An Architecture for Advance Reservations in the Internet

Steven Berson

Robert Lindell

Robert Braden

USC Information Sciences Institute

July 16, 1998

Abstract

Much work has been done on establishing resource reservations over the Internet [3, 2, 8], but almost all of this work has consisted of establishing a reservation that begins immediately and has indefinite duration. An important issue is how to request an *advance reservation*, a reservation made now for resources that are to be reserved in the future. In this paper, we propose an architecture for Advance Reservation Servers (ARS), and describe their operation. By using ARS, no reservation state or scheduling state needs to be set up in the routers until the reservation is established. No multicast routing state (for multicast sessions) needs to be set up, either, until the reservation is established.

1 Introduction

The IETF has been defining Quality of Service (QoS) models [20, 16] and a setup protocol, RSVP [3], to achieve integrated services in the Internet. The work to date has been primarily concerned with setting up *immediate reservations*, that is, the QoS takes effect immediately and remains in effect for an indefinite duration. There are a variety of circumstances in which users will want to

Ex.1030
CISCO SYSTEMS, INC. / Page 1 of 21

make an *advance reservation*, reserving a QoS now that will take effect in the future. One of the most important purposes of advance reservations is to allow a group of people to have a "virtual meeting" such as for a video conference. Other potential applications of advance reservations include remote lectures, multi-player games, and multi-person visualization. In all of these applications, the larger the group, the more important it is to make an advance reservation to ensure that the required QoS will be available for all users when needed. Therefore, we expect advance reservations to be of particular importance for multicast applications in addition to unicast applications such as Internet telephony, or remote monitoring.

One approach to advance reservations is to augment RSVP reservation messages with a start time and a duration. Ordinary reservation messages (i.e. no start time and no duration) would be assumed to start immediately and have an indefinite duration. The RSVP advance reservation signalling mechanisms would be a relatively straightforward modification of existing RSVP.

There are three main concerns with using an augmented RSVP mechanism for advance reservations, which will be discussed below.

1. The amount of state in routers is inflated.

2. The sending and receiving applications need to be active during the entire advance reservation time.

3. The senders and receivers must be known in advance.

The first issue is the amount of state in routers due to advance reservations. State is required in routers to maintain the knowledge of which flows have reservations. In addition, multicast routing state is also required for multicast sessions, since multicast routing state is created on demand. The amount of state can be roughly estimated as follows. Assume the average session duration is 30 minutes, that users schedule one week in advance, and that 10 hours per day are scheduled. If we

Ex.1030
CISCO SYSTEMS, INC. / Page 2 of 21

assume that only 10 percent of reservations are made in advance, then the amount of reservation and routing state in routers would be increased by a factor of 10. To reduce the amount of state in routers for advance reservations, we propose an architecture for off-loading the advance reservations from routers into Advance Reservation Servers (ARS). This architecture provides for a centralized ARS for each Autonomous System (AS) or domain. All advance reservation data for a domain is stored at the ARS. Resources and state in the routers are only reserved at (or shortly before) the time the reservation is scheduled to take effect.

The second issue is that applications need to be active during the advance reservation interval. Normally with RSVP, an application runs the entire time a reservation is in effect. It would be unrealistic to require an application to run for the entire time an advance reservation is in effect. Having a server-based architecture solves the problem with keeping an application running. The sender or receiver registers the advance reservation with the ARS. The ARS keeps the advance reservations active and ensures that actual reservations can be established at the advance reservation start time.

The last issue is that the session participants must be known in advance. There are two ways that participants in a session become known, either some site lists all the participants, or the participants make themselves known by registering with an advance reservation server. Once an application has registered with an ARS, information about the participant is distributed to other servers participating in the session.

Instead of participants announcing themselves, each registering with their ARS, one participant may list all senders and receivers of a session. If all participants in a session are explicitly listed by one of the participants in the session for an advance reservation, the local ARS will distribute that information to all the ARS's on the tree. Note that the sender and/or receiver addresses could be CIDR blocks of addresses.

