5th IEEE Workshop on Metropolitan Area Networks, May 10-13, 1992, Taormina, Italy

A Parallel High Performance Transport System for Metropolitan Area Networks

Torsten Braun

Institute of Telematics, University of Karlsruhe

P. O. Box 6980, W7500 Karlsruhe, Germany

Phone: +49 721 608-3982, Fax: +49 721 60 60 97

Email: braun@telematik.informatik.uni-karlsruhe.de

Abstract

Metropolitan Area Networks (MANs) provide bandwidths in the range of several hundreds of MBit/s up to 1 GBit/s. Applications, e.g., in the multi-media area, could take advantage of the network performance, but, currently, communication systems are hardly able to support high bandwidth. Even processing of the transport oriented layers is not fast enough. To overcome the transport oriented protocol processing bottleneck, this paper presents a transport system called PATROCLOS (parallel transport system for cell based networks), which is based on the utilisation of parallel architectures. Because conventional communication protocols do not support parallelism very efficiently, the transport system is especially designed to be appropriate for an implementation on a hybrid multiprocessor architecture. PATROCLOS is mainly designed for networks with small transmission units like ATM or DQDB and provides a reliable transport service. Adaptation layer processing is integrated into the transport system. Therefore, below the transport system only media access units are required.

1. Introduction

Due to advanced light wave based transmission systems, the bandwidth of modern high speed networks has evolved faster than the performance of communication controllers, which have to process communication protocols of end systems and transit systems. On the other hand, there are also several applications, which require bandwidths up to several hundreds of MBit/s, e.g., in the area of medicine. Furthermore, protocols used inside transit systems, where traffic of several subnetworks may be concentrated, must be very simple to allow complete implementation in hardware. Functionality will be limited to switching, routing or relaying functions, and simple quality of service support. Complex functions have to move from the transit systems to the end systems.

Nowadays, even in high speed networks, protocols are used that were developed decades ago for networks with up to five orders of magnitude lower bandwidths and extremely higher error rates. Several protocol mechanisms are not appropriate for faster networks and may even hinder from the use of the network performance. Other protocol functions can be simplified in modern high speed networks to allow more efficient protocol implementations.

Moreover, the protocol architecture itself restrains efficient implementations in VLSI or on multiprocessor architectures. Communication protocols and communication system architectures have to be developed hand in hand taking into consideration the requirements of applications and networks. Therefore, the design of PATROCLOS is especially influenced by MAN characteristics, e.g., increased bandwidth-delay-product, new switching techniques, and media access methods with small transfer units, lower error rates, and a large number of connection and message identifiers.

2. Design Issues

2.1. Influence of High Speed Network Characteristics

Several characteristics of high speed networks influence the design of communication system architectures and protocols significantly.

High Bandwidth Delay Product

MANs are designed to be appropriate for large extents in the area of hundreds of kilometres. Their bandwidths are developing towards 1 GBit/s.

In this case, the bandwidth-delay-product, which is a measure for the amount of data being on the transmission line between sender and receiver, affects several protocol mechanisms such as connection establishment (implicit establishment vs. handshakes), congestion control (pre-emptive vs. reactive control), flow control (rate vs. window based), and error control (forward error correction vs. retransmission (go-back-n or selective)).

Moreover, the sender and the receiver have very different views of connection states, if the offered bandwidth is used completely in those networks. Therefore, it does not seem to be useful to exchange connection state information, such as sequence numbers for sent user data packets, acknowledgements, flow control parameters etc., for each packet. Connection state information can also be exchanged after requests like in XTP [SAW90] or periodically like in SNR [SNR90].

Switching and Media Access Methods

Based on the advanced transmission technology, networks with new switching techniques, e.g., ATM, and media access methods, e.g., DQDB's pre-arbitrated and queued-arbitrated access, have been developed to support different types of applications on the network level. Management functions have to map quality-of-service required at the transport level to access parameters and services of the network.

A transport system may have to support strong delay requirements, e.g., delay jitter. Bandwidth reservation and appropriate network access mechanisms support jitter control functions of the transport system, which have to be placed close to the transport service interface, because jitter can occur if transport system processing delay is not constant.

ATM and DQDB networks use small transfer units (cells) with a fixed size of 53 bytes to support very different bandwidth and timing requirements. The overhead for small data units is reduced. Because protocol processing of many small transfer units requires very high performance, the protocol functions on cell level have to be very simple to allow implementation in VLSI and should be limited to segmentation and reassembly (SAR) functions. Complex protocol functions will be located rather at end systems than at transit systems.

