Presented at
Fourth International Workshop on Network and Operating System Support for Digital Audio and Video
November 1993, Lancaster, UK

Reservation Protocols for Internetworks: A Comparison of ST-II and RSVP

Luca Delgrossi, Ralf Guido Herrtwich, Carsten Vogt, Lars C. Wolf

IBM European Networking Center
Distributed Multimedia Solutions
Vangerowstr. 18
D-69115 Heidelberg

{lars, luca}@ibmpa.awdpa.ibm.com, {rgh, vogt}@dhdibmip.bitnet

Abstract: Although efforts on reservation protocols for internetworks started quite some time ago, the research community is now becoming particularly active in this area, as proved by the high interest generated by protocols such as ST-II and RSVP. These two protocols, starting from different assumptions, have the common goal of providing guaranteed communication by reserving network bandwidth. This paper provides a short comparison of the two protocols. It describes and compares their mechanisms, focusing on the data forwarding, multicast, and quality of service aspects for multimedia communication. Rather than trying to decide which protocol is superior, we have identified the classes of applications which are better supported by one or the other protocol.

1 Introduction

For multimedia communication, i.e., the transmission of digital audio and video, three issues are widely believed to be important:

1. fast data forwarding,
2. multicast, and
3. service guarantees.

Each multimedia communication system should address these issues.

Fast data forwarding is needed to arrive at a minimum end-to-end delay as it is necessary for conversational multimedia services such as video conferencing. It is also convenient for highly interactive multimedia applications, e.g., for video editing, because short delays yield fast responses. Also, the faster the communication system can transfer a data packet, the fewer packets it needs to buffer.

Multicast is especially important for multimedia distribution services. In addition to reducing the required network bandwidth, multicast reduces the management overhead in routers and endsystems.

To provide service guarantees, resource management techniques have to be used. Without resource management in endsystems and routers, multimedia communication systems cannot provide reliable quality of service (QoS) to users. Transmission of multimedia data over unreserved resources (resources such as network bandwidth, buffer space, and CPU processing time have to be considered) leads to dropped or delayed packets. This has been already reported by other authors [6]:

> *"Anyone who has watched, or attempted to listen to, the Internet multicasts of recent IETF meetings can easily understand how badly a transmission is disrupted when it packets are dropped, and appreciate the potential benefits of a bandwidth reservation mechanism."*

Due to the effect on the quality of service provided, resource reservation plays a dominant role among the three issues listed. Therefore, multimedia communication protocols are often also referred to as reservation protocols.[1]

1.1 Reservation Protocols

Any resource reservation approach consists of two components:

- a resource reservation protocol for end-to-end negotiation, and
- a set of resource administration functions for the individual resources.

A resource reservation protocol performs no reservation of required resources itself, it is only a vehicle to transfer information about resource requirements and to negotiate QoS values users desire for their end-to-end application.

This means that resource administration schemes such as HeiRAT [15] are independent of and required for any of these protocols to make them useful. As bandwidth management functions need to be added throughout the network, one has to be aware that there is *no* solution using reservation techniques that does *not* require modifications in existing endsystems and routers. The introduction of bandwidth reservation always requires changes in the existing systems.

Besides resource reservation, resource administration systems must provide mechanisms for enforcing and scheduling resource access during the transmission phase. This affects all resources that contribute to the multimedia communication process (or at least those resources that potentially can become a bottleneck in the data transmission). Systems using reservation schemes properly have, hence, also to be modified in this respect. Real-time scheduling and multi-level priority queueing are typical techniques that need to be put in place.

The focus of this paper is on reservation protocols, i.e., the negotiation vehicles for QoS parameters, not on actual resource administration. We compare two different resource reservation protocols: ST-II and RSVP. These protocols are not the only existing reservation protocols [1, 7, 12], but renewed (in case of ST-II) or nascent (in case

[1] It should, however, be noted that reservation is not the only possible approach to multimedia communication. Another technique, typically referred to as media scaling [4], adjusts the amount of multimedia data according to the currently available network bandwidth. In terms of the media quality provided, scaling is inferior to a reservation approach as it cannot guarantee a consistently high QoS. A problem is also that scaling techniques depend on the media encoding used. However, media scaling is a very flexible and robust way of multimedia communication.

of RSVP) interest in these protocols in the current IETF work makes them the most interesting reservation protocols.

