

Survey of Error Recovery Techniques for IP-based Audio-Visual Multicast Applications

Georg Carle and Ernst W. Biersack
Institut EURECOM
06904 Sophia Antipolis
FRANCE

g.carle@ieee.org, erbi@eurecom.fr

Abstract

IP-based audio-visual multicast applications are gaining increasing interest since they can be realised using inexpensive network services that offer no guarantees for loss or delay. When using network services that do not guarantee the Quality of Service (QoS) required by audio-visual applications, the recovery from losses due to congestion in the network is a key problem that must be solved.

This survey gives an overview of existing transport-layer error control mechanisms and discusses their suitability for use in IP-based networks. Additionally, the impact of IP over ATM on the requirements of error control mechanisms is discussed. Different network scenarios are used to assess the performance of retransmission-based error correction and forward error correction.

1 Introduction

The success of the World Wide Web has accelerated the already fast growth of user population, host count, and amount of traffic on the Internet. However, this success frequently leads to network congestion and loss and threatens to slow down the possible next success story: the widespread use of the Internet for audio-visual communication. Web traffic is based on TCP, which uses a closed loop congestion control algorithm; continuous media (CM) applications typically use UDP, which does not provide a congestion control mechanism. Therefore, CM applications may significantly aggravate Internet congestion.

Without changing the current best-effort service model, the rapid growth of the Internet makes powerful real-time error control mechanisms for audio-visual applications a necessity.

In the past, a widespread believe has been that *retransmission-based* error control is not suitable for interactive audio-visual applications. While the development of retransmission-based error control mechanisms for CM applications has been a research topic for several years, it was only recently that this topic became of widespread interest, leading to a significant number of improvements. This includes a large number of proprietary protocols that are currently being introduced as part of CM Internet applications, and as Web browser plug-ins. While the first generation of CM applications that deploy error control is limited to simple point-to-point communication (unicast), recent developments also use multicast network services. As more and more end users get access to the MBONE (IP Multicast Backbone, [20, 35]), error control schemes that allow the successful provision of multicast audio-visual services over the Internet are of significant economic importance.

Existing surveys on protocols for reliable multicast services by B. Levine and J.J. Garcia-Luna-Aceves [5] and by K. Obraczka [46] are limited to protocols for fully reliable services and ignore real-time aspects. A descriptive survey of protocol functions and mechanisms for multipoint communication is given in [19], including an overview of existing approaches for fully reliable multicast services. In contrast to existing work, this article investigates error control for real-time services and also discusses the influence of ATM-specific vs. IP-specific network properties onto the suitability of mechanisms for real-time error control.

This survey presents general requirements for error control mechanisms for audio-visual applications by defining important application scenarios. Subsequently, the impact of IP and ATM network properties is discussed. An overview of existing protocols that provide reliable multicast services for audio-visual applications is given and the error control mechanisms used are analyzed. We introduce a taxonomy that allows to judge the suitability of these approaches in representative application and network scenarios. We use this taxonomy to present guidelines for the selection of the approach that is most suitable for a given scenario. Finally, shortcomings of existing protocol mechanisms are identified and improvements are presented.

1.1 Transport Protocol Requirements

When discussing the transmission of continuous media across packet switched networks, it is useful to recall the most important characteristics of continuous media streams such as audio, video, or animation:

- *Strict timing requirements*: If the data are not delivered before a certain point in time they are useless.
- *Some tolerance to loss*: the amount of loss that can be tolerated depends on the medium, the coding techniques used, and on human perception.
- *A certain periodicity*: Video, for instance, consists of a fixed number of frames per second. When transmitting CM across a network this periodicity is normally lost.

Data transmitted across packet switched networks are normally subject to delay, delay jitter, re-sequencing of packets, and loss of packets.

When designing a protocol for the transfer of continuous media across packet networks, a set of mechanisms are available to correct the impairments introduced by the network layer. These mechanisms can improve the quality of the network service in order to provide a Quality of Service (QoS) that is acceptable for the CM application.

1.1.1 Multipoint Transport Services

Existing protocols may be classified according to the degree of reliability they provide: A **fully reliable** multicast transport service ensures the successful delivery to *all* receivers of the group, while a reliable **real-time** multicast transport service reduces the remaining error probability within a given *delay budget* (see table 1).

It can further be distinguished whether a service involving a single source (1:N or **multicast** service) is provided, or whether multiple sources are supported (M:N or **multi-peer** service).

Application examples that use a fully reliable multicast transport service with a single source are data dissemination (multicast-FTP) such as software updates, distribution of movies, or WWW push caching.

Examples employing a fully reliable multi-peer service are CSCW (Computer-Supported Cooperative Work) applications such as shared white-boards and group editors, and distributed computing on workstation clusters as for example used for distributed simulations.

Examples using a reliable real-time multicast transport service with a single source are audio-visual conferencing, as well as dissemination of audio and video streams.

Examples using a reliable real-time multicast transport service with multiple sources are DIS (Distributed Interactive Simulation) and distributed VR (Virtual Reality) applications [40].

Transport service type	<i>Fully reliable multicast</i>	<i>Real-time multicast</i>
<i>Single source: 1:N</i>	Multicast-FTP; Software update	Audio-visual conference; Continuous Media Dissemination
<i>Multiple Sources M:N</i>	CSCW; Distributed computing	DIS; VR

Table 1: Multicast transport service types and example applications.

1.2 Application Scenarios

A typical transport protocol for non-real-time applications is supposed to reliably deliver a data stream. The requirements of non-real-time applications are mostly very general. This makes it possible for a single protocol such as TCP to be used by a large variety of applications that require a reliable unicast communication.

In contrast, real-time applications mostly have specific requirements, which, for a specific application scenario, frequently make certain protocol mechanisms more suitable than others.

Application scenarios can be defined based on the characteristics of the Application Data Units (ADUs, [14]) that are exchanged:

- short or long (e.g. for audio, or video);
- same or different size (e.g., depending on encoding scheme);
- same or different impact on application-level QoS (e.g., MPEG I,P, and B frames);
- same or different time interval between the availability of two consecutive ADUs at transmitter (e.g., when using a to hardware or software encoder);
- same or different time interval of relevance at receiver (e.g., an MPEG I frame is relevant for a complete Group of Pictures (GOP)).

Additionally, application scenarios can be distinguished according to the *end-to-end delay requirement*, leading to a transport-level delay budget. Some parameters characterizing an application may vary over several orders of magnitude, such as bandwidth requirements for video streams. Delay requirements usually vary over less than two orders of magnitude.

Depending on whether the application is interactive in nature or not, the delay requirements differ. Interactive applications, such as audio or video conferences, can not tolerate a delay of more than a few hundred milliseconds [34]. Non-interactive applications can tolerate much higher delays in the order of a second or more, since this delay is only noticeable as start-up delay, but is transparent *after* start of the play-out.

Typical delay requirements for audio-visual applications are:

- tight (up to 200 ms);
- medium (200 to 500 ms);
- loose (larger than 500 ms).

Application scenarios can also be distinguished according to the reliability requirements:

- high (less than one ADU per minute or hour lost or corrupted);
- medium (one ADU per minute or hour lost or corrupted);
- low (more than one ADU per minute or hour lost or corrupted).

The tolerable loss rate depends very much on the type of medium and on the type of coding technique used. We can distinguish two main classes of coding techniques: (i) techniques where each sample is coded *independently*, such as PCM for audio or JPEG for video and (ii) techniques where mainly the *differences* between samples are coded, such as ADPCM for audio or MPEG for video. Coding techniques of the second class are more sensitive to loss since the loss of a single sample may also prevent the correct decoding of the subsequent samples.

In case of video another class of coding is of interest, which is called **hierarchical** video coding or **layered, scalable** or **sub-band** coding [13]. In hierarchical video coding, the source information, which is represented as a digital signal, is partitioned into sub-streams or layers. The lowest layer carries the signal that is essential for the reconstruction of the video at the receiver. The other layers contain additional information that will improve the quality of the video signal received. The different layers can, for instance, correspond to different bit rates, frame sizes, or frame rates. Such a coding technique allows a graceful adaptation of the video streams to the available resources. The MPEG1 standard with its I, P and B frames can be viewed as a hierarchical coding where the I-frames make up for the lowest layer. The MPEG2 standard [53] offers four scalability modes: Spatial scalability, temporal scalability, data partitioning, and SNR scalability [36].

Due to this high variety of application-specific ADU characteristics and requirements for delay and reliability it is difficult to design error recovery protocols that are well suited for the whole range of parameters. Instead, solutions are typically targeted for specific application scenarios.

To assess the suitability of certain protocol mechanisms, we present three different application scenarios. Each application scenario is characterized by ADU size, availability, life span, priority, delay budget, and reliability requirements. Table 2 summarizes the classification of the selected reference application scenarios.

1.2.1 Interactive Audio

Interactive audio is characterized by small ADUs of constant size with typically a single priority. The ADUs are available at equidistant time intervals and have identical live spans (time intervals of relevance) at the receiver. (One exception is the application *freephone*, see [8], which uses voice samples that are encoded using two different encoding schemes.) Application-level reliability requirements are medium to high and the overall delay budget is medium to small. Typical values for interactive audio applications are PCM encoding according to G.711 with data rate of 64 Kbit/s and a delay budget of 200 ms.

1.2.2 Interactive Compressed Video

Typical examples are video streams that are compressed in real-time using the compression standards H.261 or MPEG. ADU sizes are medium to long and may vary. Different ADUs can have a different impact onto application-level QoS. In case of variable encoding delays, ADUs may be available for transmission at varying time intervals. In MPEG, even though a Group of Pictures (GoP) will be displayed in a sequence such as for example IBBPBBPBB, the availability after coding will be IPBBPBBPBB. In addition, ADUs can have different time intervals of relevance at the receiver. In MPEG, an I-Frame is relevant for a complete GoP, while the time interval of relevance for B-frames is smaller. Therefore, different error recovery mechanisms may be appropriate for different ADUs

within a single data stream. Typically, interactive compressed video has data rates of 1.5 Mbit/s (MPEG-1) and a delay budget of 500 ms.

