UMI Number: 9730388

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

ABSTRACT

HIGH SPEED PROTOCOLS FOR DUAL BUS AND DUAL RING NETWORK ARCHITECTURES

by
Yaling Zhou

In this dissertation, two channel access mechanisms providing fair and bandwidth efficient transmission on dual bus and dual ring networks with high bandwidth-latency product are proposed. In addition, two effective priority mechanisms are introduced to meet the throughput and delay requirements of the diverse arrays of applications that future high speed networks must support.

For dual bus architectures, the Buffer Insertion Bandwidth Balancing (BI_BWB) mechanism and the Preemptive priority Bandwidth Balancing (P_BI_BWB) mechanism are proposed. BI_BWB can significantly improve the delay performance of remote stations. It achieves that by providing each station with a shift register into which the station can temporarily store the upstream stations' transmitted packets and replace these packets with its own transmissions. P_BI_BWB, an enhancement of BI_BWB, is designed to introduce effective preemptive priorities. This mechanism eliminates the effect of low priority on high priority by buffering the low priority traffic into a shift register until the transmission of the high priority traffic is complete.

For dual ring architectures, the Fair Bandwidth Allocation Mechanism (FBAM) and the Effective Priority Bandwidth Balancing (EP_BWB) mechanism are introduced. FBAM allows stations to reserve channel bandwidth on a continuous basis rather than wait until bandwidth starvation is observed. Consequently, FBAM does not have to deal with the difficult issue of identifying starvation, a serious drawback of other access

mechanisms such as the Local and Global Fairness Algorithms (LFA and GFA, respectively). In addition, its operation requires a significantly smaller number of control bits in the access control field of the slot and its performance is less sensitive to system parameters. Moreover, FBAM demonstrates Max-Min flow control properties with respect to the allocation of bandwidth among competing traffic streams, which is a significant advantage of FBAM over all the previously proposed channel access mechanisms. EP_BWB, an enhancement of FBAM to support preemptive priorities, minimizes the effect of low priority on high priority and supports delay-sensitive traffic by enabling higher priority classes to preempt the transmissions of lower priority classes. Finally, the great potential of EP_BWB to support the interconnection of base stations on a distributed control wireless PCN carrying voice and data traffic is demonstrated.

HIGH SPEED PROTOCOLS FOR DUAL BUS AND DUAL RING NETWORK ARCHITECTURES

by
Yaling Zhou

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Computer and Information Science

May 1997

APPROVAL PAGE

HIGH SPEED PROTOCOLS FOR DUAL BUS AND DUAL RING NETWORK ARCHITECTURES

Yaling Zhou

Dr. Dennis Karvelas, Dissertation Advisor — Date
Assistant Professor of Computer and Information Science Department, NJIT

Dr. James McHugh, Committee Member — Date
Professor of Computer and Information Science Department, NJIT

Dr. Nirwan Ansari, Committee Member — Date
Associate Professor of Electrical and Computer Engineering Department, NJIT

Dr. Constantinos Manikopoulos, Committee Member — Date
Associate Professor of Electrical and Computer Engineering Department, NJIT

Dr. Rakesh Kushwaha, Committee Member — Date
Senior Member of Technical Staff, AT&T, Homdel, NJ

BIOGRAPHICAL SKETCH

Author: Yaling Zhou

Degree: Doctor of Philosophy

Date: May 1997

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer and Information Science,
  New Jersey Institute of Technology, Newark, NJ, 1997

- Master of Science in Computer and Information Science,
  Fudan University, Shanghai, P.R. China, 1992

- Bachelor of Science in Computer Engineer,
  Zhejiang University, Hangzhou, P.R. China, 1989

Major: Computer and Information Science

Presentation and Publications:

Dennis Karvelas and Yaling Zhou,
"Support of Integrated voice/data wireless personal communication services using a high-speed dual ring MAN," *Proc. of SPIE International Symposium on Voice, Video, and Data Communications.*, Boston, Nov. 1996, Vol. 2917B, pp. 492-502.

Yaling Zhou and Dennis Karvelas,
"An Effective Buffer Insertion Mechanism for Medium Access Delay Control on High Speed Dual Bus Architectures", *NJIT Technical Report, TR-97-9,* also submitted to the journal of *Computer Networks and ISDN Systems.*

Yaling Zhou and Dennis Karvelas,
"A Distributed Control Channel Access Mechanism with MAX-MIN Flow Control Properties", *NJIT Technical Report, TR-97-10,* also submitted to the *Journal of High Speed Networks.*

Yaling Zhou and Dennis Karvelas,

"Queuing Analysis of a Buffer Insertion Mechanism for High Speed MANs", *NJIT Technical Report, TR-97-11,* also submitted to the *Telecommunication Systems Journal.*

Yaling Zhou and Dennis Karvelas,

"Fair Bandwidth Allocation on High Speed Dual Ring Architectures", *NJIT Technical Report, TR-96-10.*

Yaling Zhou, Bole Shi, and Yangyong Zhu,

"A Design Tool for Relational Database Schema SRSP," *Journal of Fudan University (Natural Science),* Mar. 1992, vol. 31, No. 1, pp. 105-110.

Yaling Zhou and Dennis Karvelas,

"Buffer Insertion Bandwidth Balancing for Broadband Dual Bus Networks," submitted to the *22nd Annual Conference on Local Computer Networks,* Minneapolis, Minnesota, November 1997.

Yaling Zhou and Dennis Karvelas,

"Fair and Efficient Transmission on High Speed Dual Ring Networks Employing Spatial Reuse", submitted to the *Sixth International Conference on Computer Communication and Networks,* Las Vegas Nevada, September 1997.

This dissertation is dedicated to
my beloved husband
Zhu Liu

ACKNOWLEDGMENT

I would like to extend my sincere gratitude to Dr. Dennis Karvelas, who not only served as my research supervisor providing valuable and countless guidance, resources, insight, and intuition, but also constantly gave me support, encouragement, and reassurance.

Special thanks are given to Dr. James McHugh, Dr. Nirwan Ansari, Dr. Constantinos Manikopoulos, and Dr. Rakesh Kushwaha for serving as members of my committee. Moreover, I wish to thank Dr. Peter Ng for his encouragement, guidance and help throughout the course of this work.

I would also like to thank my parents, Xiangxi Zhou and Shuzhu Chen, my brother, Guoyi Zhou, and my son, Justin Liu, for their love and moral support.

Finally, I would like to express my deepest appreciation to my loving husband, Zhu Liu, for his love, his continual support over the years, and his indefatigable efforts in helping me to complete this work. He has always been a consistent source of my strength. This work would not have been accomplished without him.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF TABLES
(Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES
(Continued)

GLOSSARY

ACF: Access Control Field

AD: Access Delay

ASD: Access/Storage Delay

ATM: Asynchronous Transfer Mode

BB: Busy Bit

BF_CTR: Buffer Counter

BI_BWB: Buffer Insertion Bandwidth Balancing mechanism

B-ISDN: Broadband Integrated Service Digital Network

BIU: Base Interface Unit

BMOD: Buffer Modulus

B_queue: Busy slot queue (in the queueing model)

BS: Base Station

BWB_CTR: Bandwidth Balancing Counter

BWB_DQDB: Bandwidth Balancing mechanism for DQDB

BWB_MOD: Bandwidth Balancing Modulus

BWM: Bandwidth Manager

CD_CTR: Count Down Counter

CIU: Control Interface Unit

DB: Data Base

DBTAR_CTR: Debit TAR Bit Counter

DQDB: Distributed Queue Dual Bus

EP_BWB: Effective Priority Bandwidth Balancing mechanism

GLOSSARY
(Continued)

FBAM: Fair Bandwidth Allocation Mechanism

FCFS: First Come First Served

GBW: Guaranteed Bandwidth mechanism

GFA: Global Fairness Algorithm

GPI_BWB: Global Priority Information Bandwidth Balancing mechanism

GUI: Gateway Interface Unit

HIU: Home Interface Unit

ID_RRS: I.D. based Request Removal Strategy

LAN: Local Area Network

LFA: Local Fairness Algorithm

LMOD: Local Modulus

LOCAL_CTR: Local Counter

LPS: Lower Priority Segment buffer

L_queue: Local segment queue (in the queueing model)

MAC: Medium Access Control

MAN: Metropolitan Area Network

MU: Mobile Unit

NRM: No Restriction Mechanism

NSW_BWB: No Slot Wasting Bandwidth Balancing mechanism

NSW_IUT: No Slot Wasting with Immediate Use of TAR bit

NTAR_R: Number of Available TAR Segment Register

P_BI_BWB: Preemptive priority Buffer Insertion Bandwidth Balancing mechanism

GLOSSARY
(Continued)

ES_CTR: Erased Slot Counter

PCN: Personal Communication Network

PIN: Personal Identification Number

QPSX: Queued Packet and Synchronous circuit eXchange

REQ_QS: Request Queue Size

RF: Request Field

RFES: Request Field Erasure Station

RG_CTR: Registered Counter

RQ_CTR: Request Counter

R_queue: Queue of arriving requests (in the queueing model)

S_i: station "i"

SN_RRS: Section Number based Request Removal Strategy

ST_CTR: Segment Transmission Counter

TAR: Transmit Additional Request

TBP_GFA: Time BAsed Priority Global Fairness Algorithm

TIU: Trunk Interface Unit

TTRT: Target Token Rotation Time

UNRG_CTR: Unregistered Counter

VCI: Virtual Circuit Identifier

VIU: Visitor database Interface Unit

GLOSSARY
(Continued)

e_i : probability that a TAR = 1 bit (on a busy slot) is erased by station S_i

n_i : probability that station S_i transmits a local TAR segment without inserting a request for it.

p_{ik} : probability that station S_i does not insert a request for a local segment that S_i transmits in the k_{th} slot from the instant this segment arrived at S_i's L-queue.

r_i : probability that station S_i inserts a request into a passing slot on the reverse bus.

t_i : probability that an arriving busy slot at station S_i carries a TAR = 1 bit.

$\bar{t}_i$: probability that an arriving busy slot at station S_i carries a TAR = 0 bit.

u_i : probability that an arriving slot at station S_i is idle and unreserved, i.e., it can be written by S_i.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The introduction of high-speed switching, optical fiber transmission, and ATM networking has opened up new opportunities for the development of high speed communication networks such as the Broadband Integrated Service Digital Network (B-ISDN) that can provide a diverse array of communication services in an integrated fashion. Such services will include file transfers, voice and video transmission, high capacity workstation interconnection, and LAN interconnection. Moreover, they will support the communication requirements of intensive data-processing applications such as image processing, numerical scientific parallel computations, multimedia database retrieval, video mail, interactive design, real-time simulations, and tele-conferencing. This wide variety of services will generate flows of information with very different traffic characteristics. Therefore, a major challenge for designers of the next generation of high capacity networks is the efficient allocation of the enormous available bandwidth among a large number of competing traffic sources with diverse throughput and delay requirements.

Various access mechanisms have been recently proposed for the efficient share of a high capacity channel in the local area environment [2], [59], [79], [80], [87]. However, these mechanisms cannot be directly extended to higher bandwidths and longer distances. The reason is that all of them follow a cyclic operation in order to introduce fairness in bandwidth allocation. This operation can guarantee for each station the same number of transmission opportunities during each cycle but requires a round trip propagation delay gap for distinguishing successive cycles. Consequently, the performance of these systems deteriorates significantly as the size of the network increases, i.e., the round-trip propagation delay and overhead delay gap per cycle increase. Equation (1.1) provides a quantitative measure of the effect of various system parameters on the efficiency (maximum channel utilization) of cyclic type mechanisms [74]:

$$\rho_{max} = \frac{1}{1 + \frac{t_{ov} C}{N l_m}} \tag{1.1}$$

In equation (1.1) N is the number of stations, l_m the maximum number of bits that each station can transmit during each cycle, C the channel capacity, and t_{ov} the round trip propagation delay. Equation (1.1) clearly shows that both the channel capacity and the network size have a strong negative effect on the maximum system utilization. For instance, consider an 100 Mbps 2 km ring network with N = 20 stations. Let us assume that each station can transmit up to $l_m = 20,000$ bits during each cycle and that the signal propagation delay is 5 μsec/km. Then $t_{ov} = 2 \times 5 = 10$ μsec and equation (1.1) provides a maximum utilization ρ_{max} of almost 1. Now let us assume that the transmission speed of the channel increases to C = 1 Gbps and the network size to 100 km, i.e., $t_{ov} = 100 \times 5 = 500 \mu$sec. The corresponding system utilization will now decrease to 0.44. We can improve ρ_{max} by increasing the maximum number of bits (i.e., l_m) that each station can transmit during each cycle. However, the higher the value of l_m the more unfair the system will become to lightly loaded stations.

The previous discussion clearly demonstrates the limitations of the cyclic operation in networks with high bandwidth-latency product. New medium access control protocols are needed whose performance is not sensitive to network size, the number of connected stations, or channel capacity. In addition, these mechanisms must provide effective priorities which can satisfy the diverse throughput and delay requirements of a wide variety of applications that future networks will support. We provide now a brief description of the main characteristics that appropriate Medium Access Control (MAC) mechanisms for Gbps networks must have. These are:

Simplicity: The MAC mechanism must be simple, i.e., its operation should not require significant processing since it will operate at very high speeds.

Minimal overhead for scheduling: The amount of control information which is needed for scheduling should be minimal. Furthermore, it should not be affected by the system parameters.

Fairness: Stations with similar traffic characteristics should acquire similar bandwidth and encounter similar delays regardless of their locations on the network.

Minimal effect of system parameters on performance: The throughput and delay performance should not be affected by system parameters such as network size, number of connected stations, packet size, traffic characteristics, or channel capacity.

Support of effective priorities: In the presence of multiple priorities, classes of traffic with low priority should not affect the performance of high priority.

The objective of this dissertation is to introduce and investigate the performance of effective MAC mechanisms for high speed dual bus and dual ring network architectures that can demonstrate the afore mentioned characteristics. In the following sections, we provide a brief introduction of the dual bus and dual ring network architectures.

1.2 Dual Bus Architecture

A dual bus network consists of two unidirectional buses on which information travels in opposite directions. Nodes are connected to both buses, as shown in Fig.1.1

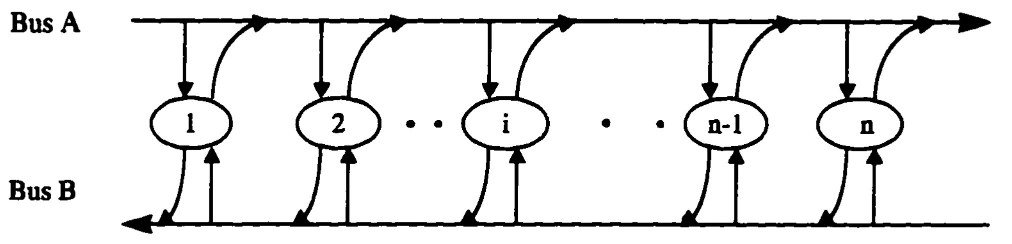

Fig.1.1 Dual bus network architecture

The first station on each bus generates fixed size slots which travel downstream. A segment is the unit of information and is equal to the data field of the slot. The Busy Bit (BB) in the Access Control Field (ACF) of the slot indicates whether a slot is currently empty (BB = 0) and can be written by a station, or busy (BB = 1), i.e., it has already been written by an upstream station.

It is evident from Fig.1.1 that if a station wants to send data to another station located to its right, it will transmit onto bus A. Otherwise, it will transmit onto bus B. In the following discussions, we focus on the transmissions on bus A. The operation regarding the transmissions on bus B is identical. **Forward bus** and **forward channel** will be used interchangeably for bus A, and **reverse bus** and **reverse channel** for bus B. Furthermore, a station S_j is said to be upstream from a station S_i when S_j can see the slots on bus A before S_i.

The main advantage of the dual bus topology is that it can eliminate the t_{ov} overhead required by the cyclic operation, thus enabling a maximum utilization of 1 (1.1) regardless of the values of the system parameters. However, dual bus topology introduces severe fairness problems. That is, the locations of the stations on the bus drastically affect their throughputs as well as the delays their packets will encounter. For instance, in Fig.1.1, if station S_1 is overloaded and keeps on transmitting packets on bus A, it will never allow a downstream station to see any idle slot. It is evident that an effective Medium Access Control (MAC) mechanism is needed to ensure fair transmissions in the network. Consequently, various MAC algorithms have been proposed for dual bus networks. However, investigation of their performances, which has been conducted in [13], [14], [19], [20], [89], and [93], clearly shows their limited success. That is, these mechanisms are either not robust (they are fair only under certain types of loading) or too complex to implement in a high speed network. An extensive survey of the research work in this area, together with a discussion on the limitation of the existing MAC mechanisms, is presented in Chapter 2.

1.3 Dual Ring Architecture

In this dissertation, we will also investigate access mechanisms for high speed dual ring network architectures. Our interest in dual ring networks has been motivated by their ability to offer much higher aggregate throughputs than dual bus networks. A dual ring consists of two unidirectional rings on which information travels in opposite directions. The stations are connected to both rings, as shown in Fig.1.2.

The shortest path routing rule is used in deciding the ring on which a station will transmit a packet. For instance, in Fig.1.2, if station S_1 needs to send packets to station

S_3, it will send them through ring A because it provides the shortest path. If station S_1 needs to send packets to station S_8, it will use ring B.

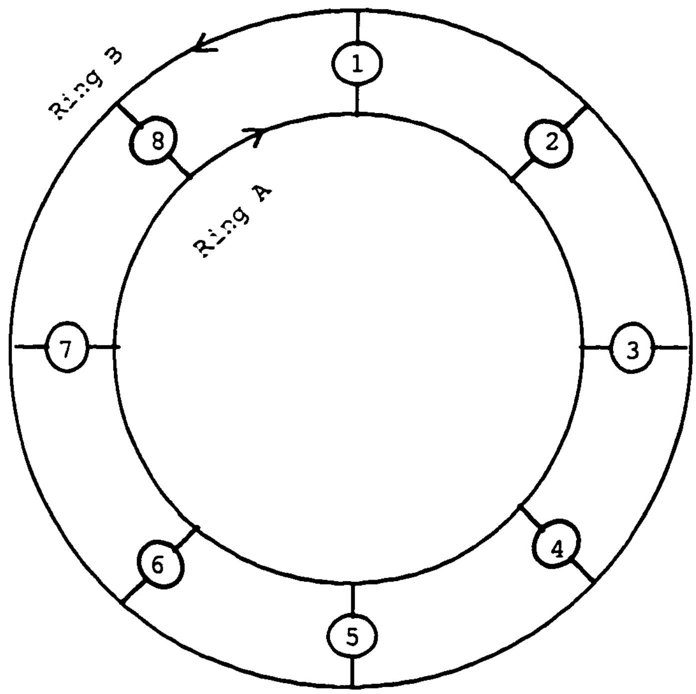

Fig.1.2 Dual ring network architecture

There are two approaches for removing the written slots from the ring. In the first one, called ***source release***, the transmitting station is responsible for resetting the slot. In the second one, called ***destination release,*** the destination station of the transmission is responsible for resetting the written slot. The advantage of ***destination release*** is that it enables the same slot traveling around the ring to be reused by other stations. For this reason the name Spatial Bandwidth Reuse is also being used for this slot removal technique. It is evident that ***destination release*** allows concurrent transmissions by stations on non overlapping ring segments. For instance, in the case of ring A of Fig.1.2, S_1 can transmit to S_4 continuously and at the same time S_5 can transmit to S_8 continuously. Consequently, ***destination release*** may significantly increase the aggregate throughput of the system. For instance, in the case of a uniform traffic pattern, i.e., when each station transmits to all other stations with the same probability, each written slot will travel on the average half a ring and the aggregate throughput of ring A will be twice the channel bandwidth. ***Destination release*** comes, however, at the cost of a higher ring latency since each station must delay every passing slot in order to look at its destination address and decide on whether it should remove it or not. In the case of ***source release***, this is not necessary since the size (in slots) of the ring is known and each station can anticipate the return of a written slot.

Destination release combined with a dual ring network architecture employing shortest path routing can increase even more the aggregate throughput of the system. The reason is that in this case, each station will always select the shortest path to destination, and slots will be more efficiently reused. For instance, in the case of the uniform traffic pattern mentioned above, each packet will now travel, on the average, only $1/4$ th of the ring. That is, the aggregate throughput per ring will be 4 times the channel bandwidth, i.e., the aggregate throughput of the system will be 8 times the channel bandwidth. This higher bandwidth comes at the cost of a higher complexity since each station must now keep a routing table to decide every time on which ring it must transmit its packets.

Despite their high throughputs, dual ring networks also suffer from severe fairness problems. For instance, unrestricted transmission may result in bandwidth starvation for certain stations. As an example, consider Fig.1.3 which shows station S_1 transmitting to station S_4 on ring A while station S_4 transmits to S_2 on ring B. In this case, station S_3 will not be able to transmit any packet on either ring. It is evident that a fairness mechanism is needed to regulate the stations' access to the network. The most prominent mechanisms that have been proposed for this purpose in the literature include those of MAGNET [55] and ORWELL [29] rings, as well as the Global [18] and Local [16] Fairness Algorithms; GFA and LFA, respectively. A discussion on those mechanisms (and their limitations) will be presented in Chapter 2.

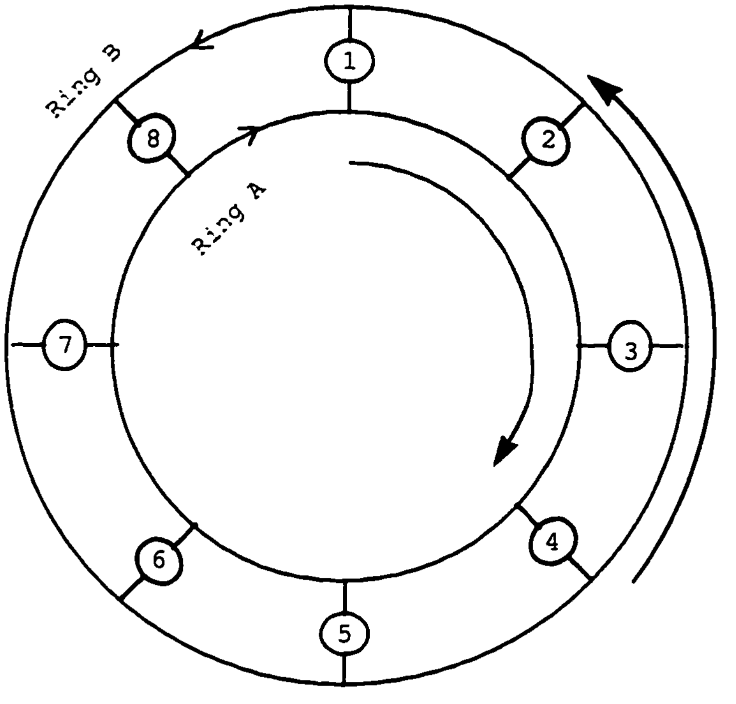

Fig.1.3 Example of bandwidth starvation

1.4 Multiple Priority Traffic

High speed networks are expected to support a wide variety of applications with diverse throughput, delay, and jitter requirements. Effective priority mechanisms that can meet these requirements are, thus, of paramount importance. Several priority mechanisms, including the one proposed for DQDB [85], the GPI_BWB [38] and TTR priority mechanisms [2], have been proposed for high speed networks and will be briefly discussed in

Chapter 2. Their limitations have motivated us to make the introduction of effective priority mechanisms as one of the major objectives of this dissertation.

1.5 Dissertation Contributions

The main objectives of this dissertation are: a) to propose and investigate the performance of channel access mechanisms that can improve fairness and introduce efficient bandwidth transmission on dual-bus and dual-ring networks with high bandwidth-latency product, b) to introduce effective priority mechanisms that can meet the diverse throughput and delay requirements that the high speed networks of the future must support.

First, we introduce the Buffer Insertion Bandwidth Balancing (BI_BWB) mechanism that can improve, on dual bus networks, the downstream stations' delay performance. This mechanism tries to combine the advantages of the recently proposed NSW_IUT [48] and BWB_DQDB [85] mechanisms by enabling downstream stations to have a faster access to the channel. This is achieved by providing each station with a shift register into which the station can insert incoming written slots. In this way, idle slots are created which the downstream station can use to transmit its packets. It is evident that with the inserted slots, the train of channel slots will snake its way in and out of many stations and will increase the latency of the bus. This extra latency will be decreased whenever idle slots, reserved by the station, arrive at this station; at this instant, the inserted extra buffer space will be removed. The advantage of the BI_BWB method is that it enables downstream stations to access the channel immediately (as in the case of BWB_DQDB) but without the need of wasting any channel bandwidth, a drawback of the BWB_DQDB mechanism.

The second contribution of this dissertation is a queuing analysis of the BI_BWB mechanism that can provide very good estimates for the average delay at each station. The proposed queueing model is capable of capturing the interdependencies among the busy slots on the forward channel, the slot reservations on the reverse channel, and the distance between stations. It can derive estimates for the stations' access delays which are in good agreement with simulation results.

The third contribution of this dissertation is the proposed Preemptive priority BI_BWB mechanism (P_BI_BWB). This mechanism is an enhancement of BI_BWB which can provide preemptive priority capabilities to higher priority classes in the following way. It allows stations to use their shift registers to temporarily remove the low priority traffic from the channel until the transmission of the high priority traffic is complete. At the same time it reserves enough idle slots to ensure the retransmission of the buffered low priority segments later on. In this way, the effect of low priority traffic on high priority traffic is completely eliminated. In fact, the operation of each traffic class becomes similar to that of a single priority system with channel bandwidth being the unused bandwidth by all higher priority classes.

The fourth contribution of this dissertation is the introduction of the Fair Bandwidth Allocation Mechanism (FBAM) for dual ring architectures. This mechanism enables stations to make slot reservations on a continuous basis rather than when bandwidth starvation is observed. Consequently, it does not encounter the difficult task of detecting starvation, which is a serious drawback of the recently proposed Local Fairness Algorithm (LFA) [16]. Other significant advantages of FBAM over LFA include the following: a) it can provide a max-min throughput fairness which is the optimum bandwidth allocation that an algorithm may achieve, b) its performance is less sensitive to the system parameters, c) its operation requires a significantly smaller number of control bits. Finally, its max-min fairness behavior allows the analytic derivation of accurate estimates for the stations' throughput in the case of overload traffic conditions.

The fifth contribution of the dissertation is the introduction of the Effective Priority BWB mechanism (EP_BWB) for dual ring networks. This mechanism is an extension of the FBAM mechanism and can provide effective priorities on high speed dual ring networks. Its operation can minimize the effect of low priority traffic on high priority traffic and, for this reason, it can meet the stringent delay requirements of real-time traffic.

Finally, the sixth contribution of the dissertation is the investigation of the performance of the EP_BWB mechanism in a more real world environment. Our motivation for this performance analysis is due to the current world-wide interest in wireless Personal Communication Networks (PCNs). We investigate the ability of a dual ring network, under the EP_BWB mechanism, to support the interconnection of base stations on a distributed control wireless PCN carrying voice and data traffic.

1.6 Dissertation Outline

The organization of the rest of the dissertation is as follows: In Chapter 2, we provide a brief survey of the research work in the area of high speed MANs and discuss the advantages and limitations of the existent MAC mechanisms. In Chapter 3, we introduce the Buffer Insertion BWB mechanism for dual bus networks. We discuss our motivation for its introduction, provide a detail presentation of the BI_BWB access algorithm, and investigate its throughput and delay performance. We also provide a queueing analysis for

BI_BWB, which can derive accurate estimates for the average segment delay at each station. In Chapter 4, we introduce the Preemptive priority BI_BWB mechanism (P_BI_BWB) which can eliminate completely the effect of low priority traffic on high priority traffic; it is, therefore, appropriate for serving real-time traffic. In Chapter 5, we introduce the Fair Bandwidth Allocation Mechanism (FBAM) for dual ring network architectures. We demonstrate its fairness, with respect to throughput and delay, and compare its performance with that of the Global and Local Fairness Algorithms under various traffic load configurations. In Chapter 6, we present the Effective Priority Bandwidth Balancing (EP_BWB) MAC mechanism. We investigate its throughput and delay performance under various traffic scenarios and demonstrate its ability to support effectively delay-sensitive traffic. In Chapter 7, we provide a brief description of a distributed control wireless PCN system and investigate its performance when a dual ring network employing spatial reuse is used for the interconnection of its microcellular base sites. Finally, in Chapter 8, we present our conclusions and suggestions for future work.

CHAPTER 2

PREVIOUS WORKS IN HIGH SPEED MANS

2.1 Introduction

High speed MANs, based on the dual bus and dual ring topologies, have recently become a very active research area. Several medium access control mechanisms have been proposed for allocating their channel bandwidth. In this chapter, we provide a brief literature review of the research work in this field. In section 2.2, we describe the main Medium Access Control (MAC) algorithms which have been proposed for dual bus networks. In section 2.3, we present the main MAC algorithms which have been introduced for dual ring networks. Finally, in section 2.4, we discuss some priority mechanisms which have been proposed for supporting multiple priority traffic.

2.2 MAC Mechanisms for Dual Bus Networks

In this section, we first provide a brief description of the Distributed Queue Dual Bus (DQDB) MAC mechanism [85] and elaborate on its fairness problem. Then, we discuss the three most effective Bandwidth Balancing Mechanisms that have been proposed to deal with the fairness issue. The objective is to provide a useful insight into the DQDB operation and the various approaches one can follow in order to improve it.

2.2.1 The DQDB MAC Mechanism

The dual bus architecture was first introduced in the Queued Packet and Synchronous Circuit Exchange (QPSX) [11] network. The objective was to introduce a high speed Metropolitan Area Network (MAN) whose maximum throughput would remain 1 regardless of its size, number of station it connects, packet size, or channel bandwidth. The idea was so attractive that IEEE formed the IEEE 802.6 committee to prepare a high speed network standard for the metropolitan area which would be based on QPSX. This IEEE 802.6 standard [85] became known as the Distributed Queue Dual Bus (DQDB) network because of its medium access control method which tries to form a distributed queue of waiting packets in order to transmit them in a First Come First Served (FCFS) order. Although FCFS is indeed achieved when the signal propagation delay is negligible (i.e., the size of the network is small), it is not possible as the size of the network increases. In fact, extensive investigations of the DQDB performance have demonstrated serious problems in the ability of the DQDB access mechanism to be fair to the competing for the channel users. That is, the location of a station in the network has a very strong effect on the amount of bandwidth this station may acquire and/or the delay that its packets will encounter.

The fairness problem of DQDB led to an explosion of research activity in the area of MANs that was mainly directed towards understanding the DQDB operation and overcoming its limitations. A wide variety of access mechanisms were proposed with many of them having only a very limited success. The main difficulty in introducing a fair and efficient MAC mechanism in the MAN environment arises from the large distances involved which delay the propagation of control information thus affecting the stations' view of the channel activity. Consequently, many of the access mechanisms which were proposed to alleviate the DQDB fairness problem required the transmission of a considerable amount of control information while others performed very well only under certain types of traffic load. When the traffic load generated by the stations changed, their performance could deteriorate significantly.

In DQDB each slot consists of an one byte Access Control Field (ACF) and a 52 bytes segment. The segment is divided into a 4 bytes segment header and a 48 bytes segment payload. The segment payload is further divided into a 2 bytes header, a 44 bytes segmentation unit, and a 2 bytes trailer. That is, the maximum amount of user data information that can be carried by each slot is 44 bytes. Thus, if the size of a packet generated by a station is greater than 44 bytes, it will have to be fragmented into blocks of 44 bytes. The format of the DQDB packet (or slot) is shown in Fig.2.1.

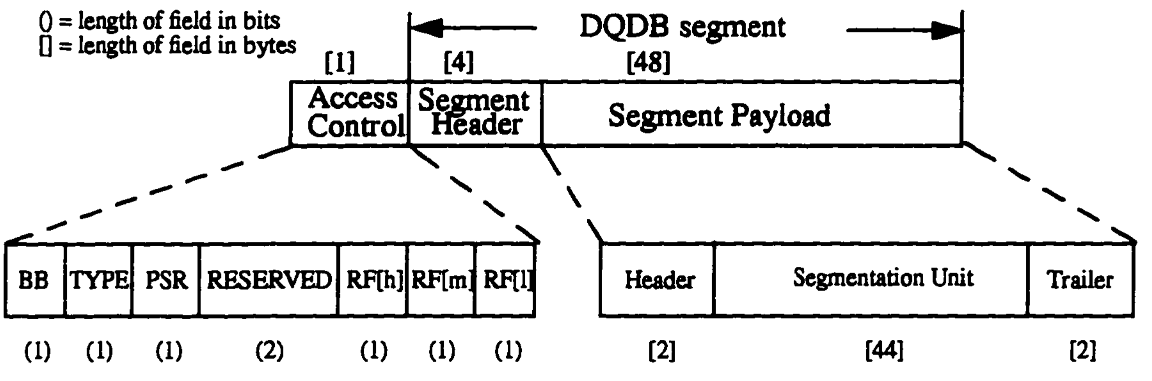

Fig.2.1 DQDB slot format

Important bits of the slot Access Control Field (ACF) are: a) the Busy Bit (BB) that indicates whether a slot is idle (BB = 0) or busy (BB = 1), i.e., it has already been written by an upstream station, b) the request fields (RF[h], RF[m], RF[l] for high, medium, and low priority, respectively) which indicate, on the reverse bus, whether a downstream traffic source of priority "i" requests (by setting RF[l] = 1) an idle slot from the upstream stations. In the sequel, we focus on the DQDB operation in the presence of a single priority traffic; i.e., only the RF[l] field can be set by an active station.

The DQDB mechanism [85] enables a downstream station to send a reservation request upstream for the first segment in its queue. Two counters per station, the Request

Counter (RQ_CTR) and the Count Down Counter (CD_CTR) control the transmission of the station on the forward bus. When a station is idle (i.e., does not have any packets to send), it increases its RQ_CTR by 1 for every request that it observes on the reverse channel and decreases this RQ_CTR by one (if it is greater than 0) for every idle slot it observes on the forward channel. In this way RQ_CTR keeps track of the downstream stations that have made requests for the transmission of their segments. When a new segment becomes first in the queue of a station (i.e., a new packet arrives at an idle station or a segment has just been transmitted and the next segment in the queue becomes first in the station's queue), the station transfers the content of RQ_CTR to CD_CTR and resets its RQ_CTR to 0. At the same time, it sends a request on the reverse bus to notify the upstream stations of the new queued segment. From this instant, the station decrements CD_CTR for every empty slot that arrives on the forward channel (bus A) and increments RQ_CTR for every request bit that arrives on the reverse channel. Whenever CD_CTR becomes 0, the station transmits its queued segment in the next idle slot that arrives on the forward channel.

The major advantage of the DQDB operation is that its maximum throughput is not sensitive to network parameters, i.e., it remains 1 regardless of the network size, number of connected stations or traffic patterns. However, DQDB suffers from a serious fairness problem. This is the extremely strong effect that the location of each station on the bus has on its throughput and delay performances. In order to illustrate this unfairness let us consider the dual bus network of Fig.1.1 but with only two stations, S_1 and S_2, separated by a distance of 20 slots. Assume that initially only station S_1 is active and overloaded, and acquires all the channel bandwidth. Then S_2 becomes active. S_2 will send a request for the first segment in its queue. This request will have to travel 20 slots upstream to S_1 to reserve one idle slot. 20 slots later, this idle reserved slot will arrive at S_2 and enable this station to transmit its first segment and send a new request for its next segment. That is, S_2 can receive only one slot every 1 round trip propagation time (i.e., 40 slots) while S_1 can transmit on all the other slots of this 40 slot cycle. Extensive investigations of the DQDB mechanism [37], [93] have shown that this unfairness is exacerbated as the network size, number of connected stations, or packet size increase. Furthermore, it has a detrimental effect on the ability of the network to support real-time traffic. For these reasons, a Bandwidth Balancing mechanism for DQDB (BWB_DQDB) was introduced which we briefly describe in the next section.

