

Reprint

A Dynamic Buffer Structure for LAN Gateways

J. Koutsonikos, T. Antonakopoulos and V. Makios

The 21st IEEE Communication Theory Workshop

RHODES, GREECE, JUNE 1991

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.

"A Dynamic Buffer Structure for LAN Gateways "

J. Koutsonikos, T. Antonakopoulos and V. Makios
Laboratory of Electromagnetics, University of Patras
26500 Patras, Greece

Abstract : In this paper, we propose a new dynamic buffer structure for use in LAN gateways. This structure is called dynamic because it uses the statistic behaviour of the attached networks, the LAN and the Backbone, to allocate the gateway resources. The method utilizes the differences between the protocol profiles of the two networks and optimizes the implementation parameters for better system performance. Using the simulation method, the operation of the proposed structure is analyzed and the influence of the system parameters is indicated. The performance of that structure is also compared with the performance of a commonly used buffer structure and, as the results demonstrate, an enhanced performance is derived.

I. INTRODUCTION

The technology of computer networking is one of the basic elements of today's state-of-the-art distributed processing evolution. The computer networking combines the new advances of communication technology and the demand of supporting new services and applications in a distributed environment, with the need to use the existing processing systems. This interworking has to be done at reasonable cost and has to fulfil the high throughput requirements of the interworking systems. The problem of interworking becomes very complicated when heterogeneous networks are concerned and various levels of interworking are implemented [1], [2].

In this paper, it is considered the internal structure of an interworking device, which, for the sake of simplicity, will be called gateway. (It is known from the literature that some terms of the interworking terminology have been used for different and sometimes contradictory purposes. The term 'gateway' has been applied for simple interworking models [3] as well as for the Application Layer Relay functions [1] and it has been accepted as a general term). Usually the gateways are considered to have a straight-forward structure and the flow of information inside a gateway is deterministic and independent of the traffic conditions [3], [4], [5], [6]. Each packet arrives from the source LAN and is buffered until the internal server is available to add the appropriate information. Then the packet is transferred to the output queue and waits to take the access of transmission into the destination network. The most important parameters of the performance of a gateway are the imposed delay and the required buffer space. These factors are in contradiction with the need to minimize the cost, keeping the 'quality of service' in acceptable limits. The cost increases as the required throughput increases, because more powerful processors are used and more buffer space is needed, to keep the packet rejection rate low.

In the next section, the proposed buffer structure is described and the traffic parameters, which affect the flow of information inside the gateway, are discussed. In Section III, the performance of this structure is highlighted using bursty type of traffic. The results of this analysis are compared with a simple buffer model and the advantages of the dynamic buffer structure are indicated. Section IV summarizes this discussion and some topics for further work are proposed.

II. THE BUFFER STRUCTURE OF THE GATEWAY

The main purpose of a High Speed Backbone Network is to interconnect various Local Area Networks (LANs) and high processing computing systems,