Network Errors

Congestion avoidance and recovery mechanisms become very important, because the probability of cell losses will be higher than of bit errors. Error recognition can be performed on cell level by checksums and sequence numbers. Assuming low error rates, error recovery performed on cell level has no significant advantages with regard to bandwidth overhead or delay compared to error recovery on frame level. For that reason, error recovery can be performed on a higher level, the so-called frame level [LAI91].

Connection and Message Identifiers

Multiplexing inside the transport system can be avoided, if the network supports multiplexing by a large number of connection and message identifiers, e.g., VPI (virtual path identifier), VCI (virtual channel identifier), MID (multiplex or message identifier). Virtual paths can bundle several message streams, MIDs allow multiplexing of data flows inside a connection.

2.2. Architectural Issues

The high performance transport system presented in this paper is intended for cell based MANs such as ATM or DQDB with complete functionality of layers between media access and transport layer. High performance is achieved by

- reduction of layering overhead and
- support of parallel protocol processing.

2.2.1. Reduction of the Layering Overhead

Highly layered systems are not very efficient, because of the so-called layering overhead.

- The large number of layer interfaces reduce efficiency, because the delivery of interface data units is costly.
- Protocol functions like connection management, checksum calculation, segmentation and reassembly etc. are often performed more than once in different layers.

The transport system performs functions that are usually located at the layers 2 - 4 according to the OSI reference model. For integration of ATM or DQDB networks into OSI protocol stacks, so called adaptation or convergence layers (AAL: ATM adaptation layer, DQDB-CF: DQDB convergence functions) are necessary. The layering overhead is further increased.

PATROCLOS combines all protocol functions above the media access and below the transport service interface. It is an adaptation layer, which offers a reliable transport service in contrast to the MAC service offered by AAL type 4, or DQDB-CF.

2.2.2. Parallel Protocol Processing

Implementations of standardised protocols like OSI and TCP/IP on parallel architectures lead to performance gains, but mainly pipelined parallelism can be achieved [ZIT91] because of the subdivision of OSI communication systems into protocol layers and sublayers. Furthermore, the various protocols, which are based on single finite state machines, are not designed to support parallelism. SNR and XTP, which are the most popular protocols based on parallel designs, have influenced the design of the transport system presented in this paper.

SNR [SNR90] is an end-to-end protocol for high speed networks. The protocol is decomposed into seven processes. Connection state parameters are exchanged periodically by control packets independent of user data transfer.

XTP [SAW90, CHE91] comprises functions of the transport and the network layer. The XTP specification is based on several finite state machines (FSMs), which can run in parallel. An implementation of XTP on a multiprocessor environment based on parallel FSMs [BRZ91] is able to process sender and receiver part in parallel without significant interactions. Also, processing of information and control packets can be performed partially in parallel.

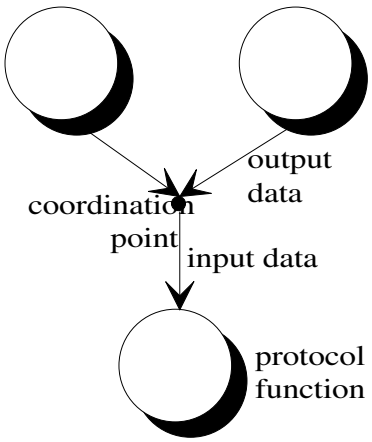

Figure 1 : Coordination Point

The main reason for restricting parallelisation of conventional communication system architectures are the so called ***coordination points***. At a coordination point several functions have to be finished before other functions can proceed. Coordination points are caused by:

- Highly layered systems

  At the layer interfaces of highly layered systems, all functions, which are necessary for generating a SDU, have to be performed before a SDU is delivered from one layer to the other.

- Use of common resources

  Many control functions use common resources such as common control packets, common subfunctions, and common data bases for connection state and routing information. With the use of common resources, consistency problems and access conflicts restrict parallelism. The demand to avoid multiplexing of common resources [FEL90] has effects on protocol design: the control functions of a protocol should operate on independent state variables and data, e.g., by avoiding of the use of common control packets by different control functions.

- Integration of control and data transfer functions

  A higher degree of parallelism can be achieved by decoupling transfer control and data transfer processing. Protocol functions can be classified into data transfer functions, which process SDUs of/for a higher layer, and transfer control functions, which control and manage data transfer, e.g., by connection management, error control etc. [CLT90].

