

Reprint

Design, Implementation and Performance Analysis of an ETHERNET to LION Gateway

T. Antonakopoulos, J. Koutsonikos, V. Makios

NATO ASI Series F: Computer and System Sciences

VOLUME F72, "HIGH -CAPACITY LOCAL AND METROPOLITAN AREA NETWORKS"

1990, pp. 455-469 (ISBN 3-540-53767-8)

Copyright Notice: This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted or mass reproduced without the explicit permission of the copyright holder.

Design, Implementation and Performance Analysis of an ETHERNET to LION Gateway

T. Antonakopoulos, J. Koutsonikos and V. Makios

Laboratory of Electromagnetics, University of Patras, Patras, Greece.

Abstract: The introduction of the new High-Speed LANs has not only to take into account the integration of the new services but also to provide powerful interconnections with traditional LANs like the ETHERNET. This paper addresses the ETHERNET internetworking solution adopted for a Local Integrated Optical Network (LION). The architecture of this internetworking solution is described, both in hardware and software terms, and the real time requirements are highlighted, describing the implemented system. Finally, the Gateway performance analysis is given and the selection of the implementation parameters becomes obvious.

Keywords: High-Speed LANs, ETHERNET, Internetworking.

I. Introduction

Integrated Services Local Networks (ISLNs) connect usually a few hundred of nodes, supporting thousands of individual users and integrate different kinds of traffic, encompassing data, voice and images [1]. The interconnection needs between these systems become essential and the requirement for economic and flexible access to external homogeneous networks, through Bridges (BRG) and heterogeneous networks, through Gateways (GTW), becomes critical as far as the network performance is concerned [2].

LION [3] is intended to integrate the above mentioned services covering areas of diameter in the range of a few hundred meters up to ten kilometers. Thus, a two-level architecture has been adopted and a new high-performance medium access protocol has been developed, based on the so-called hybrid-switching technique, which provides both circuit and packet switching capabilities.

The stream traffic is supported through a transparent "bearer" service at the MAC sublayer of the International Standards Organization (ISO) model for Open Systems Interconnection (OSI) and the I.450, I.451 ISDN protocols. The packet traffic is supported by an OSI protocol profile, as following: the 2a OSI sublayer (MAC) is provided through a specially developed Access Protocol and hardware based on the Hybrid Switching concept, due to the integration of stream and packet

traffic and the expected workload; the 2b OSI sublayer follows the LLC type1 protocol; the Network Layer is based on the inactive Internet Protocol (IP) for the LION users and the active IP for the external users; finally, the Transport Layer follows the Transport Class 4 Protocol (TR4). The impact of the internetworking problems is mirrored on the protocol profile choice since the network is considered as a "distributed end system" resulting to a connectionless approach up to layer 3 (network layer). The implemented ETHERNET to LION Gateway (ELGTW) provides access to the network layer and supports the required address transformation and relay functions.

The Gateway's hardware structure is described in section II, where the need for the development of a special interface board is highlighted. In section III, the used communication software structure is explained along with the system scheduler, emphasizing the way the information exchange is performed under the required time constraints. In the last section, the Gateway's performance analysis is given, based on simulation methods. The implementation parameters, like the buffers' length, CPU allocation scheme and the priorities of the various tasks are estimated and the way the Gateway was implemented becomes obvious.

II. The Gateway Hardware Structure

The internetworking problem requires a careful choice between a protocol profile, matching that of the external network and a protocol profile, conceived to support efficiently the internal communications. The critical parameters are specified by traffic requirements, especially in view of the huge workload of an ISLN. The LION node architecture is shown in Fig.1, including the ETHERNET Gateway, based on the network node multiprocessing architecture. This structure offers several benefits in terms of modularity, flexibility and capability to adapt to the specific characteristics of the interconnected networks. The ETHERNET-LION Gateway takes advantage of the adopted communication protocol profile, based on the ISO 8473/AD1 Internet Protocol [4]. End-systems exchange data units in a connectionless mode. The crossed subnetworks are only requested to provide a data pipeline, on which data are routed independently. The convergence between ISO IP and the underlying LLC1 is a simple one-to-one primitive mapping. In this interconnection scheme, the Gateway is not burdened, but it has only to perform addressing scheme adaptation, encapsulation-decapsulation functions and buffering.

The hardware implementation of the Gateway is shown in Fig. 2 [6]. It includes an ETHERNET Controller and an Interface Module. The ETHERNET Controller, which is based on the MVME330 board of Motorola Inc., includes a MC68000 CPU at 10 MHz, Dynamic RAM of 512 Kbytes, PROM of 64 Kbytes, where the Operating System Kernel and the communication protocol software are loaded, and the LANCE chip, which is the hardware implementation of the CSMA/CD access protocol.

The Interface Module (IM) has been developed to accommodate the following functions:

- i) Intergateway arbitration mechanism for dynamic bus allocation and bus interconnection,
- ii) The needed control mechanisms (special interrupt handling for the Gateway and the network node communications),
- iii) Dual port RAM for the implementation of a mailbox intercommunication system between the Gateway and the network node,
- iv) FIFO memories (receive and transmit), which provide fully independent operation of the Gateway and the network node, and
- v) Additional ROM requirements for the Gateway software.