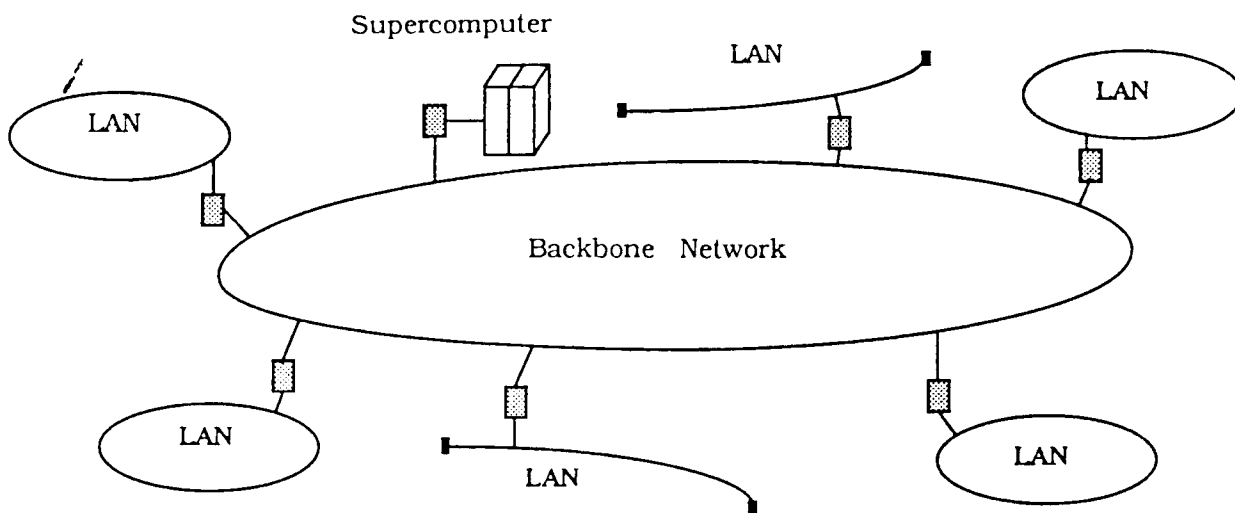


Fig.1. The backbone network architecture

like supercomputers, and to formulate the necessary communication environment for fast and reliable information exchange. In Fig. 1 the topology of such a network is depicted. In a backbone network different interworking aspects are involved, concerning the connection of homogeneous or heterogeneous LANs, the interfacing of the supercomputers, the connection to MANs, etc. In the following analysis, the interworking between homogeneous LANs will be considered, while the traffic of the other sources will be assumed that occupies a portion of the available bandwidth.

The LAN gateways are considered to have a simple architecture, as it is shown in Fig. 2a. There is a single Receive Queue (RQ) and a single Transmit Queue (TQ) [5]. The RQ stores the received packets until the internal packet server is available to process and store them into the TQ.

The internal packet server performs the protocol processing, which concerns the protocol profile of the LAN and the backbone network, without any packet multiplexing. When two internal packet servers are available (as it

is shown in Fig. 2b), there is another logical queue, the Internal Queue (IQ). In this configuration, the first server, which implements the protocol profile of the connected LAN, processes and stores the packets into the IQ. Then the second packet server, which implements the protocol profile of the backbone network, is responsible to process and store the packets into the TQ. This architecture needs one processor and a time-sharing mechanism, in order to execute the various submodules of the protocol profile [6]. In this discussion, the processors which implement the two MAC interfaces are not concerned.

The traffic between the various interconnected LANs via the backbone network supports person-to-person communication, exchange of CAD/CAM data, file transfer, data base query, etc. This means that the traffic is consisting of packets generated in a bursty manner and by interactive traffic. The packets generated by the bursty traffic have the same destination for the duration of a burst and their multiplexing in the backbone level can decrease the required processing time and increase the system throughput. It is also known that for LANs interconnected in a backbone network (which is usually a High-Speed LAN), the connection devices, like the gateway, are much more likely to be the system bottleneck than the access protocol used in the backbone network [5]. In order to increase the gateway throughput without increasing its cost, the architecture shown in Fig. 2c is proposed.

A. The Proposed Structure

The proposed architecture uses more than one logical Internal Queues. Each IQ is devoted to a specific 'destination' LAN. When a packet arrives from the source LAN, it is stored into the RQ until the RQ-server processes it. During that process, the destination LAN is recognized and the packet is stored into the respective IQ. The IQ stored packets (called r-packets) are then multiplexed to formulate a new packet (called t-packet). The t-packets are processed by the IQ-server and stored in the TQ for transmission in the

backbone network when it has the right of transmission. It must be emphasized that a packet can be rejected due to buffer overflow, only when it is received from the interconnected network. In the internal transfers there is no possibility of rejection, since the transfers are logical, which means that the information field of the packet is stored to a specific physical memory and the header is processed in each stage. Each packet is associated with a 'message descriptor' which contains its length, the starting address and status bits. The TQ buffer handles each t-packet using the 'message descriptors' of the respective r-packets.

