Issues of Reserving Resources in Advance

Lars C. Wolf, Luca Delgrossi, Ralf Steinmetz, Sibylle Schaller, Hartmut Wittig

IBM European Networking Center
Vangerowstr. 18
D-69115 Heidelberg

{lars, luca, rst, schaller, wittig}@heidelbg.ibm.com

Abstract: Resource management offers Quality-of-Service reliability for time-critical continuous-media applications. Currently existing resource management systems provide only means to reserve resources starting with the reservation attempt and lasting for an unspecified duration. However, for several applications such as video conferencing the ability to reserve the required resources in advance is needed. This paper explains a model for resource reservation in advance. We identify and discuss issues which must be resolved in resource reservation in advance systems. Some of the possible scenarios to be considered are described and we show how the resource reservation in advance scheme can be embedded in a general architecture.

1 Introduction

Computer systems used for continuous media processing have to be able to cope with streams with data rates of several Mbits/s and to provide timely processing guarantees, for instance such that an endsystem shall synchronize audio and video streams up to a granularity of about 80 ms [11].

Since available system resources are not abundant, applications have to be 'protected' such that they have access to the required resources in time because otherwise the user will notice a drop in the presentation quality. Hence, a means to manage the available system resources is necessary.

Resource management provides a way to offer applications reliability with respect to *Quality-of-Service* (QoS). A resource management system controls the access to scarce system resources needed for audio and video data processing. It checks whether additional service requests can be satisfied, if yes, the required resources are reserved for that application, if not, the request is rejected. The importance of resource management and the need of respective techniques is now widely accepted in the research community [9].

1.1 Motivation

Today existing resource management systems, for instance, HeiRAT [13], offer functions which only allow to reserve resources for a time interval which starts with the reservation attempt and which usually lasts for an unspecified time.

For several application scenarios this model of immediate reservations is not appropriate. Consider, for instance, a meeting room (conferencing) scenario which has to be supported by multimedia systems. Traditionally, a meeting will be scheduled for a specific time in a selected room in order to allow for arrangement of all participants. To be sure that the room will be available at the scheduled time, a reservation is made, in some form of a meeting room calendar, *before* the meeting starts. The time between the reservation and the meeting itself can vary from short intervals, e.g., hours or few days, to very long periods, e.g., months. In addition to 'one time events', meetings such as department meetings occur periodically. To support these 'virtual meeting room' scenarios the resource reservation system must offer mechanisms to reserve in advance the resources needed during the conference, i.e., network, router, and end-system resources.

Resource Reservation in Advance (ReRA) is not only needed for conferencing but for other scenarios such as video-on-demand as well. This resembles a video rental scenario where a user 'orders' a video for a specific time which means for video-on-demand that the resources necessary to retrieve, transfer and present the video have to be reserved in advance, i.e., video server, network, router, and end-system resources.

1.2 Related Work

Despite the fact that the necessity of mechanisms for ReRA seems to be accepted in the research community, at least to our knowledge, only few work has been done so far in the field of ReRA.

Ferrari, Ramaekers and Ventre are among the first who notice in [6] that ReRA is a useful concept. They describe the parameters for the start and duration of a reservation but go not further into the requirements of ReRA nor do they discuss design aspects.

Campbell, Coulson and Hutchinson describe in [1] their Quality of Service Architecture and specify start and end time parameters for 'forward reservations', however, they also state in the article that these parameters have been omitted so far and remain for further study.

Reinhardt gives in [10] a straight-forward extension of the resource reservation protocol ST-II [12] to exchange, within the flow specification, the necessary information about start and end time of a reservation in advance and describes shortly some problems to be solved within ReRA. He does not discuss a general model and possible scenarios.

This proceedings contain two other articles about ReRA. Ferrari, Gupta and Ventre from the Tenet group at the University of Berkeley describe in [5] a scheme for advance reservations of real-time connections. They concentrate on connection establishment, resource partitioning, and a mechanism to manage effectively the table of all set advance reservations, but do not give a general architecture.

Degermark, Köhler, Pink and Schelén show in [3] an extension of the admission control algorithm for predicted service suggested by Jamin, Clark, Shenker and Zhang [2, 7]. A general model or architecture is not presented in their work.

1.3 Contents and Outline of this Paper

This paper discusses a model for ReRA, identifies the issues to be resolved and describes some of the possible scenarios to be considered. Since ReRA seems to be a complex topic, we believe that a complete solution to all its related problems requires deep understanding and discussion within the research community. We hope to brighten the discussion on this interesting and important research field.