Ex.1030
CISCO SYSTEMS, INC. / Page 3 of 21

A potential receiver for an advance reserved session that has neither registered with their ARS, nor been announced by another participant may still attempt an immediate reservation when the reservation becomes active. If there are sufficient resources available, the receiver will be able to make the reservation. And even if there are not sufficient resources, the local advance reservation management policy might allow late arriving receivers of an advance reserved session, preemptive priority over immediate reservations. This is a policy issue, but the architectural mechanisms need to be provided to deal with this case.

1.1 Inter-Domain Multicast Routing

Advance resource reservations are closely related to Internet routing since the reservations need to be established where the traffic will be flowing. Current Internet multicast routing algorithms [5, 18, 1, 12] have problems with scaling and with intra-domain multicast protocol independence. Newer hierarchical multicast routing schemes [11, 17, 10] are being developed which have better scaling properties and allow the lower level of the routing hierarchy to independently choose an interior multicast routing protocol for inside their domain. The IETF is currently developing MASC/BGMP (Multicast Address-Set Claim/Border Gateway Multicast Protocol) as the inter-domain multicast routing protocol for the Internet. While our advance reservation server architecture would work with other inter-domain multicast routing schemes, we will concentrate on MASC/BGMP in this paper.

MASC/BGMP allocates blocks of multicast addresses to domains. These blocks of addresses are then advertised using multicast extensions to BGP called MBGP. Sites select multicast addresses from the address block assigned to the site's domain. The domain containing the multicast address that is assigned is called the root domain for that multicast address. Using MASC/BGMP, an inter-domain route for any multicast address can be found from any domain towards the root domain

Ex.1030
CISCO SYSTEMS, INC. / Page 4 of 21

of the multicast tree. Since BGP mechanisms are used, the routes generated are guaranteed to be free of loops.

MASC/BGMP routes use a bi-directional tree. The bi-directional tree means that the links used to receive data are the same links used to send data. A bi-directional tree simplifies advance reservations since the reservations can be made by the sender or the receiver, as the reverse route (e.g. from the receiver towards the sender) is the same as the forward route. Note that with unicast routing the forward routes can be different than the reverse routes. As will be seen in section 2.2, the asymmetry in unicast routing requires some extra mechanism for creating advance reservations.

1.2 Admission Control for Advance Reservations

We assume that there is some amount of resources that can be reserved in advance. These resources may have policy restrictions that allow different individuals to reserve differing amounts of resources. The available amount of resources and the related policy restrictions will typically be administratively configured. We assume that the ARS knows the topology and capacity of the links of the domain (e.g. by participating in the intra-domain routing protocol or by manual configuration). From the intra-domain routing and capacity information, and from the existing set of advance reservations at the ARS, an admission control decision can be made for a newly arriving advance reservation. The advance reservation admission control decision is made by looking if there are sufficient resources for the new reservation during the interval in which the new reservation will be active. From the policy information, the ARS can determine whether the application requesting the advance reservation has the permission to request an advance reservation for these resources. If both the admission control and the policy control succeed, then the ARS accepts the advanced reservation. In spite of the advance reservation, there is a possibility that when an advance reservation is scheduled to begin, there will not be sufficient resources available. While it is a policy

Ex.1030
CISCO SYSTEMS, INC. / Page 5 of 21

decision what action to take in this case, a typical policy would be to preempt one (or more) of the immediate reservations to allow the advance reservation to acquire the reserved resources.

Since immediate reservations are likely to be preempted if resources are overbooked, immediate reservations may need to account for upcoming advance reservations. The accounting for advance reservations can either be done by statically partitioning the amount of resources that can be reserved in advance or by having the immediate reservation system (e.g. RSVP) periodically exchange load information with the ARS. This exchange of load information can take the form of a direct inquiry from the RSVP daemon to the ARS or a multicast load status report from the ARS to all the RSVP daemons.

2 Advance Reservation Architecture

The architecture we propose is both server based and routing domain based. A server based architecture allows advance reservations to be made without using any reservation state in the routers. A server based architecture also allows applications to request an advance reservation and then exit, rather than requiring the application to run during the entire advance reservation period. Once the ARS sends a confirmation message for the reservation, the ARS will hold the reservation until the reservation is cancelled or expires.