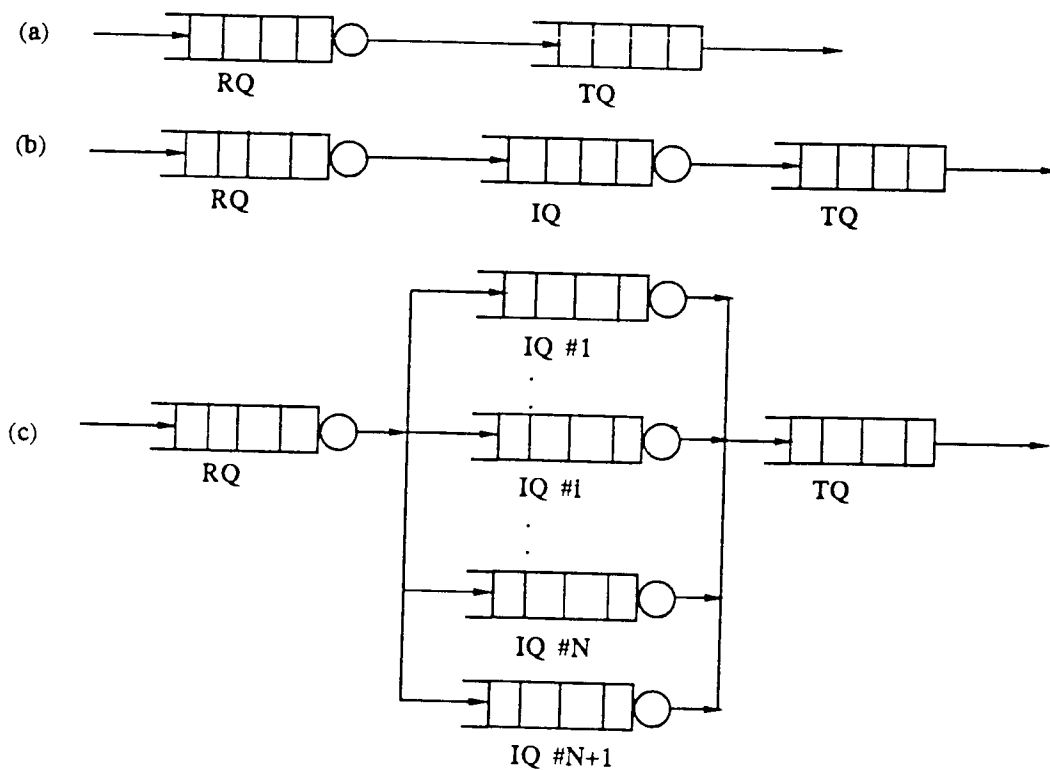


Fig.2. Various buffering schemes for gateways

B. The Structure's Parameters

The first approach to the problem of the number of IQ buffers is to use the same number (minus one) as the number of interconnected LANs, in order to have a dedicated IQ buffer for each of the possible destination LANs. The disadvantages of this approach are: i) it requires the knowledge of the number of the interconnected LANs, and ii) each time a new LAN is connected or a LAN is disconnected, the redefinition of the buffers has to be done. A modification of this approach is to include a 'learning capability' in the gateway. A type of 'learning capability' has already been used in the transparent spanning tree architecture [7]. In the buffering mechanism, the 'learning capability' of the gateway is the detection of the interconnected LANs using the destination address of each packet of the connected LAN (or the source address of each packet from the backbone network) and the allocation of a logical IQ buffer to that address. The disadvantage of that method is that it becomes very complicated as the number of the interconnected LANs increases and the required processing time degrades the throughput of the gateway.

In order to overcome these disadvantages and approach a more realistic model, the following architecture has been developed. There are $N+1$ predefined logical IQs. The N IQs are the dynamically allocated buffers, while the $(N+1)$ -IQ acts like 'bypass'. The gateway has the 'learning capability' in order to update an internal address map and uses the 'Statistic Window Method' to route the received r -packet to the appropriate IQ. Each one of the N IQs is allocated to a specific address temporarily and the allocation changes under the supervision of the 'Statistic Window' Handler. If the destination address of a packet doesn't match with any of the currently used destinations of the N dynamically allocated IQs, the packet is stored in the $(N+1)$ -IQ for further processing. In these N IQs, the t -packet formulation is performed using a number of r -packets, while in the $(N+1)$ -IQ each r -packet formulates a t -packet.

In order to route the r-packets to the IQ-buffer appropriately and to decide in which destination LAN an IQ buffer will be allocated, the 'Statistic Window' method is used. From the included 'learning capability', each time a new address is recognized, a Destination Counter for that address is generated and is inserted into the structure of the Destination Counters. These Destination Counters are used to determine the allocation of each IQ buffer to a specific LAN address. These counters indicate the number of address detection during a 'time window'. This 'time window' covers the last M-processed packets of the RQ buffer. The operation of the so called 'Statistic Window' method is conceptually represented in Fig. 3.

Suppose that there are T recognized destination addresses and the system has N available IQs for dynamic allocation. The T-counters form the 'window vector' V , which has the following format:

$$V_i = \{ n_{i,1}, n_{i,2}, \dots, n_{i,T} \} \quad (1)$$

where $n_{i,j}$ represents the value of the counter j in the time instant i.

