultaneously, it may be adequate to reserve only enough resources to handle a few simultaneous audio channels. RSVP is able to capture these application service requirements exactly, by using

[2] Clark [1] characterizes the concept of soft state in support of type of service as follows; "It would be necessary for the gateways to have flow state in order to remember the nature of the flows which are passing through them, but the state information would not be critical in maintaining the desired type of service associated with the flow. Instead, that type of service would be enforced by the end points, which would periodically send messages to ensure that the proper type of service was being associated with the flow. In this way, the state information associated with the flow could be lost in a crash without permanent disruption of the service features being used."

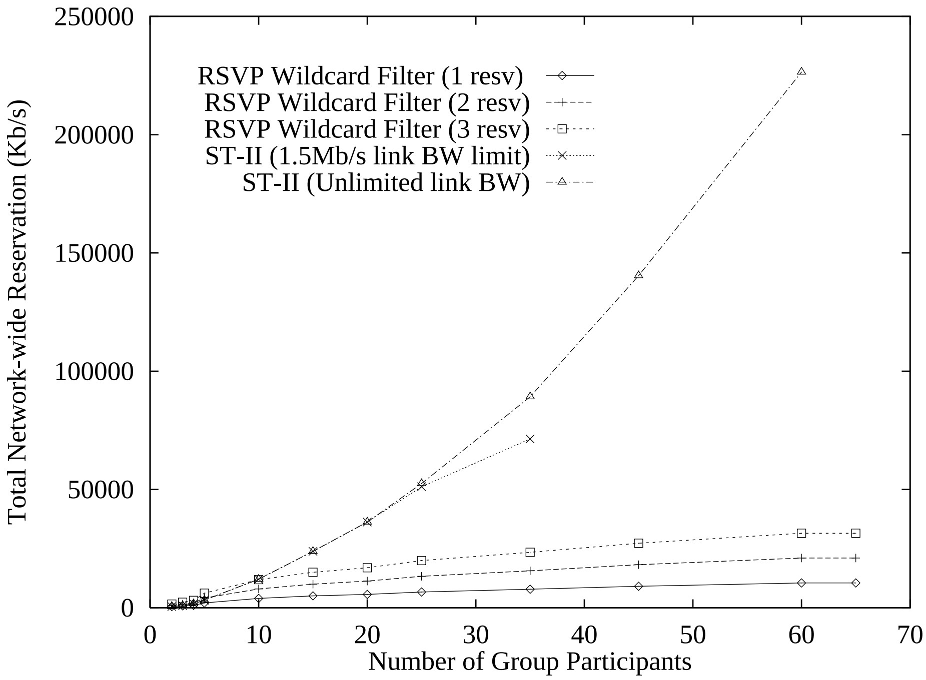

Figure 1: Comparison of Resource Requirements in Support of (Mixing) Audio Conference

a wildcard reservation, and requesting sufficient resources to carry the maximum number of simultaneously active sources. ST-II requires that a separate stream reservation be established for each audio source.

In this section we compare the total network-wide resource allocation for the two reservation protocols to support the audio conference communication style. We model a hierarchical network containing 60 routers interconnected via 82 links, and vary the number of audio sources from 2 up to 65. Each audio source was randomly distributed among the 60 nodes,[3] and is modeled as a request for a 64Kb/s PCM audio stream. Figure 1 presents the total network-wide resources allocated under the two reservation protocols to support audio conferences of various sizes. For the RSVP wildcard reservation style, we show the resource requirements for when each group member requests a reservation for 1, 2, or 3 voice streams worth of bandwidth; this represents the limit on number of simultaneous speakers. For ST-II we show the resources reserved when each network link is modeled as having 1.5Mb/s capacity, and when link bandwidth is unlimited.

The small slope of the RSVP plots highlights how the wildcard filter reservation is able to capture the application's communication requirements, to improve network resource allocation efficiency, while ST-II requires a large resource over-allocation to support the group. Under the RSVP model, adding a new member involves reserving resources only along new links required to "splice" into the distribution mesh.[4] As a group becomes more "dense" (group membership covers a higher percentage of the total network nodes), the average number of network hops required to splice into the distribution tree decreases, resulting in a smaller overhead per new member. In contrast, adding a new member under the ST-II model requires splicing into N-1 existing distribution trees to receive each of the other participants, and also setting up a new distribution tree from the new participant to all existing members.

Allocating an independent resource reservation for each ST-II source effectively places an upper limit on the maximum group size that can be supported. Figure 2 shows the number of reservations established on each active distribution link for an audio conference composed of 20 participants randomly distributed among the 60 end-sites. The labels on each active link represent the number of 64Kb/s reservations allocated along the link in each direction. The figure shows that for the common case where an end site has a single access link to the internet, for a group size of N, N-1 reservations must be established on the local access

[3] The random placement function, used throughout the simulations, selected a random order for adding participants to unique nodes; this precludes multiple participants at a single node until all nodes have at least one participant.

