National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Congestion Control Mechanisms for LANs Interconnected through a High-Speed Backbone

by

Naél B. Hirzalla, B.A.Sc.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

MASTER OF APPLIED SCIENCE

Ottawa-Carleton Institute of Electrical Engineering

Department of Electrical Engineering
Faculty of Engineering
University of Ottawa

August, 1993

ISBN 0-315-89624-8

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

to my family .....

Contents

List of Figures

List of Tables

ABSTRACT

In this thesis, we propose two novel congestion control mechanisms, specially designed for high-speed internetwork systems. These systems consist of a high-speed network which interconnects a few low-speed LANs. Proper coupling of subnetworks having a large difference in transmission speeds has been identified as one of the major challenges to overcome. The design of these control mechanisms is done by keeping in mind that a fair and simple control mechanism is necessary when dealing with high-speed communications systems. A performance evaluation of both mechanisms are carried out by simulation. The first mechanism is suitable for short-haul networks where information exchange can be performed quasi-instantaneously. The second mechanism is an open loop control to be used in systems where the ratio of the propagation time to the packet or cell transmission time is relatively high.

Acknowledgments

I would like first to start this by expressing my sincere appreciation to my thesis supervisor, Dr. Luis Orozco-Barbosa, for his patience and his guidance during my thesis work. Special thanks to Dr. Nicolas D. Georganas for introducing me to the exciting field of congestion control, and for his directions. I would also like to thank both Dr. Roger Kaye and Dr. Oliver Yang for their valuable comments.

My special thanks are due to the University of Ottawa staff in general, and Ms. Lucette Lepage and Ms. Michéle Roy in particular, who are working at the Department of Electrical Engineering.

The friendship and help of all my past and current colleagues is also acknowledged, especially Dr. Sudahkar Ganti, and Dr. Anil Gupta.

I am truly grateful to my beloved wife *Sana*, for her consistent support, without which this work would not have been possible for me.

I am thankful to the Canadian International Development Agency (CIDA) and Jordan University of Science and Technology (JUST) for their financial support.

Acronyms

BF	Buffer Full
B-ISDN	Broad Band Integrated Service Digital Network
B_j	Output buffer capacity of bridge j
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DQDB	Distributed Queue Dual Buses
FDDI	Fiber Distributed Data Interface
GLB	Generalized Leaky Bucket
IEEE	Institute of Electrical and Electronics Engineers
ISDN	integrated Service Digital Network
IVD	Integrated Voice/Data
K_i	Input buffer capacity
K_o	Output buffer capacity
LLC	Logical Link Control
L_{ij}	Input buffer threshold for the traffic stream from LAN 'i' to 'j'
L_l	Lower threshold at the output buffer
L_m	Maximum packet number per stream allowed into the input buffer
L_u	Upper threshold at the output buffer
MAN	Metropolitan Area Network
M_{ij}	Token pool size associated to the traffic stream from LAN 'i' to 'j'
M_r	Red token pool size
NACK	Negative acknowledgment
N_{ij}	Number of levels associated to the traffic stream from LAN 'i' to 'j'
OSI	Open System Interconnection
PDU	Protocol Data Unit
P_{ij}	Routing probability from LAN 'i' to 'j'
QNAP2	Queuing Network Analysis Package 2
QPSX	Queue Packet and Synchronous Circuit Exchange
RF	Reset Full message

SAP	Service Access Point
SF	Set Full message
STM	Synchronous Transfer Mode
S_{ij}	Share parameter associated to the traffic stream from LAN 'i' to 'j'
TA	Terminal Adapter
THT	Token Holding Time
TRT	Token Rotation Time
T_OPR	Operative Target Token Rotation Time
T_{ij}	Traffic stream from LAN 'i' to 'j'
UNI	User Network Interface
VCI	Virtual Channel Identifier
VLSI	Very Large Scale Integration
VPI	Virtual Path Identifier
WAN	Wide Area Network
β_{ij}	Token generation rate associated to the traffic stream from LAN 'i' to 'j'
γ_{ij}	Red Token generation rate associated to the stream from LAN 'i' to 'j'
λ_{ij}	Mean arrival rate of the traffic stream from LAN 'i' to 'j'
μ_B	Access rate of the backbone network
tprop	Network propagation delay
transp	Packet transmission time

Chapter 1
Introduction

1.1 Motivation

Network interconnection has become increasingly important within the field of data communications during the past decade [BIE90]. The term network interconnection has been used broadly to mean any technique which enables systems on one network to communicate with, or make use of services of systems on another network.

Local Area Networks (LANs), such as Ethernet, or Token Ring, allow users to share storage, computer, and peripheral resources. They are considered to be the backbone for a corporate or university network. A LAN must be capable of interconnecting a large number of stations until network limitations are reached for a single LAN, in terms of distance or number of attachments, [BAC88]. As a result, the means of interconnecting subnetworks

together become necessary. In some situations also, it would be preferable to distribute the workstations and servers over several local subnetworks, in order to distribute the load. These subnetworks can then be interconnected by a backbone network running at a higher speed. Such a backbone network, typically a MAN (Metropolitan Area Network), may even be used to interconnect remote LANs distributed over a large geographical area within a city, [VAL89]. A MAN running at speeds of 100 Mbps, 155 Mbps, or more can interconnect lower speed LANs over small, as well as large areas in metropolitan area and over 50 km in diameter. This study focus on a high speed backbone network interconnecting LANs; one of the most important MAN applications nowadays.

In systems of LANs-MAN interconnection, there are speed mismatches between the backbone and the attached subnetworks. Indeed, up to a large number of stations, it is of no problem for the backbone network to carry all the load to the destination LANs. However, simultaneous transmission to the same destination may yield a temporary overload causing information loss at a bridge, an interconnection device. This may result in packet retransmission and eventually congestion. It could be argued that this problem may be solved by large buffers. For many reasons, network engineers generally do not like long queues within the network. In addition, bridges commercially available, generally have a very limited buffering capacity. Furthermore, even large memories can not definitely solve the problem of congestion.

There are various schemes proposed to control such problem with various objectives. These objectives include among other things simplicity, maximum system throughput, and lower mean packet delay. In the literature survey chapter, a few of these schemes are described. The major drawbacks of each scheme are also described.

It was found that none of the described schemes, as well as any of the previously proposed scheme in the literature, aim to decrease the effect of a traffic load increase from one LAN on other traffic from another LAN, and sharing the same destination bridge. This

effect could be in terms of higher mean delay and packet loss probability.

1.2 Objectives

The main goal of this work is to define novel congestion control schemes. The study has been focused on the study of control schemes for high-speed networks interconnecting LANs. The main objectives of the schemes can be summarized as follows:

- To provide a fair share of the network resources.
- To limit the amount of overhead, required by the schemes to operate.
- To allow more flexibility in setting the control parameters.
- To react to the traffic needs whenever it is possible.

1.3 Thesis Outline

In the following chapter, a general overview is presented. A brief overview on the congestion problems frequently encountered in computer networks is first given. A few common used IEEE 802 LANs are then explained briefly. The DQDB and the FDDI networks are next explained. These networks, the DQDB and the FDDI, were intended initially to be used as backbone networks for interconnecting other LANs. A section on network interconnection is placed at the end of the chapter. In that section a general overview on the levels of interconnection and bridging criteria are presented.

In Chapter 3, a literature survey is presented. Three of the mechanisms for congestion control in systems of interconnected LANs-MAN are briefly discussed. The first scheme described, is a simple and straightforward one. It is based on a flow control scheme proposed for DQDB in July 1987 IEEE 802.6 meeting. This scheme aims to be simple. It is

a kind of choke packet scheme. Whenever a bridge buffer, where packets have to wait to access the LAN, reached its capacity, the other bridges are asked to interrupt their transmission towards that one.

The Longest Input Buffer First is another scheme described. This scheme aims to maximize the total system throughput by favoring the offending traffic over the non-offending one during high load periods. This is done by directing the bridge with the longest queue to send packets into the backbone to the destination bridge.

A third scheme is also reviewed in that chapter, the Shortest Output Buffer First. It aims to decrease the mean packet delay experienced by traffic among uncongested bridges. This scheme was based on the idea of routing a customer to the shortest queue.

In this thesis, two different new schemes are proposed in Chapters 4 and 5. The first one requires the bridges to exchange information regarding the status of their buffers. Based on the last state of a specific output buffer, a bridge will decide whether to release or to hold a packet arriving or requesting to enter the backbone.

The capacity of the output buffer is virtually divided among the other bridges into partitions. These shares are easily overruled during low traffic loads. On the other hand, they are in effect during high loads. This will ensure that if one traffic is misbehaving the other traffic from different LANs are less affected.

The second scheme proposed in Chapter 5, is an open loop preventive control. Thus no information is needed to be exchanged between the bridges. It is simply an extension of the well known scheme, the Virtual Leaky Bucket. We introduced a multithreshold input buffer in the bridge. This will allow the *Red* tokens to be generated according to the traffic needs. This scheme will allow the control to react to the traffic needs for more bandwidth. Furthermore, more flexibility in setting the control parameters is allowed.

The first scheme proposed in this thesis is suitable for short haul backbone, where the thresholds defined could be adjusted to compensate with small propagation delay. On the other hand, for long-haul backbone where the propagation delay to transmission time ratio is relatively high, the second scheme is more suitable. Finally in chapter 6, the conclusion emphasizes the different results obtained from our research and some remarks for possible future work.

Chapter 2
Congestion Control and LAN/MAN Technology

2.1 Introduction

Packet switched networks have changed considerably in recent years. Two main factors have played an important role to this change: 1) the dramatic increase in the capacity of communications links, and 2) the introduction of new computer-based services.

The use of new technologies, such as fiber optics, has been accepted as the preferred solution for future growth of the communications network. The fiber deployment has resulted in increasing the bandwidth of the networks. Coupled with recent advances in VLSI and transmission technologies, communications technologies are expected to realize broadband integrated services digital networks.

On the other hand, the development of sophisticated computer-based applications has

spurred the need for more bandwidth. It is now accepted that fast packet switched networks will form the basis for multimedia high speed networks that will transmit data, voice, and video through a common set of backbone nodes and links. Both factors have significant impacts on the design of the protocols and control procedures applied to the network.

The main objective of this chapter is to provide the basic framework of this thesis. The first part of the chapter presents a brief overview of the congestion problems frequently encountered in computer networks. This overview also includes a (simple) taxonomy of the congestion control mechanisms. Thereafter, the chapter briefly reviews some of the most relevant LAN and MAN technologies.

2.2 Congestion Control: an overview

A packet-switched network may be thought of as a distributed pool of productive resources (e.g., channels and buffers) whose capacity may be shared dynamically by the competing users or stations, wishing to communicate with each other [GER80]. Dynamic resource sharing is what distinguish packet switched networks from the circuit switching networks, in which network resources are dedicated to each user for an entire session. The dynamic sharing allows more flexibility in setting up connections and more efficient use of the network resources. Indeed, these advantages do not come without any risk, unless good control is exercised on the users or traffic demands. If the user exceeds its allowed capacity, the network may easily be abused, and an unpleasant congestion effects occurs.

2.2.1 What is network congestion

Network congestion is a state of a network in which some network resources is

oversubscribed or over demanded, [GER80,HAA91]. In other words, network congestion results in a real loss of the overutilised resource. In most cases, the resource utilization measure is assumed to be the goodput, which is the rate of useful data delivered by the network, whereas the throughput is the total rate of data delivered by the network including duplicated or error data.

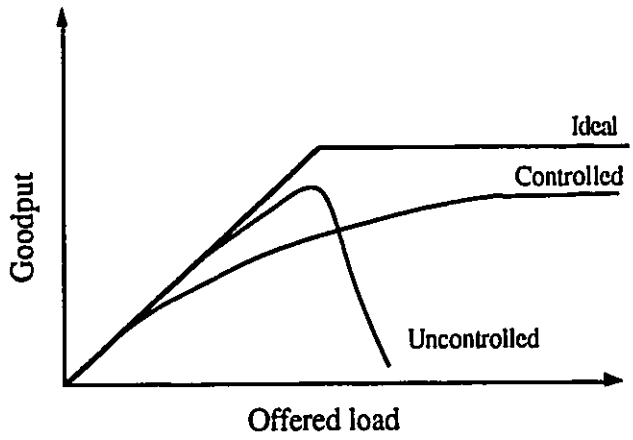


Figure 2.1 *Goodput vs. load;*

Figure 2.1 shows the goodput versus the offered load of a network for the three cases which demonstrates the phenomenon of resource loss due to congestion. The lower curve shows the dramatic decreased due to congestion "uncontrolled curve". The upper curve, "ideal curve", shows the ideal behavior of the network and represents the goal for any congestion control applied. In practice, one may be satisfied with relatively some loss of the resource due to congestion. Congestion controls are necessary in these cases to prevent that dramatic decrease, and to get as close as possible to the ideal curve with, or without the knowledge of the traffic pattern the network will experience. The knowledge of the traffic pattern will actually make the congestion control problem simple to solve. On the other hand, as in most cases, the traffic is either completely unknown or poorly predicted. Thus, a congestion control scheme is needed to be robust, i.e., the control architecture should be

relatively insensitive to any changes in the assumptions, and enough to perform relatively well in "nearly all" cases, [GER80]. This is why some researchers believe that a single scheme may not be adequate to solve the congestion control problem, and that a combination of methods is needed [HAA91,ECK91].

One of the common examples in congestion is buffer overflow. Buffer overflow happens when the rate of data arriving increases over the rate of serving. This could happen if data units arrive from different inputs at the same time to that buffer, or if one input sends data units with a relatively high rate as a result of a misbehaving act by a source. Thus units of data, cells or packets, will fill up the buffer blocking the way against others, leading to packet loss. Those lost packets may be retransmitted which may increase the problem further more, if it is not controlled.

One can state the main functions of a congestion control in a packet switched network as follows:

1- prevention of goodput degradation and loss of efficiency due to network overload, caused by the traffics that have already been accepted and are exceeding the network capacity,

2- fair allocation of resources among competing users, and

3- speed matching between the network and its attached users to prevent buffer congestion at the exit point.

2.2.2 Categories of congestion control schemes

There are two basic classes of congestion control schemes, [HAB91]. The first one is called Reactive Control, while the second is called Preventive Control and sometime Pro-active

Control.

Reactive Control Schemes

This kind of control strategy reacts to a congestion problem after it occurs or when it is about to occur. Its purpose is to reduce the degree of node congestion to an appropriate level. When the congestion problem is detected, the control scheme will take an appropriate action to solve this problem, such as by sending a feedback signal to the sources informing them of the situation, and throttling the input traffics. There are numerous schemes to achieve this goal which differ in the manner by which the feedback message is conveyed [JAI90,GER80].

Window and rate based control are examples of the reactive-type flow control. In window-based control the destination sends a feedback signal, and accordingly, the sender limits the window size, hence the number of transmitted packets. In rate based control, the destination, however, specifies the number of packets per second that the sender can transmit.

Preventive Congestion control

Unlike reactive control schemes where the control is invoked upon the detection of a congestion problem, preventive control schemes try to prevent the network from reaching an unacceptable level of congestion. Admission control to the network and traffic enforcement actions are examples of the preventive type control, [JAB90,EVE90].

Admission control determines whether to accept or reject a new connection at the time of a connection request, as in [GAL89,ECK91]. This decision is based on the traffic characteristics of the new connection as well as the current network load. However, for proper control, admission control is not sufficient alone. Users may deliberately exceed the

traffic volume declared at the connection or the call set up, and may easily overload the network, [LUA88]. Therefore once violation is detected, the traffic flow is enforced by discarding or buffering violating packets. For this reason, the normal approach to execute this type of strategy is to implement it at the access control points, which are the entrance points to a network, see Figure 2.2.

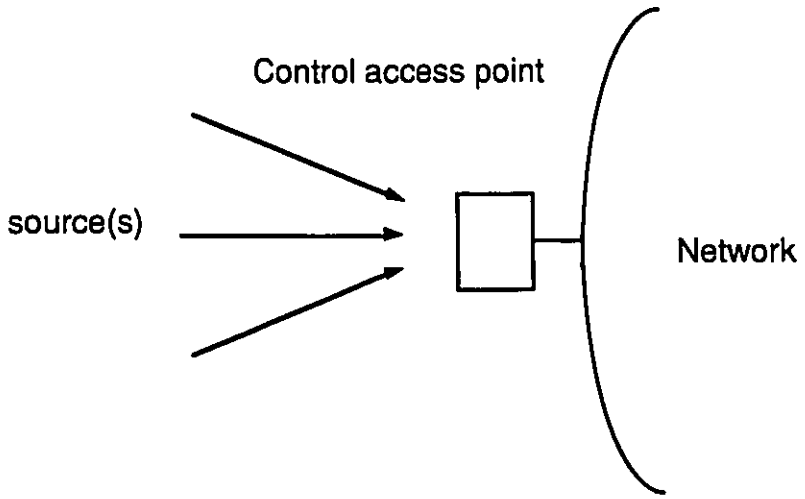


Figure 2.2 *Control access point*

2.3 Family of IEEE 802 Standards

Figure 2.3 shows the different standards defined by the IEEE 802 committees. At the physical and media access control layers, the IEEE 802.3 (CSMA/CD), the IEEE 802.4 (Token Bus), and the IEEE 802.5 (Token Ring) standards have been already established. Moreover, committees are presently working in the definition of the two other standards: IEEE 802.6 DQDB MAN and IEEE 802.9 IVD (Integrated Voice/Data) LAN interface. At the upper part of the data link control layers, there is the IEEE 802.2 Logical Link Control (LLC) sublayer which has been already established. Finally, IEEE 802.1 describes the relationships among the previous standards as well as internetworking and network management issues. In the next subsections, some of the standards are reviewed. A more detailed description of

OSI reference model and B-ISDN is given in Appendices A and B, respectively.

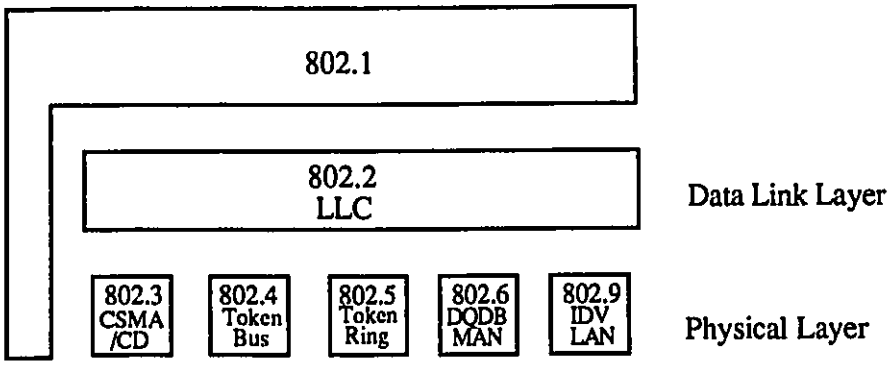


Figure 2.3 *IEEE family of standards*

2.3.1 IEEE 802.3

General Description

The IEEE 802.3 CSMA/CD (Carrier Sense, Multiple Access with Collision Detection) standard [ANSI85C] is based on the use of a common bus shared (see Figure 2.4). For the Ethernet specification, it calls just for a coaxial cable physical channel driven at 10 Mbps, whereas, the IEEE 802.3 standard is intended to encompass several medial types and techniques for signal rates from 1 Mbps up to 20 Mbps. Each station, in the CSMA/CD protocol, must listen to the bus, before transmitting. If there is no current detected on the medium, the station can transmit its queued packets. If it happened and the message collides with that of another station, then each transmitting station intentionally sends a few additional bytes, called jam signal, to guarantee that all transmitting stations on the network will detect this collision. The station remains silent for a random amount of time, backoff, before attempting to transmit again; the delay before retransmitting is assumed to be a random number of defined slot times distributed uniformly over the range of 0 to $2n$, with 'n' is the

nth retransmission attempt, until 'n' reaches 10. The retrial interval then remains at 0 to 2 *10 until 'n' reaches 15. At this point the collision event is reported as an error to the higher layers. The CSMA/CD protocol defines a slot time which largely determines the dynamics of collision handling. However, the slot time should be larger than the sum of the physical layer round-trip propagation time and the jam time generated by the station having detected a collision.

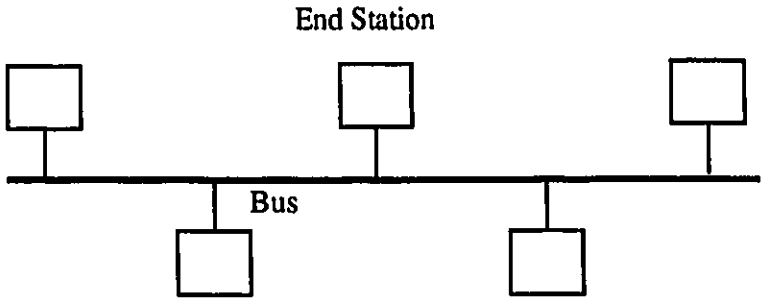


Figure 2.4 *CSMA/CD bus topology*

Performance of IEEE 802.3 CSMA/CD

In evaluating the CSMA/CD performance, the parameter $a=\tau/m << 1$ has been found to have a large influence, 'τ' as being the end-to-end propagation delay along the bus while 'm' being the frame transmission time [BUX81]. For acceptable system throughput, it has been shown that the parameter 'a' should be kept low. If 'a' decreases, collisions are more readily detected. On the other hand, if 'a' increases towards 1, which means that the propagation delay tends to be as long as the frame transmission time, the probability of collision is increased causing system degradation. Thus, the parameter 'a' explains the fact that CSMA/CD implementations like Ethernet, must be limited to a low speed, short bus length and use a minimum frame length. In Ethernet, the speed has been fixed to 10 Mbps while the bus length has been bounded to 1.5 km and the slot time has been fixed to 512 bits [SCH87]. It is also important to point out that the minimum access time by a station in

CSMA/CD, and unlike other types of access method, is only the waiting time which is needed to detect an idle period, whereas there is no guaranteed limit on the maximum access time.

2.3.2 IEEE 802.4 Token-Passing Bus

General description

The IEEE 802.4 Token Passing Bus, or simply Token Bus access method can be implemented on the same type of media as the CSMA/CD protocol; however, its access method is based on the use of a token to regulate the access of the bus. The station having the token, has a momentary control of the medium. The station can then send its data units, frames, until its queue is empty or its token holding time, THT, expires to prevent any station from dominating the network. Subsequently, the station passes the token to another station residing on the same medium. Each station should have a logical number and know its predecessor and its successor. This allows transferring the one and only token in a descending or ascending order. This is done by sending a token MAC control frame, which contains the address of both the sender station and the destination one. Furthermore, the station having the lowest logical number will send the token to the station having the largest logical number. By this, a logical token-passing ring will be formed between all the stations, Figure 2.5.

Various MAC control frames are defined by IEEE 802.4 standard. One is used for transferring the token between stations another is used for reestablishing the lost token. Other control frames are also used to insert a new station or reject a physically disconnected station from the logical ring. In particular each station sends periodically MAC control frame, called "solicit successor", which invites new stations to be integrated in the logical token ring. The

IEEE 802.4 standard also, defines eight service classes of priority, class 7 being the highest and class 0 being the lowest. Furthermore, four access classes are defined that correspond to four request queues for storing frames requesting transmission: access class 0 for service classes 0 and 1, access class 2 for service classes 2 and 3, and so on.

When a station receives the token, it first serves the highest access class queue. The station can send frames until THT for that station expires. If this timer expires before sending all the queued frames, the station must release the token to the next station on the logical ring. However, if the station sends all the frames of the highest priority queue before the THT expires, then the station begins to serve the lower priority queues. The service of the three lower access classes is based on the use of a timer, called token rotation timer (TRT). The TRT is used to allocate more or less time for transmitting frames, depending on the time spend by the token in one logical ring rotation. The access class service algorithm consists of loading the residual value of the TRT into the THT, and reloading the same TRT with the target rotation time for that access class. Then, if the THT has a remaining positive value the station can transmit frames at this access class until either the THT expires or the access class queue is empty. When either event occurs the station begins to serve the next lower access class. Only after doing the same with all the lower access classes, the token will be passed to the next station. Thus, the TRT method allows lower access class traffic to fill in the bandwidth which is not used by higher access class traffic. Bit rates of 1,5, and 10 Mbps are suggested for implementation by the IEEE 802.4 standard. For example, the MAP/TOP standard (Manufacturing Automation Protocol/Technical and Office Protocol) adopted the token-passing bus running at 5 and 10 Mbps.

