IWACA'94

Advance Reservation of Network Resources for Multimedia Applications

Wilko Reinhardt

Technical University of Aachen (RWTH Aachen), Dept. of Computer Science (Informatik IV)
Ahornstr. 55, 52056 Aachen, Germany
email: wilko@informatik.rwth-aachen.de

Abstract. To guarantee Quality of Services for continuous media like audio and video, intermediate nodes in networks must be able to reserve resources like buffer capacity, CPU time and bandwidth. Several methods have been introduced to verify that new connections can be accepted by the network without violating guarantees made for already established connections. One drawback of all proposed admission control algorithms is that they accept the reservation only at connection establishment time. It is not possible to reserve resources in advance. This paper introduces an algorithm that allows the user to reserve the desired QOS in advance. The paper further describes how the signalling of advance reservation can be realized using the Internet Stream Protocol ST-2.

1 Introduction

Communication support of distributed multimedia applications imposes several challenges on today's communication systems. The requirements of these applications reveal a lack of performance of traditional protocol stacks. Major reasons are contradictory demands made by the various media types. On one hand, isochronous data streams from live video and audio sources request high bandwidth, timely delivery of the data and low delay variation (jitter). On the other hand, asynchronous streams resulting e.g. from the transfer of still images or textual documents need a reliable transmission but can accept some delays and are insensitive to jitter. Broadband networks (e.g. B-ISDN, DQDB, FDDI, FDDI II, etc.) were developed in recent years, providing sufficient bandwidth to support the various requirements. Thus, the problem lies within the functionality and performance characteristics of the network and transport layers [5]. A new generation of protocols is necessary to bridge the functional and performance gap within these layers. XTP [10] and ST-2 [8] are two of the very promising approaches besides experimental approaches from several research laboratories [3,9]. The common part of all these protocols is the extended support of Quality of Service (QOS) parameters, allowing the user processes to define their exact requirements for the requested connection. In accordance with the QOS values the protocol functionality is configured and/or resources are reserved to provide a guaranteed service.

The 'real' support of QOS, i.e.. not just OSI's 'request and pray' approach is a key issue for sufficient communication support of multimedia applications. Currently, several extended QOS semantics are under discussion:

- *Threshold QOS:* The service user specifies 'worst case' values which are acceptable as the lowest quality. If values fall below these thresholds the user process will be informed so that it can decide to accept the lower quality or abort the connection. The QOS concept introduced by the OSI 95 project [1] proposes a more rigorous semantics, called *compulsory QOS*: If only one of the 'worst case' values is exceeded the connection will be aborted immediately.
- *Guaranteed QOS:* [2] introduced the guaranteed QOS semantics which has a rather strong legal flavour. When a service provider guarantees a performance value to the service user, it commits itself to provide that performance and to pay a penalty if the negotiated value can not be achieved. On the other hand the service user is obliged not to exceed the negotiated value.

Guaranteed QOS assumes that all intermediate nodes from the source to the destination allocate appropriate resources like bandwidth, buffer space and CPU time. These resources are exclusively used by the data streams the reservation was made for. A protocol for the negotiation of the QOS values and the corresponding reservations is needed. Examples of resource reservation protocols are ST-2 or RSVP [11]. The reservation protocol is a vehicle to transmit the user demands to the several network nodes. In general, it works independently from the routing and the admission control algorithms. The admission control algorithm is responsible to check if sufficient resources are available to serve the desired new connection without violating agreed performance parameters for already established connections.

A common drawback of all resource reservation protocols is that there is no possibility to reserve the resources in advance. This paper introduces an approach to solve this problem by presenting the necessary changes in the admission control procedure and the resource reservation protocol. Chapter 2 gives a brief introduction to ST-2 as an example for a reservation protocol. Chapter 3 describes an extension of intermediate nodes to manage the advance resource reservation. The necessary extensions of the resource reservation protocol are explained on the basis of ST-2 in chapter 4. The paper closes with a discussion of the approach in chapter 5.

