HIGH PERFORMANCE GROUP COMMUNICATION SERVICES IN ATM NETWORKS

Georg Carle and Stefan Dresler

Institute of Telematics
University of Karlsruhe, D-76128 Karlsruhe, Germany
E-Mail: {carle,dresler}@telematik.informatik.uni-karlsruhe.de

ABSTRACT

Advanced applications, such as distributed multimedia applications, require efficient communication subsystems providing a variety of services. Existing communication systems face increasing difficulties in fulfilling these requirements. In particular, the efficient provision of reliable group communication services in ATM-Networks remains a major unresolved issue. This paper presents a novel framework for support of multipoint communication in ATM networks. Two adaptation layer protocols are presented that provide reliable multicast services. The first one, called RLMCP (Reliable Lightweight Multicast Protocol), is a simple and efficient adaptation layer protocol for the Service Specific Convergence Sublayer of AAL5. It uses a frame-based ARQ scheme and is suitable for virtual connections with low cell loss rates. The second one, called RMC-AAL (Reliable Multicast ATM Adaptation Layer), features cell-based ARQ and cell-based FEC. A new network element, called the Group Communication Server (GCS), is presented for implementing the adaptation layer protocols in network nodes. It allows for hierarchical multicast error control and support of heterogeneous scenarios. The framework permits to select the combination of error control mechanisms most suitable for the requirements of a specific communication scenario. The functionality of end systems and group communication servers is described, and a basic implementation architecture is presented. Based on this architecture, approximations for the processing delays are presented when the different error control schemes are applied. Finally, the influence of the different error control schemes onto the selection of an appropriate memory management strategy is investigated.

1 INTRODUCTION

Upcoming applications, for example distributed multimedia systems, computer-supported cooperative work (CSCW) applications, and virtual shared memory systems require reliable high performance multipoint communication services. Quality of service (QoS) issues of importance are not only throughput, delay, and delay jitter, but also differences of delay and reliability within the group. A key problem that must be solved to provide a reliable multipoint service is the recovery from cell losses due to congestion in the switches. The probability for cell loss may vary over a wide range, depending on the strategy for usage parameter control (UPC) and call admission control which is applied. It is still an open question how low cell loss rates can be guaranteed for bursty multicast traffic, while using network resources efficiently. Cell losses caused by buffer overflows do not occur randomly distributed, but show a highly correlated characteristic [1]. If a reliable service in ATM networks is based on traditional transport protocols like TCP, severe performance degradations may be observed [2]. For the provision of a reliable multipoint service, the probability for losses increases for a growing number of receivers. However, there are still no convincing concepts for reliable high-performance group communications in ATM-networks. Therefore, the provision of reliable group communications requires the development of efficient protocols and of communication systems that achieve high performance even under conditions with high cell losses.

This paper focuses on design and assessment of error control mechanisms for correction of cell losses for group communication. Section 2 gives an overview on related protocols for error recovery. In section 3, the proposed framework for reliable multicast communication is presented. Section 4 presents performance results of different error control schemes and suggestions for memory organisations that seem most adequate.

2 ENHANCED SERVICES IN ATM NETWORKS

2.1 Unreliable Multipoint Services in ATM Networks

Applications may require the following types of multipoint communication: one-to-many, many-to-one, and many-to-many. ATM networks directly support multicast communication by point-to-multipoint virtual connections. However, many-to-many (i.e., multipeer) communication can be provided only indirectly in ATM networks [3]. The following two techniques are for multipeer communication in ATM networks. In the first alternative, known as VC mesh, each transmitter in a group has to establish a point-to-multipoint connection to all of the other nodes of the group. N point-to-multipoint connections are required for a group with N transmitters. This solution does not scale well for large groups. For large, long-lived groups, numerous virtual channels need to be maintained. Receivers joining or leaving a group require modifications of every multicast tree, causing unwanted delay. In the second alternative, each node establishes a point-to-point connection to a so-called Multicast Server [4], [5]. A point-to-multipoint connection from the Multicast Server to every member of the group is used to transmit messages to the members of the group. This requires N point-to-point connections and one point-to-multipoint connection, improving the scalability significantly.

2.2 Reliable Services in ATM Networks

According to the B-ISDN protocol reference model, mechanisms for error recovery may be integrated into the Service Specific Convergence Sublayer (SSCS) of the adaptation layer for provision of an assured mode service [6]. Up to now, only two SSCS protocols that offer error control mechanisms are specified by ITU. The Service Specific Connection Oriented Protocol (SSCOP) is an SSCS protocol that offers an assured mode service for signalling. The protocol provides end-to-end flow control and recovery of lost or corrupted data frames by selective retransmissions. However, SSCOP does not support assured mode multicast connections. For AAL1, an SSCS with FEC is proposed [6], based on a Reed-Solomon-Code applied to blocks of 128 cells that allows the regeneration of up to four missing cells (long interleaver method). Alternatively, a short interleaver method which uses blocks of 16 cells and which allows the regeneration of a single cell loss is under discussion. Additional FEC schemes for ATM were proposed and investigated in [7] and [8]. However, there remain still a number of open questions concerning the combination of FEC and ARQ in ATM networks.

Transport layer protocols that are suitable for a connectionless network layer, like TCP, TP4 and XTP, allow the provision of reliable services in ATM networks, but are not very well suited to a homogeneous ATM environment. The error control mechanisms of these protocols are very general and not designed for the connection-oriented transmission of ATM cells and AAL frames. These transport protocols need to tolerate packets delivered out of sequence by the network layer. An adaptation layer protocol may benefit from the in-sequence delivery of the ATM-layer service and may use sequence number gaps for error detection. Another problem arises due to the fact that ATM signalling differs conceptually from signalling in traditional transport protocols. ATM is based on out-of band signalling, while conventional transport protocols are based on in-band signalling. If these protocols are to be used in ATM networks, mapping of transport layer connection control to ATM signalling needs to be performed [9].