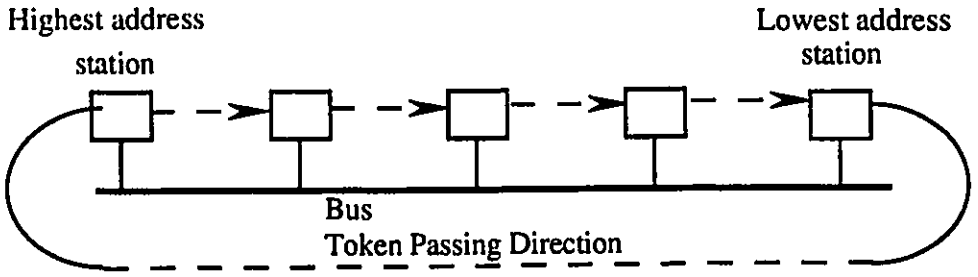


Figure 2.5 *Token Bus topology*

Performance of the token-passing bus

The major disadvantages of the token passing bus method over the CSMA/CD are the complexity of the protocol and the waiting time for a station which has to send frames under low traffic intensity [BUX81]. However, the token passing method is more fair and allows the use of several classes of priority. On the other hand, some performance work [PANG89] has showed that, under heavy load conditions, the station service time will oscillate in one of many periodic patterns which motivates further research to dynamically fix the TRT period for improving network sharing and utilization. Concerning the maximum access time by a station, the Token Bus method guarantees a limited maximum by introducing the THT timer.

2.3.3 IEEE 802.5 Token-Passing Ring

General description

The IEEE 802.5 Token Passing Ring, or Token Ring [ANSI85A] is quite similar to the Token Bus; however, a physical ring is used instead of a bus, see Figure 2.6. To start transmitting, each station has to capture the one and only token passing on the ring. When

the frame transmission is completed the station passes the token to the next station on the ring, and so on. Each station is either in transmit or repeat state. In transmit state, it sends out its own frames after capturing the token. It transmits until it has completed transmission of all its queued frames or until the transmission of another frame cannot be completed before the token holding time (THT) expires. In repeat state, the station sends back each received frame bit by bit onto the ring. It may copy the frame while repeating it if it recognizes its own address. It may also modify some control bits to indicate that the frame was correctly received. Like the IEEE 802.4 standard, the IEEE 802.5 standard defines a priority mechanism which comprises eight levels of priorities. However, the IEEE 802.5 priority mechanism is more centralized. Hence, only one priority level is recognized on the whole ring at a given time, and this is indicated by the passing token. When a station grabs the token it can only send frames having the same or a greater priority level than that of the token. In addition to that, the ring priority is indicated by the token. The token ring priority mechanism is based on the use of reservation bits contained in each token ring frame header, and on the use of stack registers in each station. The reservation bits are used by the stations requesting for higher token priority. This may force the next station generating the token to increase the token ring priority. The stack registers are used by the station generating the token to store the new and old priority levels in order to bring back or reset the ring to the original service priority, at a later stage. The IEEE 802.5 committee defined the standard of a 16 Mbps token ring [GLA89]. In addition to this increase in speed, a new token release which allows multiple frames to be on the ring at a given time, but only one free token, is also proposed. Indeed, work stations will be authorized to transmit a token immediately after sending a frame, without having to wait for the frame to return, which is not the case in the present IEEE 802.5 standard.

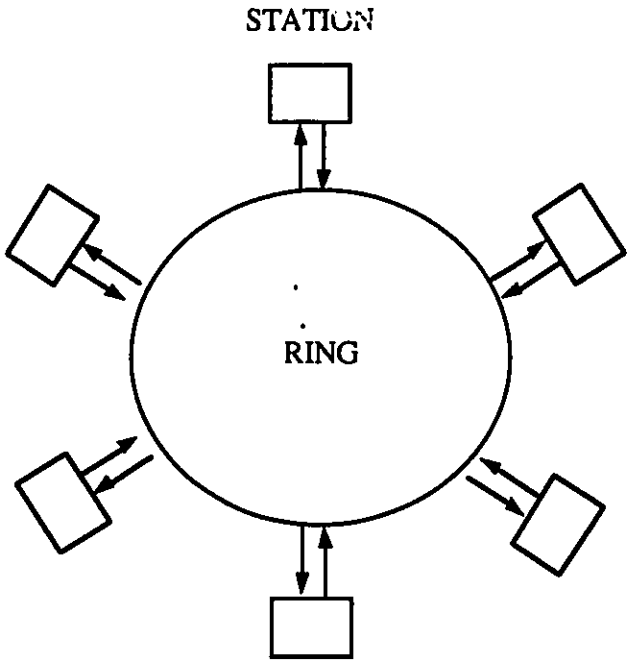


Figure 2.6 *Token Ring topology*

Performance of IEEE 802.5

Token Ring access method gives better performance compared to the CSMA/CD method, especially under heavy traffic loads, which lead to an inefficient operation of the CSMA/CD method due to the increase in the number of collisions. The difference between them is amplified if the ratio of propagation delay to packet transmission time (parameter 'a' discussed earlier) is increased, such as in the case of increasing the network speed. The Token Passing Bus and the Passing Ring access methods under the use of only one priority in the network should lead to a similar performance, since the concept of the logical ring is similarly applied to both cases. On the other hand, it is interesting to compare the performance of the priority mechanism used by each network. Indeed, the token passing ring access restrict stations to only send frames having equal or higher priority than that of the token, while in the case of the token passing bus access method, the sending station, if its

THT has not expired, can transmit low priority frames, although some other stations may have higher priority frames queued for transmission. Similar to the Token Bus and unlike the CSMA/CD methods, the Token Ring limits the maximum time a station will wait to gain access on the ring by the THT timer concept.

2.4 Metropolitan Area Network (MAN)

In recent years, *Metropolitan Area Networks* (MANs) technology has evolved as an extension of LANs [IEEE88]. Broadly speaking, a MAN is a network capable of providing high-speed switched connectivity across distances typical of those found within a metropolitan area. Furthermore, this connectivity is of such a nature as to allow different types of traffic (e.g. voice, video, data) to be carried simultaneously.

In the early eighties, the ANSI X3T9.5 study group was founded with a charter to develop interface standards for future high performance storage systems [BUR88]. At that time, it was predicted that this set of standards, known as Fiber Distributed Data Interface (FDDI), will be widely used as a backbone for other LANs.

For 802.6 the background may be even more surprising; as its inception in 1982, the project 802.6 working group was driving by the preceived need of the satellite communication industry to link its ground stations with customer sites within a few tens of miles of its earth stations, [MOL90]. In [MOL90], after listing the applications most frequently cited for MANs, the author states : "... of these (applications), it is LAN interconnection which has come of age and is contributing the most toward the push of MANs."

2.4.1 Fiber Distributed Data Interface (FDDI)

FDDI is a high speed ring network offering a bit rate up to 100 Mbps, see Fig. 2.7. It uses unidirectional transmission within the fiber. Its medium access is controlled by a token passing protocol [ROS86]. More precisely: FDDI uses a time token protocol. The token holding time, i.e., the maximum time available for transmission, is given by the difference between the token rotation time measured in the previous cycle and a value called "Operative Target Token Rotation Time" (T_OPR). If the previous token rotation took longer than T_OPR, then the station receiving the token must not transmit the data but real-time traffic.

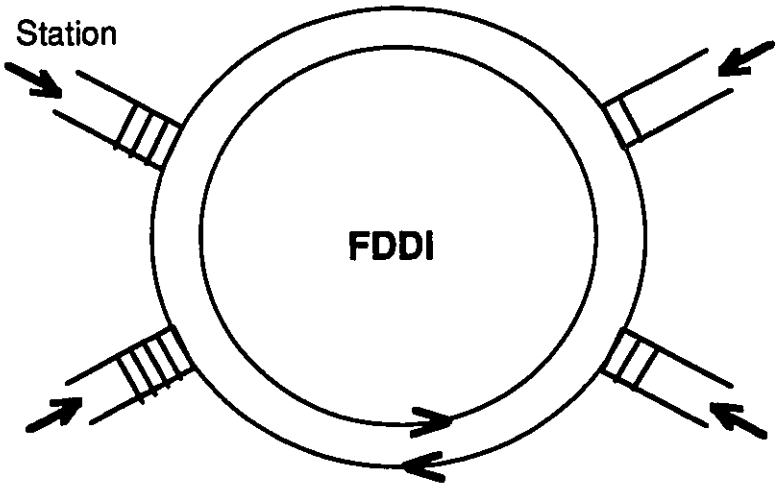


Figure 2.7 *FDDI Topology*

As shown in numerous publications [BUX87,MAR91] the heavy load performance of the FDDI decisively depends on T_OPR. The reason is that a large T_OPR allows more efficient operation. In contrast to this, for light to moderate load the average packet delay is almost independent of T_OPR.

2.4.2 Distributed Queue Dual Buses (DQDB)

In 1982, the IEEE 802.6 committee started elaborating a standard for defining a MAN. A slotted ring approach was first proposed [IEEE85]. Isochronous and non-isochronous slots were respectively defined for supporting voice or video and data communications. Where a slot refers to a fixed-length data unit periodically generated. Each isochronous slot was assumed to be allocated by a master node upon demand. At the same time, each empty non-isochronous slot was assumed to be grabbed by the first node seeing the slot and having a packet queued for transmission.

By 1986, the working group adopt another proposal under the name of DQDB (Distributed Queue Dual Buses) [IEEE88]. This proposal is based on a protocol called QPSX (Queue Packet and Synchronous Circuit Exchange) after an Australian proposal, [BUD86,NEW86,NEW88]. DQDB still defines isochronous and non-isochronous slots, but the access to the non-isochronous slots is based on a reservation scheme.

The purpose of IEEE 802.6 committee had been originally to propose a high speed network driving at speeds over 100 Mbps, integrating packets and circuit switching, and allowing to interconnect lower speed networks such as IEEE 802 LANs. A consensus emerged within the IEEE 802.6 committee on the type of the protocol to be used as a MAN standard, and DQDB was selected. The DQDB architecture comprises two unidirectional contra buses allowing full duplex communications between each pair of nodes, see Figure 2.8. Each node of the network is attached to both buses.

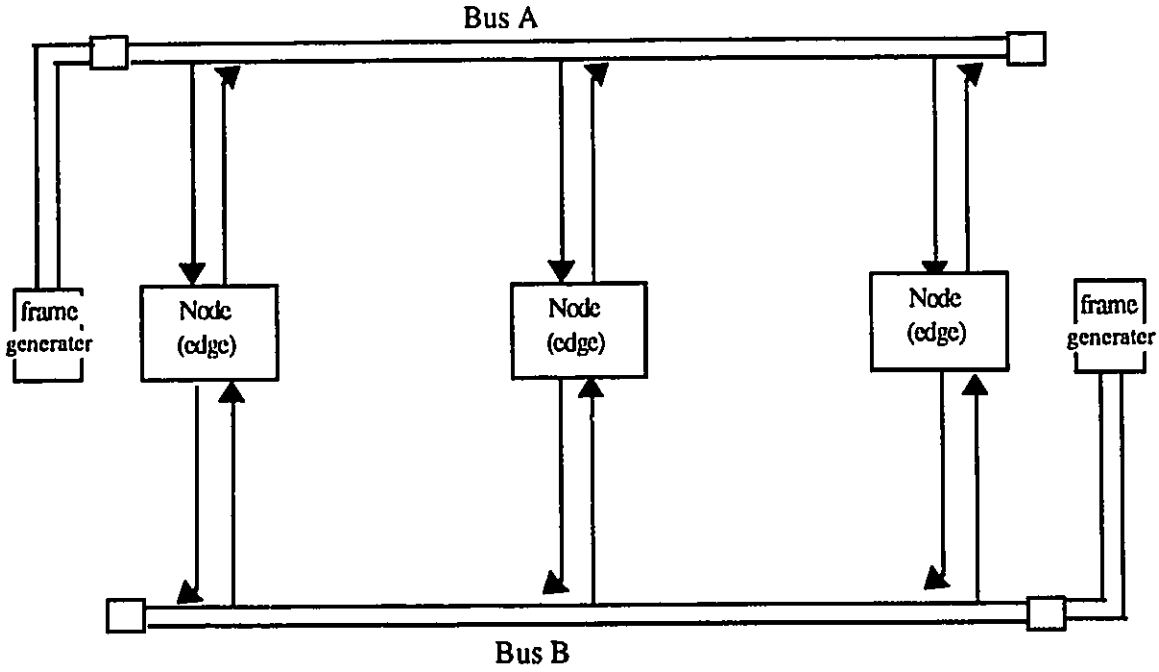


Figure 2.8 *DQDB topology*

The DQDB capacity is dynamically allocated to meet the demands of the isochronous traffic. The capacity that is not allocated to isochronous traffic can be shared by packet switched traffics. The timing structure of DQDB is based on implicit frames of 125 μsec which comprises a given number of fixed slots shared between isochronous and non-isochronous communications. Initially, slots of 32 octets were defined [IEEE87A]. However, in order to facilitate the interconnection of DQDB with B-ISDN (Broadband Integrated Services Digital Network) [CCITT89], the slot length has been increased to ATM (Asynchronous Transfer Mode) cell length which is presently 53 octets [CCITT88,RID88].

2.5 Network interconnection

For over a decade, various studies and investigations have been centered around the interconnection of both Wide Area Networks (WANs) and Local Area Networks (LANs). However, the future environment, in which internetworking will be intended, is likely to be

very different. This is due to the emergence of Metropolitan Area Networks (MANs) and Broadband ISDN (BISDN). These new technologies will enable the construction of the global information network, which will be possibly made of several thousands of subnetworks connecting millions of end nodes. The bandwidth available, which is now in the range of hundreds of Megabit per second, will affect the design of protocols to be suitable for this new environments.

2.5.1 Levels of interconnection

Networks are connected at various levels of the OSI model, [SUN90]. The devices and techniques used for interconnecting are different from one level to another. Alternatives exist all the way from the lowest (physical) level to the highest (application) level. When different networks and protocols are involved, the interconnection involves a conversion process between services provided for comparable functions in each network. The complexity of this process and the quality of end-to-end services resulting are largely determined by the level of interconnection chosen. The following subsections summarize the key features of each major alternative.

Physical level: Interconnection devices operating at the physical level are generally called repeaters. They forward individual bits of packets as they arrive, perhaps translating from one medium to another (e.g., baseband coaxial cable to optical fiber). The resulting interconnected system functions essentially as a single network at the data link level. This approach is typically used to interconnect several physically separated segments of a LAN system.

Link level: Interconnection devices operating at the link level receive entire frames from one link, examine the link level protocol header, and possibly forward the frame onto another link. They are typically called bridges or more specifically MAC bridges since they operate at

the Media Access Control level. They are used to relay MAC frames from one subnetwork to another, since the major motivation for their use is to interconnect subnetworks with different MAC protocols.

Network level: Traditionally, interconnection at the network protocol level has been a WAN problem, where different networks had independently developed different protocol mechanisms for the variety of network level functions such as routing, error handling, segmenting, and congestion control. Network level routers can be used mainly to route packets among subnetworks, and often called gateways.

Transport level: In OSI architecture, the transport service is supposed to be an end-to-end service, so transport level gateways are supposed to be a violation of the architecture. Nevertheless, they may be practical benefit when common upper level protocols are in use but different transport protocols are available.

Higher level: interconnections at levels higher than the Transport level are used also to interconnect different subnetworks, however they are less commonly used.

2.3.2 Bridging Classification Criteria

In general, bridges can be classified according to the following criteria, [TANT90]:

- The OSI level at which interconnection is performed: as discussed earlier.
- The number of subnetworks interconnected: *dual port bridges* are used to interconnect only two subnetworks; whereas, *multiport bridges* are used to interconnect more than two subnetworks.

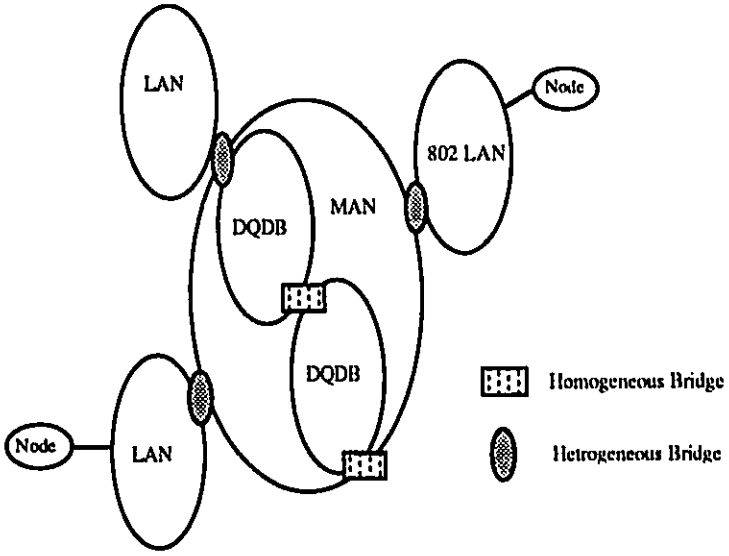


Figure 2.9 *An Interworking System*

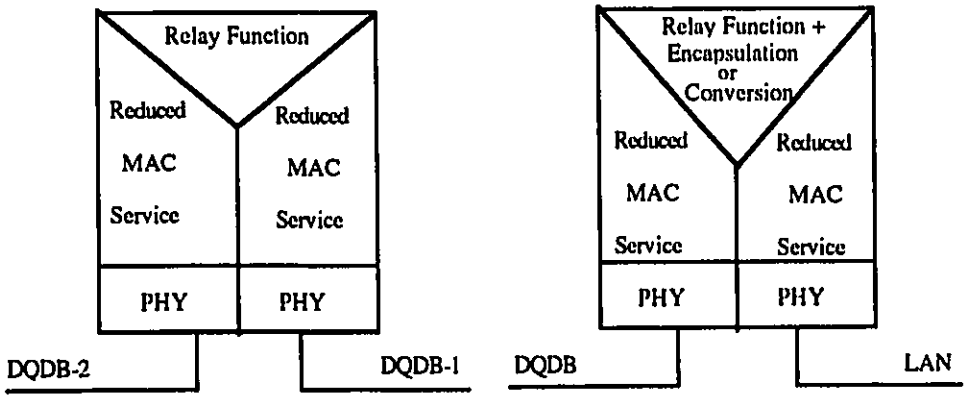


Figure 2.10 *Heterogeneous and Homogeneous bridges*

- The uniformity of OSI levels: *uniform bridges* transmit packets at their output at a level corresponding to that of the packets they receive. *Non-uniform bridges* do not adhere to that principle. In other words, a non-uniform bridge receiving LLC frames may process and eliminate the LLC header and relay a MAC frame to its output.

- The type of interconnected subnetworks: *homogeneous bridges* relay frames among subnetworks using the same frame format, e.g., when connecting two or more DQDBs, whereas, *heterogeneous bridges* relay frames between heterogeneous subnetworks, e.g., when connecting DQDBs and Token Rings, see Figure 2.9.

- The procedure used to transform frames: *encapsulating bridges*, add new headers and trailers to the received frame without deleting its original header or trailer. *Converting bridges* replaces the original header and trailer by new ones which correspond to the MAC of the output subnetwork.

- The physical integration: *Integrated bridges* are usually built in one integral device, whereas split or sometimes called *remote bridges* are usually made of two or more units.

Chapter 3
Congestion Control Mechanisms for Systems of Interconnected LANs-MAN

3.1 Introduction

The main objective of this chapter is to review the current trends in the area of congestion control for systems interconnecting LANs through a MAN. In the first section, a description of the system topology under study is given. This section also highlights the main reasons giving raise to congestion in such kinds of systems. An overview of the recent congestion control mechanisms follows. This overview includes a classification as well as an analysis of each one of the different mechanisms presented.

Fig. 3.1 shows the general topology of a system consisting of '*n*' LANs and a MAN, a high speed backbone network, interconnected via bridges. The bridges provide the protocol

services and mechanisms for assuring a proper operation of the overall communications system. Each bridge comprises a set of two buffers, the input and the output buffer (cf. Fig 3.2). The input buffer is used for holding the packets traveling towards the backbone network, while the output buffer holds the packets arriving from the backbone network and waiting to access the LAN directly attached to the bridge.

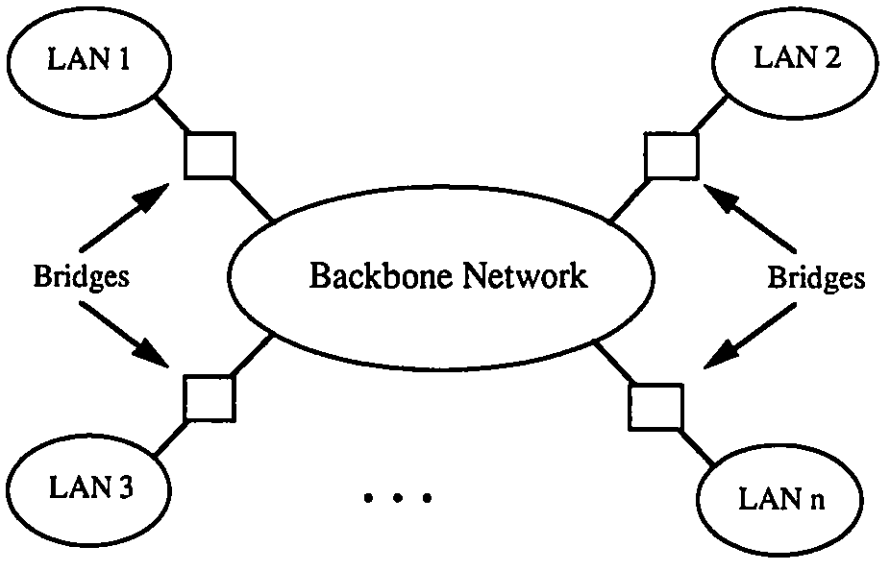


Figure 3.1 *Network Topology*

The most important feature of a system of interconnected LANs-MAN, that the model has to capture, is the speed disparity between the LANs and the MAN, [GUM92,GER92]. The service rate of the output buffer is determined by the capacity as well as the access time of the LAN. This time or the access delay depends on the bit rate as well as on the medium access control (MAC) mechanism of the LAN. Furthermore, this delay is a function of the local load being applied on the LAN. Same for the input buffer, where its service rate is a function of the load on the backbone, its capacity and MAC mechanism.

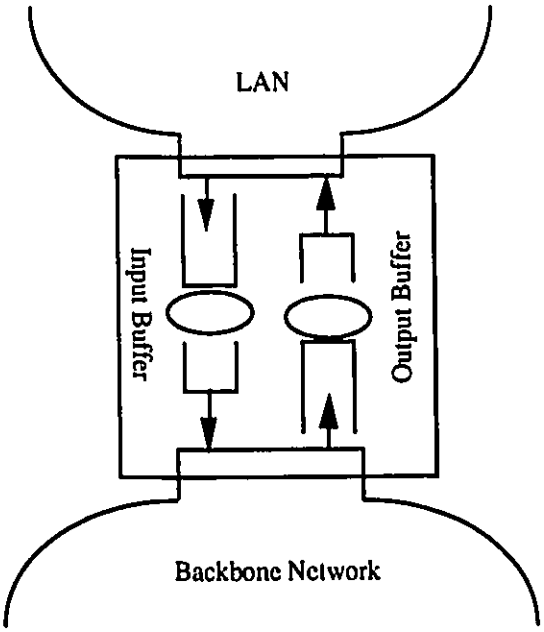


Figure 3.2 A bridge

In systems of interconnected LANs-MAN, the bit rate of the MAN (the backbone network) is anticipated to be much greater than the LANs' bit rates. Data messages are transmitted from one LAN to another, via the backbone network, as a sequence of data packets. Packets are lost whenever the capacity of the buffers is exceeded. Due to the speed disparities of the LANs and the MAN, loss is most likely to occur at the output buffers. Buffer overflow in the bridges will thus be a common phenomenon if no proper flow control is applied to regulate the traffic flow across the bridges. The mechanisms to be put in place to recover lost packets are out of the scope of this thesis. It is simply assumed that high-level protocols will have to be provided.