[4] Note that the reservation across the distribution mesh is based upon the *union* of the links in all distribution trees; not the sum, such as in ST-II.

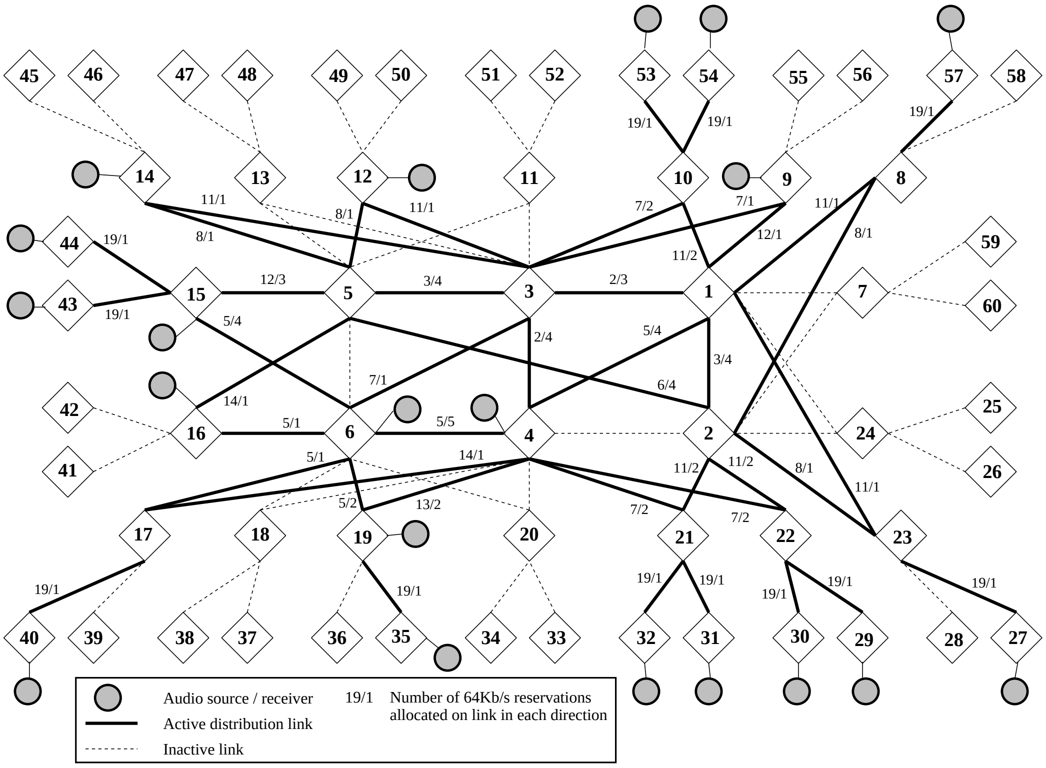

Figure 2: Link Reservations for ST-II Audio Conference (20 participants)

link. Making the optimistic assumption that the packet service algorithm can maintain QoS guarantees at 100% link utilization, the maximum group size is limited to

$$Maximum\ Group\ Size = \frac{Bottleneck\ Link\ Bandwidth}{Single\ Stream\ Resource\ Request} + 1$$

participants. The plot of ST-II reservations in Figure 1 illustrates this limitation when link capacity was fixed at 1.5Mb/s for each access links.[5] A 1.5Mb/s bottleneck link, and 64Kb/s audio stream, limits group size to a maximum of 24 participants; once this number is exceeded resource allocation requests will be rejected, due to insufficient available resources on the bottleneck links. This phenomena is visible in Figure 1 for group sizes of 25 and 35, where the resources allocated for ST-II with 1.5Mb/s links falls below the total resource allocation requested, as indicated by the discrepancy when compared to the ST-II plot with unlimited link capacity. In the simulations, for a group size of 25, 2.3% of the link resource allocation requests were rejected; for a group size of 35 this rose above 19%. RSVP wildcard reservations do not encounter this scaling problem, the maximum resource reservation across all links is limited to the number of simultaneous channels requested.

A final observation on supporting audio conferences is to note that the resource requirements of a RSVP wildcard reservation are bounded, while ST-II resource requirements are unbounded. For RSVP, once there is a group member at each network node, resources along the complete distribution path have been allocated and no further resources need be allocated to accommodate additional group members; this is evident in the plot in Figure 1 by the zero slope line when going from 60 to 65 group participants. Under ST-II, adding a new group member always requires allocating a reservation from the new member to all existing members.

3.2 Supporting Heterogeneous Groups

In a global-scale internetwork, receivers, as well as the paths used to reach the receivers, can have very different properties from one another. Network and host technologies are likely to span several orders of

[5] This is a very common scenario in today's Internet, where many end-sites and regional networks are based on T1 interconnects.

