A Transputer Based OSI-Gateway for LAN-Interconnection across ISDN

Torsten Braun

Institute of Telematics
University of Karlsruhe
W-7500 Karlsruhe, Germany

Martina Zitterbart*

IBM Research Division,
Thomas J. Watson Research Center
Yorktown Heights, NY 10598, U.S.A.

Abstract

Interconnections of remote local area networks (LANs) across public networks are of growing interest. This paper presents an OSI (Open Systems Interconnection) gateway to interconnect LANs across Integrated Services Digital Network (ISDN). The OSI connectionless network protocol (CLNP) supports interconnection of different subnetworks. The most important component of the gateway is an adaption layer to integrate ISDN into the gateway structure. The main task of this adaption layer is mapping the service provided by the connection oriented ISDN subnetwork to the service expected by CLNP and vice versa. It is designed to fulfil requirements on fairness, low costs, and acceptable performance. The gateway components are implemented on a multiprocessor platform built with Transputers.

1. Introduction

Local area networks are being used in many different applications spanning office and industrial environments. Often users located on different, remote LANs want to communicate with each other. Remote LANs have to be connected across public networks, e.g. ISDN. Public networks and LANs are connected by interworking units (IWUs) or gateways.

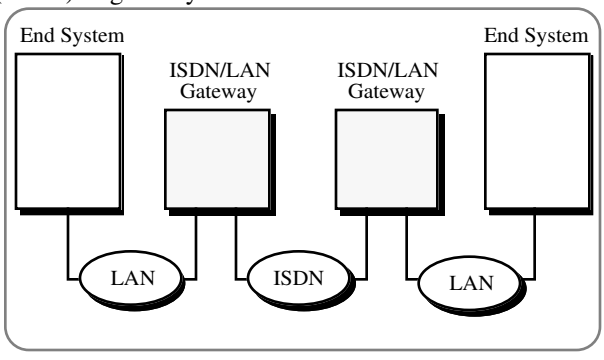

Figure 1: LAN Interconnection across ISDN

This paper presents a gateway to interconnect IEEE 802.x-LANs across ISDN. It is implemented on a multiprocessor architecture that is realized by a Transputer network.

The paper is organized as follows: Section 2 discusses differences between LANs and ISDN. In this section the frame relay approach, a special gateway type based on the connectionless network protocol (CLNP), and the integration of an ISDN subnetwork into this gateway type are also described. The latter approach has been chosen to realize the LAN/ISDN interconnection.

Gateway systems must be very powerful because data traffic of several underlying subnetworks is concentrated at a gateway. For implementations of communication system protocols Transputer networks are well suited [1]. For that reason, the gateway presented here has been implemented on Transputer networks. The implementation environment is briefly presented in section 3. Section 4 discusses a parallel gateway architecture model based on Transputer networks for the special gateway type described in section 2.

Integration of the connection oriented ISDN to the connectionless network protocol is the most important task of the presented gateway. An adaption layer integrates ISDN into the gateway. This adaption layer, which is described in detail in section 5, is totally transparent to end systems and has only to be implemented in gateway systems. Especially the adaption layer's tasks to provide fairness, low costs, and high performance are examined. Moreover, performance results of implementations on several Transputer networks are presented in this section. Finally, section 6 concludes this paper.

2. LAN/ISDN Interconnection

First, in this section the networks to interconnect, i.e. LANs and ISDN, are compared. Second, two approaches for the interconnection of these networks are discussed: the frame relay and an OSI-layer-3-gateway based on the connectionless network protocol.

2.1. Comparison of ISDN and LAN

LANs and ISDN are very dissimilar networks (cf. table 1).

*The work was performed at the University of Karlsruhe.

	LAN	ISDN
owner	private	public
extent	local, limited	world wide
network control	distributed	centralized
signalling	in band	out of band
data transfer model	multiplexing of virtual channels or messages on a single medium	switching of channels
access layer	MAC layer	network layer
topology	ring, bus, star	partially meshed network
transmission rate	4-100 Mbit/s	64 kbit/s per B-channel
bit error rate	low (10^{-8}-10^{-9})	high (10^{-5}-10^{-6})

Table 1: Comparison of LANs and ISDN

Often connectionless protocols on layers 3 and 2 are preferred for LANs, while in ISDN connection oriented protocols are used. There are also ongoing efforts to make LAN and ISDN protocols more compatible to allow easy interconnection by the so called frame relay approach, which is described in the next subsection.