When the service of a new packet is completed in the RQ-server, the 'time window' shifts one position and a new 'window vector' is generated. This 'time window' shift is represented by the Δ vector, where

$$\Delta_i = \{ m_{i,1}, m_{i,2}, \dots, m_{i,T} \} \quad (2)$$

The $m_{i,j}$ symbolizes the variation to the state of the j counter at the time of arrival i. The $m_{i,j}$ takes the values 1, 0 and -1. The value "1" is taken when the new packet is related with the j-counter, while the value "-1" is taken when the oldest packet of the 'time window' before the shift, was related with the j-counter. The $m_{i,j}$ takes the value "0" when there was no variation concerning the counter j or both the new packet and the oldest one had the same destination address, the address which is related with counter j.

This operation is described by the following equation:

$$\mathbf{V}_{i+1}[j] = \mathbf{V}_i[j] + \Delta_i[j], \quad j=[1,2,\dots,T-1,T] \quad (3)$$

As it was mentioned previously, the operation of the 'Statistic Window' and the allocation of the IQs is performed by the 'Statistic Window' Handler. This Handler is a software submodule in the gateway organization, which is called by the RQ-server when the processing of an r-packet is under completion (the 'Statistic Window Handler' is the last function in this processing stage). Suppose that the counter 1 was decremented by 1, the counter j was incremented by 1, as it is shown in Fig. 3, and that $n_{i,\min}$ is the lowest counter value allocated to an IQ buffer at time i . After the update of the Destination Counters, the buffer reallocation procedure begins. The Destination Counters are sorted in an incremental fashion and the value of the $n_{i+1,\min}$ is calculated. If two or more counters have the same value, the sorting is based on the time this value was reached and the type of variation. That counter, which received this value first, has the lowest position. If two counters receive the same value at the same time, the one who had the lower value in the previous instant, will have the higher position in the new sorting list.

Then, the value $n_{i+1,1}$ is compared with the value $n_{i,\min}$ of the previous moment. If the $n_{i+1,1}$ is lower than the $n_{i,\min}$, then the $n_{i+1,\min}$ of the $(i+1)$ moment becomes the $n_{i+1,1}$, otherwise it remains the $n_{i,\min}$. Following, the $n_{i+1,j}$ is compared with the $n_{i+1,\min}$ and if it is greater, the respective IQ buffer is deallocated from the (min) Destination and it is allocated to the (j) Destination. If no buffer has been deallocated after that procedure, the system remains in the previous allocation state and the normal packet processing continues.