There is a multitude of policies and schemes under which a flow control scheme can operate. Some mechanisms will perform better than others in meeting a defined set of objectives. This set of objectives includes, among other things, simplicity, feasibility, fairness, response speed to congestion, and implementation cost. However, most of the policies focus primarily on obtaining a higher network throughput (goodput) during resource

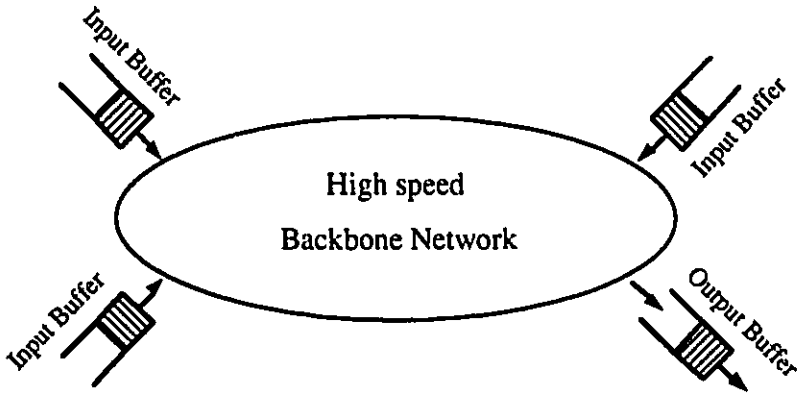
over demanding, which implies minimizing the probability of having to turn away traffic, and not considering the unfairness problem.

3.2 The Simple Scheme

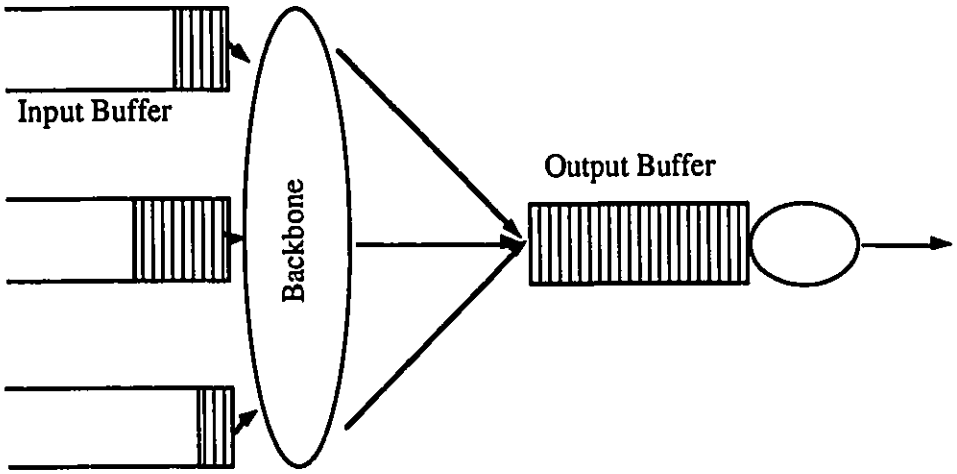
3.2.1 The mechanism

A simple and straightforward scheme was proposed to control the traffic entering the backbone network through the bridge [IEEE87B]. The operation of this scheme is depicted in Fig. 3.3. It simply allows the traffic to fill up the output buffer until its capacity is reached. Packets transmission is then interrupted whenever the destination output buffer is full. It is then resumed as soon as some output buffer space is made available. At this time, packets are transmitted to the output buffer on a First In First Out (FIFO) basis, i.e., the first packet that was held at any of the input buffers due to lack of space in the output buffer is the first one that should be transmitted, followed by the second packet, and so on.

The Simple Scheme was defined after a flow control mechanism proposed for the QPSX (DQDB) standard in July 1987 IEEE 802.6 meeting, [IEEE87B]. In this latter mechanism, the output buffer accepts as many packets as its capacity allows. Further arrivals are then discarded and negatively acknowledged. Acknowledgment is performed on a per packet basis. The source (the input buffers in our model) keeps transmitting packets up to a maximum window size and ceases transmission whenever a negative acknowledgment (NACK) is received. The NACK contains a ticket number assigned by the output buffer in the order of arrival at the buffer. This ticket number is used by the input buffers to determine the message to be retransmitted, and the input buffer that should retransmit. In this case, a FIFO order is enforced.



(a)



(b)

Figure 3.3 *The simple scheme mechanism*

In other words, as described by Inai et al, [INA92], the simple scheme is a kind of a choke packet scheme. Each bridge observes its output buffer. When the output buffer becomes full, the bridge sends a buffer-full, (*BF*), message to the other bridges to

suspend their transmission towards this bridge. When an output buffer space becomes available, the bridge then sends a receive-ready, (RR), message. Each bridge should therefore know the status of all the other output buffers, full or not full, i.e., being allowed or not to transmit a packet towards a specific bridge.

The Simple Scheme prevents packet loss at the output buffer. Packets are held at the input buffers whenever the output buffer is full. The packet loss probability is expected to be lower than that obtained for a system without any control mechanism. For instance, the traffic of a highly active LAN will be retained at the input buffer reducing the possibility of overflowing the capacity of the output buffer. The packet loss of the overall system traffic is therefore reduced.

3.2.2 Drawbacks

In the Simple Scheme, one of the major drawbacks is the high degree of overhead introduced during high load periods. According to this scheme, a bridge has to inform all the other bridges as soon as an overflow condition is detected in its output buffer. It does so by sending a *(BF)* message to all the other bridges. As soon as some buffer space is freed in the output buffer, the bridge is required to inform the others of the space availability. A *(RR)* message will be sent. The bridges are then allowed immediately to resume the transmission towards that output buffer. Under a heavy load condition, the output buffer is more likely to fill up once again. This implies that the whole aforementioned procedure will be repeated. This will lead to a highly costly overhead as the bridge will keep informing all the other bridges of each single packet arrival and departure at its output buffer.

Another drawback of this scheme may result when an input buffer overflows. Under this scheme, packets whose destination output buffer is filled up are kept at their input buffers. These packets may stay for a relatively long time at the input buffers blocking the

way against other packets. New arrivals will then be discarded even if there is enough space at their destination output buffers.

Moreover, traffic from different input buffers sharing the same destination output buffer will be affected by any misbehaving act of one traffic, if that traffic increases its rate heavily. They have to pay a high price for the sake of the offender traffic. The price paid will be translated into both, a higher packet delay and a relatively higher loss probability. This leads to an unfair traffic control.

3.3 Longest Input Queue First

3.3.1 The mechanism

In [WON89], Wong and Schwartz have proposed a congestion control scheme (Fig 3.4). The goal of this scheme is to maximize the total throughput of a system under heavy load. By assuming zero propagation delay between the input and the output buffers, they have shown that their scheme is near optimal for controlling the traffic in systems of interconnected LANs-MAN, in order to maximize the system throughput. The idea behind this scheme is to preserve a crucial resource: the output buffer space. This resource is to be used exclusively by those inputs that are in danger of overflowing.

The scheme, referred to as the longest input queue first, states that new arrivals should be kept at the input buffers. This scheme further specifies that under normal conditions at most one packet should be allowed to remain in the output buffer. By normal conditions, it is understood that none of the input buffers is in danger of being overflowed.

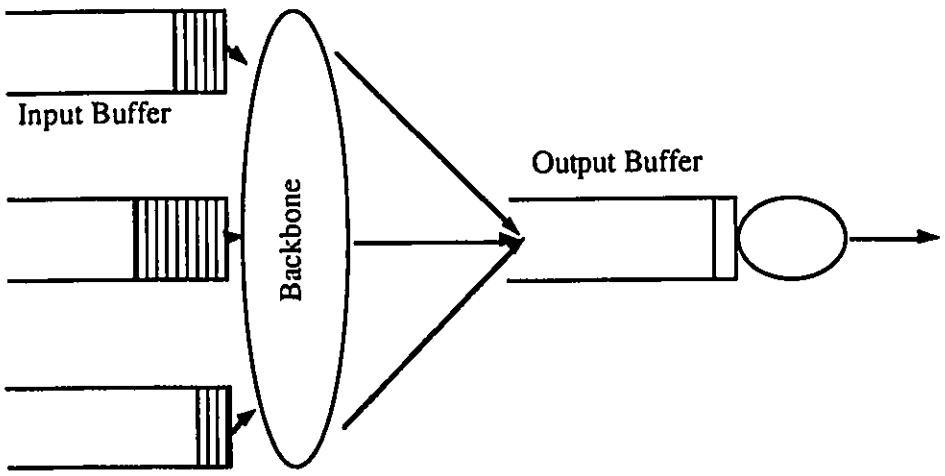


Figure 3.4 *The Longest Input Queue First Scheme*

The operation of this scheme can be briefly summarized as follows: as soon as the output buffer gets empty, a packet from one of the input buffers should be immediately transferred to the output buffer. The packet to be transferred to the output buffer will be chosen among all the packets requesting access on the backbone. The input buffer exhibiting the largest queue size will be given the priority to transmit over all the other input buffers. The reason for this choice is simple: the input buffer with the largest queue size is the most likely to be in danger of overflowing (for equal buffer capacities). In the event of equal sizes, a random choice is made. Under heavy load conditions, when an input buffer gets full, it is allowed to send a packet to the output buffer in order to make space available for new arrivals. However, the packet will be considered lost if the output buffer is also full.

By preserving spaces at the output buffer for those input buffers that are in danger of overflowing, this scheme is expected to improve the overall system performance: lower overall packet loss probability and a higher system throughput. The results of the study reported in [WON89] have shown a significant improvement on the throughput of the output buffer as compared to the one provided by the simple scheme. Indeed, this scheme is based

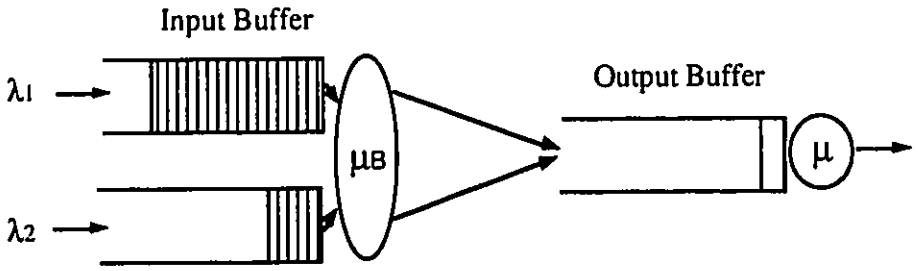
on a basic principle: the ability to thwart off temporary input data bursts by allowing the traffic streams to 'borrow' the output buffer space.

3.3.2 Drawbacks

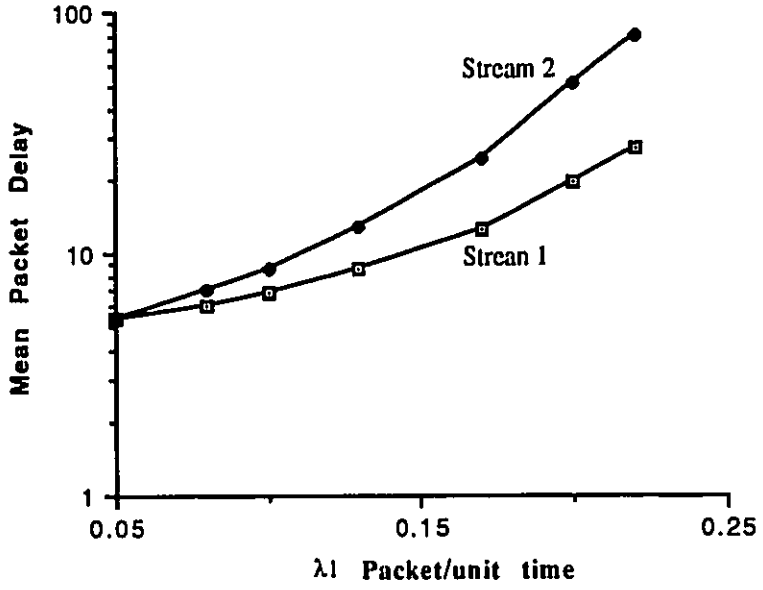
The major disadvantage of this scheme is its unfair policy. In order to increase the total system throughput, this scheme favors the increasingly demanding traffic streams, "*the offenders*", over the non-offender traffic streams. The offenders will experience relatively low mean packet delay and packet loss. Under this scheme, the non-offender inputs must hold their packets for a while, in order to be able to get access onto the backbone. For example, consider a system consisting of two input buffers and one output buffer, see Fig 3.5. Consider the case where the first input buffer receives packets from continuously highly demanding sources over a certain period of time with a high mean rate of λ_1. Further, assume that the incoming traffic to the other input buffer is low over the same period of time with a mean rate of λ_2. Those packets, that belong to the second stream, will have to wait until they can compete with the first input queue size, which seems to be for a relatively quite long time.

To see this, assume λ_2 is equal to *0.05* , μ_B*=4.0*, and μ*=0.3 packets per unit time*, in Figure 5.3a. λ_1 is set initially to *0.05* and increased up to *0.22 packet per unit time*,, and the arrival processes are considered Poisson. Figure 3.5b shows the mean packet delay for both streams 1 and 2 versus λ_1. This Figure shows the higher mean delay experienced by stream 2.

In other words, the non-offender traffic will have to pay a high price: a higher mean delay to allow the offender a better access to the network, and a higher system throughput.



(a)



(b)

Figure 3.5 *Higher priority for higher rate in Longest Input Queue First Scheme*

Another drawback is the high amount of overhead needed by the scheme. Under normal conditions, as soon as the packet found in the output buffer departs, the bridge has to direct the longest queue bridge to send a packet. This will end by sending a message for each departing packet at the output buffer. In addition to that, the available buffer spaces should be known for each bridge, which will increase the overhead further.

3.4 Shortest Output-Queue First

3.4.1 The mechanism

Another scheme has been recently proposed by Inai and Ohtsuki [INA91]. The goal of this scheme is to ensure fast packet delivery to those bridges whose output buffers have available spaces, even if another output buffer is overflowing. In their scheme, each bridge should continuously inform the other bridges with the number of available spaces in its output buffer. At the input buffer, the highest priority in transmission is given to the packets whose destination bridge has the largest available buffer space. This scheme is based on a proposal by Ephremides et al [EPH80], which states that the optimal routing decision is to route a packet to the shortest queue path. In order for this scheme to operate, every bridge should be aware all the time of the queue length of the output buffer of every other bridge.

Under this scheme, packets destined to congested bridges are kept waiting at their input buffers. These packets may stay in the buffers for a relatively long time due to the low transmission priority, and eventually preventing other packets, associated to other traffic streams, from entering the input buffer. In other words, when a long queue of packets destined to congested bridges is built up in the input buffer, the packets destined to uncongested bridges may be discarded due to input buffer overflow.

In order to overcome this problem, Inai and Ohtsuki have enhanced their scheme. The number of packets associated to a traffic stream are monitored at the input buffers. At

any time, the number of packets per stream is limited to a given number (smaller than the buffer capacity). The modified scheme will thus ensure some buffer space at the input buffers for those packets whose destinations are not congested.

In order to demonstrate the validity of their approach, Inai and Ohtsuki considered the case when the rate of a certain traffic, say T_{ij}, the traffic flowing from LAN *'i'* to LAN *'j'*, in a system of five LANs, is constantly increased while all the other traffic streams are kept constant. They also assumed a ratio of 1:3 between the capacity of a LAN and that of the backbone network. Their results indicated that their scheme reduces the mean packet delay of the traffic between uncongested bridges in comparison to the simple scheme described earlier. Their scheme showed some reduction in the packet loss probability as well.

3.4.2 Drawbacks

It is obvious that to achieve the goal of the scheme the propagation delay should equal to zero in order for each bridge to know the exact number of spaces available at each of the other output buffers.

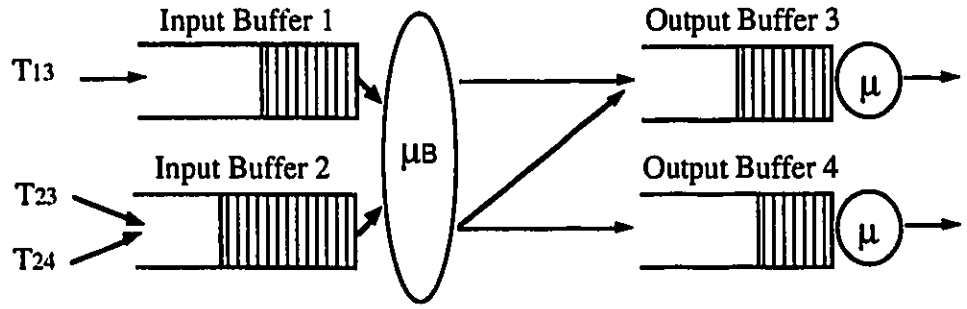


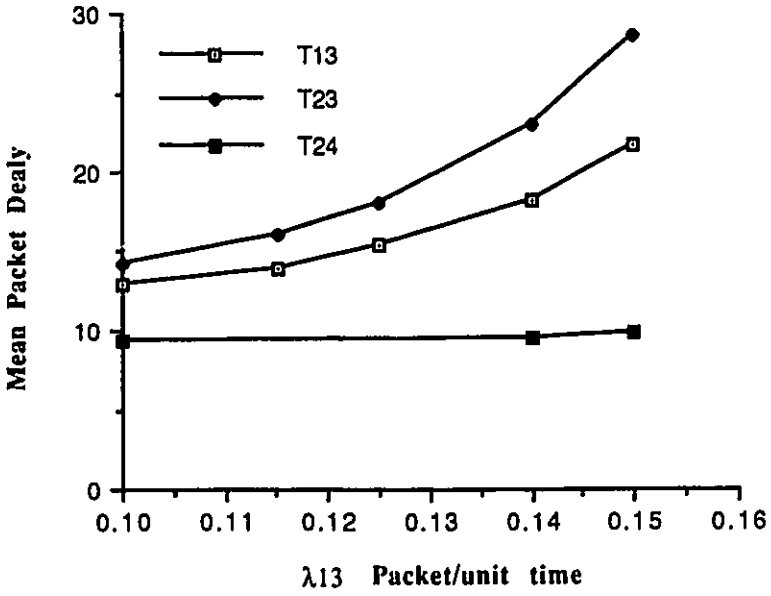
Figure 3.6 *Scenario Depicting Unfairness (Shortest Output -Queue First Scheme)*

This scheme still fails to be fair to all the traffic sharing the same output buffer. Since

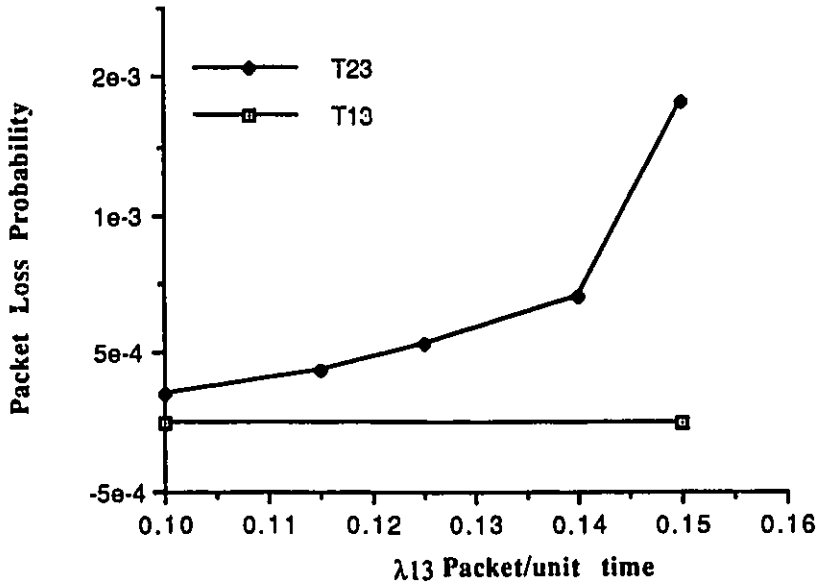
the idea behind this scheme is to guarantee a fast packet delivery between uncongested bridges, the highest priority in transmission at the input buffer is given to those packets destined to the shortest output queue length. The priority mechanism is only applied at each input buffer independently. Consider the scenario depicted in Fig. 3.6 where two input buffers are competing to get access on the backbone. Assume that the first input buffer only holds packets destined to output buffer 3, while the other input buffer holds packets destined to both output buffers. Moreover, take the case when the packet arrival rate of T_{13} is higher than that of T_{23}. It is obvious that the scheme will achieve its goal by favoring traffic T_{24} over T_{23}. However, since the packets of traffic T_{13} are allowed into the output buffer without control, traffic T_{23} will be heavy penalized in terms of packet loss and mean delay.

To see this effect, assume λ_{23} and λ_{24}, the mean arrival rates of T23 and T24 respectively, are equal to *0.1*, μ_B*=0.45*, and μ*=0.3 packets per unit time*, in Figure 3.6.λ_{13} is set initially to *0.1* and increased up to *0.15*. Figure 3.7 shows the mean packet delay and the loss probability of the various streams versus λ_{13}. It could be easily explained the increasing value of the mean delay experienced by T13. This increase in the mean delay is because of the rate increase of that particular traffic. However, the price paid by T23 in terms of higher mean delay and loss probability is a result of the unfair policy.

In cases when the backbone network capacity is much higher than that of a LAN, which is usually the case, it is obvious that this scheme will end up behaving exactly like the simple scheme described earlier: packets destined to output buffers having at least one buffer space available will find no trouble in accessing the backbone network. On the other hand, packets destined to a full output buffer will have to wait for a space availability. Thus the control messages sent by each bridge does not have to carry the number of available spaces at its output buffer. This is because the input buffers will simply use this information to determine if there is a space available at a particular output buffer or not. Thus, this scheme exhibits similar drawbacks to the ones of the simple scheme.



(a)



(b)

Figure 3.7 *Shortest Output-Queue First*
a) Mean Packet Delay vs. λ_{13}; b) Packet Loss Probability vs. λ_{13}

3.5 Conclusion

The control schemes presented in this chapter are characterized by their inability to deal with a fair allocation of the resources. All of them 'work' for the benefit of the offender at the expenses of other users (traffic). In certain cases, some of these schemes have shown undesirable behaviors ranging from the generation of an excessive number of control packets to an improper allocation of buffer space. Thus the need for a fair as well as simple mechanism becomes important. In the next chapter, a new scheme is proposed. The new control mechanism proposed has as main aims: 1) fairness, by providing a fair sharing of resources, and 2) simplicity, limiting the overhead introduced in particular during high load periods. The overall quality of service of a system of interconnected LANs-MAN will depend on the success of the mechanism to meet its objectives. This scheme works on a share principle. The amount of the information exchanged among the bridges is limited and carefully controlled during high load periods. This last characteristic makes the proposed scheme suitable to control the fair sharing of buffer space.

Chapter 4
A Share-Based Congestion Control Scheme

4.1 Motivation

The main aim of this chapter is to present a novel congestion control scheme for systems interconnecting LANs to a MAN. In this kind of systems, congestion is mainly due to the speed disparity between the LANs and the backbone network (the MAN). Congestion is characterized by an increasing number of packets being held at the bridges, and resulting in packet loss due to lack of buffer space.

The previous chapter briefly described some relevant congestion control schemes for systems interconnecting LANs to a MAN via bridges. In this description, the main features as well as the drawbacks of each one of the schemes were highlighted. In the three schemes, the bridges are required to exchange information regarding the state occupancy of their buffers. These schemes rely on the feedback information sent by the destination bridge (the

output buffer) to the source bridges (the input buffers) in order to work properly. For this reason, these schemes are not suitable for systems where the ratio between the latency and the packet transmission time may be important. This kind of situation are mainly to be found in systems interconnecting high-speed networks [KLI92].

Another major drawback of these schemes is their inability to fairly share the system resources among all the traffic streams. For instance, some of these schemes may give a higher priority to a highly active traffic stream over streams exhibiting low or moderate activities. These latter traffic streams will experience relatively higher packet delay and loss. The inability to control this kind of situation makes these schemes unsuitable to be used in systems where resource sharing is an important feature. Thus, the need for a fair and simple control policy becomes essential.

The main motivation for designing a new control scheme is twofold. First, the control scheme must guarantee a fair sharing of the system resources, in particular, the sharing of the critical resources: the (output) buffer space. This is done by virtually partitioning the buffer space. In this way, low and moderate active traffic streams are not severely affected by highly-active streams. Moreover, the scheme is flexible enough to allow the highly active streams to use any unused buffer space. This allows enough flexibility and guarantees fairness.

Secondly, a control scheme for high speed network must be simple. All the control schemes presented in the previous chapter, introduce a lot of overhead as stated. The *Simple Scheme* requires a bridge to inform all the other bridges continuously with its space availability during high load periods; in the *Longest Input Queue First Scheme*, the input buffer queue size of each bridge must be known by all the other bridges in the system; and in the *Shortest Output Queue First Scheme*, the number of available spaces in the output buffers must be communicated among all the bridges. This requirement, the continuous exchange of information among bridges, may severely affect the system performance.