  Often, control functions are closely coupled with data transfer, e.g., if flow control parameters are exchanged for each data packet. Control packet exchange can be decoupled from data transfer by explicit requests for control packets or by periodic exchange. In networks with high bandwidth-delay-product and different connection state views of sender and receiver, it does not seem to be necessary to perform transfer control functions like acknowledgement for each packet sent or received.

- Data Dependencies

  Data dependencies among different protocol functions occur when a protocol function must wait until its input data have been processed by former protocol functions, i.e., when input data of a protocol function are changed by other functions. Data changing functions may change the contents or the size of the data. Functions like segmentation/reassembly or (de)multiplexing may be eliminated or reduced. Fixed header formats avoid complex generation and parsing functions. Trailers are only useful for functions like checksum calculation, which are performed "on the fly".

3. Architecture and Implementation of PATROCLOS

Based on the impacts of high speed networks on transport system design and on the parallel design issues, a model for a parallel transport system for cell based networks is developed. This transport system is called PATROCLOS, which is mainly designed to allow a high degree of parallelism.

3.1. Functionality

PATROCLOS provides a reliable transport service and comprises functions that usually are performed between media access and the transport service interface. The transport system operates only on frames and cells. Frames are the logical units for error recovery, flow control, and all other control functions. The size of a frame has to be chosen regarding efficiency requirements of all control functions. Segmentation and reassembly (SAR) as well as other size changing functions like multiplexing are performed at the interfaces of the transport system only.

Segmentation and reassembly on cell level have to be implemented very efficiently, e.g., in gigabit networks more than 2 millions of cells have to be processed per second. Hardware support will be required, and hence, the data formats must be very simple. SAR functions similar to those of the AAL-SAR sublayer are necessary at the interface between the transport system and the network. A higher degree of parallelism can be achieved, if (de)multiplexing would be performed below SAR functions. Other size changing functions, e.g., concatenation and separating, have to be avoided inside the transport system or should be performed also close to the interfaces.

The transport system's functions layers are classified into

- transfer control and
- data transfer functions.

Data transfer functions are divided into

- data access and
- data control functions.

Data access functions process (manipulate or analyse) data units and can be divided into the subclasses *data manipulation functions* and *data analysis functions*. Most of the data manipulation functions, e.g., PDU composition, are located at the sender part of the transport system. Data analysis functions such as duplicate recognition and packet format analysis are located mainly at the receiver side.

Data control functions do not manipulate or analyse the data units. They decide, the actions to be taken on the data units, e.g., an incoming data unit can be discarded, forwarded to another network, or delivered to the above layer. Data control functions can also be divided into two subclasses called *send control functions* (sending of control and user data packets, quality-of-service support) and *receive control functions* (forwarding, PDU discarding, resequencing).

Transfer control functions (connection establishment and termination, connection data management, flow control, congestion control, rate control, acknowledgement, synchronisation message exchange, route management) do not operate on user data as the classes described above. They control the connection and its user data flow and exchange control information for control and management purposes.

Figure 2 presents the logical architecture of PATROCLOS. On sending side, TSDUs are received via the transport service interface and may be segmented. The PDUs (frames) of the transport system are generated by the data manipulation functions (user data) and the transfer control functions (control information). Control data frames have sizes of only one or two cells. The send control functions determine the next frame to be delivered via the segmentation function to the network considering connection states and quality-of-service parameters.

On receiving side, arriving cells are reassembled to frames, which are delivered to the data analysis functions or to the transfer control functions. The data analysis functions check for errors such as duplicates, lifetime errors etc. The results are delivered to the transfer control functions and to the receive control functions. The transfer control functions update connection data and perform functions like acknowledgement or flow control dependent on the results of the analysis. The receive control functions decide the actions to take on the frame, e.g., forwarding, discarding, or delivering to the upper layer considering delay jitter requirements. If necessary, user data frames are reassembled before delivery to the transport system user.

