An Architectural Comparison of ST-II and RSVP

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

[1]Computer Science Dept.
University of Southern California
Los Angeles, CA 90089
{mitzel, estrin}@usc.edu

[2]Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
{shenker, lixia}@parc.xerox.com

[3]Hughes Aircraft Co.
P.O. Box 92919
Los Angeles, CA 90009

[4]USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

Abstract

This paper presents a comparative analysis of two resource reservation protocols, ST-II [6] and RSVP [7], in support of an Integrated Services Packet Network (ISPN). We use simulations to examine 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 comparison 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 based on datagram packet delivery such as the TCP/IP protocol suite exhibit several distinctive characteristics. Datagram networks maximize network utilization by multiplexing multiple data streams, can provide multipoint communication, and provide robustness by adapting to network dynamics. However, datagram networks provide only a 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.

Initial work in supporting multicast 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 [6], which was specified as an experimental protocol within the Internet community. A more recent proposal targeted at supporting the resource reservation requirements of an ISPN is the RSVP protocol [8]. The RSVP protocol is currently in the design phase, an IETF working group has been formed to evolve the protocol along the standard track.

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, and look at the protocol overhead associated with accommodating group membership dynamics. Section 5 concludes with a summary and a few comments on future work.

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 the ISPN was to extend the current IP point-to-point service model with QoS support; however, the goal of the ISPN architecture is to provide efficient support for 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 communication 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 assumes they are provided by the underlying network. 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 into the ST-II protocol adds some processing overhead it does not affect the resource allocation or protocol messaging 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 into 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 emphasized 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 [6] 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 generates 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 or 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 defined by the protocol specification but is instead performed 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 are 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 acknowledgments 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 attempted.

The only service model directly supported by ST-II is that of a homogeneous reservation over a point-to-multipoint simplex distribution tree. We call this the *Independent Streams* reservation style; a separate and independent resource reservation is allocated for each distribution tree. The ST-II protocol specification defines the concept of a group of streams, which may be useful in defining more sophisticated reservation styles. Groups can be used to express relationships among individual streams or for performing operations on the group as a whole. However, the group mechanism is an 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

RSVP [7] 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 accommodate heterogeneous receiver requirements.

RSVP incorporates a datagram messaging protocol with periodic refreshes to maintain *soft state*[2] in intermediate switches 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 orphaned reservations and state to be automatically timed out and recovered.

RSVP models a reservation as two distinct components, a resource allocation and a packet filter. The resource allocation specifies *what amount* of resources

[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.

[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."

is reserved while the packet filter selects which packets can use the resources. This distinction between the resource reservation and packet filter, and an ability to change the packet filter without changing the resource allocation enables RSVP to offer several different *reservation styles*. A reservation style captures application-level communications requirements; these dictate how reservation requests from individual receivers should be aggregated inside the network. At the moment RSVP has defined 3 reservation styles, these are *Wildcard*, *Fixed Filter*, and *Dynamic Filter*; other styles may be identified as new multicast applications with different needs are developed. A Wildcard reservation indicates that a source specific reservation is not required and that 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. When a source specific reservation 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.[3] The Dynamic Filter reservation allows a receiver to modify its packet filter over time. This requires that sufficient resources be allocated to handle the worst case when all receivers take input from different sources.

3 Static analysis

The protocol descriptions in Section 2 noted that ST-II and RSVP are similar in that they model a data stream as a simplex point-to-multipoint distribution tree. However, the RSVP protocol incorporates heterogeneous receiver requests and multiple 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 service model of the two protocols to the application communication requirements and compare the network-wide resource requirements for supporting the application.

3.1 Supporting self-limiting applications

A number of multipoint-to-multipoint applications have application-level constraints that prohibit all

[3] Note that while the Independent Streams and Fixed Filter reservation styles result in equivalent reservations, we use distinct names to distinguish the mechanistic differences.

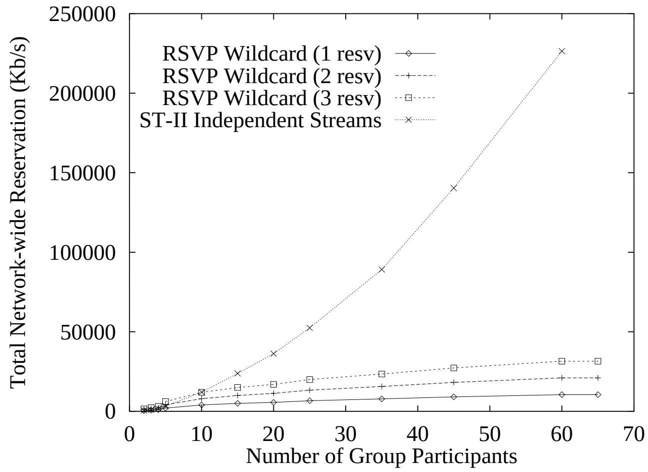

Figure 1: Resource requirements in support of n-way audio conference.