2.2.2 The BWB_DQDB Mechanism

The BWB_DQDB mechanism has also been included in the IEEE 802.6 standard [85]. This mechanism can provide the requested throughput to lightly loaded stations while, at the same time, can evenly distribute the remaining channel bandwidth among the overloaded stations. BWB_DQDB achieves this by requiring each station to increase the value of its RQ_CTR by an extra 1 every time it transmits BWB_MOD segments onto the channel. A Bandwidth Balancing Counter (BWB_CTR) is needed to indicate when a station should increase its RQ_CTR value. The artificial increase in the RQ_CTR value allows one extra empty slot to pass downstream and be written by the first active downstream station with CD_CTR = 0. In this way, upstream stations allow a greater number of idle slots to go to the downstream stations and the system can reach a steady state where the fair bandwidth allocation is achieved.

It is evident that the BWB_DQDB operation may waste channel bandwidth since in the absence of active downstream stations, the empty slots that upstream stations allow to pass will not be written by any station. Indeed, a performance analysis of BWB_DQDB, conducted in [93], has shown that as the value of BWB_MOD decreases the amount of bandwidth which is wasted increases. On the other hand, a small value of BWB_MOD can bring the system much faster to the steady state where the fair bandwidth allocation is achieved. It has also been shown in [93] that if different values of BWB_MOD are assigned to the various stations, their steady state throughputs will become proportionate to their BWB_MOD values. Nevertheless, the required bandwidth wastage and slow convergence speed to the steady state are major drawbacks of the BWB_DQDB mechanism and significantly affect its capability to support effectively multipriority traffic. The main reason is its slow responsiveness to changes of the traffic load which cannot protect the high priority traffic from transient overloads of the low priority traffic, although it can guarantee the steady state bandwidth requirements of the high priority traffic. Consequently, if the high priority traffic is generated by real-time applications (such as digital voice) it may not be capable of meeting its stringent delay requirements. These reasons motivated the introduction of the NSW_BWB and NSW_IUT bandwidth balancing mechanisms whose operation does not require the wastage of channel slots.

2.2.3 NSW_BWB and NSW_IUT MAC Mechanisms

The objective of the NSW_BWB [49] and NSW_IUT [48] mechanisms is to enable stations to know whether an idle slot will be written by a downstream station before they allow it to pass. Thus, no slot will be wasted, a small value of BWB_MOD can be used and the system can converge much faster to the steady state. The operation of both mechanisms requires the use of one additional control bit, the Transmit Additional Request (TAR) bit, in the ACF of each slot. According to both mechanisms, whenever a station S_i transmits its BWB_MODth segment, it sets the TAR bit to 1 in the written slot instead of increasing its RQ_CTR by one. The first active downstream station S_j that has segments in its queue (for which requests have not been sent) can erase the TAR = 1 bit and transmit an extra request upstream. This request will be seen by station S_i, which will now increase its RQ_CTR by one and allow an empty slot to pass to S_j. It should be noticed, however, that the request sent by station S_j will be seen not only by S_i but also by all other stations which are upstream from S_i which will also increase their request counters. The NSW mechanisms compensate these upstream stations by not allowing station S_i (which was responsible for the transmission of the extra request by S_j) to send a request for the next waiting segment in its queue, i.e., the one that follows the transmission of the TAR = 1 bit. This next segment can be transmitted only when S_i sees an idle slot that has not been reserved by the downstream stations (i.e., its RQ_CTR = 0), or when it observes a TAR = 1 bit; which will enable S_i to send an extra request and reserve an idle slot on the channel.

The NSW mechanisms can balance the bandwidth because they can guarantee that all overloaded stations will observe the same number of TAR = 1 bits on the channel. In both of them, a station is allowed to erase a TAR = 1 bit if and only if it is certain that will return it to the channel. That is, the station has enough segments in its queue whose transmission will enable this station to send a TAR = 1 bit downstream. In NSW_BWB, erasing a TAR = 1 bit is the only way for sending an extra request upstream. This approach, however, discriminates against the lightly loaded stations. For instance, consider a lightly loaded station with eight segments in its queue and a BWB_MOD value of eight. Let us also assume that a sequence of TAR = 1 bits arrives at the forward channel. Since the station has only 8 segments, their transmissions will generate only one TAR = 1 bit, and therefore this station can erase only one of the passing TAR = 1 bits. In contrast, a heavily loaded station, with a long queue, will be able to erase all of them and transmit a much greater number of requests on the reverse channel. For this reason, the NSW_IUT mechanism was introduced that allows a station to send an extra request upstream every time it sees a TAR = 1 bit on the channel. If the station's queue is also large enough and the station knows that will return a TAR = 1 bit onto the channel, it will also erase the passing TAR = 1 bit. Otherwise, it will allow the TAR = 1 bit to pass to the downstream stations. The investigation of the performance of the NSW_IUT mechanism in [48] has shown that it can significantly improve the delay performance of lightly loaded stations. Furthermore, it can drastically reduce the effect that the station location has on both throughput and delay performance.

The main problem of the NSW mechanisms is that downstream stations are required to send requests upstream and to wait for the reversed slots to arrive before they can start transmitting their segments. This requirement was introduced by the desire to eliminate bandwidth wastage. In this dissertation, however, we will introduce a buffer insertion technique, which we will call Buffer Insertion BWB mechanism (BI_BWB), that has the potential of providing a fair bandwidth allocation similar to NSW_IUT, (i.e., it does not waste any channel bandwidth) and, at the same time, can allow downstream stations to have a much faster access to the transmission medium. The key idea in BI_BWB is to allow each station, that has segments in its queue and observes a TAR = 1 bit, to delay the passing busy slot into a local shift register thus creating an idle slot on which it can write its own segment. In this way, the station can access the channel immediately. Notice, however, that because the size of the bus has now increased by one slot, the station will also send one request upstream to reserve an idle slot whose arrival at the station will enable this station to restore the size of the bus to its normal size. The detailed description of the BI_BWB operation will be presented in Chapter 3.

2.3 MAC Mechanisms for Dual Ring Networks

In this section, we provide a brief description of the most prominent mechanisms that have been proposed in the literature and elaborate on their limitations which clearly demonstrate the need for more efficient mechanisms.

2.3.1 The Global Fairness Algorithm (GFA)

Variations of this access mechanism have been proposed for MAGNET [55], ORWELL [29], ATMR [69] and METARING [18]. The Global Fairness algorithm [18] views the entire network as a single resource and tries to provide all stations with equal transmission opportunities. Its operation is based on a control message, called SAT (from SATisfied), which is forwarded around the ring and regulates the access of the stations to the network. We point out that the SAT is transmitted in the opposite direction of the data, i.e., the data transmission on ring A is regulated by the circulating SAT on ring B. Between two successive arrivals of the SAT messages at a station, the station can transmit at least l segments and at most k segments; where l and k are system parameters. A segment transmission counter (ST_CTR) at each station is reset to 0 every time this station forwards the SAT upstream. Whenever a station transmits a segment, its ST_CTR is increased by one. If a station sees an empty slot in the channel, it is allowed to transmit a segment only if its own ST_CTR is less than k. If a station S_i receives a SAT, it will forward the SAT upstream immediately unless it has packets waiting in its queue and its ST_CTR is less than l. In this case it will hold the SAT. Since every station may only transmit at most k segments before it receives another SAT message again, the upstream from S_i stations will eventually become idle, and S_i will be able to see idle slots and transmit its packets. As soon as a station has transmitted l segments, or its output queue is empty, it will forward the SAT message upstream immediately.

The main drawback of the Global Fairness Algorithm (GFA) is the sensitivity of its performance to the system parameters. For instance, consider a 100 km, 100 Mbps network consisting of 100 stations with a signal propagation delay equal to 5 μsec/km and a slot size equal to 500 bits. Then, the transmission time of a slot will be $500/100 = 5\mu\text{sec}$ and the round trip delay 100 slots.

It is evident that in order for GFA to maintain the ring utilization to 1, a value of k equal to 100 slots should be selected. In this way, in the presence of only one active station on the ring, the SAT will return to the active station at the instant it transmits the last of a group of 100 segments thus enabling this station to renew its 100 segment quota. Hence the station will transmit continuously. But then, in a worst case traffic scenario, a station may have to wait for each one of a half of its upstream stations[1] to transmit 100 segments

1. This is because for the other half of the upstream stations, the shortest path to reach the stations which are downstream from our tagged station will be through the other ring; not the one the tagged station wants to transmit its segments.

before it can see empty slots and start the transmission of its own segments. Thus, the requirement for efficient channel utilization may result to significant delays for certain stations.

2.3.2 The Local Fairness Algorithm (LFA)

The Local Fairness Algorithm (LFA) [16] has been introduced to provide high throughput without compromising the stations' ability to access the channel. This mechanism views the network as a distributed collection of communication resources. It is triggered only when potential bandwidth starvation is observed and is usually restricted to the ring sections that contain the stations which are competing for the channel. The three major issues in this mechanism are the following: a) how to detect the existence of starvation, b) how to find the ring segments that contain the competing stations, and c) how to detect the end of the starvation.

LFA [16] provides the following answers to the above questions: A station is in starvation when it has packets to transmit but it cannot access the network because it finds the medium to be continuously busy. Each station alternates between two modes of operations: a) the ***nonrestricted mode***, in which a station can transmit at any time as soon as it sees an empty slot and b) the ***restricted mode***, in which a station can transmit up to ***k*** segments. Normally, each station is operating in the ***nonrestricted mode***. When a station becomes starved, it switches to the ***restricted mode*** of operation and generates a starvation announcement control message which is sent to its upstream stations on the reverse channel. Each upstream station S_i that receives this control message enters the ***restricted mode***. Furthermore, if the channel is busy at this instant (due to transmissions by other upstream stations), S_i forwards the control message upstream. Otherwise, S_i removes the message from the channel thus terminating the overloaded ring section. The ***restricted mode*** ends when each of the involved stations has sent at least l segments. It should be noticed that deadlock may occur when all the stations in the ring network have entered the ***restricted mode***. Such a case may appear when no station sees an empty channel at the instant it receives the starvation announcement message which, in this way, is forwarded around the ring. The LFA mechanism solves this deadlock problem by including a REQ_ID parameter in the control message that is sent upstream. The REQ_ID identifies the station that generates the control message (i.e., requests the switching to the ***restricted mode***) and thus is responsible (i.e., after a complete rotation) for removing it from the channel.

The LFA algorithm can achieve a higher network throughput than the global fairness mechanism in most of the cases. Nevertheless, it suffers from three serious problems:

a) It is difficult to determine when starvation starts. The statement used in the LFA [16] to define starvation, i.e., "starvation is observed when a station has something to transmit but cannot access the network because its upstream link is continuously busy", is very vague. The problem is that it is very difficult to define quantitatively what "continuously" means. For instance, one can define "continuously" as the situation where a station sees the channel busy for w consecutive slots. Notice, however, that in such a case an upstream station may regulate its transmission in such a way so that it can acquire most of the channel bandwidth; i.e., this station can write on $w - 1$ slots and then allow an idle slot to pass by so that its downstream station will never trigger the ***restricted mode*** of operation.

b) It is not really fair. The maximum throughput of a station (under LFA) is still strongly affected by the channel speed and the location of the station in the network. For instance, consider the two station ring segment shown in Fig.2.2. Assume that both stations S_1 and S_2 are overloaded, the distance between them is ***d*** time slots, and that S_2 becomes starved when it observes ***w*** consecutive busy slots on ring A. At this instant S_2 will send the starvation announcement control message to S_1 (through ring B). When this message arrives at S_1 (***d*** time slots later), S_1 will transmit ***k*** additional segments (on ring A) and then stop, waiting for the SAT message from S_2; we should notice that S_1 has transmitted ***d+k*** segments from the moment S_2 observed starvation. S_2 will transmit ***l*** segments, and then send the SAT signal upstream which will arrive at S_1 ***d*** time slots later and enable S_1 to start transmitting. Until the busy slots written by S_1 arrive at S_2, S_2 will have the opportunity to write on an additional ***d*** slots. Then S_2 will have to wait for ***w*** busy slots to pass by before it becomes starved again and generates the starvation announcement control message. It is evident that during each cycle of the above operation (where the cycle is defined by two consecutive transmissions of the starvation announcement control message), station S_1 transmits ***2d+k+w*** segments while S_2 transmits ***l+2d*** segments. Since $k > l$ and $w > 0$, the throughputs of the two stations will be different.

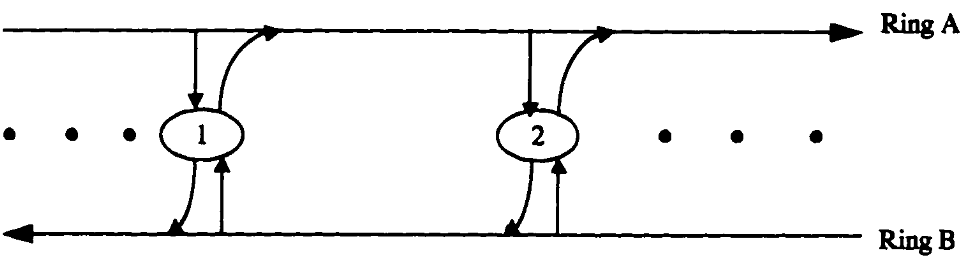

Fig.2.2 Throughput unfairness under the Local Fairness Algorithm

c) The maximum number of stations that can be connected into the network is determined by the size of the REQ_ID field in the Access Control Field of the packet. Although this might be a minor problem in the presence of one traffic class (for instance, 10 bits enable connection of $2^{10} = 1024$ stations), it may become more significant in the case of multipriority classes of traffic.

2.4 Priority Mechanisms

In this section, we provide a brief description of some prominent priority mechanisms which have been proposed for high speed networks and discuss their limitations.

2.4.1 DQDB Priority Mechanism

The DQDB priority mechanism has been included in the IEEE 802.6 MAN standard. It requires a separate request bit (see Fig.2.1), request counter (RQ_CTR), and countdown counter (CD_CTR) for each priority class. If a class is idle, its RQ_CTR will count its own and higher priority requests on the reverse channel. If a class is active, its RQ_CTR will count only the requests of its own priority, while its CD_CTR will count the requests of higher priority.

The main problem of the DQDB priority mechanism is that it cannot guarantee higher throughputs and lower delays for higher priority classes in a high speed long distance network [7], [14], [89], [90]. That is, the station location continuous to have a very strong effect on both throughput and delay performance.

2.4.2 The Global Priority Information BWB (GPI_BWB) Mechanism

The GPI_BWB mechanism [38] is based on BWB_DQDB, and is among the most effective priority mechanisms. GPI_BWB allows a high priority class to consider the busy slots of lower priority that it observes on the channel as equivalent to idle slots that it should allow to go downstream. In this way, higher priority classes allow fewer slots to go to the downstream lower priority classes and their steady state throughput performance becomes independent of the presence of the low priority traffic. However, the station location still has a strong effect on the high priority traffic delay. Moreover, because this mechanism also converges slowly to the steady state (since it is based on BWB_DQDB), it cannot effectively protect time-critical traffic from sudden overloads of the low priority traffic.

2.4.3 The Timed Token Rotation (TTR) Priority Mechanism

The TTR mechanism [2] is used in the FDDI dual ring networks. It assigns a different value of Target Token Rotation Time (TTRT) to each class. Higher priority classes acquire a larger value of TTRT. The operation is token passing. The maximum amount of traffic a class can transmit, every time it captures the token, is determined by the difference between the value of TTRT assigned to this class and the duration of the most recent token rotation. Since high priority classes have larger values of TTRT they can transmit more traffic during each token rotation, even completely shut off the low priority traffic transmissions. This mechanism is fair because of the token passing, but it is not efficient for network with high bandwidth-latency product. The reason is that the token has to go around the entire network which introduces a ring latency sensitive overhead. In addition it does not support bandwidth spatial reuse.

2.4.4 The Time Based Priority Global Fairness Algorithm (TBP_GFA)

The TBP_GFA has been proposed for dual ring architectures in [94]. It tries to combine the features of the TTR priority mechanism of section 2.4.3 with those of the Global Fairness Algorithm (GFA). Its operation is based on two control signals: SAT and ASYNC_EN, which circulate in the opposite ring than that of the traffic they regulate. ASYNC_EN with attributes GREEN, YELLOW or RED, is used for enabling or disabling the transmission of the asynchronous traffic while the SAT is used for ensuring fairness among the stations generating synchronous traffic according to the Global Fairness Algorithm. The key idea in TBP_GFA is the introduction of a T_{min} time period which is equal to R times the round trip propagation delay, where R is an input parameter. If the synchronous traffic is not satisfied during this T_{min} period, the transmission of asynchronous traffic will be halted. The GBP_GFA operation is also based on network-wide fairness cycles. Therefore, it is also very sensitive to the ring propagation delay.

CHAPTER 3

THE BUFFER INSERTION BWB MECHANISM FOR DUAL BUS ARCHITECTURES

3.1 Introduction

The primary objective of the Buffer Insertion Bandwidth Balancing (BI_BWB) mechanism is to improve the delay performance of downstream stations in large dual bus networks. The NSW_IUT mechanism [48] has already provided a significant improvement over the NSW_BWB mechanism in the delay performance of the downstream stations. This is because it allows stations, even when they are lightly loaded, to send extra requests upstream as soon as they observe the TAR = 1 bits on the channel. However, these requests still have to travel upstream and force the upstream stations to allow idle slots to go downstream. That is, in the presence of overloaded upstream stations, it will take a round trip propagation delay before an active downstream station starts seeing idle slots. For instance, in a large MAN of 200 km with a 5 μsec/km signal propagation delay, it may take $200 \bullet 2 \bullet 5 = 2000\mu\text{sec} = 2\text{msec}$ for a station at the end of the bus to observe an idle slot, if the most upstream station is heavily loaded. Such a delay may not be tolerable for some real-time applications. In Fig.3.1, we provide a more quantitative feeling of the effect that the location of a station on the bus has on its average packet delay under both the NSW_IUT mechanism and the BWB mechanism of DQDB (BWB_DQDB). We consider an 1 Gbps, 20 stations network with a distance between neighbor stations equal to 10 slots. The slot size is 53 bytes, the signal propagation delay 5 μsec/km, and the value of the BWB_MOD parameter equal to 2.

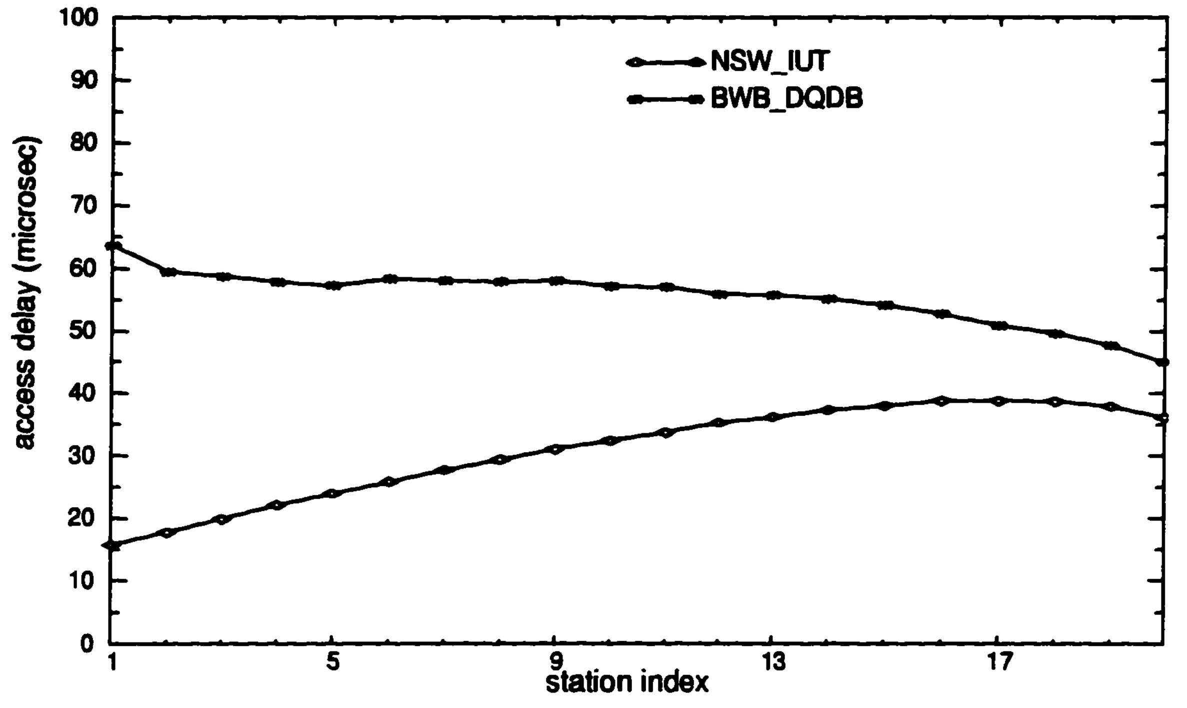

Fig.3.1 Delay comparison of BWB_DQDB and NSW_IUT mechanisms

In Fig.3.1, we consider stations that generate independent segments. The total offered load is 0.9 segments/slot uniformly distributed among the stations. Fig.3.1 shows that under NSW_IUT, the delays encountered by the downstream stations are higher than those encountered by the upstream stations. In contrast, under BWB_DQDB, the upstream stations encounter higher delays. This is due to the significant number of idle slots that upstream stations allow to go downstream under the BWB mechanism. Because a large number of these slots is wasted, the delays of the stations are significantly higher under BWB_DQDB.

The objective of the Buffer Insertion BWB mechanism is to allow downstream stations to have almost immediate access to the channel. The main idea on which the BI_BWB operation is based is the following: stations can temporarily store incoming written slots locally and replace them with their own transmissions. In order to achieve that, a station delays a written slot into a shift register and retransmits it onto the channel as soon as it has completed its own transmission. We see that the BI_BWB operation increases the bus latency by one slot every time a station inserts its own packet segment into the transmission path. The bus latency is decreased in the following way: every time a station inserts an idle slot, it also sends a request upstream to reserve an idle slot from the upstream stations. The arrival of this slot at the station will enable this station to decrease the bus latency by one slot.

The BI_BWB operation tries to combine the advantages of both the BWB_DQDB and NSW_IUT mechanisms. We remind the reader that the main advantage of BWB_DQDB is that it forces each station to immediately allow an idle slot to pass by so that downstream stations can use it. Its disadvantage is that because of the potential bandwidth wastage, the station must use a relatively large value of BWB_MOD. NSW_IUT, on the other hand, does not waste any channel bandwidth, and thus it can use a small value of BWB_MOD. Its disadvantage is that a downstream station must first make a reservation and then wait for the reserved idle slot to arrive. Consequently, in overload conditions, downstream stations will have to suffer an initial round trip propagation delay before accessing the channel. In addition, stations do not have any control over the delay of their packets. They can transmit only when they see an empty and unreserved slot. That is, once an upstream station has inserted a segment, the slot that this segment occupies will not be available to any other station. Thus access delays of the downstream stations depend only on the total load of the network and their locations on the bus.

The proposed BI_BWB can avoid bandwidth wastage while enabling downstream stations to access the channel as fast as they would do under the BWB_DQDB mechanism using a small value of BWB_MOD. By replacing the upstream stations' segment transmissions on the channel with its own queued segments, a downstream station can access the channel immediately even when its location follows that of an overloaded station. Once the bus latency has increased by one slot due to a buffered segment, there are several parameters which can be set to determine when the station must remove its inserted buffer space from the bus path. Different values of these parameters will have a different impact on the delays of the stations in the network. Consequently, the BI_BWB mechanism has a much greater flexibility in controlling the stations' delay performance than the NSW_IUT mechanism. In fact, NSW_IUT becomes a special case of the BI_BWB mechanism that we introduce. The organization of the rest of this chapter is as follows. In section 3.2, we present the main characteristics of the BI_BWB mechanism. In section 3.3, we provide the detailed description of the BI_BWB access algorithm and in section 3.4 we elaborate on its operation. In section 3.5, we use simulation results to investigate its throughput and delay performance. In section 3.6, we present a queuing analysis that can derive accurate estimate of the segment delay. Finally, in section 3.7, we present our conclusions.

3.2 The BI_BWB Mechanism

The BI_BWB operation incorporates features of the NSW_IUT [48] operation. That is, it also allows each station to send multiple requests upstream and divides its waiting segments into ***registered*** and ***unregistered*** segments. The segments for which requests have been sent are called ***registered***. The segments for which requests have not yet been sent are called ***unregistered***. Two counters, the RG_CTR and UNRG_CTR, count the number of ***registered*** and ***unregistered*** segments, respectively, at each station. When a station sends a request, its UNRG_CTR decreases by 1 and its RG_CTR increases by 1.

There are two ways a station can send a request upstream: a) when a segment becomes first in the queue, and b) when the station observes a TAR = 1 bit on the channel. As in NSW_IUT, the TAR bit has been introduced into the Access Control Field (ACF) of the slot to allow waste free bandwidth balancing operation among the upstream and downstream stations. This is achieved by providing each station with a Bandwidth Balancing Counter (BWB_CTR) which is increased by 1 every time the station transmits a segment. When BWB_CTR becomes equal to the bandwidth balancing parameter BWB_MOD, the station sets the TAR bit to 1 on the written slot and resets its BWB_CTR to 0. However, when an active station receives a segment with TAR = 1 from upstream, instead of forwarding the passing TAR segment downstream, it delays the busy slot into a local shift register (called the Insertion Buffer) thus creating an idle slot. This idle slot corresponds, in a way, to the idle slot that an upstream station allows to go downstream in the case of the BWB_DQDB mechanism. This idle slot can now be used either by a downstream station (if the station's RQ_CTR > 0) or the station itself to transmit a segment. We should keep in mind that a station observing a passing TAR = 1 bit always sends an additional request upstream whenever its UNRG_CTR > 0. This means that if a station writes on an idle slot which was created by inserting an upstream busy slot into its shift register, the segment it transmits is always a registered segment, i.e., a segment for which a request has been sent. Therefore, whenever an idle slot is inserted into the transmission path by a station, a request is always sent upstream for a free slot reservation regardless of whether the inserted idle slot is going to be used by a downstream station or the station itself. This reservation is made either by a downstream station, when RQ_CTR > 0, or by the station itself, when RQ_CTR = 0 and the station transmits a segment. This reserved idle slot will enable this station, upon its arrival, to decrease the bus latency back to its original size.

Another action that a station observing a passing TAR = 1 bit on the channel may take is to reset it to 0. The station will do this if it is certain that it can return a TAR = 1 bit to the channel through the transmission of its currently waiting segments. For this purpose, the parameter N_TAR is introduced which provides the number of TAR segments at the station's queue. TAR segment is a segment whose transmission will make the BWB_CTR equal to BWB_MOD, and therefore will have its TAR bit set to 1. It is evident that the value of N_TAR is given by:

$$N\_TAR = \lfloor (BWB\_CTR + UNRG\_CTR + RG\_CTR) / (BWB\_MOD) \rfloor \qquad (3.2)$$

where $\lfloor X \rfloor$ is the integer part of X.

In order for a station to determine whether it should reset a passing TAR = 1 bit, it must also know (in addition to the value of N_TAR) how many TAR = 1 bits it has already erased, and therefore owes to the downstream stations. This information is provided by the Debit TAR Counter (DBTAR_CTR) which increases by 1 every time the station resets a TAR = 1 bit, and decreases by 1 (if it is greater than 0) every time a station transmits a TAR = 1 bit onto the channel. It is now evident that a station can reset a passing TAR = 1 bit, if and only if, N_TAR > DBTAR_CTR. We conclude this section by pointing out that, in the case of BI_BWB, a station can transmit a segment only when its request counter

RQ_CTR is equal to 0. The objective is to give a higher priority to the service of requests, and thus slightly compensate the downstream stations for their unfavorable locations on the bus. In the sequel, we provide a detailed description of the buffer insertion and buffer size reduction mechanisms.

3.2.1 The Buffer Insertion Mechanism

When a busy slot carrying a TAR = 1 bit is seen on the forward channel and the station's transmission queue is not empty, the station is allowed to increase its buffer size by storing the written slot of the channel into a local shift register. This creates an idle slot which can be used: a) by a downstream station (if RQ_CTR > 0), b) by the station itself, or c) by the local shift register thus reducing the size of the insertion buffer by 1. A flag, called **Tx_Order** with value 0 or 1, is used to indicate whether the station is allowed to transmit a queued segment or decrease its insertion buffer size by 1. Initially, the value of **Tx_Order** is 0, which means the station is allowed to transmit from its own queue.

We should notice that the Buffer Insertion Mechanism maintains the order of the upstream written slots in the following way. Once a station has inserted its shift register into the transmission path, all the upstream busy slots will have to go through this shift register. That is, the train of written slots snakes its way in and out of this station.

3.2.2 Buffer Size Reduction Mechanism

Whenever a station receives an idle slot from upstream and its RQ_CTR is 0, it has the option of either transmitting a segment from its own queue or reducing the shift register size by transmitting the first segment of its insertion buffer. If the station always transmits the segments in the buffer first, upstream stations may be favored. In contrast, if the station always transmits the segments in its own queue first, downstream stations will be favored.

The BI_BWB mechanism introduces two new parameters, LMOD and BMOD, to control the number of segments the station can transmit from its local queue and insertion buffer, respectively. Two counters, the LOCAL_CTR and BF_CTR, are used to count the number of segments the station has transmitted from the corresponding local queue and buffer. LOCAL_CTR and BF_CTR are reset to 0 whenever their values become equal to LMOD and BMOD, respectively. In order to distribute evenly in time the transmission of the local and buffered segments, the afore mentioned **Tx_Order** flag is used to determine the queue from which the station should transmit. If $\left(\frac{LOCAL\_CTR}{LMOD}\right) < \left(\frac{BF\_CTR}{BMOD}\right)$, then **Tx_Order** = 0 and the station will transmit from its local queue. Otherwise, **Tx_Order** = 1 and the station will transmit from its insertion buffer. In order to clarify this operation, let us assume that LMOD = BMOD = 2 and that LOCAL_CTR = BF_CTR = 0. Let us also assume that both the transmission queue and insertion buffer have more than 2 segments, and that RQ_CTR = 0, i.e., no reservation has been made by any downstream station. Then, upon the arrival of the first idle slot, the station: a) will transmit a segment from its insertion buffer since $\left(\frac{LOCAL\_CTR}{LMOD} = 0\right) = \left(\frac{BF\_CTR}{BMOD} = 0\right)$, b) increase BF_CTR by 1. Upon the arrival of the second idle slot, the station: a) will transmit from its local queue since $\left(\frac{LOCAL\_CTR}{LMOD} = 0\right) < \left(\frac{BF\_CTR}{BMOD} = \frac{1}{2}\right)$, b) set LOCAL_CTR = 1. Since $\left(\frac{LOCAL\_CTR}{LMOD} = \frac{1}{2}\right) = \left(\frac{BF\_CTR}{BMOD} = \frac{1}{2}\right)$, the arrival of the next idle slot will result in a transmission from the local queue and so on. It is easily seen that for general values of LMOD and BMOD, the above algorithm will allow a station to transmit LMOD segments from its local queue for every BMOD segments it transmits from its insertion buffer. Furthermore, their transmission will be evenly distributed in time.

It must be pointed out that the values of LMOD and BMOD must satisfy the following two conditions in order for the bandwidth balancing to hold: a) BMOD > 0, b) $\frac{LMOD}{BMOD} \leq BWB\_MOD$. We should keep in mind that BMOD = 0 means that a station will always transmit from its own local queue before it can transmit any segment from its insertion buffer. Thus, if we allow BMOD to be equal to 0, an overloaded station can buffer all the upstream segments and replace them with its own transmissions. Consequently, bandwidth balancing will never be achieved. Therefore, the value of BMOD must be greater than 0. Condition "b)" must also hold since the arrival of a TAR = 1 bit indicates that an upstream station has transmitted its BWB_MODth segment and is willing to allow an idle slot to go downstream. A downstream station, which has buffered this TAR segment and has finished the transmissions of its own BWB_MODth segment, must return the buffered upstream segment back to the channel. Otherwise, it will acquire more bandwidth than the upstream stations. That is, a station can transmit at most BWB_MOD segments before it returns one buffered segment to the channel. The BI_BWB operation allows the transmission of at most LMOD local segments before the transmission of BMOD buffered segments. Therefore, the value of LMOD must be at most $BMOD \bullet BWB\_MOD$ in order for bandwidth balancing to be achieved.

It is evident from the previous discussion that the BI_BWB mechanism enables a station, through the selection of the BMOD and LMOD parameters, to have a much greater control over its channel access delay (relative to the other access mechanisms). The effect of the values of these two parameters on the delay will be examined during our performance investigation of BI_BWB.

3.3 The BI_BWB Access Algorithm

In this section, we present the BI_BWB access algorithm. First, we describe the various parameters and counters. These are:

- ***LOCAL_CTR:*** Local Counter. It increases by one for every segment which is transmitted by the station's local queue. It is reset to 0 whenever LOCAL_CTR = LMOD and BF_CTR = BMOD or the insertion buffer is empty.
- ***BF_CTR:*** Buffer Counter. It increases by 1 for every segment which is transmitted from the insertion buffer. It is reset to 0 whenever BF_CTR=0 and LOCAL_CTR=LMOD or the station's local queue is empty.
- ***Tx_Order:*** This is a flag whose value indicates the queue from which the station should transmit a segment. **Tx_Order** = 0 means that the local queue has higher priority. **Tx_Order** = 1 means that the insertion buffer has higher priority.
- ***RG_CTR:*** Registered Counter. It counts the number of registered segments in a station's queue. A registered segment is a segment for which a request has been generated. This request may have already been sent upstream or it may be waiting in the station's request queue for transmission on the reverse channel.
- ***REQ_QS:*** Request Queue Size. It provides the number of requests that a station has to send upstream.
- ***UNRG_CTR:*** Unregistered Counter. It counts the number of unregistered segments in a station's queue. An unregistered segment is a segment for which a request has not yet been generated.

- ***BB:*** Busy Bit. It indicates whether a slot has already been written by an upstream station on the forward channel.
- ***RF:*** Request Field. It indicates whether a slot on the reverse channel is carrying a request sent by a downstream station.
- ***RQ_CTR:*** Request Counter. It provides the number of slot reservations that have been made by the downstream stations. RQ_CTR increases by 1 for every request seen on the reverse channel and decreases by 1 (if RQ_CTR > 0) for every idle slot seen on the forward channel. It should be noted that under BI_BWB, a station can transmit if and only if its RQ_CTR = 0.
- ***TAR bit:*** Transmit Additional Request (TAR) bit. It is set to 1 every time a station transmits the last of a group of BWB_MOD segments. It allows downstream stations to send extra requests upstream.
- ***BWB_CTR:*** Bandwidth Balancing Counter. It increases by one every time a station transmits a segment. If BWB_CTR becomes equal to BWB_MOD, the station will set the TAR bit to 1, on the transmitted segment, and reset the BWB_CTR to 0.
- ***NTAR_R:*** Number of Available TAR Segments Register. It provides the number of TAR segments in the station's queue; see equation (3.1).
- ***DBTAR_CTR:*** Debit TAR Bit Counter. It provides the number of TAR = 1 bits the station has erased and must return to the channel. It increases by one whenever a station resets to 0 a passing TAR = 1 bit and decreases by 1 (if it is greater than 0) whenever a station sends a TAR = 1 bit onto the channel.