Existing transport protocols also frequently lack adequate support for reliable multipoint services. XTP offers support for reliable multicasting by a list-based algorithm and the so-called bucket algorithm. However, error control based on the bucket algorithm has significant shortcomings, as shown in [10]. TP++ [11] is an example for a transport protocol that is suitable for ATM networks. It uses retransmissions in combination with FEC for error recovery (type I hybrid ARQ). At present, it is the only transport protocol for high speed networks with FEC. However, it is only capable of unicast communication. Up to now, no protocol that combines ARQ and FEC was presented for multicast communication in ATM networks.

While transmission capacity was growing enormously over the last years, protocol processing and system functions in the transport component turned out to be a performance bottle-neck. High performance communication subsystems, based on parallel protocol processing, and hybrid architectures with hardware components for time-critical operations [12] allow for the provision of a service with high throughput and low latency. For highest performance, complete VLSI implementations of transport subsystems are planned [13]. The performance bottle-neck of the transport component that can be observed for point-to-point communication is even more crucial for reliable multipoint connections. For a growing number of receivers, processing of an increasing number of control packets (known as the implosion problem), and management of a large amount of status information needs to be performed. Therefore, the provision of reliable multipoint services in ATM networks will benefit from the development and implementation of multipoint error control mechanisms tailored for this target environment.

3 EFFICIENT PROVISION OF RELIABLE MULTIPOINT SERVICES

It can not be expected that there exists a single solution for reliable multicasting which is satisfying under all circumstances. Instead, it is important to allow the selection of the error control mechanisms most suited for a specific environment. The framework presented in this section describes how frame-based ARQ, cell-based ARQ, and also cell-based FEC may be integrated into the adaptation layer for the efficient provision of reliable multicast services. Additionally, it describes how these error control mechanisms may be integrated into dedicated servers, and how large groups may be supported by a hierarchy of servers for better scaling properties of reliable group communication. Both performance and implementation complexity have to be considered for a selection of the error control scheme. The selection depends on the cell loss rate, the link length, the path capacity and the available buffers as well as on the application requirements for delay and throughput. In [14] and [15], results from analysis and simulation were presented which allow to identify the most appropriate error control scheme.

The framework considers a number of different group communication services: a fully reliable multicast service with assured delivery to every receiver, and a K-reliable multicast service with assured delivery to at least K receivers of a group. Additionally, a multiplexing service is provided by the group communication server for multiplexing of AAL frames from different transmitters over a single virtual connection.

3.1 Service Specific Convergence Sublayer with frame-based ARQ

The Reliable Lightweight Multicast Protocol (RLMCP) was developed for a simple and efficient provision of reliable multicasting. In ATM end systems, RLMCP can be used as a Service Specific Convergence Sublayer for AAL5. The protocol features error control by frame-based retransmissions, and window flow control. Data frames have a protocol overhead of 10 bytes in the frame header. The AAL5 trailer introduces an additional overhead of 8 bytes. The protocol uses the following data format: the first byte of the header indicates the frame type (data frame, retransmission frame, or acknowledgement). It also contains a flag for immediate acknowledgement request, and a flag indicating the last frame of a burst. The frame header contains a transmitter identifier, and a field for the length of the SSCS-PDU payload. Frames carry frame sequence numbers of 24 bits, which is sufficient for WANs at very high speeds. Frames also carry a sequence number for the lowest sequence number a transmitter is prepared to repeat. Retransmissions may be performed in selective repeat or go-back-N mode. Receivers send acknowledgements periodically, after reception of a frame in which the 'immediate acknowledgement' bit is set, or after detection of a missing frame. Gaps in frame sequence numbers and a receiver timer are used for detection of a lost frame. The 'last frame of stream' bit is for informing the receivers of the end of a burst of frames. This allows the receivers to stop their loss detection timer. Receivers may use cumulative positive acknowledgements, sending a lower window edge, and selective positive or negative acknowledgements, using bit maps with a length of 32 bytes (for a sequence of up to 256 frames). For flow control, acknowledgements contain an upper window edge for the highest sequence number a receiver is prepared to receive.

3.2 Combination of cell-based ARQ and FEC

The Reliable Multicast ATM Adaptation Layer (RMC-AAL) features cell-based ARQ and FEC for an efficient provision of reliable multicast services under conditions of higher or varying cell loss rates, and for applications with strong delay requirements. Error recovery of RMC-AAL is based on three schemes: pure ARQ, type I hybrid ARQ, and pure FEC. RMC-AAL allows to adjust the amount of redundancy depending on the requirements for throughput and delay, the cost of bandwidth, the path capacity, and the number of receivers.

The format of the RMC-AAL data frame header is identical to the frame header of RLMCP explained in the previous section. Like RLMCP, RMC-AAL uses the trailer of AAL5-CPCS, protecting the payload of a frame by the cyclic redundancy check CRC-32. In each cell, RMC-AAL has an additional overhead of one byte (2 bit for cell type CT, and 6 bit cell sequence number). Even for high speed VCs in WANs, no large cell numbering space is required, because every cell is identified by both FSN and CSN. The alternative solution of identifying cells entirely by their cell sequence numbers leads to a significantly higher overhead per cell. For example, the protocol BLINKBLT [16], which also offers cell-based retransmissions, has a per-cell overhead of 4 bytes.

The frame header also contains the discriminator byte with an identifier for the frame type, two flags, and the number of redundancy cells that follow the data frame. Redundancy cells use independent cell sequence numbers. When FEC is used, h redundant cells are generated to protect the information cells of the frame. Encoding and decoding can be based on Reed-Solomon-Codes [17], or on simple XOR-operations and matrix interleaving [7]. Retransmissions may be sent by multicast or by unicast in selective repeat or go-back-N mode. It can be selected if retransmissions are frame-based (by retransmission of data frames) or cell-based (by retransmission of frame fragments). Frame fragments consist of a fragment header cell, followed by a selection of original data cells of this frame. The fragment header cell contains the frame sequence number of the original frame. This field is called ‘Start of Bitmap’. A bitmap is used to indicate which cells of the original frame are retransmitted within the frame fragment. The field ‘Length of Bitmap’ indicates the valid length of the bitmap, and the field ‘Offset of Bitmap’ indicates the cell number of the first bit of the bitmap.