The communication between the Gateway and the LION node comes through the use of the Data Transfer Control Block (DTCB) mechanism, a mailbox like mechanism. The DTCB contains information of the transferring packet, like the starting address, the packet length, the packet status and quality of service parameters. The Controller is polling the flag of the receive DTCB,

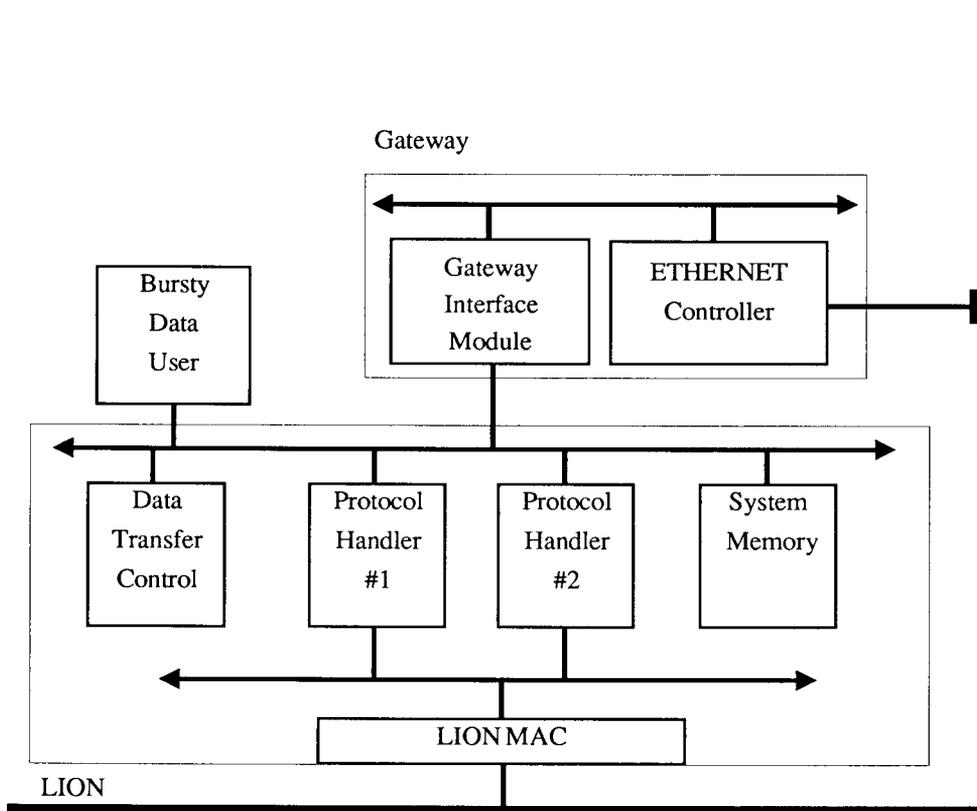


Fig. 1. The LION Node Architecture.