We propose an architecture that is based on domains as the domain model is used by inter-domain routing, to which advance reservations are closely tied. There will be an Advance Reservation Server (ARS) in each routing domain, and the ARS will participate in routing protocols. Thus an ARS will know inter-domain routes for both unicast and multicast traffic.

Figure 1 shows our advance reservation server architecture. Each domain has one (possibly replicated) ARS, and each ARS communicates with the ARS in each adjacent domain. The ARS has three basic components, advance reservation policy, advance reservation admission control, and

Ex.1030
CISCO SYSTEMS, INC. / Page 6 of 21

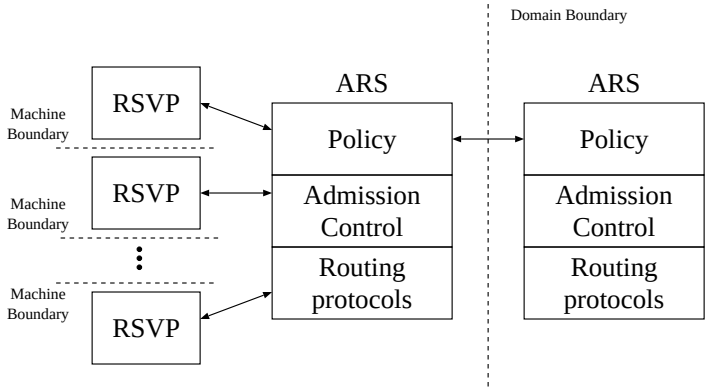

Figure 1: Advance Reservations architecture

routing protocols. The policy module describes which users are allowed to make advance reservations and for how much resources. The policy module determines whether a new reservation is permitted according to the policy rules in force. The admission control module stores the times that advance reservations will be active and the resources the reservations will need. An admission control decision for a new reservation is made by checking if the resources needed for the new reservation will be available when the new reservation will be active. The routing protocols module participates in both the interior (intra-domain) routing protocols and the exterior (inter-domain) protocols. The ARS acquires domain topology and link capacity information from the interior routing protocols, and inter-domain multicast routing information from the exterior multicast routing protocols. The inter-domain multicast routing information allows an ARS tree to be built for a session. The ARS tree is the inter-domain tree that contains all ARS's in whose domain a session participant is found along with the ARS's in transit domains. The ARS tree allows information to be distributed to the ARS's for all the participants and for all transit domains.

Each router in the domain has a server for QoS signalling and scheduling, such as RSVP (shown in the figure). The interactions with RSVP (or other signalling and scheduling code) have two main functions. The RSVP module does immediate admission control for which near-future load information from the ARS would be useful. The ARS supplies this information either in response

Ex.1030
CISCO SYSTEMS, INC. / Page 7 of 21

to a direct inquiry or by multicasting the information at regular intervals. RSVP must also verify whether an arriving reservation has an advance reservation, as a valid advance reservation may be admitted where an ordinary reservation would be rejected. To verify an advance reservation, RSVP sends the session information to the ARS, which responds with a confirmation if a valid advance reservation was in place for this session.

Unless otherwise noted, we will assume that all domains are running an ARS. Further we assume that the ARS in one domain knows the address of all the ARS's in neighboring domains. We assume that there is reliable communications (e.g. TCP) between the ARS of neighboring domains. A later section 3.2 describes relaxing these assumptions.

2.1 Multicast Advance Reservations Signalling

In our architecture, an application chooses a multicast address, for which an advance reservation is desired. The application can then register a list of senders and receivers for that session. In order to establish the advanced reservation, the ARS in the domain of each sender and receiver and the ARS in each transit domain needs to set up advanced reservation state. In order to set up the reservation state, a tree of all participating ARS's is built. Since in BGMP, the tree is built from bi-directional paths from the leaf domains towards the root, the ARS tree needs to be built in similar fashion. Thus the tree needs to be built from the leaves towards the root.

To build the tree, the originating ARS needs to establish a path of ARS's to the root domain. After establishing the ARS path to the root, the ARS's in the domains of all the other sender and receivers need to be notified. Finally, all the other ARS's establish paths to the root. The collection of all of the ARS paths to the root is the ARS tree.

