WoGriSt97

Proceedings of the IEEE, Vol. 85, No.12, December 1997, pp. 1915-1933

Multimedia Communication

Lars C. Wolf[1], Carsten Griwodz[1], Ralf Steinmetz[1,2]

1
Industrial Process and System Communications
Dept. of Electrical Engineering & Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany

2
GMD IPSI
German National Research Center
for Information Technology
Dolivostr. 15 • D-64293 Darmstadt • Germany

{Lars.Wolf, Carsten.Griwodz, Ralf.Steinmetz}@kom.th-darmstadt.de

Abstract: Multimedia communication deals with the transfer, the protocols, services and mechanisms of discrete media data (such as text and graphics) *and* continuous media data (like audio and video) in/over digital networks. Such a communication requires all involved components to be capable of handling a well-defined quality of service. The most important quality of service parameters are used to request (1) the required capacities of the involved resources, (2) compliance to end-to-end delay and jitter as timing restrictions, and (3) restriction of the loss characteristics.

In this paper we describe the necessary issues and we study the ability of current networks and communication systems to support distributed multimedia applications. Further, we discuss upcoming approaches and systems which promise to provide the necessary mechanisms and consider which issues are missing for a complete multimedia communication infrastructure.

Keywords: multimedia, communication, quality of service, reservation, scaling

1 Introduction

Multimedia systems have attracted much attention during the last few years in the society as a whole and in the information technology field in particular. Multimedia communication comprises the techniques needed for distributed multimedia systems. To enable the access to information such as audio and video data, techniques must be developed which allow for the handling of audiovisual information in computer and communication systems.

In this paper, we discuss requirements for the handling of such data in communication systems and present mechanisms which have been developed already or which are under development to fulfill the tasks necessary in distributed multimedia systems. We also discuss necessary issues which have not been studied in sufficient detail today and which must therefore be addressed in future.

We first describe in a more precise way what we mean by the term "multimedia". Unfortunately, "multimedia" has become a buzzword used to denote any kind of "new digital media" being manipulated or displayed by machines. This very imprecise (and very often employed) notion leads to a labelling of all types of media data computation, transmission, storage, manipulation and presentation with the term "multimedia". Since the mid eighties we have proposed (and even from time to time we imposed) a much more crisp and restricted specification.

> *Multimedia itself denotes the integrated manipulation of at least some information represented as continuous media data as well as some information encoded as discrete media data (such as text and graphics). The "manipulation" refers to the act of capturing, processing, communication, presentation and/or storage.*

As outlined in [StNa95] we understand continuous media data as time-dependent data in multimedia systems (such as audio and video data) which is manipulated in well-defined parts per time interval according to a contract.

Hence "multimedia communications" deals with the transfer, the protocols, the services and the mechanisms of/for discrete *and* continuous media in/over digital networks. The transmission of digital

---

Part of this work was done while the authors were with IBM[s European Networking Center, Germany.
This work is supported in part by a grant of: Volkswagen-Stiftung, D-30519 Hannover, Germany.

video data over a dedicated TV distribution network is not multimedia as long as it does not allow the transfer of some type of discrete media data as well. A protocol designed to reserve capacity for continuous media data transmitted in conjunction with discrete media data over, e.g., an ATM-LAN, is certainly a multimedia communication issue.

Information processing in a time-sharing environment is performed without any hard time constraints. The system responds to a user interaction as soon as possible but often lacks support for real-time data. The use of discrete media still governs traditional computing, while the integration of continuous media into existing computer environments creates the new complexity of time-dependent data processing. 'Correctness' in multimedia communications is – in addition to the traditional computer communications error handling – determined by whether deadlines are met or not.

In networked multimedia applications various entities typically cooperate in order to provide the mentioned real-time guarantees to allow data to be presented at the user interface. These requirements are most often defined in terms of Quality of Service (QoS). QoS is defined as the set of parameters which defines the properties of media streams. In accordance with [StNa95] we distinguish four layers of QoS: User QoS, Application QoS, System QoS and Network QoS. The user QoS parameters describe requirements for the perception of multimedia data at the user interface. The application QoS parameters describe requirements for the application services possibly specified in terms of media quality (like end-to-end delay) and media relations (like inter/intra-stream synchronization). The system QoS parameters describe requirements on the communication services resulting from the application QoS. These may be specified in terms of both quantitative (like bits per second or task processing time) and qualitative (like multicast, interstream synchronization, error recovery or ordered delivery of data) criteria. The network QoS parameters describe requirements on network services (like network load or network performance).

Multimedia applications negotiate a desired QoS during the connection setup phase either with the system layer or possibly directly with the network layer, if the system is not able to provide QoS for the application. If both of them are not capable of providing the desired QoS, many of today's multimedia applications try to set up an end-to-end connection and to take care of QoS by themselves. Alternatively, data is transferred with the best effort approach (see e.g. the various Internet schemes). This may happen even if the network is able to reserve a specific amount of bandwidth (but less than demanded) for multimedia applications.

Therefore, resource management systems that provide mechanisms for media data streams with guaranteed or statistical QoS have become a key issue ([Wolf96][NaSt95]). Those systems take care of the coordination of media streams and the interfacing between layers of protocol stacks as well as further mechanisms (like process and bandwidth scheduling) in order to enforce the appropriate data handling. Most of the involved mechanisms are developed for a completely error-free presentation of continuous-media data at the user interface. In today's networked environments we still encounter many data paths over networks and via communication protocols which are not capable of providing a guaranteed real-time service. In such set-ups it is a key issue to decide which data item must be presented at the user interface and which data items may be discarded. The approaches for this are known as "scaling" and "filtering" of media data streams.

In the following section we summarize the most challenging and most often found application driven demands which have impacts on multimedia communication issues. The principal issues for QoS provisioning are described in Section 3. Subsequently we survey examples of QoS in communication systems in Section 4. Resource reservation is typically seen as a very important concept for QoS provisioning, Section 5 introduces the reader to reservation issues, and Section 6 provides examples of reservation mechanisms in communication systems. Adaptive methods such as scaling and filtering can be seen as an alternative to or an enhancement of resource management based QoS; their concepts are discussed in Section 7. Methods for 'Reservation in Advance' are needed for several scenarios to create distributed multimedia applications which resemble today's systems; they are described in Section 8. Before we conclude the paper, we discuss the ability of current systems to support distributed multimedia applications and consider which issues are still missing in Section 9.

2 Requirements of Distributed Multimedia Applications

Distributed multimedia applications have several requirements with respect to the service offered to them by the communication system.[1] These requirements depend on the type of the application and on its usage scenario. For instance, a non-conversational application for the retrieval of audiovisual data has different needs than a conversational application for live audiovisual communication (e.g., a conferencing tool). The usage scenario influences the criticality of the demands. For example, a home user video-conference, say between parents and children, is not as critical as a video-conference used as part of remote diagnosis by a physician.

Furthermore, the requirements of applications regarding the communication services can be divided into traffic and functional requirements. The functional requirements are multicast transmission and the ability to define coordinated sets of unidirectional streams. The traffic requirements include transmission bandwidth, delay, and reliability. They depend on the used kind, number, and quality of the data streams. For example, a *bandwidth* of 1.5 Mbit/s is typically required for the playback of MPEG-1 coded video. The end-to-end *delay* is more stringent for conferencing than for playback applications. In the former case, the delay should not be more than 150 ms. The play out of related streams must be done in tight synchronization (< 80 ms skew), hence, appropriate measures must be obeyed during data transfer. On the other hand, the *reliability* requirements are sometimes lower than for traditional communication applications, e.g., if a fault-tolerant data encoding scheme is used. Furthermore, retransmissions (which are used traditionally for the provisioning of reliability) increase the end-to-end delay and are often worse than lost data for multimedia applications.

The traffic requirements can be satisfied by the use of resource management mechanisms. They establish a relationship between transmitted data and resources and ensure that the audiovisual data is transmitted in a timely manner. For this, during the transmission of data, the information about the resource needs must be available at all nodes participating in the distributed applications, i.e., end-systems and routers. Hence, resources must be reserved and state must be created in these nodes which basically means that a connection is established. This connection should then be used for the transmission of data, i.e., using a fixed route, because resources have been allocated on that path.

For various multimedia applications, especially in the conferencing realm, multiple receivers are interested in receiving the same data. For instance, in a talk distributed via the network, all listeners must receive the same data. Sending each person a single copy wastes resources since for parts of the path from the sender to the receivers, the same nodes are traversed. Thus, multicast should be used which provides for the transmission of a single copy of data to multiple receivers (Figure 1). In addition to reduced network load, multicast lowers also the processing load of the sender. Multicast must not be limited to a single sender; in conferencing scenarios, it is usual to have several senders which normally do not use the resources at the same time (e.g., only one person is speaking). Hence, mechanisms for `m:n` multicast allow for even reduced resource demands.

1. Other application areas than distributed multimedia applications have related service requirements, e.g., plant and other control systems, or large-scale simulations where the overall progress depends on the availability of single results.

Figure 1: Unicast vs. Multicast.

The delivery of audiovisual data to large receiver groups, as, for instance, the distribution of IETF meetings over the MBone, must also take into account that the resource capabilities towards and at the participants can vary widely – from high-speed network links and fast workstations to low-end PCs connected via relatively narrowband links. Therefore, support for heterogeneous systems must be provided, heterogeneous with respect to networks as well as to end-system capabilities. An approach to handle this heterogeneity is the filtering of data streams – dropping data in the network which can either not be transmitted due to a lack of bandwidth or which cannot be presented by the end-system due to a lack of compute power.

3 Quality of Service

A system designed for the presentation of audiovisual data must take timing constraints into account which are due to the characteristics of the human perception of such information. Therefore, an overall QoS must be provided to ensure that such constraints are fulfilled. Since distributed multimedia applications need end-to-end QoS, all hardware and software components participating in this process (retrieving, transmitting, processing, and displaying the data) must handle the data accordingly – from the local resources at the sender side via the transport system, including all network components, to the local resources at the receiving side. This applies to end-systems, servers, and networks as well as to system software and applications.