2 ST-2: A Brief Overview

The experimental Internet stream protocol, version 2 (ST-2) as defined in RFC 1190 [8] is a revised version of the Internet stream protocol introduced in IEN-119 [4]. It provides especially realtime applications with guaranteed and predictable end-to-end level of performance across networks. Thus, multimedia applications are enabled to communicate on existing infrastructures and request QOS parameters according to their demands. Within the OSI reference model, ST-2 can be classified as a network layer protocol, and according to the ARPANET reference model as an Internet protocol. The main tasks performed are routing and signalling of data streams as well as the fast forwarding of data packets from the source host (called *origin* in the ST-2 terminology) to one or many remote hosts (called *targets*). Multicast communication is supported at the network layer by setting up a static routing tree from the origin to all targets, with the origin representing the root of the tree and the targets representing the leaves. Each intermediate node is an ST-2 router, forwarding the data downstream on the routing tree to the targets.

Data is transported over the tree as *streams*. RFC 1190 defines a stream as

- the set of paths that data generated by an application entity traverses on its way to its peer application entities,
- the resources allocated to support that transmission of data and
- the state information that is maintained describing the transmission of data.

The principle of ST-2 is to split connection management from data transfer. The stream set-up, maintenance and release are performed by the ST Control Message Protocol (SCMP). During connection establishment paths to all targets are installed. All intermediate nodes allocate, if possible, resources for the requested stream characteristics and establish connections to the next hops on the way to the targets. During the data transfer phase the data follow the established paths. In the header of each data packet the next intermediate node is identified by a hop identification number negotiated at connection set-up time, but it includes no control information. This leads to an efficient forwarding of all data packets since no control packets disturb the information flow along the path.

2.1 The Stream Protocol (ST)

The task of the stream protocol is to forward the data down the routing tree. ST provides a connection oriented but unreliable data transfer. Each ST header is 8 Bytes long, carrying 8 fields (Figure 1).

0				15	31
ST=5	Ver=2	Pri	T	Bits	TotalBytes
HID					Header Checksum
Timestamp					

Figure 1: ST Header

ST operates as an extension of the IP protocol. Therefore, the first field in the header contains a special IP version number to indicate an ST packet. The current version number of IP is 4, therefore the RFC 1190 proposes the version number 5 for the ST protocol. The second field contains the ST version number. The priority field (*PRI*) is used to indicate packets that can be dropped by an intermediate node if a stream is exceeding its resource allocations. During connection set-up the ST agents negotiate whether or not the header contains a timestamp and its semantics. If the timestamp is included in the header, the *T* bit is set. The *TotalBytes* field contains the complete length of the ST packet including header, optional timestamp and user data. The *HID* field is the hop identifier. Each hop within the stream is identified by a unique identifier. The HID represents a stream and how the packet will be forwarded to the next hop. When the packet leaves the station, the HID will be updated with the HID that indicates the stream at the next node. All HIDs related to a stream are negotiated during connection set-up. For the calculation of the *HeaderChecksum* field which covers only the ST header the standard Internet checksum algorithm is used.

2.2 The ST Control Message Protocol

SCMP is responsible for the stream control. Within the telecommunication world's terminology SCMP would be called a signalling protocol. Its task is to create, maintain and release the routing trees used by ST. SCMP messages are encapsulated in ST packets and are distinguished from data packets by a HID that is zero. SCMP messages are sent from one ST agent to a single other in a reliable mode. Each message must be acknowledged immediately, otherwise the sender of the message re-transmits several times until a timer expires.

Connection set-up is handled as follows: The origin ST agent receives a list of targets from the application that are destinations of the stream. The desired QOS parameters of this stream are described in the flow specification (called *Flow Spec*). The ST Flow Spec structure contains, among other parameters, values for the average and minimum acceptable bandwidth, acceptable and maximum values for delay, delay jitter and error rate. After receiving a connect request the agent calls the routing function that returns a set of next-hop ST agents and the parameters of the intervening networks. Using these results the ST agent is able to decide whether the selected networks will support the requested Flow Spec parameters. After the routing decision the ST agent sends connect messages to all determined next-hop agents, containing a proposal for the HID, link references, the updated Flow Spec in accordance to the reservation at the node, and an updated target list (all target lists together will deliver the original list). This procedure is repeated until all targets are reached. The target agent requests an acknowledgement for the modified Flow Spec parameters from the application, and sends back an acknowledgement message to the origin.