2.2. Frame Relay

In frame relay, interconnection of ISDN and LANs is performed on layer 2 [2]. Frames are relayed between the subnetworks by a frame relay function (cf. figure 2). Frame relay is based on the idea of using similar protocols in LANs and ISDN to support interconnection. Protocol changes in LAN and in ISDN protocols are required.

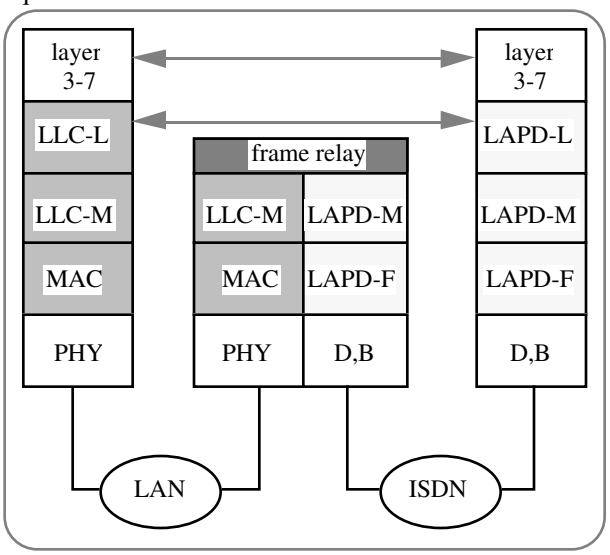

Figure 2: Frame Relay

Following this approach, layer 2 in both, LANs and ISDN, is divided into three sublayers. On one hand, ISDN layer 2 consists of three LAPD (Link Access Protocol for D-channels) parts: LAPD-F, LAPD-M, and LAPD-L. The tasks of LAPD-F, i.e. the frame sublayer, are for example frame inspection, error detection, and frame delimiting. LAPD-M is called the addressing sublayer and handles frame multiplexing and logical addressing. The address field is named data link connection identifier (DLCI). LAPD-L is a HDLC-type data link control protocol and provides in-sequence and error-free frame transmission across the user-network interface.

On the other hand, the logical link control (LLC) of LANs is divided into LLC-M and LLC-F. The third sublayer of layer 2 is the MAC layer comprising similar functionality such as LAPD-F. Analogous to LAPD-M, the LLC-M is the addressing sublayer. The LLC address field consists of destination and source parts, called service access points. Again, the LLC-L sublayer is a HDLC variant that is defined close to LAPD-L.
It is intended to make LLC-L and LAPD-L compatible to interconnect LAN and ISDN. So, LAN and ISDN are interconnected above the addressing sublayer by relaying frames between LLC-M and LAPD-M. The most important task of the frame relay function is address mapping.

This approach has the drawback that protocols used in ISDN and in existent LANs have to be modified. For that reason, frame relay is not considered further in this paper. The next subsection discusses an interconnection model that does not require new protocols in LANs or in the ISDN at all.

2.3. Connectionless OSI-Layer-3-Gateway

The International Organization for Standardization (ISO) has defined an internetworking architecture subdividing the network layer into three sublayers 3a, 3b and 3c. Layer 3a (subnetwork access function, SNACF) is responsible for the access to the underlying network. Layer 3b (subnetwork dependent convergence function, SNDCF) adapts the service provided by the subnetwork to the service required by layer 3c (subnetwork independent convergence function, SNICF).

The main tasks of layer 3c are addressing, relaying, routing, segmentation, and reassembly. It is obvious that in gateways congestions can occur. This connectionless OSI-layer-3-gateway provides no flow control mechanisms. Congestions are avoided by discarding packets. Higher layers such as transport protocols, e.g. OSI transport protocol class 4 (TP4), have to perform flow control functions and mechanisms to recover from frame losses.

LANs standardized by IEEE usually provide above the MAC layer a protocol stack consisting of LLC protocol type 1, CLNP (layer 3c) and TP4. Layers 3b and 3a are empty. CLNP is able to act as a convergence layer and expects a connectionless network service.

In contrast to LANs, the specific D-channel protocols at layers 2 and 3 have to be used for ISDN. Because the main goal of the gateway approach presented in this paper is to interconnect LANs across ISDN, modifications of used protocols should be avoided. Using CLNP as the convergence layer of the gateway, only services of LANs and ISDN have to be adapted to the expected service. The protocol structure of the gateway is depicted in figure 3.