Number of Low Quality Receivers	ST-II Resource Allocation(Kb/s)	RSVP Resource Allocation(Kb/s)
0	2944	2944
10	2944	2656
20	2944	2176
30	2944	1600
40	736	736

Table 1: Resource Requirements to Support 40 Receiver Heterogeneous Audiocast

magnitude in terms of bandwidth and processing capabilities. In this environment it may not be reasonable to assume that all receivers in a group possess the same capacity for processing incoming data, or desire the same QoS from the network.

Applications involving wide-spread distribution services such as cable-TV distribution, or broadcasting of an audio/video lecture may be able to accommodate additional participants by incorporating support for heterogeneous receiver capabilities. An application may employ a hierarchical coding scheme, or provide multiple data streams utilizing different media encodings, to present varying signal quality levels to the receivers. Each receiver is then able to determine its QoS requirements based on local constraints. Applications taking advantage of this heterogeneous environment require a reservation service model incorporating heterogeneous resource reservations, to minimize network resource utilization.

ST-II and RSVP accommodate heterogeneity very differently. Under the ST-II service model, a data source must view the stream as a homogeneous distribution path. After stream setup, the source application must conform to the minimum resource allocation, forcing all participants to suffer with the least capable or least demanding receiver. To satisfy the most demanding receiver, the source must allocate the maximum requested resources along all links. RSVP's receiver-initiated reservation scheme propagates Reservation requests from a receiver, up the sink tree toward the source, "splicing" into the distribution tree. The reservation establishment process reserves the minimum resources, on each link, required to satisfy the QoS requirements of all downstream receivers. Thus, RSVP incorporates support for heterogeneous reservations directly in the protocol, in a manner transparent to both end-points.

In this section we compare the total network-wide resource requirements to support a heterogeneous mix of receivers participating in an audio lecture. The 60-node network model introduced in Section 3.1 is used, and the lecture is modeled as a single data source transmitting a "high quality" 64Kb/s audio stream, that also contains a sub-band 16Kb/s "low quality" audio stream.[6] Figure 3 shows the link reservations installed by RSVP to support 40 randomly selected receivers of the audiocast, using fixed-filter reservations; 20 receivers request the full 64Kb/s stream, and 20 receivers request the 16Kb/s "low quality" audio sub-channel. This diagram depicts the ability of the RSVP service model, using receiver-initiated reservations, to install a heterogeneous resource reservation across the distribution path. Only those branches leading to receivers requesting the "high quality" audio stream require the high bandwidth reservation, this can result in significant resource savings. For the scenario illustrated, only 29 of the 46 links in the multicast distribution tree require a high bandwidth reservation, resulting in greater than a 25% savings in network resource allocation, when compared to a homogeneous distribution tree.

Table 1 presents the total network-wide resource requirements for both ST-II and RSVP to support the 40 receiver audiocast, with various numbers of low quality receivers. The ST-II stream exhibits an "all-or-nothing" effect, due to the protocol's limitation of treating the stream as a homogeneous distribution path. As long as there is at least one demanding receiver, the maximum resources must be allocated along all links, this ensures the QoS for the demanding receivers is met. Under the RSVP service model, the total network-wide resources reserved reflects the minimum allocation required along the distribution path to satisfy all receivers QoS requests. As the number of low quality receivers increases, additional branches in the distribution path shed their high quality resource reservation, resulting in a gradual decrease in total network-wide resource allocation.[7]

[6] Specification of the mechanisms to encode/decode this stream, and the filter to select the sub-band audio are outside the current discussion.

[7] Note that in the worst case scenario of a linear network, RSVP allocation is identical to the ST-II case, while the best case scenario of a fully connected network yields allocations that are linear in the number of low quality receivers.