It is often trivial to establish a path from the originating ARS to the root ARS as the originating ARS is typically in the root domain. To contact the ARS of a sender or receiver from the root

Ex.1030
CISCO SYSTEMS, INC. / Page 8 of 21

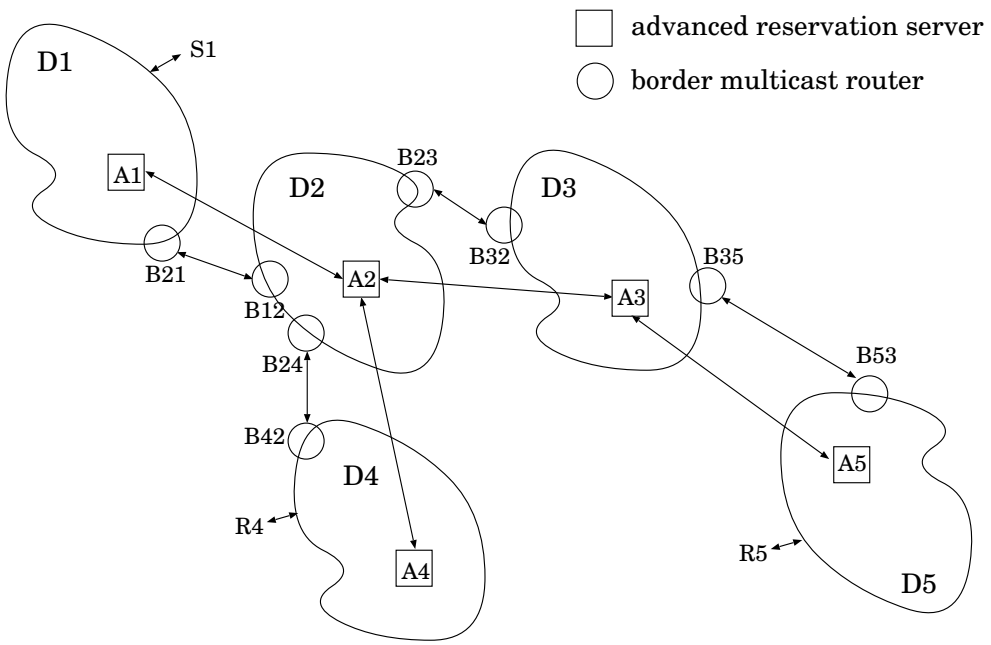

Figure 2: Multicast advance reservations signalling

domain, it is necessary to use inter-domain unicast routing to find the next domain on that path. Each ARS know the identity of the ARS in each adjacent domain, so the ARS can forward the message to the next ARS on the path to the sender or receiver. Each ARS continues to forward in this fashion until the message reaches the ARS of the sender or receiver. The final part of the construction of the ARS tree is for the sender or receiver to use BGMP to build the tree back towards the ARS in the root domain. The branch of the tree ends either when it meets an existing branch of the tree or when it meets the root domain. Once all of the branches are complete, the ARS tree for this session is complete.

An alternative way to determine the participants in a session is to distribute a session address out-of-band (e.g. using a session manager like sdr) to potential participants. Interested participants can contact the ARS in their domain to make an advance reservation. In this case, the procedure for building the ARS tree consists only of establishing a path either to the ARS in the root domain or to the first ARS already on the tree.

Ex.1030
CISCO SYSTEMS, INC. / Page 9 of 21

The operation of the advance reservation multicast signalling architecture is shown in figure 2. There are five domains in the figure represented as D1-D5. In each domain, there is an advance reservation server represented as A1-A5. The border multicast routers are labeled as Bxy, where x and y are the domains which the border routers connect. Thus, B35 is in domain 3 and connects to domain 5.

Consider a new multicast session, and assume that D3 is the root domain. A sender S1 in domain D1 initiates a reservation by contacting ARS A1 with the session parameters, the reservation parameters, and the list of receivers (R4 in D4 and R5 in D5). ARS A1 first does admission control and policy control. Assuming success, A1 needs to establish an ARS path to the root ARS. BGMP gives the next domain as D2, and A1 knows the identity of ARS A2 in domain D2. A1 forwards the message to A2 which does admission control and policy control before accepting the reservation. Similarly, the message arrives at the root ARS, A3.