The new mechanism proposed herein, does not require each bridge to know the exact output buffer size in the other bridges. Instead, each bridge needs only to know that the number of packets in an output buffer has reached a certain level. This mechanism and the information required for its operation are explained in the following sections. A performance study of this new scheme via simulation is also given. This study focuses on the ability of the scheme to provide a fair service. The success to provide a fair service is measured in terms of the delay and packet loss probability faced by the different streams, [FEN91].

4.2 A Share-Based Congestion Control Mechanism

4.2.1 System status

As already stated, one of the main aims of the control scheme proposed herein is to guarantee a fair sharing of the system resources. The control scheme is exercised at the entry points of the backbone, i.e., at the input buffers. The control scheme relies on the status of the destination output buffer in order to decide whether an arriving packet is to be sent to the output buffer, or to be held at the input buffer.

In the scheme, a bridge uses two bits to keep track of the status of the (output) buffer space of each other bridge attached to the backbone. Each pair of bits will reflect the state of the corresponding output buffer. For example, in a system comprising *'n'* bridges, each bridge requires $2(n-1)$ bits to represent the status of the output buffers of all the other bridges.

These two bits are named: FULL and EMPTY bits. Each one of these bits can be in either of the two states, namely the set state (represented by the Boolean value *'1'*), or the reset state (expressed by the Boolean value *'0'*). The possible output buffer states are depicted in Table 4.1. An output buffer can be in one of three states. The state *'11'*, i.e., both bits, FULL and EMPTY bits, are set, is not used. It is important to notice that having the

FULL (EMPTY) bit set does not necessary mean that the corresponding output buffer is full (empty).

In order to keep up to date these two status bits, every bridge must inform all the other bridges with the change of status of their output buffers. The information exchange among the bridges may be done in different ways. One way is to use a separate channel, or by means of high priority control packets transmitted through the high speed backbone network, or even by piggy-backing it on any data packet.

It is worth to mention that each packet in the FDDI MAN backbone is returned to the sender (the bridges in this case). Thus, the control information may be transmitted by the destinations to the sources by piggy-backing them into the data packets flowing in the opposite way. However, in other MANs, packets are not sent back to the sources, e.g. DQDB. In this kind of networks, the use of special 'choke' packets with higher priority should be used instead.

In this study, it is assumed that each bridge keeps track of its output buffer. It will send messages to either set or reset the corresponding bits found in the other bridges, whenever its output buffer reaches a certain state. Table 4.2 shows the three messages used in the scheme.

For a bridge to know when to send a message, two thresholds in the output buffer have been defined: L_u, called the upper threshold, and L_l known as the lower threshold. By defining these two thresholds, or levels, in the output buffer in terms of packets, assuming fixed size packets, three different states could be distinguished depending on the queue size: when the queue size is less than L_l, larger than L_u, or in between.

Table 4.1 *Output buffer status bits*

FULL / EMPTY	0	1
0	NORMAL Packets may be sent to the output buffer if necessary	ALMOST FULL Packets are held at the input buffer
1	ALMOST EMPTY Packets should be sent to the output buffer	

Table 4.2 *Control Messages*

Type	**Status bits**
SF	Set FULL bit
SE-RF	Set EMPTY bit and Reset FULL bit
RE	Reset EMPTY bit

Initially, assume that the number of packets in a certain output buffer is equal to zero. When the number of packets exceeds the lower threshold, L_l, the bridge sends a *RE* message to all the other bridges. Upon receiving the message, the bridges reset the corresponding EMPTY bit.

At the time the number exceeds the upper threshold, L_u, the bridge sends a *SF* message to the other bridges, informing them of this change. Upon receiving the *SF* message, the bridges set the corresponding FULL bit.

In the third case, if the queue size of the output buffer decreases below the lower threshold, L_l, the bridge must inform all the other bridges by sending a *SE-RF* message. Upon receiving the message, the bridges set and reset the corresponding EMPTY and FULL bits, respectively. The conditions and actions discussed above are summarized in the following algorithm:

IF *a packet arrives at the output buffer* THEN
BEGIN
    IF *new queue size* = L_u THEN *send(SF);*
    IF *new queue size* = L_l+*1* THEN *send(RE);*
END
ELSE IF *a packet departs* THEN
    BEGIN
    IF *new queue size*=L_l THEN *send(SE-RF);*
END;

Therefore, five transitions between the states are allowed. These transitions discussed above are drawn in Figure 4.1.

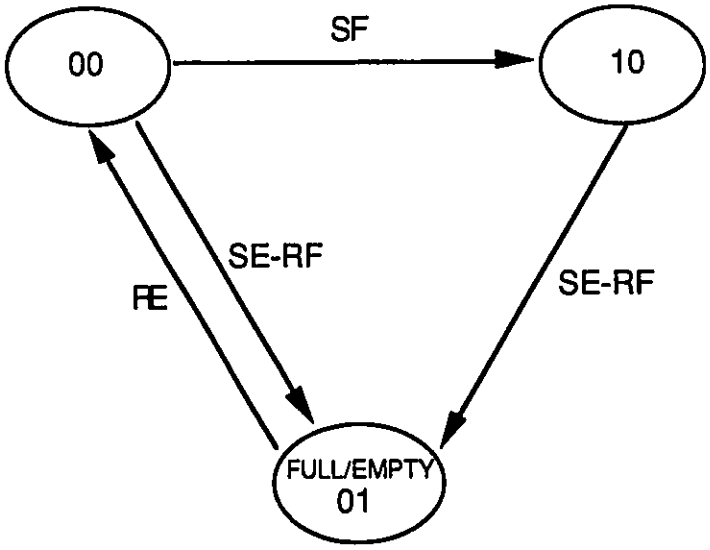


Figure 4.1: *State transitions at a source bridge*

4.2.2 The congestion control procedure

In the following, a buffer capacity is referred to in terms of number of packets, assuming the use of fixed size packets. The system is initialized with the status bits of all the bridges set to *'01'* (Almost empty state); it is assumed that the output buffers are initially empty. When a packet passes over a LAN, it is extracted or filtered by the bridge attached, if its destination is recognized. The corresponding EMPTY bit associated with its destination is then checked. If the EMPTY bit is found set, the packet is allowed to enter the backbone. Otherwise, the packet is held in the input buffer. By holding the packets at the input buffers, the mechanism preserves some space at the output buffers to be used by those inputs whose capacity may be exceeded.

Upon receiving a *SE-RF* message from bridge *'j'*, (which indicates that bridge *'j'* has enough space in its output buffer), bridge *'i'* is allowed to transmit up to S_{ij} data packets to

that bridge. S_{ij} is defined as the *share parameter*. It defines the minimum amount of output buffer space at bridge *'j'* (in packets) that bridge *'i'* may use. This parameter is set at the time of initializing the system and is defined for each pair (source bridge, destination bridge). In other words, the output buffer of bridge *'j'* is virtually divided into n-1 partitions. The size of each one of these partitions is given by the parameter S_{ij}. For a given bridge *'j'*, the summation of all the partitions, namely the buffer sharing, must be set so as to satisfy the following relation:

$$BufferSharing = \sum_{\substack{i=1 \\ i \neq j}}^{n} S_{ij} \leq (L_u - L_l) \quad \text{for output buffer of bridge } 'j',\ 'j' = 1,2,\ldots n \qquad (1)$$

This relation indicates that the summation of all the partitions for a given output buffer of bridge *'j'*, must lie between the lower threshold, L_l, and the upper threshold, L_u. This eliminates the possibility of having to reject any of the S_{ij} packets that belongs to the stream from LAN *'i'*, where $i=1,2,\ldots n$, at the output buffer of bridge *'j'*. This could be explained as follows: when the queue size of the output buffer of bridge *'j'* reaches the lower threshold L_l, a *SE-RF* message is sent. Every other bridge *'i'* will be allowed to send up to S_{ij} packets to bridge *'j'*. If all the bridges happen to send their full share, S_{ij}, relation (1) guarantees that the number of packets to be sent plus the number of packets already found at the output buffer of bridge *'j'* will not exceed the upper threshold, L_u. This guarantees that non of these packets will be lost due to output buffer overflow.

Moreover, the mechanism allows any buffer space left to be available to those bridges having more packets to transmit. Assume the case when bridge *'i'* has more than S_{ij} packets in its input buffer, when the *SE-RF* message is received from bridge *'j'*, and at least one of the other bridges has less packets than its share value out of bridge *'j'*. In this case, bridge *'i'* may make use of the space left at the output buffer of bridge *'j'*. However, bridge *'i'* will have to transmit these additional packets with a lower priority than the one used for transmitting the S_{ij} packets. The use of a lower priority guarantees that none of the additional

packets will be transmitted before transmitting all the S_{kj} packets, where $k=1,2,...n$, $k \neq j$, from all the bridges. By this, a fair share of the buffer space is enforced among the traffic from different LANs sharing the same destination LAN. This is of particular interest during periods of high activity.

When the output buffer reaches the upper threshold, bridge *'j'* is required to inform all the other bridges by sending a *SF* message. Upon receiving the message, the bridges stop the transmission of any packet directed to bridge *'j'* immediately.

It is important to highlight the use of the upper and lower thresholds: L_u *and* L_l. The parameter L_u is mainly used to avoid the overflow of the output buffers. A bridge should be able to prevent output buffer overflow by anticipating any abnormal situation. It is well known that the ratio between the latency and the bandwidth plays a major role in the performance of high-speed networks. In this kind of networks, the bridges may continue to transmit their packets while the output buffer is full, and a control packet for stopping the transmission is on its way. Obviously, these packets will be lost at the output buffer due to lack of space. For this, the parameter L_u is introduced and can be set up by realizing that the maximum number of packets in transit to a given bridge *'j'*, at any time, is given to satisfy the following relation:

$$B_j - L_u \geq p_j \times \frac{tprop}{transp} \qquad (2)$$

where B_j and L_u are the size and the upper threshold of the output buffer *'j'* in terms of packets, respectively. *transp* is the packet transmission time, and *tprop* is the network propagation between bridge *'j'* and the farthest input buffer transmitting to it. p_j is the routing probability to bridge *'j'* and is given by (3). Equation (2) shows that the amount of buffer space beyond the upper threshold are left for those packets that may be on their way to the output buffer.

$$p_j = \frac{1}{n} \sum_{i=1; i \neq j}^{n} P_{ij} \qquad (3)$$

where P_{ij} is the probability that a packet arriving at the input buffer of bridge *'i'* will be destined to LAN *'j'*. In turn, P_{ij} is given by:

$$P_{ij} = \frac{\lambda_{ij}}{\lambda_i} \qquad (4)$$

where λ_{ij} represents the mean arrival rate of packets from LAN *'i'* to LAN *'j'*, assuming that all the arrival processes are Poisson, and

$$\lambda_i = \sum_{j=1, j \neq i}^{n} \lambda_{ij} \qquad (5)$$

On the other hand, the latency of the backbone, access time plus propagation delay, is assumed negligibly small compared to the packet transmission time over a LAN. In this case, the output buffer is always busy, whenever there is at least one packet in the system. This is because whenever the output queue size reaches L_l, *a SE-RF* message is immediately sent to the other bridges. Upon receiving this message, the bridges will resume the transmission of packets towards that bridge. Hence, the time needed for the *SE-RF* message to reach the bridges, and for a packet to be transmitted and reached its destination output buffer is very small compared to the time needed for L_l packets to be transmitted onto the LAN. Therefore, new packets will reach the bridge even before its output buffer is completely emptied.

Furthermore, packets with the corresponding EMPTY bit of the destination buffer not set, i.e. not *'0'*, may fill the input buffer during high loads, and block the entrance against other

packets. To overcome this problem, when the input buffer becomes full, the packets in the buffer are scanned in the order of arrival. The first packet found with a corresponding FULL bit *'1'* is allowed to enter the backbone in order to leave a space behind for a new arrival.

4.3 Performance Study

The main objective of this study is to evaluate the mean delay and packet loss probability of the various streams, as well as the amount of overhead introduced by the scheme. The packet loss probability is defined as the ratio between the number of packets being lost to the total number of packets being generated per a traffic stream. The mean delay is defined as the average time elapsed, from the time a packet is received at the input buffer of a source bridge, up to the time the packet leaves the output buffer of the destination bridge.

4.3.1 The simulation model

Fig. 4.2 shows the queuing network model representing the internetwork system. Each input buffer is connected to a LAN via a single port. Multiple traffic streams having different destinations arrive at each one of the various input buffers. It is assumed that packets arriving at an input buffer will be routed to any of the output bridges with a probability given in Eq. (4).

Arriving packets at the input buffer will either be stored or discarded depending on space availability. For the purpose of comparison between different schemes, a Poisson packet arrival process is assumed. In practice, the rate and order of packet transmission depend not only on the congestion control scheme being used, but also on the rate and the access method employed by the MAN.

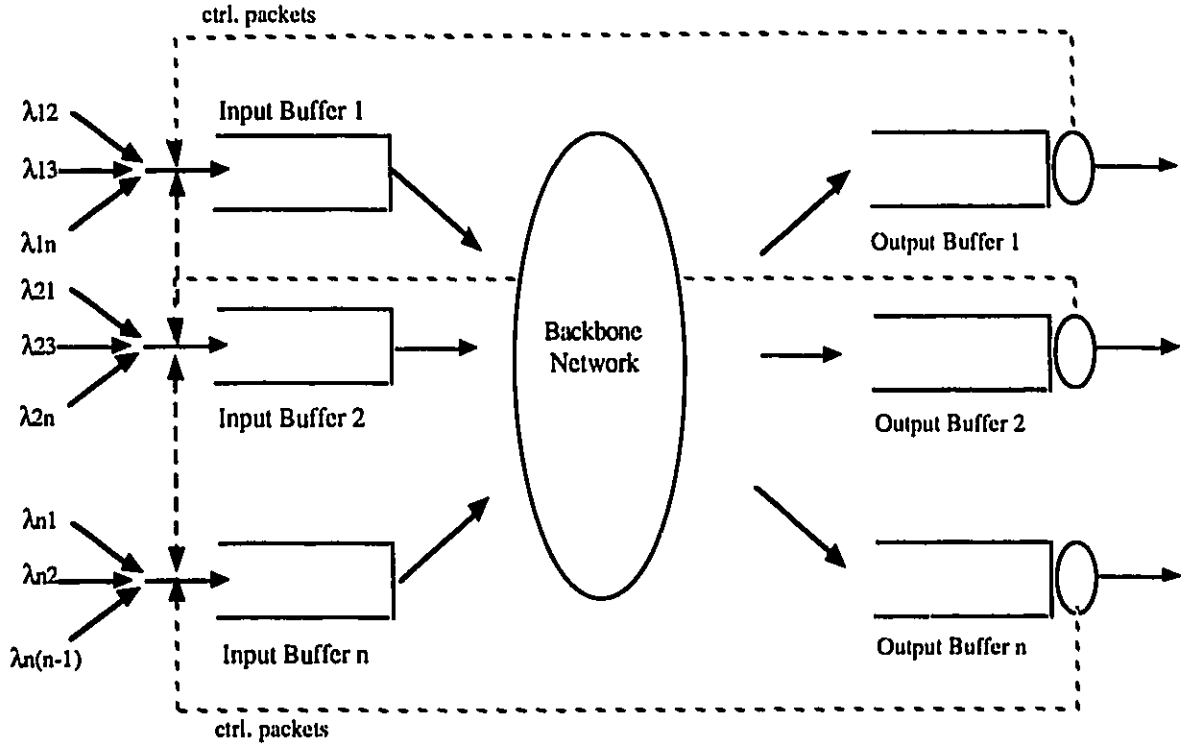


Figure 4.2 *Simulation model*

For simplicity, and because several studies have shown that the performance of a DQDB MAN is close to that of the M/D/1 queuing model, the MAN has been modeled as a single server with a deterministic service rate of mean μ_B [HAN92]. Under normal conditions, a packet having been transmitted through the backbone network will arrive at the destination bridge where it will be queued until it gains access onto the LAN connected to the bridge. The service rate of the output buffer is determined by the rate of successful access onto the LAN. Each output buffer is modeled as a single server with an exponential service time of mean μ^{-1}, [GUM92,WON89].

4.3.2 Assumptions

Throughout this work it has been assumed the use of a constant packet length, as well as error free transmissions. In order to study the effect of a traffic stream from a LAN over the traffic streams incoming from other LANs sharing the same destination bridge, a system comprising five LANs ($n = 5$) has been judged appropriate. By choosing $n=5$, it has been possible to evaluate the effect of a traffic from one LAN on the traffic from the other LANs. This should provide more accurate results by taking the average over a number of samples. The mean service rate of each output buffer has been set to $\mu=0.3$ *packets per unit time*. In this kind of systems, the capacity of the MAN is likely to be much higher than that of a LAN, or the rate of an output buffer, typically from ten to twenty times higher [WON89]. Therefore, μ_B has been set to 4.0.

The study has been mainly focused on the performance of the scheme when the output buffers (the potential system bottlenecks) are subjected to traffic loads of over 70%. Therefore, $\lambda_{ij} = 0.055, i \neq j \leq n$ which gives:

$$\lambda_j = \lambda_i = \sum_{\substack{j=1 \\ j \neq i}}^{5} \lambda_{ij} = 0.22 \qquad (6)$$

therefore

$$\rho = \frac{0.22}{0.3} = 0.73 \qquad (7)$$

where ρ, according to the queuing theory, is defined as the traffic intensity (λ/μ) or the utilization.

To study the effect of a misbehaving traffic stream, i.e. a traffic stream increasingly requesting for more output buffer space, the mean rate of the traffic stream going from LAN 1 to LAN 2, λ_{12} , has made varied from 0.05 to 0.15. For the sake of clarity, this traffic stream is referred to as T_{12}. As the traffic load λ_{12} is increased, the overall generation rate of the packets going to LAN 2 will exceed the access rate to LAN 2, through the output buffer. A backlog of packets will be created in the output buffer of the bridge interconnecting the MAN to the LAN.

The size of the output and input buffer, at each bridge, has been set to $K_o = K_i = 9$ packets. By assuming the use of a short-haul backbone network throughout this chapter, the propagation delay is considered negligibly small, and the upper and lower buffer thresholds have been set accordingly to $L_l = 1$ *and* $L_u = 9$.

By assuming all the traffic streams have the same traffic characteristics, and λ_{ij} is the same for all streams, the output buffers of all the bridges have been virtually divided into $(n-1)$ equal partitions, by setting:

$$S_{ij} = \frac{L_u - L_l}{n - 1} = 2 \qquad for \;\; i,j = 1,2,\ldots n \;\; and \;\; i \neq j \qquad (8)$$

4.4 Results and Discussion

The performance results of the proposed congestion control mechanism are compared to both the performance results of a system without any congestion control in place and those of the Simple Scheme. It is worth to notice that the Simple Scheme is equivalent to the Shortest Output Queue First Scheme when used in a high speed backbone network (cf. Section 3.4.2). Moreover, it is unsuitable to compare our scheme with the Longest Input Queue First Scheme. This is because the main goals of the schemes are incompatible. The goal of the Longest Input Queue First Scheme is to maximize the system throughput. This is done by favoring some traffic streams over others which clearly opposes the main goal of the proposed scheme.

For the sake of clarity in presenting the results, the system condition without any congestion control is referred to as Scheme A, the one implementing the Simple Scheme as Scheme B, and the Share-Based Control Mechanism as Scheme C.

The results that will be shown are all taken as the average of many points, either from the same simulation run or different runs. Each simulation have been executed to produce around 5×10^5 packets per traffic stream. That is, around 10^7 packets in total. The simulation model has been implemented by using the simulation language QNAP2 [POI84].

4.4.1 Packet Loss and Delay

Figures 4.3 and 4.4 show the packet loss probability for traffic T_{12} and T_{i2}, where $i = 3, 4$, and 5, as a function of λ_{12}. In Scheme A, packets at the input buffer are transmitted to the destination bridge as soon as they gain access to the backbone. By increasing λ_{12}, packets that belong to traffic T_{12} and T_{i2} are rejected by the output buffer of bridge 2 due to lack of space. This is reflected in an increase of the packet loss probabilities for traffic T_{12}

and T_{i2}. For Schemes B and C, packets destined to output buffers under congestion are held at their input buffers. This has a significant impact on the packet loss of traffic T_{12} and in particular traffic T_{i2}. By comparing Schemes B and C, it can be seen that Scheme C penalizes the misbehaving traffic, T_{12}, more in order to decrease its effect on the other traffic streams.

In Fig. 4.4, Scheme B fails to control the loss experienced by the traffic T_{i2} for high loads of T_{12}, whereas, Scheme C limits the loss of T_{i2} by penalizing the misbehaving traffic more. This effect arises because, in Scheme C, for higher demands on an output buffer, the transmission towards that buffer is stopped and resumed later by allowing each bridge to send its share value of packets first, to that output buffer. In Scheme B, packets are released in global FIFO order among all the inputs. This will allow higher demanding traffic to get the lion share out of the output buffer. This is not done without any price paid by other traffic sharing the same output buffer.

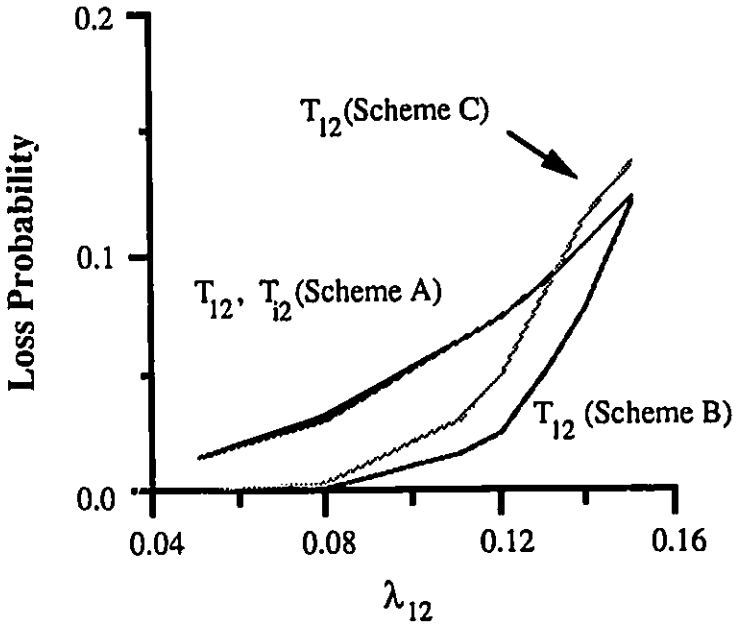


Figure 4.3 *T_{12} packet loss probability vs. λ_{12}*

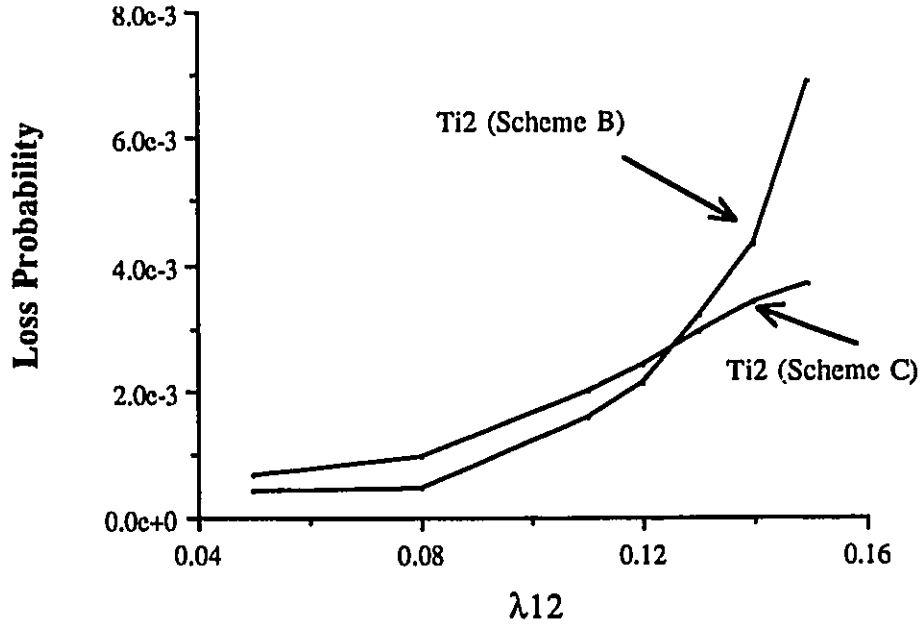


Figure 4.4 *T_{i2} packet loss probability vs. λ_{12}*

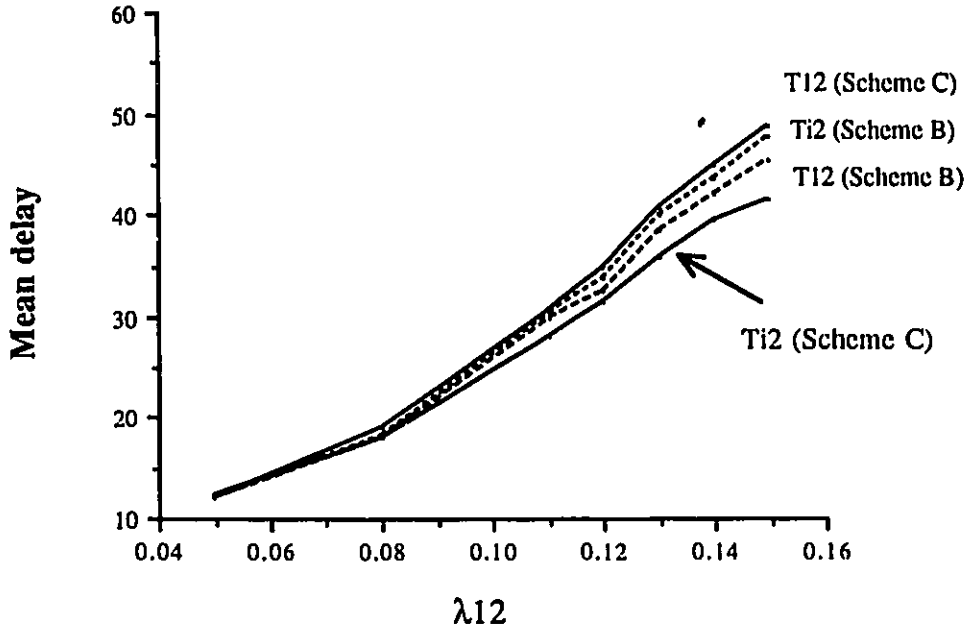


Figure 4.5 *Mean delay vs.* λ_{12}

In Fig. 4.5, it can also be noticed the price paid by traffic T_{i2} in terms of mean delay. Moreover, the delay experienced by traffic T_{i2} in Scheme B is higher than that of the traffic T_{12}. On the other hand, in Scheme C, traffic T_{i2} experiences lower mean delay compared to traffic T_{12} and to Scheme B, which seems more appropriate as the traffic load of T_{12} increases.