Most of the participating resources are shared among users and various processes. One approach would be to (over-) design them based on peak demands such that collisions between demands of multiple applications can never occur. Then it would not be necessary to provide any resource management functionality. Yet, such a scenario would result in huge costs and low resource utilization and, hence, is typically not practical. Thus, to provide a constant QoS during the run-time of an application, resource reservation and scheduling techniques must be applied. Another technique is to use filtering and scaling mechanisms which adapt the generated workload to the available resources by changing the characteristics of the transmitted data stream, e.g., lowering the frame rate of a video stream. Such mechanisms allow a smooth decrease in quality and are described in Section 7. Here, we discuss resource management techniques based on reservation and scheduling.

3.1 QoS Provisioning Steps and Components

In order to provide QoS by using resource reservation and scheduling, the following steps must be performed in turn at each system and component participating in the end-to-end application:

- *QoS specification*: the workload (i.e., the amount of traffic) and the expected QoS (e.g., the delay) must be specified to enable the system to determine whether and which QoS can be provided.
- *Capacity test* and *QoS calculation*: When an application issues its QoS requirements, the admission control of the system must check whether these demands can be satisfied taking existing reservations into account. If so, the best-possible QoS which can be provided is calculated and the application is given a certain QoS guarantee accordingly.
- *Reservation* of resource capacities: According to the QoS guarantees given, appropriate resource capacities, as, e.g., transmission or processing bandwidth, must be reserved.

- *Enforcement* of QoS guarantees: The guarantees must be enforced by the appropriate scheduling of resource access. For instance, an application with a short guaranteed delay must be served prior to an application with a less strict delay bound.

This functionality can be divided into two distinct phases. The set-up phase (also called 'QoS negotiation') consists of the first three steps. The specified QoS requirements 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').

If a connection-oriented approach for the provisioning of QoS during data transmission is used, the *QoS negotiation* steps are typically part of the connection setup. If no connections but (soft-state based) flows are used, these steps are performed as part of the flow setup, they mark nevertheless the beginning of QoS support for data transmission because no QoS can be provided without these reservation.

Overall, several resource management components interact to provide QoS assurance: Applications, QoS translators, admission control, resource scheduler. Additionally, further components are needed, for example, a resource reservation protocol to communicate QoS specifications among participating systems and a resource monitor which measures the availability of resources and whether indeed the promised QoS is provided.

3.2 QoS Classes and Layers

Several classes of QoS are typically distinguished, the extreme on one side is a hard "guaranteed QoS" where the reservation is based on peak requirements, the other is the "best-effort" approach where no reservation is made at all (and should therefore, at least with respect to QoS provisioning, better be called "no-effort"). Between these exist various forms of weaker QoS (statistical, predictive) based on average case and predicted assumptions. The hard QoS guarantee requires more resources and is, hence, more costly, than the 'weaker' approaches. Yet, in the latter cases, resources may not be available when needed for processing leading to worse quality.

The notion of QoS is very different at the various system layers. At the application layer, for instance, QoS parameters are based on media quality descriptions and requirements, e.g., for the workload parameters such as frame size (e.g., height, width, color specification) and frame rate may be used; the end-to-end delay relates to the final presentation of data to the user and loss might be specified in terms of 'visibility' to the user. In the network, the workload description consists of packet size and rate; loss may be described using measures for bit error or packet error rate. Each of these layers needs the QoS specified in its own terms which means that a mapping between them is necessary. While this mapping is an important issue for all networked multimedia applications, no overall solution has been found yet, but only partial approaches for simple conversions, e.g., between transport and network layer, have been devised.

In addition to this layer dependence, the notion of QoS is qualitative and quantitative. For example, delay, throughput, rate, and buffer specifications are quantitative parameters on different architectural layers whereas interstream synchronization, ordered delivery and error recovery are qualitative parameters on different levels of abstractions. Some of the qualitative parameters, such as lip synchronization, can be mapped to quantitative parameters if user characteristics are taken into account.

3.3 QoS Specification

In general, three QoS parameters are of main interest with respect to the transport of continuous-media data: *bandwidth, delay* and *reliability*.

Bandwidth, as the most prominent QoS parameter, specifies how much data (maximum or average) is to be transferred within the networked system. In general, it is not sufficient to specify the rate only in terms of bits, as the QoS scheme shall be applicable to various networks as well as to general-purpose end-systems. For instance, in the context of protocol processing, issues like buffer management, timer management, and the retrieval of control information play an important role. The costs of these operations are all related to the number of packets processed (and are mostly independent of the packet size), emphasizing the importance of a packet-oriented specification of the data rate. Information about the packetization can be given by specifying the maximum and the average packet size and the packet rate. **Delay** as the second parameter specifies the maximum delay observed by a data unit on an end-to-end transmission. **Reliability** pertains to the loss and corruption of data. Loss probability and the method for dealing with erroneous data should be specified.

All three QoS parameters are closely related: The smaller the overall bandwidth of a resource is compared to its load, the more messages will be accumulated in front of it and the larger the buffers need to be to avoid loss. The larger the buffers become, the more likely messages need to wait to be serviced, that is, the larger the delay will get.

Jitter, the delay variance, is the fourth QoS parameter typically considered. It is the result from varying delays during processing and transmitting the data. It can be smoothed by buffering at the receiver side which, however, increases the end-to-end delay.

The parameters used within the workload description specify *what amount* of data the source intends to transmit. This must be viewed in the context of a *workload model* which specifies *how* the source generates data and feeds it into the system. One example for a workload model is the *Linear Bounded Arrival Process (LBAP)* model [Cruz91]. In that it is assumed that the data to be sent as a stream of packets is characterized by the three parameters:

- S = maximum packet size,
- R = maximum packet rate (i.e., maximum number of packets per time unit), and
- W = maximum workahead.

The workahead parameter W allows for short-term violations of the rate R by defining that in any time interval of duration t at most $W + t*R$ packets may arrive as part of a stream. This is necessary to model input devices that generate short bursts of packets, e.g., disk blocks that contain multiple multimedia data frames, as also to account for any clustering of packets as they proceed towards their destination (for work conserving systems).

When describing QoS demands, it is useful to specify an interval [*required, desired*] in which the QoS provided by the system shall lie, i.e., a minimum QoS below which the application cannot run properly and a maximum QoS that is needed by the application for returning a very good quality. In comparison to the use of a single value for QoS demand specification such an interval provides for more flexibility and, hence, increased acceptance probability.

3.4 Role of Resource Reservation Protocols

Besides the local resource management mechanisms at the participating end systems and routers, reservation protocols are needed to exchange and negotiate QoS requirements between these systems. These demands are accumulated in a *FlowSpec* (Flow Specification). Reservation protocols perform no reservation of required resources themselves. They are only the vehicles to transfer information about resource requirements and to negotiate QoS values between the end-systems and the intermediate network routers – they leave the reservation itself to local resource management modules. The initiator of a resource reservation is not necessarily a sender in the enforcement phase. E.g., with RSVP the reservation initiator is the data receiver. Nevertheless, the network nodes need to know always the direction of data flows for making reservations, e.g., for physical transmission lines with asymmetric capacity, and generally, for asymmetric reservations.

In Section 5, we will discuss such protocols and their use for end-to-end QoS provision in some more detail.

3.5 Interoperation of the Involved Modules

The individual resource management systems need not necessarily work identically. They must be able to communicate using reservation protocols and should have a similar understanding of QoS specifications to avoid errors which might occur if specifications are translated between various forms. The QoS requirements may be mapped to resources in different ways at distinct nodes. In order to maintain QoS, subsystem builders establish a service policy. E.g., implementations might re-use the resource reservation mechanisms for, say, peak rate services until the definition of newer, say, VBR, services is finalized and a viable policy has been investigated.

3.6 QoS Models

A wide-variety of QoS models and architectures has been developed, e.g., Tenet (at UC Berkeley and ICSI) [BFMM94], HeiTS/HeiRAT (at IBM ENC, Heidelberg) [VoHN93][VWHW97], QoS-A (at Lancaster University) [CaCH94], etc. In the following we briefly describe the approach followed in the Internet community due to its foreseen influence on the future use of the Internet.

The Integrated Services (IntServ [RFC1633]) activity approaches, in relation with the work on the RSVP protocol, a general solution for QoS guarantees in the future Internet. The RSVP protocol is used to transport FlowSpecs that adhere to Intserv rules. Two types of descriptions are used for the QoS specification: the traffic specification (TSpec) describes the behavior of a flow, and the service request specification (RSpec) describes the service requested under the condition that the flow adheres to the restrictions of the TSpec.

On this basis, various services are defined. Currently, Guaranteed QoS and Controlled-Load service are under investigation, others have been discussed but are not further studied at the moment.[2] The specification of guaranteed QoS requests that the maximum end-to-end delay of a packet shall be strictly limited to the given value in the RSpec, under the condition that the flow sticks to a certain traffic pattern. In the average behavior it is limited by a token bucket model, in peak behavior it is limited by a peak rate parameter p and an interval length T so that no more than p*T bytes are transmitted in any interval T, and in packet size it is restricted by a minimum counted size m (all smaller packets are considered to be of size m) and a maximum size M (all larger packets are considered to be a violation of the contract). This specific service is supposed to be useful for applications with hard real-time restrictions such as audio transmissions. If all hops of a communication path for which this service is

2. Unfortunately, the IntServ working group has not published results yet. The information given here is based on the following internet drafts: the service template is presented in "Network Element Service Specification Template", (S. Shenker, J. Wroclawski, Internet-Draft, Work in Progress), the guaranteed QoS in "Specification of Guaranteed Quality of Service" (C. Partridge, S. Shenker, R. Guerin, Internet Draft, Work in Progress), the controlled-load service in "Specification of the Controlled-Load Network Element Service" (J. Wroclawski, Internet Draft, Work in Progress).

requested accept the service request, it is also ensured that the communication is lossless because the queue size reserved for the flow in each router can be set to the length parameter of the token bucket.