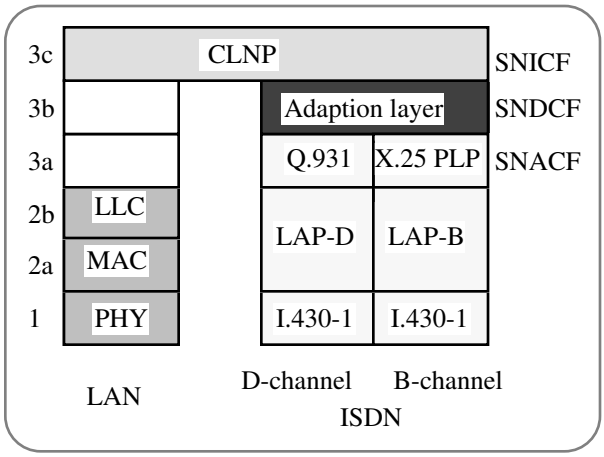

Figure 3: Gateway Protocol Model

An advantage of the suggested gateway protocol structure is the transparency of the inserted adaption layer. The adaption layer is needed only in gateway systems to integrate connection oriented networks such as ISDN. Changes in LAN end system protocol stacks are not required. Other advantages of this gateway structure are the simplicity of the used protocols and thus high achievable performance [3], [4]. [5] discusses a similar LAN/ISDN gateway based on the internet protocol (IP).

Especially gateways have to be implemented efficiently to fulfil the high performance requirements, because data traffic of several networks is concentrated at a gateway system. In [1], [6] high performance protocol implementations based on Transputer networks are examined. Also the gateway presented in this paper is implemented on Transputers that are well suited for communication protocol processing. The implementation environment is briefly discussed in the following section.

3. Implementation Environment

The Transputer [7] is a family of high performance microcomputers including a processor, a timer, on-chip memory, an external memory interface, and four communication links on a single chip. One of its most significant features is the ability to perform efficient multi-tasking in hardware. The Transputer uses processes as fundamental standard software building blocks. Processes are able to communicate using the synchronized and unbuffered rendez-vous mechanism. Due to this rendez-vous-based communication, Transputer networks are loosely coupled multiprocessor systems with distributed memory. Transputer networks are built by connecting the links of different Transputers.

During the project described in this paper a parallel Transputer C Compiler (3LC) [8] is used, which provides the most significant features of the OCCAM language including the datatype CHAN for interprocess communication channels, instructions for message passing across these channels, process management, and timer access.

This C Compiler provides two types of processes: tasks and threads. A task is an independent unit with private memory. Cooperation with other tasks is done by message passing across channels exclusively. A task is the smallest unit that can be mapped to a single Transputer. Tasks themselves may consist of several threads. Threads are able to cooperate either by message passing or by operation on common data variables. In the latter case access conflicts are avoided by semaphores.

Configurations, i.e. the placement and connection of tasks, are described in a text file. Based on this information the development system generates automatically the boot program for the described Transputer network.

The used Transputer boards for PC/AT systems can execute the loaded software without any interactions with the PC system. Transputer based communication controller boards are available as prototypes or as commercial products for several networks such as Ethernet, Token Bus, FDDI, and ISDN. A gateway can be implemented with Transputers exclusively by connecting the Transputer based controller boards via the links or even via Transputer networks. No interaction with the PC system is required. In the next section a Transputer based gateway model is examined.

4. Transputer Based Gateway Model

Based on several concepts of parallelism in communication systems a parallel gateway architecture can be derived. The essential concepts are presented in the following. A detailed discussion is also given in [1].
Communication systems are typically structured in hierarchical protocol layers, as suggested by the OSI reference model. Thereby, structural and functional parallelism are distinguished. Structural parallelism can be derived directly from the layered model and functional parallelism is based on the configuration of protocol functions inside the protocol entities [1].

The protocol layers operate in a pipelined principle: each packet moves step by step through the protocol layers. Thereby, several packets may be processed concurrently in different protocol layers. Some of the layers even can be divided into sublayers, e.g. the network layer 3 into the sublayers 3a, 3b and 3c. Such a

layer pipeline performs overlapped computation of sublayers to exploit temporal parallelism. Additional to the hierarchical structuring of sublayers, the protocol entities of those layers can be subdivided internally into different pipelines.

First, dividing the communication system protocols into a send and a receive part results in a send and a receive pipeline (cf. figure 4). The send and the receive pipeline are not totally independent of each other. In the next step, pipelines for each subnetwork can be derived. The gateway has to forward received packets from one network to another one.