We now provide a complete presentation of the BI_BWB access algorithm by describing the reaction of each station to various events.

a) ***New packet arrival:*** the UNRG_CTR is increased by the number of segments in the packet.

b) ***A segment becomes first in the transmission queue:*** if UNRG_CTR > 0 and BWB_CTR < BWB_MOD-1, a request will be sent upstream, RG_CTR will increase by 1 and UNRG_CTR will decrease by 1.

c) ***A slot arrives at the forward bus:***

c1: If the slot is busy and its TAR bit is set to 1, the station will do the following:

i) If **Tx_Order** = 0 (i.e., transmission from local queue) and the local queue is not empty, the station will buffer the channel segment at the end of its insertion buffer and reset the busy bit to 0. Otherwise (**Tx_Order** = 1), the station will increase its BF_CTR by 1. In addition, if $\left(\frac{LOCAL\_CTR}{LMOD}\right)$ is less than $\left(\frac{BF\_CTR}{BMOD}\right)$ and $(LMOD \neq 0)$, the station will set the Tx_Order to 0. Finally, if $BF\_CTR \geq BMOD$ and either the local transmission queue is empty or $LOCAL\_CTR \geq LMOD$, the station will set both BF_CTR and LOCAL_CTR to 0.

ii) If UNRG_CTR > 0, the station will send a request upstream, decrease UNRG_CTR by 1 and increase RG_CTR by 1. In addition, if NTAR_CTR is greater than DBTAR_CTR, the station will reset the TAR = 1 bit of the buffered segment and increase DBTAR_CTR by 1.

c2: If the channel slot is still busy (i.e., has not been reset by the conditions in "***c1***") and the station's insertion buffer is not empty, the station will buffer the segment in the channel and transmit the first segment in its buffer.

c3: If the slot is free and RQ_CTR > 0, the station will decrease its RQ_CTR by one and let the idle slot go by. Otherwise (i.e., RQ_CTR = 0), the station will transmit a segment either from its own queue or from its insertion buffer, depending on the value of the **Tx_Order** flag. If **Tx_Order** = 0 or the insertion buffer is empty, the station will transmit from its own queue. If **Tx_Order** = 1 or the station's own queue is empty, it will transmit from its insertion buffer.

d) <u>A slot is seen on the reverse bus:</u>

d1: If the slot carries a request, the station will increase RQ_CTR by one.

d2: If the request field of the slot is empty and the station's REQ_QS is greater than zero, the station will decrease REQ_QS by one and send a request upstream.

e) <u>The station transmits a segment from its local queue:</u>

e1: First, the station will increase BWB_CTR by 1. If BWB_CTR becomes equal to BWB_MOD (i.e., the station transmits a TAR segment), then: i) it will reset the BWB_CTR to 0, ii) if DBTAR_CTR > 0 or both DBTAR_CTR and RG_CTR are 0, it will set the TAR bit to 1, iii) if DBTAR_CTR > 0, it will decrease DBTAR_CTR by 1.

e2: If RG_CTR > 0, the station will decrease RG_CTR by 1. Otherwise, it will decrease UNRG_CTR by 1.

e3: LOCAL_CTR will increase by one. If $\left(\frac{LOCAL\_CTR}{LMOD}\right) > \left(\frac{BF\_CTR}{BMOD}\right)$, the station will set **Tx_Order** = 1. If $LOCAL\_CTR \geq LMOD$ and either the Insertion Buffer is empty or $BF\_CTR \geq BMOD$, the station will set both the BF_CTR and LOCAL_CTR to 0.

f) ***The station transmits a segment from its insertion buffer:*** the station will increase its BF_CTR by 1. If $\left(\frac{LOCAL\_CTR}{LMOD}\right) < \left(\frac{BF\_CTR}{BMOD}\right) and (LMOD \neq 0)$, the station will set the Tx_Order = 0. If $BF\_CTR \geq BMOD$ and either the local transmission queue is empty or $LOCAL\_CTR \geq LMOD$, the station will set both the BF_CTR and LOCAL_CTR to 0.

3.4 BI_BWB Mechanism Discussion

Our performance analysis of the BI_BWB mechanism has shown that it can provide the required bandwidth to the lightly loaded stations, and evenly distribute the remaining bandwidth among the overloaded stations. This behavior is expected since buffering upstream segments and replacing them with the station's own transmissions is just a reordering of transmissions in the network. BI_BWB does not provide more transmission opportunities to each station than the corresponding NSW_IUT mechanism. The main difference between the two mechanisms is that BI_BWB buffers the passing TAR = 1 segments which are returned gradually into the channel. In the following, we will discuss the buffer size required by the BI_BWB operation and the throughput that BI_BWB provides to each station in the network.

3.4.1 Buffer Size

According to the BI_BWB operation, a station may store a segment at its insertion buffer for at most one roundtrip propagation time between the station and the upmost active station. This is because that: a) whenever a station buffers an upstream segment, a request has been sent or will be sent upstream for this segment; b) when the idle slot reserved for this segment arrives, the station either has transmitted the segment or will use this idle slot to transmit it. Consequently, the maximum size of a station's insertion buffer should be the roundtrip propagation time (in slots) between the station and the first active station in the bus.

We should notice that the BI_BWB operation guarantees that a request will be sent for every buffered segment because a station will buffer an upstream TAR = 1 segment only when it is the local queue's turn to transmit (**Tx_order** = 0) and this queue is not empty. If the first segment of the local queue is a registered segment, then a request has already been sent. If it is an unregistered segment, an extra request will be sent upstream and the first segment in the queue will become a registered segment.

The idle slot which is created will be either used by the station or, if RQ_CTR > 0, left to pass by to the downstream stations. In this last case, and because RQ_CTR > 0, a request has already been sent upstream by a downstream station that will reserve an idle slot and compensate our tagged station for the slot it has allowed to pass downstream. Thus, again, the request sent by the tagged station itself will create an idle slot whose arrival will enable this station to decrease the insertion buffer.

3.4.2 Bandwidth Allocation

It is evident from the previous discussion that under the BI_BWB operation the stations that insert busy segments into their shift registers create enough requests to guarantee the eventual removal of their registers from the transmission path. Therefore comparing BI_BWB to NSW_IUT, the BI_BWB operation simply rearranges the segment transmission order but not the average rate at which stations transmit segments on the forward bus and requests on the reverse bus. Consequently, BI_BWB has the same bandwidth allocation capabilities of the NSW_IUT mechanism. That is, it can guarantee the requested throughput to underloaded stations and provide throughputs for the overloaded stations which are proportionate to the values of their BWB_MOD parameters.

The BI_BWB operation is based on the presence of an Insertion Buffer. Therefore, it is very important to investigate the effect of the Insertion Buffer size on performance. We have examined the effect that the network size, station locations, and BWB_MOD parameters have on the average Buffer Insertion size ($\overline{BI_{size}}$) when all stations in the network are overloaded. We have found the following relation among these system parameters. When all the stations in the network are overloaded, and each station S_i always completes the transmission of its BWB_MOD[i]th segment before it returns a buffered upstream TAR = 1 segment to the channel (i.e., LMOD = BMOD • BWB_MOD[i]), then a very good estimate of the average buffer length of a station S_i is:

$$\overline{BI_{size}}(i) = \left(\sum_{i}^{N} THR_j\right) \bullet (D_{i-1,i} - 1) \tag{3.3}$$

where in equation (3.2), $D_{i,j}$ is the distance (in slots) between stations S_i and S_j (S_i is upstream of S_j), N is the total number of stations on the bus, and THR_i is the normalized throughput of station S_i. THR_i can be computed from the values of BWB_MOD at each station (i.e., BWB_MOD[i] for station S_i) through:

$$THR_i = \frac{BWB\_MOD\,[i]}{\sum_{K=1}^{N} BWB\_MOD\,(K)} \tag{3.4}$$

For instance, if we have three stations in the network with $D_{1,2} = 40\ \text{slots}$ and $D_{2,3} = 20\ slots$, and BWB_MOD[1] = BWB_MOD[2] = BWB_MOD[3] = 2, then the throughput of each of the stations will be $THR_1 = THR_2 = THR_3 = \frac{1}{3}$. Consequently, the average buffer size of station S_3 will be $\overline{BI_{size}}(3) = \frac{1}{3} \bullet (20-1) = 6$ and that of station S_2 will be $\overline{BI_{size}}(2) = \left(\frac{1}{3}+\frac{1}{3}\right) \bullet (40-1) = 26$. These values are similar to the corresponding values obtained from simulation results

3.5 BI_BWB Performance Analysis

In this section, we will use simulation results to investigate the performance of the BI_BWB mechanism in both overload and underload conditions. The primary performance parameters that are widely used in the investigation of access mechanisms for dual bus networks are the station's throughput and average segment or packet delay. The segment (packet) access delay is defined as the time interval from the instant a segment (or packet) arrives at the station until the instant this segment (or last segment of the packet) completes its transmission onto the channel. This delay measure is sufficient for expressing the performance of the various medium access control mechanisms, since other factors of the delay such as propagation delay to the destination station, or end of the bus, are fixed. However, in the case of BI_BWB, the above definition of delay alone may no longer provide a satisfactory measure of the delay performance. This is because transmitted segments may be delayed inside the shift registers of intermediate stations before they reach their destination or the end of the bus. Therefore, in our investigation of the BI_BWB mechanism, we will consider what we have called Access/Storage delay. This delay, in the case of individual segments, includes the above access delay plus the time a segment spends inside the shift registers of the downstream stations. Thus, in our definition of the Access/Storage delay, we also do not include the source to destination propagation delay which is a fixed component of the delay. In the case of a packet, the Access/Storage delay is the time interval from the instant the packet is generated until the instant its last segment arrives at the destination, minus the source to destination propagation delay.

3.5.1 Transient Analysis of BI_BWB

In this section, we use simulation results to investigate the transient behavior of the BI_BWB mechanism. We consider a high-capacity network of 155 Mbps, a slot size of 53 bytes, and a signal propagation delay of 5 $(\mu s)/(km)$. In Fig. 3.2, we show the convergence speed of the basic BI_BWB when there are three stations present on the bus and at a distance of 20 slots, corresponding to a cable length of 10.9 km between neighbor stations. The values of the BMOD and LMOD parameters at each station are 1 and 2, respectively. We also assume an infinite insertion buffer size. The horizontal axis represents time, measured in slots, with ticks appearing in multiples of the end-to-end propagation delay. Initially, only station S_1 is active and overloaded. It acquires all channel bandwidth. At time

$t = 5 \bullet t_{prop} = 200$, station S_2 becomes active and tries to acquire all the channel bandwidth. Finally, at $t = 10 \bullet t_{prop} = 400$, station S_3 becomes active and tries to acquire all the channel bandwidth.

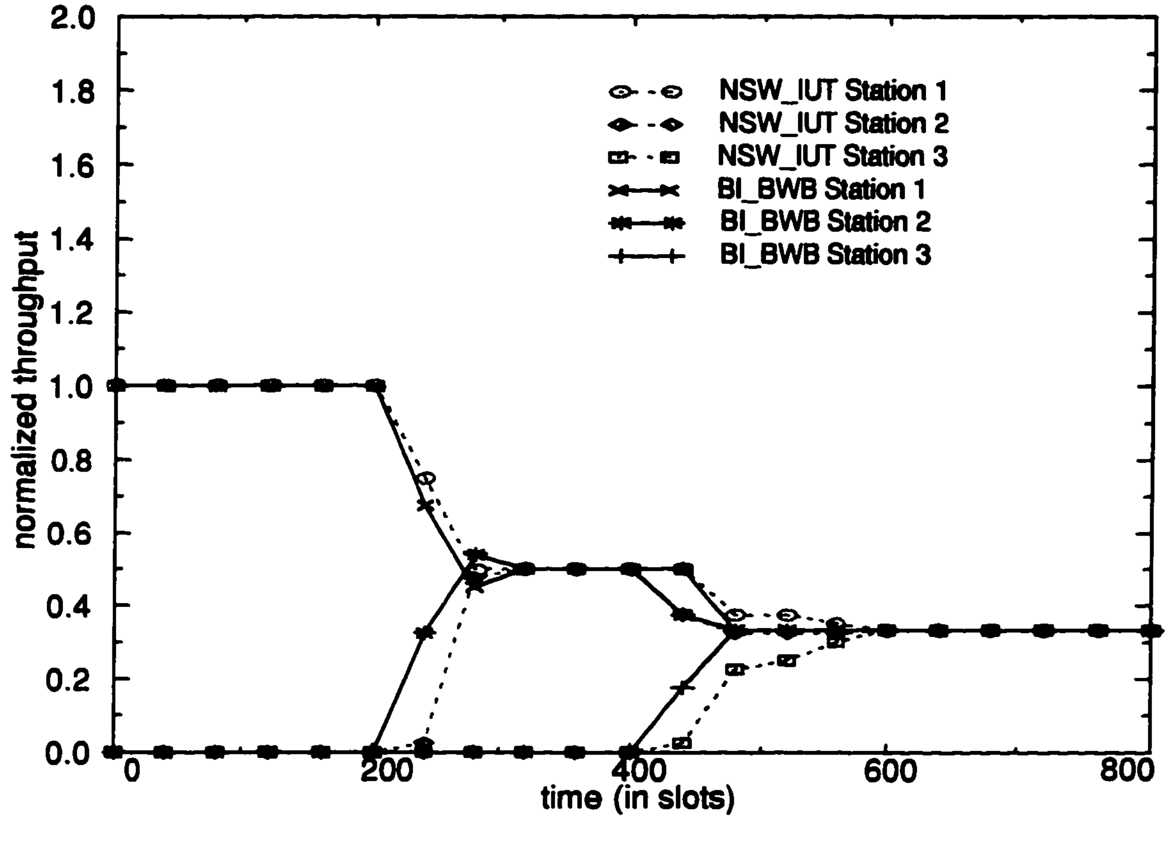

Fig.3.2 Throughput performance. Comparison of BI_BWB and NSW_IUT. $D_{12} = D_{23} = 20$ slots

Fig.3.2 shows that BI_BWB enables downstream stations to have almost immediate access to the channel and reach steady state much faster than in the case of the NSW_IUT mechanism. We point out that our performance analysis of the transient behavior of BI_BWB has shown that as the ratio $\frac{LMOD}{BMOD}$ increases the convergence speed to steady state also increases. The reason is that a higher value of LMOD enables a station to replace more upstream segments with its own local segments before it has to return them back to the channel. For instance, in the network of Fig.3.2, LMOD/BMOD = 2. Then, when station S_2 becomes active, it can buffer an incoming TAR=1 segment immediately. Furthermore, it will keep this upstream segment buffered until it completes the transmissions of LMOD =2 local segments. It should be noted that had LMOD been 0, the station would not have been able to buffer any upstream segment and the converging speed of the BI_BWB would have been the same with that of NSW_IUT. We also notice that the throughput of station S_2 is slightly higher than 0.5 at first. This is due to the existence of the insertion buffer. At the time station S_1 is active and S_2 is idle, the LOCAL_CTR of station S_2 is 0 and the BF_CTR of S_2 is greater than 0. When S_2 becomes active, it will always transmit from its local queue first until the value of $\frac{LOCAL\_CTR}{LMOD}$ become greater than $\frac{BF\_CTR}{BMOD}$. As a result, the throughput of S_2 is slightly higher at first.

3.5.2 Delay Performance of the BI_BWB Mechanism

In this section, we use simulation results to investigate the delay performance of the BI_BWB mechanism. We first consider a simple two station network in order to illustrate some of the advantages of the BI_BWB operation. Then, we look into the multistation case.

<u>A Two Station Network</u>

We consider a network with only two active stations S_1 and S_2 at a distance D_{12} = 30 slots. The BWB_MOD parameter for both stations is 2. We assume that S_1 is overloaded and tries to transmit segments on any free slot that passes by. S_2 is underloaded generating one packet every one round trip propagation delay. That is, S_2 generates a new massage at the instants 31, 91,151, 211 and so on, where the unit of time is the slot. Table 3.1 shows the packet delay at station S_2 when the packet size is 1 segment, 5 segments and 10 segments, respectively.

Table 3.1 A two station network. Delay comparison of S_2 under BI_BWB and NSW_IUT. S_1 is overloaded. S_2 generates 1 packet every one roundtrip propagation delay

Access Mechanism	Average Packet Delay at S_2 (microsec)		
	l_p of S_2 = 1 sgmt	l_p of S_2 = 5 sgmt	l_p of S_2 = 10 sgmt
BI_BWB	5.46	26.0	51.9
NSW_IUT	170.5	192.5	221.1

We see that the delays encountered by the underloaded downstream station S_2 are, under BI_BWB, much smaller than the corresponding delays under the NSW_IUT mechanism.This is because in the case of NSW_IUT, S_2 must first send its requests, wait for these requests to travel upstream to reserve idle slots from S_1, and then wait for the reserved idle slots to arrive. In contrast, in the case of BI_BWB, S_2 has immediate access to the channel.

We now consider the above network but with both stations underloaded. Each of S_1 and S_2 generates a new packet every 100 slots. S_1 at t = 0, 100, 200, 300, etc., and S_2 at t= 31, 131,231, 331, etc. The packet size of station S_1 is 60 segments. Table 3.2 shows the average packet delay at stations S_1 and S_2 when S_2's packet size is 1 segment, 5 segments and 10 segments, respectively. We see again the ability of BI_BWB to provide for the downstream station a much faster access to the channel. Furthermore, the much lower delay of S_2 comes only at a minor increase of the delay of S_1.

Table 3.2 Two station network. Delay comparison of BI_BWB and NSW_IUT. Both S_1 and S_2 are underloaded

Access Mechanism	Packet Delay (microsec)					
	l_p of S_2 = 1 sgmt		l_p of S_2 = 5 sgmt		l_p of S_2 = 10 sgmt	
	S_1	S_2	S_1	S_2	S_1	S_2
BI_BWB	166.5	2.73	177.5	35.5	191.1	79.2
NSW_IUT	163.8	163.4	163.8	174.7	163.8	188.4

Multiple Station Network

We consider an 155 Mbps network with a slot size of 53 bytes, a signal propagation delay of 5 $(\mu s)/(km)$, and 15 connected stations. The distance between neighbor stations is 2 time slots. All stations have the same traffic load. We call this type of load **constant load.** Furthermore, they have the same packet size (20 segments/packet), and the same value of BWB_MOD (= 2). The aggregate load generated by all stations on the forward bus is 0.85 segments/slots. In Fig.3.3, we show the station delays under the BI_BWB operation when different values of LMOD and BMOD are used. Furthermore, we compare these delays with the corresponding delays in the case of NSW_IUT.

Fig. 3.3 clearly shows the greater flexibility that the BI_BWB mechanism has in controlling the stations' delay performance. It also shows that the greater the value of LMOD/BMOD, the higher the upstream station delays. This is because a higher value of LMOD enables a station to store a greater number of upstream segments in its shift register and therefore increases the upstream stations' storage part of their access/storage delays. We point out that extensive simulation results, we have run, have shown that the delay behavior shown in Fig. 3.3 is not affected by the network size. It should also be noticed that when the value of LMOD is 0, the operation of BI_BWB becomes identical to that of the NSW_IUT mechanism. That is, NSW_IUT is a special case of BI_BWB.

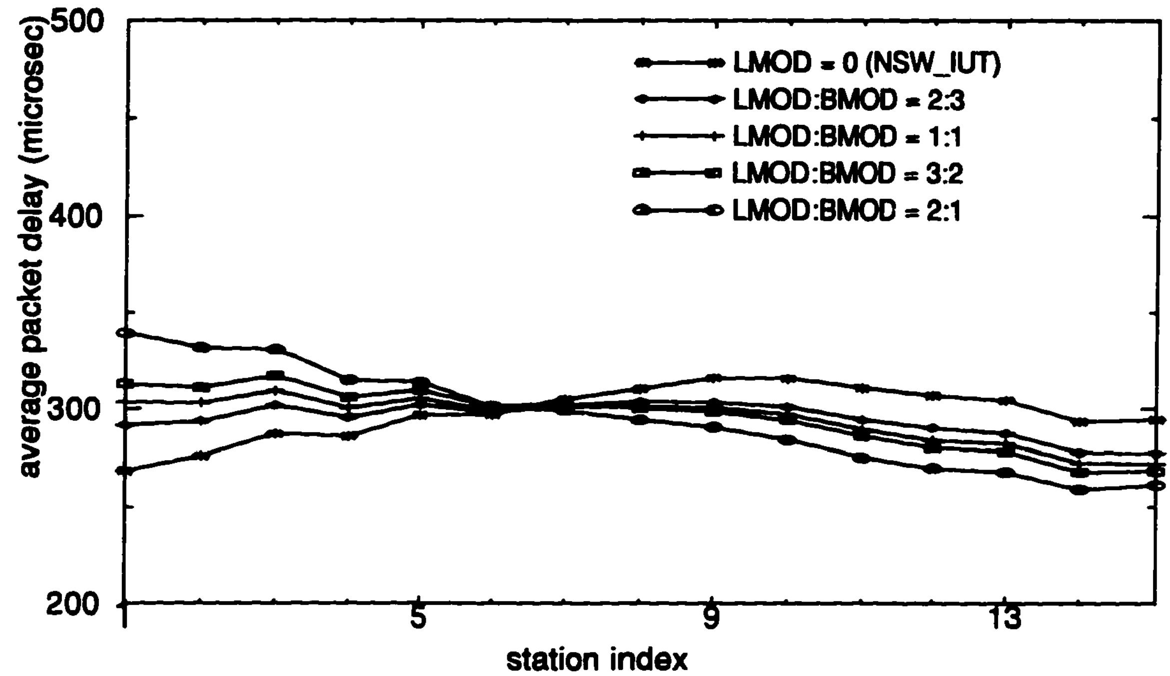

Fig.3.3 Packet delay comparison of BI_BWB and NSW_IUT. Total offered load = 0.85 sgmt/slot. Constant packet size of 20 segments. Constant load

3.6 Queueing Analysis of the BI_BWB Mechanism

In this section, we provide a queueing analysis for the BI_BWB mechanism that can derive accurate estimates for the average segment delay at each station. We use a similar queuing model to the one used for the analysis of NSW_IUT in [74]. That is, we use a two-state Markov chain to describe the request and busy slot arriving process at each station. The major difference between the two models lies on the busy slot arrival process that the segments of station S_i observe. In the case of NSW_IUT, if a busy slot arrives at station S_i, this station will not be able to write on this slot. In the case of BI_BWB, a station can insert the busy slot into the shift register and create an idle slot on which it may transmit a segment. In our queuing model, we try to include this buffer insertion behavior in the two state Markov process that describes the arrivals of the busy and empty slots at station S_i. Thus our corresponding two-state Markov chain does not really describe the busy slots that arrive at station S_i, but rather the slots that the local segments of this station see after the insertion buffer.

We consider a network of N stations, indexed from *1* to N, with distance between neighbor stations equal to d slots ($d \geq 1$). We consider the transmissions on the forward bus. We assume that the number of segment arrivals at each station follows the Poisson distribution. Each station is modelled as a multiqueue single server queueing system. The arrival process at each queue tries to encapture the interprocess dependencies among the various stations, the effect of the presence of the insertion buffer, the effect of the TAR bit, and the effect of the request transmission mechanism on the reverse bus.

3.6.1 Queuing Model for Each Station

Fig. 3.4 shows the three-queue single server model which is used to describe the behavior of each station. In this figure, the R-queue models the arriving requests on the reverse bus. The L-queue models the local traffic generated at the station. Finally, the B-queue models the bus slots that are coming out of the station's insertion buffer. We should notice that if this buffer is empty, then the B-queue describes the same sequence of busy slots which have been written by the upstream stations. Thus the arrival of a B-type customer is equivalent to the transmission of a segment from the head of the insertion buffer. The service discipline of each queue is First-In-First-Out (FIFO). Furthermore, the service time of all customers of all queues is deterministic and equal to the duration of one time slot.

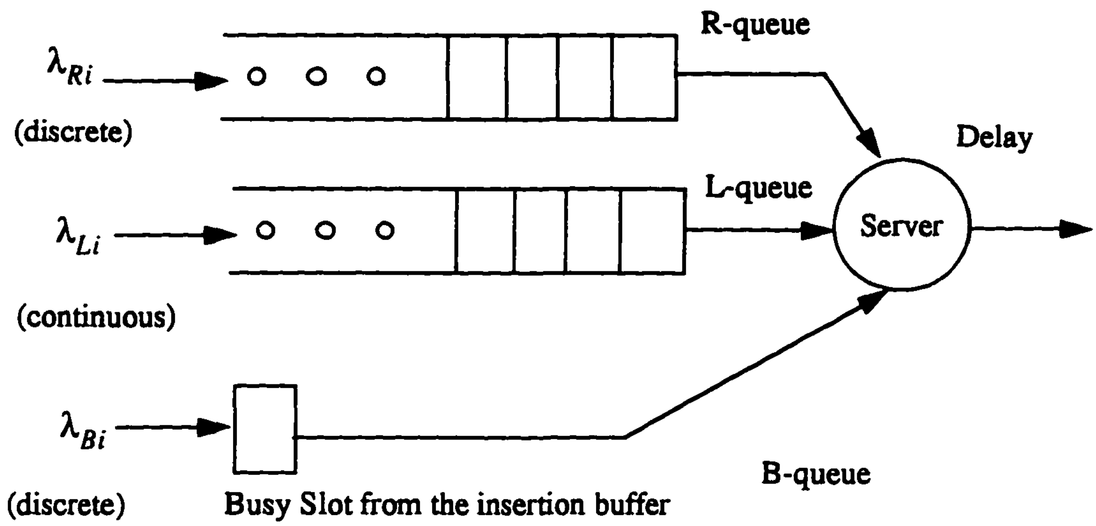

Fig.3.4 Queueing model for station S_i

In our model, the L-queue customers arrive according to Poisson distribution with mean λ_L . B and R queue customer arrivals are each modeled as a separate first order two-state Markov chain with mean arrival rate λ_B and λ_R , respectively. We have chosen this model because it can efficiently describe the variance of the arrival processes. Fig. 3.5 shows the Markov chain which is needed to model the arrivals of the B and R queue customers. State "1" indicates the arrival of a customer during a slot. State "0" indicates no customer arrival during a slot. The mean arrival rate, λ_B or λ_R, is equal to the steady state probability that the chain is at state "1". The parameter $\gamma = p_{11} - p_{01}$ describes the burstiness of the arrival process.

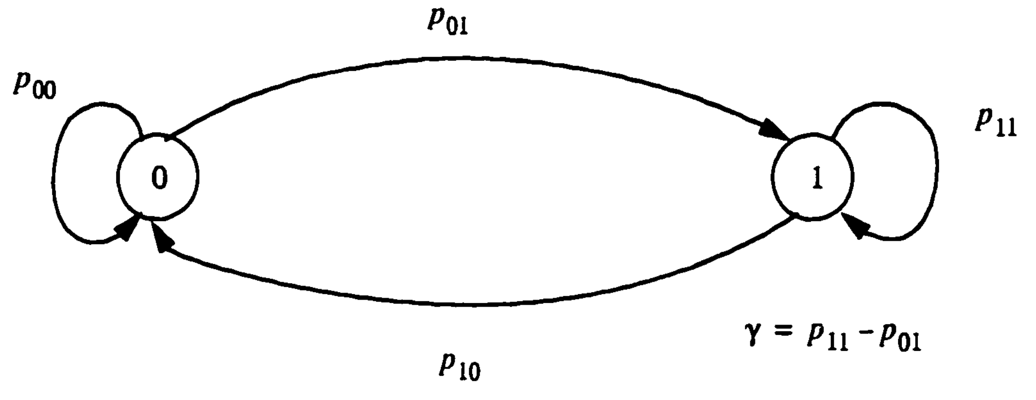

Fig.3.5 Markov chain modeling of R-queue and B-queue processes

Let γ_B, γ_R be the burstiness parameter for B and R queue customers, respectively. In the following sections, we will first derive estimates for the average access delay AD_i at each station S_i. Then, we will evaluate the average delay d_n that each segment of S_i spends inside the insertion buffer of a downstream station S_n. The total average access/storage delay ASD_i of S_i can then be computed by:

$$ASD_i = AD_i + \sum_{n=i+1}^{N} d_n \tag{3.5}$$

3.6.2 Segment Access Delay at Each Station

Let AD_X denote the average access delay that a customer experiences in an equivalent FIFO queuing system, in which the arrival process is the superposition of the arrival processes of customers that belong to set X. The access delays of B, R and L queue customers are then $AD_{Li}, AD_{Bi}, AD_{Ri}$, respectively. Because our system is work conserving, we have:

$$\frac{1}{\lambda_{Li}+\lambda_{Bi}+\lambda_{Ri}}(\lambda_{Li}\bullet AD_{Li}+\lambda_{Bi}\bullet AD_{Bi}+\lambda_{Ri}\bullet AD_{Ri}) = AD_{\{Si,Bi,Ri\}} \tag{3.6}$$

According to the BI_BWB operation, busy slots that are coming out of the insertion buffer will always be served first. That is, B-queue customers have absolute priority over L and R queue customers, and their delay is equal to the service time (i.e., $AD_{Bi} = 1$). If an idle slot comes out of the buffer, the BI_BWB will always let the idle slot to service a downstream request queue first before it services its local queue. That is, R-queue customers have absolute priority over the L-queue customers. Since the B and R queue customers have absolute priority over the L-queue customers, the presence of the L-queue customers does not affect their delay. Thus, the following equation holds:

$$\frac{\lambda_{Bi}}{\lambda_{Bi}+\lambda_{Ri}}\bullet AD_{Bi}+\frac{\lambda_{Ri}}{\lambda_{Bi}+\lambda_{Ri}}\bullet AD_{Ri} = AD_{\{Bi,Ri\}} \tag{3.7}$$

Equations (3.5) and (3.6) enable the computation of AD_{Ri} and AD_{Li}, provided we can derive estimates for λ_{Bi}, λ_{Ri}, $AD_{\{Bi,Ri\}}$ and $AD_{\{Bi,Ri,Li\}}$ (λ_{Li} and AD_{Bi} are known from the previous discussion). $AD_{\{Bi,Ri,Li\}}$ and $AD_{\{Bi,Ri\}}$ can be derived using the delay expressions which have been derived in [92]. In [92], a FIFO discipline queueing model in which the arrival process is the superposition of M arrival processes has been studied. Each of the arrival processes in that system has been modeled as a first order two-state Markov process with arrivals occurring at the slot boundaries. It has been shown in [92] that the average access delay AD_X experienced by the customers is given by:

$$AD_X = \frac{\sum_{n=1}^{M} \sum_{m>n}^{M} \left[1 + \frac{\gamma_n}{1-\gamma_n} + \frac{\gamma_m}{1-\gamma_m}\right] \lambda_n \lambda_m}{\sum_{n=1}^{N} \lambda_n \bullet \left(1 - \sum_{k=1}^{N} \lambda_k\right)} + 1 \tag{3.8}$$

In our queueing model (of Fig. 3.4), all L, B and R queue customers can start their service at the next slot. Since both requests and busy slots arrive at slot boundaries, we can use equation (3.7) to calculate $AD_{\{B,R\}}$. If the local segments arrive also at discrete time instant, we can also use equation (3.7) to calculate $AD_{\{B,R,L\}}$. However, local segments arrive according to the Poisson distribution. On the average, they have to wait for $1/2$ slot time to reach the boundary of next slot. As a result, we may use the following formula to compute $AD_{\{B,R,L\}}$:

$$AD_{\{B,R,L\}} = \frac{\lambda_{Li}}{2\left(\lambda_{Li} + \lambda_{Bi} + \lambda_{Ri}\right)} + AD'_{\{B,R,L\}} \tag{3.9}$$

where $AD'_{\{B,R,L\}}$ is the access delay computed by equation (3.7) assuming local segments arrive at the slot boundaries. It should also be noted that the burstiness parameter γ_L for the L-queue customers is 0. In the following sections, we define the arrival rates and the transition probabilities for each one of the two-state Markov chains that describe the arrival process of the B and R queue customers.

3.6.2.1 The R-queue Arrival Process: In this section, we first derive analytic estimates for the average arrival rate λ_{Ri} and then for the burstiness parameter γ_R. We use the following notations:

- t_i: probability that an incoming busy slot at station S_i carries a TAR = 1 bit.
- $\bar{t}_i$: probability that an incoming busy slot at station S_i carries a TAR = 0 bit.
- e_i: probability that a TAR = 1 bit (on a busy slot) is erased by station S_i.
- r_i: probability that station S_i inserts a request into a passing slot on the reverse bus.
- u_i: probability that an incoming slot is idle and unreserved slot, i.e., it can be written by S_i.
- p_{ik}: probability that S_i transmits its local segment in the k_{th} slot from the time the segment arrived at the station's L-queue without inserting a request for the transmitted segment.
- n_i: probability that S_i transmits a local TAR segment without inserting a request for it.

Mean Arrival Rate λ_R

The mean arrival rate of the R-queue at station S_i is the summation over all requests from the downstream stations of S_i, i.e., we have the following equation:

$$\lambda_{Ri} = \sum_{j=i+1}^{N} r_j \tag{3.10}$$

According to the BI_BWB operation, each station S_i inserts a TAR = 1 bit every M segments it transmits. Furthermore, S_i observes all the TAR = 1 bits sent by upstream stations and not erased by another upstream station. So we have that:

$$\bar{t}_i = \sum_{j=1}^{i-1} \lambda_{Lj} - t_i = \sum_{j=1}^{i-1} \lambda_{Lj} - \left(\frac{\sum_{j=1}^{i-1} \lambda_{Lj}}{M} - \sum_{j=1}^{i-1} e_j \right) \tag{3.11}$$

However, S_i may not insert a request for every arriving local segment. For instance, a request is not sent upstream when a TAR segment is transmitted while RG_CTR = 0, i.e., the TAR segment is transmitted into an unreserved slot. Therefore, we have:

$$r_i = \lambda_{Li} \left[\frac{M-1}{M} + \frac{1}{M}(1 - n_i) \right] \tag{3.12}$$

where

$$n_i = \sum_{k=1}^{\infty} p_{ik} \tag{3.13}$$

Simulation results have shown that in the case of independent segment transmissions, the probability of more than one segments waiting at the station's queue is very small. Therefore, in our analysis we may assume that a segment can send a request upstream after it has become first in the L-queue. Under this assumption, a segment can be transmitted in the k_{th} slot after its arrival without inserting a request on the reverse channel, if and only if it is a TAR segment, the k_{th} slot is an unreserved slot, and no unreserved or busy slot with TAR = 1 is seen during the preceding k-1 slots. The probability that no unreserved or busy slot with TAR = 1 is seen during the first k-1 slots is $\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1}$. The probability that the k_{th} slot is an unreserved slot is $u_i = 1 - \lambda_{Ri} - \sum_{j=1}^{i-1} \lambda_{Lj}$. Consequently,

$$p_{ik} = u_i \left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} = \left(1 - \lambda_{Ri} - \sum_{j=1}^{i-1} \lambda_{Lj}\right)\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} \tag{3.14}$$

By replacing the expression of n_i in equation (3.11) using equation (3.12) and (3.13), the following expression for r_i is derived:

$$r_i = \lambda_{Li}\left(\frac{M-1}{M} + \frac{1}{M}\left(1 - \frac{1 - \sum_{j=1}^{i-1} \lambda_{Lj} - \lambda_{Ri}}{1 - \bar{t}_i - \lambda_{Ri}}\right)\right) \tag{3.15}$$

Since we can use equation (3.10) to replace the $\bar{t}_i$ in the above equation, equation (3.14) is in fact a non-linear system of N-1 equations (i = 2 up to N) with r_i and e_i unknown. It is evident that $\lambda_{RN} = 0 = e_1$. Furthermore, each station will erase exactly one TAR = 1 bit for every extra request it inserts, and insert a regular request for each of its non TAR segments. Thus, the following equation can be written:

$$r_i = \frac{M-1}{M}\lambda_{Li} + e_i \tag{3.16}$$

Equation (3.15) provides the additional equations needed for the solution of the nonlinear system (3.14). We can then derive the mean arrival rate for station S_i by using equation (3.9).