Receivers send acknowledgements periodically, after reception of a frame in which the ‘immediate acknowledgement’ bit is set, and after detection of cell loss. An upper window edge allows for window flow control. Receivers may use cumulative positive acknowledgements of frames by sending the frame sequence number of their lower window edge. Additionally, they may use bit maps with a length of 32 bytes for selective acknowledgement of frames and of individual cells. A frame sequence number and a field for the length of the bit map identifies the position of the bit map within the window.

3.3 Group Communication Server (GCS)

The presented error control mechanisms can be used in ATM end systems, as shown in Figure 1(a). Additionally, the mechanisms can be integrated into intermediate systems called Group Communication Servers (see Figure 1(b)). The deployment of Group Communication Servers with multicast error control mechanisms allows to provide reliable high-performance multipoint services for a wide range of parameters [14]. Further improvements of performance and efficiency may be achieved by using GCSs hierarchically, as shown in Figure 1(c).

The proposed GCS integrates a range of mechanisms that can be grouped into the following tasks:

- Efficient use of network resources by multicast error control within the network;
- Processing support for multicast transmitters;
- Support of heterogeneous and hierarchical multicasting;
- Multiplexing support for groups with multiple transmitters.

For the first task, performing error control in the server permits to increase network efficiency and to reduce delays introduced by retransmissions. Allowing retransmissions originating from the server avoids unnecessary retransmissions over common branches of a multicast tree. The integration of FEC mechanisms into the GCS allows regeneration of lost cells and reinsertion of additional redundancy for adjusting the FEC coding scheme according to the needs of subsequent hops.

For the second task, the GCS releases the burden of a transmitter that deals with a large number of receivers, providing scalability. Instead of communicating with all receivers of a group simultaneously, it is possible for a sender to communicate with a small number of GCSs, where each of them provides reliable delivery to a subset of the receivers. Integrating support for reliable high performance multipoint communication in a server allows better use of such dedicated resources.

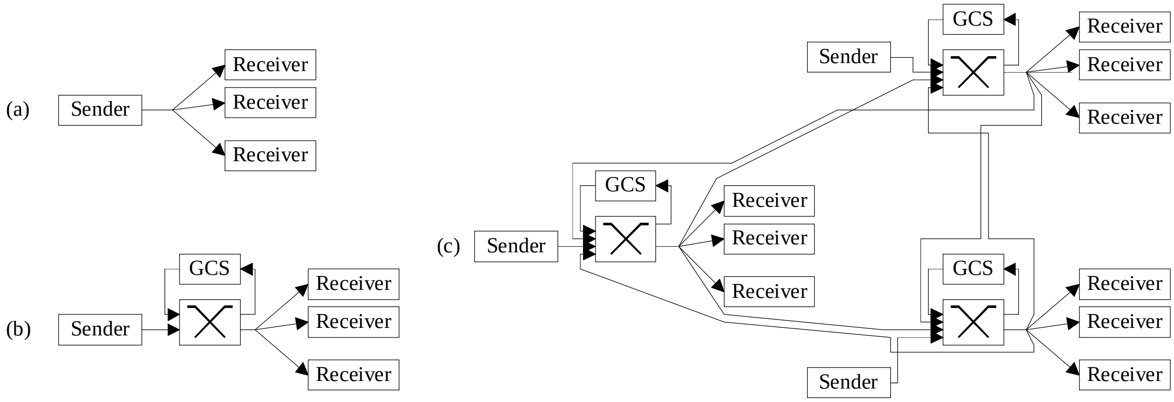

Figure 1 Three example scenarios

For the third task, a GCS may diversify outgoing data streams, allowing conversion of different error schemes and support of different qualities of service for individual servers or subgroups. The group communication server will offer the full range of error control mechanisms provided by the reliable multicast adaptation layer. For end systems, it is not required to implement the full functionality of RMC-AAL. It will be sufficient to have access to a local GCS for participation in a high performance multipoint communication over long distances. The error control mechanisms of individual end systems have only negligible influence onto the overall performance, as simple error control mechanisms are sufficient for communication with a local GCS. If an additional priority field is used in the frame format, the server is able to distinguish packets of different priorities. One example application would be hierarchically encoded video. For information with different priorities, different FEC codes may be applied inside one VC, or specific frames may be suppressed for certain outgoing links. The GCS also allows to support heterogeneous groups that use both RLMCP and RMC-AAL. For this purpose, functions for conversion between different frame formats are provided. One example for this functionality would be frame-based error control within LANs, and cell-based error control for interconnection of the LANs.

For the fourth task, the GCS provides support for multiplexing of frames onto a single point-to-multipoint connection. This allows to reduce the number of required VCs significantly for large groups with many transmitters [4]. Virtual LANs frequently require this multiplexing functionality. If LAN Emulation [5] is used in a local ATM network, a GCS might be incorporated into a LAN Emulation Server (LES) or Broadcast and Unknown Server (BUS), thus making it possible for applications to ensure the reliable delivery of broadcast messages to all peers.

The Group Communication Server may operate in three different modes. In forwarding mode, every frame is processed first by the GCS before being forwarded to the receivers. In case of simple 1:N multicasting, increased performance may be achieved in the bypass mode. In this mode, an ATM switch that supports multicasting will forward data directly to the server and the receivers, reducing the processing load of the server and the overall latency. In both modes, the GCS detects errors earlier than the receivers, and can report an error to the source with lower delay. Both modes also support processing of acknowledgements. For this purpose, every receiver may maintain an individual virtual channel to the GCS. The GCS will either perform the required retransmissions, or will forward retransmission requests to the source. If a window-based flow control scheme is enforced that includes the GCS, retransmissions by the GCS can be guaranteed. However, buffer limitations in the GCS may limit performance in this case. The third mode, called multiplexing mode, is more complex, but allows the provision of a multipeer service with multiplexing of messages from different transmitters over a single virtual connection. In all three modes, the receivers maintain individual unicast VCs to the GCS for acknowledgements. A hierarchy of servers allows for good scaling properties for large groups and high path capacities.