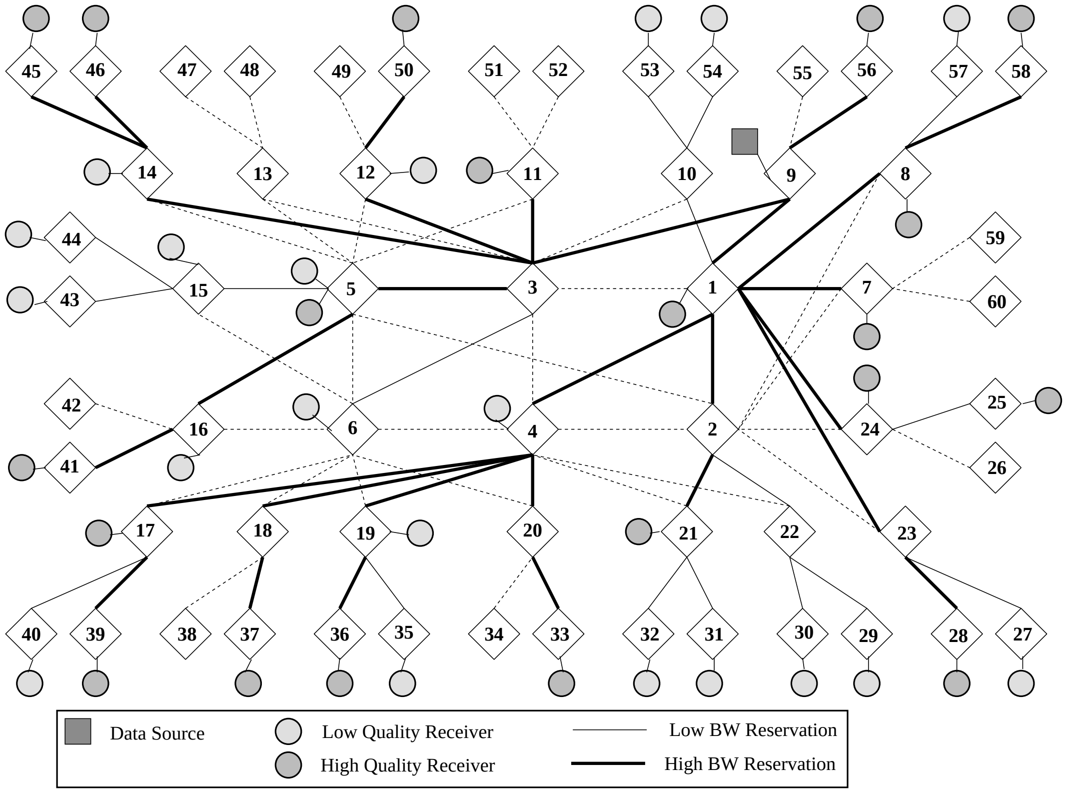

Figure 3: Link Reservations for RSVP Heterogeneous Audiocast (40 receivers)

3.3 Supporting Channel Selection

In large multiparty conferences a receiver may be unable to accommodate data streams from all active participants simultaneously, but would like the ability to select dynamically a subset of the sources to receive at any time. This restriction on number of simultaneous sources may be due to bandwidth limitations, display or codec hardware, or the inability of the user to assimilate information from all sources concurrently. We term this communication style *channel selection*. From the user's perspective there are two possible service models, assured channel selection and non-assured channel selection. A key characteristic of assured channel selection is that once a receiver has established its reservation, it should be guaranteed that a change request will not be denied. The non-assured channel selection model eliminates the constraint that a change request must be guaranteed to succeed. We define three alternative mechanisms for channel selection; *simultaneous fixed* and *dynamic filter* reservations for assured channel selection, and *fixed with replacement* reservations for non-assured channel selection.

The simultaneous fixed channel selection mechanism is based on allocating sufficient resources for *all* sources at setup time. This implements the assured channel selection model, sufficient resources are always available when switching between different sources; however, it also requires the receiver to accept data streams from all sources and discard the current "non-selected" streams. For an N-way conference, this reservation mechanism is equivalent to that presented in Figure 2 in Section 3.1, and suffers from the same group scaling limitations.

Another assured channel selection mechanism is the dynamic filter reservation style; which decouples the resource allocation and packet filter to provide for improved resource utilization. For a single receiver, reservation setup requests at most m resource allocations on a link (m is the maximum number of simultaneous sources), the receiver may then modify the associated packet filters to switch among sources. The actual number of resource allocations on a link is limited to the maximum number of non-overlapping reservations; this is the sum of all downstream receiver requests limited by the number of upstream sources. Figure 4 shows the number of reservations established on each active distribution link to support channel selection using dynamic filter reservations. The conference depicted is the same as that originally introduced in Figure 2,