data sources from transmitting simultaneously; one example is an audio conference. In an audio conference there is typically only one person speaking at a time because when more than a few speakers are simultaneously active the result is usually unintelligible. Therefore, instead of reserving sufficient 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 communication requirements exactly using the Wildcard reservation and requesting resources for the maximum number of simultaneously active sources. ST-II requires that an Independent Stream reservation be established for each audio source.

In this section we compare the total network-wide resource allocation to support an n-way audio conference under the two reservation protocols. We model a hierarchical network containing 60 routers interconnected via 82 links and vary the number of conference participants from 2 up to 65. Each audio source was randomly distributed among the 60 nodes[4] 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 when each participant requests a reservation for 1, 2, or 3 voice streams worth of bandwidth; this represents the limit on the number of simultane-

[4] The random placement function used throughout the simulations selects 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.

ous speakers. For ST-II we show the resources reserved when each source establishes an Independent Stream to all receivers, and network links are modeled as having unlimited capacity.

The small slope of the RSVP plots highlights the efficiency in adding participants using the Wildcard reservation style. Resources are reserved only along new links required to "splice" into the distribution mesh.[5] As a group becomes more "dense" (the group membership covers a higher percentage of the total network nodes) the average number of new links required to splice into the distribution tree decreases, resulting in a smaller overhead per new member. In contrast, adding a participant under the ST-II model requires splicing into $N-1$ existing distribution trees and setting up an independent distribution tree from the new participant to all existing members. The disparity between the Independent Streams and Wildcard plots represents a resource over-allocation, which is shown to rapidly diverge as the group size increases.

Allocating an independent resource reservation for each ST-II source effectively places an upper bound on the maximum group size that can be supported. For a group of size N a participant must allocate $N-1$ reservations to receive from all sources. In the common scenario of a host at the network periphery with a single access link, all $N-1$ reservations must be accommodated on the same link. Making the optimistic assumption that the packet service algorithm can maintain QoS guarantees at 100% link utilization, the group size is thus limited to

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

participants. Repeating the ST-II simulations presented in Figure 1 with link bandwidth limited to 1.5Mb/s confirmed that resource allocation requests begin to be rejected (this is termed *call blocking* in the telephony literature) for group sizes greater than 24.[6] RSVP Wildcard reservations do not encounter this scaling problem. The maximum resource reservation across all links is limited to the number of simultaneous sources requested, which is independent of the size of the group.

One final observation is to note that the total resource requirements of the RSVP Wildcard reservation are bounded, while ST-II resource requirements are unbounded. Under RSVP once there is a participant at each network node, resources for the complete distribution mesh have been allocated and no further resources need to be allocated to accommodate additional group members. This is evident in the plots in Figure 1 by the zero slope line when going from 60 to 65 group participants. ST-II always requires allocating an independent reservation from the new participant 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 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 may then determine its QoS requirements based on local constraints.

ST-II and RSVP accommodate heterogeneity very differently. Under the ST-II service model a data source must view the entire stream as a homogeneous distribution path. After stream setup the source 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. This 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 listening to an audio lecture. The 60-node network introduced in Section 3.1 is used again and

[5] Note that the total resource allocation under Wildcard reservation is based upon the *union* of the links in all distribution trees, while it's based upon the *sum* under Independent Streams.

[6] $\left\lfloor \frac{1.5Mb/s}{64Kb/s} \right\rfloor + 1 = 24$.

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 in support of 40 receiver heterogeneous audio lecture.

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. Two alternatives for supporting this application are to send the "high quality" and "low quality" components on separate multicast trees, or to send the entire data stream over a single multicast tree. Sending the entire data stream on a single multicast tree and forwarding only the components required to satisfy all downstream receivers provides the most efficient support of the application. This is the model we investigate.[7]

Figure 2 shows the link reservations installed by RSVP to support 40 randomly selected receivers of the audio lecture using a Fixed Filter reservation, 20 receivers request the full 64Kb/s stream and 20 receivers request the 16Kb/s "low quality" audio subchannel. This diagram depicts RSVP's ability to install a heterogeneous resource reservation across the data distribution tree. 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 a 27.7% 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 audio lecture, 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 RSVP the total network-wide resources reserved reflects the minimum allocation required along the distribution tree to satisfy all receivers QoS requests. As the number of low quality receivers increases additional branches in the distribution tree shed their high quality resource reservation resulting in a gradual decrease in total network-wide resource allocation.[8]

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

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 does not provide such a guarantee, and a change request may be denied.