1.2 ST-II

ST-II [13] is an experimental Internet protocol designed in 1990 to transmit real-time simulation data. It is a successor of the ST protocol [8] which was specified by J. Forgie in 1979. Most of the original development work of ST-II was done at BBN. There exist at least four operational ST-II implementations and at least nine additional implementations are currently under development and should be available by the end of this year. These implementations use different hardware (among others workstations from SUN, IBM, DEC, and HP, routers from BBN and Wellfleet); some are kernel-level and some are user-level implementations. They are used mostly for audio and video communication in different networks (e.g., DARTNET or the Defense Simulation Internet), in several research projects, and in products for distributed video retrieval. A working group to revise the ST-II protocol was initiated at the recent IETF meeting in Amsterdam.

1.3 RSVP

RSVP [16] is a new protocol designed in 1993 by researchers from Xerox PARC and USC. To our knowledge, first implementations are underway after simulations have shown the feasibility of the approach.

1.4 Contents of This Paper

The next section of this paper compares the general approaches taken in the two protocols and shows the differences. After this, the similarities are discussed. Sections 4, 5, and 6 concentrate on the concepts chosen for the three important multimedia communication issues, namely data forwarding, multicasting, and QoS concepts. The paper concludes with a discussion of the advantages of either protocol and specifies which protocol may be used for which purpose.

2 General Differences

The major difference between the protocols is their position in the protocol stack. ST-II is a full internetwork protocol which contains data handling and control messages. It replaces IP at the internetwork layer. RSVP, on the other hand, is a companion protocol to IP which controls the way in which an IP implementation sends packets. It contains only protocol elements for control, not for data transfer.

These different designs in respect to specified functionality result from dissimilar modularization approaches. RSVP follows the approach to modularize to the largest possible extent, thus, it provides functions to transmit reservation information and not more. This allows the combination of RSVP as the resource reservation protocol with different routing and data transmission protocols. On the other hand, this approach prevents using knowledge about resource availability in the other modules, e.g., for routing.

ST-II's approach is to provide an integrated solution, combining data transmission and resource reservation. It allows to take advantage of resource availability knowledge in data transmission and routing. Such an integrated solution can make routing decisions considering desired QoS and resource availability in neighboring routers while a system consisting out of independent modules (as with RSVP) does not allow this in a clean way.

ST-II is a connection-oriented protocol which requires corresponding state information for connections to be held. Similar to ST-II RSVP stores on each system participating at the transmission of a stream information about existing streams, however, this information is called 'soft-state' which means that state information is kept on a time-out basis and has to be refreshed periodically.

Considering the reservation mechanism, another important difference between ST-II and RSVP is the direction in which the reservation proceeds. ST-II is sender-oriented, thus, the sender transmits a flow specification to the targets which contains information about resource requirements. Intermediate routers and targets may adjust the flow specification with respect to available resources before the flow specification is transmitted back to the sender.

RSVP is a receiver-oriented protocol in which the receiver describes its resource requirements in a flow specification that it sends in a RESERVATION message. This flow specification is forwarded in the direction of the sender. However, it is assumed that a sender has issued a PATH message before, providing information about its outgoing data. The RESERVATION message identifies the portion of data a receiver wants to obtain from the original stream. This message does not need to be passed all the way back to the source but rather just to an intermediate node that has information about the data flow available to which the receiver wants to connect.

The core difference between the original ST-II and RSVP reservation models is, hence, that using RSVP, a sender does not necessarily know who receives its data. With a sender-oriented protocol such as ST-II, the sender is always aware of its peers. This allows for dealing with privacy (e.g., for phone calls, sensitive data) and charging. On the other hand, it can make the source a bottleneck (see also Section 5).