The specification for controlled-load service requests that all network elements behave under any circumstances for a reserved flow (that describes its traffic characteristics) as they would for a best-effort flow in a situation of light load and without congestion. The TSpec for this service is identical to that of the guaranteed QoS. An RSpec is not defined. The application of this service for multimedia communication is considered useful because multimedia applications such as some video conferencing systems have proven to work well on lightly loaded networks without reservations although they fail under heavy load. For such applications, this service can be used to simulate conditions of light load.

For being used with RSVP, all of the service models based on the Intserv template have to provide a comparison and merging function for their traffic and request specifications. This is necessary (firstly) because of the Intserv model, in which flows can always have multiple sources that share resources. Secondly, this is necessary because RSVP is a receiver-oriented protocol, which implies that individual receivers may issue QoS requests that differ strongly from the requests of others but that have to be merged at an intermediate hop.

4 Quality of Service in Communication Systems

4.1 Local-Area Networks

QoS can only be guaranteed in a networked environment if it is supported at the data link layer of a communication system. The widespread Ethernet networks have never been able to guarantee any kind of QoS due to the indeterminism of the CSMA/CD approach towards sharing of network capacity. The reason for this is that the collisions, which occur in the bus-based Ethernet if two stations start sending data on the shared line at the same time, lead to delayed service time. Although even the 10 Mbit/s version can sustain one single high-quality (Main Layer – Main Profile) MPEG-2 video stream, an interference of data traffic to other end-systems on the same network cannot be prohibited. Using Ethernet switches (Figure 2), instead of a bus-based topology, a star-wired network topology can be configured where each link connects a single end-system to the switch. Then, collisions can only occur inside of the switch and can be resolved by appropriate design of the switching unit. If an end-system participates in two or more concurrent communication sessions, one transmission can be delayed or aborted. Hence, the probability of collisions is reduced but guarantees cannot be given. A different approach would be to avoid collisions by controlling the beginning of transmissions from each station, e.g., in software using tightly synchronized clocks.

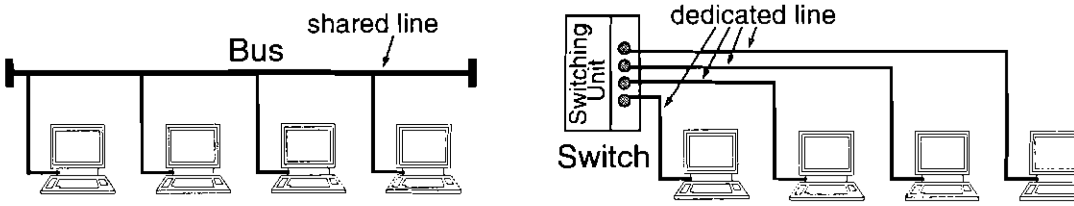

Figure 2: Bus-based Ethernet vs. Switched Ethernet

Fast Ethernet exploits the same configuration as 10 Mbit/s Ethernet and increases the bandwidth with the use of new hardware in hubs and end-stations to 100 Mbit/s but provides no QoS guarantees either. Isochronous Ethernet (Integrated Voice Data LAN, IEEE 802.9) and Demand Priority Ethernet (100Base-VG AnyLAN, IEEE 802.12) can provide QoS, yet their market potential and influence are questionable [Stue95].

Token Ring networks with 4 Mbit/s and 16 Mbit/s and the similar 100 Mbit/s FDDI networks are both capable to support QoS to some extent with the help of management applications for central admission control. By exploiting the priority mechanisms of the Token Ring networks, a limited number of multimedia data streams can be delivered with a higher priority than the regular traffic ([VoHN93], [VWHW97]). The priority mechanism guarantees that the sender with the higher priority can always acquire the token to send a frame, but it guarantees also that the token must be released for use by the station with the next lower priority after transmitting at most one frame of the maximum MTU size. A similar approach is used by FDDI's synchronous mode.

4.2 Network Layer – the Internet Protocol

The provision of QoS has not been considered in IP Version 4, today's most important network layer protocol. It is designed to provide flexible, self-repairing communication. The type-of-service field in the IP header is typically unused; furthermore, the options in this field give more indications than detailed information usable for QoS support. Overall, support for continuous media was not an issue at the time of IP's design. A few details, however, have been included in the meantime. Multicast communication is provided by defining a set of multicast addresses in the IP address space. Multicast groups are maintained by adding and removing IP addresses to and from the multicast group using IGMP [RFC1112]. A multicast-capable router uses link-layer multicast support to forward a packet or, if that is not possible, forwards the data packet to multiple destinations.

Version 6 of IP, the successor to the current IPv4 protocol, does not contain QoS support by itself but has been equipped with hooks which can be used by other means to set up reservations. The concept of a pseudoconnection, called *Flow*, is introduced which is a packet stream between sender and receivers. Flows may be established by means external to IPv6, e.g., by a reservation protocol such as RSVP. The header of each packet contains a *Flow Label* which can be used to identify for each packet to which flow it belongs. After a router has determined the flow a packet belongs to, it can identify the QoS to be supported and the resources allocated for its processing. In addition to the flow concept, IPv6 has a priority field which can be used by routers to process packets according to their urgency, e.g., high-priority packets containing data of interactive applications are preferred over low-priority traffic.

4.3 Real-time Transport Protocol

One of the Internet protocols that can be used in conjunction with reservation models at the network layer is the Real-time Transport Protocol (RTP) [RFC1889]. RTP is an end-to-end protocol for the transport of real-time data. An important application type supported by RTP is multi-party conferencing because of its support for synchronization. framing, encryption, timing and source identification. RTP has its companion RTP Control Protocol (RTCP), which is used to interchange QoS and failure information between the QoS monitor applications in the end-systems.

RTP does not define any kind of QoS itself and does not provide re-ordering or retransmission of lost packets. However, it provides a sequence number that enables the application using RTP to initiate such steps. RTP is typically used directly on top of UDP/IP or on top of ST-2. In the former case, QoS can be guaranteed by the use of RSVP's reservation mechanisms for the UDP datagrams. In such a combination, the RTP stack provides the information necessary to make educated guesses about the behavior of the data stream based on RTP's knowledge of the data format. In addition to the base RTP specification, a number of companion documents exist that provide encapsulations for various continuous media formats such as M-JPEG or MPEG. Hence, RTP itself provides no real QoS support; it relies on other appropriate protocols and mechanisms.

4.4 Telecommunication Systems

Outside the Internet, wide-area multimedia communication is generally based on telecommunication networks. Analog telephone systems play only a minor role in multimedia communication due to their limited bandwidth. For wide-area connections, the bit failure rate in analog communications is exceptionally high, which may become fatal even for multimedia streams. Without these problems, the connection-oriented approach of telephony would be well-suited for multimedia communication.

ISDN (Integrated Services Digital Network) is replacing analog telephony, at least in Europe and East Asia. As the name states, its goals are to provide for the integration of various services besides telephony. The bandwidth definitions for ISDN are made according to SDH (Synchronous Digital Hierarchy). The smallest data channel in this hierarchy is the B-channel with a fixed guaranteed data rate of 64 kbit/s – this way, an audio sample is transmitted every 125μs. The data transmission is frame-oriented with variable-length frames. In contrast to the Internet protocols, ISDN uses out-of-band signalling as usual in telephony. The D-channel provided for that signalling offers a data rate of 16 kbit/s. The signalling traffic often does not fully occupy the D-channel. The spare capacity is used for additional data transmission services, but with lower priority than for signalling.

The *basic rate access* for telephony in Europe brings two B-channels and a D-channel to the end-user. An arbitrary number of end systems can be connected to the network terminator, but only 10 can be addressed individually. Point-to-multipoint connections are supported for B-channels. Multiplexing multiple connection on a single B-channel is not supported, restricting the number of simultaneous data connections to two. Additional data packet services via the D-channel remain available even when both B-channels are in use. A guaranteed throughput of up to 128 kbit/s can be achieved by combining the two B-channels.

The *primary rate access* can be used to increase the number of parallel connections or to support a higher data rate. In Europe it comes with 30 B-channels, one D-channel for signalling and one channel for synchronization, each offering a data rate of 64 kbit/s. Some combinations of B-channels are supported to form a single channel providing for a higher throughput. The H0-channel at 384 kbit/s is typically used for commercial video conferencing systems that are based on ISDN. The combination of all 30 B-channels of a primary rate access into a single channel with a throughput of 1920 kbit/s is also supported and is named H12-channel.

Because of the connection-oriented approach with guaranteed throughput and low loss rates, ISDN-based long-distance video-conferencing is actually more common today than conferencing over computer networks. But ISDN is supposed to be only one step on the way to better service integration and higher data rates in the future Broadband-ISDN (B-ISDN). The technological basis for this is ATM (Asynchronous Transfer Mode) technology.

The development of ATM is motivated by the merge of the computer networks and the telecommunications approaches towards multimedia communication. Two groups work on the standardization of ATM: the ITU-T, which is the international standardization organization for telecommunication, and the ATM Forum, which is a consortium of industrial and research organizations. Where data communication issues are concerned, the ATM Forum is currently the standardization committee with the higher relevance. It can be expected that the ITU-T aligns its work with the proposals of the ATM Forum. Consequently, in case there are differences between the definitions of ITU-T and ATM Forum, the approaches of the latter are presented in this paper.

ATM is connection-oriented and uses a small-size basic unit called *cell*. ATM does not ensure that no cells are lost, yet it guarantees that the cell order is always maintained in a connection. QoS is conceptually negotiated between three entities: the calling party (initiator of the connection), the network and the called party. The calling party requests a connection to the called party with a SETUP message in which it provides its QoS requirements to the network and to the called party.

The QoS parameters supported for ATM connections (which use identically-sized cells) differ slightly from those that are considered in networks that are based on variable-length packets. The ATM parameters are:

- Sustainable rate: The minimum number of cells per seconds that must be supported by the network for the entire length of a connection.
- Peak rate: The number of cells that must be expected at each node in the network in rapid succession (one burst).
- Maximum burst length: The length of an interval in which at most one burst must be expected by a network node.
- Cell loss ratio: The maximum rate of lost or corrupted cells that an application can accept for a connection.
- Maximum end-to-end delay: The restriction on the sum of all waiting times that each cell can spend in the queues between the sender and the receiver of the cell.
- Maximum cell delay variation: The maximum difference in end-to-end transmission time that two cells of a connection can experience.