1.2.3 Dissemination of Stored Video

For the dissemination of stored videos, slightly different application characteristics can be observed, as compared to interactive video. The MPEG encoding standard plays a dominant role for this application class. Video frames are stored in files in the sequence they are displayed (for example in the sequence IBBPBBPBB) and are frequently available at equidistant time intervals. The delay budget for this application class is relatively high due to the non-interactive nature of the application. For video dissemination that meets widespread quality expectations typical data rates are 6 Mbit/s (MPEG-2). A delay budget of 1-5 s can be assumed.

Application Scenario	A1 interactive voice	A2 interactive video	A3 non- interactive video
ADU size	short, fixed	long, variable	long, variable
ADU priorities	single	multiple	multiple
ADU availability interval	constant	variable	constant or variable
Time interval of relevance	constant	variable	variable
Data rate (typical values)	low 10 .. 64 Kbit/s	medium-high ~ 1.5 Mbit/s	high ~ 6 Mbit/s
Delay budget (typical values)	tight < 200ms	medium 200 .. 500ms	loose > 500ms

Table 2: Classification of reference application scenarios.

1.3 Mechanisms for Error Recovery

The basic mechanisms available to recover from the loss or corruption of data packets are: Automatic Repeat reQuest (ARQ) or Forward Error Control (FEC).

1.3.1 Automatic Repeat reQuest (ARQ)

Using ARQ, a lost PDU will be *retransmitted* by the sender. ARQ-based schemes consist of three parts:

- Lost data detection: Loss can be detected by the receiver (gap-based loss detection or timeout) or by the sender (timeout).
- Acknowledgment strategy: The receiver sends acknowledgments that either indicate which data have been received (positive ACKs) or which data are missing (negative ACKs referred to as NAKs) and should be retransmitted.

Retransmission strategy that determines which data are retransmitted in case of loss. The two best known retransmission strategies are Go-Back N and selective retransmission, which trade off simplicity of the receiver implementation and transmission efficiency [24].

1.3.2 Forward Error Control (FEC)

The idea of FEC is to get the transmission right the first time. For this purpose, FEC transmits together with original data some redundant data, called **parities**, to allow reconstruction of lost packets at the receiver. The redundant data is derived from the original data using techniques from coding theory [38]: Using the exclusive OR (XOR) operation allows *one* parity packet to be computed for a given set of original packets; using Reed-Solomon codes, multiple independent parities can be computed for the same set of packets. Reed-Solomon codes allow to achieve optimal loss protection, but lead to higher processing costs than schemes based on XOR operations. It is possible to compute several parity packets with lower processing cost than Reed-Solomon codes, but also with sub-optimal loss protection, by arranging original packets in a matrix and computing XOR parities over rows, columns or diagonals [58], or by constructing new codes based on XOR operations [39].

FEC schemes do not need a return path. The recovery of lost data by reconstruction at the receiver requires very little time, which makes FEC attractive for applications with real-time requirements. The FEC transmitter sends k data packets - defining a **Transmission Group** (TG) - and adds an additional h redundant parity packets. The ratio h/k is referred to as **over-code** and indicates the amount of redundancy added. Unless the network drops more than h of the $h+k$ packets sent, the receiver can reconstruct the original k information packets.

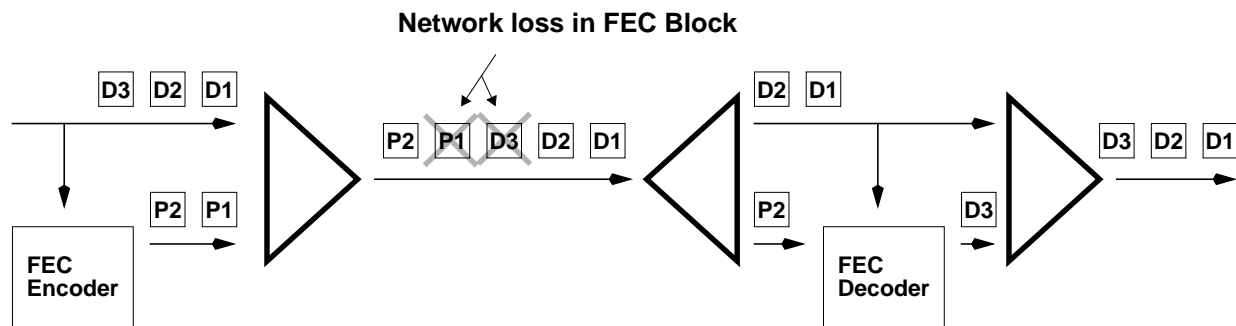


Figure 1: Operation of FEC.

Figure 1 shows the operation of an FEC system for $k=3$ and $h=2$. In this example, the FEC encoder produces two redundant packets (P1, P2) for every three data packets sent. If, for instance, one data packet (D3) and one parity packet (P1) are dropped, the receiver can reconstruct the data packet (D3).

FEC can be very effective even for a small h/k ratio, reducing the loss probability by several orders of magnitude [6]. For a given h/k ratio, losses of consecutive packets can be corrected with a higher probability for large values of k . However, the reconstruction delay at the receiver increases for large k , as a longer time interval passes until all packets required for reconstruction are arrived and as processing costs increase with larger k [55, 45].

Using parity for loss repair is particularly efficient for the case of multicast transmission, since a single parity packet can be used to repair the loss of different data packets seen by different receivers [41, 45] (see figure 2).

However, these advantages are more effective for non-real-time applications with large ADUs than for real-time applications. For real-time applications with large ADUs, such as an I-frame of a MPEG encoded video streams, it is feasible to segment a large ADU into a large number of packets, forming one or several TGs with relatively large k . For real-time applications with small ADUs, such as voice applications, within the given delay budget only a small amount of data is of relevance. As the per-packet overhead in IP-based networks is relatively large, segmentation of ADUs into several packets is usually not feasible. Instead, more frequently several small ADUs will be placed within a single packet. This limits the possible size of a TG significantly. Frequently, interactive voice applications will use small TGs of size $k=2$ or $k=3$, as larger TGs would introduce too much delay.

1.3.3 Hybrid Error Control (ARQ/FEC)

A major difficulty when using FEC is to choose the right amount of redundancy in face of changing network conditions. Also, sending redundant data consumes additional bandwidth. In order to overcome this problem, ARQ and FEC can be used in combination:

- One possibility for combining ARQ and FEC, referred to as **hybrid ARQ type II** [18], is not to send any redundant data with the first transmission, but to send parity data when a retransmission is required. As shown in [45], this approach is very bandwidth-efficient for reliable multicast to a large number of receivers. Figure 2 gives an example where different packets are lost for different receivers. Error recovery by multicast retransmission of the *original* data packets requires retransmission of *all* lost packets. On the other hand, retransmission of a *single* parity packet allows all receivers to recover their lost packet.

For a growing number of receivers and uncorrelated loss, the mean number of losses a single parity packet can repair is also growing. Multicast error control is treated more generally in the third section.

- Another approach that combines ARQ and FEC, referred to as **hybrid ARQ type I**, immediately sends a certain amount of redundant data using FEC. If the loss rate obtained after reconstruction at the receiver is still too high, ARQ is used to retransmit. Using this approach it is possible to assure with a high probability that a large number of receivers obtain the data without retransmissions, which is attractive for real-time audio-visual services.

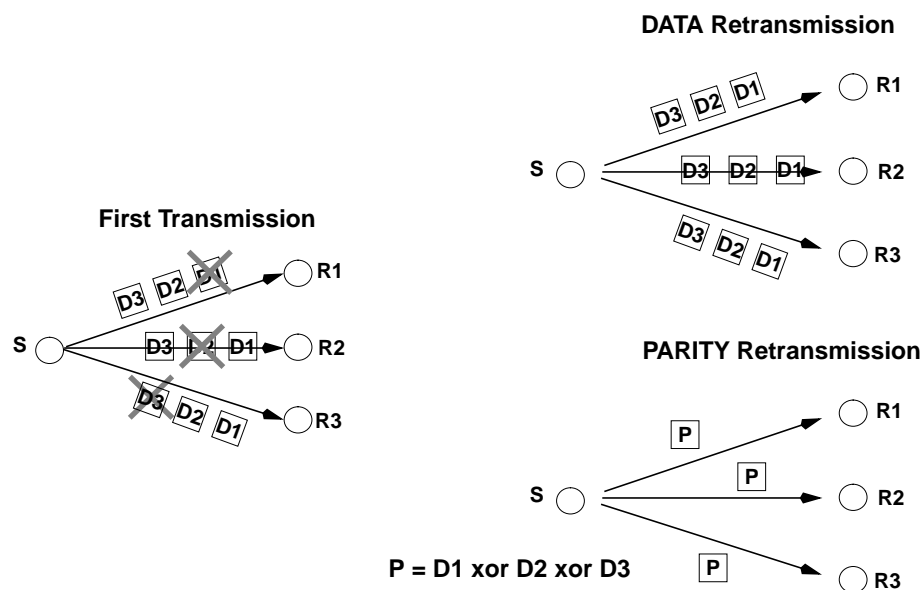


Figure 2: Reliable multicasting with hybrid error control: Recovering from losses of different data packets at different receivers, using a single parity packet.

We would like to note that it is often not sufficient to only consider the mean number of packets lost, since the nature of the loss process, such as independent losses or correlated losses, has an impact on the performance of the error recovery techniques used. For a given loss rate, FEC is able to reconstruct a higher number of lost packets when the packet losses are uncorrelated, while Go-Back-N ARQ achieves a higher throughput efficiency when the packet losses are highly correlated.

1.4 Mechanisms for Error Concealment

Applications such as audio or video transmission can use, in case of data loss, approximation or interpolation techniques to *disguise* the loss. The kind of mechanism used and their effectiveness

depends very much on the medium. For examples applied to audio see [8] and for video see [23, 33, 62].