3.4 System Types

The new AAL protocols and the GCS can operate in a number of ways, allowing several combinations. Of the possible alternatives, the following four system types are of importance.

(A) **End system with frame-based error control:** The simplest system type are end systems deploying RLMCP with frame based ARQ (cf. Figure 1(a)). These systems are appropriate for ATM services with very low cell loss rates, and for transmitters which can handle acknowledgement processing without additional support by servers. End systems in such a scenario will subsequently be referred to as type (A) systems.

(B) **End systems with cell-based error control:** For deployment of RMC-AAL with cell-based error control, more complex end systems (called type (B) systems) are required. These systems offer advantages over simpler systems in all cases with higher cell loss rates.

(C) **Servers with frame-based error control:** A GCS deploying frame-based error control is called a type (C) system. This system type, which can be used in the scenarios (b) and (c) of Figure 1, remains relatively simple. It is appropriate in all cases with relatively low cell loss probability.

(D) **Servers with cell-based error control:** For higher cell loss probabilities, an acceptable throughput can be provided by using cell-based error control. This allows to provide reliable multipoint services even to large groups and for relatively unreliable ATM connections, as for example provided by UBR services. GCSs offering cell-based error control will be referred to as type (D) systems. These systems have higher implementation costs than the other three system types.

3.5 Implementation Architecture

Figure 2 shows a functional architecture for the GCS. The functionality is distributed over a number of modules. The modules that perform frame-based ARQ functionality are shaded in Figure 2. Modules that are required only by cell-based ARQ have a lighter background. Modules that have to perform additional functionality in FEC mode are marked by 'FEC'. The *Frame Manager* is responsible for storing and retrieving cells and frames. The *ACK Manager* processes acknowledgements and manages status information of individual receivers and the group. The *Send Manger* (two blocks on the lower left) schedules between ordinary transmissions and retransmissions.

The modules are specified as individual finite state machines with local status information. The modules exchange messages which contain control information together with either a pointer to the payload of a cell, or the payload itself. The specification allows an implementation of the GCS using

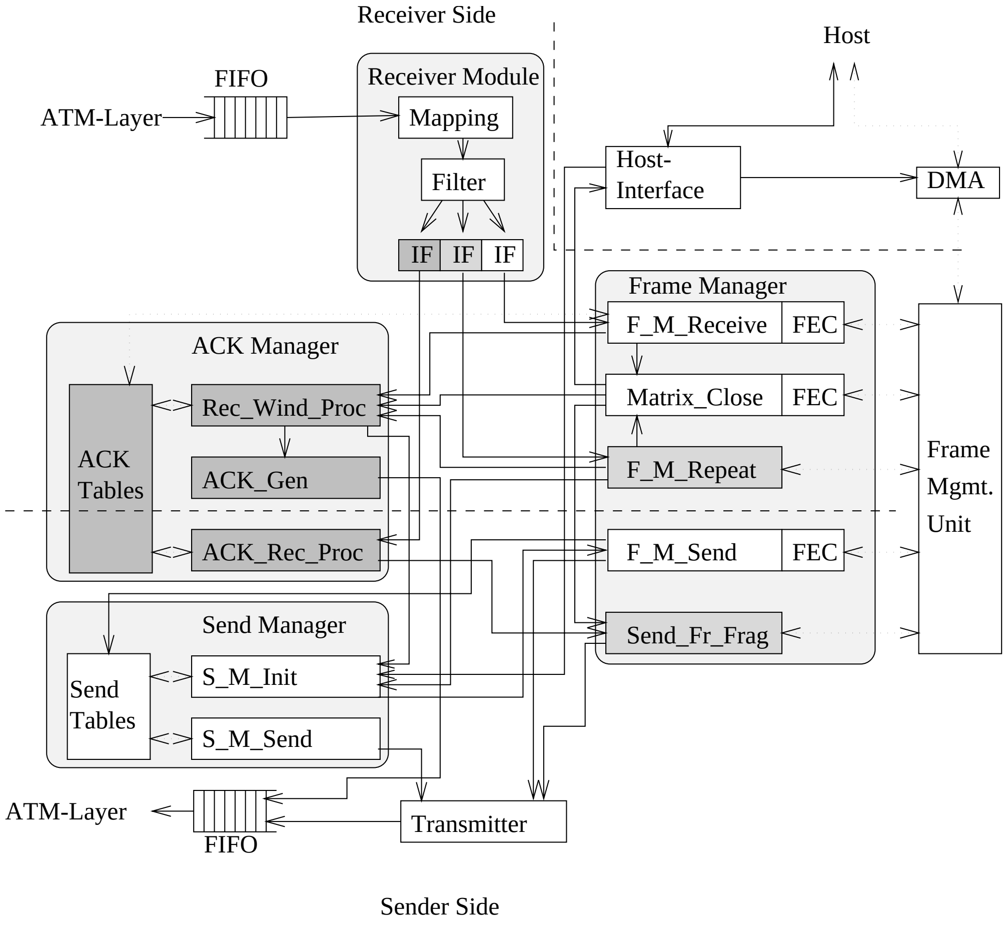

Figure 2 Architecture for the Group Communication Server

a single processor, or a parallel architecture. Implementing the GCS in software on a single processor, each module may be mapped onto an individual thread. A parallel implementation may use general purpose processors, or specialised, microprogrammable units [18]. For demultiplexing, a content addressable memory (CAM) is suitable for mapping of the large VPI/VCI address space onto smaller internal identifiers. Performance enhancements are possible by dedicated hardware support for filtering, and for processing of the bit maps. Additional hardware components are proposed for CRC, FEC, buffer management, and list and timer management [19].