The paper is organized as follows: Section 2 provides a short description of a common resource management scheme; Section 3 introduces the notion of ReRA and presents the base model; Section 4 deals with issues of ReRA and Section 5 shows how the ReRA scheme can be embedded in a general architecture.

2 Resource Management

A complete discussion of resource management is, due to space limitations, out of the scope of this paper (see, for instance, [8] for a discussion of resource management). Here, we only describe shortly those parts relevant to the ReRA scheme.

The resource management component on each system which is part of an application must provide certain functionality for each "active" resource (i.e., CPU, network adapter):

- *Interpretation and translation* of the application specified QoS in metrics applicable to the affected resources.
- *Capacity test* to check whether the available resource capacity (taking the existing reservations into account) is sufficient to handle the new request.
- *QoS computation* to calculate the possible performance the resource can provide for the new stream.
- *Resource reservation* to reserve the required resource capacities.
- *Resource scheduling* to perform the scheduling of the resource during data processing such that the QoS guarantees are satisfied.

Figure 1 illustrates how this functionality can be distinguished into two resource management phases. In the set-up phase (also called 'QoS negotiation') applications specify their QoS requirements (e.g., throughput and delay). These parameters are used for capacity test and QoS computation which finally results either in resource reservation or in rejection of the reservation attempt if the QoS cannot be met due to a lack of resources. After the negotiation phase has been successfully completed, in the data transmission phase, the resources used to process the user data are scheduled with respect to the reserved resources (also called 'QoS enforcement').

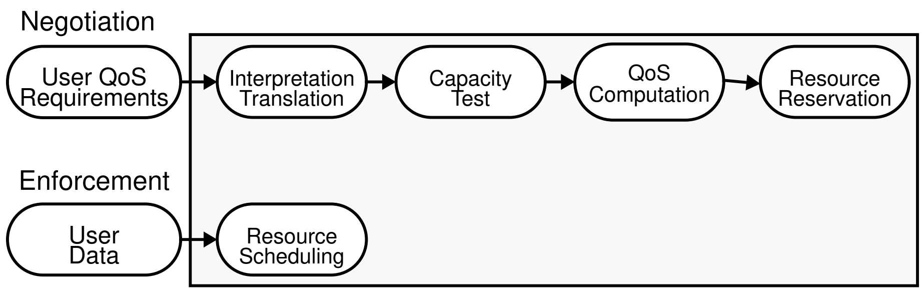

Figure 1: Resource Management Phases.

In a ReRA system, the negotiation phase is not in close vicinity to the enforcement phase and the resources are not reserved for immediate but for delayed use.

To achieve QoS provision for a distributed multimedia application, resource management is applied to all resources on the transmission and processing path, from the sending host via gateways or any other computers and networks to the receiving host. Resource reservation protocols such as ST-II [12] and its more recent version ST2+ [4], and RSVP [14] offer the functionality for QoS provision in distributed systems.

3 A Model for Resource Reservation in Advance

This section introduces the notion of ReRA and presents a basic ReRA model. To distinguish ReRA schemes from other reservation schemes, e.g., existing reservation techniques, we classify reservations based on two key factors:

- whether the resources are exploited at reservation time, and
- whether the reservation duration is known at reservation time.

This leads to the simple matrix presented in Table 1:

Table 1: Classification of Reservation Schemes.

		Reservation Duration	
		Known	Unknown
Resource Usage	Immediate	non-ReRA / ReRA	non-ReRA
	Deferred	ReRA	?

Traditional resource management systems (non-ReRA) assume that the resources are immediately used after they have been successfully reserved and no assumptions are made on the duration of the reservations. A ReRA scheme, on the contrary, is characterized by deferred resource usage and reservations of known duration (which might possibly be enlarged).

We feel that it is difficult to implement ReRA schemes if the duration of the reservation cannot be determined at reservation time. Therefore, we include a question mark in the correspondent table entry. In case of immediate usage and known duration, both schemes can be realized. We clarify this point at the end of this section, after introducing the ReRA model.

3.1 Basic Model

To provide an appropriate model for ReRA, we start from the common reservation scenarios of everyday life. In such scenarios, appropriate actions are required as part of the reservation, e.g., we have to specify at what time and for how many persons we intend to reserve. Here, we introduce a simple model to define these actions and regulate the interaction between the reservation requestor (i.e., the client application itself or a ReRA agent acting on behalf of the application, cf. Section 5) and the service provider (e.g., network and server applications). The model is shown in Figure 2.