The traditional method to provide assured channel selection is to allocate an independent reservation for each source, which is just the Independent Streams reservation style discussed in Section 2.1. The receiver can then switch between channels by selecting the desired incoming stream. The channel selecting, or filtering of incoming data, is done entirely at the receiver.

RSVP introduces the *Dynamic Filter* reservation style, which allocates sufficient resources on each link so that the receiver can always select, without failure, any set of m sources (where m is the maximum number of simultaneous sources). Once the resource allocation is fixed a receiver may dynamically modify the associated *filter*, which chooses which packets get to use that resource. Thus, the filtering is done within the network. The actual resource allocation on each 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.

The third channel selection alternative is to make a new reservation every time a new channel is selected (and then to tear down the old reservation). This

[8]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.

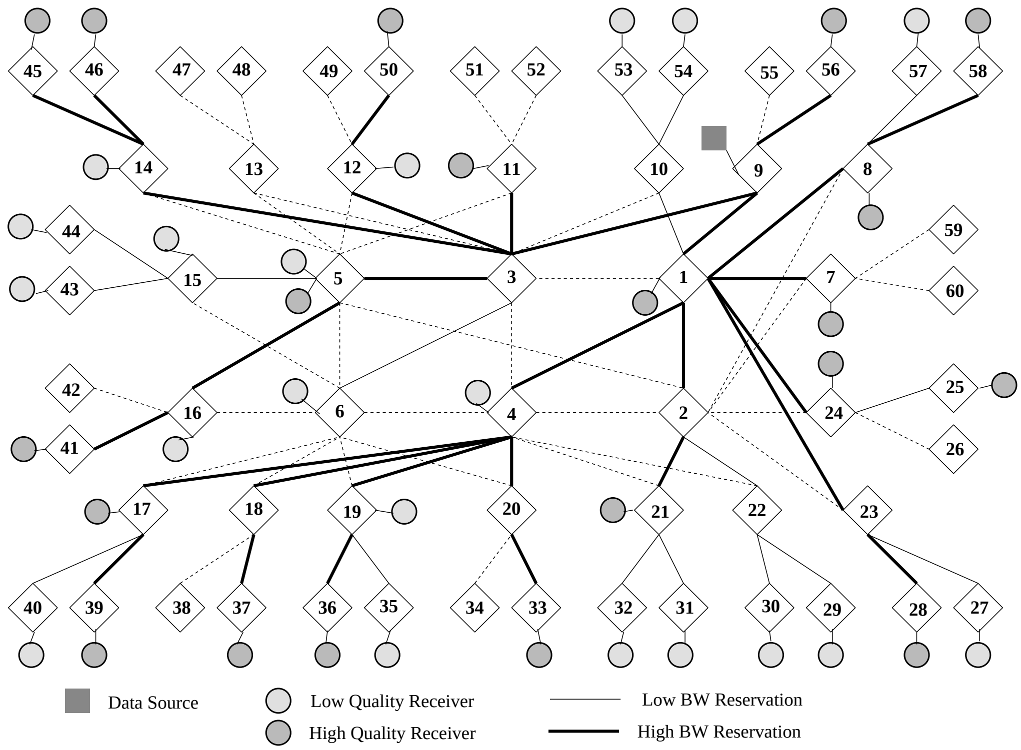

Figure 2: Link reservations for RSVP heterogeneous audio lecture (40 receivers).

provides the non-assured service because the new request may be blocked. We call this the *Chosen Source* reservation style since it only reserves for the currently chosen sources. Resources are reserved along the distribution tree from each source to the set of receivers that are currently tuned into that source, and the trees from different sources are independent. Because the Chosen Source reservation style reserves for only the currently selected sources it provides a useful lower bound for the resource consumption required by assured service.

Table 2 presents the total network-wide resource allocation required by each of the channel selection mechanisms to support the participants of the n-way conference introduced in Section 3.1. While the Chosen Source reservation style does not provide assured switching among the sources, it is presented to quantify the overhead in the assured channel selection schemes, as indicated in the "Overhead Ratio" column. For the simulations conducted, the resource overhead incurred using the Independent Streams mechanism can be quite substantial and it increases 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.[9]

The only requirement on the reservation protocol to support the Independent Streams or Chosen Source 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 distinction is currently provided only by the RSVP Dynamic Filter reservation style.