Error recovery and error concealment techniques can also be applied in combination [54]. Since concealment techniques are not strictly error recovery schemes, we will not discuss them any further.

To compensate delay jitter in case of a transmission over a network that uses work conserving service strategies such as FIFO or round-robin, the only choice left is to buffer the incoming packets at the receiver. Such buffering will increase the end-to-end latency and may be in conflict with the real-time requirement.

1.5 Model of the End-to-End Delay of a CM Data Transmission

In the past it was frequently debated whether real-time applications can be based on protocols with retransmissions. In [16], it was shown by simulation that a real-time retransmission scheme is feasible within the end-to-end delay constraints of packet voice transmissions for typical LANs (Local Area Networks), even for relatively high network access delays and protocol processing delays in transmitter and receiver. Similar end-to-end delay properties can be achieved for larger distances in case of lower network access delays (i.e., using network adapters with higher bandwidth) and shorter processing delays, which are possible today by technology advances.

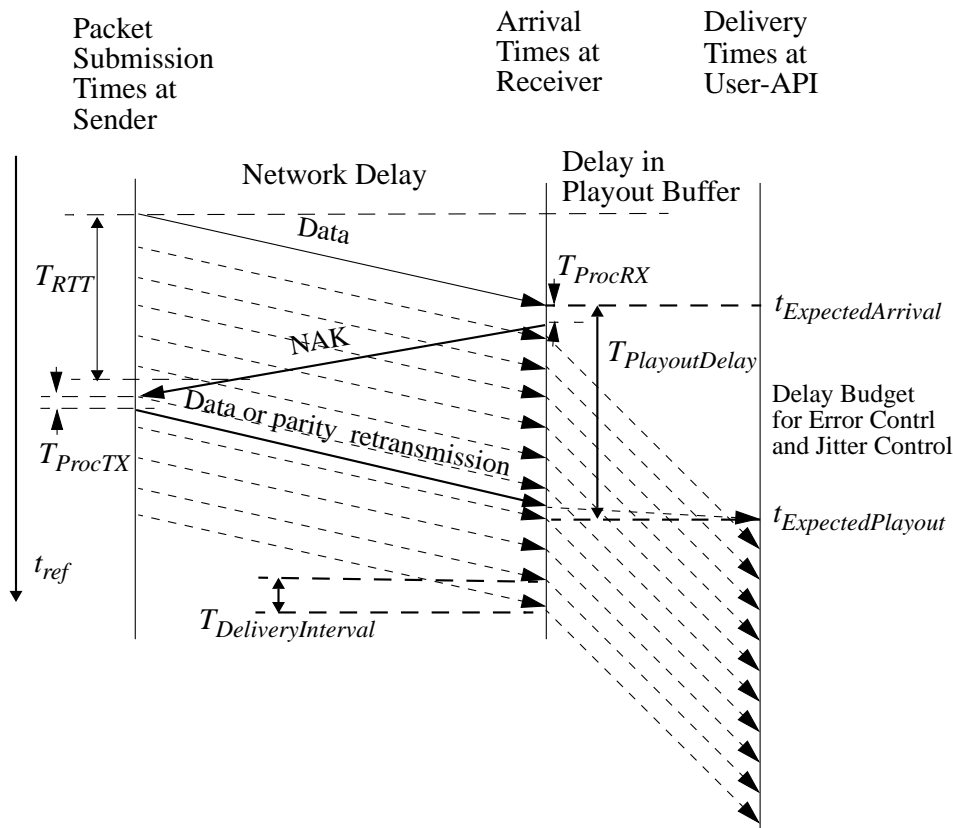


Figure 3: Diagram of error control within delay budget.

We see in figure 3 how a given delay budget can be used both for jitter control and error control. In the figure, the delay in the playout buffer ($T_{PlayoutDelay}$) is adjusted such that the expected playout time ($T_{ExpectedPlayout}$) of a packet will be met even in the case where a retransmission is necessary. The periodic packet submission times at the sender correspond to the delivery time interval ($T_{DeliveryInterval}$) at the transport service access point (Transport-SAP) of the receiver user API. The delay budget at the receiver must be dimensioned in a way that the network delay due to the initial transmission, the retransmission request (NAK), the first retransmission, the processing delay at the transmitter

(T_{ProcTX}), and the processing delay at the receiver (T_{ProcRX}) are taken into account. This will assure that the retransmission arrives at the receiver before the expected playout time.

2 Network properties

In order to meet the reliability requirements of a real-time application, it is possible to use a network service that directly provides the required reliability, without additional error control mechanisms in the transport layer. This can be ensured by reserving network resources for the application, or by dimensioning the network in a way that the residual error probability is sufficiently small (over-engineering).

In situations where a network service does not meet the reliability requirements of the application directly, additional error control mechanisms are required. By providing sufficient transport protocol processing capability with a low latency, it is possible to meet delay requirements of many audio-visual applications even after one or two retransmissions. This strategy potentially offers better utilization of network resources in particular for highly bursty sources.

2.1 IP-based Networks

In the following, we present the most important properties of IP networks with respect to the provision of real-time network services for audio-visual applications.

IP networks offer a datagram service with best-effort service quality with no guarantees for loss rate, delay, and in-sequence delivery. This service model is based on the hope that all traffic sources are well behaved. The congestion control algorithm of TCP ensures such a behavior. However, as more and more open-loop applications based on UDP are used, higher losses can be observed, and the future of this service model is seriously threatened.

One potential solution to overcome such problems is the support for resource reservation within the network, as offered by RSVP [68]. Therefore, in a subsequent section we present ATM networks, which provide support for reservations, and discuss their suitability for audio-visual applications. ATM networks currently play an important role in providing the underlying network services for IP networks. In the future, it can be expected that ATM networks also play a key role in the implementation of IP integrated services, as described in the following. Such future global IP networks will hide the diversity of the underlying network technology [59].

2.1.1 Measurements of IP Best Effort Service Model

To assess the suitability of the existing IP best effort service for supporting audio-visual applications, it is of interest to have detailed knowledge about typical quality impairments.

There is a significant number of existing publications that investigate the loss characteristic of the current Internet. IP losses for unicast communication are covered by Bolot [7] and Paxson [51]. IP losses for multicast communication over the MBONE are analyzed by Yajnik [66] and by Handley [25].

The measurements of [66] showed a relatively high loss probability in the access area and rather low loss probabilities in the backbone area. In such loss scenarios, error control schemes are attractive that involve servers, or that apply local recovery. However, certain backbone links like the perpetually congested US/UK link may cause high losses in the backbone area, limiting the effectiveness of these error recovery schemes.

2.1.2 IP Integrated Service Model

In the near future, Internet service providers will support different service classes, which offer still the same service type 'best-effort-service', but with different service qualities (i.e., delay characteristic, loss rate and characteristic).

In a future Internet, the widespread use of reservation protocols, such as RSVP, can be expected. In combination with access control techniques and scheduling mechanisms in all network nodes, reservation allows the provisioning of IP services with guaranteed quality of service. However, such networks will need some kind of tariffing to make the use of a service with guaranteed QoS more expensive than the use of a service without QoS guarantees. Therefore, users still will be motivated to apply powerful error control mechanisms in order to be able to use an inexpensive service class. In order to guarantee the required quality, *all* nodes on the path from the transmitter to the receiver need mechanisms to support QoS guarantees. Since it can not be expected for a relatively long time that IP services with guaranteed QoS will be ubiquitously available, powerful error control mechanisms will also continue to play an important role in the future.

2.2 ATM-based Networks

In contrast to IP networks, which are based on a connection-less network service, ATM networks offer a connection-oriented network service.

ATM networks provide all mechanisms required to offer services with guaranteed service quality. ATM can therefore be used for implementing IP integrated services. Several options exist for how to map the IP Integrated Service Model (supporting the service types best effort (BE), controlled load service (CLS) and guaranteed service (GS)) onto the ATM service model (supporting constant bit rate (CBR), variable bit rate (VBR), available bit rate (ABR), unspecified bit rate (UBR), asynchronous block transfer (ABT) and potential additional services; see [15, 22]).

While CBR and VBR service classes are well suited for audio-visual applications, proposed tariffs give a high incentive to use ABR and UBR service classes instead. One example is the proposed set of ATM tariffs for VBR, CBR, ABR, and UBR by Walker et al. [64], where the price per volume for VBR traffic is two orders of magnitude higher than the price per volume for UBR traffic. Due to space limitations, we do not discuss these alternatives in detail. Instead, we focus on the impact of ATM networks onto the error recovery mechanisms.

2.2.1 Error Characteristics in ATM Networks

For the provisioning of reliable services in ATM networks, error control mechanisms are required that detect and correct corrupted and lost cells. Depending on the origin of the errors, different error characteristics may be observed. As a consequence, different error control mechanisms may be identified to be more appropriate in a certain case.

Relatively high cell loss rates in an ATM network may occur for different reasons: One reason is the use of the Fast Reservation Protocol with Immediate Transmission (FRP/IT) for the ABT service class. Another is the assignment of different priorities to different VCs.

In high-speed wide area ATM networks, aggressive control algorithms (leading to relatively high loss rates) offer potential advantages such as higher network utilization, which can be traded off with additional overhead for error control they imply.

For multicast ABR services, different service models are possible. A conservative multicast service would limit the transmission rate according to the most congested link of a multicast tree, in order to achieve very low cell loss rates. In contrast, an aggressive ABR multicast service may use higher transmission rates, while producing cell losses at some congested links. This can lead to an even

higher packet loss due to the following reason: If a single cell of an AAL3/4 or AAL5 frame is lost, typically the whole frame is discarded.

2.2.2 Special Properties of ATM Networks Relevant for CM Error Control

The following special properties of ATM networks can be exploited by suitable error control mechanisms:

- ATM networks guarantee in-sequence delivery of cells within a connection.
- Reservation in combination with UPC (Usage Parameter Control) and/or scheduling algorithms in the nodes makes it possible to give bounds on delay and error probability [47].
- In ATM networks, VCs with different QoS properties can be set up to the same receiver. This can be exploited to obtain increased reliability for retransmissions, or to ensure low delay retransmissions.