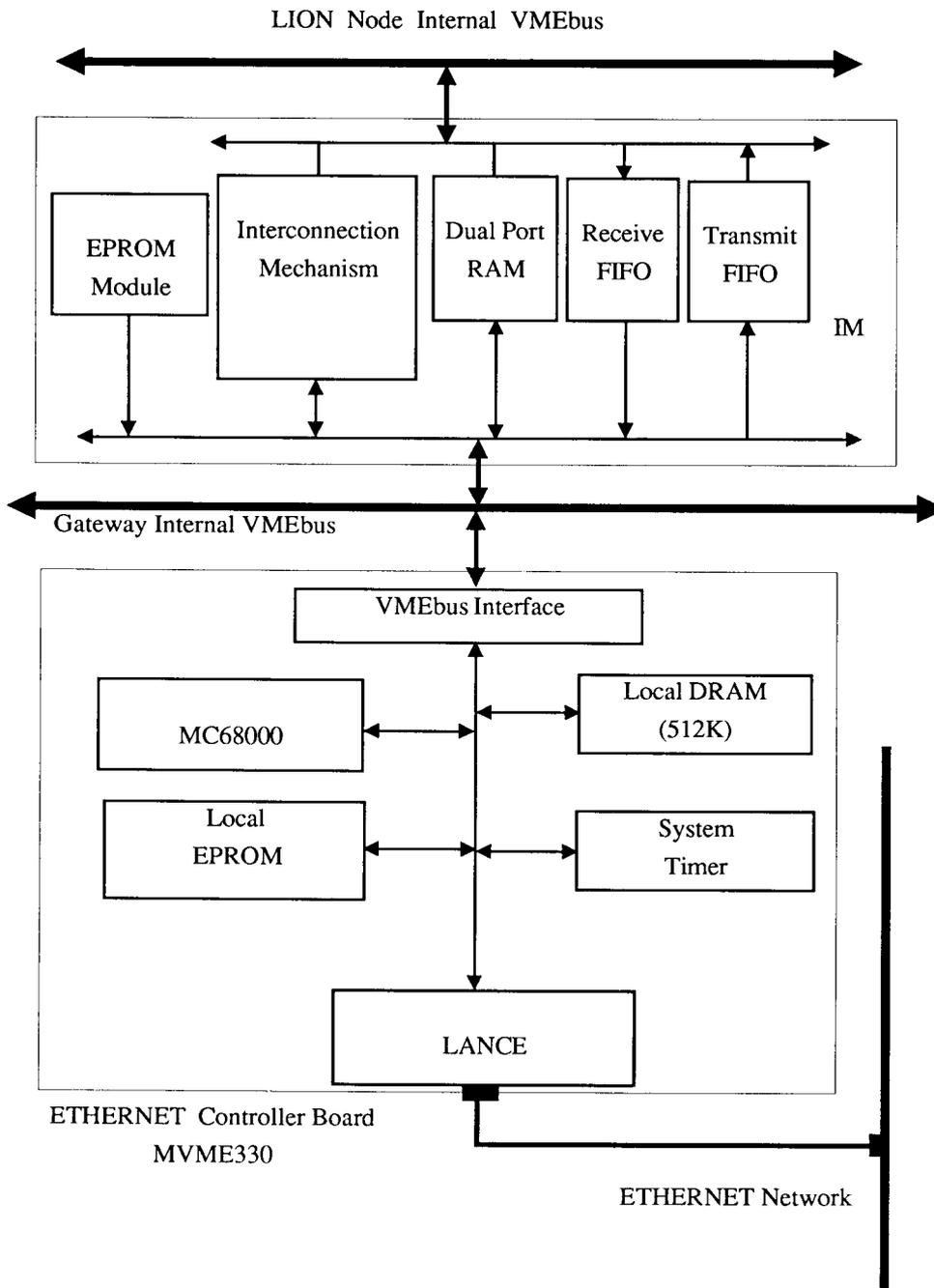


Fig. 2. The Gateway Structure.

which indicates, when set, that the LION node has a packet to send to the Gateway, then it sends an interrupt to the node and, when the transfer is completed, it copies the packet data in the local memory. When the Gateway has a packet to send to the node, the Controller copies the packet data to the transmit FIFO of the Interface Module, informs the DTCB with the packet length and the packet address, sets the flag of the transmit DTCB and sends an interrupt to inform the node to start the DMA transfer of the packet.

The transmission of a packet from the ETHERNET Controller to the LION node, using the FIFO memories of the Interface Module, is shown in Fig.3. The Controller stores the packet in the transmit FIFO and generates an interrupt to the node, using the interbus connection mechanism. The node recognizes the interrupt and receives the packet. The signal FIFO-E is the empty flag of the transmit FIFO.

In Fig.4 the transmission of a packet from the LION node to the ETHERNET Controller is shown. The node informs the respective DTCB that wants to transfer a packet to the Gateway and the Gateway generates an interrupt when the receive FIFO is available. Handling that interrupt, the node stores the packet to the FIFO and the Gateway's CPU, following a polling scheme, recognizes the end of packet storage and transfers the packet to the internal RAM.

In order to increase the performance of the Gateway and minimize the impact of the Gateway's workload to the LION node, a two VMEbus structure is used. The Intergateway arbitration mechanism in the Interface Module allows the dynamic bus interconnection following the DTCB mechanism requirements. When the Gateway wants to generate an interrupt to the node, both in receive and transmit mode, the Gateway's CPU has to use the system's interrupter which is located in the system controller. During that period, the Intergateway arbitration mechanism connects the two buses, the IM's arbiter is disabled and the system becomes a single bus system. After the interrupt generation, the system becomes again a two bus system, allowing simultaneous transfer in the two buses. This mechanism is transparent for the other node users as well as the node itself. This organization can be used for multiple gateways implementation and for a universal gateway structure. In Fig.5 the operation of the Intergateway arbitration mechanism is shown.

The communication between the CPU and the LANCE chip is attained through dynamic receive and transmit descriptor rings. When a packet is ready to be transmitted to the ETHERNET network, the CPU moves the packet to a free transmit buffer, informs the respective transmit message descriptor with the length of the packet and informs the LANCE for the existence of a ready for transmission packet. The transmission of a packet from the local memory to the ETHERNET network is shown in Fig.6. The CPU informs the LANCE with the parameters of the transmission and the LANCE gains the control of the internal bus of the MVME330 board. Then it transmits the packet in bursts of 16 bytes (16-bit transfer mode), using the internal DMA capability.

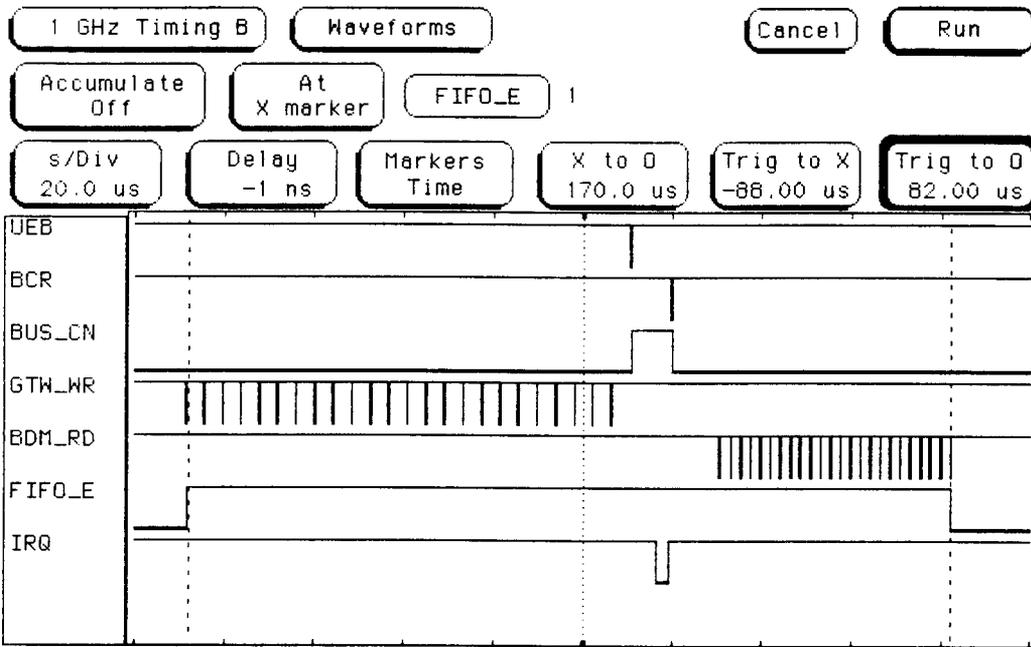


Fig. 3. ETHERNET to LION Transmission.