Burstiness Parameter γ_R

In the remaining of this section, we will derive the bustiness parameter γ_R for the R-queue customers. We assume that each station S_i generates requests according a Poisson distribution with mean r_i. If S_i wants to insert a request on the reverse bus, it will set the first slot with empty Request Field (RF) it observes to 1. This means that S_i can insert a request only at the end of a train of RF = 1s from its downstream stations. Let L_i be a random variable that is equal to the length of a run of RF = 1 bits, followed by an RF = 0 bit, that are observed by station S_i. It has been shown in [83] that:

$$E[L_i] = \frac{1}{1-\lambda_{Ri}} \qquad (3.17)$$
$$E[L^2_i] = \frac{1}{(1-\lambda_{Ri})^3}$$

It is also evident from the two-state Markov chain of Fig.3.5 that:

$$P\{L_i = l\} = \begin{cases} p_{00} & l = 1 \\ p_{01}p_{11}^{\,l-2}p_{10} & l > 1 \end{cases}$$

$$P_{00} = 1 - P_{01} \qquad (3.18)$$
$$P_{10} = 1 - P_{11}$$

Using equations (3.17), we can express $E[l]$ and $E[l^2]$ in terms of P_{01} and P_{11}, and derive the following equations:

$$E[l] = 1 + \frac{P_{01}}{1 - P_{11}}$$
$$E[l^2] = 1 + \frac{3 - P_{11}}{(1 - P_{11})^2} P_{01} \tag{3.19}$$

Combining equations (3.16) and () , we can derive the following expressions for the transition probabilities of the R-queue's two-state Markov chain:

$$P_{01} = \frac{2\lambda_{Ri} - 2\lambda^2{}_{Ri}}{2 - \lambda_{Ri}}, \quad P_{11} = \frac{3\lambda_{Ri} - 2\lambda_{Ri}^2}{2 - \lambda_{Ri}}, \quad P_{00} = 1 - P_{01}, \quad P_{10} = 1 - P_{11} \tag{3.20}$$

3.6.2.2 The B-queue Arrival Process: In this section, we will derive the arrival rate λ_{Bi} and the bustiness parameter γ_B for the B-queue customers. This process is drastically affected by the network size, the presence of the request bits on the reverse channel, and the presence of the TAR = 1 bits on the forward channel. Because of all these complicated interdependencies among the different processes, it is extremely difficult to provide an accurate description of the arrival pattern of the BB = 1 bits that each segment at S_i sees after the insertion buffer. In our analysis, we concentrate on the busy slots going out of the insertion buffer when the station is active. We say that S_i is active if and only is at least one of its R or L queues is not empty.

Let Z be a random variable that describes the length of a run of busy slots, followed by an empty slot that comes out of the insertion buffer of S_i since the time instant an R or L type customer has arrived. For instance, if a local segment arrives at slot time T, and the first empty slot comes out of the insertion buffer at slot time T+z, then Z = z+1. In the following, we will first estimate $E[Z]$ and $E[Z^2]$. Then, we will describe a two-state Markov chain that can provide the same mean and variance of consecutive busy slots with the random variable Z. The mean arrival rate λ_{Bi} generated by this Markov chain is equal to the probability that a B-queue customer is in state 1. From Fig. 3.5 we can get:

$$\lambda_{Bi} = \frac{P_{01}}{P_{01} + P_{10}} \tag{3.21}$$

Calculation of $E[Z]$ and $E[Z^2]$

In order to calculate the $E[Z]$ and $E[Z^2]$, we classify the L and R -queue customers into three categories according to the time instants they insert their requests. Type-a customers are those who request a slot from upstream stations as soon as they arrive. Type-b customers are those who eventually insert a request before they receive service, but not at the time they arrive. Type-c customers are the rest of the customers that are transmitted without sending a request upstream. We should notice that all R-queue customers are also Type-a customers.

Let P_a, P_b and P_c be the probability that the next R or L queue customer, scheduled for transmission, is of type a, b, and c, respectively. Let also each of Z_a, Z_b and Z_c denote the conditional random variable which is equal to Z, given that the next R or L queue customer to be served is of type a, b, and c, respectively. It is evident that $E[Z]$ and $E[Z^2]$ are given by:

$$\begin{aligned} E[Z] &= P_a E[Z_a] + P_b E[Z_b] + P_c E[Z_c] \\ E[Z^2] &= P_a E[Z^2{}_a] + P_b E[Z^2{}_b] + P_c E[Z^2{}_c] \end{aligned} \tag{3.22}$$

We now compute estimates for P_a, P_b and P_c. The definition of type-a customers include all R-queue customers and L-queue customers for which a regular request is inserted. Hence P_a can be given by:

$$P_a = \frac{\frac{M-1}{M}\lambda_{Li} + \lambda_{Ri}}{\lambda_{Li} + \lambda_{Ri}} \tag{3.23}$$

All L-queue customers that insert an extra request because a TAR = 1 bit was seen on the forward bus during their waiting time are of type b. All the L-queue TAR segments that are transmitted before a TAR = 1 bit is seen on the forward bus are of type c. The probability n_i that an L-queue customer is transmitted without inserting a request is given by equation (3.12) Therefore, we have that:

$$P_b = (1-n_i)(1-P_a) = \left(1 - \frac{1 - \sum_{j=1}^{i-1} \lambda_{Lj} - \lambda_{Ri}}{1 - \bar{t}_i - \lambda_{Ri}}\right)(1-P_a) \tag{3.24}$$

$$P_c = n_i(1-P_a) = 1 - P_a - P_b$$

In the sequel, we compute estimates for $E[Z_a]$ and $E\left[Z_a^2\right]$. Recall that Z_a is the length of a sequence of busy slots that come out from the insertion buffer (followed by an idle slot) from the instant a type-a customer arrives at S_i. Z_a depends on whether the slot that comes out of the insertion buffer of S_i, at the time the type-a customer arrives, is idle or not.Let"$Z_{a|0}$)", " $Z_{a|1}$ " be the length of Z_a given that the slot is idle or busy, and P_{0a} be the probability that this slot is idle; we may assume that P_{0a} is equal to

$1-\sum_{j=1}^{i-1}\lambda_{Lj}$. Then, we can write the following equations for $E[Z_a]$ and $E[Z_a^2]$:

$$\begin{aligned} E[Z_a] &= P_{0a}E[Z_{a|0}] + (1-P_{0a})E[Z_{a|1}] \\ E[Z_a^2] &= P_{0a}E[Z_{a|0}^2] + (1-P_{0a})E[Z_{a|1}^2] \end{aligned} \tag{3.25}$$

In order to compute estimates for the first and second moments of $Z_{a|0}$, $Z_{a|1}$, we first model the two-state Markov process for the B-queue customers when there is no insertion buffer and no station inserts any request upstream. In this case, the order in which stations S_1 to S_{i-1} access the forward bus does not affect the distribution of busy slots seen by station S_i. Thus, we may assume that S_1 can access the bus first, S_2 can write on the idle slots that S_1 allows to pass, and so on. Each station generates busy slots according to the Poisson distribution with mean λ_{Li}. The mean arrival rate λ_{Ba} for station S_i's busy queue is summation of all the arrivals upstream of S_i, that is, $\lambda_{Ba} = \sum_{j=1}^{i-1}\lambda_{Lj}$. By following a similar procedure to the one which led to equations (3.19), we can derive the following transition probabilities for the sequence of busy slots which arrive at the station and are seen by type-a customers:

$$P_{01} = \frac{2\lambda_{Ba}-2\lambda_{Ba}^2}{2-\lambda_{Ba}}, P_{11} = \frac{3\lambda_{Ba}-2\lambda_{Ba}^2}{2-\lambda_{Ba}}, P_{00} = 1-P_{01}, P_{10} = 1-P_{11} \tag{3.26}$$

We now take into consideration both the requests inserted by S_i and its downstream stations as well as the presence of the insertion buffer. Assume that the sequence of busy slots that comes out of the insertion buffer of S_i evolves according to the two-state Markov chain of equation (3.24), and is interrupted by both the requests that are inserted by S_i through S_N and station S_i's local transmissions. Let X be the length of a run of busy slots followed by an idle slot that segments of S_i see after the insertion buffer from an arbitrary chosen time instant. X depends on whether the slot that comes out of the insertion buffer just before the commence of the observation period was idle, i.e., X depends on the current state of the Markov chain. Let $X_{|0}$, $X_{|1}$ be the length of X given that the preceding slot is idle or busy, respectively. Let also P_{empty}, P_{busy} be the probability that the next slot is empty, given that the current slot is empty or busy, respectively. If there is no insertion buffer, P_{empty} and P_{busy} can be approximated by:

$$P_{empty} = P_{00} + (1 - P_{00})\lambda_{Ri-1}$$
$$P_{busy} = P_{10} + (1 - P_{10})\lambda_{Ri-1} \tag{3.27}$$

With the insertion buffer, upstream busy slots carrying TAR = 1 bits may be stored and replaced with the station's local segments. Accordingly, from the station's point of view, a busy slot carrying a TAR = 1 bit that can be replaced with the station's own transmission is equivalent to an idle slot. The arrival rate of upstream busy slots carrying TAR = 1 bits that can be replaced by the station's own transmissions is $\frac{LMOD}{BMOD + LMOD} \cdot t_i$. Therefore, P_{empty} and P_{busy} should be modified as follows:

$$P_{empty} = P_{empty} + (1 - P_{empty})\left(\frac{LMOD}{BMOD + LMOD} \bullet t_i\right)$$
$$P_{busy} = P_{busy} + (1 - P_{busy})\left(\frac{LMOD}{BMOD + LMOD} \bullet t_i\right) \tag{3.28}$$

where P_{empty} and P_{busy} on the right hand sides of equation (3.27) are given by equations () . It is evident that

$$P\{X_{|0}=j\} = \begin{cases} P_{empty} & j=1 \\ P_{busy}P_{01}P_{11}^{j-2}(1-\lambda_{Ri-1})^{j-1} & j>1 \end{cases}$$

$$P\{X_{|1}=j\} = \begin{cases} P_{busy} & j=1 \\ P_{busy}P_{11}^{j-1}(1-\lambda_{Ri-1})^{j-1} & j>1 \end{cases} \tag{3.29}$$

We can now use the above equations to estimate the conditional first and second moments of Z_a, assuming that the stations are located D_d slots apart. According to the operation of BI_BWB, a type-a customer inserts its request as soon as it arrives at S_i. This request may affect the sequence of busy slots after the insertion buffer that S_i sees only after $2D_d$ slots. Let P_e, P_b be the probability that the first $2D_d$ slots seen by S_i are busy given that the preceding slot was idle or busy, respectively. These two probabilities can be given by:

$$P_e = 1-\sum_{j=1}^{2D_d} P\{X_{|0}=j\} \qquad P_b = 1-\sum_{j=1}^{2D_d} P\{X_{|1}=j\} \tag{3.30}$$

Given that the first $2D_d$ slots seen by S_i are busy, the arrival process of the busy slots at S_i after $2D_d$ slots have passed, is identical to the arrival process of the busy slots at S_{i-1}. Based on this fact, we can derive the following recursive expressions for the first and second moments of station S_i's $Z_{ai|0}$, $Z_{ai|1}$:

$$
\begin{aligned}
E[Z_{ai|0}] &= \sum_{j=1}^{2D_d} jP\{X_{|0} = j\} + \sum_{j=2D_d+1}^{\infty} jP_e P\{Z_{(ai-1)|1} = j - 2D_d\} \\
&= \sum_{j=1}^{2D_d} jP\{X_{|0} = j\} + P_e(E(Z_{(ai-1)|1}) + 2D_d) \\
E[Z_{ai|1}] &= \sum_{j=1}^{2D_d} jP\{X_{|1} = j\} + \sum_{j=2D_d+1}^{\infty} jP_b P\{Z_{(ai-1)|1} = j - 2D_d\} \\
&= \sum_{j=1}^{2D_d} jP\{X_{|1} = j\} + P_b(E(Z_{(ai-1)|1}) + 2D_d) \\
E\left[Z^2_{ai|0}\right] &= \sum_{j=1}^{2D_d} j^2 P\{X_{|0} = j\} + P_e\left(E\left(Z^2_{(ai-1)|1}\right) + 4D_d E(Z_{(ai-1)|1}) + 4{D_d}^2\right) \\
E\left[Z^2_{ai|1}\right] &= \sum_{j=1}^{2D_d} j^2 P\{X_{|1} = j\} + P_b\left(E\left(Z^2_{(ai-1)|1}\right) + 4D_d E(Z_{(ai-1)|1}) + 4{D_d}^2\right)
\end{aligned}
\tag{3.31}
$$

We now compute estimates for $E[Z_b]$ and $E\left[Z_b^2\right]$. Type-b customers are TAR segments for which requests are inserted before their transmission. Let Y be a random variable that describes the elapsed time from the instant a type-b segment arrives at the station until it sees a TAR = 1 bit and insert an extra request. Some of the type-b segments will be transmitted immediately after the arrival of a TAR = 1 segment while the others will have to wait. Let P_0 be the probability that a type-b customer is transmitted immediately after the arrival of a TAR = 1 bit. P_0 can be given by $\frac{LMOD}{BMOD + LMOD}$. For those segments that have to wait after their extra requests have been inserted, we may assume that they behave like type-a customers. Furthermore, we assume that the time interval from the instant a segment arrives until it sees a TAR = 1 bit, and the time interval from the instant an extra request is sent until the first idle slot is seen, are independent. Then, we may write the following expressions for $E[Z_b]$ and $E\left[Z_b^2\right]$:

$$E[Z_b] = (1-P_0)E[Z_{a|1}] + E[Y]$$

$$E\left[Z_b^2\right] = (1-P_0)^2\left(E\left[Z_{a|1}^2\right] - E[Z_{a|1}]^2\right) + E\left[Y^2\right] - E[Y]^2 + E\left[Z_b^2\right] \qquad (3.32)$$

According to the operation of BI_BWB, a TAR segment can transmit its extra request in the k_{th} slot if and only if the k_{th} slot carries a TAR = 1 bit from upstream, and no unreserved or busy slot with TAR = 1 is seen during the first *k-1* slots. This probability can be given by $t_i\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1}$. The probability that a segment of S_i is transmitted by inserting an extra request is $\sum_{k=1}^{\infty} t_i\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} = \frac{t_i}{1-\bar{t}_i-\lambda_{Ri}}$. Then the probability that the length of Y is equal to k would be:

$$P\{Y=k\} = \frac{1-\bar{t}_i-\lambda_{Ri}}{t_i} t_i\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} = \left(1-\bar{t}_i-\lambda_{Ri}\right)\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} \qquad (3.33)$$

From equation (3.32) , we can calculate $E[Y]$ and $E\left[Y^2\right]$. By replacing the expression of $E[Y]$ and $E\left[Y^2\right]$ in equations (3.31) , $E[Z_b]$ and $E\left[Z_b^2\right]$ can be calculated.

Finally, we compute estimates for $E[Z_c]$ and $E\left[Z_c^2\right]$. Type-c customers are the TAR segments for which no extra request has been sent on the reverse channel, i.e., an unreserved slot arrives before a TAR = 1 bit is observed. The probability that a slot after the insertion buffer is an unreserved slot is $\left(1-\lambda_{Ri}-\sum_{j=1}^{i-1}\lambda_{Lj}\right)$. The probability that no unreserved or busy slot with TAR = 1 is seen during the first k-1 slots is $\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1}$. Hence the probability that Z_c is equal to k slots can be given by:

$$P\{Z_c = k\} = \left(1 - \sum_{j=1}^{i-1} \lambda_{Lj} - \lambda_{Ri}\right)\left(\bar{t}_i + \lambda_{Ri}\right)^{k-1} \tag{3.34}$$

from which the calculation of $E[Z_c]$ and $E\left[Z_c^2\right]$ is straightforward.

Transition Probabilities of the Two-state Markov Chain

In this section, we will describe the two-state Markov chain for the B-queue customers. Let A be the time interval starting from a randomly selected moment until the first idle slot is generated. The length of A should have mean value equal to $E[Z]$ and second moment equal to $E\left[Z^2\right]$. Let also $P\{A_{|0} = k\}$, $P\{A_{|1} = k\}$ be the conditional probability that A = k, given that at the commence of this time interval the Markov chain was at state 0, 1. It is evident from the two-state Markov process of Fig.3.5 that:

$$P\{A_{|0} = k\} = \begin{cases} P_{00} & k = 1 \\ P_{01}P_{11}^{k-2}P_{10} & k > 1 \end{cases}$$

$$P\{A_{|1} = k\} = \begin{cases} P_{10} & k = 1 \\ P_{11}^{k-1}P_{10} & k > 1 \end{cases} \tag{3.35}$$

From equation (3.34), we can derive the following equations:

$$E[A_{|0}] = \frac{P_{10}+P_{01}}{P_{10}} \qquad E[A_{|1}] = \frac{1}{P_{10}}$$

$$E\left[A^2_{|0}\right] = \frac{P_{10}^2 + P_{10} + 2}{P_{10}^2}P_{01} + 1 - P_{01} \qquad E\left[A^2_{|1}\right] = \frac{2 - P_{10}}{P_{10}^2} \tag{3.36}$$

Let P_0, P_1 be the probabilities that the Markov chain is in state 0,1. It is evident that P_0, P_1 can be given by:

$$P_0 = \frac{P_{10}}{P_{01}+P_{10}} \qquad P_1 = \frac{P_{01}}{P_{01}+P_{10}} \tag{3.37}$$

Thus we have that:

$$\begin{aligned} E[Z] &= P_0 E[A_{|0}] + P_1 E[A_{|1}] \\ E[Z^2] &= P_0 E\left[A^2_{|0}\right] + P_1 E\left[A^2_{|1}\right] \end{aligned} \tag{3.38}$$

Equations (3.37) can provide the transition probabilities for the two-state Markov chain that describes the arrival process of the busy slots at station S_i. The burstiness parameter γ_B can also be calculated after the transition probabilities are computed. The mean arrival rate λ_{Bi} of the B-queue customers generated by this Markov chain should be equal to P_0. However, since in our analysis we consider only the busy slot arrival process at S_i when this station is active, the estimated arrival rate λ_{Bi} of B-queue customers which is given by equation (3.20) may be in some cases higher than $\sum_{j=1}^{i-1} \lambda_{Lj}$. In such cases, we force λ_{Bi} to be equal to $\sum_{j=1}^{i-1} \lambda_{Lj}$.

3.6.3 The Buffer Delay of Each Station

In this section, we derive estimates for the average time d_i that an upstream busy segment spends inside station S_i's insertion buffer. Let AD_{0i} be the access delay of station S_i when there is no buffering in the network. Since in this case no station will encounter any buffer delay, the access/storage delay of S_i is the access delay of S_i. Simulation results have shown that the sum of the access/storage delays experienced by all the stations in the system with buffering is almost identical to the sum of the access delays in the system without buffering. Thereupon, we have assumed them to be the same. In the case of buffering, each segment of station S_i experiences $AD_{0i} - AD_i$ less access delay than in the system without buffering. This difference can be considered as the sum of the buffer delays at S_i experienced by all the stations upstream from S_i. Simulation results have also shown that the segments of the different upstream stations of S_i encounter similar storage delays at S_i. Since there are (i-1) stations upstream from S_i, we may assume that the average storage delay d_i encountered by each one of them at station S_i will be:

$$d_i = \frac{AD_{0i} - AD_i}{i - 1} \tag{3.39}$$

We can now substitute (3.38) to (3.4) and derive an estimate for the average access/storage delay ASD_i of station S_i.

3.6.4 Model Accuracy

In this section, we investigate the accuracy of our queuing analysis under various offered loads, network sizes, and values of LMOD and BMOD. In all figures, we assume that BWB_MOD = 2. We compare the analytically derived delay estimates with the corresponding ones produced by simulations results.

In Fig. 3.6, we consider a 20 station, 155 Mbps dual bus network with end to end propagation delay 19 slots. The distance between neighbor stations is 1 slot. Each station transmits to any other stations with the same probability. In this way, the load generated by each station is proportionate to the number of its downstream stations. We have used the term **linear load** for this type of load. We compare simulation and analytically derived results for three different bus utilizations, i.e., 0.70, 0.80, 0.85. Fig. 3.6 clear shows that there is a very good agreement between analytic and simulation results.

In Fig. 3.7 we consider the same network and loading condition of Fig.3.6 but with a distance between neighbor stations equal to 2 slots. Finally, in Fig. 3.8, we consider the network of Fig. 3.6 but with each station generating the same amount of traffic; i.e., we consider a **constant load**. Both Fig. 3.7 and Fig. 3.8 demonstrate a good agreement between analytic and simulation results.

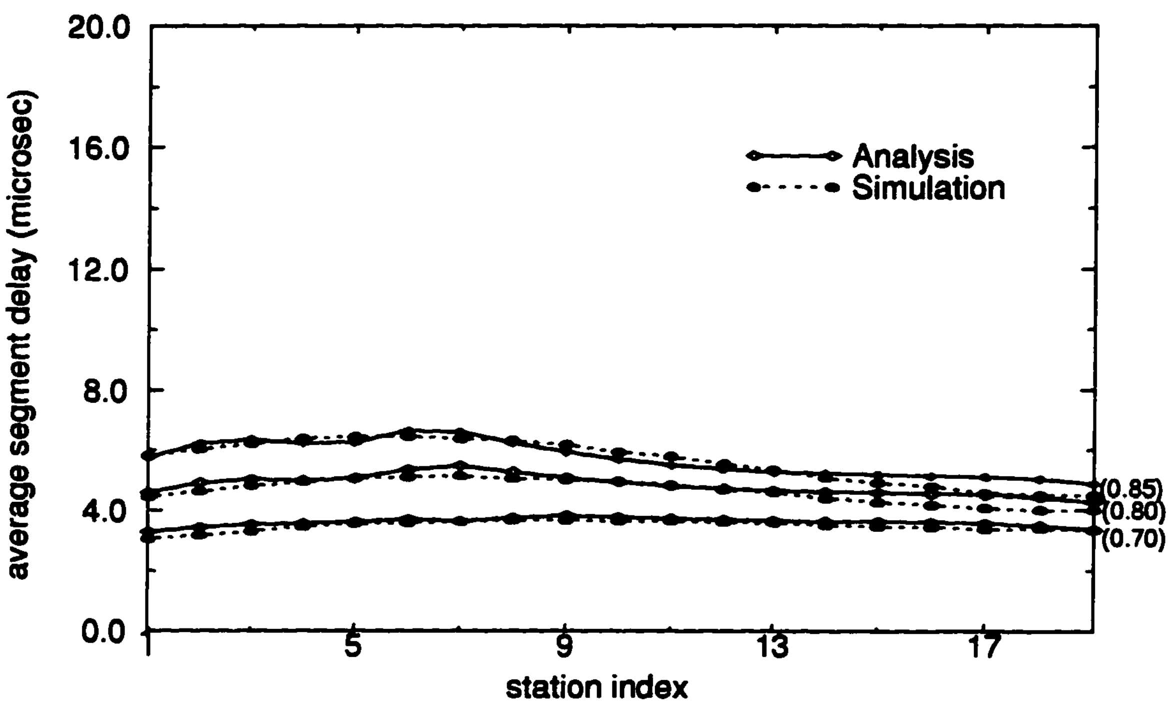

Fig.3.6 Comparison of analytic and simulation results. The distance between neighbor stations is 1 slot. LMOD:BMOD = 2:3. Linear load

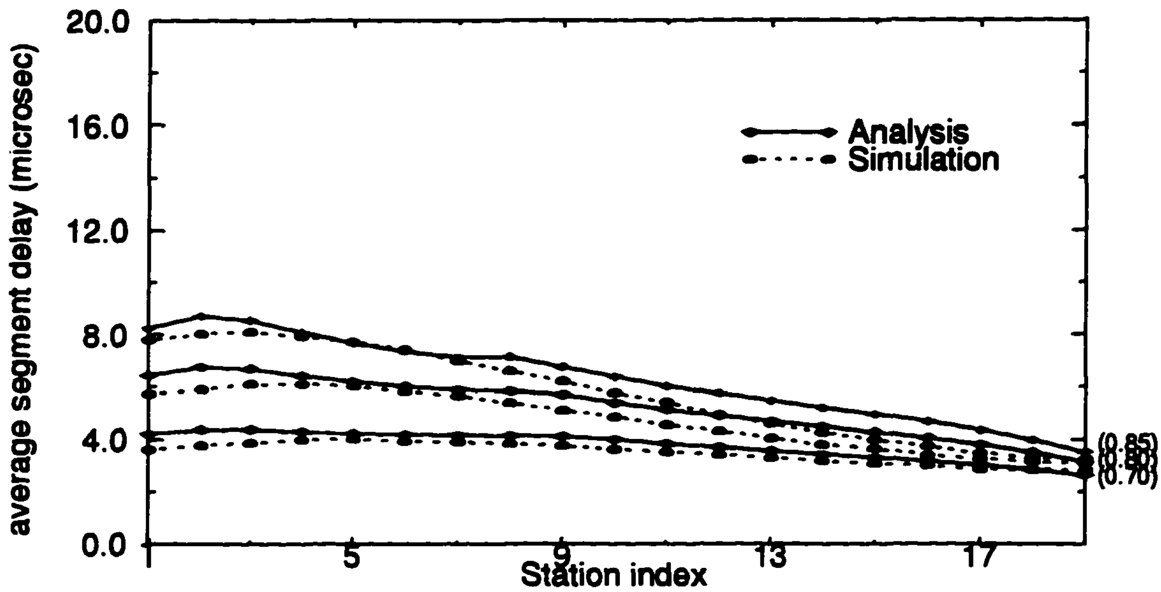

Fig.3.7 Comparison of analytic and simulation results. The distance between neighbor stations is 2 slots. LMOD:BMOD = 3:2. Linear load

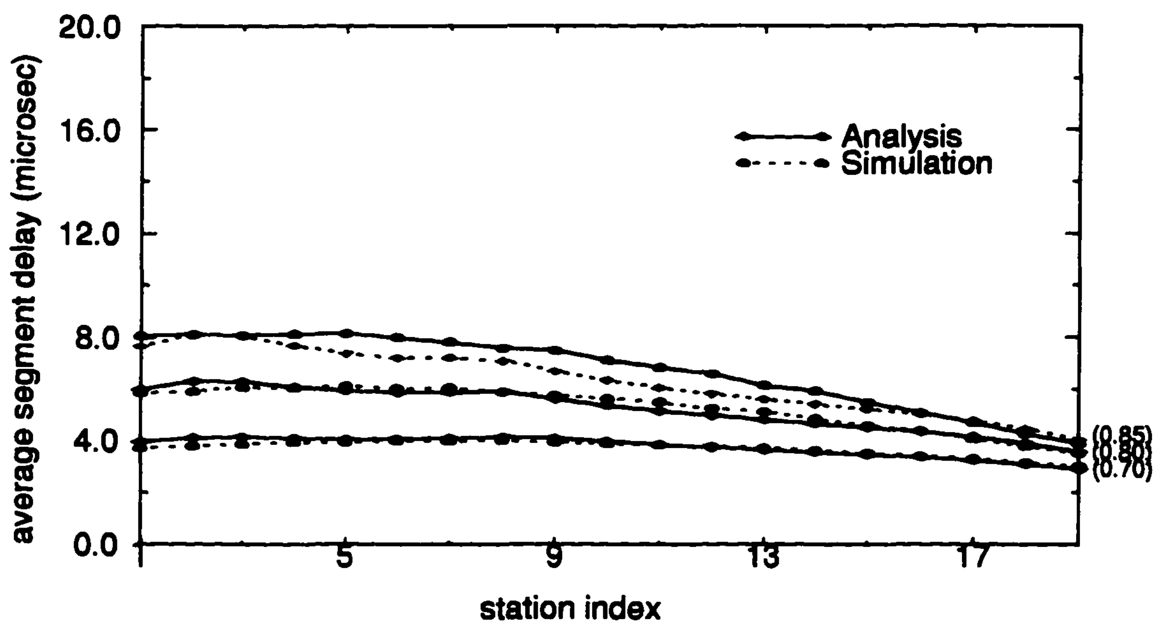

Fig.3.8 Comparison of analytic and simulation results. The distance between neighbor stations is 2 slots. LMOD:BMOD = 1:1. Constant load

3.7 Conclusions

In this chapter, the Buffer Insertion Bandwidth Balancing (BI_BWB) mechanism for dual bus networks has been introduced. BI_BWB combines the advantages of both the BWB_DQDB and NSW_IUT mechanisms by allowing downstream stations to have immediate access to the channel without requiring the wastage of channel slots. Furthermore, it can exercise greater control over the delays encountered by the various stations by selecting appropriate values for the LMOD, BMOD, and BWB_MOD parameters. We have investigated the throughput and delay performance of BI_BWB and we have compared it with the corresponding performance of NSW_IUT. Our investigation has shown that BI_BWB can converge faster to the steady state as well as provide a more fair delay performance. Finally, we have introduced a queuing analytic model for BI_BWB which can capture the interactions between busy slots, requests, and buffer insertion operation and provide accurate estimates for the stations' segment delay.

CHAPTER 4

THE PREEMPTIVE PRIORITY BI_BWB MECHANISM

4.1 Introduction

In this section, we investigate the ability of BI_BWB to provide effective priorities. We assume that each station in the network can support different priority classes of traffic with each class having its own queue. It should be pointed out here that devising effective and bandwidth efficient priority mechanisms for MANs is an extremely difficult problem. This is due to the large distances involved which significantly increase the propagation delay of the feed-back information, and thus drastically affect the responsiveness of the network to the changes of the traffic load. Consequently, most of the priority mechanisms, which have been proposed in the literature are either not effective, i.e., the performance of the high priority classes may be strongly affected by the location and traffic characteristics of the low priority classes, or their operation may require the wastage of significantly amount of channel bandwidth.

The most effective and bandwidth efficient priority mechanism that has been introduced for the dual bus network topologies is P_NSW_IUT [74]. Its operation does not waste any channel bandwidth and can preempt the low priority traffic transmissions. However, still, high priority traffic in downstream stations may have to wait for an initial delay, which is equal to the round-trip propagation delay, before it can access the medium. The presence of the insertion buffer in the case of BI_BWB can eliminate this problem. It enables a station to store the low priority segments on the channel until the transmissions of the high priority traffic is complete. The organization of the rest of this chapter is as follows. In section 2, we present the Preemptive Priority BI_BWB mechanism (P_BI_BWB).

In section 3, we investigate its throughput and delay performance. Finally, in section 4, we present out conclusions.

4.2 The Preemptive Priority BI_BWB Mechanism

The preemptive Priority BI_BWB (P_BI_BWB) mechanism has two main objectives: The first is to guarantee fair access among users of same priority. The second is to eliminate the effect of low priority traffic on high priority traffic. According to this mechanism, the highest priority traffic class can acquire all the channel bandwidth. A lower priority traffic class can receive some bandwidth only after all higher priority classes have satisfied their bandwidth requirements.

P_BI_BWB considers each station to consist of separate substations with one traffic class per substation. The ordering of these substations is always from the highest priority class to the lowest priority class. This means that if a class P_i has higher priority over class P_j, then class P_i will always see the slots on the forward bus before class P_j. In Fig.4.1, we show the basic components of a station in the case of three priority classes.We see that each priority class (substation) P_i has its own transmission queue and its own set of counters such as the Request Counter (RQ_CTR[i]), Bandwidth Balancing Counter (BWB_CTR[i]), etc. The functions of these counters are similar to those of the corresponding counters in the case of BI_BWB with the exception of RQ_CTR[i], which counts not only the requests of priority P_i, but also the requests of higher priority. Fig. 4.1 also shows that each of the high and medium priority classes has, in addition to its Insertion Buffer, a Lower Priority Segment (LPS) Buffer in which it can store the lower priority segments of the forward bus when it replaces them with its own transmissions. We point out that the main difference between the LPS buffer and the Insertion buffer is the following. In the LPS buffer the substation stores segments of lower priority, while in the Insertion Buffer it stores segments of the same priority. Finally, Fig. 4.1 shows that each priority class has been provided with a request queue REQ_QS[i] which describes the number of priority P_i requests that priority class P_i must send upstream. It should be noted here that each class P_i can send requests of the same or lower priority only.

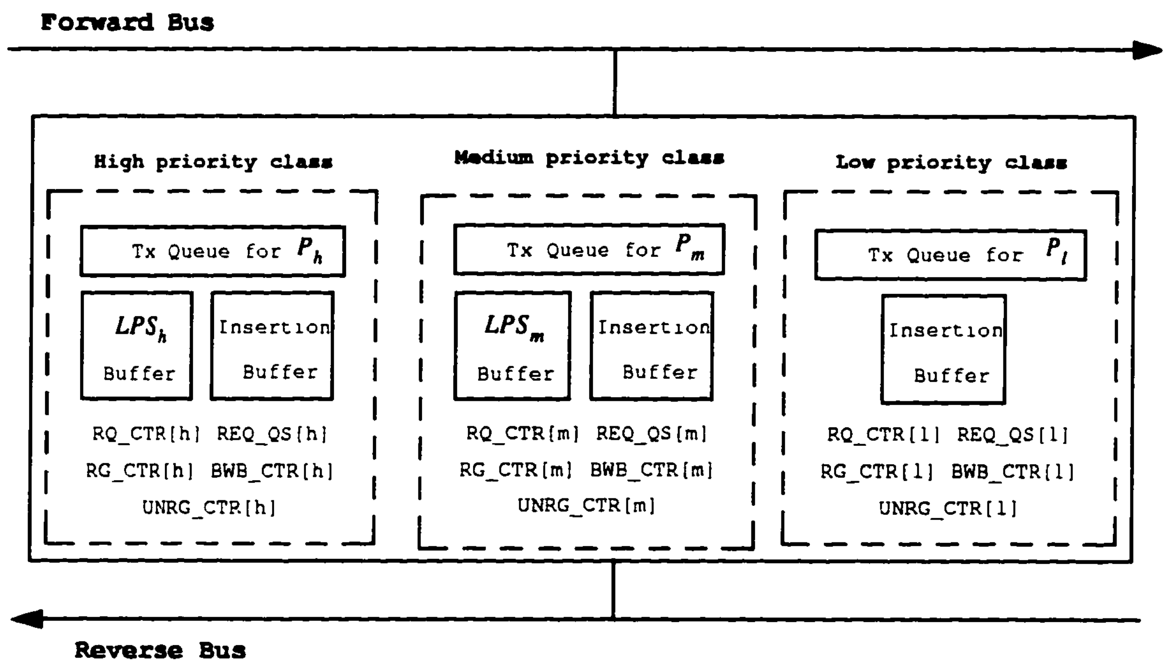

Fig.4.1 Basic station structure under the P_BI_BWB mechanism

In order to implement the proposed P_BI_BWB mechanism, the Access Control Field (ACF) of the slot must carry a separate Busy Bit and a separate Request Bit for every priority class, as well as one TAR bit. Then, in the case of three priority classes, our priority mechanism requires 7 control bits. However, the DQDB slot (see Fig. 2.1) only has six control bits available, i.e., the three Request Field (RF) bits, the one Busy Bit, and the two

Reserved bits. We can still implement our priority mechanism using only six bits in the following way. We can maintain the three RF bits of the DQDB slot, use one bit for the TAR bit, and use the remaining 2 bits to describe the priority of the slot. Then, a 00 will indicate an idle slot, a 01 will indicate a low priority busy slot, an 10 will indicate a medium priority busy slot, and an 11 will indicate a high priority busy slot.

The operation of every substation in the case of P_BI_BWB is very similar to the operation of a station under BI_BWB, i.e., in the case of a single priority traffic. Therefore, in the following, we focus on the differences between these two operations. These mainly reside in the transmission of requests and the operation of the LPS buffer.

In the case of P_BI_BWB, when a segment of priority P_i becomes first in the substation's local queue, it will send a request of the same priority upstream (i.e., REQ_QS[i] will increase by 1). In addition, whenever it transmits an unregistered priority P_i segment, it will also send a priority P_{i-1} request upstream. This is to ensure that enough free slots will be reserved that will guarantee the transmission of the buffered low priority segments.

The second difference between the operation of a substation P_i under P_BI_BWB, and the operation of a station under BI_BWB, is the following. Whenever a substation P_i sees a busy slot of priority P_j which is lower than P_i, and has knowledge of the presence of a priority P_i segment in the network (i.e., P_i's local transmission queue or insertion buffer are not empty, or RQ_CTR[i]>0), the substation P_i will buffer the busy slot into its LPS_i buffer and create an idle slot.

The last difference between the operations of the two mechanisms is that under P_BI_BWB, a substation P_i's request counter RQ_CTR[i] counts not only the requests of priority P_i (as in the case of BI_BWB), but also the requests of priority higher than P_i.

We now summarize the operation of a class (substation) P_i in the case of P_BI_BWB by describing its reaction to various events.

a) **New packet arrival:** class P_i behaves as in section 3.3.

b) **A segment becomes first in the transmission queue:** class P_i behaves as in section 3.3.

c) **A slot arrives at the forward bus:** if this is a busy slot of lower priority P_j, the following step "c4" must be introduced in the algorithm of 3.3. Otherwise, class P_i will behave as in section 3.3.