The models that have been applied in the ATM area to guarantee QoS have changed greatly during the development of ATM. While the ITU specified 4 service classes in [I.362], the definition that is relevant for current implementations is provided by the ATM Forum. It can be found in version 4.0 of the ATM Traffic Management Specification [AtmF96]. Five *service categories* are distinguished:

- UBR - unspecified bit-rate
- ABR - available bit-rate
- nrt-VBR - non-real-time variable bit-rate
- rt-VBR - real-time variable bit-rate
- CBR - constant bit-rate

The distinction of the two best-effort service categories UBR and ABR is not intuitively clear. The UBR service imposes the weakest requirements on the network. It is appropriate for all non-real-time applications that emit discrete blocks of data without relevant restrictions on the end-to-end delay or jitter. To provide such a service, it is not necessary to allocate any resources for the entire life-time of a connection. Similarly, no information on delay, jitter or loss requirements are negotiated. It may be a good choice to provide a peak rate value for a connection although it is not necessary for using this service. This specific value is used by the user-to-network interface for shaping the traffic to stay below this limit. If the requested peak rate is accepted by the network, it guarantees that all nodes in the network can handle the succession of cells that is inserted into the network. An excessive cell loss by overloading of an intermediate node of the network is prevented by this practice.

The ABR service is a modification of the UBR service; it has been specifically designed to minimize loss and to provide fairness among connections. In order to do this, the network applies a rate-based feedback flow control mechanism, which works as a throttle at the sender side of a connection. Depending on aspects such as the application and the network configuration, this service may offer a better flow control mechanism in an ATM environment than an end-to-end window-based flow control mechanism.

The third non-real-time service, nrt-VBR, requires the specification of a sustainable rate and a peak rate for a data stream as well as a loss rate. For this service, the implementation in the switches is not supposed to be optimized for delay or jitter reduction. The rate specification is included at connection setup time to improve packet loss and the end-to-end delay. The reason for this is that making a reservation for a connection without real-time demands preserves some bandwidth from being allocated by another reservation. The switch must adhere to the traffic contract and must process the cells in the queue before an overrun can occur. Otherwise, the switch would violate the loss requirements specified in the contract. Thus, the overall delay experienced by the application is limited although no specific delay requirements are specified in the traffic contract.

Two service classes are available for transmission of time-critical data: rt-VBR and CBR. The CBR service has been part of the service descriptions from the initial designs of the ITU. It exists for the transmission of data with real-time requirements which is generated by sources that insert a continuous stream without relevant bursts into the network. This kind of traffic can be described easily by a fixed rate that remains the same for the connection period. Further parameters in the QoS specification are the acceptable loss rate, the maximum end-to-end delay and the maximum jitter.

Rt-VBR has been specified by the ATM Forum for those applications with real-time requirements which would reserve excessive amounts of bandwidth if they would use the CBR service. For example, this is the case in a conferencing system that applies a variable bit-rate encoding such as Motion JPEG. Since some loss is acceptable in such a scenario, statistical multiplexing schemes can be applied in the switches for this service category. At connection setup time, the application can specify its requirements in some or all of the supported parameters: acceptable loss, maximum delay and jitter, sustainable rate, peak rate and maximum burst size. In case that the application does not accept any loss, the only difference from using the CBR service is that UBR and ABR connections can be accepted by the network in addition to a full load of CBR and rt-VBR connections. In contrast to the CBR service, a difference between sustainable and peak rate allows the calculation of remaining throughput that can be used for non-real-time connections.

Currently under discussion are various services that support re-negotiation of QoS. Renegotiation means that after a connection is established and an initial QoS is guaranteed, the network is able to handle end-system requests which ask for a modification of the provided QoS. The reason for the introduction of such an ability is the fact that traffic characteristics in real-time applications may change over time. If these changes are predictable but large (e.g., this is the case for a stored movie with scene changes), a reservation based on the initial service categories is always excessive, at least in its peak rate reservation. If these changes are not predictable (e.g in a conferencing scenario), the initial reservations may be either excessive or too low. In both cases, a renegotiation mechanism can reduce excessive reservations for the entire time that a connection exists. The approaches that are currently under discussion in the ATM Forum are labelled RCBR and RVBR, respectively. Even if such services will be specified, the original CBR traffic will still remain important because some applications exist that can not accept that a renegotiation towards higher resource demands fails. For these applications, the CBR service is the only viable choice.

5 Resource Reservation Issues

Internet services have historically always been connectionless, fair, and unreliable on a hop-by-hop basis because of IP's philosophy and characteristics. The discussion whether reservation is necessary or not has been going on for many years. The defenders of reservation mechanisms point out the pros for the user in terms of reliability and individual throughput. They argue that multimedia traffic cannot be supported in today's networks and that this is a problem that needs new technical solutions.

The critics of reservation pointed out that bandwidth will shortly be large enough to accommodate all network traffic without congestion, that reservation makes the Internet unfair, or that reservation mechanisms lead to waste of resources (that are reserved but unused). This specific set of arguments has been silenced in the last few years due to the explosion of the Internet in the private and commercial sector, which has turned into unexpected increase in traffic and thus, congestion. It seems now that any kind of bandwidth that can be provided in the visible future will be drained by the new services without a reduction of congestion.

The more viable argument seems to be the assumption that reservation mechanisms based on Internet protocols will become redundant because all long-range traffic will be handled by ATM networks, which define their own reservation mechanisms. Unfortunately for this argument, many existing networks of the Internet are not ATM networks but will continue to exist for some time. If a reservation in an ATM or any other backbone should be made for an application with end-systems attached to a non-ATM network, this still requires non-ATM protocols for the communication.

In Section 3 we already described the components involved and the basic steps performed in order to provide QoS support using resource reservation. In this section, we discuss the basics of reservation protocols which are used to exchange information about resource requirements and desired QoS.

5.1 Reservation of Resources

The decision whether an incoming reservation can be accepted is made within the admission control module of the resource management. The resource management must internally maintain the overall state of the node's resources as well as store sufficient information of the original requests to be able to release and modify reservations later (on request, in case of preemption, or in case of an unannounced breakdown). While requests dealing with only one data stream can be handled by the admission control of the network node, the interaction of more than one stream is subject to policy control where precedence among streams are considered, e.g., streams may be preempted in presence of resource requests of higher priority.

The node must map the functions of the supported reservation protocols onto the internal representation of the network node's resource management system. Furthermore, it needs such functions as activity monitoring, extracting information from the routing module, adding/removing an end-system to/from an active stream or modifying the characteristics of a stream. Parts of these are protocol-dependent functions, e.g., adding an end-system to a stream.

It would be beneficial if a node's resource management system would not be tied to a specific reservation protocol, but if various protocols could be used easily, e.g., employing a particular protocol for a specific application scenario. In theory, resource reservation and scheduling are indeed protocol-independent activities. Yet, the model chosen for the characterization of the workload influences the services which can be offered, and today's reservation protocols are not completely independent of such models. Therefore, it is not possible at the moment to support different reservation protocols within one node unless the basic models and assumptions used within the protocols are quite similar.

5.2 Reservation Styles

Reservation is applicable in many areas of communication. Depending on the particular application, the reserved stream may involve two or more parties which can be organized in one of the following approaches:

- Single Sender / Single Receiver (Unicast)
- Single Sender / Multiple Receivers (1:n Multicast)
- Multiple Senders / Multiple Receivers (m:n Multicast)

The latter two approaches are combined with the notion of group concept. The single 1:n multicast leads to a multicast tree. If m different systems want to send data, m different multicast trees must be established, leading to considerable effort in the network nodes and communication paths. If now only a few of these m sources actually transmit data, say k (k<m), only resources for k instead of m trees are needed. Hence, the m:n multicast allows for a better sharing of resources by establishing the routes for m independent multicast trees but reserving the capacity for at most k concurrent transmissions. A similar sharing could be achieved by a stream grouping scheme, where the membership within a particular group must be announced during the tree establishment.

A difference among current reservation protocols is the direction in which the reservation occurs, whether it is sender-oriented, receiver-oriented or neither. Sender-oriented means that the sender of the data is the entity that initiates the reservation setup. Therefore, the sender must know the receiver addresses for the reservation setup. In contrast to this, in the receiver-oriented approach the receivers must know the address of the data source. The sender may have no knowledge of the participating receivers; the availability of this information for the sender is often not necessary but might be useful for some applications. Furthermore, information about the resources that are required for the data transmission must be given to the receivers to enable them to perform adequate reservations. Finally, the reservation might be considered as a network management issue done by a third party operating as a mediator between the senders and receivers; this entity must be informed about senders, receivers and QoS specifications and is a uncommon approach.

The sender is the entity which starts the overall setup process in the sender-oriented as well as in the receiver-oriented approach; this is obvious for the former case, in the latter case, as stated above, the sender must give sufficient information about the data streams to the network and to the receivers to enable them to perform appropriate reservations. Hence, the difference is more when the reservation is set, in the sender-oriented approach this is done on the first pass (from sender towards receiver), in the receiver-oriented approach it is performed during the second pass (from receiver towards sender).[3] This leads to a trade-off of (i) blocking some resources on the first pass which will be released on the second pass versus (ii) encountering a certain reservation rejection probability because someone else might have reserved all resources in the meantime between the first and the second pass.

It is an important issue whether management tasks, e.g., adding a receiver to a multicast tree, are performed in a central or in a distributed manner because it affects the scalability of the system. Sender-oriented reservation can lead to substantial management workload at the sender if a large number of receivers participate in the transmission and generate control messages to be processed by the sender such as 'join' or 'leave' operations. This problem restricts the scalability of sender-oriented reservations to small-to-medium size. As a solution, sender- and receiver-oriented reservations can be combined, e.g., the reservation starts with a reservation for some receivers, later, additional receivers who want to join the transmission can be added at routers without involvement of the sender [DHHS94].