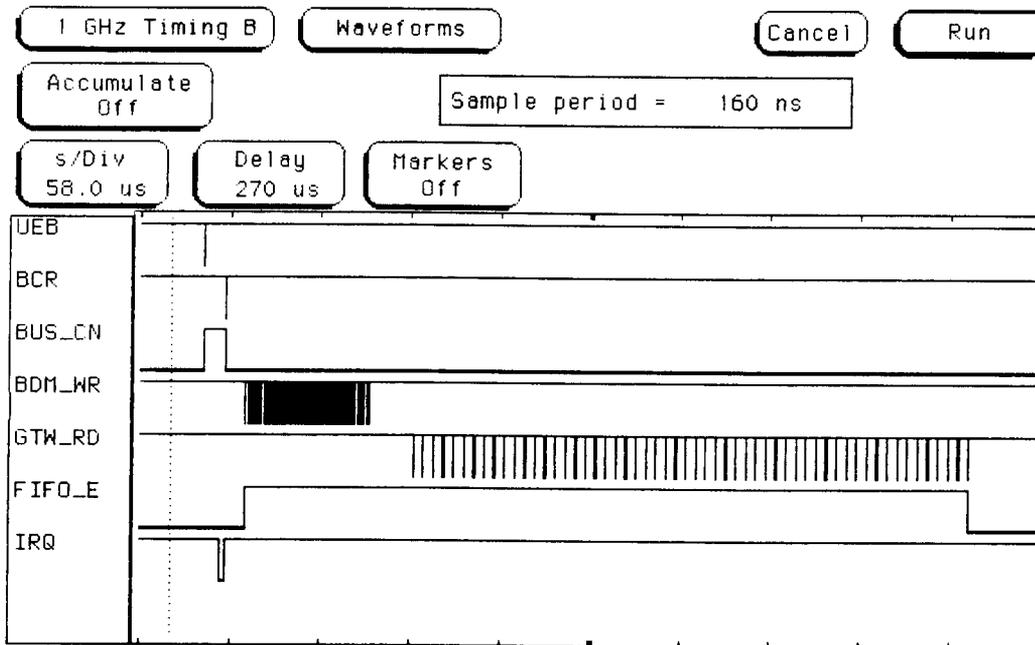


Fig. 4. LION to ETHERNET Transmission.

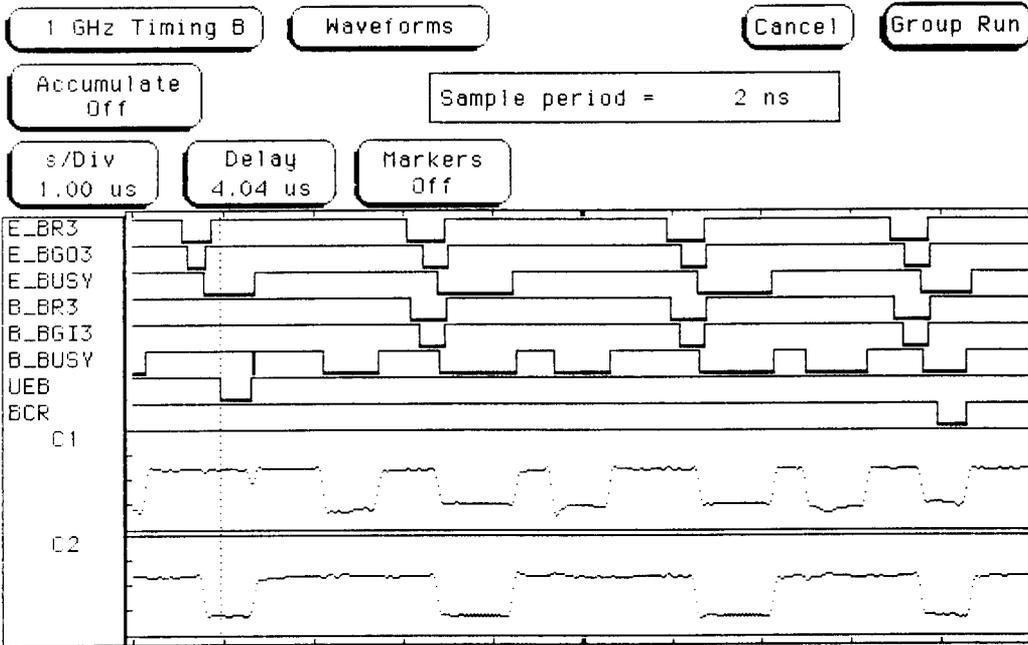


Fig. 5. The Intergateway Arbitration Mechanism.