> **c4**. If either RQ_CTR[i] > 0 or any of P_i's local transmission queue or insertion buffer is not empty, the substation will buffer the slot into the end of its LPS_i buffer. Otherwise, i.e., RQ_CTR[i] = 0 and both local queue and insertion buffer are empty, then if class P_i' LPS buffer is not empty, P_i will buffer the slot segment at the end of its LPS_i and transmit the first segment from the LPS_i buffer.

d) **A slot is seen on the reverse bus:** step "**d1**" should be modified as follows.

> **d1**. If the slot carries a request of priority P_i or higher, class P_i will increase its REQ_QS[i] by 1.

e) **The station transmits a segment from its local queue:** the following step should be added.

> **e0**. If the transmitted segment is an unregistered segment, class P_i will send a request of priority P_{i-1} upstream.

f) **The station transmits a segment from its insertion buffer:** class P_i behaves as in section 3.3.

4.3 P_BI_BWB Performance Analysis

In this section, we investigate the effectiveness of P_BI_BWB and show that higher priority substations requires finite size LPS buffers in order to remove the lower priority class segments. It is evident from our previous discussion that the operation of P_BI_BWB applies BI_BWB to all traffic sources of the same priority. Therefore, it can evenly distribute the available bandwidth among them. In addition, it enables higher priority classes to remove the lower priority busy slots from the channel. In this way, each class P_i operates as if it were the only priority class present in the system with channel bandwidth being the remaining bandwidth after all higher priority classes have satisfied their bandwidth requirements.

It should be pointed out that the P_BI_BWB operation ensures the transmission of a sufficient number of requests that will ensure the return of the buffered segments to the channel. This is achieved by allowing a priority P_i substation to store a low priority busy slot only when at least one of the following three conditions holds, i.e., when a) P_i's RQ_CTR[i] is greater than 0, b) P_i's insertion buffer is not empty, or c) P_i' local transmission queue is not empty. If "a)" is true, a priority P_i request has already been sent upstream. Therefore, an idle slot will be reserved for the low priority buffered segment. If "b)" is true, then it is also guaranteed that an idle slot will be reserved for the lower priority buffered segment. This follows from the corresponding discussion on BI_BWB in section 3.2. Finally, if "c)" is true, then the new step "e0" of the previous algorithm, which describes the substation operation, guarantees that a request will be sent for the buffered low priority segment regardless of whether the transmitted local segments is registered or not. Since a request is always sent for a buffered segment, the maximum size of the LPS_i

will be equal to the number of slots in one round-trip propagation delay between substation P_i and the most upstream active lower priority substation.

4.3.1 Transient Analysis of P_BI_BWB Mechanism

In this section, we use simulation results to investigate the effectiveness of P_BI_BWB under transient traffic conditions. We consider a high-capacity network of 155 Mbps, a slot size of 53 bytes, and a signal propagation delay of 5 $(\mu s)/(km)$. In Fig. 4.2, we show the convergence speed of P_BI_BWB when there are three stations S_1, S_2, S_3 present on the bus an at a distance of 20 slots (i.e., $D_{12} = 20 = D_{23}$), corresponding to a cable length of 10.9 km. The horizontal axis represents time, measured in slots, with ticks appearing in multiples of the end-to-end propagation delay t_{prop}. Initially, only station S_1 is active and overloaded with low priority traffic and it acquires all channel bandwidth. At time $t = 5 \bullet t_{prop} = 200$, station S_2 becomes active and overloaded with high priority traffic. Finally at time $t = 10 \bullet t_{prop} = 400$, the high priority class of station S_3 becomes active and tries to acquire all the bandwidth. Fig. 4.2 shows that the high priority traffic source at S_2 can acquire all the channel bandwidth as soon as it becomes active. That is, the presence of the upstream active low priority class at S_1 does not affect the transmissions of the downstream high priority class at S_2. It should be noted here that we measure the throughput of each class every one end-to-end propagation delay, i.e., 40 slots. Thus, the throughput of S_2 during the interval (200, 240) is the value of its normalized throughput characteristic curve, shown in Fig. 4.2 at t = 240. Fig. 4.2 also shows that during the same interval (200, 240), station S_1 has a throughput of 0.5 segment/slot. This is because S_1 has the opportunity to write on 20 time slots after the instant S_2 becomes active; it takes 20 time slots for the requests which are transmitted by S_2 to arrive at S_1 and prevent S_1 from transmitting. These slots will be buffered at S_2 and will not affect S_2's throughput. Finally, Fig. 4.2 shows that one round trip propagation delay after the instant the high priority station S_3 becomes active, S_2 and S_3 evenly share the channel bandwidth.

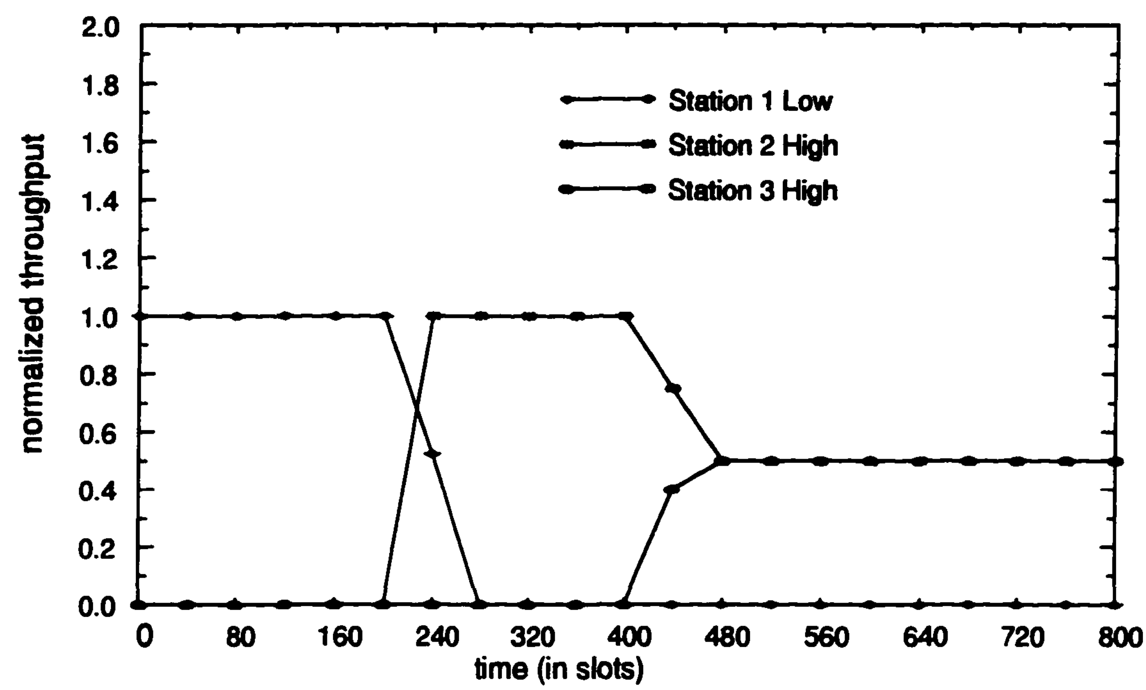

Fig.4.2 Throughput comparison of P_BI_BWB. $D_{12} = D_{23} = 20$ slots. Station S_1 is overloaded with low priority segments and station S_2 and S_3 are overloaded with high priority segments

4.3.2 Delay Performance

We first consider a network with only two active stations S_1 and S_2 at a distance equal to 30 time slots ($D_{12} = 30$). The BWB_MODs parameter of each station is 2. S_1 is overloaded with low priority segments and tries to write on every idle slot. S_2 is underloaded and generates a high priority packet every one round trip propagation delay, i.e., every 60 slots. Specifically, S_2 generates high priority packets at t = 31, 91,151, 211, etc., where the time is measured in time slots. Tables 4.1 shows the average packet delay at station S_2 when the packet size l_p is 1 segment, 5 segments, 10 segments, and 20 segments, respectively. Tables 4.1 clearly shows that P_BI_BWB is a more effective priority mechanism.

Table 4.1 Delay comparison of P_BI_BWB and P_NSW_IUT. S_1 is overloaded with low priority traffic. S_2 generates 1 high priority packet every one roundtrip propagation delay

Access Mechanism	Average Packet Delay at S_2 (μsec)			
	l_p = 1 sgmt	l_p = 5 sgmt	l_p = 10 sgmt	l_p = 20 sgmt
P_BI_BWB	4.87	15.8	29.4	56.7
P_NSW_IUT	169.3	180.2	193.8	221.1

In the following, we use simulation results to investigate the delay behavior of P_BI_BWB in the presence of many active stations. We consider an 155 Mbps network with 15 active stations and interstation distance of 2 slots. The slot size is again 53 bytes and the signal propagation delay is 5 $(\mu s)/(km)$. Each station supports high and low priority traffic both of which have BWB_MOD = 2. The aggregate generated load by the high priority is 0.85 segment/slot. Each high priority source generates the same amount of traffic (i.e., we have constant load) transmitting fixed size packets of 20 segments. In Fig. 4.3, we compare the average packet delay performance of the high priority class under P_BI_BWB and P_NSW_IUT in two cases, when the low priority is idle and when it is overloaded. Fig. 4.3 clearly shows that P_BI_BWB is a very effective priority mechanism, since the low priority class, even when it becomes overloaded, does not have any effect on the high priority delay. In contrast, in the case of P_NSW_IUT, the effect of the low priority class is evident.

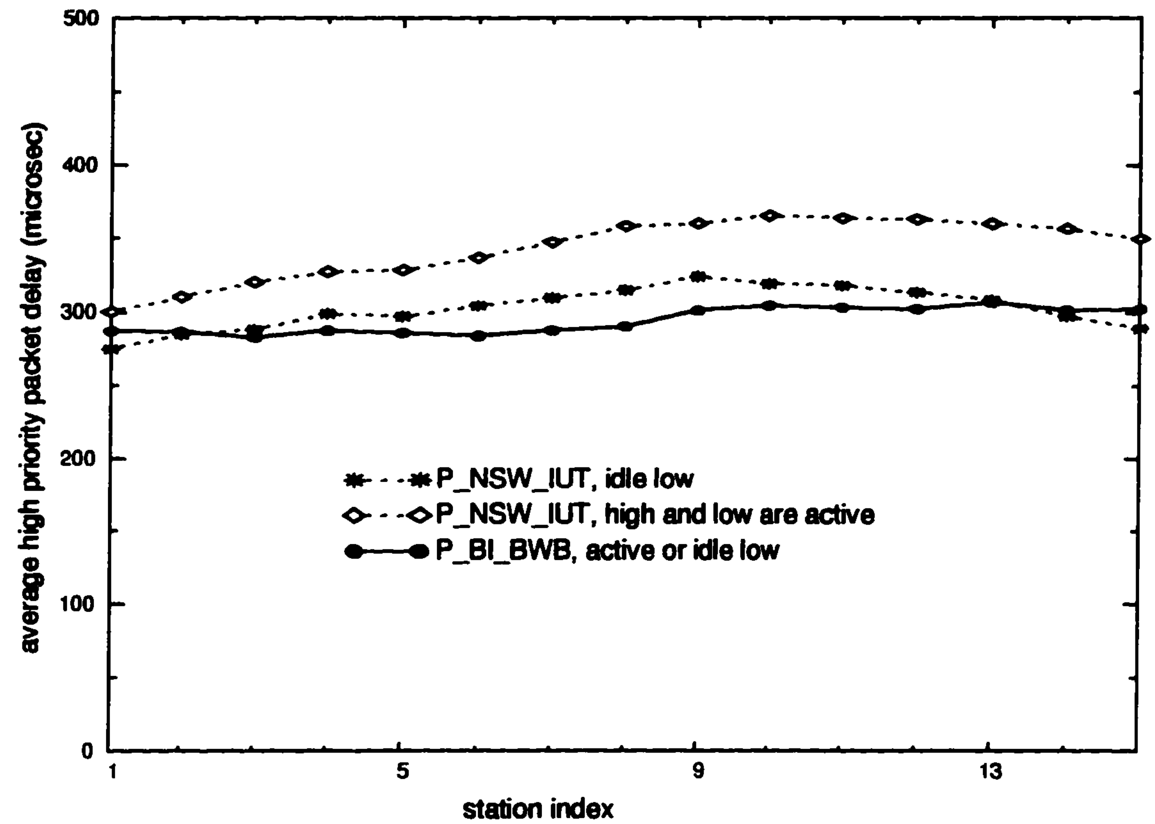

Fig.4.3 High priority delay comparison of P_BI_BWB and P_NSW_IUT. 15 stations, high priority underloaded with total offered load = 0.85 sgmt/slot. Constant load. BWB_MOD = 2, LMOD:BMOD = 3:2. Distance between stations 2 slots. Low priority can be idle or overloaded. Constant packet size of 20 segments

In Fig. 4.4, we compare the performances of P_BI_BWB and P_NSW_IUT when high and low priority classes are underloaded. Both of them generate fixed size packets of 20 segments. The aggregate load generated by the high priority class is 0.3 segment/slot and by the low priority class is 0.55 segment/slot. Fig.4.4 clearly shows that P_BI_BWB and P_NSW_IUT are very effective priority mechanisms. Under both of them, the high priority class encounters significantly lower delays. However, in the case of P_NSW_IUT, the location of a station seems to have a rather strong effect on the average high priority packet delay.

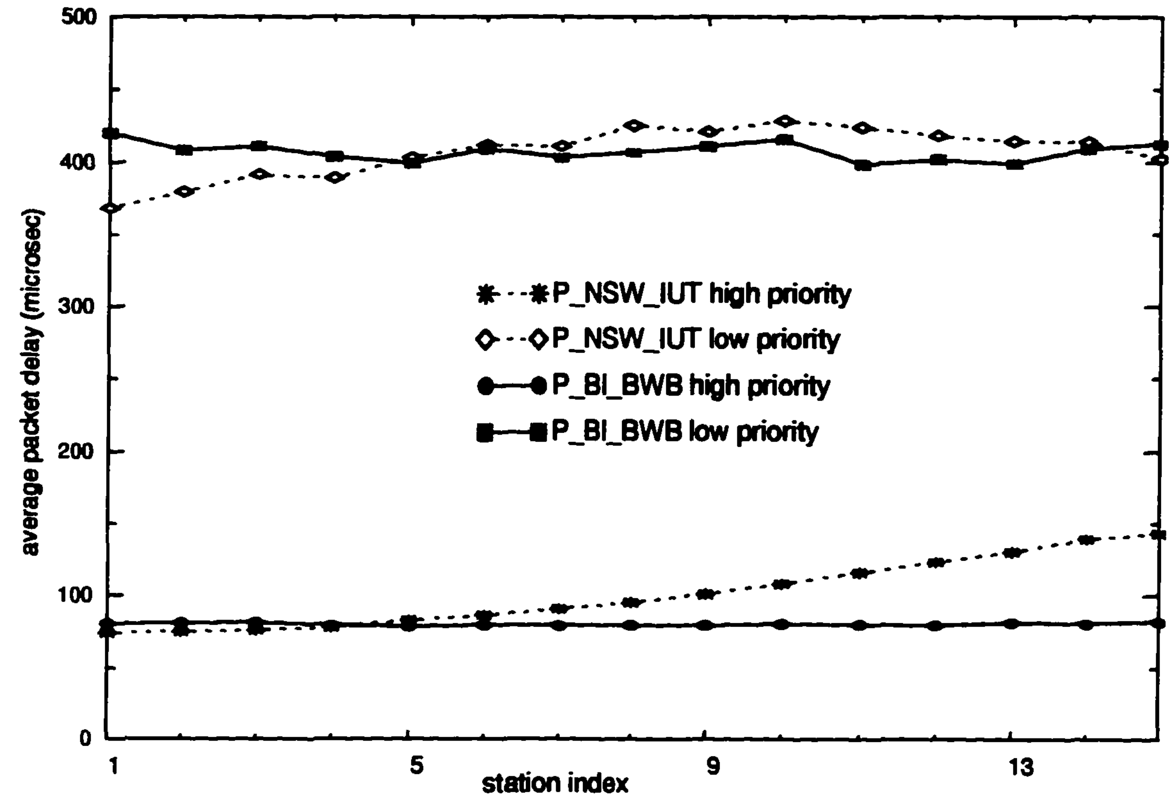

Fig.4.4 Delay comparison of P_BI_BWB and P_NSW_IUT. 15 stations, high priority load = 0.3 sgmt/slot, low priority load 0.55 sgmt/slot. Distance between stations 2 slots. BWB_MOD=2, LMOD:BMOD=3:2

4.4 Conclusions

In this chapter, we have introduced the effective Priority Buffer Insertion Bandwidth Balance (P_BI_BWB) mechanism for dual bus network architectures. This mechanism has the ability to eliminate the effect of low priority traffic on high priority traffic by enabling downstream high priority stations to temporarily remove the low priority busy slots from the channel until the transmission of the high priority traffic is complete. P_BI_BWB achieves that by allowing stations to store the low priority busy slots of the channel into a buffer, and by making sure that enough slots are reserved to guarantee the retransmission of the stored slots later on. Our simulation results have shown that P_BI_BWB is a very effective priority mechanism, which demonstrates a much better delay performance than the existing P_NSW_IUT mechanism. It does not only eliminate the effect of the low priority on high priority, but also minimize the effect of the station location on performance. Furthermore, it can evenly distribute the channel's bandwidth among overloaded traffic sources of the same priority.

CHAPTER 5

THE FAIR BANDWITH ALLOCATION MECHANISM FOR DUAL RING ARCHITECTURES

5.1 Introduction

Ring architectures offer superior reliability, availability, and serviceability, even in the face of physical damage to the network. Failed stations or transmission links can be isolated through the use of appropriate protocols such as the reliability mechanisms introduced in [46]. Ring architectures inherently impose no restriction on the number of stations that can be accommodated by the network. Furthermore, they offer significant performance advantages. These include insensitivity to load distribution, low arbitration time, bounded access delay, and no requirement for long preambles. Concurrent access and spatial reuse enable simultaneous transmission over disjoint sections of the ring and can drastically increase the throughput of the network. Counter rotating dual ring topologies allow stations to select the shortest transmission path, thus improving even more the system performance. Hence, they appear to satisfy best the requirements of high speed networking when high connectivity and large extents are required.

In this chapter, we introduce a new medium access control mechanism for dual ring architectures employing spatial reuse: the Fair Bandwidth Allocation Mechanism (FBAM). Its main feature is that it allows each active station to continuously send requests for bandwidth reservation. This is in contrast to the recently introduced Local Fairness Algorithm (LFA) [16] that switches to the bandwidth reservation mode only when bandwidth starvation is observed. As a result FBAM is more responsive to changes of the traffic load. Furthermore, it does not encounter the problem of determining when starvation is observed. The organization of the rest of the chapter is as follows. In section 5.2, we introduce the main features of the FBAM mechanism. In section 5.3, we provide a detailed description of its operation. In section 5.4, we use simulation results to investigate FBAM's throughput performance. Furthermore, we compare its performance with the corresponding performance of the Global and Local fairness algorithms which have been introduced in [18] and [16], respectively. Finally, in section 5.5, we present the conclusions.

5.2 Main Features of the FBAM Operation

The FBAM operation borrows features from the NSW_IUT [48] operation. That is, it also allows each station to send multiple requests upstream and uses the TAR control bit to introduce bandwidth balancing without requiring the wastage of channel slots.

5.2.1 Request Removal

A very important issue on dual ring networks is the removal of requests. This is not an issue on dual bus networks because requests are discarded at the end of each bus. However, in the case of a ring architecture, requests may go around the ring forever, keep the RQ_CTRs of the stations always greater than 0, and prevent these stations from transmitting. We point out that the removal of requests is not as straightforward as one might think. For instance, a simple request removal mechanism could have been the following: a station could reset a passing request bit on the reverse channel, if it has just allowed an idle and unreserved slot (station's RQ_CTR = 0) to pass downstream on the forward channel. The reasoning behind such an approach is that the idle slot would serve the downstream station which has sent the request, and thus there is no need for this request to propagate upstream. Although such a technique may work well under light traffic conditions, it may lead to a deadlock situation. Under heavy traffic conditions, such a deadlock may start with requests which are sent by some stations and travel for more than one ring before their removal; this may occur because all stations are observing busy bits on the forward channel before the arrival of the requests on the reverse channel. These requests would increase the RQ_CTRs of the various stations and eventually a situation may arise where the same requests go around the reverse channel and are never removed because the RQ_CTRs of the stations are always greater than 0.

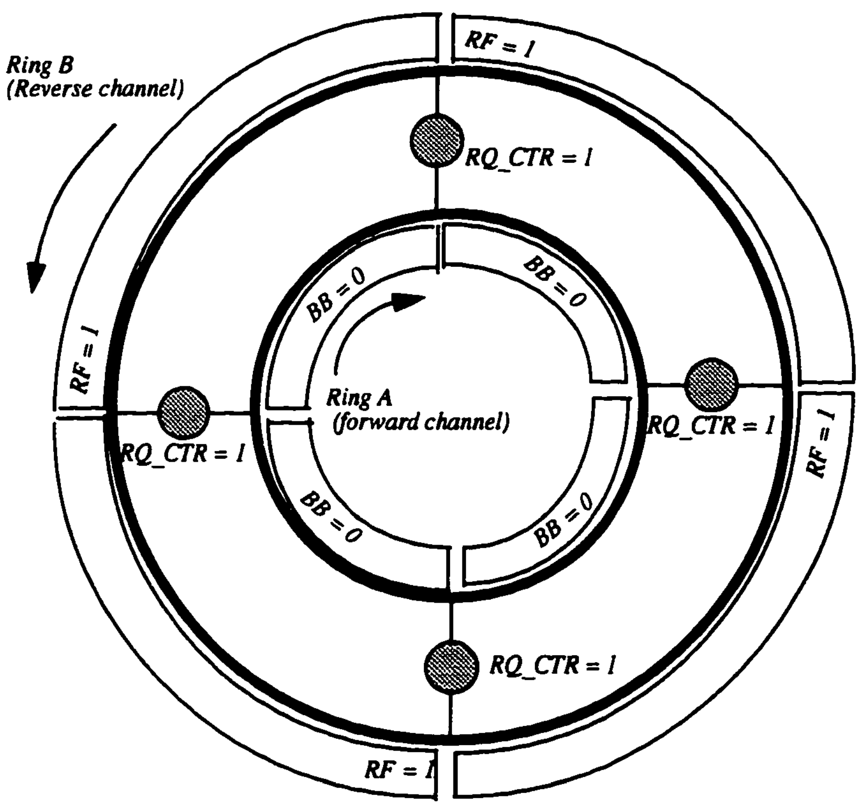

Fig.5.1 A deadlock situation

In order to illustrate such a situation, we refer to Fig.5.1 where a dual ring consisting of 4 active stations is shown. The distance between neighbor stations is 1 slot and all slots on the forward channel are idle; i.e. their Busy Bits (BB) are 0. However, the Request Fields (RF) on the reverse channel are all equal to 1. Furthermore, the RQ_CTR of each station is equal to 1 and, we assume, that each station looks first at the reverse channel and then at the forward channel. It is evident that under the above conditions: a) the RQ_CTR of each station will always change its value from 1 to 2 (when an RF = 1 bit is observed) and then from 2 to 1 (when the next BB = 0 is observed), b) the RF = 1 bits will never be reset to 0 (since RQ_CTR>0, i.e., no station sees an idle and unreserved slot), c) the throughput of the system will be 0.

The above discussion clearly indicates that the removal of requests should not depend on the traffic conditions. A mechanism is required which must be as robust in removing requests as is the end of the bus in the dual bus architectures. The Local Fairness Algorithm [16] guarantees request removal by introducing the station's ID in the request. Then, if a station observes a request with its own ID, it removes it from the bus. In this way, the maximum distance that a request can travel is one ring. The main disadvantage of this approach is that the request field should be extended to more than one bit, i.e., it must be long enough to identify every station in the network. Since the frame format should not be affected by system parameters, such as the number of active stations, a large number of bits must be reserved for the request field, so that the network can handle a large number of station; for instance, 10 bits are needed for 1024 stations. Furthermore, if new versions of the MAC protocol are anticipated to support multiple priority traffic, the request field of the slot should be even larger.

In the proposed FBAM mechanism, we introduce a different request removal approach. Our method is motivated by the fact that a request need not travel more than half a ring. This is because in a dual ring network, the bandwidth starvation of a station S_i can be caused only by another station S_k which is at most half a ring away, measured in number of hops. Overloaded stations at greater distances will select the other ring for the transmission of their packets and therefore will not affect the performance of S_i. A straightforward approach for ensuring that a request will not travel more than half a ring is to allow a station S_i, sending a request, to introduce in the request field of the slot the ID of the station S_k which should reset it, i.e., the ID of the station which is half a ring upstream from S_i. We have used the term I.D. based Request Removal Strategy (ID_RRS) for this method. The disadvantage of this approach is similar to that of the request removal approach used by the Local Fairness Algorithm; i.e., the request field must be large. Therefore, we propose here another method which we have called the Section Number based Request Removal Strategy (SN_RRS). The SN_RRS approach requires a much smaller request field and operates as follows: A request field of k bits is introduced into the Access Control Field (ACF) of each slot. A station that wants to insert a request will wait until it observes a slot with its request field equal to 0 and set the least significant bit of this field to 1. Also introduced into the ring is a set of special stations which are responsible for releasing the request fields. We call these stations Request Field Erasure Stations (RFES). Each RFES increases the value of each passing request field (which is not 0) by 1 and resets to 0 every request field with all of its k bits equal to 1. To illustrate this operation, let us consider the 20 stations network of Fig.5.2 with RFES stations S_1, S_6, S_{11} and S_{16}. Furthermore, let us assume that the request field is equal to 2 bits ($k = 2$), and that station

S_2 inserts a request on the reverse channel, i.e., it changes a passing 00 request field to 01. The first RFES observing this request field (i.e., station S_1) will increase its value to 10. The next RFES (station S_{16}) will make it 11 and, finally, station S_{11} will reset it to 00.

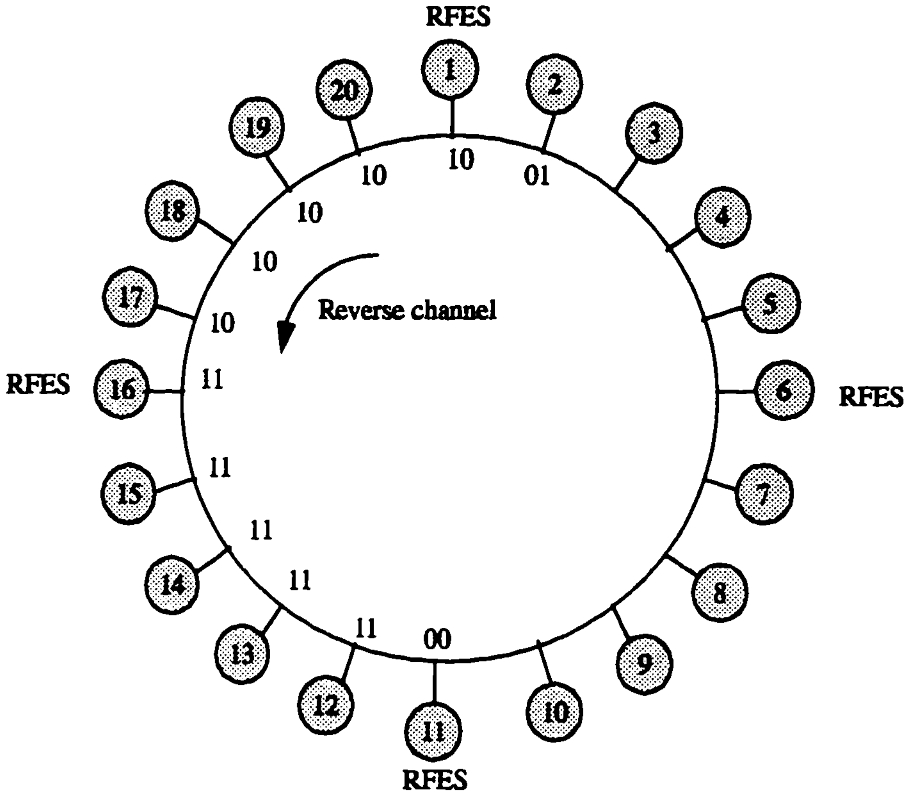

Fig.5.2 A request removal example

It it evident from Fig.5.2 that if an RFES wants to insert a request (for instance, station S_1), it must initialize the value of an idle request field (i.e., 00) with 10 instead of 01, i.e., it is as if this station sends a request and at the same time increases the value of the request field by 1. It should also be noticed from Fig.5.2 that the path travelled by a request becomes minimum (i.e., half a ring) when it is inserted by an RFES. For instance, in Fig.5.2, a request sent by station S_1 will be removed by RFES S_{11} which is 10 hops away. On the other hand, a request travels the longest path when it is sent by the first station which is upstream from an RFES. For instance, in Fig.5.2, a request sent by station

S_5 will be removed by RFES S_{11} which is 15 hops away; i.e., the request has traveled in this case $3/4$th of the ring. We can decrease the length of the maximum path that a request can travel by increasing the size of the request field and the number of the RFESs in the network. In a later section, we investigate the effect of the number of RFESs on performance.

5.2.2 Request Field Size and Number of RFESs

Let us assume that the request field is k bits. Then, after a station has set the request's least significant bit to 1 (i.e., it has inserted a request), this request field has to pass through $2^k - 2$ RFESs before all of its bits become 1. We want this to happen just before the station which is half a ring away. In this way, it is guaranteed that the request has travelled half a ring before it is reset by the next RFES. In order for this to happen, there must be at most $2^k - 2$ RFESs on each half ring, i.e., the total number of RFES, must be less than or equal to $2(2^k - 2) = 2^{k+1} - 4$. For instance, if a k = 2 bits request field is used, then the maximum number of RFESs will be 4. If a 3 bit request field is used, then the maximum number of RFESs will be $2^{3+1} - 4 = 16 - 4 = 12$. It is evident that the RFESs must be distributed as uniformly as possible around the ring. Furthermore, every station must have the capability of becoming an RFES so that redistribution of the RFESs is straightforward as the number of the stations on the ring increases. We finally mention that our request field erasure mechanism can easily accommodate the cases where the number of RFESs is smaller than the maximum. The only requirement in this case is the initialization of an all 0 request field with the appropriate number. For instance, if a 3 bit request field is used, and we have only 7 (instead of 12) RFESs, then a station sending a request must initialize the request field with 011 (an RFES must initialize this field with the value 100).

5.3 The Fair Bandwidth Allocation Mechanism (FBAM)

In this section, we provide a detailed description of the FBAM algorithm. Most of the parameters, counters, and registers that are required for its operation are similar to those of the BI_BWB and we refer for their description to Chapter 4. In addition, an Erased Slot Counter (***ES_CTR***) is needed per station to determine the number of requests that can be removed on the reverse channel due to the erased slots by the station on the forward channel. The operation of the ES_CTR is as follows.

Forward Channel: Whenever a station sees an idle and unreserved slot (i.e., RQ_CTR = 0 and RG_CTR ≤ REQ_QS) and either transmits a registered segment or it has a request in its request queue that has not yet been sent upstream, it increases its ES_CTR by one. A station also increases its ES_CTR by 1 when it erases a slot and: a) there exists an active downstream station (i.e., RQ_CTR > 0), or b) the station itself has a request in its request queue, or c) the station itself is going to use the slot to transmit a registered segment. Finally, when a station becomes idle and its RQ_CTR = 0, it decreases its ES_CTR by 1 for every idle and unreserved slot it sees on the forward channel.

Reverse Channel: When a station sees an RF > 0 on the reverse channel and its ES_CTR>0, it resets the RF to 0 and decrements ES_CTR by 1. On the other hand, if the RF = 0 and both ES_CTR and REQ_QS are greater than zero, the station decrements both of them by one.

5.3.1 The FBAM Algorithm

We now present the FBAM algorithm by describing its reaction to various events.

a) *New packet arrival:* UNRG_CTR increases by the number of segments in the packet.

b) *A segment becomes first in the transmission queue:* if UNRG_CTR > 0 and BWB_CTR < BWB_MOD - 1, a request will be sent upstream, RG_CTR will increase by 1, and UNRG_CTR will decrease by 1.

c) *A slot arrives at the forward channel:*

c1: If the slot is an unreserved idle slot (RG_CTR ≤ REQ_QS and RQ_CTR = 0) and the station is going to use it to transmit a registered segment (RG_CTR > 0), or there are requests present in the station's request queue (REQ_QS > 0), the ES_CTR will increase by one.

c2: If the slot has been written with a segment destined for this station, the station will reset the slot including the TAR bit (if it is 1). In addition, if this station or a downstream station is going to use the slot (i.e., RQ_CTR > 0 or RG_CTR > 0 or REQ_QS > 0), the station will increase its ES_CTR by one.

c3: If the written slot has a segment destined for a downstream station, its TAR bit is one, and the station has unregistered segments, it will send an extra request upstream, increase RG_CTR by one, and decrease UNRG_CTR by one. In addition, if the number of the TAR segments in the station's queue is greater than the number of the TAR bits that the station owes to the downstream stations, it will reset the TAR bit of the passing slot and increase its DBTAR_CTR by one

c4: If the passing slot is free and RQ_CTR>0, the station will decrease its RQ_CTR by one and let the idle slot go by.

d) ***A slot is seen on the reverse channel:***

d1: If RF > 0, the station will increase the RQ_CTR by one.

d2: If the station is an RFES, it will increase the RF by one and check whether it must reset it.

d3: If ES_CTR > 0 and RF > 0, the station will reset RF to 0 and decrease its ES_CTR by one. If RF = 0, and both ES_CTR and REQ_QS are greater than zero, the station will decrease both ES_CTR and REQ_QS by one. Finally, if ES_CTR = 0 and RF = 0 and REQ_QS > 0, the station will send a request upstream.

e) ***The station is allowed to transmit a segment:*** if a station does not have any segments waiting in its queue, the station will decrease ES_CTR by 1 (if its ES_CTR is greater than 0). Otherwise it will transmit the first segment in its queue and modify the control information according to the following:

e1: It will increase BWB_CTR by 1. If BWB_CTR becomes equal to BWB_MOD (i.e., the station transmits a TAR segment), then: a) it will reset the BWB_CTR to 0, b) if DBTAR_CTR > 0 or both DBTAR_CTR and RG_CTR are 0, it will set the TAR bit to 1, c) if DBTAR_CTR > 0, it will decrease DBTAR_CTR by 1.

e2: If RG_CTR > 0, it will decrease RG_CTR by 1. Otherwise, it will decrease UNRG_CTR by 1.

5.3.2 Algorithm Discussion

The objective of FBAM is to introduce local fairness on each ring section which accommodates transmissions by many stations while keeping the interaction among the various ring sections to the minimum. Such an interaction can be easily eliminated at the forward channel since slots with TAR = 1 bits are reset by the destination stations of the written slots. However, it is very difficult to achieve on the reverse channel, since a request from one ring segment can pass to another ring segment. Under FBAM, a request may be removed in three different ways:

a) Through the Section Number based Request Removal Strategy (SN_RRS) which ensures the removal of a request.

b) By a station which has recently seen an unreserved idle slot; i.e., when the conditions RQ_CTR = 0 and RG_CTR $\leq$ REQ_QS hold. RQ_CTR = 0 means that currently none of the downstream stations has requested a slot. RG_CTR $\leq$ REQ_QS reflects the fact that the station has not yet sent onto the channel all the requests for its registered segments. The reasoning behind this approach is that neither a downstream station nor the station itself has reserved the arriving idle slot. Therefore, it can be used by a station that has a request, and thus there is no need for this request to propagate upstream.

c) By a station which has erased a slot; certainly this is an idle and unreserved slot. In such a case, the station will increment its ES_CTR by 1, if its RG_CTR > 0 or RQ_CTR > 0. If either of the previous conditions hold, it is certain that this station or a downstream station will write on the slot. Since a request has already been sent by either of these stations, the station that erased the slot can remove a passing request on the reverse channel.

A station may remove requests either from the channel or from its own request queue (REQ_QS). For each slot period a station can remove at most one request. In addition, higher priority is always given to the removal of channel requests. The objective is to minimize the number of requests that upstream stations see, since a request is immediately removed from the channel and there is still the possibility for the waiting request at the station to be removed at a later time. It should also be noted that according to the FBAM operation, the station is not allowed to introduce its own request after it has erased a request from a passing slot on the reverse channel.