The second phase of the ARS tree construction is for the root ARS to notify all the leaf ARS's. ARS A3 uses unicast routing to route towards D5, the domain of the receiver R5. Since D5 is adjacent, A3 just sends the message to A5. Similarly, A3 forwards the message using unicast routing to A2 and then A4 (the domain of receiver R4).

The final phase is building the tree back towards the root ARS from the leaves while making the advance reservation. A5 does admission control and policy control and then forwards the message to A3 completing a branch of the tree. A4 does admission control and policy control and forwards the message to A2. A2 does admission control and policy control for the resources between D2 and D4. The ARS tree is now complete as A2 is already part of the tree on the branch from the root A3 to A1.

Another model for establishing advance reservations is to let individual senders and receivers join a session. This case is simpler than the case where a participant lists all senders and receivers,

Ex.1030
CISCO SYSTEMS, INC. / Page 10 of 21

as only the second and the final phase of the ARS tree building need to be completed.

2.2 Unicast Advance Reservations Signalling

Inter-domain multicast routing creates a shared, bi-directional routing tree. Since the tree is bi-directional, the reverse routes (i.e. routes from the receivers toward the senders) are known, and advance reservations state can be installed at the ARS's hop-by-hop from the domain of the receiver(s) to the root domain.

Inter-domain unicast routing, by contrast, uses uni-directional routes, and so the reverse routes are not known. In order to support advance reservations for unicast routes, the advance reservation messages must go from the domain of the sender to the domain of the receiver, i.e. follow the route of the data traffic.

Figure 3 illustrates the flow of messages. Assume receiver r in domain D3 makes a unicast reservation for a sender s in domain D1. As in multicast ARS tree building, A3 uses unicast inter-domain routing to find the next domain, and then A3 uses its knowledge of ARS's in adjacent domains to send the message to A2 and then to A1. A1 does admission and policy control as the first domain on the route from s to r. A1 sends the message to A5, the ARS in the next domain towards r. A5 does admission control and policy control and forwards the message to A4, and then similarly to A3. The advance reservation is now set up at ARS's A1, A5, A4, and A3.

2.3 Relation of Immediate Reservations to Advance Reservations

Immediate reservations are under the control of the user application. When the application begins, it requests a reservation. As part of the admission control at a node, the node sends the session information from the reservation to the ARS. The ARS will verify whether this session had an advance reservation or not. If an advance reservation is in place, then the reservation will (typically)

Ex.1030
CISCO SYSTEMS, INC. / Page 11 of 21

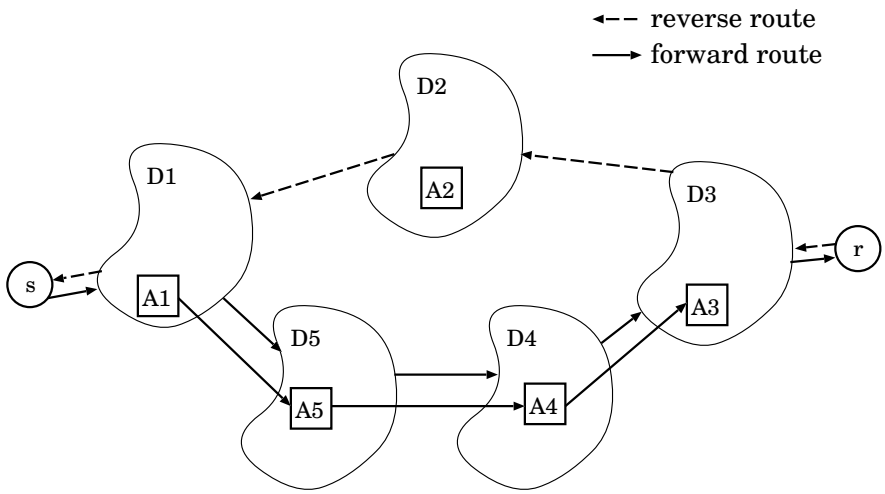

Figure 3: Unicast advance reservations signalling

be accepted even if other reservations need to be preempted.