Considering the concept of structural parallelism, a gateway consists of several receive and send pipelines each connected to a dedicated subnetwork (cf. figure 4). Thus temporal and spatial parallelism can be achieved at the same time: the receive pipeline of a subnetwork processes packets from this subnetwork in temporal parallelism, whereas receive pipelines of different subnetworks process packets in spatial parallelism. In figure 4 a sublayer is depicted using a grey rectangle, which may represent a multiprocessor architecture based on pipelines and arrays of processors [6].

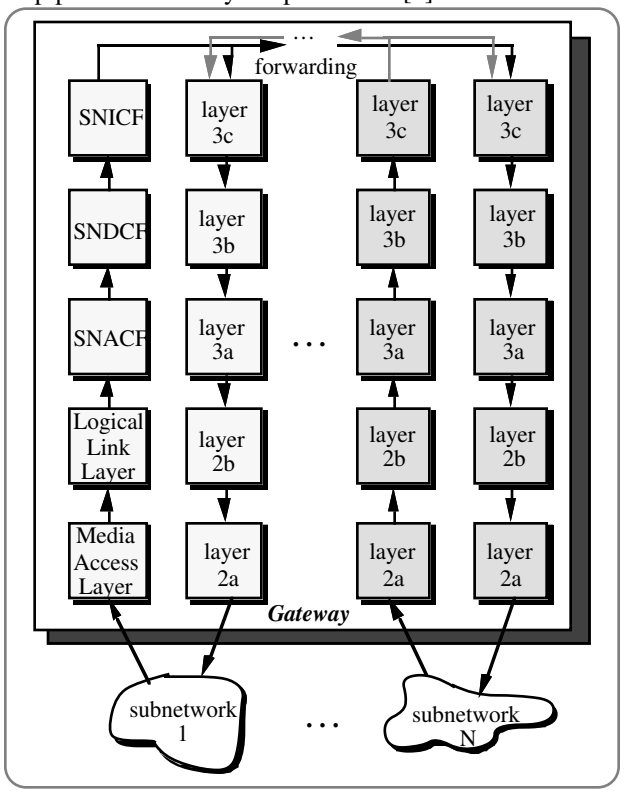

Figure 4: Parallel Interworking Unit

The connection oriented ISDN subnetwork supports another possibility to achieve spatial parallelism. For the communication between two ISDN end systems so called B-channels are necessary. Several functions can be provided not only for ISDN, but also for a single or a subset of B-channels. There may be several pipelines for each S_0 interface, as e.g., shown in figure 5. These pipelines with channel specific functions comprise layers from the physical layer to the adaption layer (SNDCF). Details are discussed below.

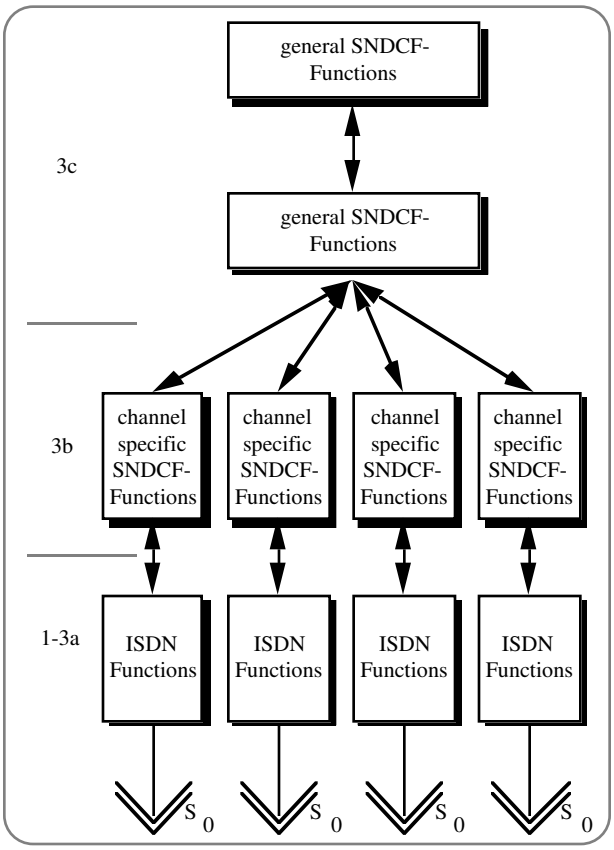

Figure 5: Pipelines with Channel Specific Functions

To avoid data movements between the processing components of a pipeline a ***global data memory*** is required. For the organization of the global data memory there mainly exist two possible solutions:

- a global data memory for all subnetworks, or
- global data memories for each subnetwork or even for channel subsets in ISDN networks.