A reservation protocol can be tightly coupled to a specific protocol used for data transmission and require that reservation setup precedes data transmission. Alternatively, the transmission may be independent of reservation setup. The former approach is a typical connection-oriented communication. The latter approach promises more flexibility, yet, it must nevertheless be possible to extract information from the packets which can then be used to determine the reserved resources for scheduling purposes. This approach is followed by RSVP which exploits information from IP packets, yet, IP can also be used without any reservation made by RSVP.

3. The 'One-Pass With Advertising' approach [ShBr95] is subsumed here under the two-pass approach where the first pass distributes information only but performs no reservation.

6 Reservation in Communication Systems

The previous section addressed the principle issues of reservations. Now we describe two resources reservation protocols which have been defined within the Internet community and also ATM's approach to resource reservation.

6.1 Stream Protocol

ST-2+ (Stream Protocol Version 2+) [RFC1819] is a connection-oriented reservation and transmission protocol. It forms a sender-oriented, unreliable, multicast protocol which supports the uni-directional delivery of data from the initiator of the *stream* (the connection) to all receivers of the stream (hence, a 1:n multicast). QoS is negotiated through the exchange of FlowSpecs between all nodes participating in a stream (sender, routers, and receivers). Once a connection is established, the sender may start sending data; transmission without connection setup is not possible (IP is not used for data transmission). The negotiated QoS and the multicast tree can be modified later. A receiver can leave the tree or the sender may drop a receiver; more receivers can be added by the sender and via receiver-oriented stream joining. To provide heterogeneous QoS to different receivers, a filtering approach as discussed in Section 7.2 could be used; yet, this is not defined in the basic protocol specification. To support m:n multicast scenarios, the notion of groups of streams has been defined – sharing bandwidth, routes, subnetwork resources or 'fate' (which means that if one stream is preempted in a node, all other should also be terminated). Yet, the specification for actual group mechanisms is only limited.

ST-2+ (and its predecessor ST-2) have been used for experiments and within some products in the first half of the nineties. A different reservation protocol, RSVP, found broad industry support since that time. Therefore, the importance and use of the ST protocols diminished.

6.2 Resource ReSerVation Protocol

The Resource ReSerVation Protocol (RSVP) [ZDES+93] is a reservation protocol in the Internet suite which is used to transport FlowSpecs (that adhere to Intserv rules) between resource managers. RSVP adds reservation to the existing and up-coming Internet protocols (IPv4 and IPv6) and relies on those protocols for the interchange of data. Although the primary background of RSVP can be seen within conferencing applications, its integration with the IntServ activity of the IETF demands that it solves the requirements of other applications and it is considered as a general solution for reservation inside the IntServ activities. Here we give only a brief overview about the concepts of RSVP and comment on its suitability for multimedia communication; a more detailed description of RSVP is given in another paper in this issue.

In RSVP, reservations are made for 'flows' which are identified by address information in the IPv4 header or by a flow label in the IPv6 header. During data transfer, a router which receives a packet checks to which flow it belongs and schedules the packet transmission in accordance with the reservation setup for that flow.

Instead of hard state controlled mainly by connection-setup and -release, RSVP keeps *soft state*. This state is setup by reservation messages as well, but it must be refreshed by reservation updates periodically, otherwise, if no such update is received for a while, the reservation times-out and the allocated resources are released.

RSVP's receiver groups may be large, and both the addition and removal of senders and receivers is supported. Filter specifications can be set to define which sender's packets can use the reservations made for a flow, hence, resources may be shared among senders. With the first versions of the RSVP specification it was also possible to define filters which could adapt the amount of data transmitted for a flow (forwarding the parts which matched a specified pattern and dropping the rest), this feature has been removed in later versions.

Reservations are made in a receiver-oriented style in RSVP – reservation requests are sent from a receiver of a flow towards a set of senders. In order to provide receivers with information about the flows which they want to receive and hence, to make appropriate reservations for that, senders advertise such information in a PATH message sent to all potential receivers. An end-system interested in that flow generates a RESV message (with a FlowSpec containing information about the desired QoS and the filter specification) which travels towards the sender along the reverse path of the PATH message. FlowSpecs can be merged in a router, e.g., if appropriate reservations from multiple receivers are requested. For this, a function must be provided that allows a network node to compute a FlowSpec that is larger than both of two given FlowSpecs.

Network nodes must store a large amount of information about RSVP flows such as the timers for each receiver of each flow which indicate when a reservation expires, the FlowSpecs of all receivers along with the incoming interface to calculate changes in a receiver's FlowSpec or to release resources in case of time-out or tear down, the merged FlowSpecs for each outgoing interface to identify when a change in a FlowSpec has an effect on an upstream network node, and the filtering information specified by each receiver for the source selection within a flow.

With RSVP each receiver decides itself how large a reservation it needs based on its own characteristics and requirements. This can lead to different reservations from independent receivers. Such heterogeneous reservations have often be remarked as an important feature of RSVP. Support for heterogeneity is necessary if multiple receivers with varying requirements or capabilities participate in a multicast session and where each receiver should get the best possible quality. Then different resource capacities must be reserved on the different paths towards these receivers. Yet, this must be combined with heterogeneity of the amount of transmitted data because a receiver is not interested in a reservation by itself. Received data and reservations are just means to get the data delivered at the right time. Therefore, to support heterogeneity, it must be specified which data should be forwarded and which should be dropped. Otherwise 'random' packet dropping might occur which can make the overall data stream useless, for example, the received data is worthless if always the I-frames of a MPEG video stream are discarded. Unfortunately, RSVP does not provide mechanisms for such a data discrimination.[4] Probably, most receiving applications will use the flow characteristics distributed by the sender on the first pass (resp. the advertising pass) for their reservation requests. Thus, the offered form of heterogeneity is unlikely but comes at the cost that merging must be supported by routers which increases the overall complexity.

Reservation requests in RSVP can fail even if the sender of a request is informed of its success. This is an effect of the combination of receiver-orientation and flow merging of reservations. If a second reservation request arrives at a hop on which a first reservation is already established, and the first reservation reserved a superset of the second one, the second reservation is considered successful, even if the first reservation has not yet been fully established up to the sender of the flow. If the first reservation fails at a later time, an error message is distributed to the receivers.

Communication system QoS support using reservations can only be successful (at least for deterministic guarantees) if the data transmission uses exactly the same route on which the reservations have been made (at least as long as no failure occurs). Thus, opportunistic re-routing – route changes due to slightly better metrics but which are not necessary – must be inhibited, i.e., 'route pinning' must be done, affecting the operation of IP. Route pinning has been discussed several times within the RSVP

4. As said above, the filters for adapting a flow have been removed from the protocol specification.

working group and is currently not provided. Therefore, a hard QoS guarantee cannot be given by RSVP.

Broken link failures are not handled by RSVP itself, but by IP which is used by RSVP for data transmission. In the case of a link outage, IP tries to find a different route for data forwarding which means that a 'route flapping' occurs. Data can now continue to flow via this alternative route, yet without QoS guarantee because no reservations have been made so far on that path. With the next exchange of PATH and RESV messages, reservations will be made along the new path (if sufficient resources are available). The reservations which have been set on the old path will be kept and the resources will stay allocated until they expire or until an explicit tear down is made (blocking unnecessarily resources for some time). If the link failure is only temporary and the route flaps back again then the data transmission can be served with QoS support immediately as long as the resources have not been released already.

6.3 ATM Reservation

In the connection-oriented ATM architecture, reservations are performed in two steps. First, the connection is negotiated between the originator of a connection, the targets and the network. Secondly, the resource guarantees are managed at each node of the network individually and any changes are adapted to the amount of resources that are made available to those connections that do not have a fixed reservation.

To establish a connection, the calling party specifies the resource requirements as described in Section 4.4 and it passes the request to the network. The Call Admission Control of the network determines whether the QoS requested by the calling party can be guaranteed at each intermediate node in the network. In actual implementations, each node checks the available resource capacity. It decides to permit or refuse the new connection depending on its service class. For example, a new call in the UBR service category can be accepted immediately, whereas a call in the ABR service category will result in a reduction of available throughput to other connections in this category. Therefore, it initiates a notification to all participants in active connections of that category. If the traffic in this service category exceeds the available resources, flow control mechanism are applied to throttle down all connections in the ABR category. Similarly, calls that require fixed reservations, such as a call in the CBR category, are accepted only if the resources can be guaranteed. In such a calculation, no unreserved resources for best-effort connections are put aside. If such a reaction is not possible, the new call is refused, other connections are not modified.

When a connection is established, a traffic shaper at the user side of the user-network interface controls the insertion of cells into the network, and the Usage Parameter Control (UPC) at the network side monitors the traffic which is generated at the sending station. Violations of the traffic contract by exceeding the negotiated cell rate or jitter are punished by tagging the non-conforming cells, which permits the UPC and all consecutive nodes along the path to drop them. An algorithm is used to specify a reference for the separation operation of the traffic shaper (in the end station) and the UPC entity (traffic policer at the network side at the UNI). The traffic shaper should not generate cells faster than allowed by the algorithm, otherwise they can be dropped or tagged by the traffic policer. To cope with the jitter at the insertion point, a network should define a fixed small Cell Delay Variation (CDV) tolerance. The traffic policer should not tag or discard cells arriving later than specified by this algorithm. The algorithm is known as Generic Cell Rate Algorithm (GCRA). It can be seen as a continuous-state Leaky Bucket Algorithm.