The functionality of GCSs is not necessarily restricted to pure servers inside the network. Instead, it is possible to combine an ATM end system with the functionality of a GCS. The additional functionality of end systems for the exchange of AAL SDUs with higher layers is shown in Figure 2 in the upper right corner. A host interface controller coordinates the communication between network adaptor and host memory.

4 ASSESSMENT OF PERFORMANCE AND IMPLEMENTATION

4.1 Assessment of Error Control Schemes

In order to study processing delay and implementation complexity, the implementation of a GCS on a network adaptor with the following properties was investigated: An embedded controller (32 bit RISC processor with an average performance of 100 MIPS), hardware support for segmentation and reassembly, hardware for CRC32, and hardware for FEC processing were assumed. Based on a specification of the modules in assembly-level pseudo-code, the number of instruction cycles necessary to perform the required functionality was determined for each module of the GCS. Using the numbers, the processing delay on a processor with 100 MIPS was evaluated.

In Figure 3, the processing delays are compared to the cell interarrival time of 2.74 µs for an ATM link of 155 Mbit/s. The figure shows the processing delays for GCS configurations of increasing complexity from left to right. The configuration with the lowest processing delays is a GCS with frame-based ARQ. The figure also shows the additional processing delays of a GCS with cell-based ARQ, and with additional FEC. The right edges of the bars indicate the processing delays in each component if cell-based ARQ, FEC, and multiplexing are performed. The figure illustrates several

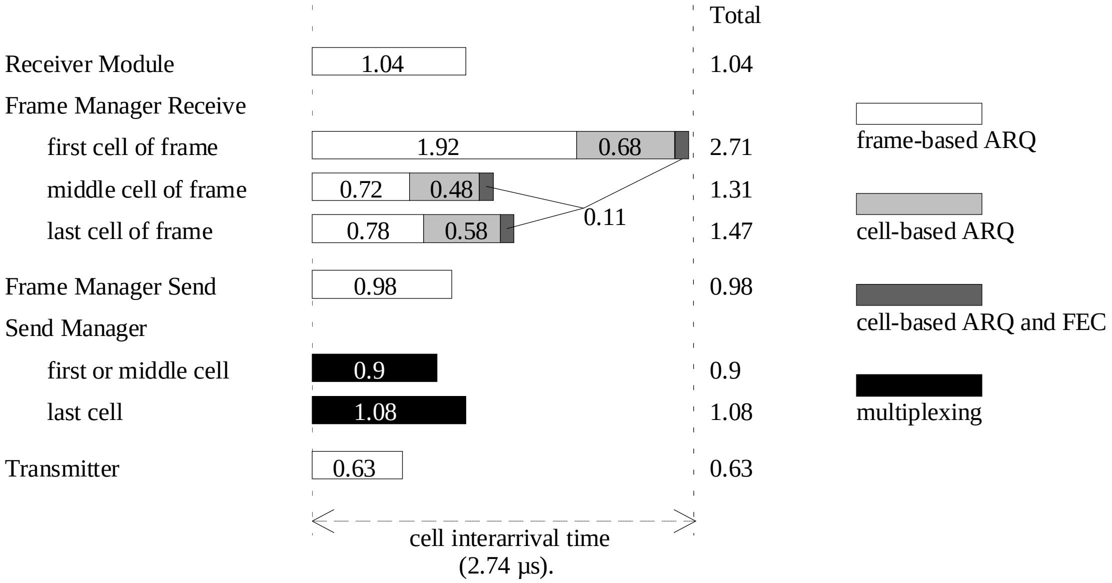

Figure 3 Processing delays (in µs) for data cells in different error control schemes

properties. First, the receiver module and the outgoing transmitter have a constant delay independent of any ARQ or FEC processing. Second, frame-level multiplexing does not take much time in any component other than the Send Manager which does the scheduling of the frames. The GCS does not provide a copy function for multicasting. This copy function for multicasting is provided by an ATM switch. Third, and most important, it can be seen that the delay is dominated by the Frame Manager Receive whenever the first cell of a frame of a connection with cell-based ARQ and FEC is processed. In this module, the processing delay of the first cell of a frame is 2.71 µs when cell-based ARQ is selected in combination with FEC. The processing delay of a cell in the middle of a frame is 1.31 µs, while the processing delay of the last cell of a frame is 1.47 µs. Thus, this component is the first candidate for optimisation, and for the deployment of hardware components for processing support.

A single processor of 100 MIPS leads to a processing bottle-neck at high loads, as the overall delay for processing of a cell by the GCS (summarising the processing times of all modules) is larger than the cell interarrival time. A set of three processors with support of a CAM and dedicated hardware to perform the filtering and the construction of outgoing cells allows for maximum load with an ATM link of 155 Mbit/s even for frames consisting of a single cell.

Not shown in the diagram are queueing delays of cells that have to wait because frames of other senders in the same group have to be sent first. Furthermore, operations caused by the processing of acknowledgements in a GCS or in a sending host are not contained in the diagram. The latter operations heavily depend on the number of receivers that acknowledge the reception of frames or cells, and on the acknowledgement strategy (e.g., NAKs might be sent as soon as possible, whereas ACKs are grouped).

4.2 Memory Management

This section discusses how the error control schemes to be supported influence the selection of the memory management scheme.

Requirements specific to error control mechanisms

Optimisation of memory management needs to consider the operations for storage and retrieval of cells and frames required by a specific error control mechanism:

- In a GCS that does not deploy error control mechanisms, only functions are needed that linearly store, retrieve, and release cells of their respective frame. Without retransmission functionality, only a small number of frames must be stored. No random access to frames or individual cells is required.
- If frame based ARQ is deployed, a larger number of frames has to be managed, depending on path capacity and packet sizes. Additionally, selective retransmissions of frames require random access to individual frames.
- Cell-based FEC requires random access to individual cells.
- For cell-based ARQ with or without FEC, the requirements of frame based ARQ and of FEC are combined. This requires random access to a large number of frames, and to individual cells.