Since retransmissions can be sent over ATM networks with higher probability of success and with lower delay, than in networks with only a single service type, the applicability of ARQ-based error control schemes will be extended to scenarios where retransmissions would not be suitable otherwise.

2.3 Comparison of ATM and IP Networks

An important characteristic of ATM networks is that delay bounds may be known (i.e., for the ATM service categories CBR, VBR). In other cases (i.e., for the ATM service categories UBR and ABR), where delay bounds are not known, the mean delay and delay variation of these service categories may be significantly smaller than mean delay and delay variation of conventional IP services.

In IP networks, which provide a best-effort type service, no delay bounds are given. However, mechanisms for access control and scheduling exist that allow to provide IP services with QoS guarantees, such as delay bounds. In order to transfer information on the required QoS parameters to all relevant network nodes and end systems, a signaling protocol is required. RSVP is a signaling protocol that supports the exchange of this information for multipoint services. Therefore, in the case of IP/RSVP-based networks, delay bounds can also be given.

3 Mechanisms for Multicast Error Control

In this section, a general discussion of mechanisms for multicast error recovery is presented and an overview on existing approaches is given.

Provision of reliable multicast services faces the following key challenges:

- Multiple receivers with heterogeneous connections and processing capabilities;
- Feedback implosion: If every receiver reports about the success or failure of the data transfer, the sender will be overwhelmed.
- Depending on where loss in the multicast tree happens, different error scenarios result. For a multicast tree with a common link and individual links to all receivers, losses on individual links result in different receivers losing different packets. Loss on a common link results in many receivers losing the same packet.

Figure 4 shows a simple example of a multicast tree. Losses on the common link from the source to the central node C are observed by *all* receivers. For large groups, error notification of a shared loss by all receivers using NAKs would overwhelm the source. When NAKs are sent by multicast, receivers may apply NAK suppression algorithms, usually trading off notification latency with reduced number of NAKs. Loss on the link from node C to R_1 will be observed first by R_1 , subsequently by R_3 . In case R_1 sends a NAK to the source immediately, R_3 may obtain a

retransmission after a smaller delay than its RTT. This indicates that for reliable multicast, retransmission-based error recovery can be performed within a recovery interval smaller than RTT of the respective receiver. Loss on an individual link such as the link from node C to R_2 will be observed only by a single receiver. In case a NAK suppression algorithm is used, receiver R_2 will issue a NAK later than in the case where no NAK suppression algorithm is used.

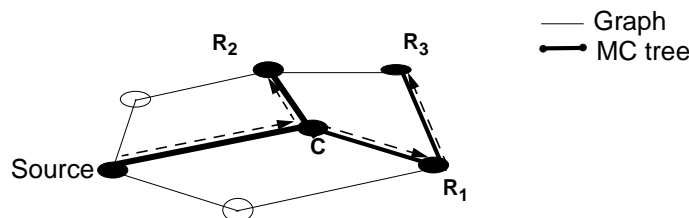


Figure 4: Network topology with multicast tree.

3.1 Multicast Error Control - Basic Mechanisms

In the following, alternative basic mechanisms for reliable multicasting are discussed. Existing approaches employ mechanisms for error detection, error notification, and error recovery which are either executed at a central point (i.e., by the source) or distributed (i.e., locally).

The approaches can be classified according to the following questions:

- Who detects loss: the sender (waiting for all ACKs), or the receiver (using NAKs).
- How to send feedback: using unicast or multicast, with the option of applying algorithms for NAK suppression.
- What is retransmitted: original data, or parity.
- Who retransmits: the source, servers within the network, or other group members.
- How to retransmit: using unicast, multicast, or subgroup-multicast.
- How to cope with heterogeneity: adapting to the slowest/most impaired receiver, or ignoring receivers that do not reach certain limits.

3.2 Loss Detection

According to the entity that performs error detection, sender-initiated protocols and receiver-initiated protocols can be distinguished.

For **sender-initiated protocols**, receivers return positive acknowledgments (ACKs). The sender is responsible for error detection and therefore must maintain and process status information associated with every single receiver. Such a scheme obviously has problems in case of large groups, as the sender has to process a large number of ACKs and as these ACKs may cause congestion and losses in the sender's neighborhood.

For **receiver-initiated protocols**, each receiver is responsible for error detection and therefore maintains its own state information. When a loss or a corrupted data packet is detected, the receiver issues a negative acknowledgment (NAK) by unicast or multicast. Generally, this approach reduces the number of NAKs and therefore reduces the processing load at the sender in comparison to the sender-initiated approach. The comparative analysis presented by [63] has shown that receiver-initiated protocols are far more scalable than sender-initiated protocols for large groups and provide a higher throughput.

3.3 Feedback

3.3.1 Feedback: Unicast versus Multicast

Receivers have two options for sending feedback: by unicast only to the sender, or by multicast to the sender and all the other receivers. In [52], these alternatives are compared for real-time multicast transport protocols with source-only retransmission of original data. Cases are identified where control messages should be sent by multicast, thereby allowing the receivers to implement feedback suppression schemes. Multicasting control messages proved to be useful when receivers are locally concentrated. When receivers are far apart, multicasting of control messages proved to have a negative impact on delay properties, due to the large propagation delay of NAKs to other receivers and due to the time receivers must wait to achieve the desired suppression of NAKs. In addition, multicasting of NAKs to receivers far apart consumes significant network resources in comparison with unicast transmissions of NAKs to the sender.

3.3.2 NAK suppression: Slotting and Damping Algorithms

NAK suppression requires a distributed algorithm to be executed by the receivers. NAK suppression is desirable since it reduces the amount of feedback the sender must process. With NAK suppression, in the best case at most one NAK per lost PDU would reach the sender.

Two generic algorithms for NAK suppression, called **slotting** and **damping**, are proposed in RTP Version 3.6 [61]. Slotting describes a mechanism where receivers send feedback via multicast either immediately, or after a delay of one or more time slots. Damping describes a mechanism in which a receiver suppresses its own feedback packet if feedback from other receivers corresponds to its own state. Slotting and damping algorithms may significantly increase the latency in a WAN environment if only a small number of receivers are affected by the same loss.

Various protocols employ the basic slotting and damping idea with some optimizations. For instance, the TMTP [67] protocol applies these mechanisms in the local scope in order to reduce delays and redundant retransmissions. The effect of localization is achieved by using the TTL field in an IP packet.

The SRM [21] protocol integrates damping and slotting algorithms to implement a receiver-based error control. Timers must be carefully set to avoid a flood of control packets in case of high loss.

NAK suppression also has disadvantages. It increases the mean delay until the first NAK reaches the transmitter, since the receivers wait a certain time interval before sending a NAK. As NAK are sent by multicast, more bandwidth may be consumed for multicasting NAKs, than in the case of sending control messages by unicast. Depending on the underlying multicast routing protocol, the required amount of status information in the routers can be much higher when every receiver is sending feedback via multicast.

3.3.3 Data retransmission versus parity retransmission

Retransmission of missing original data (ARQ) is the most popular approach. However, as shown in [45], retransmission of *parity* results in much improved throughput and bandwidth usage for reliable multicast to a large number of receivers.

Parity for retransmission can be produced by the transmitter based on a predefined transmission group, or by a sliding window with predefined length, covering the last k packets that have been successfully delivered to the receiver.

There exist cases for which retransmissions of parity packets for predefined transmission groups is not possible or not desirable. One example would be *local* retransmission of parity in cases where no member of a local group did receive all packets of a transmission group. In such cases it is

advantageous to employ an error recovery mechanism where parity information is calculated over a dynamically determined group of packets. Such mechanisms are proposed in [1], where parity is generated using XOR-operations.

3.4 Source-based versus Decentralized Retransmission

3.4.1 Source-based Retransmission

There exist a large number of protocols that employ source-based retransmission, which has the advantage of simplicity. In case of retransmission of *original* data, this approach suffers from limited efficiency in case of large groups and significant loss rates, as retransmissions may also get lost, and as large groups may require a large number of retransmissions.

An important example is the XTP protocol [61]. In case of large groups and significant error probabilities, XTP does not allow efficient error recovery and leads to high bandwidth requirements.

Additional examples for multicast protocols that employ source-based feedback processing and error recovery are MTP [4], RAMP [9] and others [19].

In [45], the protocol NP is proposed which employs a type 2 hybrid ARQ scheme. In NP, the sender transmits a TG consisting of k data packets. Each time that a receiver detects a missing packet, it requests a new parity packet from the sender until it has a sufficient number of packets (k) for decoding of all k data packets of the transmission group. In every transmission round, the sender sends the number of parity packets which corresponds to the *maximum* number of lost packets for every receiver in the previous transmission round. Feedback from the receivers is also sent via multicast, coupled with NAK suppression as in SRM [21]. The protocol NP was shown to achieve a better bandwidth usage than other protocols with source-based error recovery [45].

3.4.2 Decentralized Retransmission

Another concept used to achieve better scalability in one-to-many protocols considers distributing source responsibilities for error control over a number of dedicated servers which we call in the following Group Communication Servers (GCSs).

These servers can be used for (1) feedback processing and (2) local retransmissions, thereby improving bandwidth efficiency and latency. Additionally, GCSs can be used for (3) transformation between different error control schemes.

Improved bandwidth efficiency and reduced delay can be achieved by dividing a global multicast group into separate subgroups. These subgroups consist of a number of receivers within a local region. Instead of sending control packets directly to the sender, receivers send them to the dedicated GCS. When possible, a GCS will retransmit requested packets directly. The sender deals only with the set of GCS instead of communicating simultaneously with a large set of receivers.

