A LAN WITH REAL-TIME FACILITIES BASED ON OSI CONCEPTS

A.J. de Raaf, A. Dijkstra and S.D. Swierstra

RUU-CS-85-19

August 1985

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

A LAN WITH REAL-TIME FACILITIES BASED ON OSI CONCEPTS

A.J. de Raaf, A. Dijkstra and S.D. Swierstra

Technical Report RUU-CS-85-19

August 1985

Department of Computer Science

University of Utrecht

P.O. Box 80.012, 3508 TA Utrecht

The Netherlands

To appear in the proceedings of the Conference "Computing in the High Energy Physics", Amsterdam, The Netherlands, June 25-28, 1985, L.O. Hertzberger and W. Hoogland (editors), North-Holland Publ.Comp.

A LAN WITH REAL-TIME FACILITIES BASED ON OSI CONCEPTS

A.J. de Raaf, A. Dijkstra and S.D. Swierstra

Department of Computer Science
University of Utrecht
P.O. Box 80.012, 3508 TA Utrecht
The Netherlands

ABSTRACT

Research is being done into structured design and realization methods for Local Area Networks (LAN's). The main aim is to develop a LAN (ZWOLAN) with real-time facilities for use in laboratories and based on ISO-OSI standards. Twentenet[1]) will be used for the physical and the data link layer of ZWOLAN. Twentenet is based on a Priority based CSMA/CD data link access mechanism with guaranteed access times. An implementation model has been constructed from an FSM decomposition analysis of OSI protocols. Modular Pascal[2]) will be used as language for the realization of the network software The emphasis is on the software architecture and the reduction of the OSI protocol overhead.

Keywords: Local Area Network, real-time, OSI, protocol, service primitive, finite state machine, protocol decomposition, implementation, VME-bus, M68000 processor.

INTRODUCTION

In high energy physics laboratories, computers are being used for data acquisition, experiment control, experiment protection and analysis. For the last twenty years several computer-based systems have been used for these purposes; they range from a single computer system to CAMAC based multi-computer systems [1,2]. As a result of new developments in micro-

---

[1]) Twentenet is being developed by the Twente University of Technology, the Netherlands.

[2]) Modular Pascal has been developed by the Departments of Computer Science of the University of Groningen, the Twente University of Technology and the University of Utrecht.

processors, new and more complex micro-processor based systems are coming into use. The introduction of Local Area Networks gives researchers new ideas about the design of multi-processor systems. The LAN for a high energy physics laboratory should provide real-time facilities. Ethernet-based LAN's using CSMA/CD access methods cannot provide such real-time facilities owing to the way in which collision conflicts are solved. Token rings and time-slotted rings sometimes have facilities for real-time interactions but are in general slow.

The only widely accepted standard interface for connecting devices to computers is the RS-232/RS-442 interface. Therefore this interface is generally used for box oriented LAN interfaces, despite their low transfer rate (up to 19.2 kbps). More sophisticated (DMA based) board oriented network interfaces are manufacturer-bound and limit the variety of computers that can be connected to the LAN.

For laboratory applications in high energy physics there are hardly any LAN's which can fulfil the performance and flexibility requirements. Most of the available LAN's, which have real-time facilities under special conditions, provide only the physical and the data link layer of the OSI Reference Model. In the case of vendor-bound networks, network software for the upper layers is in general available, although at a high cost in CPU and memory load. Many manufacturers have announced that they intend to support OSI standards, but the OSI implementations that are currently available are comprehensive and are inefficient for real-time laboratory applications.

To overcome the aforementioned shortcomings a research project has been started with the aim to develop a real-time LAN (ZWOLAN) based on OSI standards. New problems arise with respect to protocol overhead and as result of incomplete specifications and definitions. The goal of the project described here is to overcome these problems and to make contributions to OSI standardisation.

THE ZWOLAN CONCEPT

The characteristics of the ZWOLAN design are:

- open structure
- real-time facilities
- minimum transmission speed 10 Mbps
- serial data transport
- based on OSI standards
- extendible by subnetting
- segment length at least 1 km
- low cost

The network topology consists of one or more backbone networks and subnets. These nets are connected via so-called "bridges". A link to another network, for example an X.25 network, can be achieved via a "gateway". The computers, terminals and other resources are connected to ZWOLAN via "Network Connection Units" (NCU's). In fact the bridges, gateways and NCU are based on identical hardware and are called Network Units (NU's). Twentenet has been chosen for the physical medium and the medium access control because it is superior to Ethernet with respect to noise immunity, real-time facilities, transmission speed, safety and coding (PR4).