In the presented gateway the latter solution with data memories for each subnetwork is realized because available communication controllers allow only local memory access.

According to the parallelism and the memory concepts the layer 3c (CLNP) is implemented on several Transputer network configurations. The implementation and performance results are described in detail in [6]. A high performance configuration on eight Transputers is able to process more than 2000 packets/s. This paper describes the layer 3b (SNDCF) implementation for ISDN integration into the CLNP.

5. SNDCF-Layer for ISDN Integration to Connectionless Network Service

5.1. Design Goals

The most interesting part of the gateway is the SNDCF layer for the integration of ISDN to the gateway. The SNDCF layer adapts the connection oriented ISDN service to the connectionless network service that the CLNP expects and vice versa. Single data packets of CLNP have to be transferred across one or more ISDN channels that have to be established and terminated. For channel establishment and termination a certain strategy is necessary because the design of SNDCF is influenced by the following aspects:

- High performance
  Available ISDN channels must have a high utilization with small idle times to achieve high throughput and low delay. Because channel establishment and termination may be very time consuming, frequent channel establishment and termination must be avoided. If several channels are available, e.g. by primary rate ISDN or by several basic rate ISDN interfaces, data traffic can be distributed to more than one B-channels.
- Low costs
  The costs are extremely dependent on the channel utilization and on the tariff structure of the ISDN provider. On one hand, it can be very costly to establish and terminate channels very frequently. On the other hand, duration of channel utilization can be the main cost factor. Also collisisons, i.e. when two gateways establish an ISDN-B-channel to each other at the same time, can lead to inefficient channel utilization.
- Fairness
  If there are less channels available than connections required, connections to establish must be selected. In this case, connections with permanent channel utilization must be terminated to avoid priorisation.

The aspects mentioned above are dependent on each other, e.g. fairness can reduce throughput, if connections with high channel utilization are terminated in favour of connections with less channel utilization. Preferred aspects have to be defined. A strategy component is necessary to achieve the desired behaviour of the gateway.

5.2. Implementation Model

The SNDCF layer must perform the conversion of the connection oriented service into the connectionless service and vice versa. It must fill the gap between the the SNICF and the SNACF layer. The ISDN interface is often realized by an ISDN application program interface (ISDN-API) dependent on the used ISDN controller board. Several German vendors developed a common ISDN-API for basic rate ISDN controller boards.

Data packets of SNICF have to be transferred across ISDN connections mapping the packets to connections. A complex connection management is necessary to fulfil the goals mentioned above. The SNDCF layer (layer 3b) is divided into two parts:

- general functions (3b2 functions) and
- channel specific functions (3b1 functions).

The interface to the ISDN-API consists of the channel specific functions. They are specific to a single or a subset of channels. Examples are establishment, monitoring, and termination of B-channels, as well as sending and receiving of data packets across B-channels. The more complex and general functions are the former ones that need information about all accessible ISDN channels.

The functionality of SNDCF layer is depicted in figure 6. The central component of the SNDCF layer is the block management including the following functions:

- distribution/collection of packets to/from blocks,
- establishment and termination of channels,
- monitoring of channel and block states,
- assignment of blocks to channels, and
- collision avoidance.

Data packets are received from the SNICF layer and are distributed to so called blocks (send- and receive-blocks) that contain queues of packets to be sent to/from the same ISDN node. In contrast to the number of send-blocks the number of actual receive-blocks is limited by the number of available channels. The send- and receive-blocks are mapped to the channels. The channel assignment component controls this mapping function using channel and block state information.

In general, there are more blocks than available channels (2 channels in basic rate ISDN, 30 channels in primary rate ISDN). Furthermore, the data rate required by one block may be higher than the data rate provided by a single channel. The block management has to select blocks to associate them to channels. At least one channel must be associated to a block to send data. It is also possible to associate a couple of channels to a block, if a very high data rate is required. Channels must be established and terminated to realize the goals mentioned in the previous subsection.