3.2. Decomposition into Modular FSMs

PATROCLOS consists of FSMs for all transfer control functions and FSMs for outgoing and incoming user data. There are FSMs for

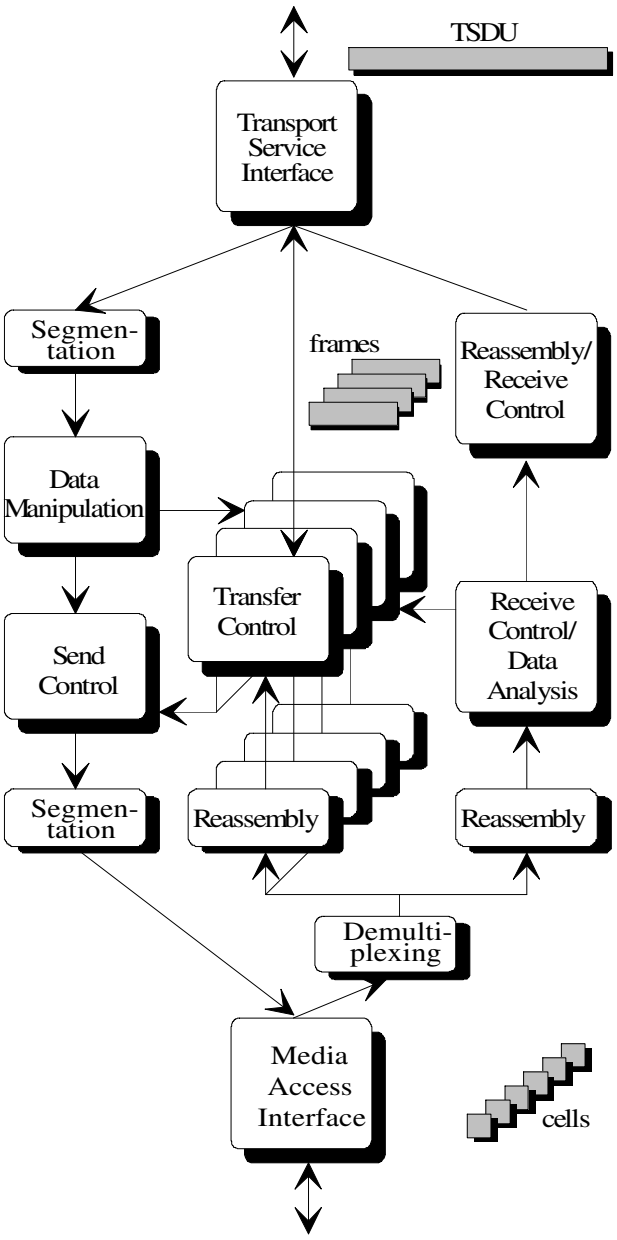

Figure 2 : Architecture of PATROCLOS

- transport user interface functions
- user data transfer
- connection management (establishment, termination, and monitoring of connections)
- flow and rate control
- congestion control and estimation of round trip time
- acknowledgement and retransmissions
- delay jitter control

In addition to parallel execution, another advantage of the decomposition into independent FSMs is their modularity. Depending on the requirements of the applications and the characteristics of the networks, strategies, mechanisms and parameters can be changed and configured very easily. Well defined FSM interfaces become important. Configuration issues of transport systems are considered in [STZ91].

FSMs of the same transport entity exchange messages for synchronisation and communication purposes. If possible, internal event signalling is performed periodically to reduce overhead.

FSMs of different entities communicate with each other by individual protocols to avoid coordination points. All FSMs, except for the transport interface functions FSM, communicate with the corresponding peer FSM. Each communication relation has its own protocol and protocol data units. Multiplexing of protocol data units by different FSMs will be avoided to support parallelism. The FSM protocols have their own error recovery mechanisms and timers if necessary. So, timer administration as well as other time consuming functions such as buffer management are distributed to all FSMs.

Transfer control is designed to be independent of user data transfer by decoupling the connection state exchange from user data traffic. Acknowledgements, connection state information exchange etc. are triggered by expired timers, requests, or error situations and not by user data frames sent or received.

The transport system consists mainly of the following FSMs:

- Connection management

  There is one FSM for the management of the complete duplex connection. The FSM has to support establishment, monitoring, maintaining, and correct termination of connections. It provides handshake-based or implicit establishment mechanisms according to the required service. The FSMs exchange frames for establishment. They have to initialise the connection and have to forward quality-of-service requirements to the other FSMs. Connection monitoring may be supported by exchanging periodic monitoring frames. The FSMs have also to guarantee secure termination and to control unique use of connection identifiers. Further division of the connection management FSM is not performed, because establishment, maintaining, and termination of connections occur successively. Several frame types are necessary to distinguish frames for the different tasks of the FSM.

- Routing and relaying

  The FSM performs functions required in gateways such as relaying, routing, route management, and monitoring.

- User data transfer

  The FSMs for user data transfer send and receive user data. The main tasks are formatting and analysis of user data packets.