3 General Similarities

Even with these important differences between the two protocols, there are several similarities. Both protocols allow heterogeneous receivers, i.e., not all receivers need to receive the same amount of data. Both can handle changes in receiver lists, targets can be added or removed from a running stream without establishing the whole stream again. RSVP and ST-II are basically simplex protocols, transmitting multimedia data in one direction only (ST-II, however, provides an optional duplex connection).

As should have been become clear from the introduction, a complete independence between reservation and data transmission is not possible. While RSVP does not deal directly with data transmission, nodes on the transmission path have to be aware of the reserved resources in a similar way as with ST-II. An important issue here is that it is necessary to know about resource availability to make routing decisions, thus, changes to routers are required for both protocols.

4 Data Forwarding Concept

Since multimedia data usually become quite large over time, efficient mechanisms to forward data are an important aspect for every multimedia communication system. As a connection-oriented protocol, ST-II performs routing decisions already during connection-setup. Hence, during data transmission only few program statements have to be executed. ST-II will need to reroute connections in case of system failures. Error handling routines in this case are costly as they need to work together with reservation.

RSVP concentrates on resource reservation only, therefore, it does not deal with data transmission. This means that the used data forwarding protocol (as well as any other component touching the multimedia message) has to be aware of the relation between a packet and the reserved resources.[2] In ST-II, such a relationship is established by the connection ID contained in every data message. As RSVP is a companion protocol to IP, there is no such information available: IP operates in connection-less mode.

This describes a potential problem for the use of RSVP: how can an IP-based flow be associated with its RSVP flow specification? One potential identification means is to use the IP address of the source. This, however, either means that only one source can exist on that system or that all streams that this source sends need to have the same flow characteristics. Another means for identification is the IP multicast address to which the stream is sent. This, however, means that multicast addresses cannot be assigned to reflect application groups (e.g., all participants of a video conference), but one multicast address has to be used for every stream.[3]

A discussion about fast data forwarding aspects of protocols such as IP multicast used in conjunction with RSVP is outside the scope of this paper. However, the effort necessary for routing decisions is not negligeable. Mechanisms such as route caching and packet prediction are essential to arrive at a short data forwarding time. With ST-II as a connection-oriented protocol this is inherently solved.

It should also be noted that with more and more connection-oriented networks such as ATM being deployed, the use of connections throughout the entire protocol stack seems to be a more natural approach.

[2] If a system takes only the network bandwidth of an outgoing link into account, the admission control component of a sender and the scheduling component of a router are the only components which have to be considered. However, multimedia messages are typically also handled by other entities such as the source and sink drivers and the application program.

[3] Alternatively, one could look into the IP message and try to identify a higher-level connection ID. Such a higher-level ID will exist in typical multimedia applications which use a audio/video transport protocol such as RTP [11] or HeiTP [3] on top of the network layer. We have worked on such a system several years ago [1] and have abandoned this approach because of its inherent layering violation.

5 Multicast Concept

Both ST-II and RSVP support multicast, but the details of their respective concepts are different: while RSVP's concept is built around a multiple-sender to multiple-target scenario, ST-II concentrates on a single-sender to multiple-target scenario and allows to cluster several streams to stream groups.

Two issues have to be considered in the discussion about multicast abilities in the two protocols:

1. how targets can be added to and removed from a stream,
2. how QoS negotiation for multiple targets is done and which constraints are imposed.

In ST-II, the sender transmits a connect message to the new target to add it to an existing stream. Hence, for large receiver groups, the load of the source in an ST-II connection is directly proportional to the number of receivers. This means that the source can become the system bottleneck. For this reason, we have designed ST-II extensions which allow for a receiver-initiated addition of targets [5]. The removal of a target from a stream may be originated by the target itself or the sender may send a DISCONNECT message to the target.