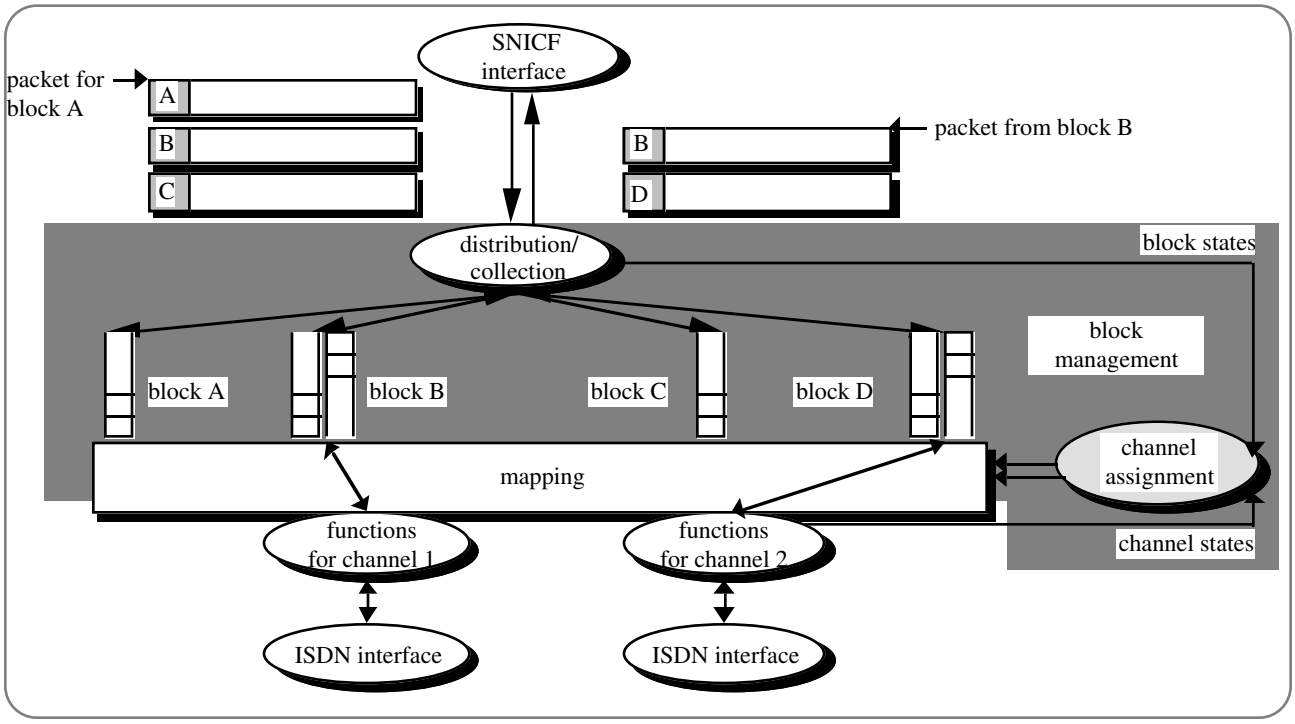

Figure 6: SNDCF Layer

The channel assignment component has to observe data traffic characteristics of the block and channel states to perform the right decisions. Examples for channel state parameters are incoming call setup, disconnect request, and idle time. Block states describe size, addresses, priority, or lifetime of the blocks. Channels are established, terminated, and associated to other blocks dependent on block and channel states. Decisions are supported by a channel assignment strategy component as shown in figure 7.

If there are waiting blocks and free channels, blocks are assigned to these channels regarding the block states. Blocks can be assigned to more than one channel, if they exceed certain size values. The block size is measured by the number or by the length of data packets.

Channel displacement of blocks is triggered by timers. A timer ("all busy timer") is started, if all channels are busy. After a certain time period the timer may expire. Then, a block is displaced, e.g. the block that was assigned for the longest period. This mechanism provides fairness to all blocks.

To reduce unnecessary channel utilizations a block is displaced, if it is idle for a certain period of time. The initial idle timer value is dependent on several parameters, e.g. the timer value can depend on tariff structures or connection establishment times. Very small timer values may lead to low throughput, if the connection establishment has large delay. Very large timer values lead to low throughput rates. A network management instance should be able to adjust all timer values.