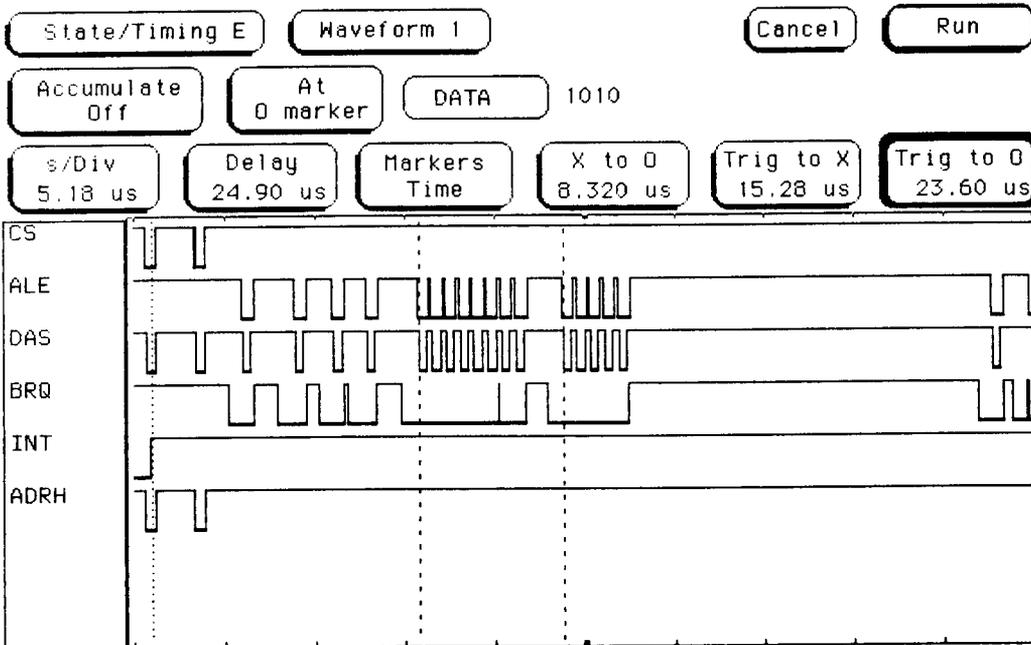


Fig. 6. ETHERNET Packet Transmission.

III. The Communication Software

The ETHERNET to LION interconnection takes advantage of the adopted communication protocol profile, based on ISO 8473/AD1 Internet Protocol.

The Gateway's software consists of three parts, namely:

- i) A modified version of a Real-Time Multitasking Operating System,
- ii) The Scheduler, a specially developed low-level software, and
- iii) The Communication Protocols Software.

As the System's Operating System, the RMS68K Kernel of Motorola, is used. The installed Kernel is a modified version of the original operating system. The main modifications concern the system's Scheduler to allow multiple CPU allocation schemes. This will become clearer in the next section, where different CPU allocation schemes will be used, to measure the Gateway's performance.

The Scheduler's drivers provide the required communication primitive and data format transformation with the external world (the LION node), during initialization for identity, quality of service and various procedural functions, required by the LION global software architecture.

The ETHERNET interface is handled and managed by a developed Kernel function, integrated in the Scheduler module, in order to provide modular and efficient handling. The responsibilities of the Kernel are:

- i) Initialization and testing of the ETHERNET interface,
- ii) Local resources management, related to the ETHERNET interface, and
- iii) Interface to the software, at the level of LLC1 sublayer, that implements the required primitives.

The Gateway follows the same protocol profile with the LION network up to the network layer. For that purpose, the XTSLIT software is used [6], in conjunction with a specially developed software, called ENV.

The XTSLIT, which is shown in Fig.7, provides the needed environment for the processing of the packets by the protocol profile used in LION. The XTSLIT besides the LLC1 and IP protocols includes the following:

- AXI: an Operating System independent module, providing the Access Interface between the external world (e.g. the LION node) and the internal software modules (e.g. IP and Layer Management),
- GBUF: an Operating System independent module, that provides the management of the buffers inside the XTSLIT,
- GTIM: an Operating System independent module, that provides the time management functions.

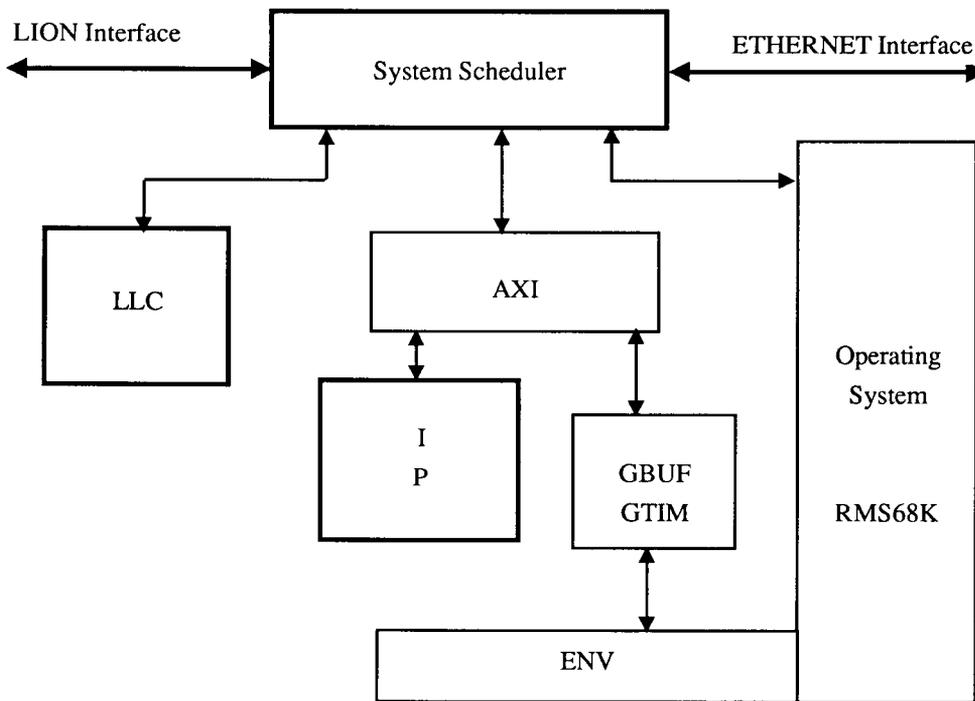


Fig. 7. Gateway's Software Structure.