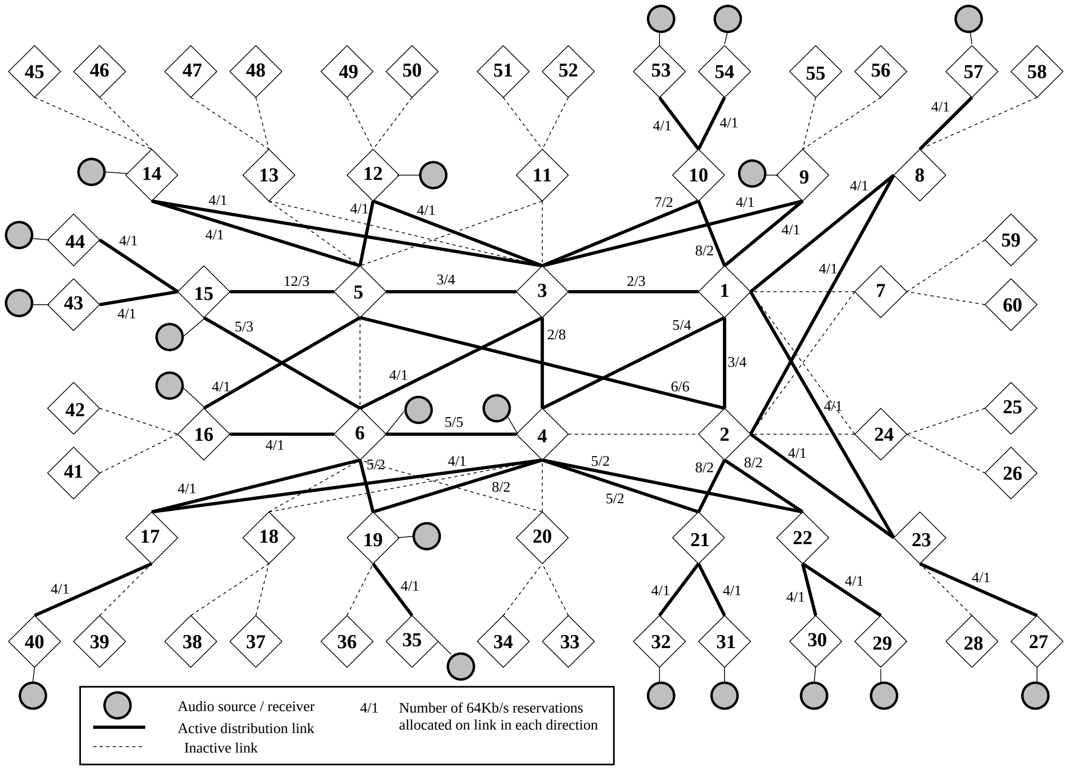

Figure 4: Link Reservations for Dynamic Filter Channel Selection (20 participants / 4 resv)

and each of the 20 members requests four simultaneous reservations. The labels on each active link represent the number of resource reservations allocated along the link in each direction. The figure shows that the dynamic filter channel selection mechanism suffers from some resource over-allocation, such as at switch 14 where 8 resource allocations were established to ensure selection over both access links. Note however that the resource allocation on each participants access link is limited by the maximum number of simultaneous sources requested, this reduces the scaling problem on the access links noted for the simultaneous fixed channel selection style.

The third channel selection alternative, fixed with replacement, allocates resources for only the currently selected set of sources. Switching is accomplished by tearing down an existing reservation, and then requesting a new resource allocation for the new source. This mechanism provides only the non-assured channel selection service; it does not provide a guarantee of success when switching sources since the new request may be blocked. Our primary interest in investigating the non-assured channel selection mechanism is that this resource allocation represents the minimum required to support the selected set of sources; this also represents a lower bound on resource requirements for assured channel selection, and can be used to assess the assured channel selection overhead.

Table 2 presents the total network-wide resource allocation for each of the channel selection mechanisms, to support selection among the participants of the N-way conference introduced in Section 3.1. Note again that the fixed with replacement channel selection mechanism does not provide assured switching among the sources; it provides a baseline to compare the overhead of the assured channel selection schemes. The Overhead Ratio represents the ratio of the resources allocated, compared to the minimum required to support the currently selected sources. We see that for the simulations conducted, the resource over-allocation incurred using the simultaneous fixed mechanism can be quite substantial, and the overhead grows as the group size is increased. The resource overhead in providing assured channel selection is much smaller using the dynamic filter mechanism; most importantly, for the class of graphs simulated, the overhead appears to be bounded as the group size is increased.[8]

[8] Note that in the best case of a linear network the overhead ratio is always 1, whereas in the worst case of a fully connected network when the receiver requests a single reservation the overhead ratio is N (where N is the number of sources).

Group Size	Fixed w/Replacement (4 Reservations) Resource Allocation(Kb/s)	Dynamic Filter (4 Reservations) Resource Allocation(Kb/s)	Dynamic Filter (4 Reservations) Overhead Ratio	Simultaneous Fixed (N-1 Reservations) Resource Allocation(Kb/s)	Simultaneous Fixed (N-1 Reservations) Overhead Ratio
5	3200	3200	1.0	3200	1.0
10	6592	8704	1.32	12032	1.83
15	9728	13184	1.36	23808	2.45
20	11840	18432	1.56	36224	3.06
25	14400	22720	1.58	52480	3.64
35	20160	32704	1.62	89152	4.42
45	26368	42048	1.59	140352	5.32
60	36416	57024	1.57	226432	6.22

Table 2: Channel Selection Resource Overhead

The only requirement on the reservation protocol to support the simultaneous fixed or fixed with replacement channel selection mechanism is that a fixed resource allocation can be established from each selected source to the receiver. Both the ST-II stream model, and the RSVP fixed-filter reservation style provide this service. For the dynamic filter channel selection mechanism, a distinction must be made between a resource allocation and the packet filter; this service is provided only by the RSVP dynamic-filter reservation style. Thus, ST-II can effectively only support the non-assured channel selection service model, since the simultaneous fixed mechanism is too costly, while RSVP can provide both the assured and non-assured channel selection service models.

4 Dynamic Analysis

In Section 3 we compared the resource requirements of the ST-II and RSVP protocols, to support a fixed set of group members. In a real, large scale internetwork environment, there may be frequent dynamic events that must be accommodated by the reservation protocol. These events include both network dynamics, such as link and router failure/recovery, and group membership dynamics, when participants join and leave the multicast group. It is extremely important that the group membership dynamics be supported efficiently, as membership change is expected to be a common occurrence, whereas topology change represents an exceptional event. In this section we describe the ST-II and RSVP mechanisms for supporting network dynamics, and compare the protocol overhead associated with accommodating group membership dynamics.