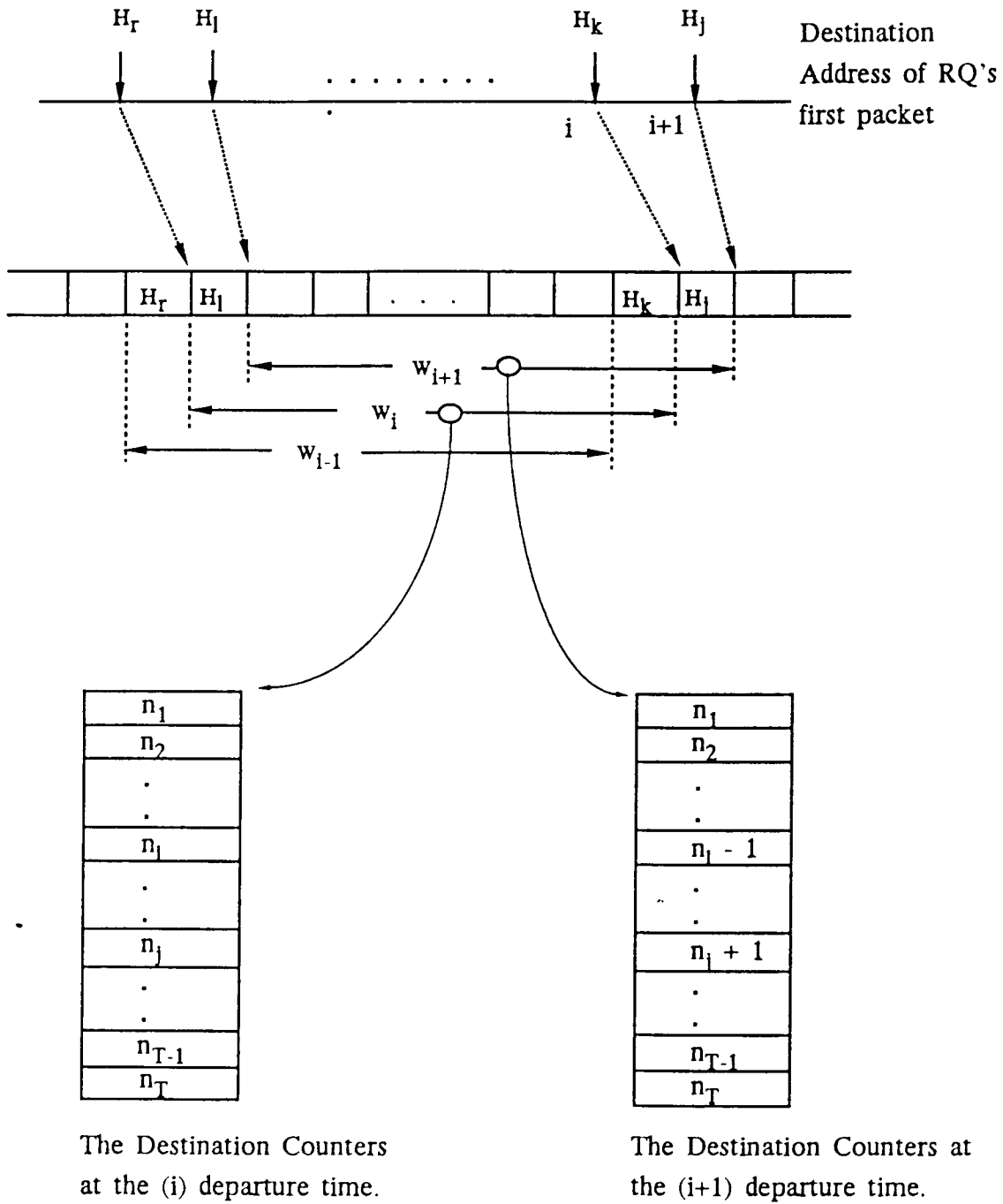


Fig.3. The 'Statistic Window' method

Usually a high-speed LAN, which in our case serves as the backbone network, has a higher allowable packet length than the interconnected LANs. Trying to utilize this characteristic, the t-packet formulation method was developed. The t-packet formulation is implemented by the IQ server as a part of the backbone's protocol profile. According to that method, a number of r-packets, which have the same destination address, is multiplexed to form a t-packet and that packet is transmitted in the backbone network. In the destination gateway, the t-packet is demultiplexed and the r-packets are "regenerated" in order to be transmitted to their final destination. When the traffic load is low, a single r-packet can formulate a t-packet and the system operates in the normal mode, the single IQ mode.

A t-packet is formulated when one of the four following conditions is met:

- i) The maximum packet length of the backbone has been achieved,
- ii) The maximum number of r-packets has been received,
- iii) The Formulation-Time has expired, and
- iv) A Formulation Indication has been received.

The incoming r-packets have various packet lengths and their length has to be taken into account when they participate in the t-packet formulation. When an r-packet is received in the IQ buffer, its length is added to the current length of the formulated t-packet. If the total length exceeds the maximum allowable length, the last r-packet is rejected from the current formulation, a t-packet is generated and the r-packet starts a new formulation round. Otherwise the packet is merged to the t-packet, the 'participation indicator' of the t-packet is incremented by one, the new t-packet length is estimated and a new r-packet is expected. The same holds when the maximum number of r-packets is reached. This condition has been included to fulfil the requirements of the high level protocols.

In order to satisfy the communication parameters, like the 'quality of service', and to restrict the maximum allowable end-to-end delay, a maximum Formulation-Time has to be defined. When a new t-packet formulation is started, a timer is preset to the Formulation-Time and its

decrease starts. When the time expires, the t-packet is formulated and the next r-packet is expected to start a new packet formulation. When the 'Statistic Window' Handler recognizes that an IQ buffer has to be allocated to a new Destination Counter, it uses a 'Formulation Indicator' to inform the IQ server to complete the current t-packet formulation.

III. THE PERFORMANCE ANALYSIS

In this section, we first define the traffic model and give the definitions of the used performance measures. Then, the simulation results are discussed and some topics of interest for further work are derived.

A. The Traffic Model

For the performance analysis of the proposed buffering scheme, traffic conditions which approach known implemented networks have been used. For the backbone network, the 'packet limited access permission' has been used, while for the traffic generated by the connected LAN, the bursty type has been considered. For all these traffic conditions, it is assumed that they follow an exponential distribution for the arrival and departure processes.