The Operating System (OS) communication to the system tasks is carried out through the Environment module (ENV) in the form of primitives exchanged between each task and the OS. The responsibilities of the ENV are:

- i) Interface for subsystem access by user entities and entity access by the subsystems.
- ii) Isolation of the Operating System and hardware environment, providing the required format transformation, drivers, utilities and monitoring interfaces.
- iii) Management and allocation of the system resources, especially buffers, queues and timers.
- iv) Intercommunication between the various system tasks, and
- v) User interface in the form of C language functions for transmitting and receiving the primitives of the various subsystems and the Real-time Multitasking Operating System.

The communication between the LLC and the IP, handled by the Scheduler, concludes the proper transformation of the packets, in order to be able to be processed in these protocols. Each processing state uses a part of the buffering scheme which is shown in Fig.8. The Operating System allows the dynamic allocation of the local memory, taking into account the workload require-

ments. Using that technique, a better memory utilization is achieved and the system performance is increased due to minimization of data flow congestions and buffer overflow.

The communication of the Scheduler with the protocol processing modules is performed through message exchanges between the tasks implementing the modules. The protocol processing modules are in "dormant" state until they receive a message to process a packet, when they "wake-up", perform the packet processing, send it back to the Scheduler and are put again in the "dormant" state, awaiting a new message. The Scheduler receives messages from the protocol processing modules, but it is never put in the "dormant" state, because it needs to poll the receive DTCB and the receive message descriptor ring of the LANCE chip for possible packets received either from the LION node or from the ETHERNET network. This is accomplished by using a "read_event" function, instead of the "get_event" function used from the protocol processing modules.

IV. Performance Analysis

The Gateway performance analysis is based on simulation methods, using the Gateway model which is shown in Fig.8. The system has three operative tasks, the Scheduler, the IP task and the LLC task. The Scheduler is always in "running" state and manages the packet transfer between the LION interface, the ETHERNET interface and the other two tasks which process the packets, following the respective algorithm. The IP and LLC tasks are in "running" state when they have a packet to process, otherwise they are in "dormant" state, waiting a signal from the scheduler to "wake them up".

The LION interface has two Single Packet FIFO Buffers for reception and transmission respectively. The ETHERNET interface has two Buffer Rings which can store up to 64 packets for each direction. The Scheduler manages these buffers following a round-robin scheme. For each protocol processing task, there are one input and one output buffer. The IP Input Buffer has a variable length depending upon the traffic workload, while the IP Single Packet Output Buffer (IP SPOB) can handle only one packet at a time. The same holds for the buffer of the LLC. This dynamic buffering scheme is achieved using the respective utilities of the RMS68K Operating System. The total available memory is about 400 Kbytes both for the static and the dynamic allocated buffers.

For simulation purposes, the following assumptions were introduced:

- The packet interarrival time follows an exponential distribution (not necessarily different for the two networks).
- The packet length has an exponential distribution and the minimum and the maximum values follow the ETHERNET standard.
- The CPU is allocated between the different tasks using a fixed length allocation scheme. If IP

or LLC have no more packets in their input buffers, the CPU is allocated to the next available task.

- The time required by the Kernel's scheduler is assumed to be negligible. Also for DRAM refreshing, bus arbitration and DTCB handling the time interval is negligible. This is a consequence of our abstract view of the system behavior.

- The network is supposed to be in steady state.

The Gateway performance will be given in terms of message delay, mean and maximum total buffer length, while the simulation parameters will be the CPU allocation scheme, the distribution of the total workload between the two networks, the priorities between the packets of the dif-