A number of proposals for protocols using GCSs with a specific functionality exist. In the Consolidated Status Protocol (CSP) [49, 50], GCSs are situated in local exchanges, combining control packets from receivers of a subgroup into consolidated control packets that are sent to the sender. In the Designated Status Protocol (DSP) [49] and Local Group Concept [27], the functionality of GCSs is realized in end-systems that are members of the multicast group. In these protocols GCSs are responsible not only for ACK processing for receivers in the subgroup, but also for performing local retransmissions in their subgroups. Based on these results of [49], the Reliable Multicast Transport Protocol (RMTP) [37] was developed, featuring local feedback processing and error recovery by local retransmissions. Another server-based protocol is the "log-based receiver-reliable multicast" (LBRM) [28] which uses a hierarchy of log servers for feedback processing and retransmission. The Group Communication Server (GCS) presented in [10] provides mechanisms for

ATM-specific hybrid error control. The integration of FEC mechanisms into the GCS allows for the regeneration of lost cells and for the reinsertion of additional redundancy for adjusting the amount of redundancy according to the needs of subsequent hops. GCS placement remains an important problem that still needs to be resolved.

3.5 Heterogeneity

A reliable multicast protocol has to support the possibility that not all receivers in the multicast group have the same processing capabilities and network resources. In such a heterogeneous environment several policies can be used. Applying a conservative policy, the multicast communication is adapted to the needs of the slowest or most impaired receiver. Some protocols are able to provide several levels of service for different receivers. Alternatively, a protocol can use a restrictive policy to eject slow receivers that represent a performance bottleneck.

Instead of using some dedicated GCS, the SRM protocol implements distributed error recovery by allowing every receiver to be a retransmitter. If a receiver indicates via a NAK that is multicast to all receivers that he has not received a certain data packet, any receiver that has successfully received the data can perform the retransmission. Multiple retransmissions by different receivers are avoided using a damping mechanism similar to the one described for feedback suppression.

3.6 Reliable Real-time versus Fully Reliable Multicast Protocols

Protocols for real-time multicast and for fully reliable multicast differ in a number of aspects which are related with the fact that after passing a given deadline, successful delivery of data to the receivers is no longer important. However, protocols for real-time multicast are based on the same basic mechanisms for feedback processing and error recovery. Within the given delay budget, feedback processing and error recovery is subject to the same performance tradeoffs as in the case of a non-real-time-service. Therefore, our performance consideration for the required bandwidth and for the delay properties give also valuable insight into the suitability of source-only vs. local error recovery for real-time services.

Transport protocols for real-time reliable multicast services have in common the tradeoff between reliability and delay budget. By increasing the delay budget available for error recovery, the residual error probability in terms of non-recoverable errors within the allowed delay can be reduced.

4 Error Control Mechanisms for CM data transmission

In this section, existing approaches for transport protocols that employ ARQ and FEC for providing reliable real-time multicast services are discussed.

4.1 ARQ-based Error Control for CM Data Transmission

4.1.1 Audio Transmission

Dempsey and Liebeherr were the first to investigate retransmission for CM applications [16, 17] for the case of a *unicast interactive voice* transmission over *local area* networks. Their approach for a protocol that performs NAK-initiated retransmissions within a given delay budget is called **Slack-ARQ**. Given round-trip times in the order of 10 msec and inter-packet gaps of 20 msec, Dempsey demonstrated that a playout delay at the receiver (introduced by delaying the first packet in a talkspurt) of about 100 msec will allow to obtain an acceptable quality voice transmission. A timely (successful) playback probability of at least 90% is achieved in case of random and burst packet loss.

For transmission over the Internet, where delay and jitter are frequently in the order of a few hundred milliseconds, the support of *interactive* voice transmission using retransmission-based protocols has not yet been demonstrated. Considering the fact that interactive voice applications require round-trip delays of less than 200 msec [34], retransmission generally does not seem a feasible option. However, the situation is different for *distribution* of voice over the Internet. Recently, Xu, Myers, Zhang and Yavatkar [65] have investigated the use of retransmission for the delivery of non-interactive voice over the Internet to multiple recipients. Given a playout delay in the order of 500 msec, retransmission for loss recovery is feasible. The authors argue that in the case of non-interactive voice, the receivers can make a trade-off between increased reliability and lower latency by choosing the playout delay appropriately.

The protocol proposed, which is referred to as Structure-Oriented Resilient Multicast (**STORM**), does *local* loss recovery to achieve scalability and lower recovery times. **STORM** works as follows:

- The receiver detects loss using gap-based loss detection and uses NAKs to request the retransmission of lost packets.
- Each receiver has a list of **parent** nodes, from which he chooses one node to send the NAK to. If that node does not reply with a retransmission, the receiver will timeout and send the NAK to another parent node.
- The NAKs and the retransmissions are done via unicast to keep the overhead due to loss recovery low.

The experiments performed with a playout delay of 200 and 500 msec indicate that **STORM** is able to reduce the loss rate seen by a receiver by at least one order of magnitude. Packet loss rate before recovery in the order of a few percent can often be reduced to zero loss after recovery.

4.1.2 Video Transmission

The previous two examples studied evaluated retransmission-based loss recovery for audio applications, where the bit rate is fairly low, i.e. in the order of a few tens of Kbit/sec. For video transmission, the bit rates seen are typically much higher. MPEG for instance requires a bit rate of about 1.5 Mbit/sec. Such a high bit rate poses the problem of **rate control** to multiple receivers: Given that across the Internet the bandwidth available between a source and a receiver is generally much lower than 1.5 Mbit/sec and that the bandwidth available to different receivers can vary by an order of magnitude or more, sending at the same rate to all receivers could cause severe congestion and high loss in parts of the network.

One way to allow for rate control and scalability, is to use **hierarchical coding schemes**, where the signal is encoded in a base layer that provides a low quality image and additional complementary layers for improved image quality. Each receiver needs to receive at least the base layer. Different layers of the video signal are transmitted on different multicast groups and each receiver chooses to receive the base layer and as many layers as the bandwidth available along the path allows. This approach is also referred to as **receiver-driven layered multicast** or **RLM** for short [42, 43].

Li, Paul, Pancha and Ammar [36] have recently proposed a retransmission-based loss recovery protocol, called Layered Video Multicast with Retransmission (**LVMR**), for non-interactive transmission of MPEG video to multiple receivers across the Internet.

- The MPEG video stream is separated into three layers: The base layer contains I-frames only, the other two layers contain P- and B-frames respectively.
- When the receiver detects a loss of a frame, he can send a NAK to request the retransmission. A retransmission is only requested if the data is likely to be received before it is required for playout. The recovery time depends on the round-trip time between receiver and the node that retransmits the frame and the processing times at reception.

- For LVMR loss recovery is local: Each receiver has a **designated receiver** (DR), to whom the receiver will send the NAK to.
- The NAKs and the retransmissions are done via unicast to keep the overhead due to loss recovery low.

The experiments performed for a playout delay of at least 1500 msec indicate that LVMR is able to recover around 80% of the losses seen by a receiver.

4.2 FEC-based Error Control for CM Data Transmission

4.2.1 FEC Schemes Suitable for IP Networks

Today, an important limit for widespread use of telephony and video-conferencing over the Internet is bad service quality due to losses in congested routers. A number of interactive applications employ FEC for real-time error control with stringent delay requirements. As IP services may already have significant one-way-trip-times due to queuing in routers, special care is required in the development of the FEC scheme in order to obtain acceptable delay properties.

For audio applications that apply codecs with relatively long sampling intervals (e.g. as used for GSM), the resulting data stream has a relatively low packet rate, further complicating the design of a suitable FEC scheme. A number of applications were developed that have an application-specific FEC scheme with good delay properties.

The INRIA freephone [8] achieves good delay properties by encoding the audio stream using two different coding standards, and by transmitting encoded samples of the same time interval in subsequent packets: The data stream of freephone contains in every packet a PCM-encoded sample of one time interval, together with a redundant version of the *previous* time interval encoded at a lower rate. This FEC scheme has the advantages of adding only little bandwidth overhead to a PCM encoded audio stream, and of not increasing the IP packet rate while achieving relatively high robustness against losses.

An example for a video-specific FEC scheme is the Priority Encoding Transmission (**PET**) developed at ICSI, Berkeley [60, 2, 3]. This technique allows a user to specify a different priority for each segment of a continuous media stream. According to the assigned priority, PET generates a different amount of redundancy for the segments and disperses user data and redundancy onto several subsequent packets. PET can be applied to the transmission of MPEG video streams such that the data of one GOP is dispersed over a sequence of packets and that I-Frames are protected with a higher amount of redundancy than P-Frames, which are protected by a higher amount of redundancy than B-Frames. A typical dimensioning would be to protect I-, P- and B-Frames with 100%, 33%, and 5% redundancy, respectively, and to disperse the data of a GOP over packets of 2 Kbyte length. PET was integrated into the MBONE video conferencing tool vic [2] and was shown to work in combination with MPEG-1 and MPEG-2 data streams.

The Real-Time Transport Protocol (**RTP**, [57]) is a transport layer protocol framework which has been developed by the IETF (Internet Engineering Task Force) Audio/Video Transport working group in order to support delivery of continuous media over the Internet. The protocol defines a data packet semantic with timing information, packet sequence numbers, and optional parameters. Various payload formats are defined for different audio and video compression standards. No specific error control mechanism is defined. However, it is possible to adapt the framework specified by RTP by defining application-specific error control mechanisms. Current contributions within the IETF describe how RTP can be combined with FEC (**RTP+FEC**, see [56]), based on the previously developed tools freephone [8] and rat [26]. Additionally, it has been proposed to combine RTP with the receiver-initiated retransmission scheme from SRM ([48]). The RTP framework also defines a control protocol RTCP (Real Time Control Protocol) which allows to collect feedback from the

receivers. RTCP also can be adapted to application-specific needs. RTP is not only widely incorporated into Internet applications, but also adopted by ITU for the H.323 [31] recommendation defining audio-visual telephone systems for local area networks.

4.2.2 FEC Schemes Suitable for ATM Networks

A number of different approaches have been presented for applying an ATM-specific FEC scheme. However, until now no approach received high attention, and no plans are known for widespread use of one of the approaches.