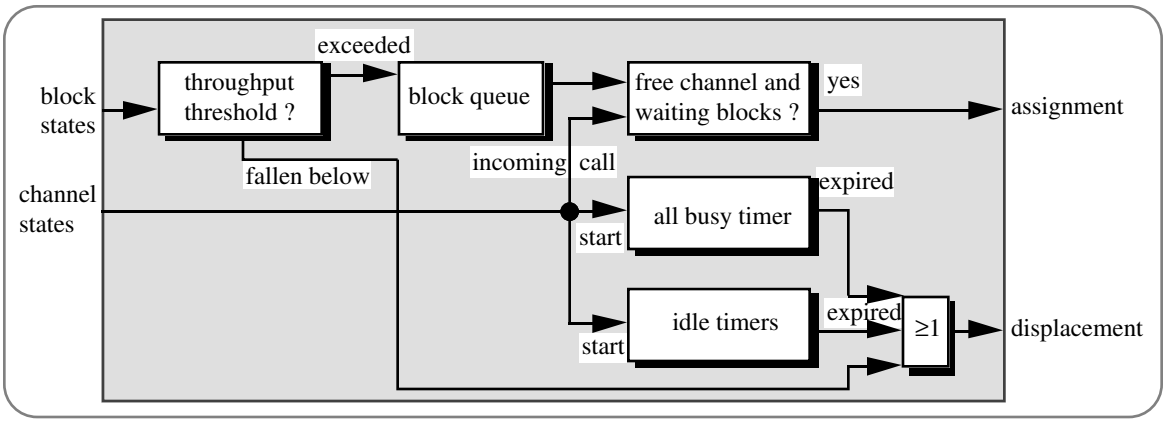

Figure 7: Channel Assignment Strategy Component

Moreover, a channel is terminated, if the corresponding block throughput falls below a certain threshold value and if more than one channel is assigned to the block.

Another block management function is collision avoidance. An incoming and an outgoing call for one channel at the same time lead to a collision. Without collision avoidance two channels are established that may not be utilized efficiently. Collision avoidance is performed by starting a collision avoidance timer after establishing a channel. If the opposite instance desires to establish a channel within this time period, a collision is assumed and only one channel is established. The value for the collision avoidance timer is estimated by the average connection establishment delay.

5.3. Implementation

Figure 8 depicts the SNDCF layer implementation [9]. The SNDCF layer consists mainly of two tasks: the block manager task and the channel manager task. There exist two different additional **interface tasks** (IF) that are responsible for adaption of the implementation to CLNP and ISDN-API.

The block manager task (BM) and the channel manager task (CM) consist of several threads. There exists always one thread for each input channel. Threads are using common data, common functions, and common output channels. Conflicts are avoided by semaphores.

The ***channel manager task*** (CM) performs channel specific functions such as establishing, monitoring and terminating channels. CM executes commands received from BM, e.g. CM opens or terminates a channel and sends packets across the channel. In receive direction CM delivers received packets to BM. CM starts the channel specific timers for the mentioned functions. There are timers for monitoring, establishing, and terminating channels as well as for monitoring of sending data by the ISDN controller software.

CM must also send and receive data packets to/from the controller interface because the ISDN controller software currently used is not able to support communications with memory managers.

In contrast to BM, CM can be multiplied for single channels or channel subsets, e.g. for each basic rate interface. In this case each pipeline can process the subset of B-channels, if it has access to a D-channel.

The ***block manager task*** (BM) is the central task of the implementation. The block management component of the model described in figure 6 is realized using this task. BM is responsible for distributing packets to blocks and for block transfer across ISDN channels. BM uses two submodules: the block assignment module (BAM) and the block and channel state data module (BCM). In this context a submodule is considered as a set of exported and internal functions operating on private data structures. The block assignment strategy as discussed in the previous subsection is implemented in BAM. BCM performs complete access to read or write channel and block state data. BCM manages and updates packets, blocks, and state information about channel and block states.

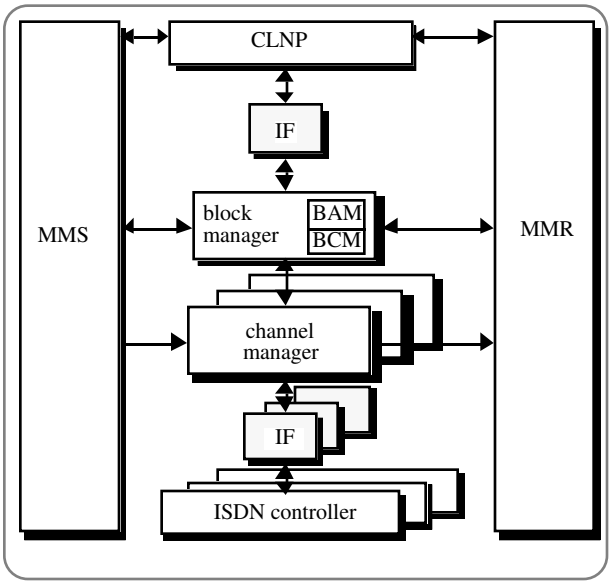

Figure 8: SNDCF Implementation

5.4. Performance results

Finally, the performance of the implementation is evaluated. In the test configuration an ISDN emulator is used that emulates access to 30 B-channels with data rates of 64 kbit/s. The load is generated by two load generators that are located in the ISDN controller emulator and above SNDCF implementation instead of CLNP. The used data packet size is selected very small to avoid effects of data copying by CM. There are three scenarios examined: sending with various destination addresses, sending with always the same destination address, and receiving with various destination addresses. The channels are always opened before the scenario.