The advance reservation state stays in place on the ARS until the advance reservation expires (i.e. start_time+duration). It is important that the state stay in place during the entire reservation period, so that the advanced reservation does not get preempted by a newer advanced reservation. After the advance reservation state expires, a former advance reservation will be eligible for preemption.

3 Advance Reservation Server Functionality

In this section we describe the major functions of an advance reservation server. We start by outlining the procedure that servers use to discover the identity of servers in other domains. Some domains may not have a server, either due to failure or non-participation. We discuss how the advance reservation system would gracefully operate in the presence of this break in service. We also discuss the issues of redundant servers and the recovery process after a failure. We then conclude the discussion with a description of the application interfaces for the advance reservation service.

Ex.1030
CISCO SYSTEMS, INC. / Page 12 of 21

3.1 Advance Reservation Server Discovery

Providing end-to-end advance reservations requires coordination between servers within the domains transited by the data path. In order to coordinate reservations, servers must know information about the other servers maintained by neighboring domains. To simplify administration, we propose a solution that provides automatic discovery of neighboring servers. We will also discuss the issue of neighboring domains which do not maintain an advance reservation server in a subsequent section.

All advance reservation servers listen for and periodically announce information to all other servers using a well known multicast address. Each server can independently estimate the number of servers, based on received announcements, and can adjust the announcement rate to keep the network overhead for server discovery at a constant value. Servers during initialization or restart should start immediately sending announcements at an initially higher periodicity to speed up the discovery process for other domains. We do not assume that the announcements are sent using a reliable transport protocol.

The advance reservation server announcement contains information about the servers unicast address and the domain that it represents. In addition, it announces both the unicast and multicast address prefixes which are assigned to that domain. Interdomain routing protocols, such as BGP and MBGP, also exchange domain and address prefix information. We choose to redundantly exchange this information as part of the server discovery process due to subtle address prefix aggregation issues. In BGP, for example, it is possible for a router to combine adjacent address spaces and drop the identity of the subdomains in the routing announcement.

An advance reservation server then combines the information obtained from advance reservation server announcements with the information about adjacent domains obtained from routing. The server identifies all adjacent domains in which it does not know the identity of the server. If

Ex.1030
CISCO SYSTEMS, INC. / Page 13 of 21

no information can be obtained about a server for a given adjacent domain, the first advance reservation server along the path from the current domain to the destination for a given address prefix is computed using the procedure described in the next section.

An ARS server must participate in the server discovery process. This does not preclude the server from maintaining additional information about other servers using a manual configuration process. The manually configured data may also be used to provide sanity checks for the information collected during the discovery process to protect a domain from malicious actions.

3.2 Domains Without Advance Reservation Servers

As previously mentioned, it is possible that a domain does not have an advance reservation server due either to failure or non-participation in the service. The server discovery process will identify for a given domain all adjacent domains which do not have servers. It is important that the advance reservation service continues to function in the presence of non-participating domains. In addition, it would be useful to report such breaks in service to the application requesting an advance reservation as real reservations may be rejected by these non-participating domains.

When a server detects that an adjacent domain does not support advance reservations, there are two choices. One approach would be to require manual configuration of the first participating domain along the data path towards the destination address range. The second approach would be for the advance reservation server to use both information from routing and the server announcement information to construct routes in an effort to determine a participating domain. While an automated solution seems preferable, there are some subtle issues related to both prefix aggregation and policy routing. At any given domain, the inter-domain routing information may be incomplete with respect to individual domain routing strategies. Although it is possible to generate a reasonable route to a destination domain, it may differ from the actual path that would be traversed by

Ex.1030
CISCO SYSTEMS, INC. / Page 14 of 21

data packets. Computing similar, but not exact routes, will lead to a higher rate of failure when a reservation is attempted.

3.3 Server Redundancy

A domain may choose to run multiple redundant servers for reliability. This not only increases the availability of the server for applications making advance reservations, but is important since routers within a domain need to contact the server for assistance in making admission control decisions. In our solution, we make no distinction between primary and secondary servers. Each server is capable of providing advance reservation services and in turn shares it's state with the other servers in the same domain. If a domain configures multiple servers, the server discovery advertisements will contain a list of all servers in the domain. Other domains freely choose any server in the list when contacting another advance reservation server. Improved load balancing would be achieved if servers are chosen in round robin from the list each time they need to contact the server in a given domain.