Architecture for Memory Organization

An assessment of the implementation costs associated with the different memory management strategies allows to identify the strategy most appropriate for the selected error control mechanisms. In this section, some example architectures for memory organization are described and compared, resulting in recommendations for selection of the memory management strategy.

The window mechanism for selective repeat ARQ requires management of frame status information. In the specification of the GCS, memory for 2^n frames is allocated as a ring buffer. This allows random access to any frame identified by its frame sequence number. One bit per frame and per receiver is used to indicate whether the specific receiver has already acknowledged the frame. In a typical configuration of the GCS, 256/8=32 bytes are needed for 256 receivers. With an additional overhead of 16 bytes per frame for window management and status information, this results in a total overhead of 48 bytes per frame.

Additional status information is required for cell-level memory management. Buffering strategies found in literature frequently propose the usage of local memory on the host adaptor, or of dual-port memory [20]. In [21], a direct mapping mechanism of two virtual address spaces is described. The mechanism does not need dedicated buffers on the network adaptor, because it directly accesses host memory. Another memory management scheme for an ATM network interface which eliminates copying can be found in [22]. None of the systems cited above and further below is tailored towards correcting cell or frame losses. For example, the management strategy of [23] is explicitly targeted for very low cell loss and cell corruption.

Figure 4 presents six different cell-level memory management strategies which can be used for implementation of the GCS, and of ATM end systems employing RLMCP and RMC-AAL. Containers are defined as memory blocks accommodating one or more cell payloads, together with valid bits for reassembly status information. The containers may be used to store redundancy cells and the AAL5 trailer in addition to the AAL SDU in order to facilitate retransmissions.

(a) Since cells of an AAL frame arrive in order, in many cases the simplest way to store cells is a linked list [24]. This is illustrated in Figure 4(a). A variation of this strategy is presented in [25], where the elements of the list do not contain the cell payload itself, but pointers to the payload.
(b) A different strategy is the allocation of a single memory block for all cells of a frame (Figure 4(b)). This also allows to keep the valid bits for all cells of the frame in a single valid bit map, which can be accessed without searching a list. However, in this strategy memory will be used inefficiently if variable size AAL frames are used together with fixed size containers.
(c) Figure 4(c) illustrates a hybrid approach, which uses a row of pointers each pointing to a single cell container (or NIL, resp.).
(d) Figure 4(d) shows another approach which uses a list of containers, each allowing to store a number of cell payloads together with a valid bit map [26]. The size of the containers should be chosen neither too big (in order not to waste memory for short frames) nor too small (in order to avoid time consuming search operations if cells at random places have to be accessed).
(e) Part (e) of the figure shows what will turn out to be one of the most promising approaches. A row of pointers allows immediate access to containers that allow the storage of e.g. 32 cell payloads each.
(f) In end systems, host memory can be used to store AAL SDUs. Figure 4(f) shows a memory management strategy holding control data like the valid bit maps in local memory, while storing the actual payload in host memory.

Table 1 gives the sizes of the data structures associated with the different strategies. For the strategy which involves host memory, column (f) of Table 1 presents only the requirements of local memory. For calculation of the amount of memory needed to store a single frame in each of the memory management strategies (see Table 2), the following frame sizes are used:

- Frame size 1 with a length of 2 cells (typical for short messages);
- Frame size 2 with a length of 196 cells (suitable for the IP MTU size of 9,180 bytes);
- Frame size 3 with a length of 1,366 cells (the maximum size of AAL5 frames).