3 Advance Resource Reservation

Most of the work in the area of resource reservation at the network layer has been done on reservation at connection establishment time. If the resources are available at that time they are reserved in accordance to the user demands. The word 'if' describes an element of uncertainty, which means that the resources are only reserved when they are available and not used by other connections. In the worst case, the connection request must be rejected or the connection will be established only with extremely reduced QOS parameters. Today, this is a common situation if we consider e.g. wide area data links. Users are not always guaranteed a suitable connection upon their request. In this situation users have to try again later or they need a lot of patience since the offered performance of the network is not sufficient. However, if we consider new applications providing e.g. video conferencing with several participants this situation is not satisfying. A date for the meeting has to be arranged so that all participant are available at the scheduled time. What happens if they are not able to set up a conference due to the fact that not enough resources are available from the network? If the users wait for better conditions on the network, a new date will have to be arranged with all problems of setting up a time schedule between several partners and the risk that it is not sure that the resources are available at the new date.

This paper proposes to integrate an advance resource booking mechanism into the reservation procedure. [7] states that it would be a helpful feature for the users if they were able to reserve the requested resources a certain time in advance. The initiator of a conference could set-up the data streams for the time the conference is scheduled. At that time the necessary resources are guaranteed from the network.

To realize this concept an adequate signalling protocol must be responsible for the negotiation of the requested resources between the source, intermediate nodes and all targets. Within the network nodes some more work has to be performed. Not only the resources used by currently active connections have to be managed but also requests for future reservations have to be examined. It has to be decided if the resources are available at the desired time and if so, the agreed performance values and the dates must be stored and managed. This work can be performed by a *resource manager*.

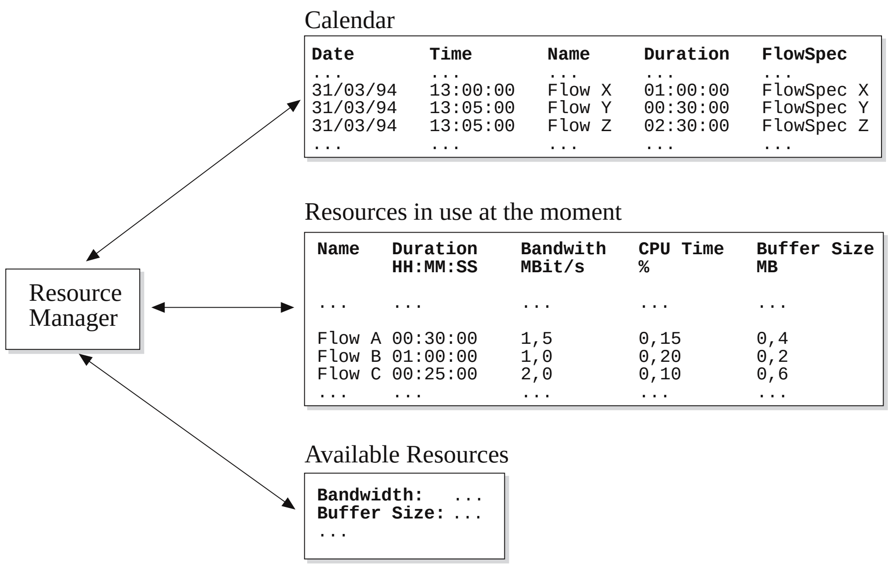

Figure 2: Scheme of the resource manager

Before a router reserves the resources, it has to check if sufficient resources are available to guarantee the desired QOS parameters. The router has to make sure that the guaranteed service parameters for existing connections are not violated. Therefore, an admission control algorithm is executed to determine whether already agreed parameters from existing connections and the requested performance values from the new connection can be guaranteed. Only in this case the new connection is accepted with the desired values.

If a new connection will violate the guarantees of other flows the desired QOS has to be reduced, or the connection request must be rejected.

The same procedure has to be performed if resources are allocated in advance. In this case the admission control algorithm must check if the new request can be accepted for the desired time without violating already confirmed reservations. Therefore, the functionality of a router has to be extended by a resource manager responsible for monitoring the available resources, the resources in use and the already allocated resources for the future. To remember the future allocation states, a *calendar* is needed that stores already confirmed reservations and their related parameters. These parameters include the date when the connection will be in place, their duration, a pointer to the name descriptor of the stream and a pointer to its agreed Flow Spec (Figure 2). All confirmed reservations are sorted in relation to their starting time. This makes it easier for the admission control algorithm to verify the situation at the particular time the new connection wishes to reserve resources in advance. The resource manager has to consider the situation at the starting time of the planed connection. It has to make sure that none of the performance guarantees for already confirmed future active connections are violated. Furthermore, it has to verify the connection request against all reservations that are made in the time period the new connection will be active. To make this verification process more efficient, connections should only be allowed to pre-reserve at certain times.