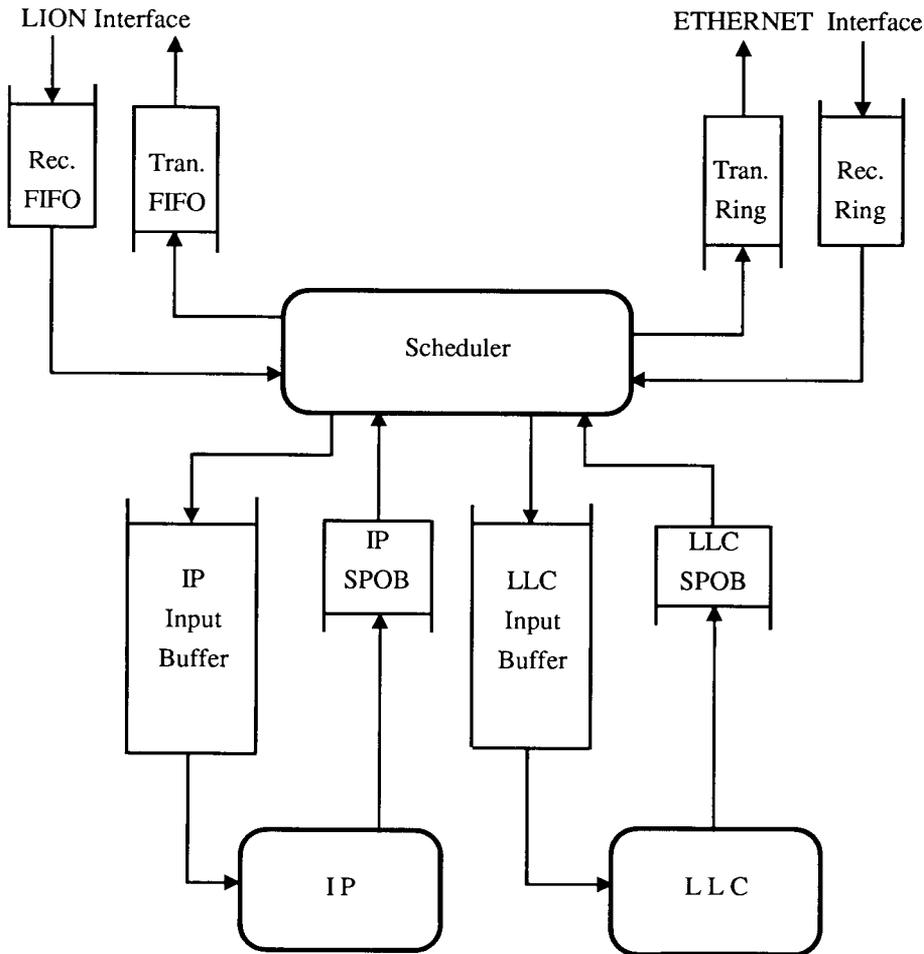


Fig. 8. The Gateway Simulation Model.

ferent networks to the same task and the task priorities. As a reference we will take the case where the load is equally distributed, the tasks have the same priorities and the CPU is allocated for one time slot to each active task at each round.

In the following figures, "case 1" represents the reference case, "case 2" represents the case where the IP task has the CPU for five continuing time slots in each round, the LLC task has the CPU for two time slots and one time slot is available for the scheduler. In "case 3" the tasks follow a nonpreemptive priority discipline. The IP task has the maximum priority, i.e. the CPU allocates the required time slots until its input buffer becomes empty. The LLC has the intermediate priority and the scheduler the lowest priority. The protocol processing time is constant and it is equal to 25 msec for the IP and 10 msec for the LLC (These values are derived from the implemented system).

In Fig.9 the throughput/delay characteristics are shown. The delay for "case 2" and "case 3" are almost the same while the delay performance for "case 1" is not acceptable. The maximum processing power of the Gateway is 28 packets/sec and it is approximated with the last two cases. The mean packet length has been considered to be 1 Kbyte long, the maximum length 1.5 Kbytes while the minimum length is 64 bytes.

In Fig.10 and Fig.11, the mean length and the maximum length of the total buffer are shown while Fig. 12 shows the packet rejection rate due to the Gateway's buffer overflow. In "case 1", the buffer overflows in medium traffic load and the rejection rate becomes unacceptable, while in the other two cases the buffer overflows when the load reaches the maximum processing power of the Gateway and the rejection rate remains low. In the developed system, "case 3" was decided to be implemented because of its simplicity and better system performance.

In order to increase the system performance, new software for IP and LLC has to be developed or a faster CPU has to be used.

V. Conclusions

The implementation of an ETHERNET to LION Gateway is presented. The hardware structure as long as the adopted communication software are described, giving emphasis to the supported ISO model. The simulation analysis gives the best values for the implementation parameters and estimates the Gateway's performance both in packet delay and required memory.

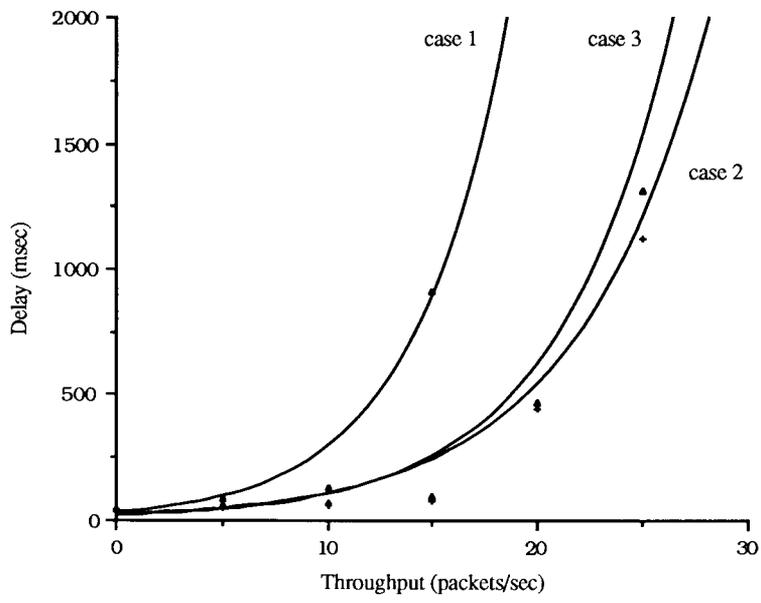


Fig. 9. Throughput/Delay Characteristics.