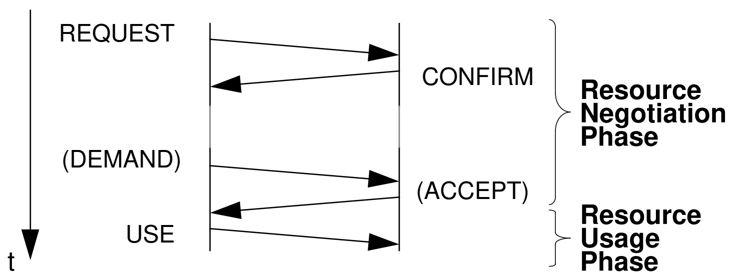

Figure 2: Reservation in Advance Primitives.

The ReRa scheme consists of two parts:

- resource reservation in advance
- usage of reserved resources

In the first part of the ReRA scheme, the client issues a REQUEST and it specifies the nature of its request by indicating how much of the resource capacities will have to be reserved for its application, i.e., it gives a *workload specification*. It also specifies the points in time that define *beginning* and *duration* of the reservation. The service provider may then CONFIRM the reservation. As part of this confirmation, it possibly provides the client with a *reservation identifier* for later client identification. This terminates the first part of the ReRA scheme.

The second phase begins shortly before the client intends to exploit its reservation. The client contacts the service provider to DEMAND the previously reserved resources. It may be requested to show some form of identification, which the service provider will ACCEPT. After receiving this acceptation, the client eventually exploits its reservation by making USE of the reserved resources.

It is possible to further simplify this scheme by eliminating DEMAND and ACCEPT. In this case, the client attempts directly to make use of the allocated resources and client identification can be associated with USE. However, we feel that the scheme described in Figure 2 is more convenient because it provides for the management system the ability to prepare the resource usage phase and generally allows for higher flexibility. For instance, it is often necessary to change reservations at the very last moment. A common example is a couple of unexpected guests for dinner making a larger table necessary. With the DEMAND and ACCEPT scheme, DEMAND can be used to adjust reservations appropriately when possible. Also, an explicit ACCEPT from the service provider is desirable because it informs the client that everything is set so that its requirements can be met.

3.2 Timing

In order to appropriately define a ReRA system, it is important to analyze the temporal relationships among the events. Consider the events in Figure 3.

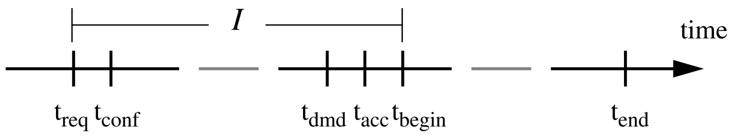

Figure 3: ReRA Model Temporal Sequence.

In our view of a ReRA system, we assume that the distance between t_{req} and t_{conf} is short, about the order of delay tolerated by Remote Procedure Calls (RPC). The same holds for t_{dmd}, t_{acc} and t_{begin}. On the contrary, t_{req} and t_{begin} are possibly very distant, possibly in the order of weeks or months. Let us call I the time interval between resource reservation and exploitation:

$$I = t_{begin} - t_{req}$$

When I is too small, making a ReRA reservation is pointless and a normal reservation scheme can be adopted. A ReRA system may define a value for I, say I_{min}, such that requests with:

$$t_{begin} - t_{req} < I_{min}$$

are rejected. In the same way, an I_{max} value for I can be defined to prevent applications to request their reservation long ahead of time, e.g., to prevent storing too much reservation state. These definitions help clarify Table 1 when both non-ReRA and ReRA are possible, the I_{min} value can be used to decide which of the two schemes to adopt.

4 Issues in Resource Reservation in Advance

4.1 Resource Management

Assuming non-ReRA and ReRA schemes will have to coexist on future systems, there are two ways of managing the resource capacity:

- the capacity is *partitioned* into two parts so that a portion is assigned to non-ReRA and the other to ReRA reservations (cf. [5]).
- non-ReRA and ReRA reservations make *shared use* of the resource capacity.