For AAL1 [29], it is envisaged to apply a Service Specific Convergence Sublayer (SSCS) with FEC as error control for real-time services. Two different schemes are proposed in [30]. One scheme is based on a Reed-Solomon-Code applied on blocks of 128 cells. The scheme allows for the regeneration of up to four missing cells. Alternatively, two corrupted cells can be detected, and the errors can be corrected. This scheme is called the Long-Interleaver Scheme and introduces additional delay which makes it not very suitable for interactive real-time applications. It leads to a delay of 128 cells at the transmitter due to interleaving. In order to allow the correction of errors, an additional delay of 128 cells is introduced at the receiver. An additional scheme (called Short Interleaver Scheme) was also proposed by ITU SG13. It has more limited error control capabilities, but better delay properties. The scheme introduces a delay of 16 cells at the transmitter and at the receiver, and allows to recover a single missing cell per block of 16 cells.

AAL1 has not been widely implemented, but AAL5 is generally supported. Therefore FEC schemes for AAL5 are of higher importance than the FEC schemes developed within ITU. In [11], a cell-level FEC scheme for the SSCS of AAL5 called FEC-SSCS is presented which can be used for a variety of application and network scenarios. The scheme allows a dynamic adaptation of the amount of redundancy. In [44], an ATM-specific FEC scheme is presented which provides a solution for an extension to the ATM Adaptation Layer AAL5 which is optimized for hierarchical data streams according to MPEG coding. The proposal applies a block-based bit interleaving scheme to spatially disperse the cell loss effect on ADUs which is proposed to be used for MPEG I-Frames. The different priorities of MPEG ADUs (I-, P- and B-Frames) are mapped onto CLP (Cell Loss Priority) and PT (Payload Type) mechanisms of the ATM layer.

5 Taxonomy of Protocol Mechanisms for Real-time Audio-visual Services

5.1 Introduction

In contrast to traditional point-to-point communication, the suitability and performance of multicast protocol mechanisms crucially depends on specific properties of the communication scenario.

Network scenarios are determined by error and loss characteristics, by delay characteristics, and by the fanout characteristics. Losses and errors can be specified further by probability and type of correlation (spatially or temporally correlated) and by diversity (homogeneous or heterogeneous distribution). Delay can be specified further by first and second moment (mean delay and delay variation) and also by diversity (homogeneous or heterogeneous distribution). The fanout characteristics can be specified further by topology (including fanout distribution within the multicast tree) and by the total number of receivers. In addition, it can be distinguished by whether receivers can directly communicate with each other or not. For direct communication among receivers, the multicast network service must support multiple sources (M:N communication). Where receivers do not need to communicate with each other, a single communication channel for feedback to the sender is sufficient.

5.2 Network Scenarios with Characteristic Properties

In the following, two representative network scenarios are discussed. The key difference between these scenarios is the location where losses occur: mostly on individual links (individual loss scenario N1), or mostly on common links (shared loss scenario N2). Models for these two scenarios are shown in figure 5. Additional classifications are possible with respect to the distribution of the loss probability (homogeneous vs. heterogeneous loss rate) and to the round-trip-time (homogeneous vs. heterogeneous RTT).

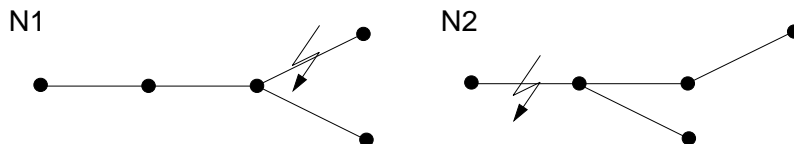


Figure 5: Diagram describing the reference network scenarios.

5.2.1 Network Scenario with Individual Loss

Individual losses play an important role for many multicast scenarios. Measurements within the MBONE, such as those performed by [66], have shown that individual losses are predominant, since the majority of losses occurred in subnets. Additional examples of individual losses are point-to-multipoint connections in wireless terminals, where receivers are located in different cells, and satellite communication with dominating downlink errors.

For multicast error recovery by retransmission of missing packets, individual losses lead to high bandwidth consumption and poor scaling for large groups. In addition, selective retransmission of missing packets requires a large amount of control messages and transmitter status information.

One solution that reduces bandwidth consumption and improves scalability is to use a hybrid error control scheme with parity retransmissions. An alternative solution is to use local retransmission.

Individual losses in combination with a *heterogeneous error rate* may lead to QoS deterioration for receivers with low error rates. Due to error recovery performed by the transmitter for receivers with high error rates, low loss receivers will unnecessarily receive some data multiple times and see a reduction of the throughput. In case of local error recovery, receivers with good link quality may perceive reduced performance or an increase in network utilization cost due to retransmissions they perform for receivers with poor link quality. Heterogeneous error rates may lead to *unfairness*, as receivers may have paid higher prices to obtain a network access with higher quality.

Fairness can be improved by scheduling at the transmitter to avoid delay for transmissions to receivers with low error rate. Appropriate scheduling policies can give priority for first transmissions over retransmissions, or can use rate-limitations for retransmissions. Further improvements are possible by scheduling that distinguishes different retransmission rounds and different receivers, or different receiver classes. Destination Set Grouping that separates receivers according to their loss rate [12] is one example for a multicast error recovery scheme suitable for heterogeneous error rates.

5.2.2 Network Scenario with Shared Loss

Shared loss is located before fanout nodes within a multicast tree, as shown by scenario N2 in figure 5. Shared loss may occur at IP multicast routers, at ATM switches of point-to-multipoint ATM connections, at the transmitter, for example by leaky bucket usage parameter control (UPC) at the network access, or may be introduced at the physical layer, e.g. at a satellite uplink.

Shared loss leads to several problems. As many receivers are subject to loss, NAK implosion may happen. Local retransmission is not appropriate for recovery from shared loss. When protocols with local recovery are used, receivers face the difficulty to distinguish individual losses - where local recovery is appropriate - from shared losses. While FEC and parity retransmission are very efficient

for recovery from individual losses (as the same redundancy can be used to correct different losses), these techniques have no impact for scaling to large numbers of receivers when used to recover from shared losses.

A suitable solution for recovery from shared loss is hybrid ARQ, where FEC with sufficient amount of redundancy is used to ensure high delivery probability at first and subsequent transmission attempts. Error detection close to the location where loss occurred (e.g. in routers or in transport-level Group Communication Servers) allow retransmission-based error recovery with small delay. In addition, a NAK concentration functionality within the network is useful to reduce NAK implosion.

Shared losses for groups with *heterogeneous RTT* introduce the problem that NAKs from different receivers, arriving at different times, refer to the same loss. This may lead to unnecessary retransmissions. To avoid them, a transmitter must perform RTT-aware NAK processing. An alternative approach to avoid unnecessary retransmissions is NAK processing by routers, servers, or receivers that is adapted to the multicast topology.

Receiver groups with heterogeneous RTT have a number of benefits for CM error control. The NAK implosion problem is less severe than in the case of homogeneous RTTs, as NAKs are dispersed over time. Additionally, error detection by a receiver or other protocol entity within the network that has a low RTT allows a fast retransmission within a time interval that is frequently below the RTT between the source and other receivers that perceived the loss. Therefore, retransmission-based error control can be applied to recover from shared losses even for delay budgets that are *smaller* than the RTT of an individual receiver. This leads to the non-intuitive conclusion that heterogeneous RTTs can have a *delay-reducing* property, which broadens the applicability of retransmission-based error recovery schemes significantly.

5.2.3 Overview of General Error Control Concepts

Based on the basic mechanisms for error recovery from ARQ and FEC, the following general error control concepts can be identified.

- **Progressive amount of redundancy:**
With increasing number of the transmission round, the amount of redundancy is increased, as the remaining delay budget decreases.
- **Priority-dependent amount of redundancy:**
The amount of redundancy can be selected by distinguishing ADUs with application-specific priority and by determining the impact of corrupted/lost ADUs onto application-level QoS (see [2, 60]).
- **NAK-initiated retransmission cycles:**
Where NAKs are used to initiate retransmissions, mechanisms to suppress unnecessary retransmissions are desirable. For transmitters that are aware of RTTs of individual receivers, this can be achieved by detecting NAKs referring to common losses for which retransmissions were performed within a specific RTT. For receivers that apply slotting and damping, the amount of unnecessary NAKs can be decreased [61].
- **Delay-budget aware calculation of redundancy:**
This concept is an open-loop technique. The delay budget defines a time interval relevance. A specified fraction of bandwidth is used to transmit (and retransmit) redundant data. Redundancy can be organized in layers. Convolutional codes are used to generate redundancy. Dimensioning of redundancy can be performed according to the rules presented in [32].

Most mechanisms allow an advantageous combination with other mechanisms. The mechanisms have been investigated, but mostly have not yet been integrated into protocols currently available for CM applications.

5.3 Assessment of Error Control Mechanisms for Different Scenarios

This section summarizes the properties of the presented error control mechanisms and gives an assessment of their suitability for different network and application scenarios.

5.3.1 Open-loop Error Control with FEC

FEC has the inherent advantage of simplicity at transmitter and receivers. This facilitates protocol implementation and makes the adjustment of protocol parameters less complex. While buffer requirements for retransmission-based error control increase with increasing path capacity, buffer requirements for FEC only depends on the choice of protocol parameters.

For varying error rates, open-loop error control with FEC has the big drawback of difficulty for dimensioning: A-priori selection of the optimal amount of redundancy is impossible. In order to be effective when needed, sufficient redundancy has to be transmitted all the time, leading to a waste of network resources as well as local resources at transmitter and receivers.

It is not only a drawback that the optimal amount of redundancy is not known a priori, but also that no precise knowledge is available about where and when errors occurred. Therefore, in an open-loop error control scheme with FEC, always a significant number of redundancy packets will be transmitted over the network without an actual need. This leads to unnecessary load, which again leads to additional losses by congestion.

FEC therefore is of particular benefit in the following scenarios:

- Large groups;
- Large RTTs;
- Feedback is impossible or undesired;
- Individual loss dominates;
- Homogeneous loss probability;
- Limited buffer.