- Flow and rate control

  For flow and rate control there are two FSMs for each connection, i.e., one FSM for each transfer direction. The FSMs control data flow by the interpretation of the peer entity's feedback signals, e.g., window and buffer sizes, and rate values. For rate control, both peer FSMs negotiate appropriate rate parameters, e.g., the receiver FSM can transmit rate parameters or credit values according to the current traffic or available buffer space. The peer's sender FSM triggers sender control functions of the same entity by signalling parameters to the send control. It may also request new credit values from the peer's receiver FSM.

- Congestion control and estimation of round trip time

  The congestion control FSM controls data flow by the interpretation of feedback signals from the network or the peer entity, e.g., congestion indications, round trip times, and inter-arrival times of test packets. The FSM may send test data to identify congestions or to estimate round trip times. Based on the test packet delays and inter-arrival times, congestion situations can be detected and the rate can be decreased [KAS91]. To avoid large reaction times in networks with high bandwidth-delay-product the FSM may also react to the feedback signals of the network nodes.

- Acknowledgement and retransmissions

  The acknowledgement / retransmission functions can also be divided into two FSMs. The sender FSM receives acknowledgements from the peer's receiver FSM. It may also request acknowledgements, or send the state of sent and acknowledged data to the peer's receiver FSM, but in high speed networks automatic sending of positive or negative acknowledgements without special requests seems to be more appropriate in order to reduce delays.

  Retransmissions are performed or initiated dependent on the acknowledgement information. The retransmission strategy and the management of unacknowledged data is implemented in this FSM. The sender FSM controls buffer of sent user data and has to release buffer space of acknowledged data. The receiver FSM sends negative or positive acknowledgements to the peer's sender FSM and requests retransmissions based on the results received from the user data receiver functions.

- Delay jitter control

  The jitter control FSMs support the timing relationship between sent and received data. Several applications require well defined data packet inter arrival times. These FSMs send and receive time stamps for transferred user data. The receive FSM has to control correct delivery of user data to the transport system user considering the received time stamps.

The FSMs are addressed by entity address extensions. The extensions are simple bit maps, each bit represents a certain FSM. This scheme allows multicast addressing between protocol entities. This can be useful if a FSM of one entity has to send the same data to several peer entity FSMs. In that case, the data are transmitted once and are distributed at the peer entity to all destination FSMs. But, mainly peer FSMs with the same functions communicate with each other.

The PDU for all FSMs, the so-called frame, consists of a general header and a FSM specific header. The general header contains addresses, identifier, type fields, and lifetime values, e.g., the FSM specific header for user data consists of length informations, sequence numbers, and several flags. By using the IP addressing scheme, the header of a user data frame has the size of exactly one cell. Control data frames have usually fixed sizes of only one or two ATM cells to be appropriate for cell based networks. In networks with larger transmission units, e.g., FDDI, multiple control frames have to be concatenated.

Because control data frames have usually fixed sizes of only one or two cells, they can be transmitted without significant overhead via ATM or DQDB networks. Furthermore, the amount of control frames consisting of only a few cells is very low compared with the larger user data frames. The additional bandwidth overhead will be very low. It seems to be reasonable to waste bandwidth in order to reduce protocol processing time, because in high speed networks bandwidth is not the limiting factor for performance.

3.3. Cooperation among FSMs

Figure 3 shows the cooperation between the most important FSMs. The connection management FSMs are responsible for connection initialisation and correct release. They have to ensure that all data, which are delivered before initiating connection termination, are sent and acknowledged.

The flow control state machine on sender side controls the user data sender by flow control parameters. The user data FSM can request new credit values.

On receiver side, the user data receiver has to signal buffer state information to the flow control FSM, which calculates new parameters for the peer's sender flow control FSM.

The user data FSM gives control on sent user data to the retransmission FSM, which is responsible for further retransmissions. On receiver side, the user data receiver signals informations about received data to the acknowledgement FSM. Event signalling between the user data transfer FSM and the acknowledgement / retransmission FSM is performed periodically with short time intervals, if possible.