The 'packet limited access permission' claims that each time the access to transmit into the backbone network is gained, the node transmits a number of packets (if they exist) and after that the channel is released to allow other nodes to transmit [8]. The permission to transmit a single packet in each access is considered as a specific case of that condition. For all these traffic conditions, it is assumed that the time between two successive accesses follows an exponential distribution.

The bursty type of traffic is used to represent the file transfer internet traffic, which is one of the most important services supported by the backbone configuration. During bursty generated traffic, fixed length packets

are received having the same destination address and an interpacket gap is assumed. The length of the transferred files is considered exponentially distributed and the files are generated using a Poisson distribution. The values of the generated bursty traffic are selected according to the assumed LAN access data rate.

The backbone network was considered to have 100 Mbits/sec data rate and 25 interconnected LANs. Its access protocol allows up to 4 packets to be transmitted in each access permission and the mean value of the exponential distribution of the interdeparture process is 60 accesses/sec. The bursty traffic was simulated using 10 independent sources having the same performance characteristics. The mean value of the interarrival processes from each source is from 0.2 bursts/sec up to 1.8 bursts/sec and the destination address of each file is determined using the uniform distribution. Each file (burst) is transmitted using fixed-length packets (1 Kbyte/packet) while the length of each file is exponentially distributed with 14 Kbytes mean length, 8 Kbytes minimum and 20 Kbytes maximum length. The total packet processing time for each packet in the simple model is 4 msec and the gateway's throughput is 250 packets/sec. We define two packet processing times: t_l is the time which is required to process the r-packet according to the LAN protocol profile, while t_b is the processing time of the backbone protocol profile. Obviously, we don't assure simple bridging processing procedures, which have low processing times (usually 300 μ sec/packet [4]) but we consider a more complicated (and time consuming) protocol architecture for the two networks. In the simulation model, it is also considered the time needed for the formulation of r-packets to a t-packet, t_f . This time is considerably lower than the protocol processing times t_l and t_b .

B. The Simulation Results

The following discussion will be restricted to two basic performance measures [4], throughput and delay, and the influence of some system parameters to these measures. Those parameters are the number of dynamic IQ buffers, the length of the 'Statistic Window' and the protocol processing time distribution. The performance of the proposed structure will also be compared with the performance of the usually used simple architecture.

The gateway throughput is defined as the mean number of packets processed per time unit, while the gateway delay is defined as the time elapsed between the reception of a packet from the connected LAN till the transmission of this packet to the backbone network. Of course, the delay is strongly affected by the considered interdeparture times and the access protocol characteristics of the backbone network. In this analysis, we also use the 'offered packet rate' quantity [4], which is defined as the mean number of packets received from the connected LAN per time unit.

Fig. 4 shows the mean packet delay as a function of the throughput for two different system considerations. As it is obvious, the proposed buffering structure improves the system performance considerably. When the traffic is low, the two systems behave almost the same, but as the traffic increases the difference becomes noticeable. This is caused by the packet multiplexing which allows the gateway to process more r-packets per unit time while the number of processed packets in the system (r-packets and t-packets) remains the same.

Fig. 5 shows the throughput-delay characteristics for various values of dynamic IQ buffers. The number of IQ buffers does not affect the system performance seriously and a small number of IQ buffers is enough to handle the incoming bursty traffic in a better way than the simple buffering model. This in fact is one of the major advantages of this method.

We next consider the gateway performance under the same conditions as above, except for various lengths of the 'Statistic Window'. We relate the

length of the 'Statistic Window' with the number of the dynamic IQ buffers and the statistical behaviour of the bursty traffic. The length of the 'Statistic Window' is considered as the product of the number of the dynamic IQs by the maximum, mean and minimum number of r-packets per burst. As it is shown in Fig. 6, when the 'Statistic Window' length decreases, the gateway's performance becomes better, because the observation of the traffic behaviour is concentrated to the latest packets. There is also a low limit under which, the system's behaviour remains the same and this limit depends on the minimum number of packets per burst.

As it was described earlier, the t-packet formulation implies that different number of r-packets and t-packets are processed, so the protocol processing time distribution has to be considered. Fig. 7 shows the influence of the protocol processing time distribution to the throughput-delay characteristics. As $t_1/(t_1+t_b)$ increases while (t_1+t_b) remains constant, more time is required to process the same number of r-packets (the r-packets are more than the t-packets) and the performance improvement decreases.