The two alternatives are briefly sketched in Figure 4. Partitioning a resource's capacity makes it easier to implement the system at the price of a non-optimal exploitation of the resources. The latter can be possibly reached by sharing the available capacity, but the system realization is in this case more complex because of the additional need of a reservation pre-emption scheme. Note that if the durations of all reservations are known (which can be achieved by the introduction of a default duration value), then it would be possible to apply the sharing strategy without preemption.

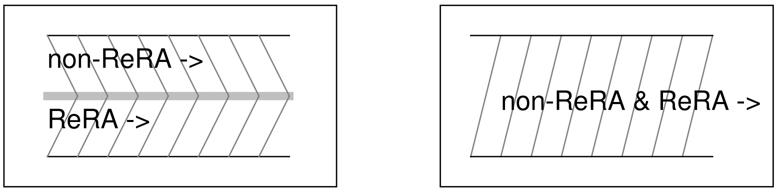

Figure 4: Examples of Resource Capacity Allocation Strategies: Resource Partitioning (left) and Sharing (right).

4.2 Reservation Duration

ReRA schemes require that the applications reserve resources over a certain time interval. The problem is, it is difficult to predict in advance how long some applications may need their reservations. In a video-on-demand system, it is usually possible to foresee the duration of a movie. Still, the user may increase this duration by pausing playout or even by stopping and rewinding to watch his favorite sequence a second time. In the same way, meetings take often longer than expected. Note that also shorter durations may be induced, e.g., by skipping through movie sequences or by rapidly adjourning a meeting.

When the actual duration does not correspond to the reservation, several issues arise:

- if the *duration is shorter*, exceeding resources should be freed and made available for other applications. In this case, resources are more likely to be made available

for immediate use and for traditional reservation requests than for new ReRA requests, because of the short notice (which is likely < I_{min}).
- if the *duration is longer*, the system may or may not have a sufficient amount of resources to serve the application with its needed QoS. If enough resources are available, one possibility is not to interrupt the service and to provide the application with the means to extend its previous reservation. If insufficient resources are available, the system may still attempt to serve the application on a best-effort basis with a degradation in the QoS.

Means to extend a previous reservation are desirable for a ReRA system, i.e., in addition to the primitives discussed in Section 3.1 a CONTINUE primitive to enlarge an already established reservation is necessary.

Sometimes, the delay can be foreseen, e.g., it becomes clear that the meeting will take longer than expected. In such cases, it may be possible to extend in advance, i.e., before it expires, a previous reservation. This will only be successful if sufficient resources are available, e.g., if no other reservation overlaps with the extended reservation. For the prolongation of the reservation, we differentiate two alternatives, (i) the management system informs the application/user that the reservation will expire and queries whether the reservation should be extended or (ii) the application has complete responsibility about the reservation state and must take appropriate action to lengthen the reservation.

The ability to extent reservations encourages applications not to book resources over too long time intervals in order to be guaranteed against unpredicted longer durations. In a cost-based ReRA system, this can also be imposed by adequate payment policies of the associated reservation costs.

If it is known before the beginning of usage that the needed reservation duration is different to the originally specified length, the DEMAND mechanism can be used to adapt the reservation to the required duration. If the duration shall be shortened, the reservation requestor might be charged for preventing other reservations. For prolongation, the necessary resources might be unavailable, however, due to the earlier request, the risk is lower than during the usage phase of a reservation.

4.3 Failure Handling

We believe that failure handling is one of the most difficult topics within ReRA, here we outline only shortly some aspects. For the handling of failures, we must distinguish when the failure occurs:

- after the reservation, but before the usage (between REQUEST and USE), or
- during the usage phase.

The second case is not different from failures within traditional reservation-based systems. The former case, however, requires special attention.

The reservation state information stored at nodes might be needed for long lasting time periods. State information must be stored in stable storage not only to protect against failures, but also since any node may be restarted between REQUEST and USE also regularly, e.g., for maintenance.

In opposite to failures occurring during data transmission, no client is running when a node notices a failure. Hence, means to inform the clients explicitly about the failure situation and whether it can be resolved in time must be provided. The failure itself might, however, not be detected at the failing node but only at a neighbor which has only partial information about the reservation state stored at the node.

4.4 Relation Between ReRA and Scripted Presentations

For presentations, e.g., specified via a script, the information is available about which parts of the presentation are needed at which time, e.g., when a particular video is must be presented[1]. This information can be used to perform ReRA for all presentation parts. Support from the ReRA system for the presentation application should be available to ease the applications task, additionally such support functions also provide more information to the ReRA management which can lead to better scheduling decisions.