5.3.2 Retransmission-based Error Control with ARQ

Retransmission-based error control is able to adapt to varying losses (temporal and spatial). At the same time, it faces problems that are avoided by FEC: Error recovery delay depends on RTT and large groups may lead to feedback implosion.

ARQ is therefore of particular benefit in the following scenarios:

- Small groups;
- Heterogeneous loss probability;
- Loss on shared links of multicast tree dominates;
- Non-interactive applications.

It can further be distinguished based on whether retransmissions are sender-based or local and whether original data or redundancy is retransmitted.

ARQ with local recovery is beneficial for large groups and well suited for individual losses and heterogeneous RTT. Due to simplicity, in all cases where the benefits of local recovery are not obvious, a sender-based recovery is preferable.

- Retransmission of original data:
Due to its simplicity, retransmission of original data is attractive in cases with a limited number of receivers and where shared losses are dominating. It is also useful for receivers with limited processing capabilities, where FEC processing leads to significant processing delay.
- Retransmission of parity information:
However, in multicast scenarios, retransmission of parity information frequently is the most attractive solution. It has the following features: It is not necessary to transmit precise information about *which* original data was lost to the transmitter, but only about *how much*, and it is possible to correct different errors at different receivers with a single retransmitted packet.

5.3.3 Assessment of Mechanisms using Application Scenarios

Table 3 summarizes a preferable choice of mechanisms, based on the suitability of the approaches for different applications and on strengths and weaknesses of the different approaches for different network scenarios. In addition, it lists example protocols that are suitable for the combination of application scenario and network scenario.

Application/ Network- Scenario	A1 <i>interactive voice</i>	A2 <i>interactive video</i>	A3 <i>non-inter- active video</i>
Individ. Loss (N1) Suitable Mechanisms Example Protocol	FEC, local recovery STORM, RTP+FEC	FEC, local recovery RTP+FEC, PET, RLM, LVMR	Local recovery LVMR
Shared Loss (N2) Suitable Mechanism Example Protocol	ARQ (low or heterog. RTT); hybrid ARQ Slack-ARQ	ARQ (low RTT); hybrid ARQ Slack-ARQ	ARQ (RTT- aware) FEC LVMR PET

Table 3: Summary for choice of suitable error control mechanisms and available protocols, depending on application and network scenario.

5.4 Communication Subsystems Supporting Different Error Control Schemes

As shown in the previous discussion, it is necessary to know application-specific and also network-specific parameters in order to select the most appropriate error control scheme. Important alternatives are pure FEC, source-based recovery by retransmission of parity, and local recovery by retransmission of missing original data.

There are two possible approaches that allow a selection of the most appropriate transport level error control scheme.

One approach is to design a single protocol that provides the full set of protocol mechanisms and that can be configured with respect to application scenario and network scenario. Configuration can be performed at connection setup time, or dynamically during the live-time of the communication.

To date no protocol has been designed that supports all error control schemes of interest and that is applicable for different CM applications. Considering the large amount of ongoing activities in protocol development, it seems likely that such a protocol, or even different protocols that offer all error control schemes of interest will be designed and successfully deployed. Typical target end systems are PCs that are powerful enough for protocol processing while meeting requirements of CM applications. A more difficult problem to resolve is the correct parametrisation of complex transport-level protocols. By implementing functionality to measure network properties and to estimate key parameters it is possible to implement algorithms for protocol configuration. Due to the high complexity and the dynamic nature of network characteristics and application/user behavior, it can be expected that a variety of alternative approaches will be realized, improving and optimizing protocol configuration.

An alternative approach to the design of a single protocol that provides all error control mechanisms of interest is to use a communication subsystem that selects between different protocols for CM error control. The large number of protocols that are concurrently supported by existing operating systems show the feasibility of this approach. However, selection of the appropriate protocol and its parametrisation is also a non-trivial task. Algorithms for this task can either be integrated within the application, or within the communication subsystem.

Both approaches will most likely co-exist in practice.

6 Conclusions

We have provided an overview of existing approaches for transport-layer error control for continuous media streams involving both, ARQ-based schemes and FEC-based schemes.

In contrast to widespread beliefs, ARQ-based schemes can be applied in a large number of cases for interactive audio-visual applications. However, several open questions can be identified that need to be solved in order to distinguish cases in which the application of ARQ-based schemes is useful from cases in which it is not.

The appropriate combination of ARQ and FEC is the key to achieve optimal performance. Hybrid ARQ where initial transmission is protected by redundant information provides improved delay characteristics. Retransmission of parity allows for scalability for large receiver groups and most efficient usage of bandwidth.

We have shown that a variety of protocol mechanisms for error control are available. However, no mechanism works best for all scenarios. As none of the existing protocols offer the full set of error control mechanisms, no protocol currently achieves the highest performance in all cases.

For better support of IP-based CM applications it is required to either develop protocols that use a larger set of error control mechanisms or to develop communication subsystems that support the selection of the most suitable protocol. In both cases, functionality is required to select the appropriate protocol mechanism or the appropriate protocol. This functionality can either be integrated within the application or within the operating system as service for all applications.

We have reviewed a variety of proposals for error recovery. We have seen that there does not yet exist a general solution viable under all network conditions and applicable for all CM applications. It is still an open question if there will ever be a single error recovery protocol for CM multicast communication as it exists with TCP for reliable point-to-point communication.

Acknowledgements

The support by Siemens AG, Munich, within a cooperation project with Institut Eurecom on *Protocol Modules for Multicast Audio-Visual Services* and the support of G. Carle by the European Commission in form of a TMR (Training and Mobility for Researchers in Europe) grant is gratefully acknowledged. We would also like to thank the anonymous reviewers for their helpful comments.

References

- [1] M. Aghadavoodi-Jolfaei, S. Martin, and J. Mattfeldt, "A New Efficient Selective Repeat Protocol for Point-To-Multipoint Communication", In *IEEE International Conference on Communication (ICC)*, Geneva, Switzerland, 1993.
- [2] A. Albanese and M. Luby, "PET - Priority Encoding Transmission", chapter in *High Speed Networking for Multimedia Applications*, Kluwer Academic Publishers, March 1996.
- [3] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority Encoding Transmission", *IEEE Transactions on Information Theory (special issue devoted to coding theory)*, 42(6), November 1996.
- [4] S. Armstrong, A. Freier, and K. Marzullo, "Multicast Transport Protocol", Request for Comments (Informational) RFC 1301, Internet Engineering Task Force, February 1992.
- [5] B. N. Levine and J. J. Garcia-Luna-Aceves, "A Comparison of Known Classes of Reliable Multicast Protocols", In *Proc. International Conference on Network Protocols (ICNP-96)*, Columbus, Ohio, October 1996.
- [6] E. W. Biersack, "Performance Evaluation of Forward Error Correction in an ATM Environment", *IEEE Journal on Selected Areas in Communications*, 11(2):631-640, May 1993.
- [7] J.-C. Bolot, "End-to-end Packet Delay and Loss Behavior in the Internet", In *Proceedings of ACM Sigcomm '93*, pp. 289-298, San Francisco, CA, 1993.
- [8] J. C. Bolot and A. Vega Garcia, "The case for FEC-based error control for packet audio in the Internet", *ACM Multimedia Systems*, 1997.
- [9] R. Braudes and S. Zabele, "Requirements for multicast protocols", Request for Comments (Informational) RFC 1458, Internet Engineering Task Force, May 1993.
- [10] G. Carle, "Adaptation Layer and Group Communication Server for Reliable Multipoint Services in ATM Networks", In *R. Steinmetz (Ed.): Advanced Teleservices and High Speed Communication Architectures; Lecture Notes in Computer Science 882*, pp. 124-138, Second International Workshop IWACA '94, Heidelberg, September 1994, Springer-Verlag.
- [11] G. Carle, H. Esaki, A. Guha, K. Tsunoda, and K. Kanai, "Proposal for Specification of FEC-SSCS for AAL Type 5", Contribution ATMF/95-0326, ATM Forum Technical Committee 'Service Aspects and Applications SA&A', Denver, Colorado, U.S.A., April 1995.
- [12] S. Y. Cheung, M. H. Ammar, and X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution", In *Proceedings of IEEE INFOCOM'96*, pp. 553-560, San Francisco, CA, April 1996.
- [13] T. Chiang and D. Anastassiou, "Hierarchical Coding of Digital Television", *IEEE Communications Magazine*, 32(5):38-45, May 1994.
- [14] D. D. Clark and D. L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols", In *Proc. ACM SIGCOMM 90*, pp. 200-208, Philadelphia, PA, September 1990.
- [15] J. Crowcroft, Z. Wang, A. Smith, and J. Adams, "A Rough Comparison of the IETF and ATM Service Models", *IEEE Network*, 1995.