4.1 Network Dynamics

Section 2 described the mechanisms incorporated into the ST-II and RSVP protocols to provide reliability and robustness in the face of network dynamics. ST-II utilizes a reliable control message protocol, and a Hello protocol to monitor neighbor ST agent health, while RSVP uses a datagram control message protocol in conjunction with a soft state refresh mechanism. A difficulty in conducting a comparison of the dynamics of the two protocols is that both rely heavily on timers (ST-II Hello interval and RSVP refresh period), which have a great effect on the protocol overhead and recovery period; and no explicit timer values are mandated by the protocol standards. Instead, we compare the design philosophies behind the dynamics support in the two protocols.

The integration of support for network dynamics, in ST-II and RSVP, are substantially different in terms of both implementation and design philosophy. ST-II incorporates a failure detection mechanism using Hello, Status, and Notify messages, these add considerable complexity to the protocol,[9] while RSVP relies on the soft state refreshes to automatically adapt without additional protocol complexity. RSVP could be modified to incorporate a failure detection mechanism also, to trigger refreshes as an optimization; however, there are more fundamental differences that distinguish the protocols. The key difference between the two

[9] Partridge and Pink [4] note that much of the functionality is overlapping, leading to the elimination of the Hello, Status, and a portion of the Notify mechanisms in their implementation.

protocols is in where recovery takes place. ST-II requires that the network be responsible for correctness, by either restoring itself or reliably contacting the source; this leads to complex protocols with strange failure modes. Clark [1] notes that systems relying on distributed state are difficult to build and few truly provide protection against failure. RSVP leaves the final responsibility for maintaining reservations with the ends; this is consistent with the current Internet philosophy of "fate-sharing"[10] among the end-points.

It should be noted that even in steady state (no network or group dynamics) there is an overhead associated with both protocols. Under ST-II this overhead is a result of each ST agent periodically exchanging one Hello message with each active neighbor. Requiring the agent to track peers separately from streams may pose a slight complication in data structure organization; however, it results in a protocol that scales independent of the number of active streams. Protocol overhead in RSVP results from the periodic reservation refreshes, which each receiver is required to generate to maintain each reservation. This seems to imply that RSVP overhead scales directly with the number of participants; however, RSVP incorporates a protocol overhead reduction mechanism called "merging" to reduce this overhead. The merging process insures that only a single reservation message is propagated over a link per refresh period. With a wildcard reservation there is only a single reservation on each link for the entire group; for a fixed-filter reservation there is one reservation for each source forwarding along a link; while a dynamic-filter requires a separate reservation per receiver. Thus, RSVP protocol overhead scales with the number of reservations, and this is dependent on the reservation style.

4.2 Group Membership Dynamics

Large multicast groups, such as global distribution of a conference or lecture, are likely to encounter frequent membership dynamic "events." These events are a result of participants tuning into and leaving the conference. In a correctly functioning internet, group membership changes are much more common than network dynamic events; therefore, it is important that the reservation protocol be able to accommodate these membership dynamics efficiently. Protocol efficiency can be evaluated in terms of messaging overhead, and latency in adapting to changes. In this section we compare the protocol overhead for ST-II and RSVP to adapt to group membership changes.

Figure 5 depicts the distribution of a lecture from a single sender to 40 randomly selected receivers. Each of the participants is modeled as joining the lecture independently, and we list the total number of protocol messages transmitted on each link, when using the ST-II protocol to install the resource reservations. These messages represent the Connect, Accept, and Ack overhead for the protocol. Note how the end-to-end messaging of ST-II places an overhead on each link proportional to the number of downstream receivers. This results in links closer to the source becoming a "hot spot," they incur a higher overhead in terms of bandwidth and protocol processing overhead. One method to reduce this overhead, if multiple receivers join simultaneously, is to list the multiple targets in a single Connect; however, there is no way to avoid the resulting implosion of Accepts.

The receiver initiated reservations in RSVP result in a very different join overhead model. Assuming homogeneous receivers, the join overhead is reduced to one protocol message, on each link, in each direction. This represents a single Path message sent by the source to build the reverse path state, and a single reservation request sent by each receiver. The key to RSVP's reduced join overhead is the merging function; as soon as the reservation request splices into an existing distribution branch, the request can be merged (discarded). The situation becomes only slightly more complicated when heterogeneous receivers are introduced; in this case the merging function must ensure the request splices into the distribution tree, and there are sufficient resources allocated. This may result in multiple reservation messages being propagated over a link, if a more demanding request is received after a less demanding reservation has already been installed.