4 Dynamic analysis

In Section 3 we compared the resource allocations 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

[9] Analysis shows that the Dynamic Filter reservation style uses exactly the same resources as the worst case of the Chosen Source reservation style, and appears to be only a constant factor worse than the average case of the Chosen Source reservation style [4].

Group Size	Chosen Source (4 Reservations) Resource Allocation(Kb/s)	Dynamic Filter (4 Reservations) Resource Allocation(Kb/s)	Dynamic Filter (4 Reservations) Overhead Ratio	Independent Streams (N-1 Reservations) Resource Allocation(Kb/s)	Independent Streams (N-1 Reservations) Overhead Ratio
5	3200	3200	1.00	3200	1.00
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.

protocol. These events include both network dynamics such as link and router failure/recovery and group membership dynamics (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. The 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, and these add considerable complexity to the protocol.[10] In contrast, RSVP relies on the soft state refreshes to automatically adapt without additional protocol complexity. RSVP could be modified to incorporate a failure detection mechanism to trigger refreshes as an optimization; however, there are more fundamental differences that distinguish the protocols. The key difference between the two 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” among the end-points.[11]

Note 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 Path and Reservation refreshes. This would seem 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 reser-

[10] Partridge and Pink [5] note that much of the functionality is overlapping.

[11] 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 if, and only if, the entity itself has failed at the same time.

vation for each source forwarding along a link, while a Dynamic Filter requires a separate reservation per receiver (bounded by the total number of upstream sources). Thus, RSVP protocol overhead scales with the number of reservations.

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. 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.

Dynamic addition of receivers under ST-II requires the generation of Connect and Accept messages between source and receiver. The end-to-end messaging of ST-II results in an overhead on each link proportional to the number of downstream receivers. This results in links closer to the source becoming "hot spots," in that they incur a higher overhead in terms of bandwidth and protocol processing overhead. Also, the explicit source interaction required for every group membership dynamics could result in a processing bottleneck at the source.

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 3 shows the total network-wide protocol overhead for ST-II and RSVP for various numbers of homogeneous receivers independently joining the audio lecture first described in Section 3.2. This 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" due to the average number of hops to splice into an existing distribution branch decreases.[12]

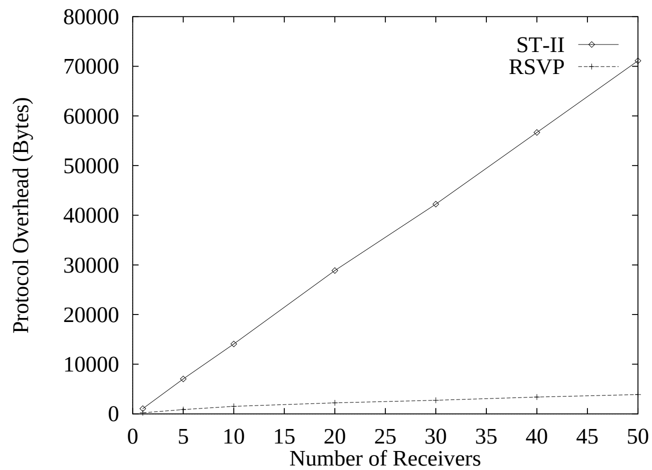

Figure 3: Protocol overhead for independent group joins for audio lecture.

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."

[12] Note that the current assumption of homogeneous receivers result in a best case scenario of one protocol message on each link in each direction. 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.

When a receiver leaves, explicit reservation teardown can release the resources immediately.

5 Summary and Future Work

We have described how the ST-II and RSVP protocols provide resource reservation establishment in support of an Integrated Services Packet Network. Both protocols utilize multicast data distribution to improve network efficiency for multipoint communication; however, we argue that a richer service model is required for the ISPN environment. Our simulations show that RSVP's support for heterogeneous receiver requests and multiple reservation styles can be exploited to obtain significant improvements in network-wide resource allocation for several common applications. If these application classes make up a significant fraction of the resource demands in an ISPN, then incorporation of RSVP could result in a substantial reduction in network resource requirements and improve scaling in terms of the number and size of groups that can be accommodated.

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 load on links closer to the source, reduces source-receiver interactions, 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 data sources; 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 reservation styles 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] Mitzel, D., Shenker, S., "Asymptotic Resource Consumption in Multicast Reservation Styles," unpublished preprint.

[5] 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.

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

[7] Zhang, L., Braden, B., Estrin, D., Herzog, S., Jamin, S., "ReSource ReserVation Protocol (RSVP) - Functional Specification," Internet Draft, March, 1994.

[8] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D., "RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine*, September 1993.