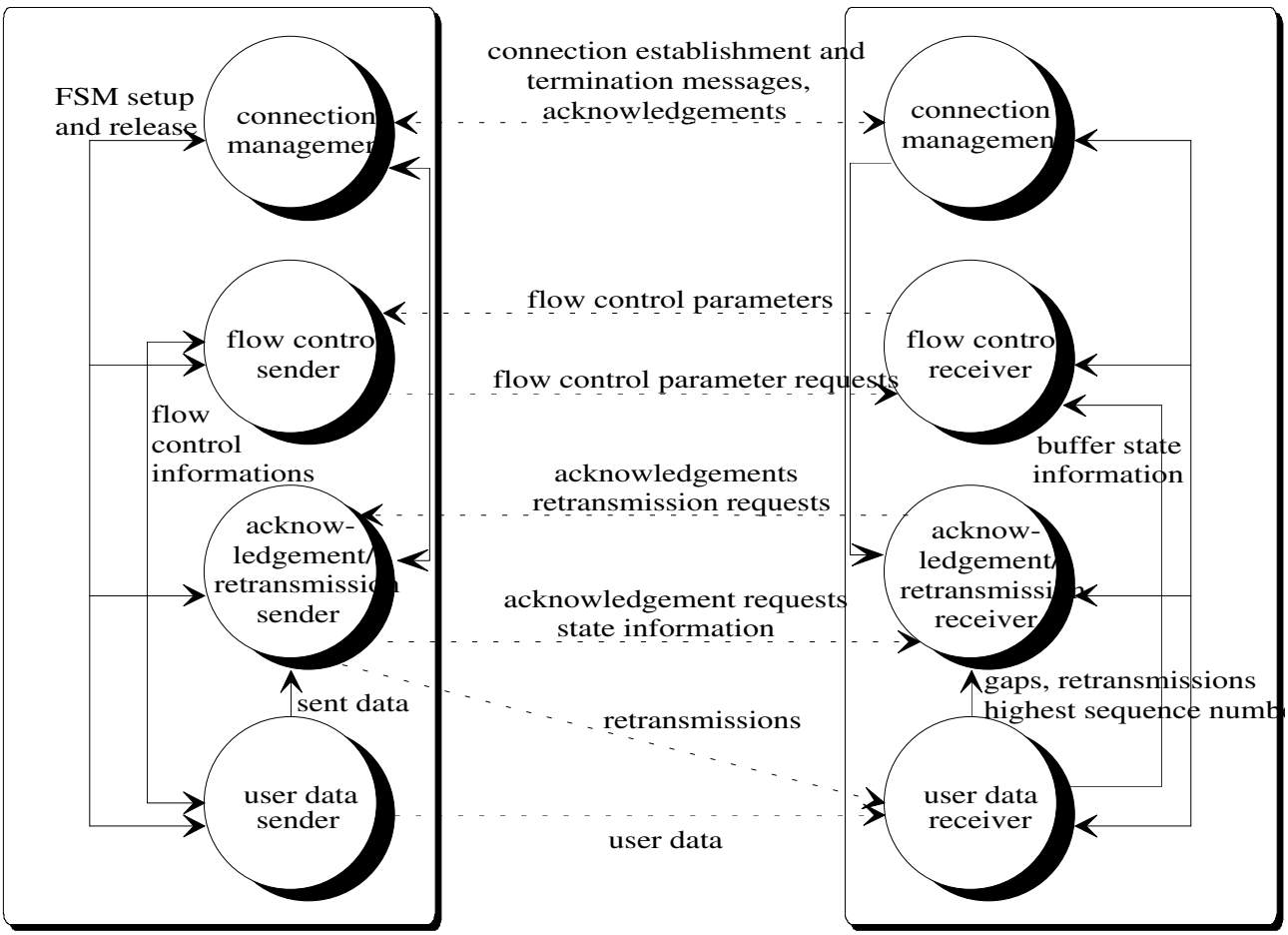

Figure 3: Cooperation among several FSMs

3.4. Implementation

PATROCLOS is appropriate for implementation on a Transputer [INM89] based multiprocessor architecture. Parallel architectures for communication subsystems are successfully built with Transputer networks [BRZ91, DIR91, KAI91, ZIT91]. The FSMs are mapped to processes, which are mapped to processors of a suitable multiprocessor configuration [SCH92]. The processes are connected by logical channels, which are mapped to software or hardware channels depending on processes to interconnect are on the same processor or on different processors. Processes of the same entity communicate with each other via message passing across channels or via common data in shared memories. Global memories for the user data packets to send and to receive are realised by multi-port memories.

For time-critical functions like checksum calculation, timer processing, or buffer management, specialised hardware units will substitute software implementations (cf. [SID91]). Furthermore, simple protocol functions on cell level (e.g., SAR processing according to AAL type 4) will be implemented by dedicated hardware processing units to support the high network bandwidth. The combination of universal processors and specialised processors results in a hybrid and heterogeneous multiprocessor architecture to provide high performance and a high degree of flexibility simultaneously.