3.4 Server Recovery

Servers maintain state for both intra-domain and inter-domain advance reservations. Since intra-domain reservation state is never shared with neighboring domains, it is very important that the server state be saved to stable storage. If a server fails and subsequently restarts, it will use common database recovery mechanisms to restore it's entire state from stable storage. In addition, if multiple servers are running in the domain, the server will immediately contact them and load all state information that was missed during the failure-restart period. If done properly, a redundant server solution could relax the strict need for recoverable stable storage. Servers could commit transactions to other servers and then use them for recovery purposes. The assumption is that not

Ex.1030
CISCO SYSTEMS, INC. / Page 15 of 21

all servers would fail simultaneously.

If a server failure eliminates all service within a domain, that domain will appear to be non-participatory in the advance reservation process. Subsequent inter-domain signalling of advance reservations will bypass that domain. Once service is reestablished within the domain, it's servers presence will be discovered in adjacent domains by the server discovery process. Independently, adjacent servers will attempt to coordinate existing advance reservations which transit the domain which previously did not support advance reservations services due to failure of the server. During this recovery process, these additional servers are informed of the existing advanced reservations which are transiting their domain. They need to perform admission control and determine whether there is adequate capacity. If there is not, that server would generate an admission control failure. This failure may occur quite some time after the original reservation request. This can present difficulties to applications since they may have ceased execution after obtaining the initial reservation confirmation. An application which receives a confirmation for an advance reservation but is informed of a break in service, should be prepared to expect a possibly delayed reservation failure notification. This may have implications on whether the application can exit immediately after receiving the initial confirmation.

3.5 Server Application Interface

Advance reservation servers will support an application interface for clients. Once a reservation is made, servers will retain the information in hard state and allow the initiating application to cease execution if desired. An application can subsequently modify or delete the reservation request. In addition, the server would provide status information, when appropriate, about a reservation request with the application.

Clients will request reservations by specifying a flow specification, reservation style, sender

Ex.1030
CISCO SYSTEMS, INC. / Page 16 of 21

and receiver lists, and the start time and duration of the reservation. The style would describe whether the flow specification distinctly describes each sender or is a shared resource specification. Reservations can be initiated by both senders and receivers.

As a final note, we assume that the existing base of RSVP aware application software will continue to grow. For this reason, we believe that it will be important to continue supporting an RSVP interface for both immediate and advance reservations. It is relatively straightforward to augment the RSVP protocol with temporal attributes. In addition, the existing RSVP application programming interface could be slightly modified to support advance reservation semantics. The difficult step is the interface between RSVP and the advance reservation server architecture. For unicast sessions, the first RSVP capable host or router along a path which received an RSVP message with temporal attributes would recognize the advance reservation request and forward the RSVP message to the server for processing. This would occur transparently to the application. Once at the advance reservation server, the request is now forwarded from server to server using unicast inter-domain routing information. At the last domain in the path, the server would reinject a corresponding RSVP message to be delivered to the destination address. The multicast case is much more difficult. Our solution avoids creating multicast routing state for the session at the time of the advance reservation. The lack of multicast routing information in the destination domains makes it impossible for the advance reservation servers to reinject an RSVP message to be received by the destination hosts.

4 Related Work

Proposals to add advance reservation support to the existing Internet Integrated Services Architecture were described in the RBONE presentation [4]. Early experiments with both Integrated Services and the emerging MBONE helped to realize the shortcoming of the current architecture

Ex.1030
CISCO SYSTEMS, INC. / Page 17 of 21

at that time. The issues raised by this early effort identity key problems, issues, and solutions and helped to shape the subsequent research in this area.

The work at the IBM European Networking Center on advanced reservations [19] lays the groundwork for many of the important issues that an advance reservation system must address. We attempt to address most of these issues in our architecture and emphasize the added complexity required to support multicast. Further work by that group described extensions for ST-2 to support advance reservations [13]. The approach was very similar to the proposed temporal extensions of RSVP. While these approaches provide a simple and straightforward extension to an existing immediate reservation system, it ignores the concerns of greatly inflating the amount of state that must be managed by routers.