From the above results, the performance figures of Scheme C show better packet loss and mean delay of traffic sharing the same output buffer with a highly demanding traffic. It punishes the traffic generated by sources which increasingly request for more bandwidth.

In Schemes B and C packets destined to congested output buffer may stay waiting in the input buffer for some time. Those packets may build up in the input buffer and eventually block the entrance of any new arrival whether their destination output buffer is full or empty. This will result in almost equal packet loss probabilities for all the traffic streams arriving at the same input buffer, i.e., from the same LAN. To decrease this effect, the number of packets destined to a certain bridge could be restricted to a maximum number. This will save some space to the new arrivals at the input buffer. Fig. 4.6 shows packet loss probability for traffic T_{1j} (j=3,4,5) and T_{ij} (i=2,3,4,5, j=1,3,4,5 and $i \neq j$), with and without input restriction. The maximum number of packets has been set equal to 7.

To explain this, consider traffic T_{1j} for the Share-Based Scheme in Fig. 4.6. The packet loss probability without input restriction is much higher than that with input restriction. This is due to the fact that packets belonging to T_{12} are filling the input buffer of bridge 1 and blocking the entrance against new arrivals of either T_{1j} or T_{12}. It is also worth to note that T_{12} in both schemes will experience a bit higher loss due to this restriction. The lower the maximum number is, the higher the loss for T_{12} and the lower the loss for T_{1j} will be.

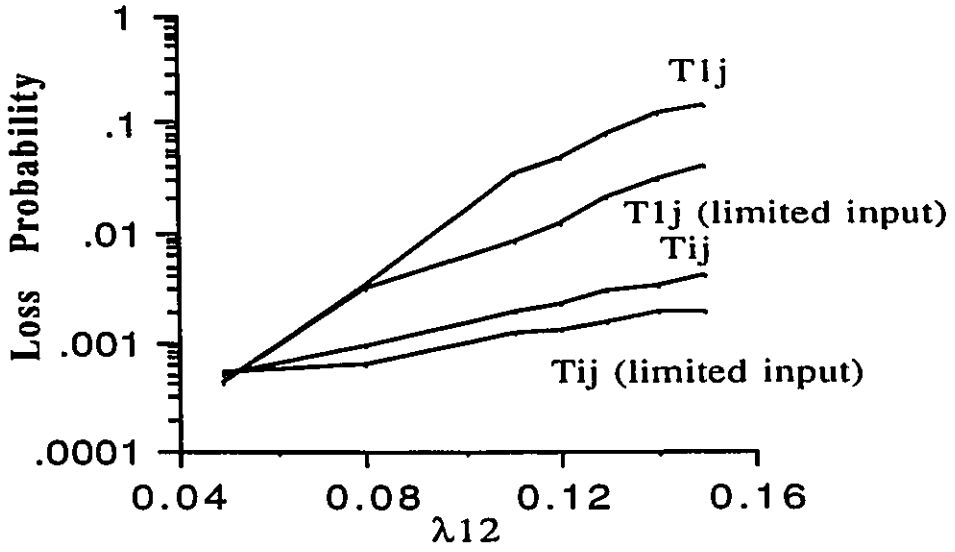


Figure 4.6 *Restricted input buffer occupancy*

One can argue that a kind of replacement scheme could be also used at the input buffers. This could decrease the packet loss experienced by T_{1j} or T_{ij} significantly. However, this will need at least two levels of priorities to be introduced at the input buffers, and may highly increase the complexity of the implementation. Therefore, the input restriction applied above, gives fairly good results given its simplicity in implementation.

4.4.1 Control overhead

The amount of overhead required for a control scheme to operate is another important feature when it comes to study the performance of such kind of schemes. In this study, the

control messages are supposed to be sent by using separate control packets. It has further been assumed that they are of fixed sizes as the data packets, and that there are always buffer space available for them.

Let R_i be the ratio of the number of control packets sent to the number of data packets received by bridge '*i*' from the other bridges. Fig 4.7 shows R_2 versus λ_{12} for both schemes B and C. Under low loads, R_2 for Scheme C is higher than that for Scheme B. This is due to the fact that Scheme C tries not to leave the output buffer idle. Under this scheme, the bridge is required to send a *SE-RF* message to indicate that there are enough space in its output buffer before this one gets empty. However, this high overhead may not be a problem compared to the high value of R_2 for high loads in Scheme B. This higher value of R_2 is because when the output buffer of bridge 2 becomes full, the bridge will send a message to the other bridges to stop the transmission. Another message will also be sent to resume the transmission when a space is made available, (cf. Section 3.2). During high loads, the output buffer will soon be filled again. The bridge is therefore required to send these two messages again, and so on. This feature makes of Scheme C particularly useful: the overhead introduced by the scheme during periods of high demands for resources is remarkably low.

It is interesting to analyze the distribution of the different control messages. In Scheme B, two messages are needed. These messages are the Buffer Full, *BF*, and the Receive Ready, *RR*. Under this scheme, a bridge issuing a *BF* message will transmit a *RR* as soon as a space becomes available. Therefore, 50% of the messages used by Scheme B are *BF* and 50% are *RR*. Under a condition of heavy load, *R* can reach a ratio as high as 2:1. This kind of situation may arise when each packet arrives at the output buffer causes the bridge to send two messages. One message is when the packet fills the buffer and another when it moves forward and leaves a space behind.

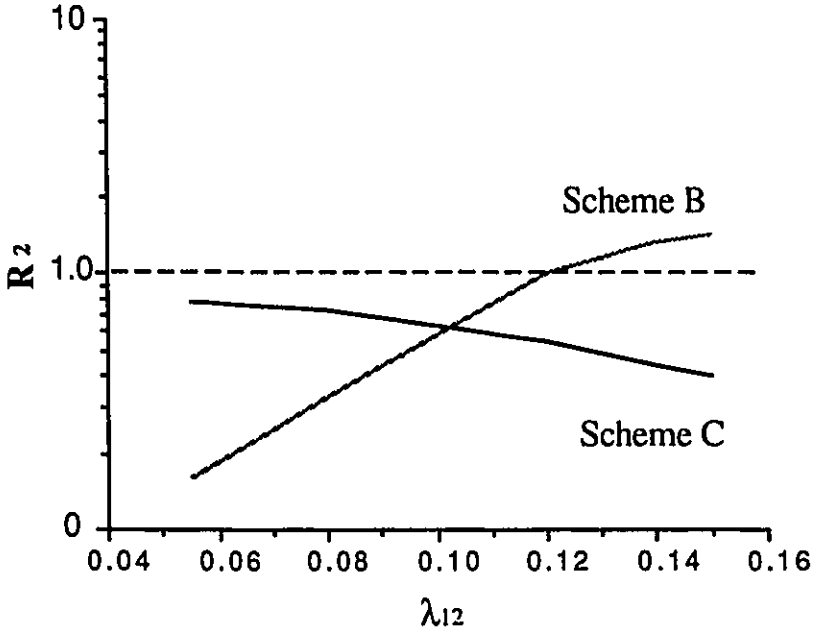


Figure 4.7 *R_2 versus λ_{12}*

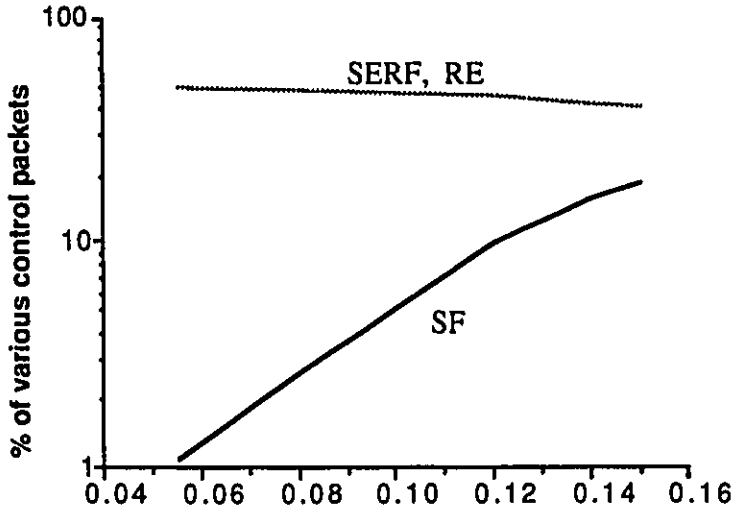


Figure 4.8 *Percentage of SF, SE-RF, and RE messages*

In Scheme C, three messages are required *RE, SE–RF, and SF*. When a *SE–RF* message is sent, it indicates that the output buffer is ready to accept already queued packets in the system or newly arriving packets. This message is usually followed by a *RE* in order to stop newly arriving packets from entering the backbone. This message is sent if the output buffer queue size starts to increase, otherwise it will leave the way open for newly arriving packets to enter the backbone. Concerning the *SF* message, it depends on the load on the output buffer. Figure 4.8 shows the percentage of all the three messages out of R_2 for Scheme C versus λ_{12}. As λ_{12} increases, more *SF* messages and less *SE-RF* and *RE* messages are sent.

It is worth to mention that in the Shortest Output Queue First Scheme, *R* is always equal to 1, since the output buffer informs all the other bridges with any change in its queue size. Thus, a control message will be sent upon receiving each packet. In the Longest Input Queue First Scheme, *R* is greater than one, since each output buffer directs the longest queue size input buffer to send a packet almost each time a packet departs from its buffer. In addition, each input buffer queue size should be known by all the bridges in the system.

4.5 Conclusion

In this chapter, a new congestion control scheme has been presented. This scheme has been designed to be used in systems interconnecting LANs to a MAN via bridges. The design aimed to define a simple mechanism to provide a fair share of the system resources. This is done by virtually partitioning the buffer space of the bridges interconnecting the MAN to the LANs.

A performance study via simulation of the proposed scheme was carried out. This study was focused on the mean delay and the packet loss probability. It also comprised an evaluation of the amount of overhead needed for this scheme to operate. The results show

that this scheme meets its main objective: a fair share of the output buffer space.

The scheme presented in this chapter relies on the information exchanged among the bridges (feedback). Different approaches are currently being undertaken in the area of congestion control of high-speed networks. One of the main approaches is the use of schemes which do not require any exchange of information. In the next chapter, this approach will be explored. A novel control scheme will be presented and evaluated. This scheme is characterized by the fact that no information exchange is required for controlling the sharing of the system resources.

Chapter 5
An Open-loop Congestion Control Mechanism for High-speed Networks

5.1 Introduction

In this chapter, a novel preventive congestion control scheme is presented. The main features of this new scheme are: its flexibility in setting the control parameters, in reacting to the traffic needs, and it aims to provide a fair sharing of the network resources. These features are particularly important for mechanisms aiming at high-speed networks. Current trends indicate that the use of preventive control mechanisms is the best choice for high-speed networks, [JAI90]. High speed packet switched networks have relatively high ratio of propagation delay to packet transmission time. Therefore, congestion control schemes based on feedback information may not work well in these networks [BAE90].

Furthermore, the ratio of processing time to packet transmission time has also been increased. As a result, protocol processing times have become a bottleneck in a high speed network environments. For this reason computational intensive control schemes are less desirable than simple schemes.

Some traffic requirements such as those for real time data that are time sensitive data, require rapid transfer, but the loss of small amount of information may be tolerable. On the other hand, data traffic can usually be slowed down to cope with the network congestion. It is likely that the real time nature of the traffic in broadband networks will require some level of bandwidth guarantee, which is estimated prior to the establishment of the connection.

New schemes are therefore required for congestion control in high speed environments. Recently, various congestion control approaches have been proposed, [WAN90, BOV92]. Most of them fall under the class of preventive control. After a connection is granted by an admission control scheme to a requesting user, the user may deliberately exceed the traffic volume declared at the time of connection set up, and may overload the network. Therefore once violation is detected, the traffic flow is enforced by discarding or buffering violating packets. For this reason, the bandwidth enforcement mechanism is needed to be implemented at the edges of the network.

In this chapter, a novel preventive traffic control scheme is proposed. In the following section, a brief overview on the main work in the area of traffic control for high-speed network is given. This overview mainly focuses on the Leaky Bucket mechanism and its variants. This mechanism has received a lot of attention in the recent years. The reason is simple: the Leaky Bucket is simple and particularly suitable for high-speed networks. Thereafter, a modified version of the Virtual Leaky Bucket Scheme as well as the new control mechanism are presented. In section 5.3, the implementation issues of these two latter schemes, the modified Generalized Leaky Bucket and the new scheme, are discussed. This is done particularly for the use of these schemes in a LAN/MAN environment. In

section 5.6 the performance of both schemes is studied by simulation. Finally, a conclusion summarizes the findings of the chapter.

5.2 Congestion Control in High-Speed Networks

The design of novel congestion control mechanisms is one of the main current issues in the area of high-speed networks. Current trends indicate that by using connection oriented services, network resources can be allocated and monitored in order to guarantee the overall grade of service required by the users. For this reason, many laboratories are focusing their efforts on the design of traffic control mechanisms. The design of simple bandwidth allocation and enforcement mechanisms is one of the most active research areas nowadays. In the sequel, an overview of the most relevant control mechanisms is presented.

5.2.1 The Leaky Bucket

The Leaky Bucket method is a typical bandwidth enforcement mechanism [WAN90, GAL89, TUR86]. This method is an end-to-end open loop preventive congestion control. It controls the traffic flow by means of tokens, which work as permits. In the original Leaky Bucket mechanism [TUR86], an input buffer was not provided. Later the use of an input buffer to hold incoming user information was suggested (see Figure 5.1). The integration of this buffer ought to provide a better control and tradeoff between waiting time and packet loss probability. Therefore, to achieve a lower probability of packet loss, packets may wait at the input buffer during periods of high loads.

The Leaky Bucket with a buffer states that an arriving packet enters a queue first. If the queue is full the packet is simply discarded. To enter the network a packet must first obtain enough tokens depending on its size from a fixed size token pool. If there are not

enough tokens, a packet must wait in the queue until enough tokens are generated. Tokens are generated at a fixed rate. This rate is set at the time of the connection setup and it depends on the traffic characteristics as well as the network load at the time of setup. In ATM or ATM-like environments, as in DQDB MANs, the packets are of fixed size and are called cells, thus each cell will request only one token.

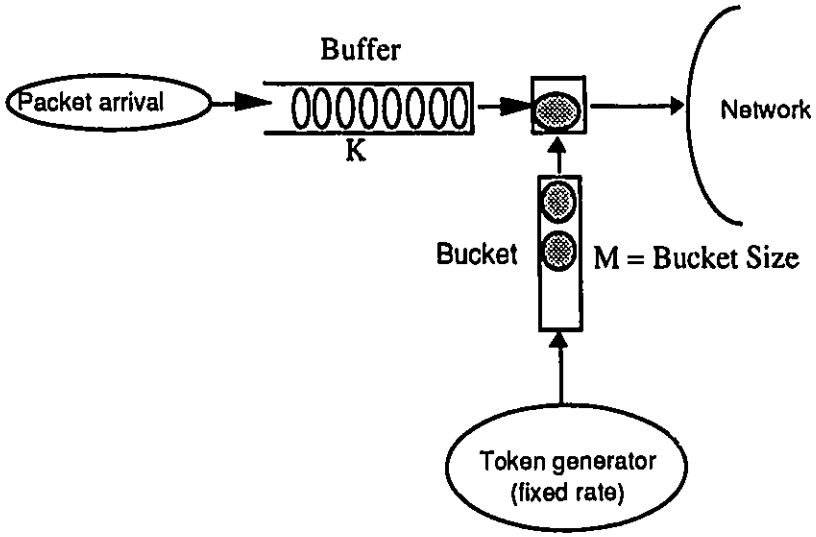


Figure 5.1 *Leaky Bucket with Input Buffer*

Because bandwidth may still be available in the network and unused, a marking method then was suggested for the original Leaky Bucket [GAL89, BAL90], and was called Virtual Leaky Bucket. The violating packets are permitted to enter the network with violation tags, rather than being discarded. By this, the mechanism allows a call or a connection to exceed its allocated bit rate, if it does not adversely affect other streams. A tagged or marked packet at a point reaching congestion will be *pushed out* so that the throughput of the unmarked packets is not significantly affected (Figure 5.2). This allows not only to take advantage of the unused bandwidth, but also a larger margin of error in determining the control parameters.

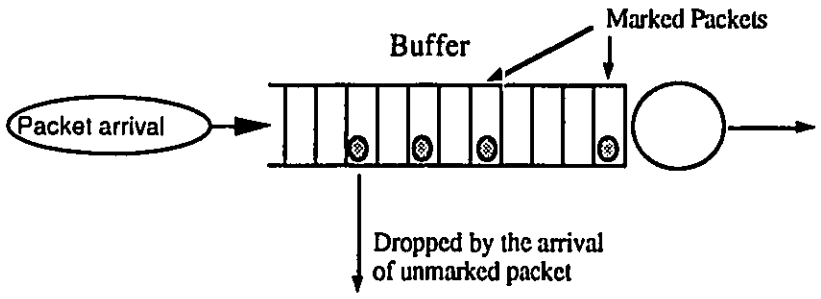
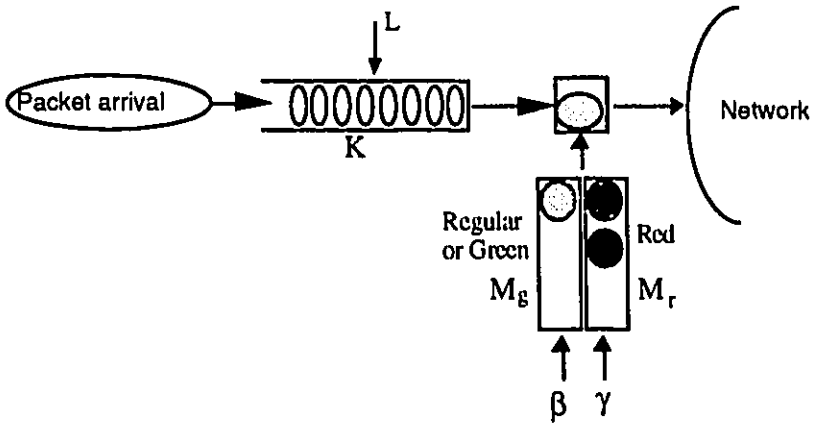
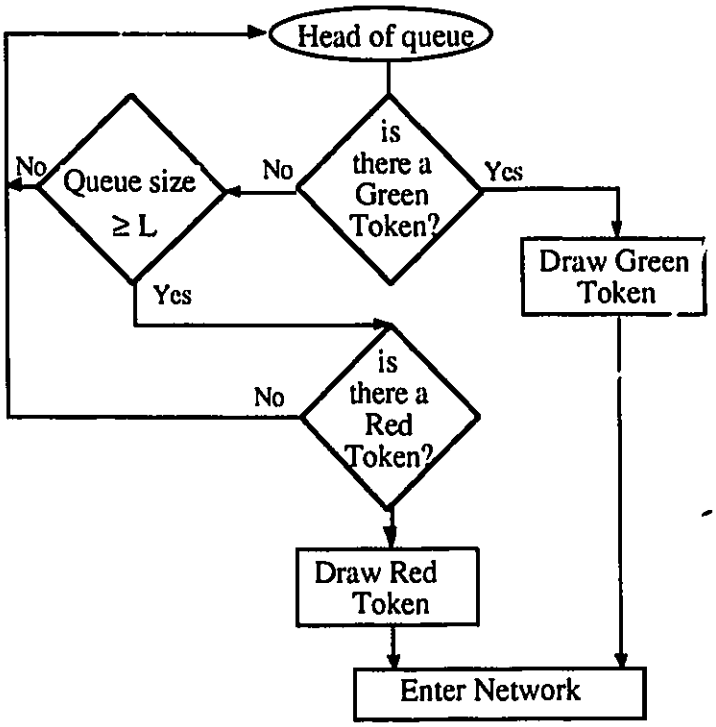


Figure 5.2 *Dynamic sharing between marked and unmarked packets*

In reference [BAL90], the Generalized Leaky Bucket Scheme was proposed. Under this scheme, a packet may draw a *Red* token only if the input queue has reached a certain level. These *Red* tokens are another type of tokens besides the regular or so called *Green* tokens. They are generated into a separate pool. Packets that could not draw a *Green* token and drew a *Red* token instead, are tagged and are considered as marked packets. Packets are allowed to draw *Red* tokens if and only if the queue size of the input buffer increases to a certain level. In other words, the Generalized Leaky bucket states that arrivals unable to draw *Green* tokens are queued in the input buffer. If the queue size reached a certain level, L, packets are then allowed into the network by drawing *Red* tokens from a separate token pool. These packets are then considered as marked packets. Marked packets are subject to be discarded at any point suffering from congestion. In the Generalized Leaky Bucket mechanism, *Green* and *Red* tokens are generated at fixed rates of β and γ respectively, as shown in Figure 5.3a. The mechanism is also summarized in Figure 5.3b. In this mechanism, the threshold L acts as an indicator for an additional bit rate or bandwidth needed by a source.



(a)



(b)

Figure 5.3 *Generalized Leaky Bucket*

5.2.2 The Multithreshold Mechanism: A new control scheme

The scheme presented herein is basically an open loop control mechanism. The traffic is enforced at the entrance of the backbone network. Hence, no feedback information is needed from the destination, or any of the intermediate stations. It controls the traffic by means of tokens, as in the Leaky Bucket. The scheme requires a buffer of size K, where the arrivals are stored while waiting to enter the network (Figure 5.4). Tokens that are generated are stored, if not used, in a finite size token pool of size M. If the pool is full, new generated tokens are discarded. For a variable packet length, in order to enter the network each packet should draw a certain number of tokens depending on its size. Each token corresponds to a number of bits called *quota*. Throughout this work, the use of a fixed size packets is assumed. Therefore, a packet requires to draw only a single token.