The use of receiver initiated reservations and reservation merging in RSVP result in two distinct advantages over the end-to-end protocol of ST-II. First, the implosion of messages at the sender causing "hot spots" is eliminated; second, the total network-wide protocol overhead is reduced. Figure 6 shows the total network-wide protocol overhead for ST-II and RSVP for various numbers of receivers joining the lecture.

[10] Clark [1] characterizes the fate-sharing model as gathering the critical state information at the end-point of the net, in the entity which is utilizing the service of the network. It is then acceptable to lose the state information associated with the entity, only if the entity itself is lost at the same time.

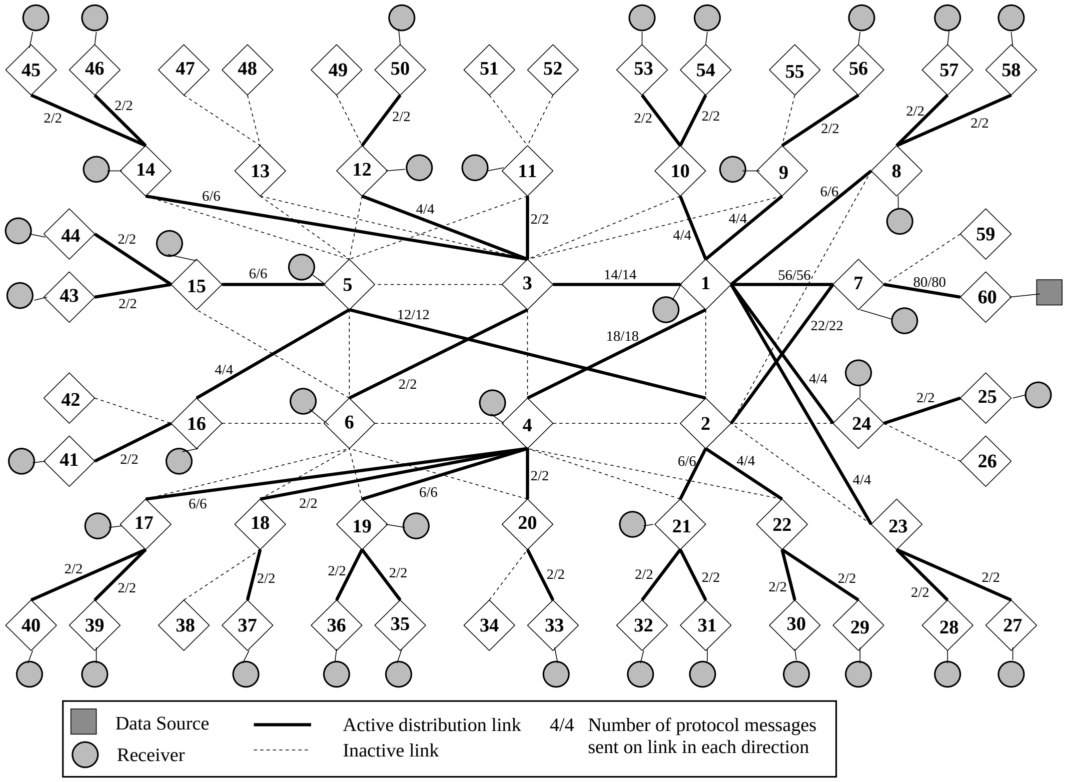

Figure 5: ST-II Protocol Message Overhead for Independent Joins (40 receivers)

The graph shows that the RSVP merging function is indeed highly effective in reducing protocol overhead; in fact RSVP becomes more efficient as the group becomes more "dense," as the average number of hops to splice into an existing distribution branch decreases.[11]

In addition to protocol overhead, another important measure of group dynamics support is the latency in reacting to group changes. RSVP latency can be "tuned" by adjustment of refresh timers, making direct comparison of latency times difficult; however, we can make some general observations regarding the two protocols. Under ST-II, the reservation setup and teardown times for a target are nominally one round trip time between source and receiver, and one end-to-end delay respectively. Latencies in RSVP are much less precise, adding a new receiver may involve an initial delay in waiting for a Path refresh, if the receiver is on a new branch in the multicast distribution tree; reservation setup time is also variable, from as little as one hop, up to an end-to-end delay, depending upon whether an existing reservation can be "spliced." When a receiver leaves, explicit reservation teardown may be possible for a dynamic-filter reservation; however, the non-reliable transport and lack of teardown support for wildcard and fixed-filter reservations[12] effectively makes the timeout delay the nominal case.

5 Summary and Future Work

We have described how the ST-II and RSVP protocols provide support for resource reservation establishment, in support of an Integrated Services Packet Network. Both protocols assume multicast datagram distribution, to improve network efficiency for multipoint communication. We have shown that incorporating application-level communication requirements in the reservation protocol can aid in reducing network

[11] Note that the current assumption of homogeneous receivers result in a best case scenario of one protocol message on each link in each direction; whereas the worst case is encountered in a heterogeneous environment when the receivers join in order from least demanding to most demanding, resulting in an overhead proportional to the number of downstream receivers on each link leading towards the sender, and one Path message on each link in the distribution tree.