The work of Schelen and Pink, et al [6, 7, 14, 15] relates most closely to the work presented in this paper. They propose a server based approach that is domain based. For multicast reservations, servers communicate with other domains assuming that multicast routing has been installed in routers. We relax this assumption to avoid burdening routers with additional routing state until the starting time of the reservation. Instead, we assume the existence of an inter-domain multicast routing protocol, such as BGMP, which provides use with information to construct the distribution trees. We also address issues that arise with signaling over asymmetric routes between two domains.

The work of the Tenet project [9] demonstrated that effective utilization of the network can be achieved using a movable partitioning between advance reservation and immediate reservation resources. Thus, it is feasible to design an architecture for advance reserverations mostly independent of the immediate reservation system, needing only infrequent coordination between them.

Ex.1030
CISCO SYSTEMS, INC. / Page 18 of 21

5 Conclusions and Future Work

In this paper, we have described an architecture for Advance Reservation Servers. By using a server based architecture, no reservation or multicast routing state is needed in the routers until the reservation becomes active. A server based architecture also allows applications to request advance reservations without the application running during the length of the advance reservation. Our architecture is domain based, which allows simple functioning with inter-domain routing which is also domain based. We are in the process of implementing a system based on this architecture, and expect to have a prototype soon.

References

[1] Ballardie, A., Francis, P., Crowcroft, J., "Core Based Trees," SIGCOMM '93.

[2] Banerjea, A., Ferrari, D., Mah, B., Moran, M., Verma, D., Zhang, H., "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences,' ' IEEE/ACM Transactions on Networking, February 1996.

[3] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205.

[4] Braden, R., Casner, S., "RBONE - Resource Administration for the MBONE," IETF April '95 Proceedings.

[5] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., Wei, L., "An architecture for wide-area multicast routing," SIGCOMM '94.

[6] Degermark, M., Kohler, T., Pink, S., Schelen, O., "Advance Reservations for Predictive Service," NOSSDAV '95.

[7] Degermark, M., Kohler, T., Pink, S., Schelen, O., "Advance Reservations for Predictive Service in the Internet," ACM/Springer Journal of Multimedia Systems, May '97.

[8] Delgrossi, L., Berger, L., "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+," RFC 1819.

[9] Ferrari, D., Gupta, A., Ventre, G., "Distributed advance reservation of real-time connections," NOSSDAV '95.

[10] Handley, M., Crowcroft, J., Wakeman, I., "Hierarchical Protocol Independent Multicast," Unpublished Draft.

[11] Kumar, S., Radoslavov, P., Thaler, D., Alaettinoglu, C., Estrin, D., Handley, M., "The MASC/BGMP Architecture for Inter-domain Multicast Routing," SIGCOMM '98.

[12] Moy, J., "Multicast Extensions to OSPF," RFC 1584.

[13] Reinhardt, W., "Advance Reservation of Network Resources for Multimedia Applications," 2nd International Workshop on Advanced Teleservices and High-Speed Communications Architectures (IWACA).

[14] Schelen, O., Pink, S., "Sharing Resources through Advance Reservation Agents," IWQoS '97.

[15] Schelen, O., Pink, S., "An Agent-based Architecture for Advance Reservations," LCN '97.

[16] Shenker, S., Partridge, C., Guerin, R., "Specification of Guaranteed Quality of Service," RFC 2212.

[17] Thyagarajan, A., Deering, S., "Hierarchical Distance-Vector Multicast Routing for the Mbone," SIGCOMM '95.

Ex.1030
CISCO SYSTEMS, INC. / Page 20 of 21

[18] Waitzman, D, Deering, S., Partridge, C., "Distance Vector Multicast Routing Protocol," RFC 1075.

[19] Wolf, L., Delgrossi, L., Steinmetz, R., Schaller, S., Wittig, H., "Issues of Reserving Resources in Advance," NOSSDAV '95.

[20] Wroclawski, J., "Specification of the Controlled-Load Network Element Service," RFC 2211.

Ex.1030
CISCO SYSTEMS, INC. / Page 21 of 21