Furthermore, the reservations are supervised by the individual nodes in the network. Whereas early implementations of ATM switches handled all traffic in a FIFO manner, newer implementations apply a policy of isolation and sharing. Queueing at the input ports is untypical for the current generation of

ATM switches. Cells are queued at the output ports, and the policy is applied only to these queues. At such an output port, two queues can be used to handle two different priority levels in order to isolate the real-time traffic from the non-real-time traffic. This provides a partial isolation of the two very different classes of data. If both variable bit-rate classes are supported and they are handled by different algorithms for statistical multiplexing in the switch, then these two priorities are not sufficient. Instead, two additional queues are required to isolate the VBR cases from each other. In this way, each service class can be handled independently of the other, and resources are not wasted, since bandwidth and queue length are dynamically re-allocated for all queues. For real-time services, a large bandwidth allocation and a short queue are appropriate to minimize delay and jitter. For the unspecified bit-rate service, a low priority, a small bandwidth allocation and a longer queue are appropriate to exploit any interval that is not used by another service. To ensure a useful partition of bandwidth among the queues, various algorithms for fair queueing have been examined for ATM switches. They are used to assign a greater amount of service time to the queues that serve the more time-critical service classes, while the less time-critical classes are allowed to exploit all remaining time. Further information on the specific fair queueing algorithms that are considered for use in ATM networks can be found in, e.g., [Gol94], [Zhang90] and [FlJa95].

7 Adaptive Mechanisms

Compression algorithms used in popular audiovisual encoding methods lead to continuous-media streams with variable bit-rates (VBR). Especially for conversational applications, these VBR requirements cannot be levelled out by preprocessing due to the timing constraints of these applications. This can potentially lead to a data rate mismatch between producer and consumer of the data. Furthermore, the support for resource management in general and for VBR streams in particular is rarely implemented in commercial products today. If support is missing altogether, applications can not make any reservations. If VBR support is missing, applications may perform CBR instead of VBR reservations, but worst case assumptions must be made in that case, which implies an overbooking. This leads to partially unused capacity and high costs. Several applications such as conferencing can deal with varying transmission bandwidth if appropriate mechanisms are employed. In this section we discuss such mechanisms which adapt the application's requirements and behavior to the available resources.

7.1 Scaling

Adaptive methods address the problem of missing or inappropriate resource management mechanisms by changing the amount of data transferred from the origin to the target over time. To perform the adaptation, a feedback control loop is introduced – the load state of network and local end-system resources are monitored and if significant changes occur, appropriate actions must be taken. For example. large delay and high loss is experienced if the network is overloaded, so the generated load must be reduced, e.g., by using a coarser coding of the input data.

The reduction can be achieved in various ways: by explicit communication between receiver and sender (the receiver informs the sender to slow down), completely in the network on a hop-by-hop basis, or by feedback from congested network nodes to the sender. Multiple systems have been developed for such scaling especially in the unicast case. [GiGu91] and [KaMR93], for instance. regulate the sender codec to adapt the amount of transmitted data, [JSTS92] describes a special queuing mechanism to adapt the bandwidth allocated by videos sent across packet-switched networks; and [HoSF93] addresses network feedback to the sender to avoid congestion in networks that cannot be supported properly by resource management.

Implementing scaling mechanisms within each single application forces programmers to construct their own mechanisms. Further, it leads to interworking problems between applications in case of several streams being scaled simultaneously and raises questions about fairness and balancing between streams. These problems can be solved by 'middleware' approaches (e.g., [KaWo94]) where media scaling methods are integrated into a general system support for multimedia (such as the Multimedia System Services defined by the IMA (Interactive Multimedia Association)).

7.2 Filtering

Sending feedback from receiver to sender informs the latter about the requirements of the receiver. Yet, how can a situation be handled where several multicast receivers require different amounts or different encodings of data from the sender? This can be the case, for example, if the network capabilities to or the processing capacities of the receivers vary. Such a scenario of heterogeneous receivers can be supported in multimedia applications by using filtering mechanisms where an intermediate network node changes the amount of transmitted information.

Filters can be introduced as a general concept (as proposed in, e.g., [Pasq93]) allowing for arbitrary operations on multimedia data in any part of the network. Such filters can, for example, be used to transform one encoding format to another, e.g., from ADPCM to PCM audio coding, which is useful if end-systems have different encoding requirements. Although the generality of the model is appealing, it can lead to several problems: long processing times may increase communication delays, security aspects may prohibit users from down-loading code for arbitrary filters into routers (limiting the approach to predefined filters), and not all intermediate nodes, for example ATM switches, may be suited to provide the required processing capabilities. To avoid (or at least reduce) some of these problems, e.g., placing potentially large processing loads on intermediate nodes, such operations could be performed by specialized nodes in the network.

A simpler filtering approach is that an intermediate node removes parts of the data and forwards only a subset of the full information, which is finally presented to the end user on the receiving side. Thus, the source always emits a full stream, but the stream may be adapted to a stream with lower quality. Figure 3 shows such a filtering tree.

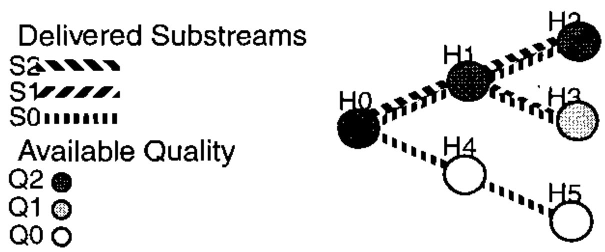

Figure 3: Substream Filtering.

The encoding of the full stream at the origin can follow one of two approaches:

- In *independently* coded streams, higher-quality parts are substitutions for lower-quality parts. For example, one substream S1 may contain complete images of the size a*b and another substream S2 may contain complete images of the size 2a*2b. To choose a different quality means to choose a different substream.
- In *hierarchically* encoded multimedia streams, higher-quality parts are additions to the lower-quality parts. For example, one substream S1 may contain images of the size a*b and another substream S2 may contain all *additional* pixels that extend the format to 2a*2b. To present data in the highest quality, both substreams must be presented.

The independent coding of streams can lead to inefficiency due to the resulting overhead of transmitting 'similar' data multiple times (often called simulcast). The hierarchical coding avoids this, yet, it can be more complex because several parts must be combined to have the full information available. Further, it requires that the full stream has been encoded into a hierarchy of information layers, as is for instance possible with MPEG-2. Then two approaches can be followed for the hierarchical coding.

In the first approach, the application splits the data into streams and sends them independently. This has been described in various papers as in [DHHH94], where ST-2 is additionally used to reserve resources for the base layers, or as in [BTSW94] and [ChGu96]. where a similar approach using an IP multicast group for each layer has been taken. A refinement of this is the use of error detection within the receiver-driven layered multicast approach as described in [McJV96]. Receivers start out to receive the base layer and add further enhancement layers until they have either subscribed to all layers or until they experience packet loss. In the latter case, they remove the less important layers again. This way, receivers search for the optimal level of subscription. And, due to the pruning mechanisms of IP multicast, only layers for which a subscription exists in a particular area of the Internet are forwarded into that area. Furthermore, these mechanisms can but must not be combined with resource reservation, e.g., reservations using RSVP could be established for some of the layers. A potential drawback is that the independent transmission of the layers can lead to differing delays among the layers (e.g., due to the use of different routes or priorities) which may cause synchronization problems because some layers depend on others. While this is probably not a serious problem today, it may become an issue in future if a large number of layers is used, e.g., due to new object-based coding schemes such as MPEG-4. And it is not possible to have any resource sharing among layers without additional grouping mechanisms.

In the second approach, such problems are inhibited. Here, the application transmits one stream, but describes the structure of the stream (e.g., as part of the FlowSpec), allowing receivers to specify filters which strip off information not to be transmitted to that receiver [WoHD95]. This dropping is performed by a filter in a network node. Since the relations between the parts are known within the network, sharing effects might potentially be gained. However, this scheme is more complex because it requires that packets are identifiable and that the streams and their relationship are specified.

8 Reservation in Advance

The resource management systems described above, offer functions which allow the reservation of resources for a time interval which starts with the reservation attempt and which lasts for an unspecified time. For several application scenarios this model of immediate reservations is not appropriate. Consider, for instance, a virtual meeting room (conferencing) scenario supported by multimedia systems, where perhaps weeks in advance of the actual 'meeting', it must be ensured that sufficient resources to hold the conference are available. To support these 'virtual meeting room' scenarios the resource reservation system must offer mechanisms to reserve in advance the resources needed for the conference, i.e., certain capacities of networks, routers, 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. In general, if resource reservation is needed, then ReRA must be provided as well.[5] However, several issues must be resolved before ReRA will find widespread support; here we can only address some of these issues and solution approaches for them. It might even be that the required overhead is too large to pay off (it might be more cost effective to reduce the blocking probability by over-provisioning resources).

5. If there is a noticeable reservation blocking probability.

8.1 Characterization and Model

Reservations can be classified based on two key factors [WDSS+95]: (1) whether the resources are exploited at reservation time, and (2) whether the reservation duration is known at reservation time. 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 extended). In case of immediate usage and known duration, either scheme can be realized.

Then the ReRA scheme consists of two parts (see also Figure 4): (1) the resource reservation in advance and (2) the usage of the reserved resources. In the first part, the client specifies its request, i.e., it gives a *workload specification* and defines the *begin* and *duration* of the reservation. 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. Then the client exploits its reservation by making use of the reserved resources. Once a session is established, the participants may either finish earlier (than previously reserved) or they may want to extend the time. The first case is simple; resources can be freed and made available for other applications. However, in the second case, if the application duration is to be extended, the system may or may not have a sufficient amount of resources to serve the application with the necessary QoS. If enough resources are available, the service should not be interrupted and the application should be provided 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.

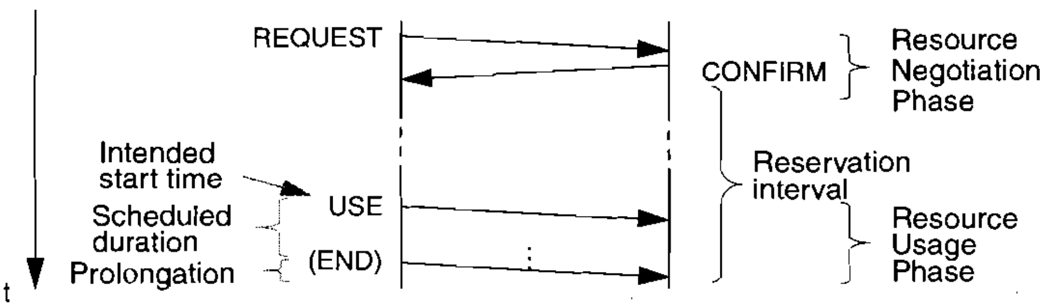

Figure 4: Reservation in Advance Steps.