Four configurations with 1, 3, 4, or 5 Transputers are evaluated. In the first configuration all tasks are running on the same Transputer. In the 3-Transputer configuration the upper and lower layers are separated from the SNDCF tasks. The memory managers and the SNICF layer are mapped to the first Transputer, the ISDN controller emulator to the second, and the other processes, i.e. the SNDCF tasks IF, CM, and BM to the third one. On the 4-Transputer configuration the upper IF process is separated. On the 5-Transputer configuration also BM has an own processor. The results in figure 9 present the throughput in packets/second.

Receiving is much faster than sending. The SNDCF layer has only to deliver packets to the higher layer. Sending is much more complex, especially in the scenario with different destination addresses. In this scenario distributing packets to blocks is very time consuming. However, the implementation is fast enough to support primary access interfaces with data rates of 2 Mbit/s.

In all scenarios it is obvious that the use of parallel architectures leads to better performance results. The configuration with five Transputers is about two times faster than the 1-Transputer configuration.

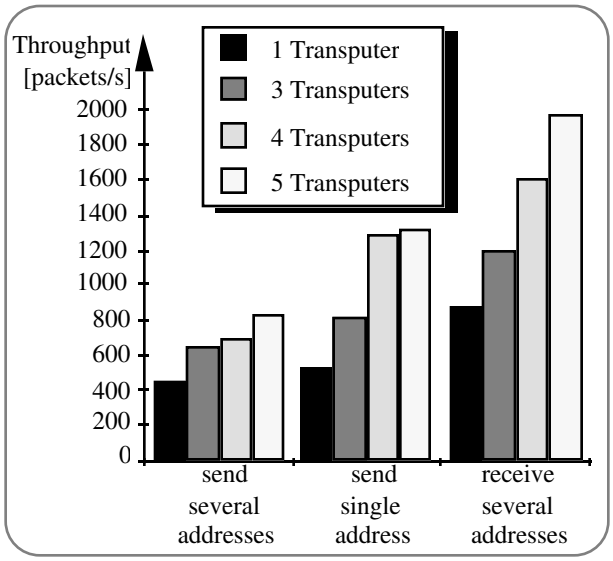

Figure 9: Performance Results

6. Conclusions

In this paper a multiprocessor implementation based on Transputer networks for a connectionless OSI-layer-3-gateway for LAN interconnection across ISDN is presented. Several concepts for a high performance gateway architecture based on a multiprocessor system are discussed. A special component of the gateway, the SNDCF layer for integration of ISDN into the gateway, and the implementation of this component is described. The performance results show that protocol implementations on multiprocessor systems lead to higher performance.

In [6] a high performance implementation of CLNP was presented. To achieve those throughput rates of nearly 2000 packets/s the SNDCF implementation has to be improved, e.g. by the support of even more parallelism.

References

[1] M. Zitterbart:
High Speed Transport Components
IEEE Network Magazine, January 1991

[2] D. Jäpel
LAN/ISDN Interconnect via Frame Relay
IEEE Globecom '88, Hollywood, Florida, November 28 - December 1, 1988

[3] D. Deniz, G. Knight
Channel Management Issues in LAN-ISDN Interconnection
in: J. Arnbak : "ISDN: Innovative Services or Innovative Technology ?", North-Holland, 1989

[4] M. de Prycker
LAN in an ISDN: Consistency with the OSI Reference Model
Computer Communications 8 (2), April 1985

[5] X. Zhang, R. Deng
Gateway Design for LAN Interconnection via ISDN
Computer Networks and ISDN Systems 19, 1990

[6] T. Braun, M. Zitterbart
High Performance Internetworking Protocol
15th Annual Conference on Local Computer Networks, Minneapolis, October 1-3, 1990

[7] INMOS Ltd.
Transputer Reference Manual
Prentice Hall, 1988

[8] INMOS Ltd.
3L Parallel C User Guide
February 1989

[9] E. Wolf
Entwicklung und Bewertung eines Ebene-3-ISDN/LAN-Gateways auf Transputern
Diploma Thesis, University of Karlsruhe, January 1991