For the determination of this certain times the following considerations can be made: The starting time of a conference and its duration is determined by the user. In general, dates are made in steps of five or up to fifteen minutes. A date at e.g. 11:00:00 h is much more likely than a date at e.g. 11:02:34 h. Since the resource manager has to control all starting times of reserved connections the overhead can be reduced significantly when only a small number of possible starting times are allowed. Consider for an example a time period of two days within which advance reservations are possible. The necessary number of comparisons can be calculated as a function of the interval size. Time intervals in steps of 1 second cause 172.800 time periods within 48 hours, whereas an interval size of 5 minutes lead to 576 time periods. A further increase of the interval size up to 15 minutes reduces the number of time periods down to 192 periods.

If we further consider the already mentioned user behaviour, an interval size of five or fifteen minutes should be acceptable for advance reservations. This leads to the consequence that the resource manager should only accept starting times for reservation which are in accordance to these time steps. With this smaller amount of possible starting points it is possible to store the complete traffic situation for all time intervals. Therefore, the admission control algorithm is able to consider the traffic situation over the time period in question, without recalculating the reservations at the beginning of all time intervals. The number of comparisons is further reduced because the tests are only necessary within the duration of the connection. The duration of a video conference will for example be two hours. If we assume possible starting points in steps of five minutes the admission control must be called twenty-four times. Since this effort will not disturb the data transfer through the intermediate agents we can assume that the additional processing overhead is acceptable.

For the signalling of the advance resource reservation a protocol is needed that transmits the user request to all involved network nodes. Existing reservation protocols can be used if they are extended by adequate features. To realise the advance booking the connection setup of the reservation protocol can be used, since all negotiations are identical to the negotiations during the immediate connection establishment. The resource allocation within the intermediate nodes is generally independent from the reservation protocol.

The existing signalling protocols can easily be enhanced to provide the signalling for advance resource allocation. Two additional parameters have to be added to the respective FlowSpec of the reservation protocol:

- The starting time of the data flow across the connection and
- the duration of the connection from the starting event.

The starting time is only used for advance resource reservation, to indicate the date and time the new connection will start. The duration parameter is needed for all requested connections to evaluate if the required resources conflict with already reserved resources in the near future. The starting time can further be used to indicate the immediate connection setup. If the value is set to zero the path is established for immediate use.

4 Signalling of Advance Resource Reservation with ST-2

Only a few changes have to be made to use the ST-2 protocol for resource reservation. First the duration and starting time parameters as described above need to be integrated in the Flow Spec. The routing path is established immediately through a connect message, and all parameters (e.g. the HIDs, Flow Spec values, etc.) and resource requests are negotiated as in the conventional ST-2 version. (Figure 3) The only difference is, that all reservations are stored within the resource manager of each ST agent if the starting time parameter is not zero. The reservations are not performed immediately, but the resources are reserved for the time the connection will be used by the applications.

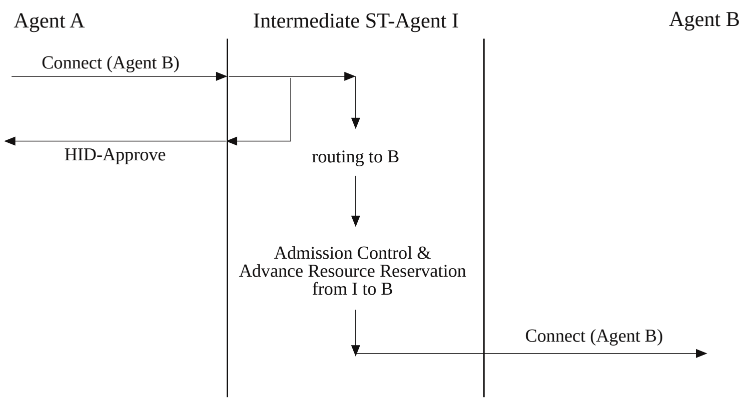

Figure 3: Processing of a connect messages at intermediate nodes

When the connection is to be activated by the users, a *FlowStart* PDU is sent from the origin to all targets. This PDU-type has to be integrated into SCMP. The FlowStart message indicates all intermediate ST agents at which resources allocated in advance for this connection that they will be used within the agreed duration (Figure 4). If the FlowStart message is not be received from the involved agents at the agreed time, the pre-allocated resources are released within a specified time interval and are available for other connections.