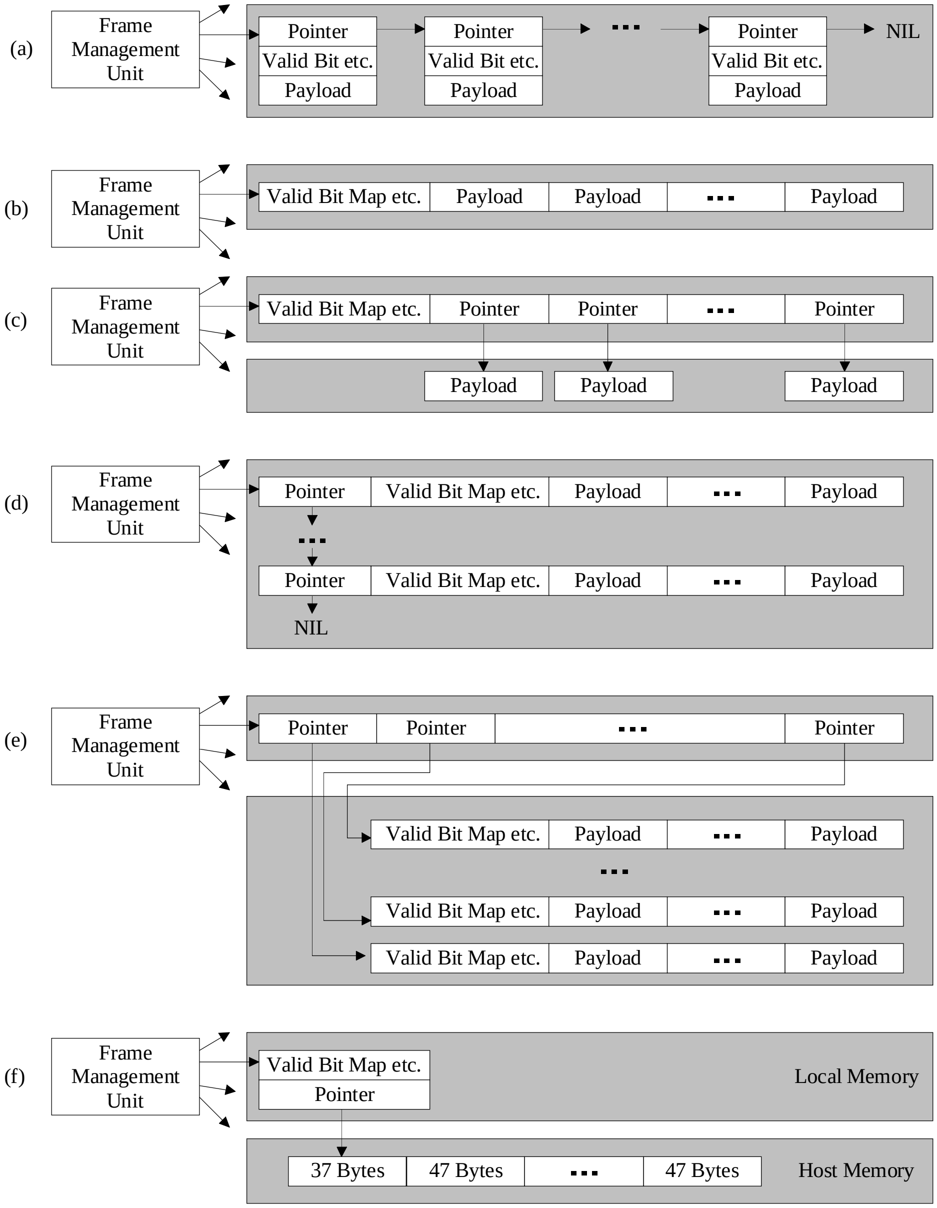

Figure 4 Management of cells of frames in GCSs and end systems

Table 2 also gives an evaluation of the strategies in terms of access delay. For different system types, different memory organisations are appropriate. A recommendation for a memory management strategy for GCSs and end systems has to take into account that for certain implementations, it may not be feasible to alter the strategy at run time depending on the current needs. In these cases, a strategy must be selected which is a good compromise for all situations that might occur. The assessment of suitable memory management strategies can be summarised as follows:

- If fixed size frames are used, strategy (b) with fixed size containers is the most advantageous approach. This strategy supports all error control schemes equally well.
- For frames of variable size, and systems that need to support all four error control schemes, strategy (e) with containers of, e.g., 32 cells is most appropriate.
- For end systems, storage of the actual payload in local memory can be avoided by choosing strategy (f). Therefore, this approach is recommended for end systems for all error control schemes.

(a) Linked list		
	per frame:	0
	per cell:	52 bytes (48 bytes for payload, 4 bytes for pointer and valid bit)
(b) Single container for 1366 cells		
	per frame:	65,740 bytes (⌈1,366/8⌉ bytes + 1 byte padding + 1,366*48 bytes)
	per cell:	0
(c) Row of pointers		
	per frame:	5,636 bytes (⌈1,366/8 bytes⌉ + 1 byte padding + 1,366*4 bytes)
	per cell:	48 bytes
(d) List of containers, 32 cells per container		
	per frame:	0
	per container	1,544 bytes (4 bytes BM + 4 bytes pointer + 32*48 bytes)
(e) Row of pointers to containers, 32 cells per container		
	per frame:	172 bytes (⌈1,366/32⌉*4 bytes for pointers)
	per container	1,540 bytes (32*48 bytes payload + 4 bytes BM)
(f) Host memory		
	per frame:	180 bytes (8 bytes (pointer) + ⌈1,366/8⌉ bytes + 1 byte padding)
	per cell:	47 bytes

BM = Bit Map; ⌈...⌉ indicates rounding up to the next integer

Table 1 Local memory needed for different strategies

	(a)	(b)	(c)	(d)	(e)	(f)
Frame size	Memory usage (bytes)					
Size 1 (2 cells)	104	65,740	5,732	1,544	1,712	180 (local)
Size 2 (196 cells)	10,192	65,740	15,044	10,808	10,952	180 (local)
Size 3 (1366 cells)	71,032	65,740	71,204	66,392	66,392	180 (local)
Suitability	+	-- (+ for fixed size frames)	-	+	+	++
Access delay						
System type (A)	+	++ (Bytes directly accessible)	+ (Access in constant time)	o	+	++ (zero copy, direct access)
System type (B)	-- (Search inefficient)	++ (Bytes directly accessible)	+ (Access in constant time)	- (Slow search)	+ (Access in constant time)	++ (zero copy, direct access)
System type (C)	+	++ (Bytes directly accessible)	+ (Access in constant time)	o	+	n.a.
System type (D)	-- (Search inefficient)	++ (Bytes directly accessible)	+ (Access in constant time)	- (Slow search)	+ Access in constant time	n.a.
Recommendation	GCS, only frame-based ARQ	GCS, all error control schems, only fixed size frames	none	none	GCS, all error control schems	End systems, all error control schems

Rating: ++ very good; + good; o average; - poor; -- inacceptable; n.a. not applicable

Table 2 Evalution of memory management strategies

5 CONCLUSION

A new framework is presented which has the potential to fulfill many requirements. For small groups and low cell loss rates, a frame-based end-to-end error control is most appropriate. In this case, RLMCP or another lightweight protocol for reliable multicast can be used as SSCS for AAL5. In case of significant cell loss, large group sizes, and higher path capacities, the new adaptation layer type called the Reliable Multicast ATM Adaptation Layer (RMC-AAL) is proposed. It is the first AAL protocol offering cell-based ARQ and FEC for reliable multicasting. It has only one byte protocol overhead per cell, plus 10 bytes overhead for the frame header. Cell-based retransmission as well as FEC allow high-performance reliable multicasting even for significant cell loss rates. For better scalability and support of heterogeneous scenarios, the deployment of a new network element called the Group Communication Server (GCS) is proposed. It allows an hierarchical approach for multicast error control and the combination of different error control schemes. Investigation of the processing delay demonstrated the feasibility of the proposed error control schemes even for very high speeds. It also revealed that cell-based error control schemes contribute little to the processing load for error-free transmissions. Different strategies to store cells of AAL frames were investigated and compared. Recommendations resulted according to the characteristics of the systems involved.

Acknowledgement

The authors would like to thank Martina Zitterbart and the members of the High Performance Networking Group for valuable discussions. Georg Carle also acknowledges the support by the Graduiertenkolleg „Controllability of Complex Systems“ (DFG Vo287/5-2).