Tokens are generated at a fixed rate of β token per time unit. This rate, which should range between the mean and the peak rate of the traffic, as well as M, which represents the maximum number of packets allowed back to back into the network, i.e., without waiting, are set at the time of connection setup. These control parameters depend on the network load, the destination capacity, and the different characteristics of the traffic specified. Drawing a token is a way of getting permission to enter the network. A packet that is not being able to draw a token is held at the input buffer until it can draw one. New arrivals are simply discarded when the buffer is full.

In the case when the queue size reaches a certain threshold, L_1, a separate token generator starts to generate *Red* tokens at a rate of γ_1, otherwise the *Red* token generator is shut off with $\gamma_0 = 0$. Packets drawing *Red* Tokens are considered as marked packets, and may be discarded at any point in the network suffering from congestion. When the queue size increases further up to a level L_2, the generation rate of the *Red* tokens is increased to γ_2, and so on, up to a preset maximum number of levels, N. The input queue size will act

then as our *input congestometer*, and the various levels of thresholds give indications of the amount of bandwidth required by the traffic.

Bandwidth enforcement is achieved by setting β and M at the time of setting up the connection. By introducing the *Red* token generator, more flexibility is allowed by allowing a larger margin of error in setting M and β. By introducing various *Red* token generation rate, further flexibility is allowed. Inadequate parameter settings may result from inaccurate declaration of the traffic characteristics by the source, inaccurate estimation of the network load by the admission control mechanism, or network load change during the call.

Moreover, only a maximum number of packets, M, is allowed back to back into the network, while smoothing the entrance of the packets following immediately. Spacing between these packets will be decreased gradually as more bandwidth is required by the traffic, but at the same time not affecting other traffic. This is because the extra bandwidth which was granted to the demanding traffic, by allowing more packets into the network, is considered as *unsafe* bandwidth. This unsafe bandwidth is subject to withdrawal if no extra bandwidth is available, by dropping the marked packets allowed into the network.

The destination or intermediate buffers will be dynamically shared by marked and unmarked packets up to their full capacity. Any packet can be stored in a buffer if a space is available. However, marked packet are *pushed out* by unmarked ones. This will guarantee that even if marked packets are stored without restrictions, no unmarked packet is lost, unless the buffer contains only unmarked packets. The buffer, however, must be accessed by pointers organized in FIFO order, in order to guarantee packets correct sequencing. As suggested in [GAL90], the processing speed could be enhanced by tabling in an auxiliary memory the location of each marked packet, together with the identifier of its predecessor, making the push out process as simple as a record elimination from a list.

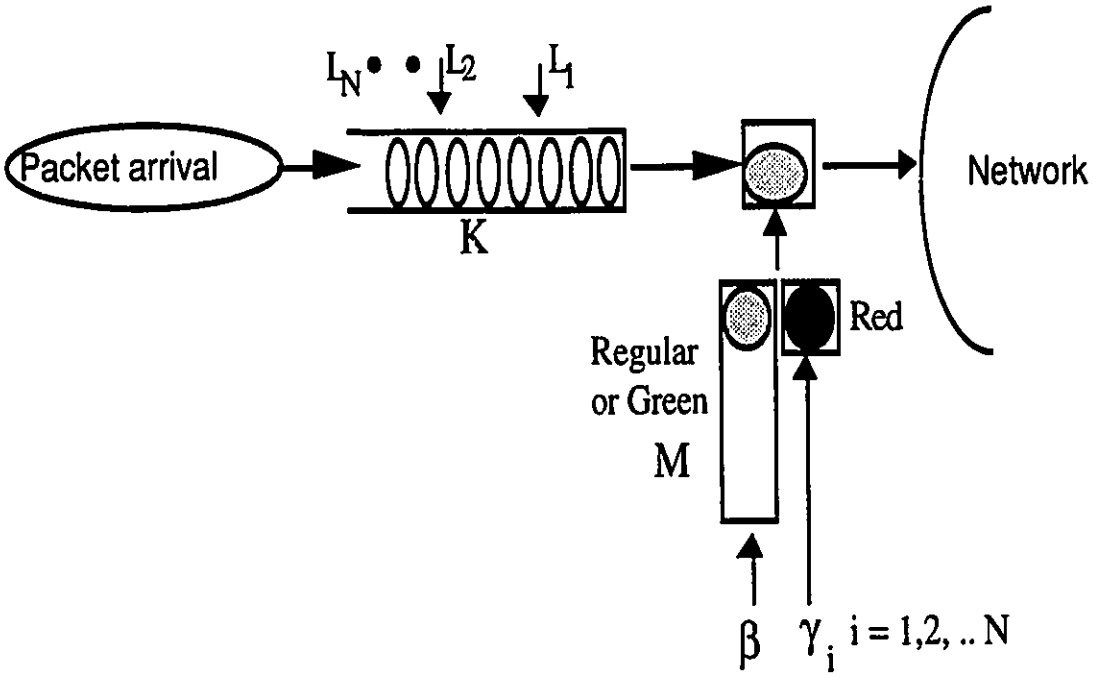


Figure 5.4 *The Multithreshold Scheme*

The way tokens can be implemented is to replace the token pool with a register of Log_2 (M+1) bits, for the *Green* tokens. The register will contain a number which indicates that of available tokens. This number is decremented when a packet draws a token, and incremented when a token is generated. Token generator is thus replaced by a counter with a maximum number of M. Thus, the register will be incremented at a rate of β.

5.2.3 The Multithreshold Scheme vs. the Generalized Leaky Bucket

There are two main features distinguishing the Multithreshold Scheme from the original Generalized Leaky Bucket: 1) the introduction of a multithreshold buffer, and 2) the use of multiple *Red* token generation rates.

In the Multithreshold Scheme, the *Red* token generation rate is increased every time the queue size reaches a given threshold, and so on up to a limit. In the Generalized Leaky Bucket, the threshold in the input buffer is unique and the rate of *Red* token generation is fixed.

In other words, the *Red* token generation in the new scheme follows the needs of the traffic for more bandwidth gradually. Thus the rate of marked packets arrivals at the destination station and at any intermediate one is a true indication for the need of extra bandwidth by a given source. This may provide the means to indicate to the control mechanism when to reset the values of the control parameters on a per connection basis.

In addition, the Multithreshold mechanism shows more flexibility in setting the control parameters. Consider a case when β is set small compared to the traffic arrival rate. Since β represents the allowed rate, packets will start to build up in the input buffer. At the time the queue size reaches L_1 extra bandwidth is allowed using *Red* tokens to enter the

network. Further bandwidth is also allowed if the queue size is increased further, and so on. On the other hand if β is set relatively large, *Red* token generation will be mostly shut off. Thus, the number or rate of marked packets received by the destination node in both cases may be used for increasing or decreasing β for a traffic.

5.3 Preventive Congestion Control Schemes for LANs/MAN Systems

5.3.1 The Multithreshold Scheme

This section focused on the use of the Multithreshold Scheme in a LAN/MAN environment. Figure 5.5 depicts the way this scheme can be implemented for controlling the traffic flowing through bridge 1.

At each bridge there is an *individual token pool* assigned for each destination bridge, with a size of M_{ij} and a generation rate of β_{ij}, where *'i'* and *'j'* represent the source and the destination bridge, respectively.

Each packet arriving at the source bridge *'i'* and destined to LAN *'j'*, enters first the input buffer. To be able to enter the backbone, each packet must first draw a token from the corresponding pool. *Priority is given to the packets in the input buffer with the largest number of tokens*. The number of tokens gives an estimation of the queue length at the destination output buffer. The larger the number of available tokens, the shorter the queue length of the destination output buffer may be.

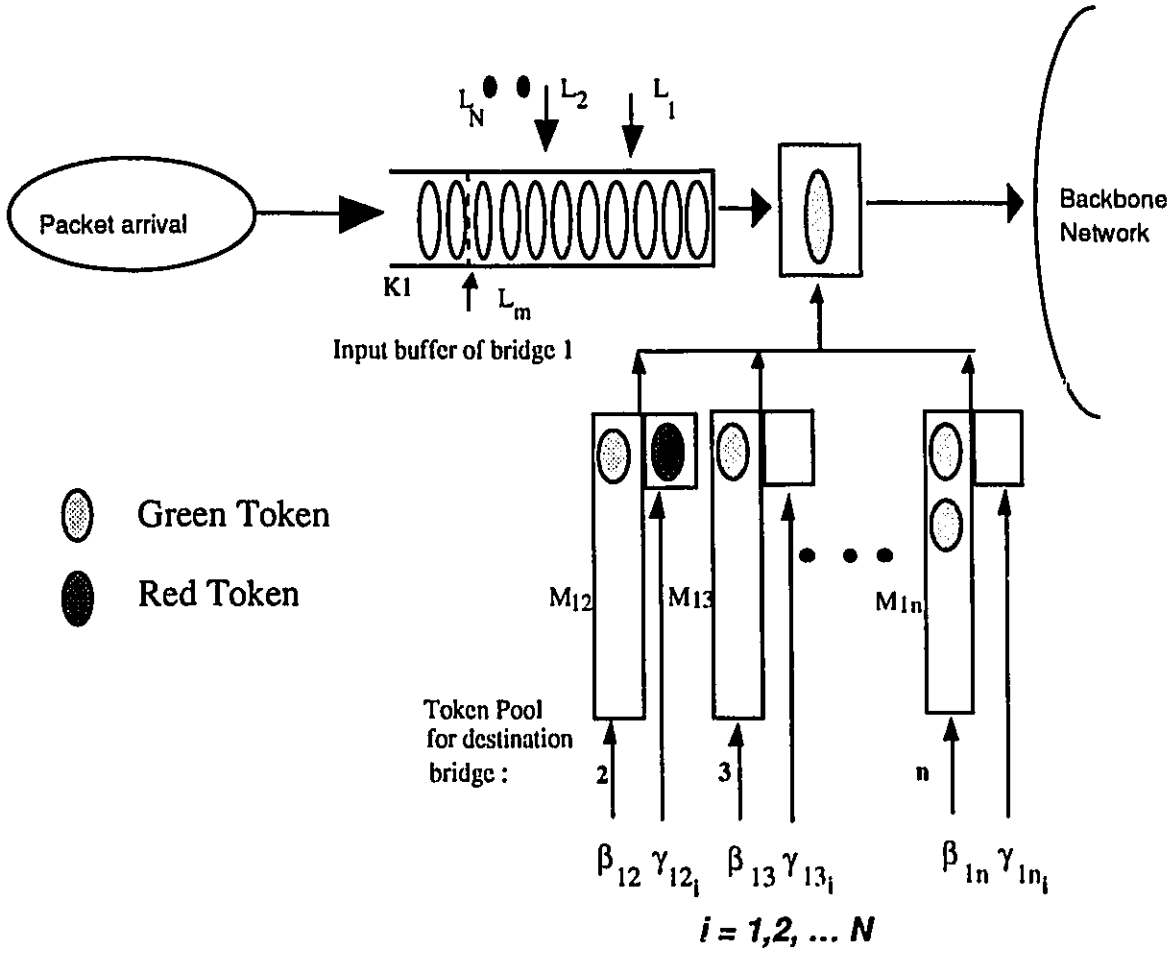


Figure 5.5 *The Multithreshold* Scheme *applied at bridge 1*

If no token is available, the packet is forced to wait in the input buffer of size K_i until a token is generated. When the number of waiting packets destined to bridge *'j'* in the input buffer of bridge *'i'* reaches a threshold L_{ij1}, the corresponding *Red* token generator will be turned on, with an initial rate of γ_{ij1}. When the number increases further and reaches L_{ij2}, the rate of the corresponding *Red* token generation will be increased to γ_{ij2}, and so on up to a maximum level, N_{ij}. Packets drawing *Red* tokens are marked and are subject to rejection from entering a full output buffer or pushed out by unmarked packets. Packets pushed out from the buffer are selected in LIFO order. This has been shown to be a good policy in selecting lower priority packets to be pushed out from a buffer, [FON93].

To prevent one traffic from dominating the input buffer and blocking the way against new arrivals of other traffic streams from the same LAN, one may restrict the maximum number of packets destined to a certain bridge to a maximum number L_m. Thus, new arrivals with the same destination exceeding this limit will be discarded.

At the source bridges, traffic T_{ij}, i.e. traffic from LAN *'i'* to LAN *'j'*, may represent either a traffic from a single source or a group of sources connected to LAN *'i'* and having destinations connected to LAN *'j'*. In either case, the parameters of the control mechanism is either set or modified upon each connection establishment. By applying the scheme at the bridges, a way of bandwidth borrowing among users, from the same LAN and sharing the same destination LAN, is allowed; a heavy traffic from one source is allowed to benefit from the unused bandwidth, or the available tokens, of another source.

5.3.2 The Modified Generalized Leaky Bucket

In the same manner as the Multithreshold Scheme, the Generalized Leaky Bucket can

be used for controlling the traffic in a LAN/MAN environment. An individual set of token pools corresponds to one destination is found at the input buffer. One pool is for the *Green* tokens and the other is for the *Red* tokens. A higher priority is also given to the packets with the largest corresponding number of tokens. To prevent new arrivals to be rejected from entering the input buffer, when it is filled with one type of traffic waiting for tokens, the maximum number of packets with the same destination bridge is limited to L_m. Due to these modifications, this scheme will be referred to as the Modified Generalized Leaky Bucket, (Modified GLB).

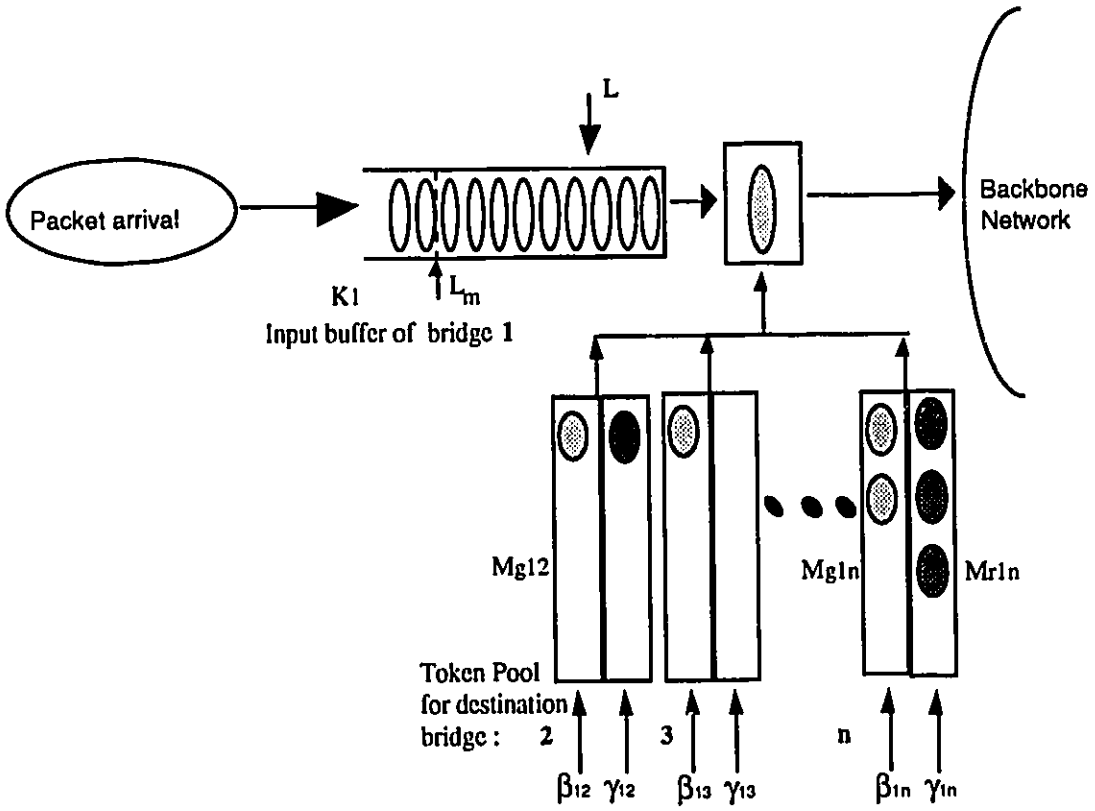


Figure 5.6 *The Modified GLB Scheme applied at bridge 1*

5.4 Performance Study

Figure 5.7 shows the queuing model used in the study. The main objectives of the study has been to evaluate the packet loss probability and the mean delay. The packet loss probability is defined as the ratio of the number of packets being lost to the total number of packets being generated per traffic stream. The mean delay is defined as the time elapsed from the time a packet enters the input buffer to the time the packet is accepted by the destination LAN.

5.4.1 Assumptions

The purpose of this study is to evaluate the performance of the new scheme when used in a LAN/MAN environment. The major assumptions are as follows:

- the system consists of multiple LANs interconnected by a MAN. In particular, a system consisting of four LANs and a MAN is considered, in order to be able to study the various traffic streams' behaviors. Each LAN communicates with all the other LANs through the MAN.

- a symmetrical model where all the output buffers are identical with mean service rate of $\mu = 0.3$ packets per unit time is considered. The MAN capacity is considered to be 10 times higher than the LANs capacity, or the output buffer rate. This is a typical ratio nowadays [WON89].

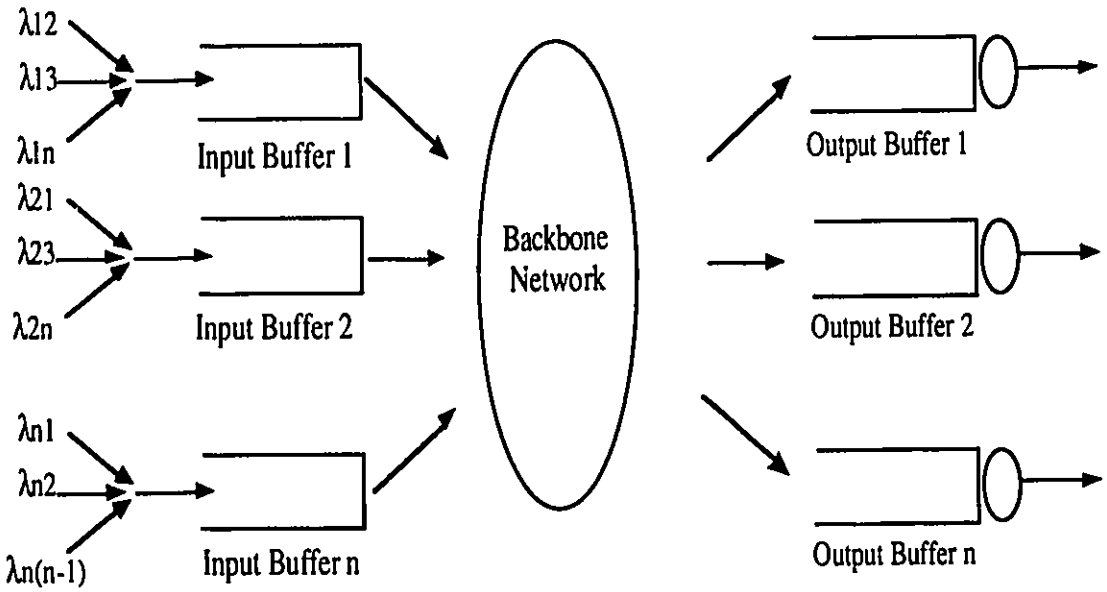


Figure 5.7 *Simulation model*

- The traffic generated by a LAN is assumed to be a Poisson process, and evenly directed to all the other LANs in the system, i.e.,

$$\lambda_{ij} = \frac{1}{n-1}\ \lambda_i \qquad (1)$$

where λ_i is the mean arrival rate of the total traffic generated by the sources on LAN *'i'*.

- the system is assumed to work under a heavy load, where λ_{ij} is set equal to 0.06 packets/unit time for i, j = 1,2,3,4 , $i \neq j$ and $(i,j) \neq (1,2)$ .i.e.,

$$\rho_j = \frac{\lambda_j}{\mu} = \frac{0.06 \times 3}{0.3} = 0.6 \text{ for output buffer of bridge 'j' = 1, 3, and 4} \quad (2)$$

The case when a traffic is violating its allowed bandwidth is taken into consideration. In particular, T_{12}, traffic from LAN 1 to LAN 2, is thus chosen to be the "offender" traffic. Therefore, λ_{12} is made to increase from 0.06 to 0.15 packets/unit time, i.e., ρ_2 of output buffer of bridge 2 is increased from 0.6 up to:

$$\rho_2 = \frac{0.06 \times 2 + 0.15}{0.3} = 0.9 \text{ for output buffer of bridge 2} \quad (3)$$

The main goal by doing this is to see the effect of a violating traffic, when its arrival rate is equal to some value over other traffic sharing the same destination path.

- considering that the ratio between the LANs and MAN's capacity is 1:10, the sizes of the output and input buffers at each bridge have been set to $K_o = K_i = 12$. Furthermore, the use of two levels (thresholds) have been considered, L_1 and L_2, and have initially been set to 4 and 6, respectively. Under normal conditions, all the traffic streams show the same arrival rate. In this case, each input buffer is equally shared among three traffic streams, i.e., 4 buffer spaces per stream. Otherwise, the control mechanism will react to the increasing demand of a stream by activating the *Red* token generator.

  The second level has been set to permit an increasingly demanding source to request for more space. This is detected at the time that the traffic stream occupies more than half the buffer size, i.e., 6 buffer spaces. Any traffic with a certain destination at an input buffer is restricted to a maximum of $L_m = 9$ buffer spaces at any time.

- β_{ij} , the generation rate of the *Green* tokens should have a value between the declared peak and the mean rate of the traffic T_{ij}. Thus, β_{ij} should be greater or equal to λ_{ij}. M_{ij} will be set to the maximum number of packets allowed back to back into the backbone

network without any control. For the case of symmetry and Poisson arrival process, granting guaranteed rate of approximately 125 % of the declared mean rate, i.e., $\beta_{ij} = 1.25 \times \lambda_{ij}$, and $M_{ij} = 2$, for all $i \neq j$, are sufficient under normal conditions.

The simulator has been developed by using the programming language QNAP2 [POI84]. Each run roughly generates 5×10^5 packets per traffic, i.e., around 6×10^6 packets in total.

5.4.2 Results and Discussion

In this section, the results of the performance study are presented. Because of the symmetry case considered, the subscripts *i* and *j* in the following will be dropped where obvious.

Figure 5.8 shows the packet loss probability as a function of the load when no control is applied. In this case, packets are generated by the sources and transmitted from the input buffer to the destination bridge as soon as they gain access to the backbone. By increasing λ_{12}, the overall rate of all the traffic going to LAN 2 increases leading to an increase in the number of packets waiting in the output buffer of bridge 2, and eventually filling it. This results in a high packet loss probabilities for traffic T_{12} and T_{i2}, where i=3,4. All the other traffic streams are practically unaffected. Notice that T_{1j} does not experience any change. This is due to the high capacity disparity between the LANs and the MAN. Packets go across the MAN fast, while they get blocked at the output buffer. This does not affect at all T_{1j} which shares only the input buffer with the "offender" traffic T_{12}.