[12] Explicit reservation teardown could be added to the protocol for these reservation styles, however it would require maintaining per-receiver state.

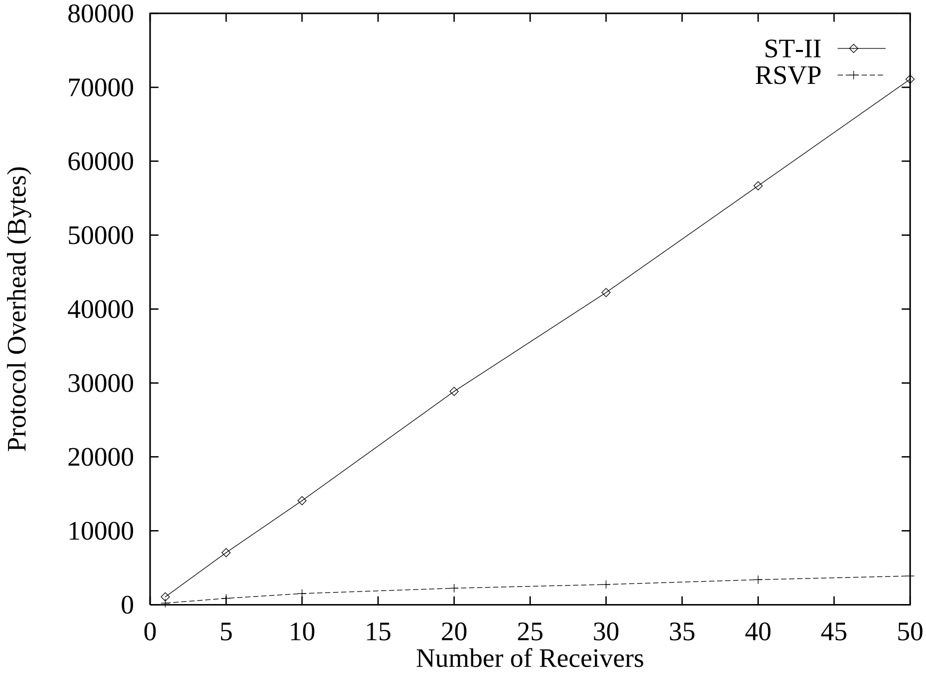

Figure 6: Comparison of Protocol Overhead for Group Joins

resource requirements, and improve scaling in terms of the number and size of groups that can be supported. The examples in Section 3 highlight RSVP's ability to incorporate a broader range of service models than ST-II.

Both ST-II and RSVP use timer based mechanisms to provide robustness in adapting to network dynamics; however, the design philosophies are quite different. ST-II requires that the network be responsible for correctness, leading to increased protocol complexity; RSVP uses a soft state mechanism, leaving end-systems responsible for refreshing state. We also showed that the receiver-initiated reservation and merging in RSVP reduces the bottleneck on links closer to the source, and reduces the network-wide protocol overhead, when compared to ST-II.

There are several features of RSVP that are currently not well understood, or that can be further improved to increase efficiency. RSVP related topics open for further investigation include:

- Channel selection is a new communication paradigm, and not well understood; what are the trade-offs between using the assured and non-assured mechanisms? What is the overhead of the dynamic filter reservation style; is it bounded for typical network topologies and group member distributions?
- Protocol overhead currently scales with the number of reservations; is it possible to further reduce this by aggregating refresh messages across groups?
- As noted in Section 4.1, a fault detection and refresh trigger mechanism could be incorporated into the protocol; how would this affect protocol complexity and recovery latency?
- Timer settings control adaptation latency, and have a large effect on protocol overhead; is it possible to dynamically adapt timers to measured network performance, to reduce protocol overhead?
- What additional service models are required to efficiently support future ISPN applications?

References

[1] Clark, D., "The Design Philosophy of the DARPA Internet Protocols," *Proceedings of ACM SIGCOMM '88*, August 1988.

[2] Deering, S., "Multicast Routing in a Datagram Internetwork," Technical Report STAN-CS-92-1415, Stanford University, December 1991.

[3] Forgie, J., "ST - A Proposed Internet Stream Protocol," Internet Experimental Notes IEN-119, September 1979.

[4] Partridge, C., Pink, S., "An Implementation of the Revised Internet Stream Protocol (ST-2)," *Journal of Internetworking: Research and Experience*, vol. 3, no. 1, March 1992.

[5] Topolcic, C., "Experimental Internet Stream Protocol: Version 2 (ST-II)," Internet RFC 1190, October 1990.

[6] Zhang, L., Braden, B., Estrin, D., "ReSource ReserVation Protocol (RSVP) - Functional Specification," unpublished draft, August 26, 1993.

[7] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D., "RSVP: A New Resource ReSerVation Protocol," to appear in *IEEE Networks Magazine*, September 1993.