5 An Architecture Exploiting ReRA Mechanisms

In addition to services needed for non-ReRA systems, e.g., multimedia communication system and resource management system, the following components are required as shown in Figure 5:

- a *ReRA agent* at each system participating in the processing of the stream, i.e., enhancing the 'standard' resource manager,
- a *user agent* providing the front end for users to inform them about incoming ReRA attempts, hence, it interacts with the ReRA agent. An incoming ReRA attempt is also combined with the announcement of a data stream, e.g., the invitation to a video conference.

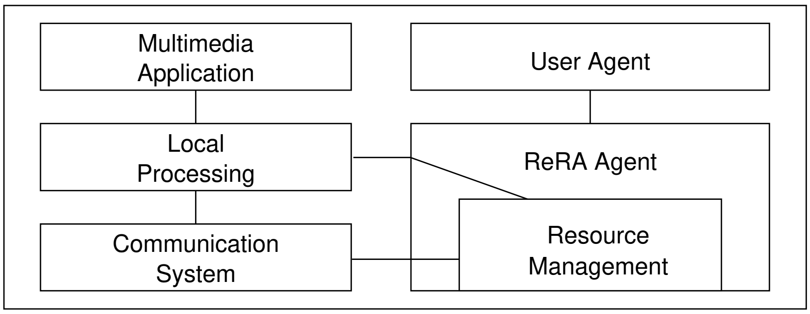

Figure 5: ReRA Endsystem Components.

5.1 ReRA Agent

The ReRA agent is illustrated in more detail in Figure 6. Compared with a traditional resource management system it contains the following extensions:

- time parameters in the flow specification exchanged via the used resource reservation protocols,

1. This information is only available if later parts of the presentation do not depend on user input. In that case, heuristics can be used to determine the most likely presentation path. However, use of ReRA beyond points where user input is required seems to be questionable.

- time parameters in the reservation database, and
- new resource management algorithms in the resource manager.

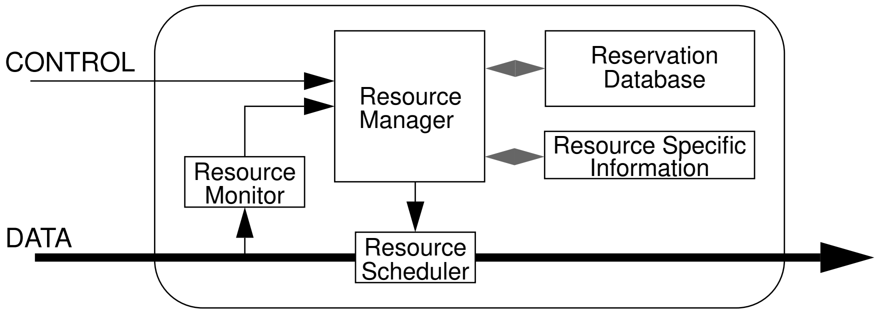

Figure 6: ReRA Agent.

In the used resource reservation protocol, the flow specification transmitted as part of the reservation setup PDUs must be extended by the parameters *start_time* and *duration*. While these PDUs are exchanged between the systems participating in the final transfer of the data stream, the resource management systems of these nodes are informed about the time and QoS parameters of the reservation request.

The reservation database has to be enhanced to store the time parameters of a reservation in advance. This includes the date for the start of a reservation and its duration. If the resource manager receives a reservation request it uses these time parameters to check the available resource capacities against the requested capacities in the time period given by the request. If requested resources exceed the available resources the resource manager rejects the reservation request.

5.2 User Agent

The user agent is similar to the user agent of a mail system. It provides the interface for the user to handle resource reservations in advance. An incoming invitation to a multimedia application (to be started sometime in the future) is presented to the user who can acknowledge or reject the invitation. Using this agent, users can also start reservation attempts themselves. The user agent should provide the ability to start automatically the application at the time the data stream has been scheduled, i.e., just before the conference begins.

5.3 Scenarios

Considering distributed multimedia applications, two different scenarios can be distinguished:

- the sender of the data stream has complete knowledge about the set of receivers, i.e., their identities,
- the sender has no knowledge about the set of receivers, i.e., the sender knows neither identity of receivers nor even whether anyone is listening.