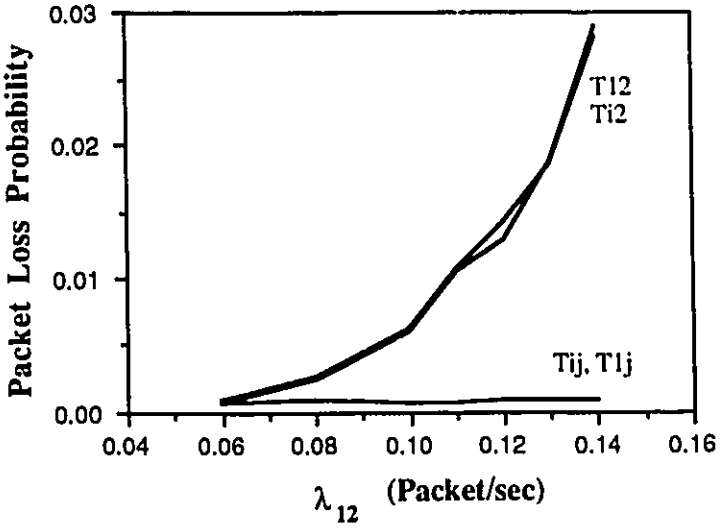
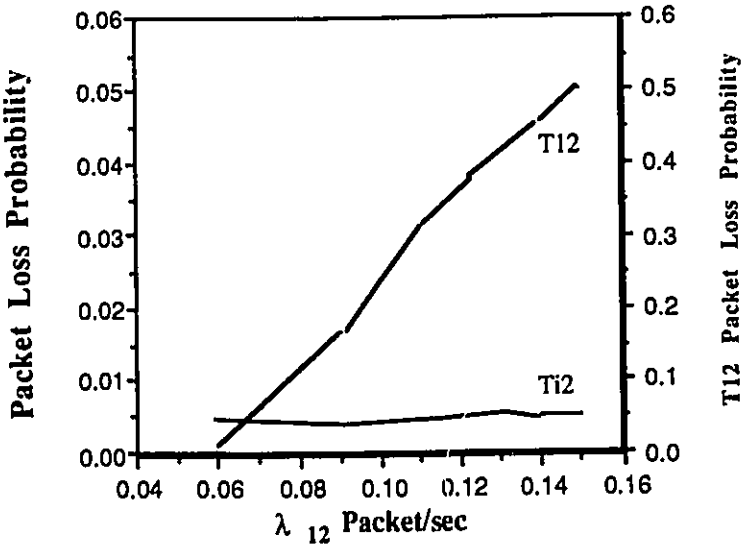
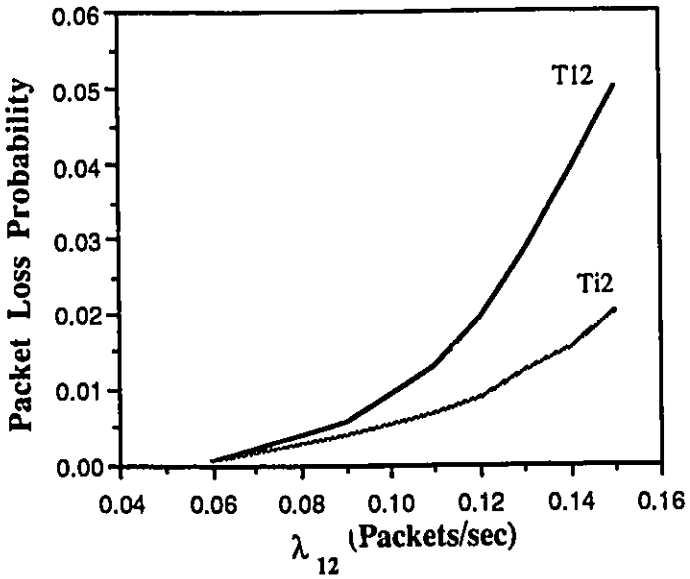


Figure 5.8 *Packet loss probability vs load when no control is applied*

In Figures 5.9a and 5.9b, the performance of two different schemes are shown. In the first scheme, packets are allowed to enter the backbone network if and only if they can draw *Green* tokens. This is referred to as the Leaky Bucket with Input Buffer scheme. In the second scheme, all packets are allowed into the backbone network without any restriction, as if no control is applied. However, those packets which could not draw *Green* tokens are considered as marked packets. They are also subject to be pushed out at the output buffers. This second scheme is referred to as the Virtual Leaky Bucket. These two schemes are also described in Section 5.2.1. The rate of token generation in both schemes have been fixed to $\beta=1.25\ \lambda_{ij}$.



(a)



(b)

Figure 5.9 a- *Leaky Bucket with Buffer*, b- *Virtual Leaky Bucket*

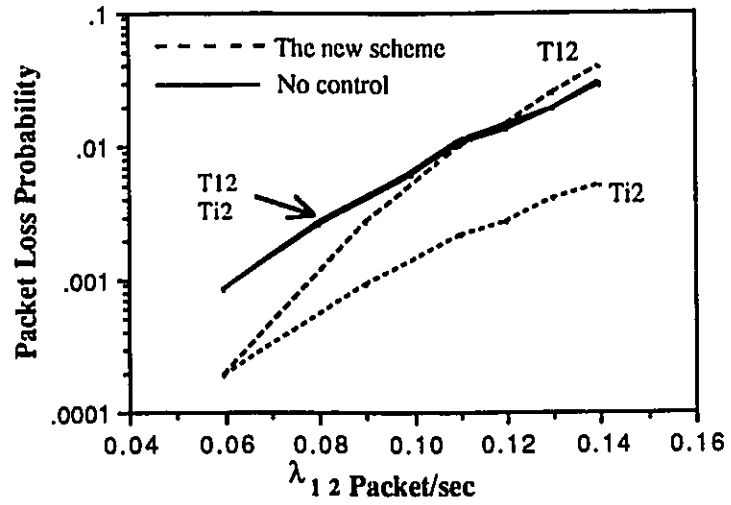
In Fig 5.9a, T_{i2} is not affected by increasing the load of T_{12}. The only loss it suffers is at the input buffer during the periods where its arrival rate exceeds β, due to the randomness of the arrival Poisson process. This randomness allows the traffic to exceed the allowed rate β from time to time. However, unused bandwidth or buffer space at the output buffer may be wasted, since packets are discarded mostly at the input buffers without considering whether a particular output buffer is full or not. This explains why T_{12} is much more penalized by its increasing rate as compared to the second scheme where no buffer space is wasted.

In the second scheme, the Virtual Leaky Bucket Scheme, all the packets are allowed into the backbone as marked and unmarked packets. Thus, packet loss is only due by the lack of space at the output buffer. However, as more and more packets are utilizing the LAN, due to a load increase of one traffic, the other traffic is affected as well. This will lead to an unfair control, see Fig 5.9b. By comparing these two figures and looking in particular at the loss experienced by T_{i2}, the Leaky Bucket with Input Buffer Scheme exhibits a lower T_{i2} packet loss than the Virtual Leaky Bucket for this type of system configuration.

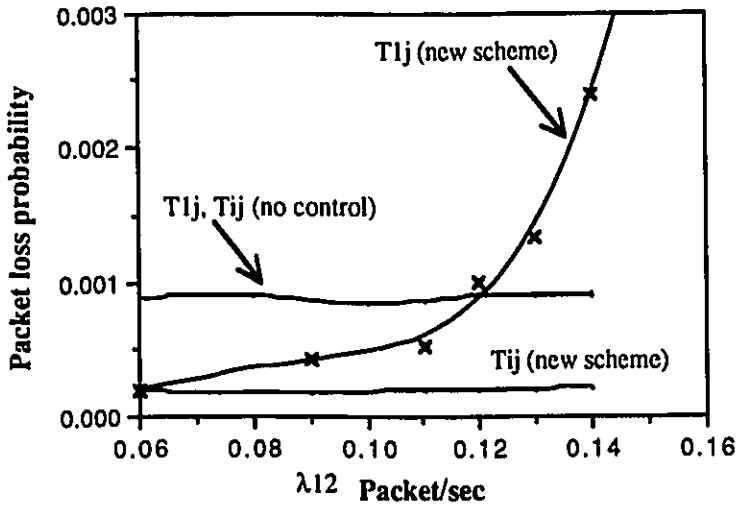
The proposed Multithreshold Scheme simply gives a kind of tradeoff between the above performances of the mentioned two schemes.

Figures 5.10a and 5.10b show the packet loss probability as a function of the load when no control is applied, compared to the Multithreshold Scheme for traffic T_{i2} and T_{12}, and for traffics T_{ij} and T_{1j}, respectively. The *Red* token generation rate has been set to $\gamma_1 = \beta$ and $\gamma_2 = 1.5\,\beta$. In the Multithreshold Scheme, packets that cannot draw tokens are held at their input buffers, until either they draw *Green* or *Red* tokens. All the traffic directed to LAN 2 from all the other LANs but LAN 1, i.e., traffic T_{i2} i = 3,4, benefit from the fact of holding the packets belonging to the violating traffic T_{12} at bridge 1. Thus, this scheme decreases the effect of the traffic increasingly requiring more bandwidth on the other traffic

sharing the same output buffer. Indeed, this is because *the Multithreshold Scheme* heavily penalizes traffic T_{12} as compared to the 'No Control' Scheme for very high loads.



(a)



(b)

Figure 5.10 *Packet loss probability vs.* λ_{12}
a)- T_{12} *and* T_{i2} *b)-* T_{1j} *and* T_{ij} *for both 'No Control' and the Multithreshold scheme*

In Fig 5.10b, T_{ij} experiences less loss and it is not affected by the increase of λ_{12}, whereas T_{1j} suffers some loss due to the fact that the input buffer of bridge 1 is mostly occupied by packets belonging to T_{12}. This effect could be decreased by decreasing L_m, the maximum number of packets, with the same destination, allowed into the input buffer. This will result in varying the loss probability of T_{ij} towards the one obtained when using the Leaky Bucket with Buffer Scheme (cf. Figures 5.9a), that is, lowering the loss probability of T_{i2} and increasing that of T_{12}. It could be also decreased by decreasing the values of either or both of the indication levels L_1 *and* L_2 . This, however, will result in varying the performance towards that of the Virtual Leaky Bucket (cf. Figures 5.9b), which has an opposite effect on both T_{i2} and T_{12}, compared to the first option; T_{i2} will experience higher loss probability, whereas, T_{12} will experience less.

In the following figures, the packet loss probability of each of the traffic versus λ_{12} in the Multithreshold and the Modified Generalized Leaky Bucket, (Modified GLB), schemes are shown. For the Multithreshold Scheme, Figure 5.11, γ_1 is set equal to β and γ_2 to 1.5β, as already stated, and the values of the thresholds are $L_1 = 4$ and $L_2 = 6$. Whereas, in the Modified GLB, L is set to L_1 *of* the Multithreshold Scheme, i.e. 4, and $M_r = M = 2$. The performance of the Modified GLB for the two values of the *Red* token generation rates are shown. In Figure 5.12, γ has been set equal to γ_1, and in Figure 5.13, γ has been set equal to γ_2.

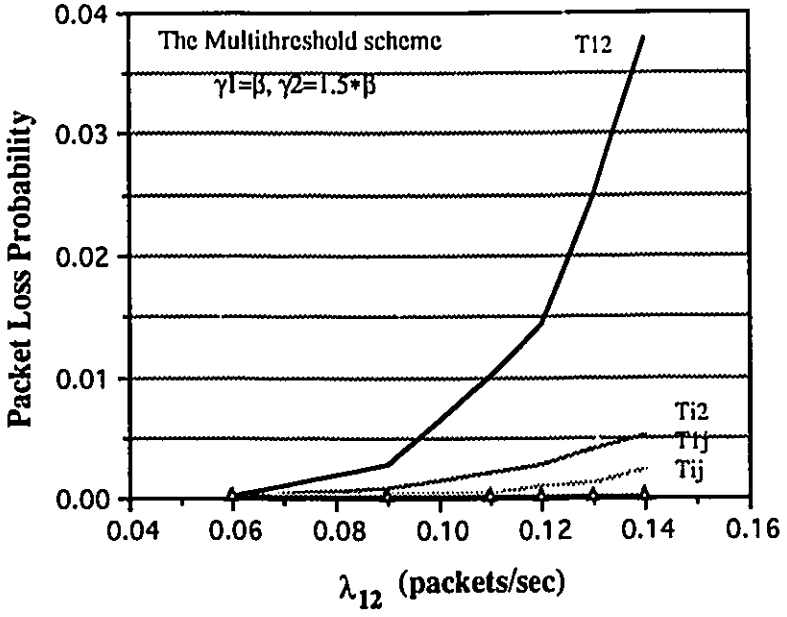


Figure 5.11 *Packet loss probability vs. λ_{12} (The Multithreshold Scheme)*

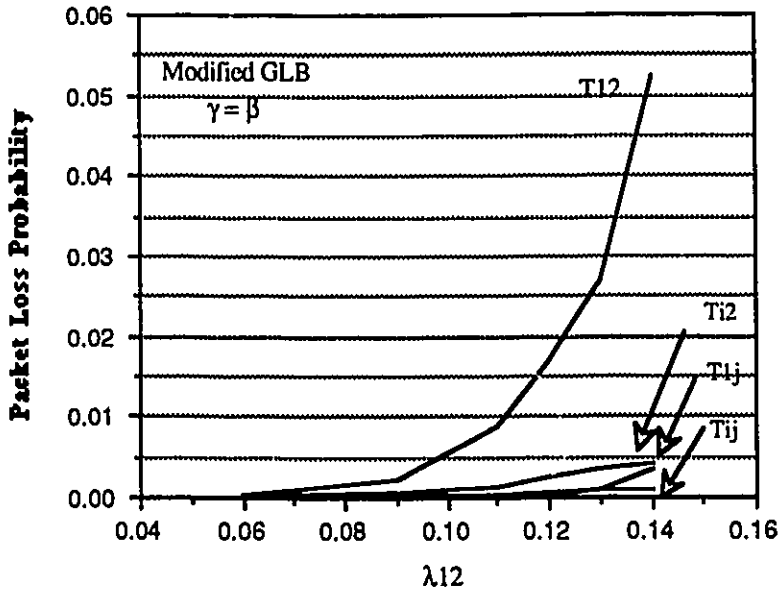


Figure 5.12 *Packet loss probability vs. λ_{12} (Modified GLB, $\gamma = \beta$)*

In Figure 5.12, γ in the Modified Generalized Leaky Bucket Scheme is relatively small and equal to β. The scheme performs better in lower loads and up to $\lambda_{12} = 0.13$. Beyond this point, the packet loss is dramatically increased at the input buffer of bridge 1 for both traffic, T_{12} and T_{1j}, due to the fact that the control mechanism used does not react to the extra needs of the traffic. Therefore, when T_{12} increases its arrival rate to 0.14, packets will start to build up quickly at the input buffer of bridge 1. Therefore packets arriving from LAN 1 will be blocked. This is because the maximum allowed rate into the backbone of both marked and unmarked packets for $\gamma=\beta$ is approximately $\beta + \gamma = 2\beta = 0.1375$. Therefore, this explains the high packet loss experienced by the traffic exceeding this value.

In Figure 5.13, γ is fixed to a relatively high rate γ_2. The *Red* tokens are almost available when needed, this leads to more marked packets accessing the backbone network. In this case the performance will vary towards that of Virtual Leaky Bucket, affecting T_{i2} more in terms of loss probability

By comparing the results shown in Fig 5.11 to 5.13, the Multithreshold Scheme shows that with one setting of parameters, its performance lies in between the above described settings of the Modified GLB Scheme. Furthermore, the use of a multirate token generator adds a higher degree of flexibility. This makes this mechanism more suitable for applications requiring an adaptive control mechanism.

Figure 5.14 shows the mean delay experienced by the traffic in the three cases discussed above. Most traffic streams have almost the same mean delay. However, T_{12} in the Modified Generalised Leaky Bucket Scheme when $\gamma = 1.5\beta$, experiences lower delay because of the less time its packets have to wait for tokens to enter the backbone network. Therefore, the Multithreshold Scheme offers a kind of tradeoff between the mean delay and the packet loss probability.

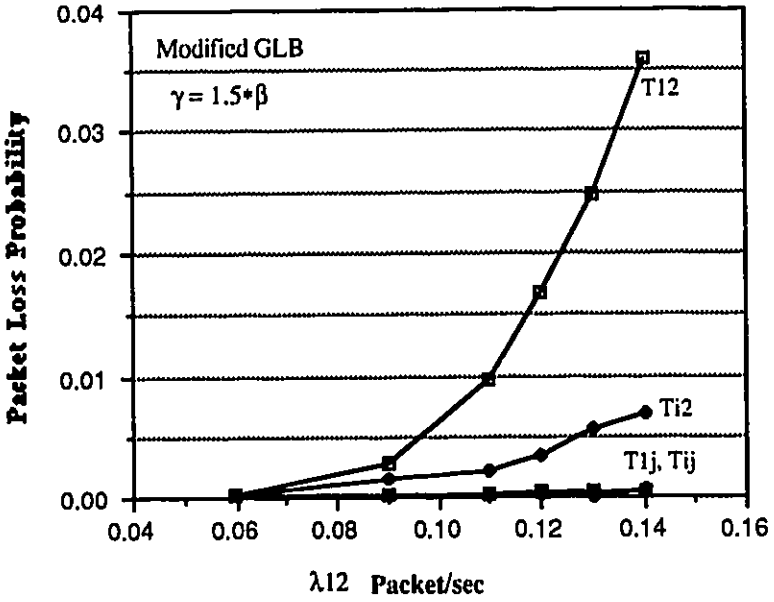


Figure 5.13 *Packet loss probability vs. λ_{12} (Modified GLB, $\gamma = 1.5\beta$)*

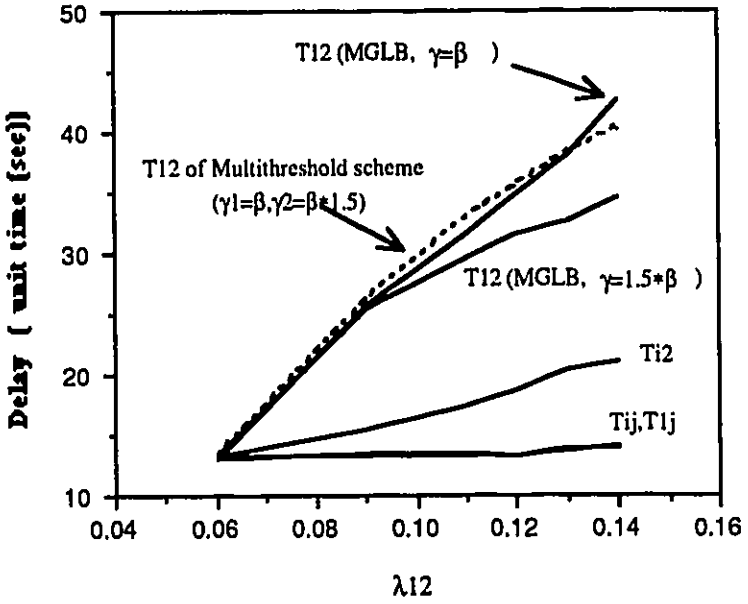


Figure 5.14 *Delay vs. λ_{12}*

In Fig 5.15, the packet loss probability is shown versus λ_{12} for the new scheme. In this figure the allowed bit rate is decreased to 1.17 of the mean rate of the traffic. The randomness in the Poisson process will cause more violation to be considered by various traffic, for a lower allowed bit rate. More violation means more packets being discarded during periods of full output buffers. This explains the higher loss experienced by the various traffic.

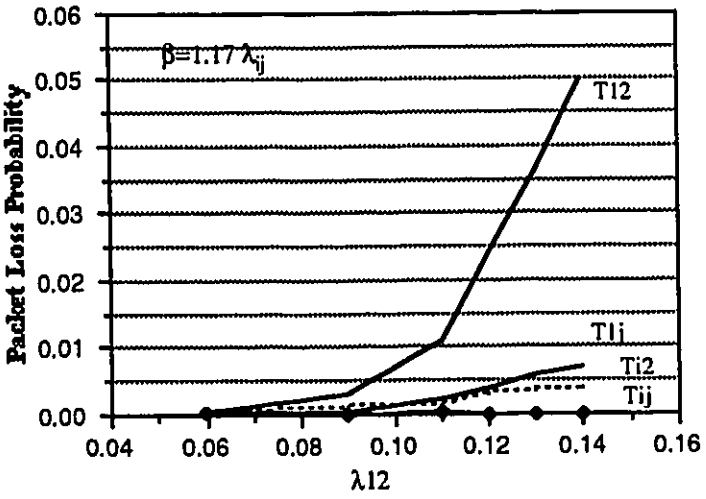


Figure 5.15 *Packet loss probability vs.* λ_{12} *(* $\beta = 1.17\ \lambda$*)*

5.5 Conclusion

In this chapter two novel preventive congestion control mechanisms have been presented, namely: the Modified Generalized Leaky Bucket and the Multithreshold Mechanism. These

mechanisms aim to provide a fair and efficient sharing of the network resources by continuously monitoring and enforcing the activity of the sources. Furthermore, these mechanisms are particularly suitable to be used in high-speed networks where the propagation delay may dominate over the transmission time as no information exchange among entities is required for the mechanisms to operate.

The use of the proposed mechanisms for controlling systems consisting of LANs interconnected to a MAN has also been investigated. This application is of particular interest as the deployment of these kinds of systems is seen as the first step to the deployment of the future Broadband-ISDN.

A performance study of the two schemes has also been conducted. The main results of this study have shown that these two schemes exhibit an excellent grade of flexibility in setting the control parameters by allowing a larger margin of error and in properly reacting to the needs of the sources.

Chapter 6
Conclusion and Suggestions for Further Research

In this thesis, we considered a MAN/LAN system. This system comprises a high speed backbone, a MAN, interconnecting lower speed LANs. We studied the buffer overload situations that may arise due to the speed difference between the MAN and the LANs. These buffer overload situations may occur at the interworking units (bridges) interconnecting the backbone MAN and the LANs.

We presented a few different mechanisms that have been proposed for such systems in the literature. We looked into each of these mechanisms separately. Their main common drawback was the unfairness of sharing the crucial resources among the sources, or users These resources are the output buffers space in this system, where the data units leaving the backbone have to wait to access the LAN attached. In addition to the high control overhead introduced by these mechanisms during high load periods.

To overcome these problems, we proposed two new mechanisms, and studied their

performances. This study was focused on the packet loss probability and the mean delay for the various traffic streams.

The first mechanism proposed is a share-based scheme. In this scheme, the crucial resources (buffers) are virtually partitioned among the various traffic streams into shares. This mechanism is also flexible enough to allow any traffic stream to take advantage of any unused buffer space. The performance of this mechanism was evaluated via simulation and compared with some other mechanisms. It has been found out that this mechanism not only decreases the effect of an offender traffic on others, but also decreases the amount of overhead required during high loads. This mechanism is particularly suitable for short-haul backbone networks where the latency in terms of propagation and processing times is not very high.

For long-haul backbone networks, an open loop mechanism has been proposed. This mechanism, the Multithreshold mechanism, is based on the Leaky Bucket Scheme. Under this scheme, packets should get permissions in order to enter the backbone. It turned out that this mechanism exhibits great flexibility in setting the control parameters, and in reacting to the traffic needs for more bandwidth over the Generalized Leaky Bucket, which was also studied.

Which mechanism is better? This question could be raised. It is obvious that the Share-Based Mechanism is only suitable for short-haul backbone since the bridges require to exchange information among each other. On the other hand, the Multithreshold mechanism is not only suitable for long-haul but also for short-haul networks. However, when short-haul networks are considered, to us it seems that the Share-Based Mechanism is better strategy. It provides the sender (the source bridge) with information about the situation at a specific destination bridge. Thus, there is really knowledge about what is going on in the stations on the way to the destination.

As follow up , it has been suggested to enhance the second mechanism to be able to reset the control parameters that were set at the time of connection setup, or the time of initializing the system. This adaptive control will allow a kind of bandwidth borrowing not only among users connected to the same LAN but also to other LANs. It is useful to take the arriving rate of the marked packets at the destination of the various traffic streams as a sign for parameter resetting. The ratio of the number of marked packets to the number of unmarked packets may also be used.

APPENDIX *A*
OSI Reference Model

A.1 Introduction

The international organization for standardization ISO created a new subcommittee for "Open System Interconnection" (OSI) [ZIM80] for standardizing heterogeneous computer networks, in 1977. This committee produced a specific layered communications architecture, the *OSI Reference Model*, which was approved in 1983 [ZIM83]. The OSI reference model is composed of seven layers, Fig 2.1. Each layer of the OSI model gives services to the layer above it. So, the (N)-entities in the (N) layer provide services to (N+1)-entities in the (N+1)-layer above. The interface between two layer entities is called the "(N)-service access point" (N-SAP). An (N)-entity can serve a number of (N+1)-entities through N-SAPs. On the other hand, each (N)-entity is assumed to have a virtual connection with a peer (N)-entity serving a remote user.

Protocol Data Units (PDUs) are defined to provide the transfer of information between peer entities via a peer protocol. Data units are first created at the application layer. Then, they are passed to the presentation layer where they are encapsulated, by adding some control bits, and called presentation PDUs. This encapsulation process may be repeated underneath, and so on. However, at the destination station, the control bits that were added by the sender (N)-entity, are only examined and removed by the peer (N)-entity, and the rest of the PDU will be send to next higher layer entity.

A.2 The Seven OSI Layers

In this section, a brief summary on each layer of the OSI reference model starting with the *application layer* is given. This layer serves the end users directly by providing distributed information services appropriate to an application, and to its management. This layer is also responsible for the semantics or meanings of the information being exchanged between the application entities. Examples of applications services are virtual terminal, file and job transfer and manipulation services and protocols [SCHW].