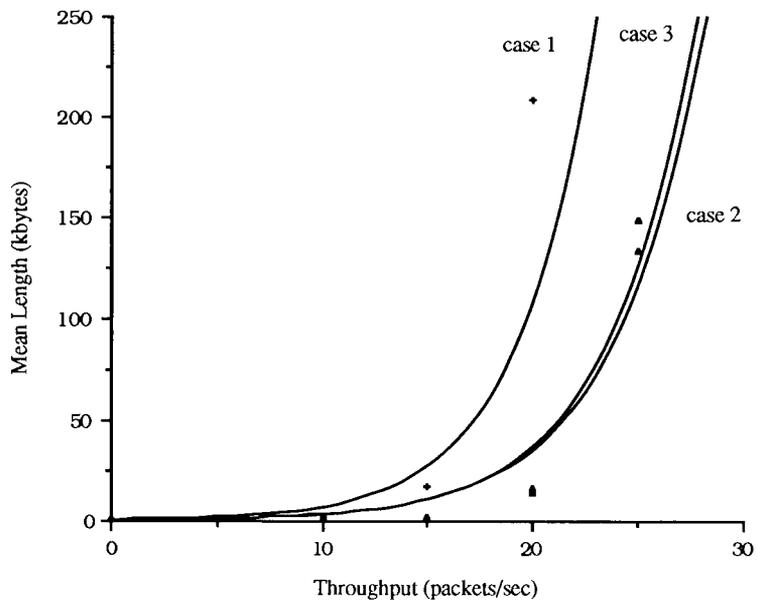


Fig. 10. Throughput/Mean Buffer Length Characteristics.

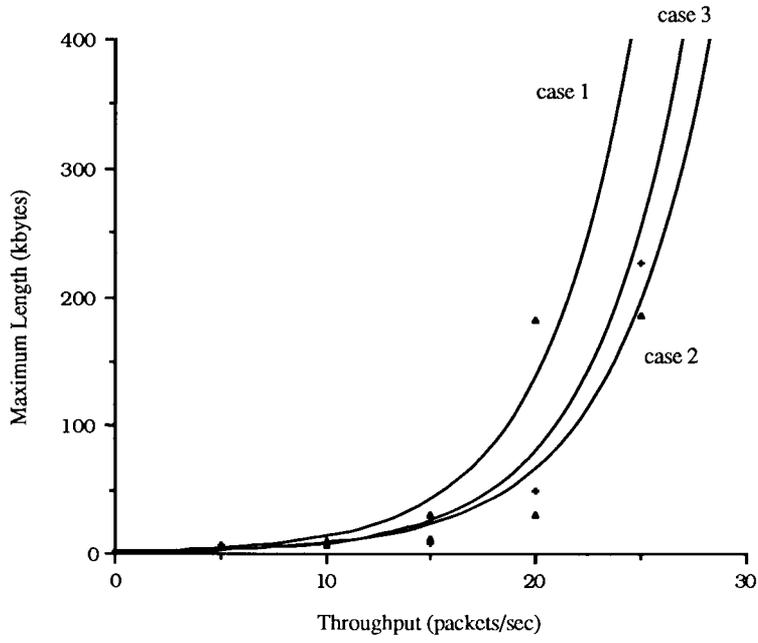


Fig. 11. Throughput/Maximum Buffer Length Characteristics.

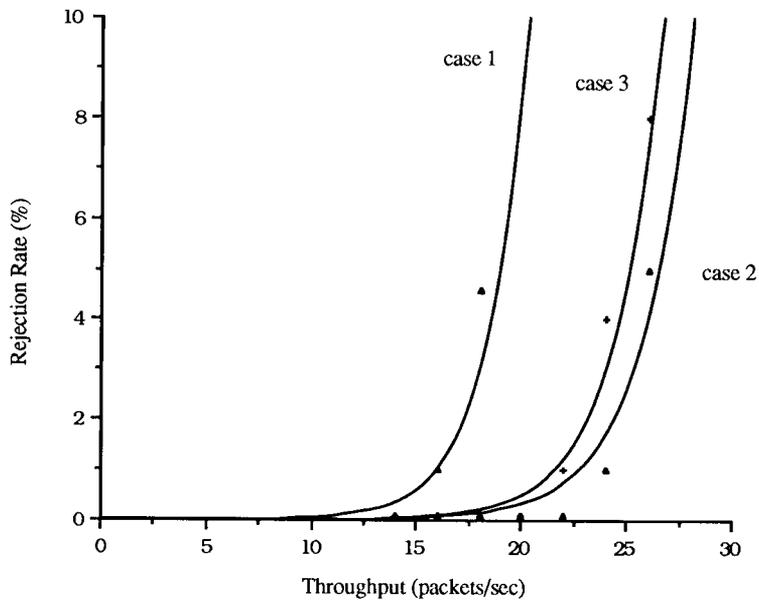


Fig. 12. Throughput/Packet Rejection Rate Characteristics.

Acknowledgement

The authors would like to thank Mr. C. Stavroulopoulos for his contribution to the development of the simulation program. This work was carried out under the partial financial support of the EEC (ESPRIT project 169, LION) as a subcontract with ALCATEL-TITN, France.

References

1. N. Corsi, A. Luvison and A. Moncalvo: "Respectives on Wideband Local Area Communication Networks", Proc. IEEE ICC '84, May 1984.
2. D.Roffinella, C.Trincherro, G.Freschi: "Interworking Solutions for a Two-Level Integrated Services Local Area Network",IEEE Journal on Selected Areas in Communications, Vol.SAC-5, No.9, pp.1444-1453, December 1987.
3. A.Luvison, G.Roulet, F.Toft: "The LION project: A status report",in Proc. 4th Annual ESPRIT Conf., Brussels, Belgium, Sept.1987, pp.1477-1489.
4. "General Specifications IP and TR4 Products", ESPRIT project 169, TITN Internal Report, March 1988.
5. J.Koutsonikos, T.Antonakopoulos, V.Pallios, V.Makios: "Analysis and Implementation of the Gateway between the ETHERNET and a High-Speed Multiservice LAN", The 8th International Symposium on Applied Informatics, (IASTED) Innsbruck, Austria, February 1990.
6. "The LION to ETHERNET Gateway", ESPRIT project 169, University of Patras, Concise Technical Report, June 1989.