The implementation architecture is depicted in Figure 4. It will be connected via a DMA unit to a host system.

On sending side, TSDUs are segmented by the segmentation processor and delivered to the data manipulation processor. The send control processor receives frames from the data manipulation processor and from the transfer control processors. It determines the next frame to send and delivers corresponding frame informations to the segmentation hardware, which performs segmentation into cells. Cell headers and trailers are exclusively generated by the segmentation hardware.

On receiving side, the reassembly hardware units have to reassemble the incoming cells to frames. A demultiplexing hardware performs demultiplexing below the reassembly level. The incoming cells are demultiplexed to the reassembly processing units of the data manipulation processor and the transfer control processors. The demultiplexing destination depends on the FSM address of BOM (beginning of message) and SSM (single segment message) cells and on the MID value of COM (continuation of message) and EOM (end of message) cells.

The reassembly hardware has to signal completely received frames as well as errors, e.g., bit errors, cell loss, and mis-sequenced cells. Error recovery, however, is not performed on a per-cell basis. If an error occurs, all cells of the corresponding frame are discarded. Received user data TPDUs are delivered via the receive control processor to the reassembly processor, which generates TSDUs. The jitter control processor controls correct delivery to the transport service user.

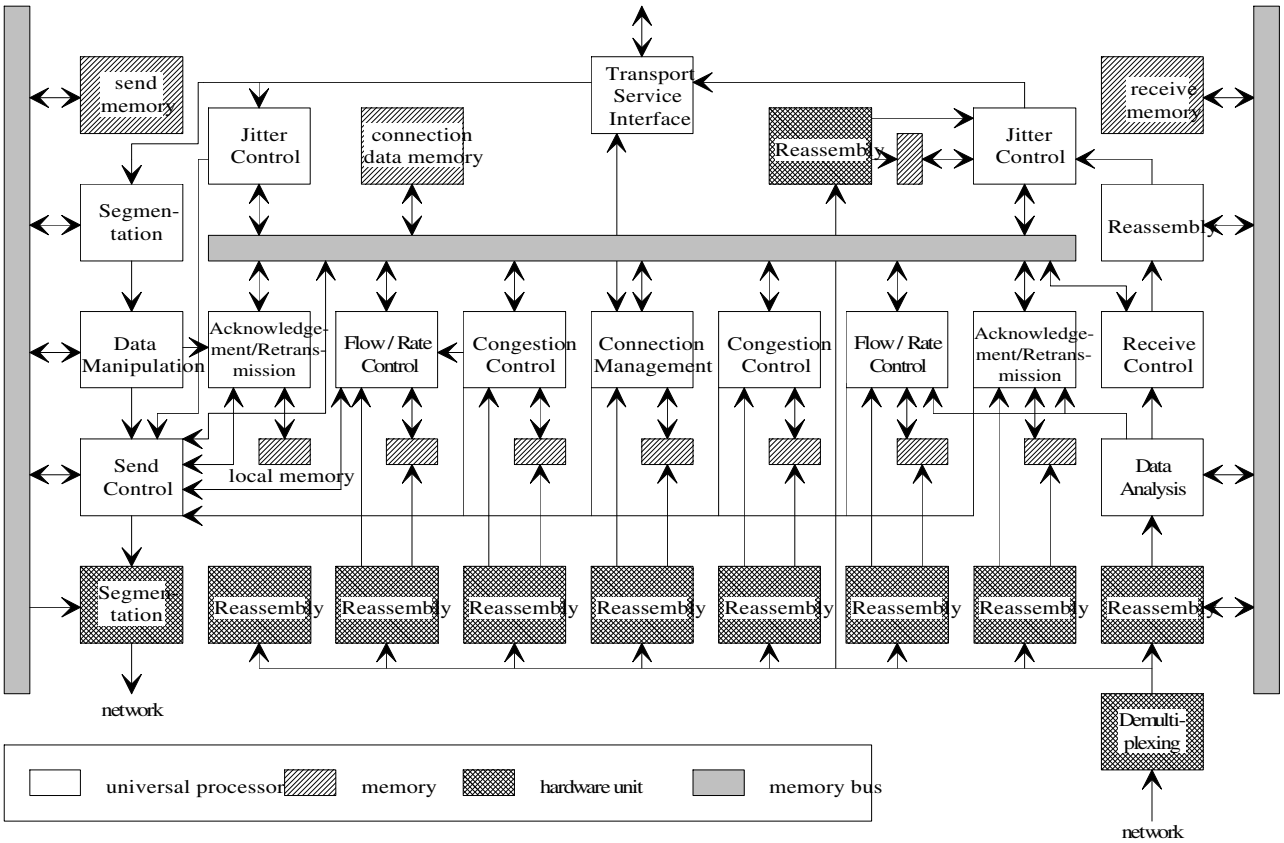

Figure 4: Implementation Architecture

4. Conclusions

This paper presents a transport system called PATROCLOS for high speed networks with small transfer units. The transport system is on top of the network access and provides a reliable transport service. It is designed to work efficiently in modern high speed networks such as DQDB and ATM. PATROCLOS is based on parallel principles to be implemented on a heterogeneous multiprocessor architecture.

The key issue is the decomposition of a protocol entity into concurrently executable finite state machines. Therefore, the protocol was decomposed into modular FSMs being able to operate almost independently from each other.

The design comprises a separate FSM for each control function and for user data processing. With respect to concurrent processing, synchronisation among FSMs of a single protocol entity was reduced by associating each FSM with an individual protocol for message exchange between FSM residing in different entities.