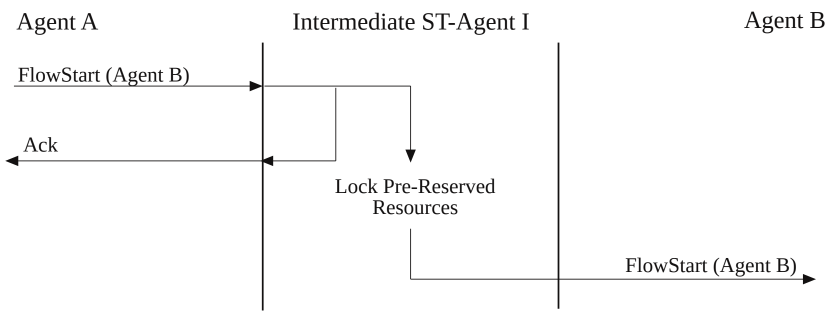

Figure 4: Processing of a FlowStart message at intermediate nodes

The FlowStart message should be integrated into the SCMP protocol since all SCMP messages are transferred in a reliable mode. An ST-2 agent that receives a FlowStart message immediately generates an ACK message to the previous hop. If the previous hop does not receive an ACK within a fixed time period it has to re-send the FlowStart message. Just to set a flag in the first packet of the data flow, indicating its start is not suitable since the transmission of ST packets is unreliable. If the first packet is lost or corrupted the intermediate agents will not be informed that the reserved resources will be used and release the reservation.

0		15	31
OpCode	Options	Total Bytes	
RVLId		SVLId	
Reference		LnkReference	
SenderIPAddress			
Checksum		0	
DetectorIPAddress			
Name Parameter			
UserData Parameter			

Figure 5: FlowStart PDU

Figure 5 depicts the structure of the FlowStart message. In addition to the mandatory SCMP parameters (OpCode, Options, TotalBytes, RVLId, SVLId, Reference, LinkReference, SenderIPAddress and Checksum) the DetectorIPAddress, Name and UserData Parameter are added. The DetectorIPAddress Parameter identifies the agent originating the message (in general the origin). It is copied from hop to hop and is primarily used for control purposes. The *name* data structure identifies the stream. It contains the origin's IP address, a unique stream identifier issued by the origin and a timestamp. The combination of the three parameters guarantees an globally unique identification of the stream. The origin is able to transmit messages from upper layer protocols with the UserData parameter. All additional parameters are also used in other SCMP PDUs and their structure is already defined in the RFC 1190. Each parameter contains a parameter ID, the length of the parameter data structure and some parameter specific values.

5 Discussion

The advance resource reservation mechanism causes additional overhead for connection set-up and resource management at the ST agents. Data streams requiring this feature, are typically long term connections transmitting video and audio streams for multimedia conferences. These streams require high bandwidth and real time delivery of data packets. New network technologies provide this high bandwidth, but we have to take into account that a lot of data streams with identical characteristics are transmitted, over several links, thus reducing the available capacity for a single stream. If we further consider that generally more than two partners are participating in a video conference an agreement for the date of this conference has to be arranged. If the connection can not be set-up due to limited available network resources this situation leads to a new time schedule for the meeting. A mechanism that provides the possibility for the users to define their requirement in advance seems to be a useful service provided by the network.

The overhead of the newly introduced advance resource reservation is limited to the connection establishment procedure. Most of the management PDUs for connection maintenance have to be exchanged anyway. Only one additional PDU that indicates the start of the data flow has to be added. Within the intermediate agent a calendar function has to monitor all reservations. The advantage of this proposal is that the data transfer is not disturbed by these negotiation, since all negotiations will be finished at the time the applications start transmitting their data.