Following the approach of receiver orientation, inspired by IP multicast, a target adds itself to an existing stream in RSVP. To address the stream to which it wants to connect, the target has somehow to be informed about the streams available. RSVP itself is not concerned with the negotiation and dissemination of these addresses. Neither is IP multicast. In a system using the two protocols, some framework for setting up transient multicast groups is needed.[4] We refer to [14] for the discussion of how to perform such an address negotiation.

The second issue is the QoS negotiation for multiple-sender/multiple-target scenarios. RSVP provides a filter concept which allows a receiver to reserve only one set of resources which may be used by streams from several senders. The receiver specifies via the filter which packets from which stream should be delivered. This way, receivers can switch data transmission 'channels'. The reservations for several receivers may be aggregated at intermediate nodes for upstream resources.

While the concept of filters that was first introduced in [10] is intriguing, it remains an open issue on how effectively the concept can be implemented. The implementation of any filter beyond a simple switch among incoming channels will raise questions about the placement of functions in the communication protocol stack. The implementation of an audio mixer, the typical example of a filter, depends on the data format used; the same applies to a filter that scales down a media stream, another common example. Encoding-dependencies in the network layer have traditionally been avoided.

[4] We found that in the past the problems of multicast address dissemination have largely been neglected. In a system where receivers have the initiative to randomly tune in to some stream source in the network, they need to get information about the sources available. As it does not seem to be a good idea to flood the network with group creation messages, an address server approach is a likely solution.

To avoid restrictions on the encoding formats to be used, one can think of user-provided filters, propagated into the network [10]. While, for performance reasons, it is questionable whether routers will execute filter code at all, it is, for safety reasons, even more questionable whether they will execute user-provided filter code. In addition, it needs to be resolved how such filter code can be provided to the router, in source code for compilation on the router or cross-compiled by the user.

ST-II provides no such filter concept, for each sender a distinct set of resources has to be reserved. Through the stream group concept, ST-II provides a basic mechanism to combine reservations for several streams, however, appropriate algorithms need to be defined and implemented. ST-II does not require distinct reservations for all targets of one stream, the upstream resources for one stream are reserved for several receivers only once on common paths. The packet priority concept of ST-II allows the implementation of scaling methods to serve different targets differently as described in [4].

Since there are no algorithms defined for RSVP's filter mechanism and also not for ST-II's stream group concept, these both have to be considered as open issues.

6 QOS Concept

The levels of QoS assurance the protocols can provide differs between ST-II and RSVP. Both can support best-effort QoS, where in the context of this paper "best effort" means that not all resources which ever might be needed are reserved, but the resources which are needed for an average workload only.

In the regular transmission case, where no router failures occur, streams which have been established via ST-II can be served with guaranteed QoS. As already said, the actual resource reservation and scheduling is independent of the reservation protocol. Guaranteed QoS, however, requires to make reservations before the data transmission starts. This is the case with ST-II.

RSVP cannot support guaranteed QoS since there exists no direct relation between the reservation and the routing and data transmission protocol. Therefore, a route may change during data transmission (even without node failure) and this means that data may temporarily be transmitted over a path of unreserved resources.

For most consumers, small drops in the perceived QoS are acceptable. The best-effort QoS, as can be supported by ST-II and RSVP is sufficient for playback applications. While recording data, e.g., in a movie production studio, even a small degradation of the QoS cannot be tolerated. The ability to provide guaranteed QoS makes ST-II better-suited for these production-level applications.

In case of router failures, both protocols re-build the route from sender to target. With RSVP, the details of this re-building depends on the used data transmission protocol; yet, the source continues to send data, which is transmitted over unreserved resources until the next set of PATH/RESERVATION messages have been exchanged. With ST-II no transmission can occur until the new route has been established which also means that the reservation has been completed. While a detailed analysis is missing yet, it seems as if the RSVP approach is less disruptive for some fault-tolerant media encodings.

7 Conclusions

ST-II and RSVP are reservation protocols which differ in several properties. Therefore, we consider it inappropriate to decide which one is “better.” They have a different maturity: while a lot of experience has already been gained with ST-II and several applications have been developed using it, RSVP has just recently appeared on the scene and time is still required for a complete evaluation.