8.2 Distribution of Announcement Information

In addition to the information about stream characteristics which are exchanged via resource reservation protocols such as RSVP (cf. Section 6), information about the date of the stream start and even basically the knowledge about its existence time must be distributed as well. Such information is today usually distributed via other means than those later used for the application; for example, the invitation to join a multi-user phone conference is given to the potential participants by contacting each person independently via a point-to-point phone call. In the Internet, the sd or sdr programs are often used for such notifications if the event is open and can be joined by anyone who is interested.

The information about announcements can be handled by a 'user agent' which 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 automatically start the application at the scheduled starting time of the data stream, i.e., just before the conference begins. ([WoSt97])

8.3 Failure Situations

With ReRA, in addition to the handling of failures in the negotiation phase and the usage phase, care must be taken of failures that may occur between these phases, i.e., after a reservation has been made but before it is used. First, the reservation state stored at end-systems and intermediate nodes might be needed for long time periods. State information must be stored in non-volatile storage. This is not only necessary as a protection against failures, but also because any node may be restarted regularly between the phases, e.g., for maintenance. Alternatively, similar to the approach followed by RSVP, the reservation may be refreshed from time-to-time. As the actual usage time approaches, refresh messages are sent more often [DKPS95]. The drawback of this approach is that reservations which have been set might be lost during a system outage and cannot be reset because others have occupied the resources in the meantime.

Furthermore, as opposed to failures occurring during data transmission, no client is running when a node notices a failure. 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. The reservation system must provide means to inform the clients explicitly about the failure situation and whether or not it can be resolved in time and the application must be able to query the correctness and availability of the reservation before it starts its usage phase.

An interesting proposal to handle these cases is described in [ScPi97] where a third party, a reservation agent, keeps track of set reservations and manages potentially occurring failures even if the endsystems are not running.

8.4 Example for a Reservation in Advance

To illustrate the steps performed as part of a reservation in advance, a short and simplifying example is presented in this paragraph. After all potential participants have been informed about the upcoming event and expressed their interest in that (by 'external' means or an announcement system as discussed in Section 8.2), the reservation is started by exchange of reservation protocol messages. The FlowSpecs carried in these messages contain time parameters to specify begin and duration of the reservation. Each node which receives the reservation message checks as usual whether the needed resource capacities are available, but it must now check against the future time slot and not the current one. Later, when the reservation should be used, the scheduling mechanisms must be informed about the active data streams. When the duration of the reservation is over, the scheduler must withdraw the resources from this stream.

8.5 Modifications to Support Advance Reservations

The issues discussed so far and the example given in the last paragraph make clear that various components of current resource management systems have to be modified to support ReRA scenarios. The interfaces of resource management systems need in addition to the QoS parameters now also specifications of the time parameters (begin and duration). These time values must be contained in the flow specification that is distributed via the resource reservation protocols to all affected network nodes ([Rein94], [Rein95]). The database of existing reservations must represent time slices (e.g., [FGV95]). For each time the set of existing or reserved streams with their QoS parameters and the free resources must be known. The admission control algorithms must take the time parameters into account; an example for such an algorithm for predictive service is given in [DKPS95]. Additional failure handling mechanisms and means to save state information in permanent storage are necessary. Furthermore, the reservation protocols must be enhanced. New PDU types to support the additional states and transitions (e.g., to indicate the usage phase) and to handle failure situations and notify neighbor nodes about them are needed.

9 Open Issues

Many pieces of an overall infrastructure for distributed multimedia applications have been developed over the last years. So far, only some parts have found their way into systems of daily use. Others will be deployed in the future while some have been put aside (for varying reasons, e.g., complexity, questionable usefulness, 'political' incorrectness, ...).

Various parts for the multimedia communication infrastructure are still missing and must be developed in the future to offer a complete solution. Examples for missing or incomplete parts, especially with respect to the QoS provisioning part of multimedia communication, are QoS routing and pricing mechanisms. Perhaps most important will be the verification of the suitability of the proposed mechanisms for large-scale use: for (few) large multicast sessions with many receivers as well as for many small, concurrent sessions.

9.1 Scalability

The multimedia communication methods designed for shared and distributed components must be scalable. With respect to multimedia applications, e.g., multicast video conferences, scalability has at least two aspects:

(1) scalability with respect to the number of participants in one application,
(2) scalability with respect to the number of concurrent applications.

The first requirement states that it must be possible to transmit a flow (distributed via multicast) to a potentially very large number of participants. This is, for instance, the case in transmissions from IETF meetings or prominent lectures. To fulfill this requirement (as discussed in Section 5.2), mechanisms for resource sharing among participants and for the aggregation of reservations must be provided, furthermore, there should be no central component which has to process requests from new participants joining resp. old participants leaving such a conference. Based on their receiver-oriented reservation and flow joining concepts, RSVP and IP multicast support this requirement. ATM offers now the new leaf-initiated join feature to reduce the load of any central component, yet, grouping concepts are still missing.

The second requirement demands that it should be possible to support many independent applications, and hence flows, – for example, thousands of video-conferences (probably with very few participants) and video-retrieval sessions. Therefore, the processing and storage effort per flow must be very small. The abilities of current protocols and architectures in this respect are not yet clear. We believe that it can be only shown by experiments because processing and storage overhead depend not only on architecture but on particular implementation as well. The soft state approach used by RSVP requires periodic message exchange per flow per receiver (potentially reduced by aggregation among receivers) to refresh the reservation and, hence, avoiding the loss of state. This needs transmission bandwidth and, perhaps more important, processing in the routers. Additionally, in order to be able to remove expired soft state, a timer must be kept per each receiver within each flow. While some of these can be aggregated (and efficient timer mechanisms such as timing wheels are known for quite a while), it can be expected that all these timers lead to some non-negligible overhead. Hard state approaches, on the other hand, avoid these problems since they neither require the permanent exchange of refresh messages nor timers per receiver (but per neighbor). However, they must keep the state all of the time and cannot, as soft state approaches may do, throw it away in case of lack of storage capacity.

Which type of scalability is more important depends on the predominant usage scenario. Currently, it seems that more attention has been given to the first issue: the scalability of one application. In future, small-sized applications will probably be more important, hence, more consideration should be given to the second issue: the scalability of concurrent applications.

9.2 Routing

QoS driven routing algorithms are needed for the efficient establishment of reservations. These algorithms suggest one or multiple suitable paths towards a given target considering a given set of QoS requirements. Then one attempts to make a reservation on such a path. Without appropriate routing mechanisms which take QoS requirements into account, the setup of reservations becomes a mere trial-and-error approach.

A QoS driven routing algorithm has to consider the currently available capacity of a resource to avoid an immediate rejection of the reservation attempt and the QoS requirements of the reservation to find a route best-suited for this QoS. It should also consider the resource load after the routing decision to avoid using up the majority of resources on this route.

Some of the problems to be solved with QoS routing are: how much state information should be exchanged among the routers; how often should this state information be updated; must there be a distinction between exterior and interior systems and if yes, how can it be made; is it possible to hide internal details of an autonomous system; can the complexity of path computation be managed ?

QoS routing is currently still in its infancy. At least in the Internet, its necessity, the ability in principle to perform QoS routing, and the proposed approaches are currently under controversial discussions. Furthermore, it seems difficult to combine QoS routing and receiver-oriented reservations. The hard-state, sender-oriented ATM camp, on the other hand, designed PNNI which provides at least some QoS routing support.

9.3 Pricing

An important issue for the future success of distributed multimedia applications is the cost for any data transmission (i.e., with or without QoS) and the question 'why should a user ask for less then the best quality' has always been answered with 'costs'. QoS methods have to take cost into account as an additional (possible negotiable) parameter. However, as discussed in [SCEH96], most research has focused on specific issues; architectural issues have most often been neglected. The issues to be attacked are among many others: who pays for a service, and how is this indicated, especially if the receiver benefits from the transmitted data; can the user specify a limit on its expenditures; how can fairness be provided such that each receiver within a multicast session pays its share, how can payment cross a firewall, how can a department or group be charged instead of the overall company ?

In addition to these aspects which apply to transmissions without QoS, further questions have to be answered in QoS provisioned systems, e.g.: how can resource consumption be "weighted" (e.g., delay vs. loss); what QoS do users accept for a specific price and which pricing schemes do they understand; how can fairness be provided such that all users – benefiting from a reservation made for a multicast transmission – share the costs in a fair manner ?

10 Conclusions

Multimedia communication has been (and certainly will be much more) used by various distributed applications: Video-conferencing, retrieval systems and video-on-demand will address all network types, LANs (e.g., in-house information systems), MANs (e.g., city information systems, campus networks) and WANs (e.g., distributed lectures).

The provision of a well understood QoS is a crucial issue for the successful delivery of audiovisual and any other time sensitive data over networks and hence, for distributed multimedia applications. Within current networks this requires mechanisms like resource reservation or adaptive mechanisms such as scaling and filtering. The need for reservations was highly controversial a couple of years ago. Now, the concept of reservation-based QoS has found wide-spread acceptance, nevertheless there are still *'reservations about reservations'*[6] whose advocates consider reservations as too complex and propose adaptive mechanisms as overall solution. Neither reservation-based nor adaptation-based QoS support would be necessary if the available system resources would become abundant. Yet, we believe that resource demand grows at least at the same pace as available resources, hence, reservation will be necessary for quality demanding applications and users in the future.

If reservation is needed at all (what the authors are convinced of) then it probably applies also to ReRA. Yet this will require modifications, add complexity to protocols and network nodes, and furthermore, requires that state information is kept in the network for quite some time. These requirements lead to questions about stability and scalability.