The amount of memory that is needed to store information about all confirmed resources is another issue. It should be clear that there will be more 'intelligence' within future routers. In normal network nodes only the determination of the routes to the next hop are performed, while in future systems the nodes will be aware of the characteristics of data flows. An overhead for the processing of advance reservation should not influence the decision, whether or not it will be supported. In the end users should pay for the additional service.

The advance reservation mechanism causes several problems that have to be solved. Within this paper these problems can only be mentioned. They are described briefly with an idea how a solution may look like.

- *Time synchronisation*: All clocks within the involved nodes have to be synchronized with a central time so that the reserved resources are available at all nodes at the same time. A possible solution is the use of the Network Time Protocol [6].

- *Cancellation of the conference*: How long must the resources be reserved without use after a scheduled time? A delay has to be agreed within which the reservation is still valid. Within this time period the resource may be used by other connections, but with the proviso that the resources may be re-allocated when the delayed connection is activated. From my point of view users have to be charged for the effort this new feature causes. Therefore, the users have to pay a fee, if they don't use already reserved resources. The signalling of the cancellation could be performed with the disconnect message like a normal disconnect request in the original ST-2 version.

- *Failure within an intermediate agent*: In the time between advance resource allocation and the actual start of the data flow one or more intermediate agents may crash, thus corrupting advance booking within this node. This will interrupt the complete routing path. The intermediate agents should inform each other about their status as usual. ST-2 provides several protocol functions that allow the interchange of status information between intermediate nodes. The notify message e.g. is used if the routing function causes problems or if the resource allocation changes. The intermediate node that receives a notify messages decides whether to solve the problem locally, or to forward the message to the previous node or the origin.

- *Exceed of reservation duration:* The proposed mechanism requires the users to define the duration of the reservation. This is also a problem for traditional meetings that are not support by communication systems. Meeting room have to be reserved and a time schedule is agreed for the agenda. For the reservation of network resources it has to be investigated whether the connection is aborted immediately or the connection is further support with best effort QOS.

6 Conclusion

Within this paper an algorithm for advance reservation of network resources was introduced. It has been considered which extension are necessary for the common reservation procedures, and how they can be integrated into existing environments. An enhancement of the reservation protocol ST-2 was presented that allows to support advance reservation. As a result of the final discussion it can be stated that such a mechanism is a desirable feature for multimedia conference applications, but there are still problems to be solved.

References

1. A. Danthine: *A new Transport Protocol for the Broadband Environment*, in A. Casaca, ed., Broadband Communications, Estoril, Jan. 1992, C-4, Elsevier Science Publisher (North-Holland), Amsterdam, 1992.

2. D. Ferrari: *Client Requirements for Real-Time Communication Services*, RFC 1193, Nov. 1990.

3. D. Ferrari, D. Verma: *A Scheme for Real-Time Channel Establishment in Wide-Area Networks*, in IEEE Journal on Selected Areas in Communications, Vol. 8, No. 3, April 1990.

4. J. Forgie: *ST - A proposed internet stream protocol*, IEN 119, MIT Lincoln Laboratory, 1979.

5. B. Heinrichs, W. Reinhardt: *The DyCE Concept - Architecture and Implementation Strategies*, in V.B. Iversen, ed., Integrated Broadband Communication Networks and Services, Copenhagen Apr.1993, C-18, Elsevier Science Publisher (North-Holland), Amsterdam 1994.

6. D. Mills: *Network Time Protocol (Version 3), specification, implementation and analysis*, RFC 1305, 1992.

7. C. Partridge, S. Pink: *An implementation of the revised Internet Stream Protocol (ST-2)* in Interworking: Research and Experience, Vol. 3, 1992.

8. C. Topolcic (ed.): *Experimental internet stream protocol*, RFC 1190, 1990.

9. C. Vogt, R. Herrtwich, R. Nagarajan: HeiRAT: *The Heidelberg resource administration technique: Design philosophy and goals*, in N. Gerner, H.-G. Hegering, J. Swobodowa, ed. Communication in Distributed Systems, Springer Publisher, 1993.

10. *XTP Protocol Definition Revision 3.6*, 11 January 1992.

11. L. Zhang, S. Deering, D. Estin, S. Shenker, D. Zappalla: *RSVP: A new resource reservation protocol*, IEEE Networks, No. 5, 1993.