5.4 Performance Analysis

In this section, we use simulation results to investigate the effect of the system parameters on the throughput performance of FBAM. We consider a 10 station, 155 Mbps network with a slot size of 53 bytes, and a signal propagation delay of 5 μ sec/km.

5.4.1 Effect of Request Removal Approach and BWB_MOD on Performance

In this section, we investigate the effect of the request removal approach as well as of the value of BWB_MOD on the FBAM throughput performance. We consider the above mentioned 10 station, 155 Mbps network with a distance between neighbor stations 4 slots; i.e., the total cable length is 40 slots (21.8 km).

In Table 3.1, we show the effect of the request removal approach on the maximum throughput THR_i of each station S_i. We compare the ID_RRS approach, in which each station inserts in the request field of the slot the ID of the station which is half a ring upstream from it, with the SN_RRS approach which uses the Request Field Erasure Stations (RFES) for removing the requests. In the SN_RRS case RFESs are stations S_1, S_3, S_5, S_8. We assume that all stations are overloaded and try to write on every idle and unreserved slot on the channel randomly selecting, every time, the destination of their transmission. Table 5.1 shows that the ID_RRS and the SN_RRS methods provide similar throughputs. That is, the SN_RRS method, which requires a much smaller request field than the ID_RRS method, is very effective.

Table 5.1 Comparison of the maximum throughput under ID_RRS and SN_RRS

Request Removal method	Station Throughput										Net
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	THR
ID_RRS	0.360	0.360	0.360	0.360	0.360	0.360	0.360	0.360	0.360	0.360	3.60
SN_RRS	0.359	0.358	0.359	0.357	0.359	0.359	0.357	0.359	0.357	0.358	3.59

In Table 5.2, we show the effect that the value of BWB_MOD has on the maximum throughput of each station under the traffic load of Table 5.1 and the SN_RRS mechanism. Table 5.2 shows clearly that the effect of the BWB_MOD parameter is minor. Since small values of BWB_MOD allow faster convergence to the steady state, where the fair bandwidth allocation is achieved, the use of a small BWB_MOD value (i.e., 2) is recommended.

Table 5.2 Effect of BWB_MOD value on throughput performance

BWB_MOD	Station Throughput										Net
Value	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	THR
12	0.354	0.359	0.354	0.357	0.358	0.355	0.358	0.354	0.357	0.360	3.57
8	0.354	0.359	0.351	0.358	0.358	0.356	0.359	0.357	0.357	0.357	3.56
4	0.357	0.358	0.357	0.357	0.360	0.357	0.355	0.357	0.356	0.356	3.57
2	0.359	0.358	0.359	0.357	0.359	0.359	0.357	0.359	0.357	0.358	3.59

5.4.2 Throughput Comparison

In this section, we compare the throughput performance of the FBAM mechanism with the corresponding performance of the Global Fairness Algorithm (GFA), Local Fairness

Algorithm (LFA), as well as with the one of the mechanism under which stations can transmit without any restriction. We have used the term the No Restriction Mechanism (NRM) for this access mechanism. We consider six traffic scenarios which are commonly used in the literature for the evaluation of the fairness of proposed access mechanisms. In our investigation, we consider two cases of network size. In the first, the distance between neighbor stations is 1 slot., in the second, 10 slots; the corresponding total length of the cable will then be 10 slots (5.5km) and 100 slots (54.7 km), respectively. In the case of FBAM, the value of BWB_MOD is equal to 2 and the request field of the slot is 2 bits. In addition, the SN_RRS request removal strategy is used and Request Field Erasure Stations (RFESs) are stations S_1, S_3, S_5, S_8. Finally we select $k = l = 12$ in the case of GFA and LFA, and $w = 2$ in the case of LFA, i.e., a station considers that bandwidth starvation begins when it is in the unrestricted mode and observes $w = 2$ consecutive slots busy. The above values of k, l, and w provide for LFA and GFA the highest aggregate throughput in the case of uniform traffic and a 5.5 km network.

- ***Traffic Scenario 1:*** It is shown in Fig.5.3. The arrows show the transmission paths of each station. Fig.5.3 shows that the ring is divided into four independent segments (i.e., segments having non-overlapping transmission paths). The first segment includes station S_1. The second, includes stations S_2 and S_3. The third, includes stations S_4, S_5 and S_6. Finally, the fourth, includes stations S_7, S_8, S_9, and S_{10}. Each station transmits to the head station of the next group, making the most downstream link in each group the bottleneck link.

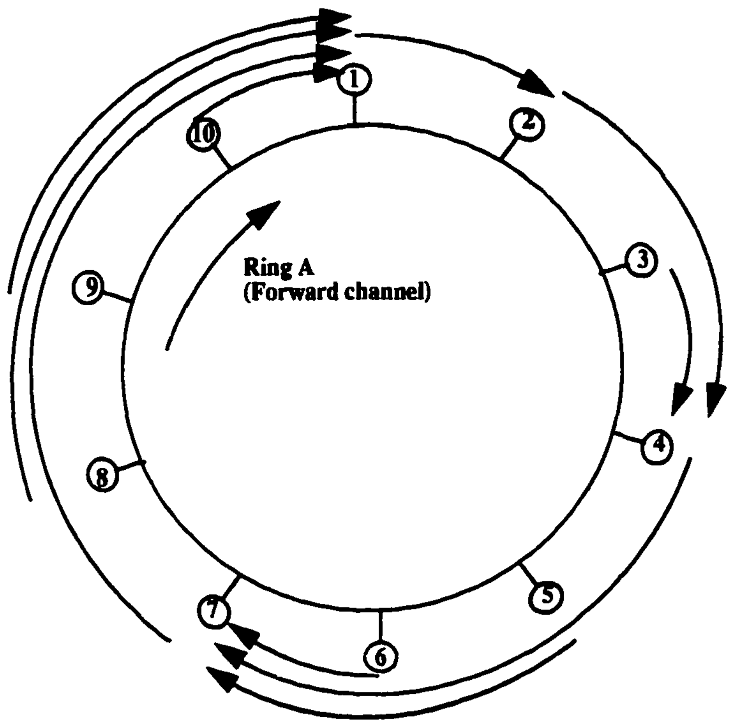

Fig.5.3 Traffic scenario 1

Table 5.3, for the short network, and Table 5.4, for the long network, show the normalized throughputs of the stations under the various medium access mechanisms. Both tables clearly show that only FBAM can evenly distribute the available channel bandwidth of each ring segment among the competing stations. Furthermore, its performance is not affected by the network size. In contrast, LFA always provides a smaller bandwidth to the downstream stations of each ring segment. Although GFA can provide all stations with the same bandwidth, its aggregate network throughput is much lower than that of the other mechanisms even in the case of the short network, where the quota $k = 12$ is larger than the round trip propagation delay (which is 10 slots). Table 5.3 shows that the GFA operation provides a throughput for each station the throughput of the stations in the most congested link, which in the case of Fig.5.3 is the link between stations S_{10} and S_1. Thus, GFA is not a bandwidth efficient mechanism. Moreover, its bandwidth performance is strongly affected by the network size. As Table 5.4 shows, when the network size increases from 10 to 100 slots, the throughput of each station drops to 0.12 segments/slot and that of the network drops to 1.2 segments/slot. Finally, both tables clearly show that under the NRM operation the throughputs of some stations become 0, i.e., bandwidth starvation is observed.

Table 5.3 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 1. Short network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	1.00	0	1.00	0	0	1.00	0	0	0	4.00
GFA	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2.50
LFA	1	0.54	0.46	0.36	0.33	0.31	0.27	0.24	0.24	0.23	4.00
FBAM	1.00	0.50	0.50	0.33	0.33	0.33	0.25	0.25	0.25	0.25	4.00

Table 5.4 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 1. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	1.00	0	1.00	0	0	1.00	0	0	0	4.00
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	1.00	0.52	0.48	0.34	0.33	0.32	0.26	0.25	0.25	0.24	4.00
FBAM	1.00	0.50	0.50	0.33	0.33	0.33	0.25	0.25	0.25	0.25	4.00

- ***Traffic Scenario 2:*** It is shown in Fig.5.4. In this scenarios the bottleneck link of each ring section of competing stations has moved to the middle of this section; it was the last link in the case of scenario 1. Table 5.5 and Table 5.6 show the corresponding throughputs of the various stations under the short and long networks, respectively. Both tables show that bandwidth starvation is again observed in the case of NRM while GFA is fair but very inefficient, especially in the case of the long network. LFA improves significantly the throughput performance of the network relative to NRM and GFA, however, it still fails to provide the same throughput to the stations of each ring section; upstream stations continue to acquire a slightly higher bandwidth. FBAM is the only mechanism which is both fair and bandwidth efficient, i.e., it provides the highest aggregate throughput.

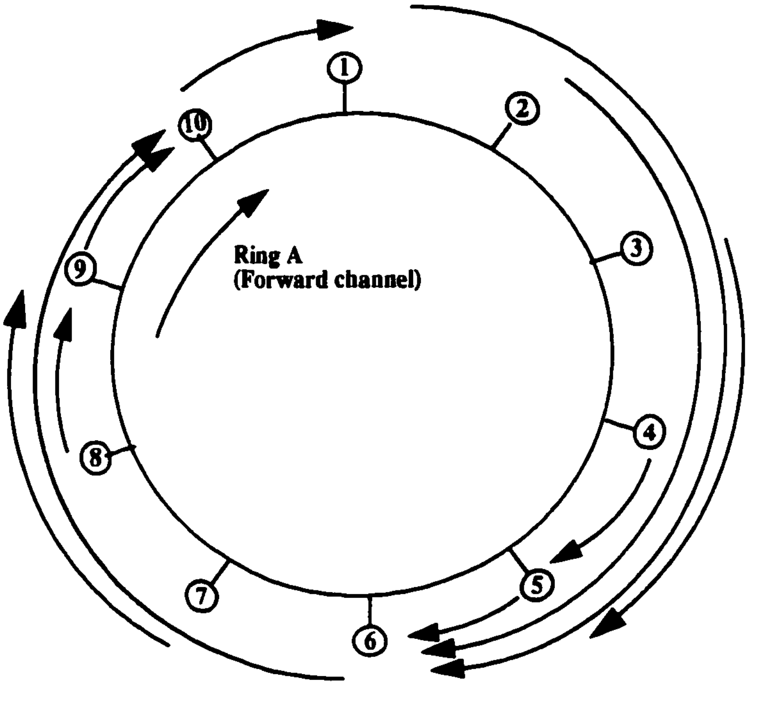

Fig.5.4 Traffic scenario 2

Table 5.5 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 2. Short network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	0	0	0	0	1.00	0	0	0	1.00	3.00
GFA	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2.50
LFA	0.27	0.25	0.25	0.23	0.48	0.36	0.33	0.31	0.64	1.00	4.12
FBAM	0.25	0.25	0.25	0.25	0.50	0.33	0.33	0.33	0.67	1.00	4.17

Table 5.6 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 2. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	0	0	0	0	1.00	0	0	0	1.00	3.00
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	0.24	0.24	0.24	0.24	0.52	0.32	0.31	0.31	0.68	1.00	4.09
FBAM	0.25	0.25	0.25	0.25	0.50	0.33	0.33	0.33	0.67	1.00	4.17

- ***Traffic Scenario 3:*** It is shown in Fig.5.5. The corresponding stations' throughputs are shown in Tables 5.7 and 5.8. Traffic scenario 3 is very similar to traffic scenario 2. The only difference is that station S_5 now has destination S_8 (before it had destination S_6). In this way, the boundaries between the various ring sections become less distinct than in the previous cases. Tables 5.7 and 5.8 clearly show that LFA now fails to balance the bandwidth among the various stations providing them with different throughputs. GFA is fair but inefficient while NRM causes bandwidth starvation. FBAM demonstrates again the best performance. It divides the stations into 3 groups. The first group only consists of S_{10} which does not compete with any other station and requires the entire bandwidth. The second group consists of S_5, S_6, S_7, S_8, and S_9. The transmission path of each one of these stations overlaps at a certain link with the transmission path of two other stations. Consequently, each one of them receives one third of the channel bandwidth. Finally, the third group consists of S_1, S_2, S_3, and S_4 whose transmission paths overlap at the link that connects S_4 to S_5. Consequently, each one of these stations acquires 1/4 th of the channel bandwidth under the long network but only 1/5 th of the channel bandwidth under the short network. A lower throughput is observed under the short network due to a smaller number of requests that are not erased effectively by the RFESs and prevent S_1, S_2, S_3, and S_4 to see the entire channel bandwidth.

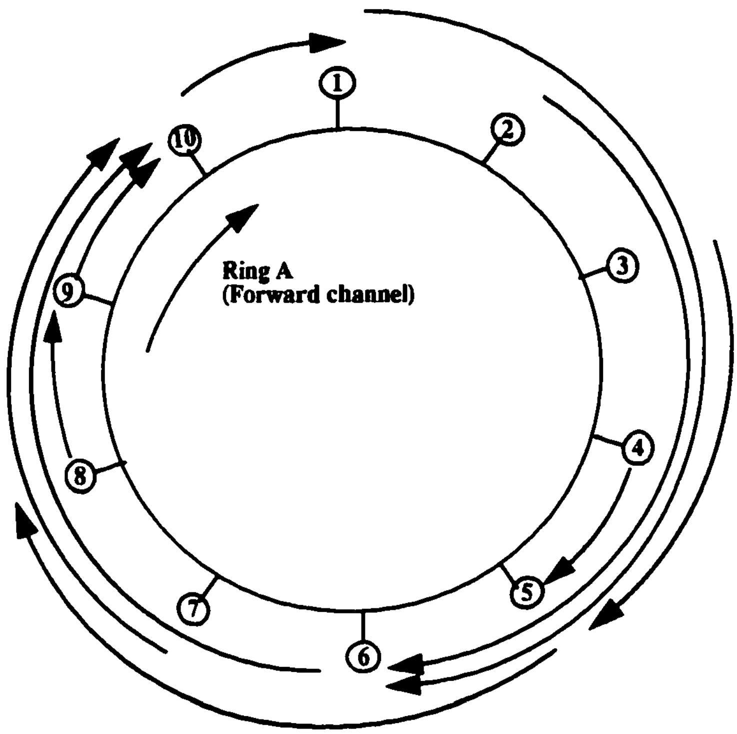

Fig.5.5 Traffic scenario 3

Table 5.7 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 3. Short network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	0	0	0	0	1.00	0	0	0	1.00	3.00
GFA	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	2.50
LFA	0.34	0.27	0.20	0.18	0.27	0.33	0.26	0.39	0.42	1.00	3.66
FBAM	0.20	0.20	0.20	0.20	0.33	0.33	0.33	0.33	0.33	1.00	3.46

Table 5.8 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 3. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	0	0	0	0	1.00	0	0	0	1.00	3.00
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	0.24	0.24	0.24	0.24	0.28	0.29	0.18	0.31	0.53	1.00	3.55
FBAM	0.25	0.25	0.25	0.25	0.33	0.33	0.33	0.33	0.33	1.00	3.67

- *Traffic Scenario 4:* It is shown in Fig. 5.6. The corresponding throughputs are shown in Tables 5.9 and 5.10. In this scenario, there are no distinct ring segments and all stations seem to belong to the same group. Consequently, as the two tables show, FBAM provides the same throughput to each station. The same is true for GFA, but not for LFA. Finally, NRM causes, again, bandwidth starvation.

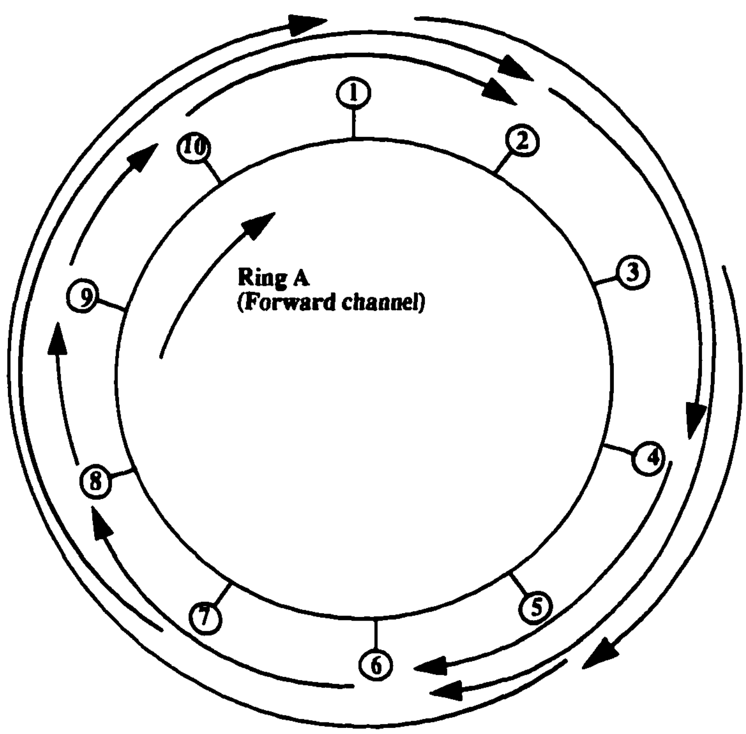

Fig.5.6 Traffic scenario 4

Table 5.9 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 4. Short network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	0	1.00	0	1.00	0	0	1.00	0	0	0	3.00
GFA	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.33	3.33
LFA	0.33	0.32	0.32	0.30	0.32	0.32	0.31	0.35	0.31	0.27	3.15
FBAM	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.33

Table 5.10 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 4. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	0	1.00	0	1.00	0	0	1.00	0	0	0	3.00
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	0.31	0.30	0.21	0.25	0.25	0.28	0.26	0.25	0.28	0.24	2.64
FBAM	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.33

- *Traffic Scenario 5:* In this traffic scenario, the ring segments are the same with those of the traffic scenario one which is shown in Fig.5.3. However, each station now evenly distributes its traffic among the downstream stations of its ring segment. For instance, station S_8 evenly distributes its generated traffic among stations S_9, S_{10} and S_{11}. The corresponding stations' throughputs are shown in Tables 5.11 and 5.12. The superiority of FBAM is again evident.

Table 5.11 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 5. Short Network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	1.00	0.50	1.00	0.33	0.50	1.00	0.25	0.33	0.50	6.42
GFA	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	4.50
LFA	1.00	0.75	0.63	0.57	0.50	0.56	0.53	0.40	0.39	0.54	5.86
FBAM	1.00	0.90	0.55	0.74	0.50	0.50	0.63	0.48	0.37	0.50	6.17

Table 5.12 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 5. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	1.00	1.00	0.50	1.00	0.33	0.50	1.00	0.25	0.33	0.50	6.41
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	1	0.75	0.63	0.61	0.41	0.59	0.57	0.32	0.36	0.57	5.81
FBAM	1.00	0.97	0.52	0.75	0.50	0.50	0.65	0.49	0.35	0.50	6.23

- ***Traffic Scenario 6:*** In this scenario, every station randomly chooses the destination of each transmitted packet. The shortest path routing rule is used to decide on which ring a given packet must be sent. If the destination station is exactly half a ring away, 50 percent of the packets will be transmitted on one ring and the rest on the other ring. Tables 5.13 and 5.14 show the corresponding stations' throughput under the short and long networks, respectively. We see that under uniform traffic, all four access mechanisms can provide the stations with similar throughputs. However, the aggregate throughput in the case of FBAM is higher.

Table 5.13 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 6. Short network (5.5 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.363	3.60
GFA	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	3.41
LFA	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	3.50
FBAM	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	3.60

Table 5.14 Throughput comparison of FBAM, NRM, GFA, and LFA under traffic scenario 6. Long network (54.7 km)

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
NRM	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	3.60
GFA	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	1.20
LFA	0.34	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.33
FBAM	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36	3.60

5.4.3 Aggregate Throughput Comparison

In order to show clearly the ability of FBAM to introduce fairness in a very bandwidth efficient way, we summarize in Tables 5.15 and 5.16 the maximum aggregate throughput provided by the mechanisms under both the short and long networks. The two tables show that with the exception of traffic scenario 3, in the case of short network, FBAM provides a higher aggregate network throughput than the other two fairness mechanisms.

Table 5.15 Aggregate normalized throughput comparison of FBAM, NRM, GFA, and LFA under various traffic scenarios. Short network (5.5 km)

Traffic Scenario	Network Throughput			
	NRM	GFA	LFA	FBAM
1	4.00	2.50	4.00	4.00
2	3.00	2.50	4.12	4.17
3	3.00	2.50	3.66	3.46
4	3.00	3.33	3.15	3.33
5	6.42	4.50	5.86	6.17
6	3.60	3.41	3.5 0	3.60

Table 5.16 Aggregate normalized throughput comparison of FBAM, NRM, GFA, and LFA under various traffic scenarios. Long network (54.7 km)

Traffic Scenario	Network Throughput			
	NRM	GFA	LFA	FBAM
1	4.00	1.20	4.00	4.00
2	3.00	1.20	4.09	4.17
3	3.00	1.20	3.55	3.67
4	3.00	1.20	2.64	3.33
5	6.41	1.20	5.81	6.23
6	3.60	1.20	3.33	3.60

5.4.4 FBAM Max-Min Fairness

A careful examination of the FBAM throughput performance shows that in most cases the FBAM mechanism provides a Max-Min throughput fairness which is the optimum one can achieve in terms of bandwidth allocation. What is more impressive is that FBAM does not require a centralized control but it can achieve MAX-Min fairness in a distributed way. We are not aware of any other distributed algorithm that can demonstrate similar properties.

The objective of the Max-Min flow control [6], [44] is to maximize the throughput THR_i of each station S_i under the constraint that an incremental increase of THR_i will not result in a decrease of the throughput THR_j of another station S_j when THR_j < THR_i. The Max-Min fairness flow control is not only optimum in terms of fairness, but also enables the analytic computation of the stations' throughputs. Let N be the number of stations in a network, l_i be the link that comes out of station S_i (with a normalized capacity of 1), and THR_i be the throughput of station S_i, under the Max-Min flow control mechanism. The Max-Min Fairness Throughput Computation Algorithm is as follows:

1. Find the first network bottleneck link.

2. If there are n stations competing for the first bottleneck link, then the throughput of each one of them will be $1/n$.

3. Compute the unused normalized capacities in the remaining links. For instance, if k of the traffic paths of the bottleneck link traverse another link l_i, its unused normalized capacity will be: $1 - k(1/n)$.

4. Compute the bandwidth allocation C_i of each link that has bandwidth available and for which there are stations competing. For instance, if m transmission paths traverse the above link l_i (in addition to the k whose throughput has been computed), then $C_i = \frac{(1-k/n)}{m}$.

5. Find $C_{min} = min(C_1, C_2 ... C_i ..., C_N)$ with $C_i \neq 0$.

6. Compute the throughput of the stations of this link. If this is the last link that has unused bandwidth, stop. Otherwise, go to step 3.

In order to clarify the above computation, we apply it in the case of traffic scenario 1 shown in Fig.5.3. The different steps of the above algorithm now become: **1.** Link l_{10} is the first bottleneck link since it has the largest number of stations (i.e., flow) competing for bandwidth. **2.** The throughput of these four stations, i.e., S_7, S_8, S_9, and S_{10} will then be: $THR_7 = THR_8 = THR_9 = THR_{10} = 1/4 = 0.25$. **3.** Unused bandwidth now have all links except the bottleneck link l_{10}. **4.** Links for which stations compete are l_1, l_2, l_3, l_4, l_5, and l_6. Their corresponding bandwidth allocations are: $C_1 = C_2 = C_4 = 1, C_3 = C_5 = 0.5, C_6 = 0.33$. **5.** Minimum C_i is $C_6 = 0.33$. **6.** Consequently, $THR_4 = THR_5 = THR_6 = 0.33$. By repeating the steps "3.", "4.", "5." and "6." of the Throughput Computation Algorithm, we can find all the remaining throughputs which are: $THR_1 = 1$ and $THR_2 = THR_3 = 0.5$.

We see that the analytically computed throughputs are identical to the ones provided by FBAM in Tables 5.5 and 5.6. Extensive simulation results that we have run have shown that in most cases, FBAM can provide max-min throughput fairness. However, a small discrepancy is observed in some cases. This discrepancy is caused by a small number of requests sent by downstream stations which are not removed effectively by upstream stations, and are passed to upstream ring sections affecting the throughputs of their stations. Tables 5.17 and 5.18 show such a discrepancy between the throughputs provided by FBAM and the MAX-MIN flow control mechanism in the case of traffic scenario 3 and 5, respectively. We see that this discrepancy is rather negligible under the long network and becomes more evident under the short network.It should be noted here that in all other traffic scenarios the FBAM and MAX-MIN flow control algorithms provide identical throughputs to all station.

Table 5.17 Comparison of FBAM and MAX_MIN under traffic scenario 3

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
FBAM (short net)	0.20	0.20	0.20	0.20	0.33	0.33	0.33	0.33	0.33	1.00	3.46
FBAM (long net)	0.25	0.25	0.25	0.25	0.33	0.33	0.33	0.33	0.33	1.00	3.67
MAX_MIN	0.25	0.25	0.25	0.25	0.33	0.33	0.33	0.33	0.33	1.00	3.67

Table 5.18 Comparison of FBAM and MAX_MIN under traffic scenario 5

Access Mechanism	Station Throughput										Network Throughput
	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	
FBAM (short net)	1.00	0.90	0.55	0.74	0.50	0.50	0.63	0.48	0.37	0.50	6.17
FBAM (long net)	1.00	0.97	0.52	0.75	0.50	0.50	0.65	0.49	0.35	0.50	6.23
MAX_MIN	1.00	1.00	0.50	0.75	0.50	0.50	0.67	0.50	0.34	0.50	6.26

5.5 Conclusions

In this chapter, the Fair Bandwidth Allocation (FBAM) mechanism for dual ring network architectures has been introduced. FBAM has the ability to minimize the interactions between ring segments that connect stations with no overlapping transmission paths, thus maximizing the aggregate network throughput. In addition, it can evenly distribute the channel bandwidth among the competing stations of each ring segment. Moreover, its throughput performance is almost insensitive to the system parameters. Our performance investigation has also shown that FBAM, which is a distributed access mechanism, can achieve in most cases a Max-Min throughput fairness, which is considered to be the optimum throughput fairness. Furthermore, in the cases where the throughputs provided for the stations by FBAM and Max-Min flow control differ, the observed discrepancy is rather small. This is a significant advantage of FBAM over all the other previously proposed access mechanisms. Finally, FBAM enables stations to send requests for bandwidth reservations continuously so that they do not encounter the rather difficult problem of deciding when bandwidth starvation occurs. In contrast, the Global Fairness Algorithm (GFA) cannot effectively utilize the channel bandwidth in the cases of traffic patterns that would allow many channel slots to be reused, and thus improve the aggregate throughput of the system. Furthermore, its throughput performance is strongly affected by the network size or channel speed. The local fairness algorithm (LFA), on the other hand, although it can prevent bandwidth starvation as well as effectively reuse the channel bandwidth, it cannot provide the stations of the same ring segment with similar throughputs. Furthermore, its performance is sensitive to the system parameters, and in many cases, its aggregate throughput is smaller than the throughput provided by FBAM. Finally, it has the difficult task of deciding when bandwidth starvation is observed.

CHAPTER 6

THE EFFECTIVE PRIORITY BWB MECHANISM FOR DUAL RING ARCHITECTURES

6.1 Introduction

The objective of this chapter is to introduce an effective priority mechanism for high speed dual ring networks that can minimize the effect of low priority traffic on high priority traffic and guarantee the stringent delay requirements of real-time traffic. The organization of this chapter is as follows. In section 6.2, the main features of the Effective Priority Bandwidth Balancing (EP_BWB) mechanism are introduced. In section 6.3, a detailed description of its operation is provided. In section 6.4, simulation results are used to investigate its delay and throughput performance in the presence of two priority classes of traffic. Finally, in section 6.5, the conclusions of our investigation are presented.

6.2 Main Features of the EP_BWB Operation

The EP_BWB mechanism has two main objectives: The first is to guarrantee fair access among users of same priority. The second is to minimize the effect of low priority traffic on high priority traffic. The first objective is achieved by incorporating the FBAM mechanism into the EP_BWB operation. The second objective is achieved by enabling a high priority class to preempt the transmissions of lower priority classes. Preempting transmissions on a distributed network, especially when the distances between stations are large, is an extremely difficult task due to the large propagation delays. This is the main reason behind the lack of effective priority mechanisms for high speed dual ring networks. It is evident, that in order for a high priority traffic source to preempt the transmissions by heavily loaded lower priority traffic sources which are located upstream, the high priority source must be able to send multiple requests upstream. In addition, it must ignore the lower priority requests which arrive from downstream stations. Although ignoring lower priority requests is straightforward, the transmission of multiple requests raises two important questions: How many requests must an active class at a station send upstream so that channel bandwidth will not be wasted? How can we prevent these requests from interfering with the transmissions of similar priority classes in the other stations' queues? The proposed EP_BWB mechanism in this chapter provides simple answers to these two questions. The key idea is the following: according to EP_BWB each priority class P_i counts only the priority P_i requests. This is in sharp contrast with the other proposed priority mechanisms in the literature where each class counts requests of its own and of higher priority. Furthermore, every time a k segment packet of priority P_i arrives at a station, this station will immediately send upstream k requests for each priority P_k which is lower than P_i. Although these requests will preempt the transmissions of the lower priority upstream traffic sources, they will not affect the transmissions of the upstream priority P_i sources. In addition, because the number of low priority requests is exactly equal to the number of segments in the arrived priority P_i packet, no slot will be wasted.

The EP_BWB operation tries to separate as much as possible the interaction between the various priority classes. Then, it will be much easier to balance the unused channel bandwidth by a high priority class among the users of the same lower priority class. Consequently, according to EP_BWB, each priority class inside a station behaves as a complete separate substation with its own parameters and counters. Furthermore, the

Access Control Field of the slot carries only one busy bit, and a separate Request Field (RF) and TAR bit for each priority.

The EP_BWB mechanism incorporates the request removal algorithm of FBAM into its operation. The Request Field Erasure Stations (RFES) are defined in a similar manner as in FBAM, and requests are removed by RFESs after they have travelled at least half of the ring. Since the Request Field (RF) of each priority requires two bits, the Access Control Field (ACF) of the DQDB slot in Fig. 2.1 is not sufficient to support all the required control bits. In Fig. 6.1, we show the proposed slot format of EP_BWB in the case of three priority classes. It should be noted that we have tried to keep the EP_BWB slot format as similar as possible to that of the QA slot of DQDB, since easy interconnection with IEEE 802.6 network is one of our primary objectives.

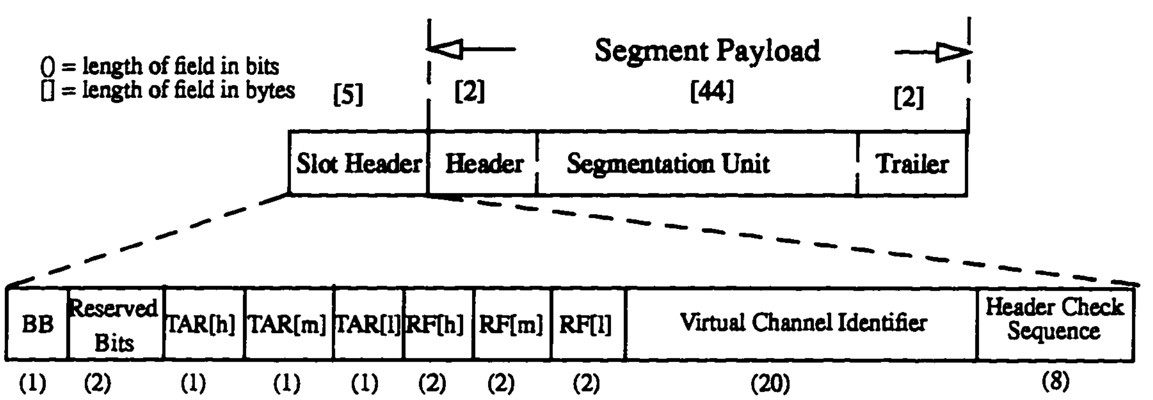

Fig.6.1 EP_BWB slot format

In Fig.6.2, we show the basic components of a station supporting three priority classes. Fig.6.2 shows the components which are responsible for the segment transmissions on one ring (Ring A) only. Another set of similar componets is needed for the reverse channel (Ring B). Each priority class P_i (where i = h,m,l for high, medium, and low priority, respectively) has its own transmission queue and its own set of counters such as the erased slot counter (ES_CTR[i]), request counter (RQ_CTR[i]), bandwidth balancing counter (BWB_CTR[i]), etc. The functions of these counters are similar to those of the corresponding counters in the case of FBAM. In addition, each priority class has been provided with an array of request queues REQ_QS[i,j] (i,j = h, m, l) which describes the number of priority P_j requests that priority class P_i must send upstream.

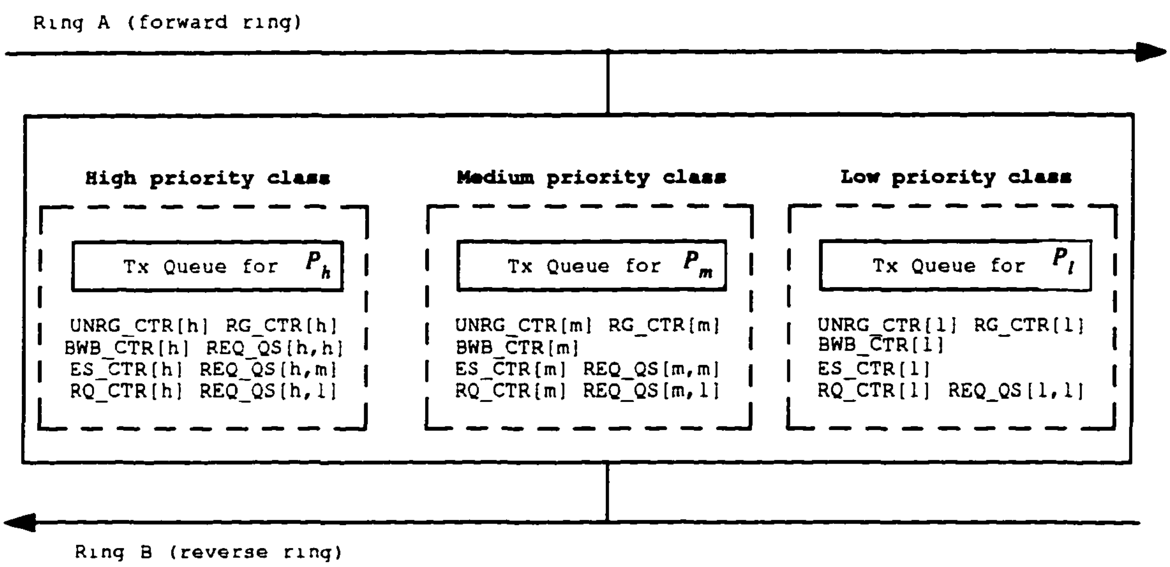

Fig.6.2 Station structure under the EP_BWB mechanism

Fig.6.2 clearly shows that each class P_i can send requests of the same and lower priority only; i.e., class P_h has three request queues (REQ_QS[h,h], REQ_QS[h,m], REQ_QS[h,l], class P_m has two request queues (REQ_QS[m,m], REQ_QS[m,l]) and class P_l has only one request queue (REQ_QS[l,l]). In the sequel, we elaborate on the various actions that a substation (priority class) can take.

6.2.1 Transmission of Requests

According to the EP_BWB operation, when a segment of priority P_i becomes first in queue, a same priority request will be sent upstream. (i.e., REQ_QS[i,i] will increase by 1). In addition, whenever a segment of priority P_i is generated at a station, regardless of whether it is first in queue or not, a separate request will be sent upstream for each priority class P_j which is lower than P_i (i.e., the corresponding REQ_QS[i,j] will increase by 1). The main objective of this approach is to ensure that a control mechanism similar to FBAM is applied to the segments of the same priority class. In addition, the higher priority class can immediately request all the bandwidth that it needs for the transmission of its segments.