References

[1] Ohta, H., Kitami, T.: „A Cell Loss Recovery Method Using FEC in ATM Networks,“ IEEE Journal on Selected Areas in Communications, Vol. 9, No. 9, December 1991, pp. 1471-1483

[2] Romanov, A.: „Some Results on the Performance of TCP over ATM,“ Second IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems HPCS’93, Williamsburg, Virginia, U.S.A., September 1993

[3] Bubenik, R.; Gaddis, M.; DeHart, J.: „Communicating with virtual paths and virtual channels,“ Proceedings of the Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies INFOCOM’92, pp. 1035-1042, Florence, Italy, May 1992

[4] Wei, L.; Liaw, F.; Estrin, D.; Romanow, A.; Lyon, T.: „Analysis of a Resequencer Model for Multicast over ATM Networks“, Third International Workshop on Network and Operating Systems Support for Digital Audio and Video, San Diego, U.S.A., Nov. 1992

[5] ATM Forum: „LAN Emulation Over ATM: Draft Specification“, LAN Emulation Subworking Group, ATM Forum Technical Commitee, August 1995

[6] ITU-TS Draft Recommendation I.363: „BISDN ATM Adaptation Layer (AAL) Specification,“ Geneva, 1993

[7] Shacham, N.; McKenny, P.: „Packet recovery in high-speed networks using coding,“ in Proceedings of IEEE INFOCOM '90, San Francisco, CA, pp. 124-131, June 1990

[8] Biersack, E. W.: „Performance Evaluation of Forward Error Correction in an ATM Environment,“ IEEE Journal on Selected Areas in Communication, Volume 11, Number 4, pp. 631-640, May 1993

[9] Kure, O.; Sorteberg, I.: „XTP over ATM,“ Computer Networks and ISDN Systems, 26 (1993). pp. 253-262

[10] Santoso, H.; Fdida, S.: „Transport Layer Multicast: An Enhancement for XTP Bucket Error Control,“ in: Danthine, A.; Spaniol, O. (eds.): High Performance Networking, IV, IFIP, North-Holland, 1993

[11] Feldmaier, D.: „An Overview of the TP++ Transport Protocol Project,“ Chapter 8 in ‘High Performance Networks - Frontiers and Experience,’ Ahmed Tantawy (Ed.), Kluwer Academic Publishers, 1993

[12] Braun, T.; Zitterbart, M.: „Parallel Transport System Design,“ 4th IFIP Conference on High Performance Networking HPN’92, Liège, Belgium, December 1992

[13] Braun, T.; Schiller, J.; Zitterbart, M.: „A modular VLSI implementation architecture for communication subsystems,“ Neufeld, G.; Ito, M. (Ed.), Protocols for High Speed Networks IV, Chapman & Hall, 1995

[14] Carle, G.: „Adaptation Layer and Group Communication Server for Reliable Multipoint Services in ATM Networks,“ in R. Steinmetz (Ed.): Advanced Teleservices and High Speed Communication Architectures; Second International Workshop, IWACA '94, Heidelberg, September 26-28, 1994; Lecture Notes in Computer Science 868, Springer-Verlag, pp. 124 - 138

[15] Carle, G.; Zitterbart, M.: „ATM Adaptation Layer and Group Communication Servers for High-Performance Multipoint Services“, 7th IEEE Workshop on Local and Metropolitan Area Networks, pp. 98-106, March 26-29, 1995, Duck Key, Marathon, Florida, U.S.A.

[16] Goldstein, F.: „Compatibility of BLINKBLT with the ATM Adaptation Layer,“ ANSI Technical Subcommitee T1S1.5/90-009, Raleigh, NC, USA, Feb. 1990

[17] McAuley, A.: „Reliable Broadband Communication Using a Burst Erasure Correcting Code,“ ACM SIGCOMM '90, Philadelphia, PA, U.S.A., Sep. 1990

[18] Carle, G.; Schiller, J.; Schmidt, C.: „Support for High-Performance Multipoint Multimedia Services,“ in D. Hutchison, A. Danthine, H. Leopold, G. Colson (Eds.): Multimedia Transport and Teleservices; International COST 237 Workshop, Vienna, Austria, November 13 - 15, 1994, Lecture Notes in Computer Science 868, Springer-Verlag, pp. 219 - 240

[19] Carle, G.; Schiller, J.: „Modular VLSI Implementation Architecture for the Scalable Provision of High-Performance Multimedia Multipoint Services,“ Third IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS’95), Mystic, Connecticut, August 23-25, 1995

[20] Davie, B.: „A Host-Network Interface Architecture for ATM,“ Proceedings of ACM SIGCOMM '91, September 4-6, 1991, Zürich, Switzerland, pp. 197-205

[21] Blumrich, M.A.; Dubnicki, C.; Felten, E. W.; Li, K.; Mesarina, M. R.: „Virtual-Memory-Mapped Network Interfaces“, IEEE Micro, February 1995, pp. 21 - 28

[22] Minnich, R.; Burns, D.; Hady, F.: ‘‘The Memory-Integrated Network Interface,’’ IEEE Micro, February 1995, pp. 11 - 20

[23] Traw, C.: „A Host Interface Architecture and Implementation for ATM Networks,“ Master Thesis, University of Pennsylvania, Pennsylvania, USA, May 1992

[24] Chao, H.; Smith, D.: „Design of Virtual Channel Queue in an ATM Broadband Terminal Adaptor,“ Proceedings of IEEE INFOCOM'92, pp. 1035 - 1042, Florence, Italy, May 1992, pp. 294-302

[25] Traw, C.; Smith, J.: „A High-Performance Host Interface for ATM Networks,“ Proceedings of ACM SIGCOMM '91, September 4-6, 1991, Zürich, Switzerland, pp. 206- 214

[26] Hill, M.; Cantoni, A.; Moors, T.: „Design and Implementation of an ATM Receiver,“ 3rd IFIP WG6.1/6.4 Workshop on Protocols for High-Speed Networks, May 13-15, 1992, Stockholm-Kista, Sweden