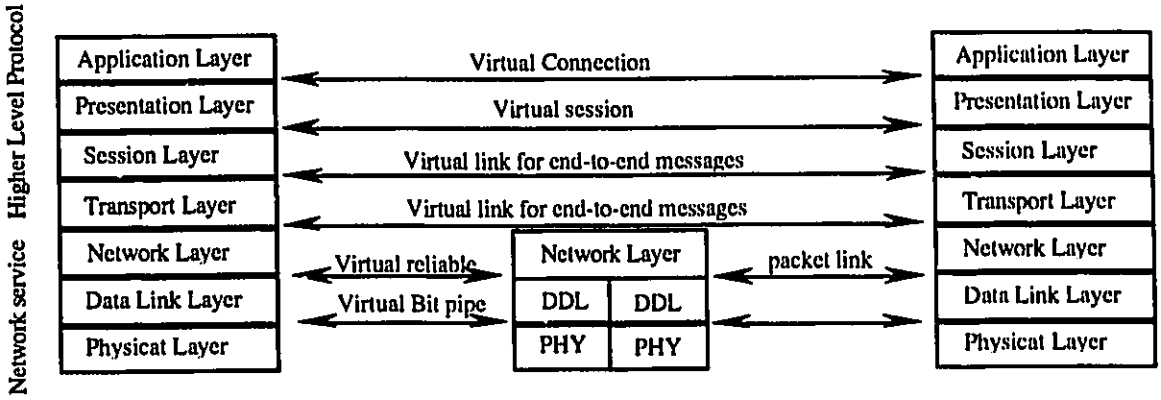


Figure A.1 *OSI Reference Model*

The presentation layer takes the responsibility of the transfer syntax of the data unit generated by or addressed to the application layer entities. Data encryption for network security and privacy purposes, data compression for reducing the number of bits to be transmitted, and code conversion for interconnection incompatible platforms with different terminals, printers, and file servers, are considered to be as its major functions.

The *session layer*, which lies under the presentation layer, manages and controls the dialogue between the end users by monitoring it. This consists of setting up a session and assuming a synchronization service to overcome any error detected. This layer also allows remote users to identify themselves in establishing communications with other users.

The *transport layer* is intended to provide a reliable and cost-effective data transfer. The services provided by this layer depends on the design of the lower layers provided. If the lower layers provide virtual circuits services, guaranteeing that messages are delivered in order and without loss from sender to receiver, the transport layer protocols become relatively simple. However, if a datagram service, which is a connectionless scheme where each packet generated by the transport layer can be routed differently through the network, is provided by the lower layers, the transport layer protocols may have to assure that packets are delivered in sequence and without loss, error, or duplication. For this purpose, various flow control schemes can be used, such as the window flow control [KNI83,TAN88]. Moreover, the transport layer is responsible for breaking and reconstructing messages coming from the session and the lower layer entities respectively. It can also multiplex several low rate session or split a high rate session into a multiple session at the network layer in order to get a higher quality of service. Finally, the transport layer is also used in gateways interconnecting networks using different packet types, and communication protocols.

The four higher layers are supposed to be end-to-end communication processes, which have to deliver the data correctly and in a recognizable form to the users. The three lower are more concerned with the network environment. In fact, these layers are not only used at the end stations, but also at the intermediate network nodes.

The *network layer* mainly provides necessary routing and congestion control procedures to ensure adequate delivery of packets. However, contrary to the other layers in which entities are defined to work in pairs, the network layer comprises entities working all together in implementing routing and congestion control. This can be done by algorithms using distributed processes. These processes must periodically interchange information in order to adequately react and prevent congestion problems. Examples of congestion control methods used at the network layer is the sliding window mechanism, a scheme where packets sent by

the source stations are chocked and/or discarded by nodes whose buffers are temporarily full. Another example are the schemes where buffer preallocation is done at the virtual circuit setup. Furthermore, similar to the transport layer, the network layer can also provide virtual circuit or datagram services, error detection and recovery.

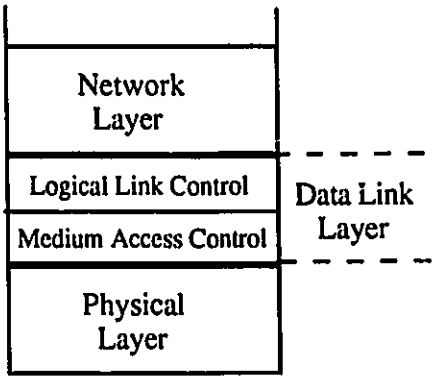


Figure A.2 *The LLC and the MAC sublayers*

APPENDIX B
Integrated Services Digital Networks (ISDN)

B.1 Introduction

ISDN involves a set of standards whose main objective is the end–to–end digital connectivity between end user. It offers bit rate from 64 Kbps up to 600 Mbps. In particular, N-ISDN (Narrowband ISDN) and B-ISDN(Broadband ISDN) have been proposed. N-ISDN is intended to provide bit rates up to 1.536 Mbps (2.048 Mbps in Europe) while B-ISDN is intended to provide 150/600 Mbps link per user for supporting broadband applications such as video communications. In the next section, the major concepts of ISDN and B-ISDN standards are reviewed.

B.2 ISDN

ISDN is a data communications network that is digital from one end to the other, i.e. end-to-end digital connectivity [AND84]. It is intended to carry all types of information including digitized voice, data, and image. On the ISDN access side, there is the network terminator (NT) which corresponds to the layer 1 of the OSI Reference Model (line transmission

termination, interface termination) [JUL86]. It can be accessed by one or several terminal equipment (TEs) through a passive bus. A large number of NTs can be physically attached in a full duplex way to the ISDN switch. On the network side, the ISDN switches can be interconnected through packet and/or circuit switched networks or networks composed of ISDN switches. In fact, each end user can be independent of the transport network by using an ISDN switch as a network interface.

Two ISDN interface are presently defined: the basic rate interface, composed of 2 B-channels (64 kbps/channel) plus one D-channel (16 kbps), for an overall data rate of 144 Kbps; the primary rate interface, composed of 23 B+D in Europe). The D-channel is mainly used for transferring control signaling in order to establish communications. The ISDN standards also define the H-channel which have a bandwidth in multiples of 64 Kbps. Hence, these channels represent the type of bandwidth offered to each NT connected to an ISDN switch.

B.2 Broadband ISDN

With the introduction of new services requiring large bandwidth such as high speed data packet (high volume file transfer services) and motion video services, there is a need to define a broadband network able to integrate these services requiring bandwidth higher than 2 Mbps. Thus, the CCITT Study Group XVIII [CCITT89] is defining the B-ISDN, Broadband Integrated Services Digital Networks, standard which is intended to provide 150/600 Mbps interface to end users or customer premises networks (see Fig.B.1). All types of services will be packetized at the terminal adapter (TA) and multiplexed at the user-network interface (UNI), and then transferred through a rout comprising a given number of switching nodes connecting the source and destination UNIs.

The CCITT study Group XVIII has adopted the ATM (Asynchronous Transfer

Mode) concept [RYD89,THO84] as transport mechanism for BISDN. In ATM, all information is conveyed within fixed-size slots called cells. Each ATM cell contains in its header a VCI and VPI (Virtual Channel Identifier, and Path Identifier) which allows the switching nodes to identify and switch the cell through the network. The VCI and VPI are determined at the virtual connection establishment. The term "asynchronous" is used since ATM is a packet switching based network. ATM cells are filled depending on the actual demand from each connection, which means that cells used by a given connection can exhibit an irregular recurrence pattern. ATM is then different than schemes like STM (Synchronous Transfer Mode) where dedicated bandwidth is preallocated for each connection (circuit switching concept). Thus, the major advantage of ATM is flexibility network efficient use, since any kind of traffic, already or eventually defined, will be integrated. For instance, ATM will be advantageous to support bursty traffic (such as compresses video) which is characterized by important variations of the bit rate generated during the maintenance of the virtual connection.

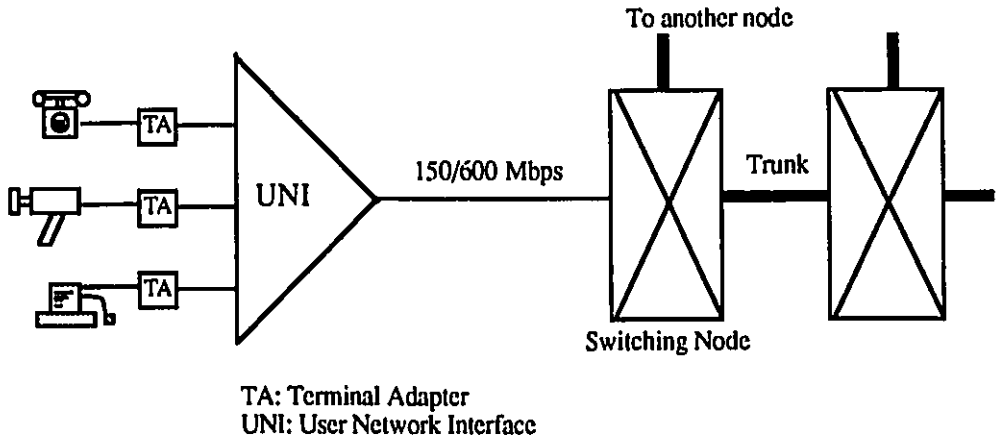


Figure B.1 *B-ISDN architecture*

Figure B.2 shows a typical example of a future telecommunications network

interconnecting public and private networks. B-ISDN will be used as a large public network interconnecting various types of LAN, and DQDB based networks. DQDB may be used as concentrator for regrouping LANs.

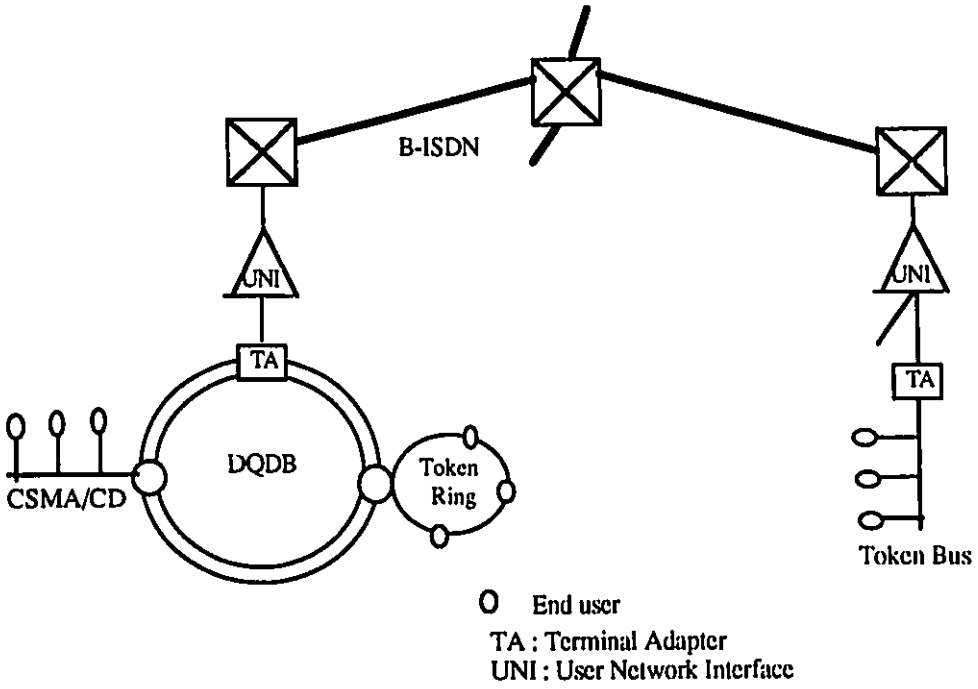


Figure B.2 *A telecommunication network*

Bibliography

[AND84] F.Andrews, "ISDN '83", *IEEE Communications Magazine*, vol. 22, no. 1, pp. 6-10, Jan. 1984

[ANSI85A] ANSI/IEEE Standard 802.5, "Token Ring Access Method and Physical Layer Specifications" *IEEE Project 802*, Local Area Network Standards, IEEE, New York, 1985.

[ANSI85B] ANSI/IEEE Standard 802.2, "Logical Link Control", *IEEE Project 802*, IEEE, New York, 1985

[ANSI85C] ANSI/IEEE Standard 802.3, "CSMA/CD Access Method and Physical Layer Specifications" *IEEE Project 802*, Local Area Network Standards, IEEE, New York, 1985.

[BAC88] F.Backes, "Transparent Bridges for interconnection of IEEE 802 LANs," *IEEE Network*, Vol. 2, No 1, January 1988, pp. 5-9.

[BAE90] J.J.Bae and T.Suda, "Survey of traffic control protocols in ATM networks," in *Proc. IEEE GLOBECOM' 90*, San Diego, CA, Dec. 2-5, 1990, pp.1-6.

[BAL90] K. Bala, I. Cidon and K Sohraby "Congestion control for high speed packet switched networks", in *Proc. IEEE INFOCOM'90*, San Francisco, CA, June 1990, pp. 520-526.

[BIE90] E.W. Biersack, "Annotated bibliography on network interconnection", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 1, Jan. 1990, pp. 22-41.

[BOV92] A.Bovopoulos, "Performance Evaluation of a traffic control mechanism for ATM networks," in *Proc. IEEE INFOCOM' 92*, Florence, Italy, May 4-5, 1992, pp. 469-478.

[BUD86] Z.L. Budrikis, J.L. Hullett, R.M. Newman, D. Economou, F.M. Fozdar and R.D. Jeffery, "QPSX: A Queued Packet and Synchronous Circuit Exchange", in *Proc. 8th Int. Conf. Comp. Comm.*, Munich, 1986, pp. 288-293

[BUR88] W.E. Burr and L. Zuqiu, " An Overview of FDDI," in *Proc. EFOC/LAN 88*, pp. 287 - 293

[BUX81] W.Bux, "Local Area Subnetworks: A Performance Comparison", *IEEE Trans. on Comm.*, vol. COM-29, no. 10, pp.1465-1473, Oct.1981

[BUX85] W. Bux and D. Grillo, "Flow control in Local Area Networks of interconnected Token Rings", *IEEE Trans on Comms*, vol. COM-33, no. 10, Oct. 1985, pp. 1058-1066.

[BUX87] W. Bux, "An investigation of the FDDI Media-Access Control Protocol", in *Proc. of EFOC/LAN'87*, pp 229 - 236

[CCITT88] "Broadband aspects of ISDN," CCITT COM XVIII Draft Recommendation I.121, June 1988.

[CCITT89] CCITT Recommendation E.164, "Numbering Plan for the ISDN Era.".

[CER78] C.Cerf and P.Kirstein, "Issues in packet network interconnection," *Proc. IEEE*, Vol. 66, pp. 1386-1408, November 1978.

[DEC92] M. Decina and V. Trecordi (Eds.), *IEEE Network: Special Issue on Traffic Management and Congestion Control for ATM Networks*, vol 6, no. 5, Sept.1992.

[ECK89] A.Eckberg, D.Lucantoni and D. Luan, "Meeting the challenge: Congestion and flow control strategies for broadband information transport," in *Proc. IEEE GLOBECOM'89*, Dallas, Texas, November 27-30, 1990, pp. 1769-1773.

[ECK91] A.Eckberg, B.Doshi and R.Zoccolillo, "Controlling congestion in B-ISDN/ATM: issues and strategies," *IEEE Communication Magazine*, September 1991, pp. 64-70.

[EPH80] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem", *IEEE Transactions on Automatic Control*, vol. AC-25, no. 4, Aug.1980, pp. 690-693.

[EVE90] C.Evequoz and C.Tropper, "End-to-end delay of multiple packet messages in window flow controlled packet switching networks," in *Proc. IEEE INFOCOM'90*, San Francisco, CA, June 1990, pp.47-54.

[FEN91] K.Fendick, D.Mitrani and M.Rodrigues, J.Seery, A.Weiss, " An approach to high-performance, high-speed data networks," *IEEE Communication Mag.* October 1991, pp. 74-82.

[FON93] N.Fonseca, J.Silvester, "A comparison of Push-Out Policies in an ATM Multiplexer," *IEEE Pacific RIM Conf. on Communications, Computers, and Signal Processing*, May 1993, Victoria, B.C., Canada

[GAL89] G. Gallassi, G.Rigolio and L.Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies," in *Proc. IEEE GLOBECOM' 89*, Dallas, Texas, Nov 27-30, 1990, , pp. 1,788-1,793, 1989

[GAL90] G.Gallassi, G.Rigolio and L.Fratta, "Bandwidth assignment in prioritized ATM networks," in *Proc. IEEE GLOBECOM' 90*, San Diego, California, Dec. 1990, pp.

[GER80] M.Gerla and L.Klienrock, "Flow control: A comparative survey," *IEEE Trans. on Communication*, Vol COM-28, No 4, April 1980, pp. 553-574.

[GER88] M. Gerla and L. Kleinrock, "Congestion control in interconnected LANs", *IEEE Network*, vol. 2, no. 1, Jan. 1988, pp. 72-76.

[GER92] M.Gerla and T.Tai, "LAN/MAN interconnection to ATM: a simulation study," in *Proc. IEEE INFOCOM'92*, Florence, Italy, May 4-5, 1992, pp.2270-2279.

[GLA89] B.Glass, "Under the Hood: The Token Ring", *Byte Magazine*, vol. 14, no. 1, pp. 363-376, Jan 1989.

[GUM92] M.Gumbold, P.Martini and R.Wittenberg, "Temporary overload in high-speed backbone networks," in *Proc. IEEE INFOCOM'92*, Florence, Italy, May 1992, pp.2280-2289.

[HAA91] Z.Haas and J.Winters, "Congestion control by adaptive admission," in *Proc. IEEE INFOCOM'91*, IEEE, 1991, pp. 560-569

[HAB91] I.Habib and T.Saadawi, "Controlling flow and avoiding congestion in Broadband Networks," *IEEE Communication Mag.*, Oct. 1991, pp. 46-53.

[HAN92] E.L. Hahne, A.K. Choudhury, and N.F. Maxemchuk, "DQDB networks with and without bandwidth balancing", *IEEE Trans. on Communs*, vol. 40, no. 7, July 1992, pp. 1192-1204.

[IEEE85] Draft Standard IEEE P802.6 Metropolitan Area Network, "Media Access Control Specifications", IEEE 802.6 Document 802.6/85-D, Revision D, August 1985.

[IEEE87A] Draft Standard IEEE P802.6 Metropolitan Area Network (MAN) Dual Buses, "Media Access Control (MAC) and Physical Layer Protocol Documents", Draft B.0, July 1987.

[IEEE87B] Attachment #25, Minutes of IEEE 802.6 MAN, Vancouver, B.C., Canada, July 13-17, 1987.

[IEEE88] Draft Standard IEEE P802.6, Distributed Queue Dual Buses Metropolitan Area Netwok (DQDB MAN), "Media Access Control (MAC) and Physical Layer Protocol Documents", Draft D.6, Nov. 1988

[INA91] H.Inai and K.Ohtsuki, "Congestion control in high speed backbone networks and interconnected LANs," *High-Speed Networking III*, North-Holland Netherlands 1991, pp. 137-152.

[INA92] H. Inai and K. Ohtsuki, "Performance study of congestion control for high-speed backbone networks", *Computer Communications*, vol. 15, no. 7, Sep. 1992, pp. 429-437.

[JAB90] B.Jabbari, "A bandwidth allocation technique for high-speed networks," in *Proc. IEEE GLOBECOM'90*, San Diego, CA, December 2-5, 1990, pp.355-359.

[JAC88] V.Jacobson, "Congestion avoidance and control," *Proc. ACM SIGCOMM'88.*, Stanford, CA, August 1988, pp.314-329.

[JAI90] R.Jain, "Congestion control in computer networks: Issues and trends," *IEEE Network Mag*, May 1990.

[JUL86] U. de Julio, "Layer 1 ISDN Recommendations," *IEEE JSAC*, vol. SAC-4, no. 3, pp. 355-359, May 1986.

[KLI92] L. Kleinrock, "The Latency/Bandwidth tradeoff in Gigabit networks", *IEEE Communications Magazine*, vol. 30, no. 4, April 1992, pp. 36-40.

[KNI83] K.G.Knightson, "The Transport Layer Standardization", in *Proc. IEEE*, Vol. 71, no. 12, Dec. 1983, pp. 1394-1396.

[LUA88] D.Luan and D.Lucantoni, "The effect of bandwidth management on the performance of a window-based flow control, " *AT&T Technical Journal*, Sep-Oct. 1988, pp 17-26.

[MAR91] P. Martini and T. Meuser, "Real-Time Traffic in FDDI-II Packet Switching vs. Circuit Switching," *in Proc. IEEE INFOCOM'91*, 1991, pp. 1413 - 1420

[MOL90] J. Mollenauer, " Metropolitan Area Networks: How they differ from LANs and WANs," article included with the *DQDB Draft* sent to the TCCC balloting committee, 1990.

[NEW86] R.M. Newman and J.L. Hullett, "Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX", in *Proc. 8th Int. Conf. Comp. Comm.*, Munich, 1986, pp. 288-293

[NEW88] R.M. Newman, Z.L. Budrikis and J.L. Hullett, " The QPSX MAN", *IEEE Comm Mag.*, vol. 26, no. 4, pp. 20-28, April 1988.

[PANG89] J.Pang and F.A.Tobagi, "Generalized Access Control Strategies for Token-Passing Systems", in *Proc. IEEE INFOCOM'89*, Ottawa, Canada, April 1989, pp. 332-341.

[PIN90] S.Pingali, D.Tipper and J.Hammond, " The performance of adaptive window flow controls in dynamis load environment," in *Proc. IEEE ICC/SUPERCOMM'90*, Atlanta, GA, 1990, pp 55-62.

[POI84] D. Poiter, "New Users' Introduction to QNAP2," *Technical Report No. 40*, INRIA, France, October 1984.

[RID88] M.J.Rider, "Protocols for ATM Access Networks," in *Proc. IEEE GLOBECOM' 88*, Hollywood, Florida, November 28-December 4, pp. 112-116.

[ROS86] F.E.Ross, "FDDI-A tutorial", *IEEE Comm. Mag.*, vol 24, no. 5, pp. 10-17, May 1986.

[RYD89] M. Ryder, "Protocols for ATM Access Methods", *IEEE Network Mag.*, vol. 3, no. 1, pp. 17-22, Jan. 1989.

[SCH87] M.Schwartz, *Telecommunication Networks*, Addisson-Wesley Publishing Company, 1987

[SID89] M.Sidi, W.Liu, I.Cidon and I.Gopal, "Congestion control through input rate regulation," in *Proc.IEEE GLOBECOM'89*, Dallas, Texax, Nov 27-30, 1990, pp. 1764-1768

[SUN90] C.Sunshine, "Network interconnection and gateways," *IEEE JSAC*, Vol 8, No.1, January 1990, pp.4-11.

[TAN88] A.S. Tanenbaum, Computer Networks, Prentice Hall Inc., New Jersey, 1988.

[TANT90] A.Tantawy, "Bridging issues in DQDB subnetworks," in *Proc. SPIE*, vol 1364 FDDI, Campus-Wide, and MANs, 1990 pp. 268-277.

[THO84] A. Thomas, J.P. Coudreuse and M.Servel, "Asynchronous Time-Division Techniques: An Experimental Packet Network Integrating Videocommunication", in *Proc. ISS'84*, Florence, Italy, May 1984, 32C.

[TUR86] J.S.Turner," New directions in communications (or Which Way to the information age?)," *IEEE Comm. Mag*, Vol 24 October 1986 pp8-15

[VAL89] R.Vallée, L.Orozco-Barbosa and N.D.Georganas, "Interconnection of token rings by IEEE 902.6 MANs," in *Proc. IEEE INFOCOM'89*, July 1989, Ottawa, Canada, pp. 934-941.

[WAN90] W. Wang , T.N. Saadawi and K.Aihara," Bandwidth Allocation for ATM networks", in *Proc. IEEE ICC/SUPERCOMM'90*, Atlanta, GA, 1990, p439-442 , 1990

[WIL91] W.S.William, L.Orozco-Barbosa and N.D.Georganas, "System unfaireness of DQDB in supporting multiple communities of interest," *in Proc. IEEE GLOBECOM' 91*, December 8-11, Phoenix, Arizona, pp.1116-1120.

[WON89] L. Wong and M. Schwartz, "Flow control in Metropolitan Area Networks", *in Proc. IEEE INFOCOM'89*, July 1989, Ottawa, Canada, pp. 826-833.

[ZIM80] H.Zimmermann, "OSI Reference Model - The OSI Model of Architecture for Open Systems Interconnection", *IEEE Trans. on Comm.*, vol. COM-28, no.4, pp.425-432, April 1980

[ZIM83] H.Zimmermann, "OSI Reference Model", *Proc. IEEE*, vol. 71, no. 12, pp. 1334-1340, Dec. 1983