6.2.2 Transmission of Segments

Each priority class operates independently on the ring. A segment of priority P_i will be sent if and only if the corresponding request counter of priority P_i (RQ_CTR[i]) is 0. When an idle slot arrives at a station, the highest priority class sees this slot first. The idle slot may be written by a lower priority class only if: a) it has not been reserved by any downstream class of similar or higher priority; b) the higher priority classes inside the same station are idle.

6.2.3 Slot Classification

An arriving slot at a station can be classified either as an unreserved idle slot, a reserved idle slot, or a busy slot. In EP_BWB the status of a slot is determined as follows:

a) A slot is considered to be an unreserved idle slot for priority class P_i, if its busy bit is 0, RQ_CTR[i] = 0, and RG_CTR[i] $\leq$ REQ_QS[i,i] (i.e., when neither a downstream station nor the station itself has sent a priority i request upstream).

b) A slot is also an unreserved idle slot, if its busy bit is 1 and its destination is this station.

c) A slot is considered to be a reserved idle slot, if its busy bit is 0 and either $RQ\_CTR[i] > 0$ or RG_CTR[i] > REQ_QS[i,i].

d) A slot is a busy slot, if its busy bit is 1 and has destination a downstream station.

6.2.4 Erased Slots Counter Update

In EP_BWB, each priority class determines whether the incoming idle slot is reserved or not, and decides on whether it should remove requests accordingly. If a segment can be sent in an incoming unreserved slot, there is no need to send upstream the set of requests that are generated for this segment. Every priority class P_i has its own Erased Slot Counter (ES_CTR[i]) that counts the number of requests it can remove from the channel. If class P_i detects an unreserved idle slot, it increments the ES_CTRs of priority P_i and lower according to the following rules:

a) If a registered priority P_i segment is going to use an unreserved slot (RG_CTR[i] > 0 or RQ_CTR[i] > 0), every ES_CTR[j] of priority $P_j \leq P_i$ is incremented by 1; this is because the registered priority P_i segment has already sent a separate request for every class $P_j \leq P_i$.

b) If an unregistered priority P_i segment is going to use the idle slot, every ES_CTR[j] of priority $P_j < P_i$ is incremented by 1; this is because the unregistered class P_i segment has already sent a separate request for every class $P_j < P_i$.

c) If a class P_i is idle (i.e., RQ_CTR[i] = 0 and RG_CTR[i] = 0 and UNRG_CTR[i] = 0), it will clear all the priority P_i request queues in the station (i.e., set REQ_QS[j,i] = 0, where priority P_j is higher than or equal to P_i); this is because there is no more priority P_i segment on the station and all the requests for downstream priority P_i segments have already been served. Therefore, all the priority P_i requests that are still left in the station are redundant.

In both cases "a)" and "b)", if an unreserved idle slot of priority P_i happens to be a reserved slot of priority P_j, ES_CTR[j] will remain unchanged. The reason is that since class P_i has occupied a reserved idle slot of priority P_j, it should return class P_j another idle slot by sending a priority P_j request upstream. Therefore, it should not increment ES_CTR[j] in the cases of "a)" and "b)".

6.2.5 Removal of Requests

For each class P_i, whose ES_CTR[i] is greater than 0, a request of priority P_i will be removed on the reverse channel and ES_CTR[i] will be decremented by 1. The reverse channel is always checked first by a station to examine if an incoming segment contains a priority P_i request or not. If it does, the request on the reverse channel is removed even if there is a request in the station's request queue. Higher priority is given to the removal of requests on the reverse channel because the requests in the request queue will have additional opportunities to be removed before their transmission. Thus, fewer requests will be transmitted onto the channel.

If ES_CTR[i] is greater than 0 and there is no request in the channel, class P_i will try first to remove a request of priority P_i from its own request queue (REQ_QS[i,i]). If

REQ_QS[i,i] = 0, class P_i will try to remove a P_i request from the next higher priority request queue (REQ_QS[i+1, i]), and so on. If one priority P_i request is removed, the ES_CTR[i] will decrease by one.

Once a request has been removed, no other class at the station can write on this request. This is to prevent unnecessary requests from being sent upstream. For instance, consider the head station of a distinct ring section with active stations. Let us assume that this head station receives continuously unreserved idle slots. It is evident that it should never send any request upstream. The reason is the following. According to EP_BWB operation, if the head station receives unreserved idle slots continuously, then the erased slot counters for all priority classes with non-empty request queues will always be positive. By not allowing this station to write on a request field, after it has removed a request of the same priority, no requests will be sent upstream and the transmissions on this ring section will not affect the transmissions of other upstream ring sections.

6.3 The EP_BWB Access Algorithm

We now provide a complete presentation of the EP_BWB access algorithm by describing the reaction of each station to various events.

a) A new packet of priority P_i arrives at the station: the UNRG_CTR[i], as well as each REQ_QS[i,j] of $P_j < P_i$, increases by the number of segments in the packet.

b) A priority P_i segment becomes first in the queue: if UNRG_CTR[i] > 0 and BWB_CTR[i] < BWB_MOD-1, a P_i request will be sent upstream, RG_CTR[i] will increase by 1 and UNRG_CTR[i] will decrease by 1.

c) A slot arrives at the forward bus:[2]

c1: If the slot is destined for the station and the station, or a downstream class P_i , is going to use the slot (i.e., RQ_CTR[i] > 0 or RG_CTR[i] > 0 or UNRG_CTR[i]>0), the station should reset the busy bit of the slot. The station should also reset the busy bit of the slot if the slot is destined for the station and P_i is currently the lowest active priority class in the station.

c2: If there are still priority P_i requests waiting in the request queues belonging to classes P_i and higher, and P_i is not going to use an unreserved slot (RG_CTR[i] = UNRG_CTR[i] = RQ_CTR[i] = 0), the station will remove all priority P_i requests (i.e., set all REQ_QS[j,i] to 0, where the priority of P_j is higher than P_i).

c3: For each class $P_j (P_j \leq P_i)$, if the slot is neither reserved by priority P_i nor by priority P_j, then the station will increase ES_CTR[j] by one, if one of the following conditions is true: a) the priority P_i class is going to use the unreserved slot to transmit its registered segment (RG_CTR[i] > 0); b) a downstream priority P_i class is going to use the slot to transmit its segment (RQ_CTR[i] > 0); c) the priority P_i class is going to use the unreserved slot to transmit an unregistered segment and P_j is not equal to P_i (i.e., UNRG_CTR[i] > 0 and $P_j \neq P_i$).

c4: If the TAR[i] bit of the segment is set and there are unregistered priority P_i segments waiting in the queue, the station will send an extra priority P_i request

2. Slots arriving on the forward bus are seen by the various classes in a descending order; i.e., higher priority classes see these slots first.

upstream. In addition, it will increase RG_CTR[i] by one and decrease UNRG_CTR[i] by one. If the number of the priority P_i TAR segments in the station's queue is larger than the number of the priority P_i TAR bits that the station owes to the downstream stations, the station will reset the TAR[i] bit in the bus and will increase DBTAR_CTR[i] by one

c5: If priority class P_i sees an empty slot (i.e., higher priority classes have not written on the slot), it will either decrease RQ_CTR[i] by one (if it is greater than 0) or transmit a priority P_i segment (if it has one).

d) ***A slot arrives on the reverse bus:*** [3]

d1: If the P_i request field of the slot is set, the station will increase the RQ_CTR[i] by one.

d2: If the station is an RFES, it will first examine RF[i] to see whether it should reset it. Otherwise, it will increase it by 1.

d3: If the P_i request bit on the channel is set or the ES_CTR[i] is greater than 0, the station will not write on the request field of the slot. The station will send a priority P_i request if ES_CTR[i] is 0, the request bit on the channel is not set, and one of the REQ_QS[j, i] of priority P_j $(P_j \geq P_i)$ is greater than 0.

d4: If the ES_CTR[i] is greater than 0, the station will try to remove a request of priority P_i in the following order:

d4a) If the P_i request field of the slot is set, the station will reset it.

3. Slots arriving on the reverse bus are seen by the various priority classes of a station in an ascending order (i.e., lowest priority class sees them first)

d4b) If the request queue REQ_QS[i,i] of class P_i is greater than 0, the station will decrease REQ_QS[i,i] by one

d4c) The station will check the REQ_QS[j,i] queues in priority ascending order. If REQ_QS[j,i] is greater than zero, the station will decrease it by one.

If the station does remove a priority P_i request, it will decrease ES_CTR[i] by one.

e) ***The station transmits a segment:*** the highest priority segment is transmitted. If this is of priority P_i the station will take the following actions:

e1: It will increase the bandwidth balancing counter of priority P_i by one. If the transmitted segment is a TAR segment (i.e., BWB_CTR[i] = BWB_MOD[i]), it will reset BWB_CTR[i] to zero, set the TAR[i] bit in the forward bus to 1, and decrease the DBTAR_CTR[i] (if it is greater than zero) by one. We should notice that the station will not set the TAR[i] bit to 1 if DBTAR_CTR[i] is zero and there are registered P_i segments in the waiting queue.

e2: If the RG_CTR[i] is greater than 0, the station will decrease it by one. Otherwise, it will decrease the UNRG_CTR[i] by one.

6.3.1 Algorithm Discussion

In the EP_BWB mechanism, high priority segments will notify upstream low priority classes about their presence as soon as they arrive at a station. Furthermore, they will notify similar priority upstream classes of their arrivals according to the FBAM mechanism. A station can thus prevent upstream low priority data from delaying the high priority segment transmissions while fairly sharing the available bandwidth with similar priority sources inside other stations. In the remaining sections, we elaborate on three issues regarding the EP_BWB operation.

6.3.1.1 Guaranteeing the Reserved Slots by Low Priority Traffic: When an idle slot arrives at a station the higher priority class can always use the slot first, regardless of whether the slot has been reserved by a high priority class or by a low priority class. If a high priority segment has used a slot which has been reserved by a low priority class, EP_BWB will ensure that the low priority class will get an idle slot later on. For instance, consider an overloaded network in which only two priority classes are active. Let us assume that a station S_i has a low priority segment for which it has already sent a request, i.e., RG_CTR[l] = 1. If S_i now generates a high priority segment before the arrival of the idle slot that has been reserved by the low priority segment, one request would be sent to each of the high priority's request queues (i.e., REQ_QS[h,h] = 1 and REQ_QS[h, l] = 1). When the idle slot that has been reserved by the low priority segment arrives, it will be used by the high priority segment. If at this time the low priority request in REQ_QS[h, l] has already been sent, the slot that will be reserved by this request can be used to serve the low priority segment. Otherwise, i.e., if REQ_QS[h,l] = 1, then according to step "*c3*" of the algorithm, whenever REQ_QS[l,l] (= 0) < RG_CTR[l] (= 1), the EP_BWB will consider the arriving idle slot to be a reserved slot by the low priority class and will not remove the request. When this request is transmitted later on, another idle slot will be reserved and used by the queued low priority segment.

6.3.1.2 Resetting the Busy Bit: When a station receives a packet with destination the station, it will reset the busy bit for spatial reuse. The slot now becomes an idle slot and the highest priority class of the station can use it. Let us assume that this class is of priority P_i.

a) If there is a priority P_i request from downstream (RQ_CTR[i] > 0), the station will allow the idle slot to go downstream.

b) If RQ_CTR[i] = 0 but there are priority P_i segments (RG_CTR[i] > 0 or UNRG_CTR[i] > 0), the station will transmit a segment.

We point out that the station will inform all of its priority classes that the slot has been erased, and therefore it is equivalent to an unreserved slot, i.e., no downstream station has sent a request for this slot.

6.3.1.3 Resetting the Request Queues: According to EP_BWB, if a class P_i is idle and detects an unreserved idle slot, it will reset all the request queues of priority P_i at the station. However, it will not reset the request queues of lower priority P_j because these requests may now be required to serve priority P_j segments although they were originally sent to inform upstream stations about the arrival of priority P_i segments. In order to clarify this, consider the following example. Assume that there is a station with two priority classes P_h (high) and P_l (low). Class P_h has 10 requests in both REQ_QS[h,h] and REQ_QS[h,l] and class P_l has 10 requests in REQ_QS[l,l]. Let us also assume that in the first 10 slots that the station receives on the reverse channel, the request fields of both high and low priority are free. Furthermore, all the subsequent slots have their low priority request fields set. Since the low priority substation (class) sees the slots on the reverse channel first, the low priority class will transmit its low priority requests first. It is now evident that the low priority requests in REQ_QS[h,l] must be used to serve the low priority segments at this station. Therefore, after the transmission of the 10 high priority segments by the station, the high priority substation must not reset REQ_QS[h,l] to 0 when it sees an idle slot on the forward channel.

6.4 Performance Analysis

In this section, we use simulation results to investigate the throughput and delay performance of the proposed EP_BWB mechanism in the presence of two priority classes of traffic. We consider a 10 stations, 155 Mbps network with a distance between neighbor stations equal to 4 slots and a slot size equal to 53 bytes. Then, the total length of the cable will be equal to 40 slots, or equivalently, 21.8 km; a signal propagation delay of 5 μsec/km is assumed. The selected value of BWB_MOD is 2. The request field of each priority class is 2 bits and RFESs are stations S_1, S_3, S_5, S_8. In our performance analysis, we are focusing on the uniform traffic scenario. That is, every time a station generates a segment, it randomly chooses the destination of the transmission among the downstream stations. The shortest path routing rule is used for deciding the ring on which each segment will be sent. If the destination station is exactly half a ring away, 50 percent of the packets will be transmitted on one ring and the rest will be transmitted on the other ring.

First, we investigate the effect of the presence of low priority traffic on the throughput of high priority traffic. Initially, we assume that only the high priority class is active and overloaded and tries to acquire all the channel bandwidth. Our simulation results show that the EP_BWB mechanism is fair and all high priority sources acquire the same bandwidth. Then, we consider both low and high priority classes to be overloaded. Our simulation results show that the high priority class continues to acquire all channel bandwidth; i.e., the throughput of the low priority class is 0. We show this behavior of EP_BWB in Fig.6.3, where we have plotted the aggregate throughput of the high priority class when the low priority class is idle and when it is overloaded. In both cases the high priority throughput is 3.6 segments/slots.

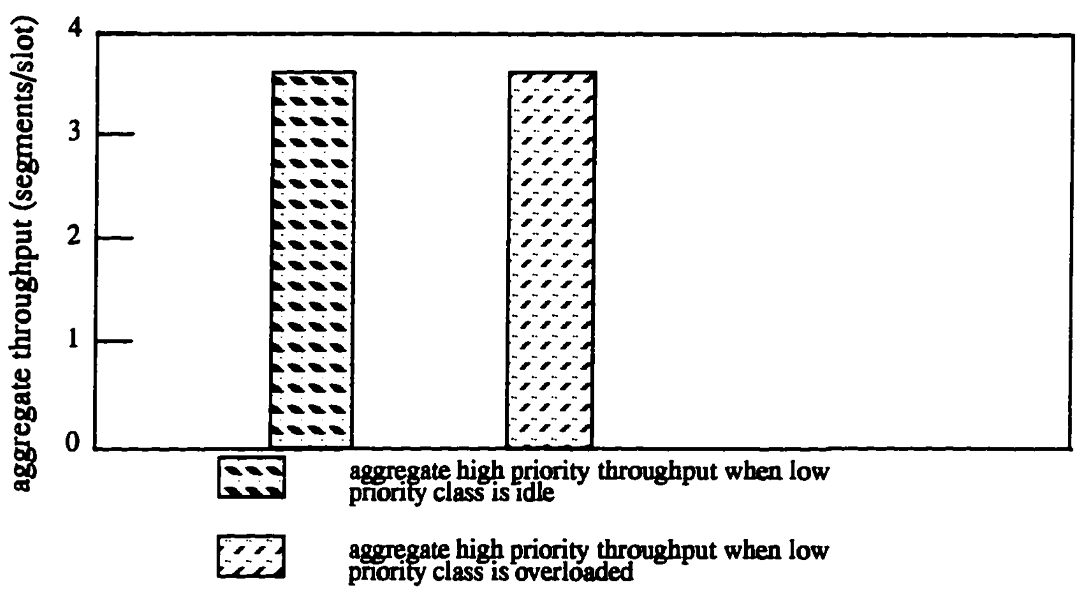

Fig.6.3 Effect of low priority on the high priority aggregate throughput

In Fig.6.4 and Fig.6.5, we consider an underloaded high priority class and show the effect of the low priority traffic on the high priority packet delay. We assume that the high priority class generates fixed size packets of 20 segments according to the Poisson distribution. We compare the high priority average packet delay in the absence of low priority traffic with its corresponding delay when the low priority traffic is overloaded. In Fig.6.4 and Fig.6.5 the aggregate high priority load is 0.8 and 0.6 of the maximum throughput

(3.6 segments/slot), respectively. In both cases the offered load is evenly distributed among the stations, i.e., we consider a constant load. Fig.6.4 and Fig.6.5 clearly show that the effect of the low priority traffic on high priority traffic is minor, which demonstrates the effectiveness of the proposed priority mechanism.

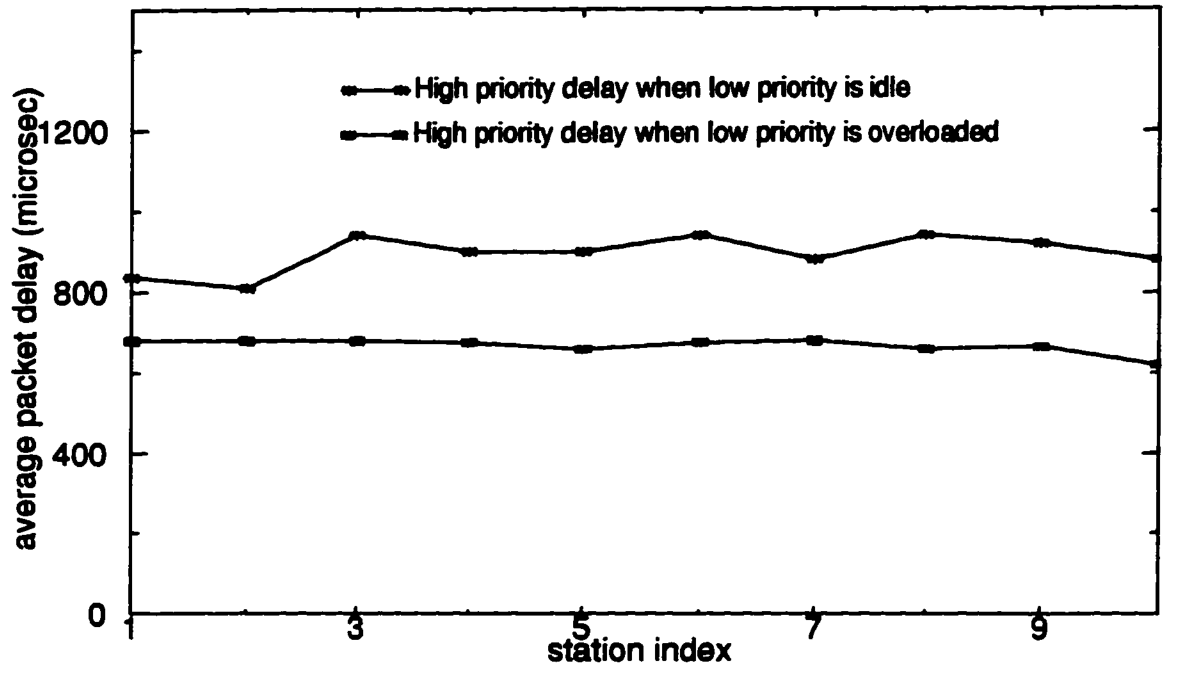

Fig.6.4 Effect of low priority traffic on the high priority traffic delay. Aggregate high priority traffic load is 0.8 of the maximum throughput (3.6 sgmt/slot)

A very interesting and rather counter intuitive behavior shown in Fig.6.4 is the observed lower delays encountered by the high priority traffic in the presence of overloaded low priority traffic; one would expect that the high priority delay would always be lower when the low priority traffic is idle. The reason for this behavior is the following. In the presence of low priority traffic, written slots by a low priority traffic source that arrive at a destination station enable this station to determine immediately that these are unreserved slots and that no requests should be sent upstream. In contrast, in the absence of low priority traffic, when an idle slot arrives at a station and the high priority class has already sent a request upstream, it has to assume that the idle slot is a reserved slot and thus misses an opportunity to remove a request from its request queue.

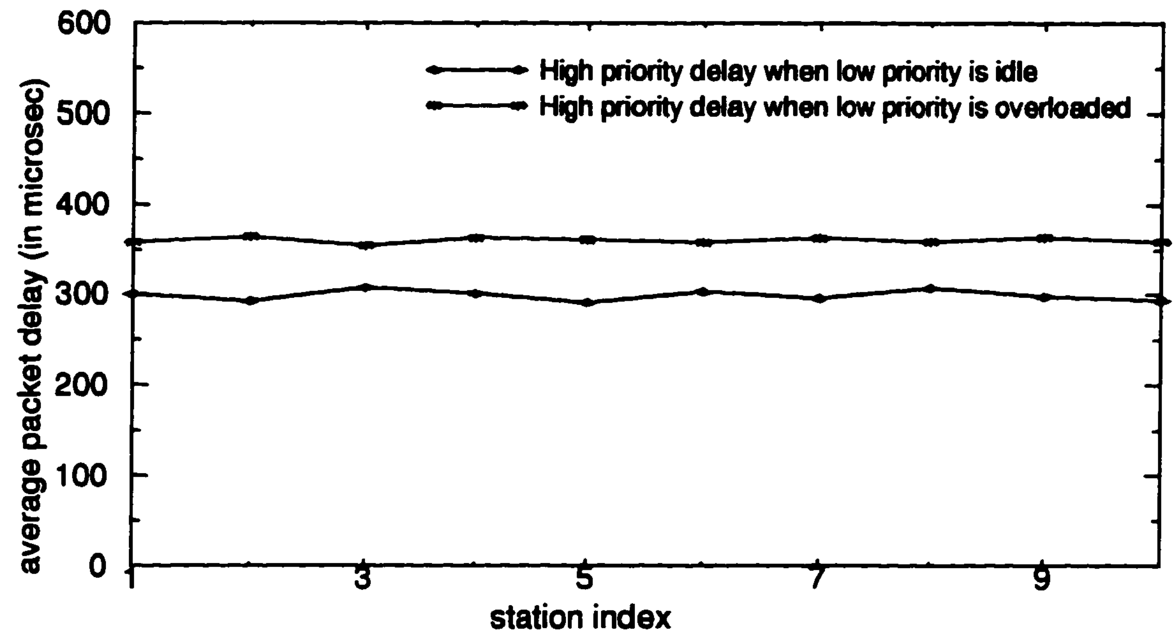

Fig.6.5 Effect of low priority traffic on the high priority traffic delay. Aggregate high priority traffic load is 0.6 of the maximum throughput (3.6 sgmt/slot)

Finally, in Fig.6.6, we compare the delays encountered by the high and low priority traffic sources in the case where the aggregate load generated by each priority class is 0.4 of the maximum throughput (3.6 segments/slot). In this case, we assume that the low priority class also generates fixed size packets of 20 segments according to the Poisson distribution. Again, the offered load of each class is evenly distributed among the stations. Fig.6.6 clearly shows that the high priority class encounters significantly lower delays. For completeness, we have also included in Fig.6.6 the high priority packet delay under an idle low priority class. Fig.6.6 shows that these delays are very similar to the ones under a

0.4 aggregate low priority load; i.e., the effect of low priority traffic on high priority traffic is again minor.

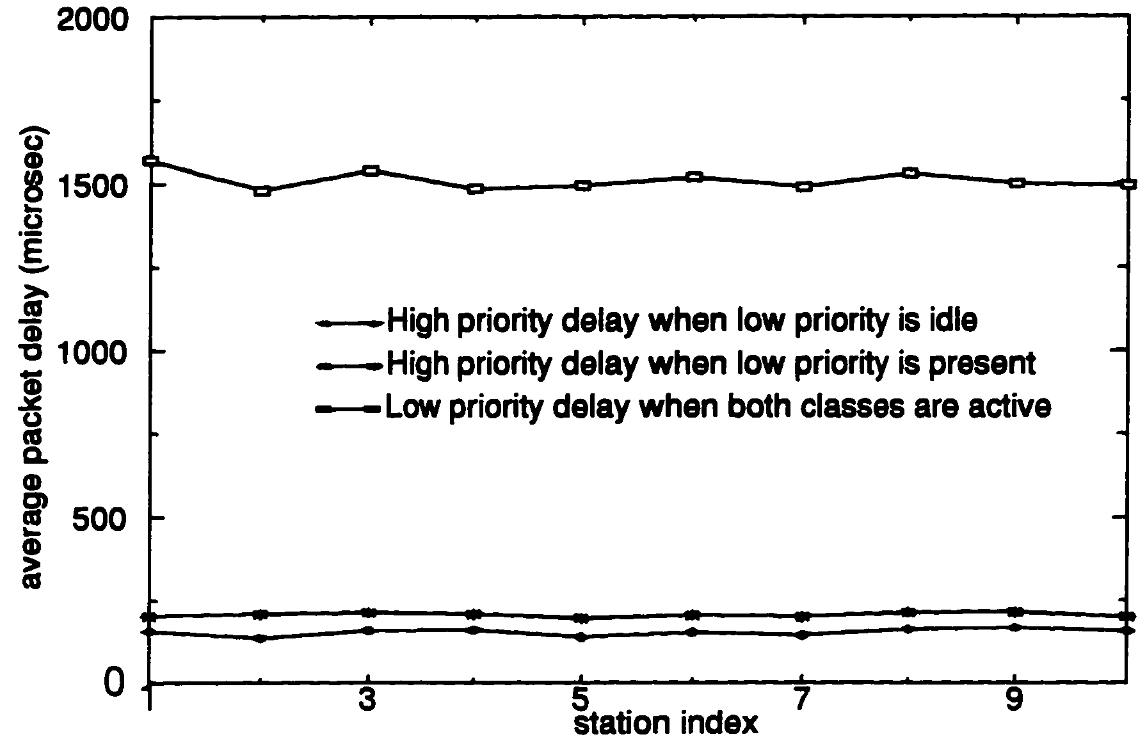

Fig.6.6 Comparison of high and low priority packet delay (20 sgmt/packet). Aggregate offered load by each of the high and low priorities is 0.4 of the maximum throughput (3.6 sgmt/slot) uniformly distributed among the stations

6.5 Conclusions

In this chapter, we have introduced the EP_BWB mechanism for dual ring architectures. EP_BWB is a very effective priority mechanism which enables high priority traffic to have almost immediate access to the transmission medium regardless of the presence of low priority traffic. We have used simulation results to investigate its throughput and delay performance in the presence of two priority classes of traffic. We have found that

EP_BWB can minimize the effect of low priority traffic on the throughput and delay performance of high priority traffic. Therefore, it has the potential of supporting real-time traffic.

CHAPTER 7

APPLICATION TO WIRELESS PERSONAL COMMUNICATION NETWORKS

7.1 Introduction

The objective of this chapter is to investigate the performance of the EP_BWB mechanism in a more real-world environment. Motivated by the current world-wide interest in Wireless Personal Communication Networks (PCN), we will investigate its applicability to the PCN environment.

Personal Communication Networks (PCN) have emerged as an important field of research activity in telecommunications [72] [71] [43] [68] [35] which is expected to continue throughout the 1990's and into the next century. The driving force behind the great research and commercial interest is the vision of providing communication services to any person, at any time, at any place, and in any form, as well as the potential revenue that these services will generate. The enabling concepts for providing universal personal communications include: terminal mobility provided by wireless access, personal mobility provided by personal numbers, and service portability.

Terminal mobility refers to the ability of a mobile terminal to access application services from any location, while in motion, as well as to the ability of the network to locate the mobile terminal as it moves. Personal mobility refers to the ability of the end user to originate and receive calls as well as access a variety of telecommunication services on any terminal in any location, and the ability of the network to identify users as they move. Personal mobility will be based on the use of a unique Personal Identification Number (PIN). Finally, service portability refers to the network capability to provide subscribed services at the terminal designated by the user. Although voice service is expected to remain a key service, a wide variety of other applications will be supported such as file transfers, electronic mail, electronic news including special services such as stock market news, video telephony, yellow pages, electronic banking, map services, etc. These services will be provided through low-power, portable personal digital assistants (palm-top computers with wireless communication technology) that can be used in home, buildings, outdoors, and in vehicles.

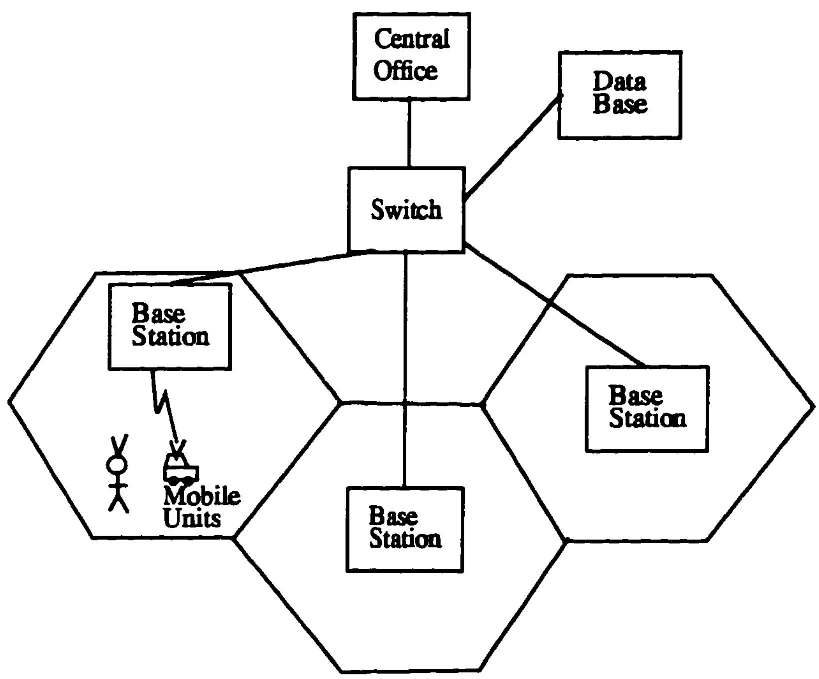

Fig.7.1 A typical cellular network

In a PCN, the covered geographical area is partitioned into a set of cells as shown in Fig.7.1. Each cell has a Base Station (BS) to exchange radio signals with wireless Mobile Units (MU). The base stations are connected to a Switch which grants user access to the wireless network and directs the flow of all information to the fixed wireline network.

Whenever a mobile unit moves from one cell to another during a call, its corresponding sessions are handed over to the new cell. This is achieved with the help of a Data Base (DB) which is internal to the switch and enables the switch to perform: a) registration of subscribers as they move through the network, b) selection of new base stations and radio frequencies for calls that require handovers. Most current networks rely on a switch based centralized control as the one shown in Fig.7.1. However, the bandwidth demand which is imposed by the multimedia services that PCNs are expected to provide requires a more efficient utilization of the available radio frequencies advocating a reduction in cell size to microcell or picocell level. While such a reduction in cell size can increase significantly the number of Mobile Units that can be supported over a geographical area, as well as lengthen battery life and improve signal quality, the frequent movement of users across cell boundaries imposes a significant burden to the network controllers. Consequently, the processing complexity needed to manage a cellular system which is made up of microcells and picocells may be excessive for a centralized control system. For this reason, packet switched architectures based on distributed network control have been proposed [36] [65] [62] [61] [40] [57] to deal with the above problem. In these architectures, the functionality of the centralized switch is partitioned into independent pieces which are distributed across a high speed Metropolitan Area Network (MAN) that provides the infrastructure for interprocessor communications. In Fig.7.2, we show such an architecture which includes the following components:

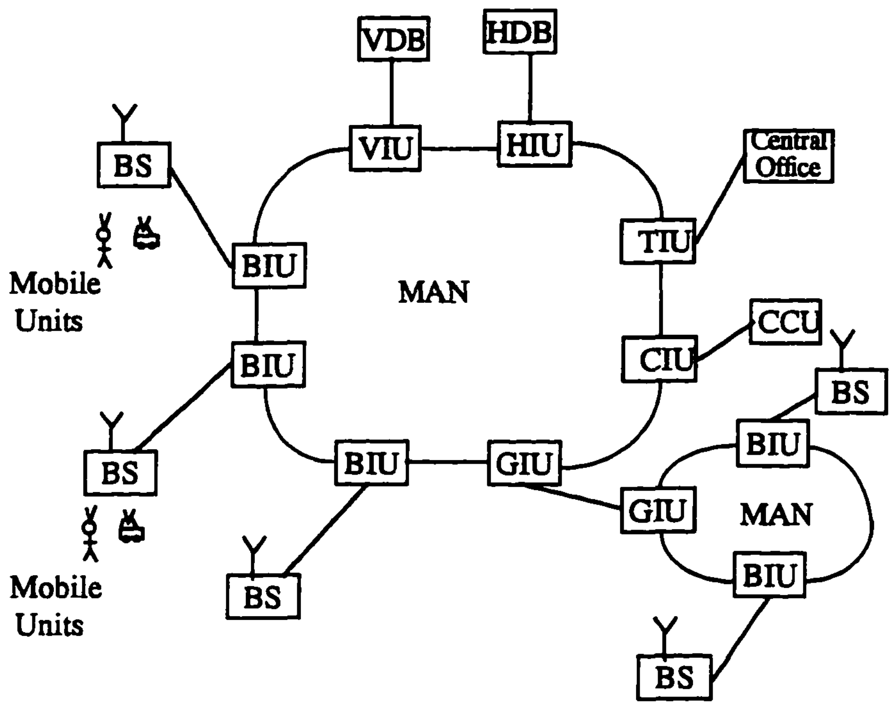

Fig.7.2 A distributed network control PCN

Base Interface Unit (BIU): It is the interface between the base station and the MAN. It provides access to the MAN and makes the conversion from the radio link protocol to MAN protocol format and vice-versa. Two types of addressing are used on the MAN. A permanent address assigned to each BIU and a Virtual Circuit Identifier (VCI) assigned to each conversation. The BIU will read the addresses of all packets as they pass by on the MAN and copy the ones which contain the permanent address of the BIU or the VCI assigned to a mobile unit using the base station.

Trunk Interface Unit (TIU): It provides the interface between the MAN and the fixed network. It converts the incoming information from the fixed network to the format of the selected radio protocol. This simplifies the protocol conversion required at the base stations. It also maintains a VCI translation table mapping MAN VCIs into public network trunk identifiers.

Cellular Controller Interface Unit (CIU): It provides the interface between the MAN and the cellular controller. The CIU is always addressed by its permanent address. The cellular controller will coordinate the allocation of VCIs to the pools of VCIs at each of the base station interface units.

Home Interface Unit (HIU): It provides access to the Home Data Base (HDB) which contains the permanent subscriber parameters of each mobile unit which was originally registered in its area of authority, which may include many cells. For a mobile unit that has currently moved to another area, it has a pointer to the appropriate Visitor Database to assist in routing incoming calls. The HIU is assigned a permanent address.

Visitor Database Interface Unit (VIU): It provides the means by which the Visitor Database (VDB) accesses the MAN. The VIU is also assigned a permanent address. The VDB contains the subscriber parameters of all foreign mobile units which are currently visiting the (multiple cells) area over which VDB has authority.

Gateway Interface Unit (GUI): It allows interconnection of MANs so that larger geographical areas can be covered.

7.2 Operation of the Distributed Control PCN

In this section, we briefly describe: a) how calls initiated by mobile or land based units can be established, b) how calls can be handed over to new base stations as a mobile unit moves from one cell to another.

7.2.1 Mobile Call Origination

A mobile wishing to initiate a call sends a *request* message to the VDB through the closest base station. The VDB finds the record for the mobile (which was retrieved by VDB from the mobile's HDB when the mobile entered the new area) and sends an *authentication* message to the BIU to determine whether the mobile will be granted service. The BIU passes this message to the mobile which returns an *authentication response*, followed by a call *setup* message containing the telephone number of the called party. The VDB verifies the *authentication response* and, then, acquires a VCI from the Cellular Controller and forwards the *setup* message, VCI, and the VDB record to the Central Office through the TIU. The Central Office sets up the connection and once it has been established, it returns a *call proceeding* message to the TIU to acknowledge the setup. The TIU creates a table entry to map the VCI to the Central Office trunk and forwards the *call proceeding* message to the BIU along with the radio VCI. The BIU adds the radio VCI to the list of VCIs which are being served by it, and forwards the message to the mobile using the source telephone number for last time. From now on, the mobile will communicate with the BIU using the radio VCI, unless a handover is needed. When one of the parties decides to hang up, a *release complete* message is sent which is picked up by the TIU which then sends a *deallocate VCI* message to the CIU. The CIU releases the VCI and sends a *deallocate VCI response* to the BIU acknowledging the deallocation of the VCI.