It is important to note that the two protocols are pieces of two different puzzles: ST-II was designed for multimedia applications with a moderate number of participants such as video conferencing or video on-demand services. It requires the presence of a resource allocator and a routing algorithm to provide a complete multimedia communication system. The most complete example of such a model that we know is the Heidelberg Transport System (HeiTS) [9]. RSVP was designed to support scenarios with very large numbers of senders and receivers. It requires a companion protocol to carry the data, a routing algorithm, and a flow specification. Both protocols require admission control and scheduling algorithms. The success of the two protocols will heavily depend on how complete multimedia communication systems can be built around them.

ST-II’s strong point is its provision of guaranteed QoS to multimedia applications. As a connection-oriented protocol its overhead during data transmission is low. The way ST-II maintains the stream’s state (without requiring the use of timers) leads to simple and efficient implementations. It seems accepted that the ST-II specifications need to be reworked. The several extensions proposed in the literature will further improve the protocol.

RSVP’s advantage is the ability of its filter mechanism to allow a receiver to switch between several data streams (channels) with one set of reserved resources only. Since filtering requires direct access to the data, it is to be seen how this mechanism can be implemented. Depending on the used data transmission protocol, RSVP might scale better than ST-II with respect to the number of receivers of a data stream. The scalability of both protocols with respect to the number of concurrent streams is an open issue.

References

[1] D.P. Anderson, R.G. Herrtwich: *Internet Communication with End-to-End Performance Guarantees*, in: J. Encarnacao (Ed.): Telekommunikation und multimediale Anwendungen der Informatik, GI-Jahrestagung, Darmstadt, Informatik-Fachberichte 293, Springer-Verlag, October 1991.

[2] S. Deering: *Host Extensions for IP Multicasting*, Internet RFC 1112, August 1989.

[3] L. Delgrossi, C. Halstrick, R.G. Herrtwich, H. Stüttgen: *HeiTP – A Transport Protocol for ST-II*, IEEE Globecom 92, Orlando, December 1992.

[4] L. Delgrossi, C. Halstrick, D. Hehmann, R.G. Herrtwich, O. Krone, J. Sandvoss, C. Vogt: *Media Scaling with HeiTS*, ACM Multimedia 93, Anaheim, August 1993.

[5] L. Delgrossi, R.G. Herrtwich, F.O. Hoffmann, S. Schaller: *Receiver-Initiated Communication with ST-II*, Technical Report 43.9314, IBM European Networking Center, Heidelberg, 1993.

[6] C. Elliott, C. Lynn: *ST-II in Practice*, Draft, June 1993.

[7] D. Ferrari, A. Banerjea, H. Zhang: *Network Support for Multimedia: A Discussion of the Tenet Approach*, TR-92-072, International Computer Science Institute, Berkeley, October 1992.

[8] J. Forgie: *ST - A Proposed Internet Stream Protocol*, IEN 119, 1979.

[9] R.G. Herrtwich: *The HeiProjects – Support for Distributed Multimedia Applications,* Technical Report 43.9206, IBM European Networking Center, Heidelberg, 1992.

[10] J. Pasquale, G. Polyzos, E. Anderson, V. Kompella: *The Multimedia Multicast Channel,* Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, November 1992.

[11] H. Schulzrinne: *A Transport Protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications.* Internet Working Draft.

[12] H. Tokuda, Y. Tobe, S.T.-C. Chou, J.M.F. Moura: *Continuous Media Communication with Dynamic QOS Control Using ARTS with an FDDI Network*, ACM SIGCOMM ‘92, Baltimore, October 1992.

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

[14] B. Twachtmann, R.G. Herrtwich: *Multicast in the Heidelberg Transport System,* Technical Report 43.9306, IBM European Networking Center, Heidelberg, 1993.

[15] C. Vogt, R.G. Herrtwich, R. Nagarajan: *HeiRAT: The Heidelberg Resource Administration Technique - Design Philosophy and Goals*, Kommunikation in verteilten Systemen, Munich, March 1993.

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