- [16] B. J. Dempsey, M. T. Lucas, and A. C. Weaver, "An Empirical Study of Packet Voice Distribution Over a Campus-Wide Network", In *Proceedings of the IEEE 19th Conference on Local Computer Networks, October 1994, Minneapolis, Minnesota, 1994*.
- [17] B. Dempsey, J. Liebeherr, and A. Weaver, "On Retransmission-Based Error Control for Continuous Media Traffic in Packet-Switching Networks", *Computer Networks and ISDN Systems*, 28(5):719 - 736, March 1996.
- [18] H. R. Deng and M. L. Lin, "A Type I Hybrid ARQ system with Adaptive Code Rates", *IEEE Transactions on Communications*, COM-43(2/3/4):733-737, February/March/April 1995.
- [19] C. Diot, W. Dabbous, and C. J., "Multipoint Communication: A Survey of Protocols, Functions and Mechanisms", *IEEE Journal on Selected Areas in Communications*, 15(3):277-290, April 1997.
- [20] H. Eriksson, "MBONE: The Multicast Backbone", *Communications of the ACM*, 37(8):54-60, August 1994.
- [21] S. Floyd, V. Jacobsen, S. McCanne, C.-G. Liu, and L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", In *Proceedings of SIGCOMM'95*, volume 25, pp. 342-356, ACM, October 1995.
- [22] M. Garrett and M. Borden, "Interoperation of Controlled-Load and Guaranteed Services with ATM", Internet-Draft: draft-ietf-issll-atm-mapping (Work in Progress), Internet Engineering Task Force, July 1997.
- [23] M. Ghanbari and V. Seferidis, "Cell-Loss Concealment in ATM Video Codecs", *IEEE Transactions On Circuits and Systems for Video Technology*, 3(3):238-247, June 1993.
- [24] F. Halsall, *Data Communication, Computer Networks, and Open Systems*, Addison-Wesley Publ., Reading, MA, 4th edition, 1996.
- [25] M. Handley, "An Examination of MBONE Performance", Technical Report, University of Southern California, Information Sciences Institute USC/ISI, ISI/RR-97-450, January 1997.
- [26] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable Audio for Use over the Internet", In *Proceedings of Inet'95*, Honolulu, Hawaii, June 1995.
- [27] M. Hofmann, "A Generic Concept for Large-Scale Multicast", In B. Plattner, editor, *Proc. International Zuerich Seminar*, Volume 1044 of *LNCS*, pp. 95-106, Springer Verlag, February 1996.
- [28] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation", In *Proceedings of ACM SIGCOMM'95*, Boston, MA, October 1995.
- [29] ITU-T, "Recommendation I.363: B-ISDN ATM Adaption Layer (AAL) Specification", Telecommunication Standardization Sector of International Telecommunication Union, Geneva, Switzerland, 1993.
- [30] ITU-T, "Recommendation I.363.X: B-ISDN ATM Adaption Layer (AAL) Specification, Types 1 and 2", Geneva, Switzerland, 1994.
- [31] ITU-T, "Recommendation H.323: Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Non-Guaranteed Quality of Service", Geneva, Switzerland, 1996.
- [32] G. Karlsson, "Layered Error Control Coding for IP Multicast", In *HIPPARCH Workshop*, Uppsala, Sweden, June 1997.
- [33] L. Kieu and K. Ngan, "Cell-Loss Concealment Techniques for Layered Video Codecs in an ATM Network", *IEEE Transactions On Image Processing*, 3(5):666-677, September 1994.

- [34] E. Klemmer, "Subjective Evaluation of Transmission Delay in Telephone Conversations", *Bell Systems Technical Journal*, 46:1141-1147, July 1967.
- [35] V. Kumar, "The MBONE FAQ", Collection of Information about MBONE, January 1997, URL: <http://www.mbone.com/mbone/mbone.faq.html>.
- [36] X. Li, S. Paul, P. Pancha, and M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Error Recovery Schemes", In G. Parulkar, editor, *NOSSDAV 97*, Springer, May 1997.
- [37] J. C. Lin and S. Paul, "RMTP: A Reliable Multicast Transport Protocol", In *INFOCOMM '96*, pp. 1414-1424, San Francisco, CA, March 1996.
- [38] S. Lin and D. J. Costello, *Error Correcting Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1983.
- [39] M. Luby et al., "Practical Loss-Resilient Codes", *InSTOC*, 1997.
- [40] M. Macedonia and M. Zyda, "A Taxonomy for Networked Virtual Environments", *IEEE Multimedia*, 4(1):48-56, January - March 1997.
- [41] A. J. McAuley, "Reliable Broadband Communications Using a Burst Erasure Correcting Code", In *Proceedings of SIGCOMM'90*, pp. 287-306, Philadelphia, PA, September 1990, ACM.
- [42] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast", In *SIGCOMM 96*, pp. 117-130, Stanford, CA, August 1996.
- [43] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast", SSC/1997/001, EPFL, Lausanne, Switzerland, January 1997.
- [44] A. Mehaoua, R. Boutaba, and G. Pujolle, "A Picture Quality Control Framework for MPEG Video over ATM", In *Proc. 5th Workshop on Protocols for High Speed Networks*, pp. 49-59, October 1996.
- [45] J. Nonnenmacher, E. W. Biersack, and D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission", In *Proceedings of ACM SIGCOMM'97*, pp. 289-300, Cannes, France, September 1997.
- [46] K. Obraczka, "Multicast Transport Mechanism: A Survey and Taxonomy", submitted to *IEEE Communications Magazine*, 1997.
- [47] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", In *Proc. IEEE INFOCOM'92*, pp. 915-924, 1992.
- [48] P. Parnes, "RTP extension for Scalable Reliable Multicast; Internet-Draft: draft-parnes-rtp-ext-srm", November 1996.
- [49] S. Paul, K. Sabnani, and D. Kristol, "Multicast Transport Protocols for High Speed Networks", In *International Conference on Network Protocols*, pp. 4-14, Boston, October 1994.
- [50] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharyya, "Reliable Multicast Transport Protocol (RMTP)", *IEEE Journal on Selected Areas in Communications, special issue on Network Support for Multipoint Communication*, 15(3):407 - 421, April 1997.
- [51] V. Paxson, "End-to-End Internet Packet Dynamics", *Computer Communication Review, Proceedings of ACM SIGCOMM'97 Conference*, Cannes, France, September 1997, 27(4):139-152, October 1997.
- [52] S. Pejhan, M. Schwartz, and D. Anastassiou, "Error Control Using Retransmission Schemes in Multicast Transport Protocols for Real-Time Media", *IEEE/ACM Transactions on Networking*, 4(3):413-427, June 1996.

- [53] A. Puri, "Video Coding Using the MPEG-2 Compression Standard", In *Proceedings of SPIE Visual Communications and Image Processing*, Boston, MA, November 1993.
- [54] G. Ramamurthy and D. Raychaudhuri, "Performance of Packet Video with Combined Error Recovery and Concealment", In *IEEE INFOCOM*, volume 2, Boston, MA, April 1995.
- [55] L. Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols", *Computer Communication Review*, 27(2):24-36, April 1997.
- [56] J. Rosenberg and H. Schulzrinne, "An A/V Profile Extension for Generic Forward Error Correction in RTP", Internet-Draft: draft-ietf-avt-fec (Work in Progress), Internet Engineering Task Force, July 1997.
- [57] H. Schulzrinne, S. Casner, R. Frederic, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", Request for Comments RFC 1889, Internet Engineering Task Force, January 1996.
- [58] N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management", In *Proc. IEEE INFOCOM '90*, pp. 124-131, San Francisco, CA, June 1990.
- [59] W. D. Sincoskie, "Broadband ISDN is happening - except it's spelled IP", *IEEE Spectrum*, pp. 32-33, January 1997.
- [60] R. Storn, "Modeling and Optimization of PET-Redundancy Assignment for MPEG Sequences", TR-95-018, International Computer Science Institute, Berkeley, CA, May 1995.
- [61] W. T. Strayer, B. J. Dempsey, and A. C. Weaver, *XTP - THE XPRESS TRANSFER PROTOCOL*, Addison-Wesley Publishing Company, 1992.
- [62] Sun, Zdepski, and Voeten, "Error Concealment for MPEG Video over ATM", In *ISO-IEC/JTC1/SC29/WG11 MPEG92 AVC-308*, July 1992.
- [63] D. Towsley, J. Kurose, and S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *IEEE Journal on Selected Areas in Communications*, 15(3):398-406, 1997.
- [64] D. Walker, F. P. Kelly, and J. Solomon, "Tariffing in the New IP/ATM Environment", *Telecommunications Policy*, 21(97):283-295, May 1997.
- [65] X. Xu, A. Myers, H. Zhang, and R. Yavatkar, "Resilient Multicast Support for Continuous-Media Applications", In G. Parulkar, editor, *NOSSDAV 97*, Springer, May 1997.
- [66] M. Yajnik, J. Kurose, and D. Towsley, "Packet Loss Correlation in the Mbone Multicast Network", In *Proceedings of IEEE Global Internet*, London, UK, November 1996, IEEE.
- [67] R. Yavatkar, J. Griffioen, and M. Sudan, "A reliable Dissemination Protocol for Interactive Collaborative Applications", In *Proceedings of ACM Multimedia*, pp. 333-344, San Francisco, CA, 1995.
- [68] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", *IEEE Network*, pp. 8-18, September 1993.

Biographies

Georg Carle

Georg Carle was born in 1965 in Stuttgart, Germany. Before joining GMD Fokus, Berlin, in October 1997, he has been working as postdoctoral researcher at Institut Eurécom, Sophia Antipolis, France. Georg Carle defended his PhD dissertation (Dr.-Ing.) on reliable group communication services over ATM networks at University of Karlsruhe in December 1996. He worked at the Institute of Telematics at University of Karlsruhe from 1992 to 1996. In 1992, he obtained a degree in Electrical Engineering from the University of Stuttgart. In 1989, he obtained the degree Master of Science in Digital Systems from Brunel University, London, U.K.

His research interests are protocol architecture and charging for integrated services broadband networks, protocol engineering for IP and ATM networks under consideration of performance, Quality of Service and tariffing aspects, and the provision of low-cost high-quality multicast audio-visual services. Georg Carle is a member of IEEE, ACM, and GI/ITG Technical Committee on Communication and Distributed Systems.

Ernst W. Biersack

Ernst Biersack received his M.S. and Ph.D. degrees in Computer Science from the Technische Universität München, Munich, Germany. From March 1989 to February 1992 he was a Member of Technical Staff with the Computer Communications Research District of Bell Communications Research, Morristown, USA. Since March 1992 he has been an Associate Professor in Telecommunications at Institut Eurecom, in Sophia Antipolis, France. His current research is on

- Scalable Reliable Multicast Transfer for very large groups and
- Architectures for Scalable High-Performance Video Servers.

In 1996, he received the outstanding paper award of the IEEE Conference on Multimedia Computing and Systems. Dr. Biersack is a member of the IEEE and ACM. He has been a program committee member for various international conferences and publicity chair for ACM SIGCOMM 97.