The former scenario occurs in point-to-point communication or in multicast communication using a sender oriented communication setup, this is the case, for instance, in

ST-II. The latter scenario is used in multicast communication where the receivers are not specified by the sender and is typical for IP multicast communication.

In both cases, the human users who will consume the presentation of the transmitted data must be informed about the intended transmission of the multimedia data and accept or reject the stream. While it would be possible, in the first scenario, to perform the announcement phase together with the reservation attempt, it should be avoided. The reason is the severe drawback that until the user decided about the delivery of the stream, the resources on the complete path from the sender to the receiver must be set aside (for the future time frame) for the case that the receiver accepts the stream. However, the reaction of the user might be delayed for an unknown amount of time – the user might even not be at the computer for days or weeks, e.g., during business trips or vacations. Keeping resources reserved for such a long time can lead to rejected reservation attempts for other applications even if finally the user decides not to receive the stream and, therefore, available resources at the time the other applications intended to run. Additionally, it complicates the implementation of the ReRA components and the recovery from system failures. Altogether, it becomes clear that a distinction into the *announcement* of the data stream and ReRA for that stream is necessary.

6 Summary

While current resource management systems provide mechanisms which offer reliability with respect to QoS, this is not sufficient since not all application scenarios we are used to from our daily live can be supported.

ReRA mechanisms are needed to for several important classes of applications. However, ReRA is more than a simple extension of current resource reservation systems. As part of the development of ReRA systems, several issues must be solved.

The integration of current reservation schemes and ReRA requires resource partitioning or the ability to preempt resource usage. Applications must be offered a variety of mechanisms to prolong and adapt reservations. Failure handling raises difficult questions and must be carefully integrated into the system architecture. The provision of reservation mechanisms is only one part of a complete ReRA system. Agents to interact with the user, for reservation request generation as well as for the presentation and handling of incoming invitations, are necessary.

Acknowledgments

The description of the relationship between ReRA and scripted presentations is based on the hint of one of the anonymous reviewers.

References

1. A. Campbell, G. Coulson, D. Hutchinson: "A Quality of Service Architecture", ACM Computer Comunication Review, Vol. 24, No. 2, April 1994, pp.6-27.
2. D. Clark, S. Shenker, L. Zhang: "Supporting Real-Time Applications in an Integrated Packet Services Network: Architecture and Mechanisms", SIGCOMM 1992.

3. M. Degermark, T. Köhler, S. Pink, O. Schelén: "Advance Reservation for Predicted Service", Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995.

4. L. Delgrossi, L. Berger (Ed.): "Internet STream Protocol Version 2 (ST2) – Protocol Specification – Version ST2+", Internet Draft – Work in Progress, IETF ST Working Group, March 1995.

5. D. Ferrari, A. Gupta, G. Ventre: "Distributed Advance Reservation of Real-Time Connections", Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995.

6. D. Ferrari, J. Ramaekers, G. Ventre: "Client–Network Interactions in Quality of Service Communication Environments", Proceedings of the Fourth IFIP Conference on High Performance Networking, University Liege, Belgium, December 1992, pp. E1-1 – E1-14.

7. S. Jamin, D. Clark, S. Shenker, L. Zhang: "Admission Control Algorithm for Predictive Real-Time Service", Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, CA, USA, November 1992.

8. K. Nahrstedt, R. Steinmetz: "Resource Management in Networked Multimedia Systems", IEEE Computer, Vol. 28, No. 4, April 1995.

9. NOSSDAV4: "Panel Discussion (D. Ferrari, K. Jeffay, D. Northcutt, J. Rosenberg)," Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster University, UK, November 3-5, 1993.

10. W. Reinhardt: "Advance Reservation of Network Resources for Multimedia Applications", Proceedings of the Second International Workshop on Advanced Teleservices and High-Speed Communication Architectures, 26.-28. September 1994, Heidelberg, Germany.

11. R. Steinmetz: "Human Perception of Audio-Visual Skew" in "Hochgeschwindigkeitsnetze und Kommunikationssysteme" W.Effelsberg, A.Danthine, O.Spaniol (Ed.), Kluwer, 1994.

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

13. C. Vogt, R. G. Herrtwich, R. Nagarajan: "HeiRAT: The Heidelberg Resource Administration Technique - Design Philosophy and Goals," Kommunikation in Verteilten Systemen, Munich, Germany, March 3-5, 1993.

14. L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala: "RSVP: A New Resource ReSerVation Protocol," IEEE Network, September 1993, pp. 8-18.