Because all the NU's use the same hardware components, the differences between a NCU, Bridge or Gateway are mainly in the software. In figure 1 a possible NCU configuration is shown. The Twentenet Controller performs the Data Link Protocol and is connected via a coupler (Physical Layer) to the coaxial cable. The NCU contains software for the functions and protocols that correspond to the OSI documents. To minimize software development on handlers have still to be developed for operating systems such as UNIX (AT&T), Versados (Motorola), RSX and VMS (DEC). The File Transfer Protocol, the Virtual Terminal Protocol and other Application Layer facilities will possibly be implemented in Modular Pascal to achieve "easy" portability over a range of systems.

Twentenet [5] is a bi-directional communication system based on a coaxial cable with a maximum length of 2 km. The data speed aimed at is 16 Mbps, with a maximum of 255 stations connected to a cable segment. The access method is based on Priority based Carrier Sense Multiple Access with Collision Detection. In Twentenet access conflicts are resolved in a deterministic way, thus providing real-time behaviour. This should be contrasted with the binary exponential backoff approach of Ethernet.

ON THE OSI IMPLEMENTATION

In general a structured design is based on a layered approach. Fundamental work on network architectures has been done by the International Standards Organization (ISO). The Basic Reference Model of Open System Interconnection (OSI DIS 7498) defines a seven- layered view of a network architecture for the interconnection of various computer systems. For each layer a Service Definition and a Protocol Specification are being developed. Most layers have been described informally and have attained the status of a Draft International Standard (DIS) or a Draft Proposal (DP). New developments, such as the connectionless communication service and the IEEE 802 LAN standards, necessitate modification and revision of the lower layers; these developments are influencing LAN designs.

A model for layer services is described in the Service Conventions, in the annex to the Transport Service Definition (DIS 8072). For each layer interaction between a service-user and a service-provider, four types of Service Primitives (Sp's) are introduced: request, indication, response and confirm. Transport entities may request services from the Network Service Provider at a Network Service Access Point (N-SAP) using N-Sp's. Likewise, the Session Layer may request a Transport Service via T-SAP's using T-Sp's. In DIS 8072 the sequences of Sp's at a single SAP are defined by both state transition diagrams and in table representations. The Sp's play a role only within a single OSI system, whereas for the interaction between peer entities Protocol Data Units (PDU's) are used. PDU's consist of Protocol Control Information (PCI) and user data.

The Protocol Specifications specify protocols in natural language and in Extended Finite State Tables (FST's). For the Transport Layer (DIS 8073) five protocol classes are specified (class 0 up to 4), each with different error behaviour in relation to user requirements. Even for the very simple class 0 protocol the FST becomes rather complex, making a structured design based on the Service Definitions and Protocol Specifications complicated. What we are looking for is a way of decomposing OSI protocols.

In DIS 7498 a (N)-Protocol is described as: "A set of formats and rules (syntactic and semantic) which determines the communication behaviour of (N)-entities in the performance of (N)-functions".

The formats describe the data structure (bits and octets) of PDU's. The rules describe the sequence of PDU's and Sp's. The rule part of the protocol specifications is described using FST's, which we shall refer to as Protocol Machine (PM).

A Transport entity communicates with its Transport Service Provider via one or more N-SAP's using N-Sp's and with its peer entity using T-PDU's. To achieve a uniform treatment of Sp-calls we propose that a PDU should be associated with every Sp-call and therefore we introduce (Sp,PDU) pairs as interaction signals. If Sp-calls deal only with subsystem interaction then an Empty PDU is provided. It is clear that an Empty (N)-PDU cannot participate in the (N)-peer-to- peer protocol. Because (N)-PDU's are received by the (N-1)-layer as (N-1)-Service Data Units (SDU's), (Sp,SDU) pairs and Empty SDU's are introduced as well.

From the Transport Protocol FST it can be deduced that as well Sp's as PDU's are used as input events for the PM. The Sp's can be divided into Network Sp's (N-Sp's) and Transport Sp's (T-Sp's). The N-Sp's describe the interaction between the Transport Layer and the Network layer while the T-Sp's describe the interaction between the Session layer and the Transport layer. Furthermore, the PDU's describe the interaction between Transport entities.

With regard to the states of the same FST, these may also be divided into a part relating to Network Connections and a part relating to Transport Connections. This sub-division of the Transport Protocol table is shown in figure 2.

```
         |   States      :  States
Input    | N-Connection  : T-Connections
---------+--------------------------------
T-Sp's   |               :
........ |...............................
N-Sp's   |               :
T-PDU's  |               :
```

Figure 2.

This observation suggests a decomposition of an (N)-PM into two parts: one describing the interaction between the (N+1)-layer and the (N)-layer and one describing the interaction between the (N)-layer and the (N-1)-layer.