7.2.2 Land Based Terminal Call Origination

The call set up procedures here are similar to those above with the main difference the paging procedure used to locate the mobile. When the Central Office receives a setup call for a mobile, it uses the destination telephone number to find the HDB with which the mobile has registered and, from there, the VDB which is currently associated with it. A *page* message is sent from the HDB to the VDB which broadcasts it to all the base stations of the MAN. Upon recognizing the paging message, the mobile sends a *page response* to its VDB via the BIU. The BIU recognizes that this is a control message from the VDB and inserts the MAN address for the VDB. An authentication procedure follows between the mobile and VDB which is similar to the one of the mobile originated call case. If it is successful, the Cellular Controller allocates a Radio VCI and the VDB retrieves the subscriber record of the mobile and sends an *authentication success* message to the Central Office. The BIU and TIU then create table entries for the new VCIs. The mobile can now begin using the Radio VCI starting with a *call proceeding* message.

7.2.3 Handover

Handovers are mobile initiated. When a mobile enters a new cell and decides to switch to another base station, it sends a *handover* message to the new base station that contains the Radio VCI and the sequence number of the last message that received from the old base station. The new base station adds the new VCI to its table of VCIs and sends a delete entry message to the old base station. From this instant, the new base station can start accepting packets from the new VCI. In addition, the new base station forwards the *handover* message to TIU which sends a *handover ack* followed by the transmission of packets that may have been lost, i.e., packets with a sequence number larger than the one included in the *handover* message and which have been transmitted by the old base station. Immediately following the *handover* message, the mobile can send packets on the new path without any loss. Similar is the handover procedure in the case where the mobile moves to another area served by another VDB. In this case, some additional control messages must be transmitted so that the entry in the old VDB is deleted and a new entry in the new VDB is added.

7.3 Voice/Data Performance in the PCN Environment

In this section, we investigate the ability of a dual ring network, running the EP_BWB protocol, to support the interconnection of the base stations in a wireless PCN under voice/data transmissions. We assign higher priority to voice traffic and assume that in addition to EP_BWB there is a Bandwidth Manager (BWM) on the MAN which reserves 20% of the channel bandwidth for call control signalling, handoff management signalling, etc., and allocates 80% of the channel bandwidth to voice/data transmissions. That is, if 155 Mbps is the channel bandwidth then the allocated bandwidth to voice/data traffic $BW_{v/d}$ will be 124 Mbps. We consider the voice conversation model of [10] where each voice source alternates between talkspurt and silent periods which are exponentially distributed with means of 1.5 sec and 2.25 sec respectively, i.e., the voice activity factor a_c is 0.4. We assume 64 Kbps PCM encoding for voice and that only talkspurts are packetized and transmitted over the network. The voice packetization interval VPI is 5.5 msec so that a voice source in talkspurt can fill up the 44 bytes of the segmentation unit field of the slot (see Fig. 6.1). The slots filled by a particular voice source are identified by the VCI value given by BWM during the call set up stage. We also assume that a single buffer is assigned to each voice source which can store only one voice packet. If this packet has not been transmitted by the time the next voice packet (from the same voice source) has been generated (after 5.5 msec), the new packet will overwrite the old packet in the buffer; i.e., the old packet will be discarded. As performance criterion we use the percentage of discarded voice packets which should not exceed 1%.

A $BW_{v/d}$ bandwidth provides $S_{v/d} = (BW_{v/d} \bullet VPI) / (53 \bullet 8)$ slots during a VPI time interval. Since voice has (through EP_BWB) preemptive priority over data, an estimate of the maximum number of voice users $N_{v,max}$ that can be supported by the system can be derived by the following inequality:

$$\frac{\sum_{j=S_{v/d}+1}^{N_{v,max}} (j - S_{v/d}) \binom{N_{v,max}}{j} a_c^j (1-a_c)^{N_{v,max}-j}}{\sum_{j=1}^{N_{v,max}} \binom{N_{v,max}}{j} a_c^j (1-a_c)^{N_{v,max}-j}} \le 0.01 \qquad (7.1)$$

In equation (7.1), the numerator provides the average number of discarded voice packets during one VPI and the denominator provides the average number of generated voice packets during one VPI. Since $a_c = 0.4$, $N_{v,max}$ can be as large as 2.5 $S_{v/d}$. Simulation results we have run have shown that (7.1) is satisfied even when the number of active voice sources $N_{v,max}$ is very close to 2.5 $S_{v/d}$. As a quick, and rather pessimistic approximation, of $N_{v,max}$ we will consider $N_{v,max} = 2.4\ S_{v/d}$. We should keep in mind that the preemptive priority capabilities of voice will prevent the transmission of data in the case of a large number of active voice users. Therefore, if it is desirable to guarantee some average bandwidth BW_d for data, a smaller (than $N_{v,max}$) number of voice calls must be accepted by BWB into the system; we should notice that the number of voice users in one ring is also the number of voice calls in the system. Let E_r be the total voice traffic in Erlangs that should be allowed into the system so that an average bandwidth BW_d becomes available for data traffic. The value of E_r must satisfy the following two conditions: a) the average bandwidth for voice $(E_r \bullet 53 \bullet 8 \bullet 0.4)/VPI$ must be less than or equal to ($BW_{v/d} - BW_d$), b) the voice packet discarding probability must be less than 0.01 or, approximately, E_r must be less than $2.4\ S_{v/d} = (2.4 \bullet (BW_{v/d} \bullet VPI))/(53 \bullet 8)$. By combining "a)" and "b)", we get the following expression:

$$E_r = \frac{VPI}{(53 \bullet 8 \bullet 0.4)}\left(min(0.4 \bullet 2.4 \bullet BW_{v/d}, BW_{v/d} - BW_d) \right) \quad (7.2)$$

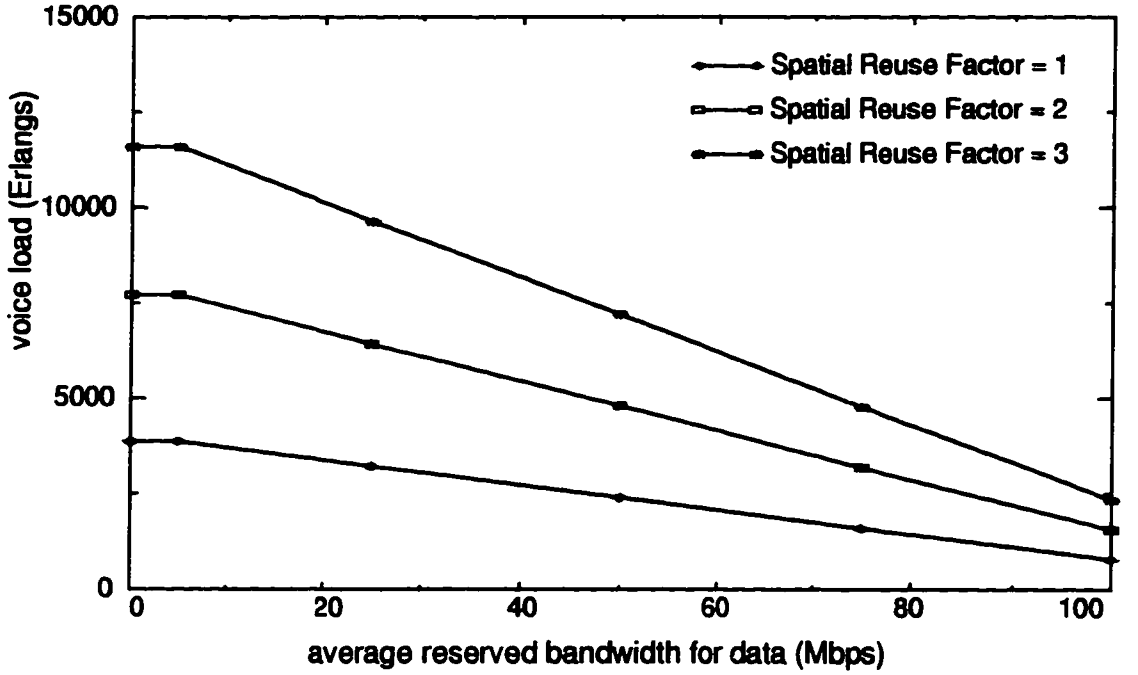

Fig.7.3 Voice versus Data load. 155 Mbps channel bandwidth 80% of which is allocated to voice/data transmissions

In Fig.7.3, we have used equation (7.2) to show the effect of the average reserved bandwidth for data (per ring) on the corresponding voice traffic load in Erlangs for different values of the Spatial Reuse Factor (SRF). SRF = j indicates that a slot, due to the locality of traffic, carries on the average j different (voice or data) segments per rotation around the ring. We see that as the traffic locality increases, the system can support significantly higher loads. In order to gain a better understanding of the presented values in Fig.7.3, we consider the downtown metropolitan area of the PCN system in [63] which consists of rectangular city blocks and for which it is assumed that: a) the two pavements of each street are 300m, b) pedestrians are spaced 1.5 m apart during the busy hour, c) 75% of the pedestrians have a portable phone, d) there are 1000 people per block inside buildings, of which 80% have a phone, e) each subscriber averages 2 calls/busy hour with mean duration of 1 minute. Under these conditions, the offered traffic per city block (including the streets) will be 36.6 Erlangs. Then, Fig.7.3 can provide estimates for the number of city blocks that the system can support by dividing its Erlang values by 36.6. For instance, in the case where SRF = 1 and the average reserved bandwidth for data is less than 10Mbps, the MAN could support the traffic of 105 blocks. If SRF increases to 3, the number of supported city blocks will increase to 315. Notice, that in the above examples we have considered a 155 Mbps channel bandwidth. Had we considered an 1 Gbps channel (i.e., the bandwidth of Metaring [18]), both the supported voice load and number of city blocks would have increased by a factor of about 6.5.

7.4 Conclusions

In this chapter, we have investigated the potential of a dual ring MAN, employing spatial reuse and running the EP_BWB mechanism, to interconnect microcellular base sites in a wireless PCN. A significant advantage of the dual ring topology for such an environment is that it facilitates, due to its shared medium, the transmission of control information and multicasting. In our investigation, we have considered the performance of the system in the presence of voice and data traffic. Our analysis clearly shows that dual ring network architectures employing the EP_BWB mechanism are very appropriate for the interconnection of the base stations in a wireless PCN because they can support the traffic requirements of a large number of microcellular base sites.

CHAPTER 8

CONCLUDING REMARKS

In this dissertation, we have proposed various channel access mechanisms that can improve fairness and introduce bandwidth efficient transmission on large high speed dual bus and dual ring network architectures. In addition, we have introduced effective priority mechanisms that can meet the diverse throughput and delay requirements of the wide variety of applications that the high speed networks of the future will support.

In Chapter 2, we have reviewed the existing MAC mechanisms that have been proposed for high speed dual bus and dual ring networks. The limitations of these mechanisms, along with the importance of supporting a wide variety of services over high speed MANs, have motivated our research interest in this area. In Chapter 3, we have proposed the Buffer Insertion Bandwidth Balancing (BI_BWB) mechanism for dual bus architectures that can significantly improve the downstream stations' delay performance. BI_BWB achieves that by enabling stations to delay, inside a shift register, incoming written slots replacing them with their own transmissions. In this way, a downstream station can have immediate access to the channel at the cost of increasing the bus latency by one slot. The downstream station can then decrease the bus latency when an idle slot, reserved by this station, arrives on the forward channel. This idle slot is reserved by a request the downstream station sends upstream at the instant it increases the bus latency. We have investigated the performance of BI_BWB and we have compared it with that of the most efficient existing access mechanisms. Our investigation has clearly shown that BI_BWB can increase the convergence speed to the steady state where bandwidth balancing is achieved, enables downstream stations to have a faster access to the channel, and allows them to have control over their access delays. We have also introduced a queuing analytic model for BI_BWB which can capture the interaction of the request bits, busy bits, TAR bits, and insertion buffers among the stations and provide good estimates for the stations' average segment delay.

In Chapter 4, we have introduced the Preemptive priority Buffer Insertion Bandwidth Balancing (P_BI_BWB) mechanism. P_BI_BWB introduces very effective priorities into the system by allowing stations to use their shift registers and to remove the low priority traffic from the channel until the transmission of the high priority traffic is complete. In addition, each class not only counts the requests of its own priority, but also counts the requests of higher priority classes. As a result, the effects of lower priority class on the higher priority class are further eliminated. Our investigation of P_BI_BWB has shown that it can fairly distribute the available bandwidth among traffic sources of the same priority, provide higher priority users with better performance characteristics, and minimize the effect of station location on performance. In addition, its operation does not require the wastage of channel slots.

In Chapter 5, we have introduced the Fair bandwidth Allocation Mechanism (FBAM) for dual ring architectures. This mechanism enables stations to make bandwidth reservations on a continuous basis instead of waiting until bandwidth starvation is observed. Thus, it does not encounter the difficult task of determining when bandwidth starvation begins, which is a serious drawback of previously proposed access algorithms such as the Local and Global Fairness Algorithms, LFA and GFA, respectively. In addition, its operation requires a much smaller number of control bits in the access control field of the slot. We have investigated the performance of FBAM under various traffic scenarios and we have found that it can minimize the interaction between ring sections containing stations with no overlapping transmission paths, thus maximizing the aggregate network throughput. In addition, it can evenly distribute the channel bandwidth among the competing stations of each ring section. Our investigation has shown that FBAM, which is a distributed access mechanism, can achieve in most cases a MAX-MIN throughput fairness, which is considered to be an optimal throughput fairness. Furthermore, in the cases where the throughputs provided by FBAM and the MAX-MIN fairness algorithm differ, the observed discrepancy is very small. This is a significant advantage of FBAM over all the other previously proposed mechanisms.

In Chapter 6, we have introduced the Effective Priority Bandwidth Balancing (EP_BWB) mechanism for dual ring architectures. The operation of EP_BWB is based on FBAM and enables high priority traffic to preempt the low priority traffic transmissions. We have investigated the performance of EP_BWB in the presence of two priority classes of traffic and we have found that it can minimize the effect of the low priority traffic on the throughput and delay performance of the high priority traffic. Thus, it can effectively support real-time traffic.

Finally, in Chapter 7, we have investigated the EP_BWB mechanism in a more real-world environment. Motivated by the current world-wide interest in wireless Personal Communication Networks (PCN), we have investigated the potential of a dual ring network employing spatial reuse, under the EP_BWB mechanism, to support the interconnection of the base stations on a distributed control wireless PCN carrying voice and data traffic. Our analysis has shown that EP_BWB can indeed provide such an interconnection supporting the traffic of a large number of microcellular base sites.

8.1 Future Work

The main objective in the area of high speed medium access control protocols is to propose and investigate fair and bandwidth efficient control mechanisms that are appropriate for supporting applications with very diverse traffic characteristics, throughput, and delay requirements. The introduction of the insertion buffer that enables downstream stations to have immediate access to the channel, as well as the transmission of multiple low priority requests by high priority traffic sources that can preempt the transmissions of upstream low priority sources, are two very significant and effective steps in this direction. They can serve as the basis for future research.

For instance, in the case of dual bus architectures, we have proposed the P_BI_BWB mechanism which allows a downstream high priority substation to store low priority segments. In addition, whenever this substation transmits, it also sends a high or lower priority request upstream, depending on whether the transmitted high priority segment is registered or unregistered. In this way, sufficient idle slots are reserved to guarantee the retransmission of the buffered segments later on. It would be very interesting to combine the buffer insertion capability of the P_BI_BWB mechanism with the multiple request transmission capability of the EP_BWB mechanism and investigate the corresponding effect on performance. It is not expected that the resulting priority mechanism will be more effective than P_BI_BWB. It is expected, however, that the size of the Lower Priority Segment (LPS) buffer will decrease. It will also be very interesting to investigate the performance of a priority mechanism that uses both LPS buffers and multipriority request transmissions on dual ring networks. It is expected that this mechanism will be more effective than EP_BWB.

In Chapter 3, we have provided an analytic queuing model for BI_BWB in the case of independent segment transmissions. Extensions of this work may include the introduction of queueing models for the P_BI_BWB, FBAM, and EP_BWB mechanisms.

Congestion control on high speed networks has been an active research area for the past few years. Its objective is to prevent network overloading that may lead to a high delay or even deadlock. Congestion control mechanisms must operate on top or in conjunction with the MAC mechanisms and try to satisfy the quality of service required by the end users. This is very important since MAC mechanisms alone cannot guarantee the throughput and delay requirements of the end users. For instance, P_BI_BWB is the most effective priority mechanism ever proposed for dual bus networks. It can eliminate the effect of low priority traffic on high priority traffic. However, this mechanism will not be able to meet the delay requirements of high priority time-critical traffic if the aggregate load generated by this traffic is greater than the channel bandwidth. Thus, the need of congestion control becomes evident. A congestion control mechanism, called the Guaranteed Bandwidth (GBW) mechanism [11], [64], has been recently proposed for dual bus architectures. Its operation assigns a *cost* to each free slot of the channel and guarantees the bandwidth each station is willing to pay for. A drawback of GBW is that a station can only write on the paid slots even when there is no other active station in the network. Therefore, it will be extremely interesting to investigate extensions of the GBW mechanism which will enable a station to transmit on unreserved (unpaid) slots. This will provide a greater flexibility in allocating bandwidth to the different network sessions and in lowering the connection cost through multiplexing. For instance, consider the case of an 155 Mbps dual bus network that supports high priority, delay sensitive applications with peak and average bandwidth requirements of 7.75 Mbps and 2 Mbps, respectively. If the GBW mechanism is used, then 7.75 Mbps must be allocated to each application, the network will be able to support only 20 simultaneous sessions, and the cost of each connection will be significant. A combination of GBW and P_BI_BWB mechanism, which allows a station to reserve 3 Mbps and compete for the remaining 4.75 Mbps, may allow the network to support a much larger number of connections while satisfying each application's throughput and delay requirements. Consequently, the corresponding network connection cost may be significantly reduced. Such an approach can be extremely useful, for instance, in the transmission of compressed video using a hierarchical coding technique [31], [47]. Hierarchical coding, also known as layered coding, divides a signal into subsignals of various importance to be coded and transmitted separately. A station can then use the guaranteed bandwidth to carry the bit stream that contains all the important structural information in the image, and compete for additional bandwidth in order to transmit the less important information which will add the necessary quality finish.

REFERENCES

1. O. Aboul-Magd, "H. Gilbert, and M. Wernik. Flow and congestion control for broadband packet networks," in *Proc. 13th International Teletraffic Conference*, pp. 853-858,1991.

2. *Fiber Distributed Data Interface (FDDI) - Token ring medium access control (MAC)*, ANSI, REV-10, Feb. 1986

3. A. Baiocchi el at., "Achieving statistical gain in atm networks with the same complexity as peak allocation strategy," in *Proc. INFOCOM'94*, pp.374- 382, 1994.

4. A. W. Berger et al., "Performance characterizations of traffic monitoring, and associated control, mechanisms for broadband packet networks," in *IEEE Globecom*, pp. 350-354, 1990.

5. A. W. Berger et al., "Performance analysis of a rate-control throttle where tokens and jobs queue," *IEEE Journal Select. Areas in Commun.* vol 9, no. 2, pp. 165-170, Feb. 1991.

6. D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc. Englewood Cliffs, NJ, 1987.

7. C. Bisdikian and A. Tantawy, "A mechanism for implementing preemptive priorities in DQDB subnetworks," *in Proc., IEEE ICC'91*, Denver, CO,pp. 1062-1067, June 1991.

8. A.D. Bovopoulos, "Performance evaluation of a traffic control mechanism for atm networks," *IEEE INFOCOM*, pp. 469-478, 1992.

9. P.E. Boyer and J.W. Roberts, "Traffic control for stepwise VBR connections in an atm network," *Integrated Broadband Commun.: Reviews from RACE*, pp. 391-398, 1991.

10. D. T. Brandy, "A Statistical Analysis of On-Off Patterns in 16 Conversations," *Bell Systems Tech. Journal*, vol. 47, January 1968.

11. Z.L. Budrikis et. al, "QPSX: A queue packet and synchronous circuit exchange," in *Proc. 8th Int. Conf. Comput. Comm. ICCC'86*, Munich, West Germany, pp. 288-293, Sept. 15-19, 1986.

12. Z.L. Budrikis, "Working Document for IEEE 802.6 CO Services Standard," *Contribution to IEEE 802.6 Working Group*, March 1992.

13. L. Campbell, P. G. Potter and M. Zukerman, "Fairness under sustained overloaded in a DQDB subnetwork with erasure nodes," *in Proc., IEEE ICC'91*, Denver, CO, pp. 1038-1042, June 1991.

14. V. Catania et. al, "Throughput Analysis of DQDB in overloaded Condition," in *Proc., IEEE ICC'91*, Denver, CO,pp. 741-747, June 1991.

15. J. H. S. Chan and D. H.K. Tsang, "Bandwidth allocation of multiple QOS classes in ATM environment," in *Proc. IEEE INFOCOM*, pp. 360-367, 1994

16. J. Chen, I.Cidon and Y. Ofek, "A local fairness algorithm for gigabit LAN's/MAN's with spatial reuse", *IEEE Journal Select. Areas in Commun.*, vol. 11, no. 8, pp. 1183-1192, Oct. 1993

17. R. Chipalkatti, J. Kurose, and D. Towsl. "Scheduling policies for real-time and non-real-time traffic in a statistical multiplexer," in *Proc. IEEE INFOCOM*, pp. 774-783, 1989.

18. I. Cidon and Y. Ofek, "Metaring-A full duplex ring with fairness and spatial reuse," *IEEE Tran. Commun.*. vol. 41, no. 1, pp. 110-120, 1993.

19. M.Conti et al., "A methodological approach to an extensive analysis of DQDB performance and fairness, "*IEEE Journal Select. Areas in Communication*, vol.SAC- 9, no.1, pp. 76-87, Jan. 1991.

20. D. Davids and T. Welzel, "Performance analysis of DQDB based on simulation," *in 3rd IEEE Workshop on Metropolitan Area Networks*, San Diego, CA, pp. 7.6.1-7.6.15, March 28-30, 1989.

21. M. Decia et al., "Bandwidth assignment and virtual call blocking in atm networks," in *Proc. IEEE ICC*, 318.6.1-318.6.8, 1990.

22. B. Doshi and H. Nguyen. "Congestion control in isdn frame-relay networks," *AT&T Technical Journal*, pp. 67:35-46, 1988.

23. B. Doshi, "Performance of in-call buffer/window allocation schemes for short intermittent file transfers over broadband packet networks," *IEEE INFOCOM*, pp. 2463-2471, 1992.

24. B. Doshi, S. Dravida, and G. Ramamurthy, "Memory, bandwidth, processing and fairness considerations in real time congestion control for broadband networks," in *Proc. 13th International Teletraffic Conference*, pp. 143-149, 1991.

25. B. Doshi and H. Heffes, "Performance of an adaptive buffer/window allocation scheme for long file transfers over wide area high speed packet networks," *IEEE Journal Select. Areas in Commun.*, vol. 9, no. 3, pp. 325-334, April 1991.

26. A.E. Eckberg, D.T. Luan, and D.M. Lucantoni, "Bandwidth management: A congestion control strategy for broadband packet networks - characterizing the throughput-burstiness filter," in *Int. Teletraffic Cong. Specialist Sem.*, Adelaide, Australia, Sept. 1989.

27. A. Eckberg, D. Lian and D. Lucantoni, "An approach to controlling congestion in atm networks," *International Journal on of Digital and Analog Commun. Sys.*, vol 15, pp. 277-288, 1990.

28. K.W. Fendwich et al., "An approach to high-performance, high-speed data networks," *IEEE Commun.Magazine*, pp. 74-82, Oct. 1991.

29. R.G. M. Falconer and J. L. Adams, "Orwell: A protocol for an integrated services local network," *Brit. Telecom Technol. Journal*, vol. 3, no. 4, pp. 27-35, Oct. 1985.

30. D. Ferguson, Y. Hymenium, and C. Nikolaou, "Microeconomic algorithms for load balancing in distributed computer systems," in *Proc. 8th International Conference on Distributed Computing Systems*, June 1988.

31. M. Ghanbari, "Two-Layer coding of video signals for VBR networks," *Journal Select. Areas in Commun.*, vol. 7, no. 5, pp. 771-781, June 1989.

32. S. J. Golestani, "Congestion-free transmission of real-time traffic in packet networks," in *IEEE INFOCOM*, pp. 527-536, June 1990.

33. S.J. Golestani, "Congestion-free communication in high-speed packet networks," *IEEE Transactions on Commun.*, vol. 39, no. 12, pp. 1802-1812, Dec. 1991.

34. S.J. Golestani, "A framing strategy for congestion management," *IEEE Transactions on Commun.*, vol. 9, no. 7, pp. 1064-1077, Sept. 1991.

35. D.J. Goodman, "Cellular packet communications," *IEEE Transactions on Communications*, pp. 1272-1280, August 1991.

36. D.J. Goodman, "Network control for wireless communications", *IEEE Commun. Magazine*, vol. 30, no. 12, pp. 116-124, December 1992.

37. E. L. Hahne et al., "Improving the fairness of distributed queue dual bus network," in *Proc. INFOCOM'90, San Francisco*, CA, pp.175-184, June 1990.

38. E.L. Hahne and N. F. Maxemchuk, "Fair access of multi-priority traffic to Distributed-Queue Dual-Bus Networks," in *Proc., IEEE INFOCOM'91*, Bal Harbour, FL, pp. 889-900, Apr. 1991.

39. D. S. Holtsinger, "Congestion control mechanisms for atm networks," *Asynchronous Transfer Mode Networks*, pp.107-121, 1993.

40. Nen-Fu Huang, Ko-Shung Chen, "A distributed paths migration scheme for IEEE 802.6 based personal communication networks", *IEEE Journal Select. Areas in Commun.*, Vol. 12, No 3, pp. 1415-1425, October 1994.

41. E.L. Hahne and K.J. Han, "Fair Access of Multi-Priority Traffic in DQDB," in *Proc. INFOCOM'91, Bal Harbour*, FL, pp.889-900, April 1991.

42. I. J. Hyun and K. J. Han, "Dynamic Bandwidth Balancing Mechanism for improving DQDB performance," in *Proc., IEEE ICC'91*, pp. 1345-1349, Denver, CO, June 1991.

43. B. Jabbari, et.al., "Network issues for wireless communications", *IEEE Commun. Magazine*, Vol. 33, No 1, pp. 28-42, January 1995.

44. J.M. Jaffe, "Bottleneck Flow Control", *IEEE Transactions on Commun*, COM 29(7), pp. 954-962, July 1981

45. D. G. Jeon, G. T. Kim, and K. J. Han, "Effectiveness of priority schemes of the DQDB and the FDDI," in *Proc., IEE ICC'91*, pp. 1340-1344, Dever, CO, June 1991.

46. M. J. Johnson. "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Comput. Networks ISDN Syst.*, vol. 11, pp. 121-131, 1986.

47. G. Karlsson, M. Vetterli, "Packet video and its integration into the network architecture," *Journal Select. Areas in Commun.*, vol. 7, no. 5, pp. 739-751, June 1989.

48. D. Karvelas and M. Papamichail, "No Slot Wasting Bandwidth Balancing with Immediate Use of TAR Bit," in *Proc. INFOCOM'94*, session 6b, Toronto, Canada, pp. 12-14, June 1994.

49. D. Karvelas and M. Papamichail, "The No Slot Wasting Bandwidth Balancing Mechanism for Dual Bus Architectures," *IEEE Journal Select. Areas in Commun.*, vol. 11, no.8, pp. 1214-1228, Oct. 1993.

50. D. Karvelas and M. Papamichail, "An effective priority mechanism for high speed MANs," in *Proc. INFOCOM'94*, session 6b, Toronto, Canada, pp. 742-749, June 1994.

51. D. Karvelas and M. Papamichail, "The guaranteed bandwidth and balancing mechanism for high speed MANs," in *Proc. IEEE INFOCOM*, pp. 696-700, 1994

52. S. Y. Kim and K. J. Han, "An enhanced bandwidth balancing mechanism of the DQDB MAN," in *Proc., IEEE ICC'91*, pp. 423-427, Denver, CO, June 1991.

53. V. Kulkarni, L. Gun, P. Chimento, "Effective Bandwidth Vector for Two- Priority ATM traffic," in *Proc. IEEE INFOCOM*, pp. 1056-1063, 1994

54. J. F. Kurose and R. Simha, "A microeconomic approach to optimal resource allocation in distributed computer system," *IEEE Transactions on Computers*, vol. 38, no. 5, May 1989.

55. A.A. Lazar, A.T. Temple, and R. Gidron, "MAGNET II: A metropolitan area network based on asynchronous time sharing," *IEEE Journal Select. Areas in Commun.*, vol. 8, no. 8, pp. 1582-1594.

56. W.E. Leland. "Window-based congestion management in broadband atm networks, "in *IEEE Globecom*, pp. 1794-1800, 1989.

57. V.C.M. Leung, et.al., "Call control and traffic transport for connection oriented high speed wireless personal communications over MANs", *IEEE Journal Select. Areas in Commun.*, Vol. 12, No 3, pp. 1376-1388, October 1994.

58. J.O. Limb and L. E. Flamm, "A distributed local area network packet protocol for combined voice and data transmission," *IEEE Journal Select. Areas in Commun.*, vol SAC-1, no 5, Nov. 1083.

59. S. Low and P. Varaiya, "An algorithm for optimal service provisioning using resource pricing," in *Proc. INFOCOM'94*, pp. 368-373,1994.

60. D.T. Luan and D.M. Lucantoni, "The effect of bandwidth management in broadband atm networks," *Performance evaluation review*,vol.20, no. 1, pp. 128-139, June 1992.

61. A.D. Malyan, et.al., "Network architecture and signaling for wireless personal communications", *IEEE Journal Select. Areas in Commun.*, Vol. 11, No 6, pp. 830-841, August 1993.

62. A.D. Malyan, et.al. "A personal communication network architecture using the IEEE 802.6 MAN", *Proceed. of ICC'92*, Chicago, IL, pp. 1247-1251, May 1992.

63. A.D. Malyan, et.al. "Network architecture and signaling for wireless personal communications", *IEEE Journal Select. Areas in Commun.*, vol. 11, No. 6, pp. 830-841, August 1993.

64. P. Martini, G. Werschmann, "Real-time communication in DQDB: a comparison of different strategies," in *Proc. 17th Conference on Local Computer Networks*, Minneapolis, Minnesota, pp. 446-453, Sept. 1991.

65. K.S. Meier-Hellstern, et.al., "A wireless service for the IEEE 802.6 MAN", *WINLAB-TR-14*, April 1991.

66. A. Mukherjee, L. H. Landweber, and T. Faber. "Dynamic time windows and generalized virtual clock: combined closed-loop/open-loop congestion control," *IEEE INFICOM*, pp. 322-332, 1992.

67. B. Mukherjee, C. Bisdikian "A Journey through the DQDB network literature," *IBM Research Report RC 17016(#75478)* July, 1991.

68. M. Naghshineh, M. Schwartz, A. Acampora, "Issues in wireless access broadband networks", *IBM Research Report 19980*, Nov. 1994.

69. H. Ohnishi, N. Morita, and S. Suzuki, "ATM ring protocol and performance, "in *Proc. ICC'89*, pp. 394-398,1989.

70. H. Ohnishi, T. Okada, and K. Noguchi, "Flow control schemes and delay/loss tradeoff in atm networks," *IEEE Journal Select. Areas in Commun.* vol. 6, no. 9, pp. 1609-1616, Dec. 1988.

71. J.E. Padgett, et.al., "Overview of wireless personal communications", *IEEE Commun. Magazine*, Vol. 33, No 1, pp. 28-42, January 1995.

72. R. Pandya, "Emerging mobile and personal communication systems", *IEEE Commun. Magazine*, Vol. 33, No 6, pp. 44-54, June 1995.

73. D. Peter and V. Frost. "Optimal packet discarding: an ATM-oriented analysis model and initial results," in *Proc. IEEE INFOCOMM*, pp. 537-542, 1990

74. M. Papamichail, "Medium access control mechanisms for high speed metropolitan area networks," Ph. D thesis, CIS Dept., NJIT, Oct. 1993.

75. S. Pingali, D. Tipper, and J. Hammond. "The performance of adaptive window flow controls in a dynamic load environment," in *IEEE INFOCOM, pp.* 55-62, 1990.

76. K.K. Ramakrishnan. "A binary feedback scheme for congestion avoidance in computer networks with a connectionless network layer," in *Proc. SIGCOMM*, pp. 303-313, Aug. 1988.

77. G. Ramakrishnan and R.S. Dighe, "Distributed source control: a network access control for integrated broadband packet networks," *IEEE Journal Select. Areas in Commun.* vol. 9, no. 7, pp. 990-1001, Sept. 1991.

78. E.P. Rathgeb. "Modeling and performance comparison of policing mechanisms for atm networks," *IEEE Journal Select. Areas in Commun.*, vol.9, no. 3, pp. 325-334, April 1991.

79. F.E.Ross "FDDI-A tutorial," *IEEE communications magazine*, May 1986.

80. F.E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE Journal Select. Areas in Commun.*, vol. 7, no. 7, pp. 1043-1051, Sept. 1989.

81. F.E. Ross, "Rings are 'round for good'," *IEEE Network Magazine*, 1987.

82. M. Sidi et al., "Congestion control through input rate regulation," *IEEE Globecom*, pp. 49.2.1-49.2.5, 1989.

83. J. Spragins, "Loop Transmission Systems-Mean Value Analysis," *Transaction on Communications*, pp. 592-602, June 1972.

84. K. Sriram. "Dynamic bandwidth allocation and congestion control schemes for voice and data multiplexing in wideband packet technology," in *Proc. IEEE ICCC*, pp. 324.3.2-324.3.7, 1990.

85. IEEE Standards for Local and Metropolitan Area Networks: Distributed Queue Dual Bus (DQDB) subnetwork of a metropolitan area network, IEEE Std. 802.6-1990, July 1991.

86. A.S. Tanenbaum, *Computer Networks*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

87. F. Tobagi, F. Borgonovo, and L. Fratta, "EXPRESSNET: A high performance integrated services local area network," *IEEE Journal Select. Areas in Commnu.*, vol SAC-a, No 5, Nov. 1983.

88. J. Turener. "New directions in communications (or which way to the information age?)," *IEEE Communications magazine*, pp. 38-45, Oct. 1986.

89. H. R. van As, "Performance Evaluations of the Bandwidth Balancing in the DQDB MAC protocol," *8th Annual EFOC/LAN Conference*, Munich, Germany, pp. 231-239, June 27-29, 1990.

90. H.R. van As, "Major performance characteristics of the DQDB MAC protocol," *SBT/IEEE Intl. Telecommunications Symposium(ITS'90)*, Rio de Janeiro, Brazil, pp. 113-119, Sept. 3-6, 1990.

91. R. J. Vetter, "ATM concepts, architectures, and protocols," *Commui. of the ACM*, vol. 38, no. 2, pp. 31-44, Feb. 1995.

92. A. M. Viterbi, "Approximate Analysis of Time-Syncronous Packet Networks," *IEEE Journal Select. Areas in Commun.*, pp. 879-890, Sept. 1986.

93. J. W. Wong, "Throughput of DQDB networks under heavy load," in *Proc. 7th Europ. Fiber Optic Commun. Local Area Network. Expo.*, Amsterdam, The Netherlands, June 1989, pp. 146-151.

94. H. Wu, Y. Ofek and K. Sohraby, "Integration of synchronous and asynchronous traffic on the metaring architecture and its analysis," in *Proc. ICC'92*, pp. 147-153, 1992