The designed parallel transport system is currently being implemented on a hybrid multiprocessor architecture using Transputers as general purpose processors and dedicated hardware for time-critical protocol functions.

Acknowledgements

The author would like to thank Dr. Martina Zitterbart and Burkhard Stiller for their contribution to the presented work.

References

[BRZ91] Braun, T.; Zitterbart, M.
A Parallel Implementation of XTP on Transputers
16th Annual IEEE Conference on Local Computer Networks, October 14-17, 1991, Minneapolis, USA, pp. 172-179

[CHE91] Chesson, G.
The Evolution of XTP
3rd IFIP WG 6.4 Conference on High Speed Networking, Berlin, Germany, March 18-22, 1991, pp.15-24

[DIR91] Diot, C.; Roca, V.
XTP/KRM Implementation on a Transputer Network
16th Annual IEEE Conference on Local Computer Networks, Minneapolis, U.S.A., October 14-17, 1991, pp. 310-320

[CLT90] Clark, D.; Tennenhouse, D.
Architectural Considerations for a New Generation of Protocols
ACM SIGCOMM ´90, September 24-27, 1990, Philadelphia, USA, pp. 200-208

[FEL90] Feldmeier, D.
Multiplexing Issues in Communication System Design
ACM SIGCOMM ´90, September 24-27, 1990, Philadelphia, USA, pp. 209-219

[INM89] INMOS Ltd.
The Transputer Databook
2nd edition 1989, Redwood Burn Ltd, Trowbridge.

[KAI91] Kaiserswerth, M.
A Parallel Implementation of the ISO 8802-2.2 Protocol
IEEE TriComm '91, Chapel Hill, NC, USA.

[KAS91] Keshav, S.; Agrawala, A.; Singh, S.
Design and Analysis of a Flow Algorithm for a Network of Rate Allocating Servers
in: Johnson, M. J. (ed.): Protocols for High-Speed Networks, II, North-Holland, 1991, pp. 55-72

[LAI91] Lai, W.S.
Recovering from Congestion Losses in High-Speed Networks
in: Johnson, M. J. (ed.): Protocols for High-Speed Networks, II, North-Holland, 1991, pp. 17-23

[SNR90] Sabnani, K.; Netravali, A.; Roome, W.
Design and Implementation of a High Speed Transport Protocol
IEEE Transactions on Communications, Vol. 38, No. 11, November 1990, pp. 2010-2024

[SAW90] Sanders, R.; Weaver, A.
The Xpress Transfer Protocol (XTP) - A Tutorial
Computer Communications Review, Vol. 20, No. 4, October 1990, pp. 67-80

[SCH92] Schmidt, C.
Design of a Transport System Based on Parallel Finite State Machines
Diploma thesis (in German), University of Karlsruhe, in preparation

[SID91] Sidenius, M.
Hardware Support for Implementation of Transport Protocols
in: Johnson, M. J. (ed.): Protocols for High-Speed Networks, II, North-Holland, 1991, pp. 251-267

[STZ91] Stiller, B; Zitterbart, M.
Parallelism in Transport Systems Supported by a Protocol Configuration Manager
1991 International Symposium on Communications, December 10-13, 1991, Taiwan, Republic of China, pp. 712-715

[ZIT91] Zitterbart, M.
High Speed Transport Components
IEEE Network, Vol. 5, No. 1, January 1991, pp. 54-63