In analogy to the theory of sequential machines [6] we define an (N)-PM P_n as a 5-tuple: $P_n = (S_n, I_n, O_n, d_n, g_n)$.

In this definition S is the set of states, I the set of inputs and O the set of outputs with d:SxI->S, the transition function and g:SxI->O the output function,

with I_n={((N)-Sp-in,(N)-SDU-in)} U {((N-1)-Sp-in,(N)-PDU-in)}
O_n={((N)-Sp-out,(N)-SDU-out)} U {((N-1)-Sp-out,(N)-PDU-out)}.

This protocol machine is the PM as defined in the FST's. In this definition we make use of (Sp,PDU) and (Sp,SDU) pairs. This PM may be decomposed into an (N)-Subsystem PM P_{sn} and an (N)-Layer PM P_{ln}.

P_{sn} is defined as: $P_{sn}=(S_{sn}, I_{sn}, O_{sn}, d_{sn}, g_{sn})$

with I_{sn}={((N)-Sp-in,(N)-SDU-in)} U {(N)-PDU-in)}
O_{sn}={((N)-Sp-out,(N)-SDU-out)} U {(N)-PDU-out)}

and P_{ln} is defined as: $P_{ln}=(S_{ln}, I_{ln}, O_{ln}, d_{ln}, g_{ln})$

with I_{ln}={((N-1)-Sp-in,(N)-PDU-in)} U {(N)-PDU-out}
O_{ln}={((N-1)-Sp-out,(N)-PDU-out)} U {(N)-PDU-in}.

We now define a decomposition operator [] which is called "join":

A PM P_n is a join of two PM's P_1 and P_2 iff P_1 and P_2 can be described as mutually asynchronized PM's, whereby P_1 interacts with the (N+1)-layer and P_2 interacts with the (N-1)-layer,

notation: $P_n(x,y) = P_1(x)\ [z]\ P_2(y)$.

The SAP-manager is part of an OSI manager and contains the layer PM's and the queues and forms part of the kernel of the network software.

The overhead associated with the layered approach of OSI may reduce the performance of a network considerably. Sp-call overhead can be reduced by constructing a fast SAP manager and OSI memory manager. For special, local applications it may be useful to implement a by-pass around the application dependent layers. PCI-overhead in a PDU is more tricky to solve. The whole PDU must be sent if applications communicate with other OSI systems. It is to be expected that most of the communication will be limited to the local network. For this sort of application the number of different PDU's used may be restricted. Once the number of different PDU headers are determined a more efficient coding may be introduced. The translation to full OSI PDU's should only be done in bridges or gateways when a PDU has to be sent to or received from an OSI system outside the local network. Inside ZWOLAN a coded form of PDU headers will be used, thus reducing the OSI overhead substantially.

CONCLUSION

There is a definite need for LAN's with an open structure. The NU's simplify the coupling of computers to the network. Because the NU contains most of the network software, little software needs to be developed for the application computers. Modular Pascal has proved to be an effective language for systems software development.

OSI implementations introduce problems (complexity and overhead) but as a result one has to pay attention to various aspects of network design. The OSI Reference Model may be used as basis for a stuctured design, but does not lead to a straightforward implementation model. An implementation model has been derived from a decompositional analysis of OSI protocols. Parts of the OSI manager have already been realized in Modular Pascal. The protocol overhead can be reduced by defining efficient protocol headers and by developing a mapping algorithm which maps the new data units onto OSI Protocol Data Units.

Acknowledgement:

Part of the work described in this paper is financed by the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

REFERENCES

[1] A.J de Raaf, A versatile data handling system for nuclear physics experiments based on PDP 11/03 micro-computers, Nucl. Instr. & Meth. 163 (1979) 313-324.

[2] A.J. de Raaf, A data storage and communication system for up to sixteen multichannel analysers, J.Phys.E:Sci.Instrum.,Vol. 16 (1983) 780-785.

[3] C. Bron, Modular Pascal Language Definition, Twente University of Technology (Netherlands), THT-Memorandum, INF-82-10 (1982).

[4] T.J. Rossingh and J.E. Rooda, ROSKIT-A real-time Operating System Kit, Twente University of Technology (Netherlands), Internal report, WTB-41 (1984).

[5] I.G. Niemegeers and C.A. Vissers, Twentenet-A Local Area Network, Twente University of Technology (Netherlands), Internal report, No. 1251.1891 (1982).

[6] J. Hartmanis and R.E. Stearns, Algebraic structure theory of sequential machines, Prentice Hall inc., Englewood Cliffs, (1966).