Resource reservation and scaling mechanisms have been an active research area with increasing dedication already during more than the last five years. Currently, the Internet does not support QoS on a wide scale. This will change in the near future due to the support of RSVP, accompanying admission control and scheduling mechanisms. Their commercial success will depend on the proof of the respective suitability for a large scale use, i.e. a huge amount of concurrent flows and numbers of participants in a flow. We believe that RSVP will find its way into day-to-day usage because of its support from the IETF and from industry. Yet, what started out as a small protocol became relatively complex over time. Questions can be raised about its appropriateness, completeness, and overall complexity which might slow-down its deployment.

The success of ATM for multimedia communication depends on the successful standardization of its signalling mechanisms, its ability to attract the development of native ATM applications and the integration of ATM with other communication systems. It seems more and more that ATM will not find its way to the desktop, i.e., there will be no native ATM applications. ATM will have its role in the backbone, so an integration with the Internet is needed. The integration of ATM *into* the Internet world is under investigation (using ATM as a subnetwork with RSVP on top of it). But if there will be native ATM applications, e.g., video-on-demand, then there is also the need for a 'side-by-side' integration of ATM and Internet protocols, however, no advanced work on that exists. Furthermore, the integration of the various network infrastructures into a global, ubiquitous network capable of providing suitable support for multimedia communications must address, besides current Internet technology and ATM, also mobile systems.

The multimedia communication systems developed so far address several of the necessary aspects. Yet, not all issues have been tackled. Solutions for these issues have to be developed and they have to find their way through the standardization activities to provide for a complete multimedia communication infrastructure which is needed to support distributed multimedia applications.

---

6. Which was, for instance, the theme of a panel at IWQoS'97 (International Workshop on Quality of Service).

Acknowledgments

The multimedia systems and networking experience gained with all our team members at the IBM European Networking Center in Heidelberg and the Darmstadt University of Technology made possible this work. We would also like to thank Barbara Lutes for improving the paper style.

Further Information

The interested reader will find further information in various journals, conference proceedings and internet publications. Here we can provide only a brief selection covering the most relevant:

Journals, Magazines & Newsletters: ACM/IEEE Transactions on Networking, ACM Computer Communications Review, ACM Multimedia Systems Journal, IEEE Journal on Selected Areas in Communication, IEEE Communications, IEEE Multimedia Magazine, IEEE Networks, Computer Communications, Internetworking, High-Speed Networking, Telecommunication Systems

Conferences & Workshops: ACM SIGCOMM, ACM Multimedia, IEEE International Conference on Multimedia Computing and Systems (ICMCS), INFOCOM, Network and Operating System Support for Digital Audio and Video (NOSSDAV), High-Performance Networking, QoS Workshop

Internet: Request for Comments (RFC) and Internet Drafts can be obtained from various hosts, e.g., by starting with the IETF homepage at http://www.ietf.org/.

References

[AtmF96] The ATM Forum: "ATM Traffic Management Specification, Version 4.0", Upper Saddle River, NJ, Prentice Hall, 1996

[BFMM94] A. Banerjea, D. Ferrari, B.A. Mark, M. Moran: "The Tenet Real-Time Protocol Suite: Design, Implementation, and Experiences", Technical Report TR-94-059, International Computer Science Institute, Berkeley, CA, USA, November 1994.

[BTSW94] J.C. Bolot, T. Turletti, S.Schröder, I. Wakeman: "Scalable Feedback Control for Multicast Video Distributionin the Internet", Proceedings of SIGCOMM '94.

[CaCH94] A. Campbell, G. Coulson, D. Hutchinson: "A Quality of Service Architecture", ACM Computer Communication Review, Vol. 24, No. 2, April 1994.

[ChGu96] N.Chaddha, A.Gupta: "A Frame-Work for Live Multicast of Video Streams over the Internet", Proceedings of the IEEE International Conference on Image Processing, 1996.

[Cruz91] R.L. Cruz: "A Calculus for Network Delay, PART I: Network Elements in Isolation", IEEE Transactions on Information Theory, Vol.37, No. 1, January 1991.

[DHHH94] L.Delgrossi, C.Halstrick, D.Hehmann, R.Herrtwich, O.Krone, J.Sandvoss, C.Vogt: "Media Scaling in a Multimedia Communication System", ACM Multimedia Systems Journal, Vol. 2, No. 4, 1994.

[DHHS94] Luca Delgrossi, Ralf Guido Herrtwich, Frank Oliver Hoffmann, Sybille Schaller: "Receiver-Initiated Communication with ST-2", ACM Multimedia Systems Journal, Vol. 2, No. 4, 1994.

[DKPS95] M. Degermark, T. Köhler, S. Pink, O. Schelén: "Advance Reservation for Predicted Service", Proceedings of Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995, Springer-Verlag LNCS 1018.

[FGV95] 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, Springer-Verlag LNCS 1018.

[FlJa95] S. Floyd, V. Jacobson: "Link-Sharing and Resource Management Modes for Packet Networks", IEEE/ACM Trans. Networking, Vol. 3, No. 4, August 1995

[GiGu91] M.Gilge, R.Gusella: "Motion video coding for packet-switching networks - an integrated approach", Proceedings of the SPIE Conference on Visual Communications and Image Processing, 1991.

[Gol94] S. J. Golestani: "A Self-Clocked Fair Queueing Scheme for Broadband Applications", Proc. IEEE INFOCOM, May 1994

[HoSF93] D. Hoffman, M. Speer, G. Fernando: "Network Support for Dynamically Scaled Multimedia Data Streams", Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK, 1993.

[I.362] ITU-T Recommendation I.362, B-ISDN ATM Adaption Layer (AAL) Functional Description, Geneva 1991.

[JSTS92] K. Jeffay, D.L. Stone, T. Talley, F.D. Smith: "Adaptive Best-Effort Delivery of Digital Audio and Video Across Packet-Switched Networks", Third International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego, USA, 1992.

[KaWo94] T. Kaeppner, L. Wolf: "Media Scaling in Distributed Multimedia Object Services", Proceedings of the Second International Workshop on Advanced Teleservices and High-Speed Communication Architectures, Heidelberg, Germany, September 26-28, 1994, Springer-Verlag LNCS 868.

[KaMR93] H.Kanaki, P.P.Mishra, A. Reibman: "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport", Proceedings of SIGCOMM '93.

[McJV96] S.McCanne, V.Jacobson, M.Vetterli: "Receiver-driven Layered Multicast", Proceedings of SIGCOMM'96, October 1996.

[NaSt95] K. Nahrstedt, R. Steinmetz: "Resource Management in Networked Multimedia Systems", IEEE Computer, Vol. 28, No. 4, April 1995.

[Pasq93] J. Pasquale: "Filter Propagation in Dissemination Trees: Trading off Bandwidth for Processing in Continuous Media Networks", Fourth International Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, UK, November 1993.

[RFC1112] S. Deering, "Host extensions for IP multicasting", RFC1112, 1989

[RFC1633] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994

[RFC1819] L. Delgrossi, L. Berger, "Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+", August 1995.

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", January 1996.

[Rein94] W. Reinhardt: "Advance Reservation of Network Resources for Multimedia Applications", Proceedings of the Second International Workshop on Advanced Teleservices and High-Speed Communication Architectures, Heidelberg, Germany, September 26-28, 1994, Springer-Verlag LNCS 868.

[Rein95] W. Reinhardt, "Advance Resource Reservation and its Impact on Reservation Protocols", Proceedings of Broadband Island'95, Dublin, Irland, September 1995.

[ScPi97] O. Schelén, S. Pink: "Sharing Resources through Advance Reservation Agents", Proceedings of International Workshop on Quality of Service, New York, NY, USA, May 1997.

[SCEH96] S. Shenker, D. Clark, D. Estrin, S. Herzog: "Pricing in Computer Networks: Reshaping the Research Agenda", ACM Computer Communication Review, April 1996.

[ShBr95] Scott Shenker, Lee Breslau: "Two Issues in Reservation Establishment", Proceedings of SIGCOMM'95.

[StNa95] Ralf Steinmetz, Klara Nahrstedt: "Multimedia: Computing, Communications and Applications", Prentice-Hall, July 1995.

[Stue95] Heinrich J. Stüttgen: "Network Evolution and Multimedia Communication", IEEE MultiMedia Magazine, Vol. 2, No. 3, Fall 1995.

[VoHN93] C. Vogt, R.G. Herrtwich, R. Nagarajan: "HeiRAT: The Heidelberg Resource Administration Technique – Design Philosophy and Goals," Proceedings of Kommunikation in Verteilten Systemen, Munich, Germany, March 3-5, 1993, Springer-Verlag.

[VWHW97] C. Vogt, L.C. Wolf, R.G. Herrtwich, H. Wittig: "HeiRAT – Quality-of-Service Management for Distributed Multimedia Systems", to appear in ACM Multimedia Systems Journal – Special Issue on QoS Systems, 1997.

[WDSS+95] L.C. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, H. Wittig: "Issues of Reserving Resources in Advance", Fifth International Workshop on Network and Operating System Support for Digital Audio and Video, Durham, NH, USA, April 19-21, 1995, Springer-Verlag LNCS 1018.

[WoHD95] L.C. Wolf, R.G. Herrtwich, L. Delgrossi: "Filtering Multimedia Data in Reservation-Based Internetworks", Proceedings of Kommunikation in Verteilten Systemen, Chemnitz, 1995, Springer-Verlag.

[Wolf96] L.C. Wolf: "Resource Management for Distributed Distributed Multimedia Systems", Kluwer Publ., Boston 1996.

[WoSt97] Lars C. Wolf, Ralf Steinmetz: "Concepts for Resource Reservation in Advance", Special Issue of Journal of Multimedia Tools and Applications (Kluwer), The State of the Art in Multimedia Computing, May 1997.

[Zhang90] L.Zhang: "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks", Proceedings of ACM SIGCOMM, Philadelphia. PA, September 1990.

[ZDES+93] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala: "RSVP: A New Resource ReSerVation Protocol," IEEE Network, September 1993.