IV. CONCLUSION

In this paper we have described a new dynamic buffering structure for LAN gateways which uses the statistic behaviour of the attached LAN and Backbone networks to handle the gateway's logical queues. Using this method, the gateway's performance is improved and more packets can be processed without any additional cost. The performance analysis of that method was derived for bursty type of incoming traffic and a 'packet limited access permission' access protocol for the backbone network.

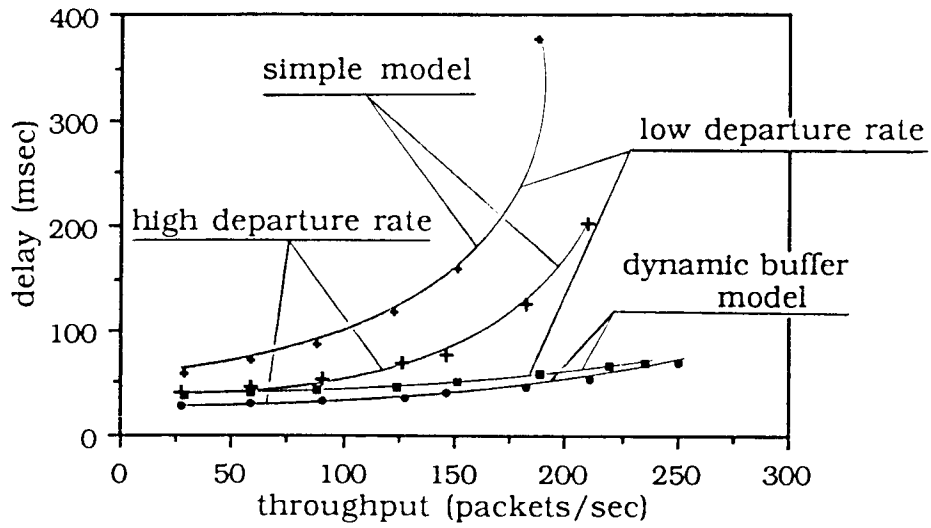


Fig.4. Throughput versus Delay for two traffic conditions

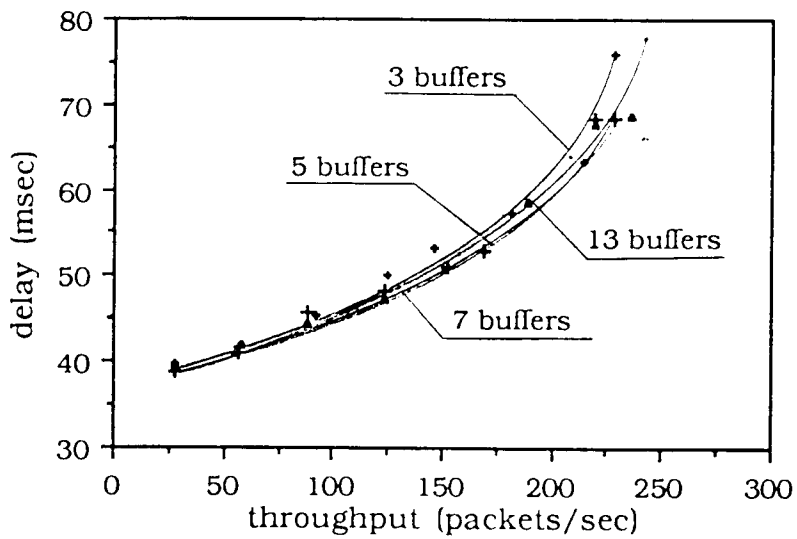


Fig.5. Throughput versus Delay for different numbers of dynamic buffers

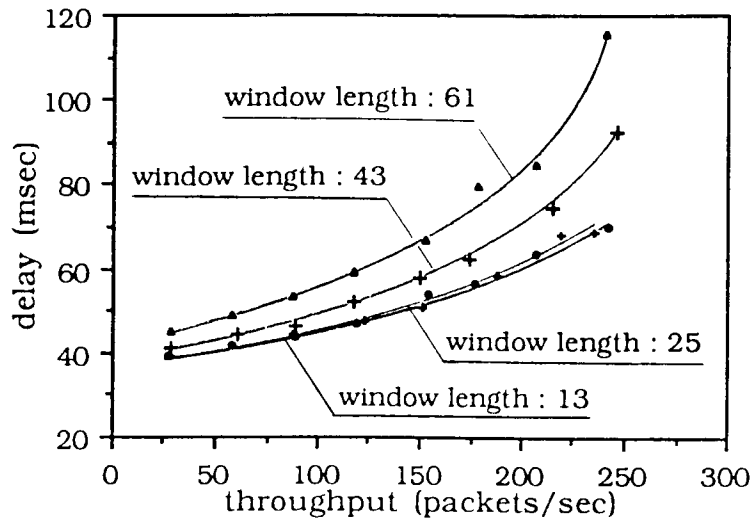


Fig.6. Throughput versus Delay for different lengths of the 'Statistic Window'

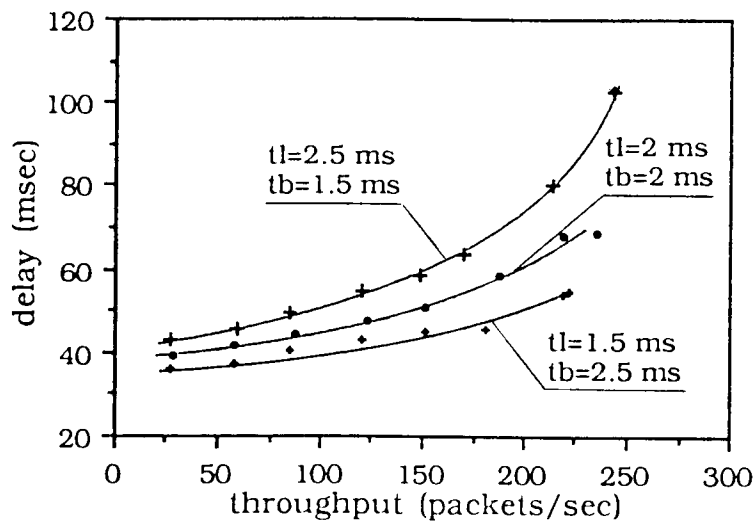


Fig.7. Throughput versus Delay for different values of the protocol processing time distribution.

In order to optimize the performance of the proposed structure, various parameters have been considered and implementation decisions were made. As the results indicate, the length of the 'Statistic Window' is one of the most important system parameters while the number of the used IQ buffers has almost no impact. The protocol processing time distribution also affects the system performance and special consideration has to be taken in the implementation of the time-sharing mechanism of the gateway processor.

As further work, we intend to analyze the proposed buffering structure using interactive and mixed traffic as well as various traffic conditions for the backbone network, i.e. the 'time limited access permission' and the 'exhaustive discipline'. The use of two processors must be considered and their allocation to the two different servers will be examined.

REFERENCES

- [1] F. M. Burg and N. Di Iorio: "Networking of Networks: Interworking According to OSI", *IEEE Journal on Selected Areas in Communications*, Vol. 7, No 7, September 1989, pp. 1131-1142.
- [2] M. Gerla and L. Kleinrock: "Congestion Control in Interconnected LANs", *IEEE Network*, Vol. 2, No 1, January 1988, pp. 72-76.
- [3] P. Martini, O. Spaniol and T. Welzel: "File Transfer in High-Speed Token Ring Networks: Performance Evaluation by Approximate Analysis and Simulation", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No 6, July 1989, pp. 987-996.
- [4] W. Bux and D. Grillo: "Flow Control in Local Area Networks of Interconnected Token Rings", *IEEE Transactions on Communications*, Vol. COM-33, No 10, October 1985, pp. 1058-1066.

- [5] P. Martini: "High-Speed Bridges for High-Speed Local Area Networks. Packets per Second vs. Bits per Second", in *Proc. INFOCOM'89*, Ottawa, Canada, April 1989, pp. 474-482.

- [6] T. Antonakopoulos, J. Koutsonikos and V. Makios: "Design, Implementation and Performance Analysis of an Ethernet to LION Gateway", *NATO Advanced Research Workshop on Architecture and Performance Issues of High-Capacity Local and Metropolitan Area Networks*, Springer-Verlag, Sophia Antipolis, France, June 1990.

- [7] M. Soha and R. Perlman: "Comparison of Two LAN Bridge Approaches", *IEEE Network*, Vol. 2, No 1, January 1988, pp. 37-43.

- [8] D. Roffinella, C. Trincherio and 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, December 1987, pp. 1444-1453.