Performance and Reliability of High Speed Integrated Services Metropolitan Networks

Michel Kadoch

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

September, 1991

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

ISBN 0-315-73665-8

ABSTRACT

Performance and Reliability of High Speed Integrated Services Metropolitan Networks

Michel Kadoch, Ph. D.,
Concordia University, 1991

We introduce new access schemes for service integration in a metropolitan environment. The physical medium is the dual bus network of the type adopted by the IEEE (802.6 Standard). This facility is a reliable, high capacity Metropolitan network using efficient distributed queueing techniques. This work describes an enhancement to distributed queueing by introducing hybrid reservation protocols amenable to integrating voice, video and different data types and compatible with the concept of asynchronous transfer mode (ATM).

The proposed integration scheme uses movable boundary type frames to guarantee different services a basic capacity while providing extra capacity on demand. Performance and reliability of the underlying protocols are analyzed, system aspects discussed different priority schemes described, and redundancy techniques to improve network performance given intermittent faults are studied.

In the second phase of the research, we consider the problem of load unbalance and fault effects in the wave division multiplexing networks reffered to as ShuffleNet that necessitates the use of adaptive routing mechanism that may relieve the networks from the congestion that otherwise may exist. We take the ideal case of complete and updated knowledge of the traffic flow matrix and find the optimal probabilities of a node using a certain route so as to optimize certain performance criterion. In one version, the criterion is equalizing the usage of all wavelengths in a mean square sense, in another the criterion is the number of hops that a typical route traverses. We also take a weighted version of the two criteria and compare it against recent contributions dealing with distributed local state information. The combined analysis simulation routine is finally repeated in the case where a few link faults take place in conjunction with load unbalance.

ACKNOWLEDGEMENTS

The quality and the extent of this research would not have been possible without the expert guidance, suggestions, encouragement, friendship and genuine concerns of my advisor and supervisor Dr. A. K. Elhakeem. To him I express my deepest gratitude.

To the memory of my parents I dedicate this work.

Table of Contents

CHAPTER 1:INTRODUCTION 1

1.1 Introduction 1

1.2 Scope of the Thesis 5

1.3 Research contributions 7

1.4 Reference 9

CHAPTER 2:EVOLUTION AND REQUIREMENTS TO SWITCHED HIGH SPEED INTEGRATED SERVICES 11

2.1 BUZZ-NET Protocol 11

2.2 EXPRESSNET and FASTNET Network Protocols 16

2.3 AMTRAC Network 24

2.3.1 Exact Model 28

2.3.2 Approximate Model 31

2.4 FDDI Network 38

2.4.1 Format of the Information and Token Frame 40

2.4.2 MAC Protocol 43

2.4.3 Analysis 46

2.4.4 Results 52

2.5 References 56

CHAPTER 3:WAITING TIME ANALYSIS IN A SINGLE BUFFER DQDB NETWORK 60

3.1 Introduction 60

3.2 DQDB Analysis 61

3.3 References 77

CHAPTER 4:DQDB REQUIREMENTS: SERVICE INTEGRATION AND RELIABILITY ISSUES FOR DUAL BUS METROPOLITAN NETWORKS ... 80

4.1 Introduction 80

4.2 Frame Structure 84

4.3 Transfer of Narrowband and Wideband Data 89

4.4 Priority Schemes 94

4.5 synchronous Transmission for Voice and Video ... 95

4.6 A Model of the Proposed Protocol 99

4.7 Computational results 121

4.8 Fault Probabilities and their Effect on Service Rate 139

4.9 Computation of the Performance Criteria under Faults 145

4.10 Conclusion 154

4.11 References 156

Appendix 4A- Delay in a Network of Queues 161

Appendix 4B- The Weibull Model 163

CHAPTER 5: SHUFFLENET REQUIREMENTS: ROUTI℥G TECHNIQUES FOR LIGHTWAVE SHUFFLENET UNDER FAULTS AND UNBALANCED LOADS ..164

5.1 Introduction 164

5.2 Network Description 165

5.3 The Optimum Routing Policy 167

5.4 The Random Routing Policy 172

5.5 The Greedy Routing Policy 172

5.6 Buffering Delay and Reliability under Faults .. 173

5.7 Averag␣ Queue Size and Buffer Overflow 175

5.8 Traffic Flow Distribution and Traffic Intensity 176

5.9 Results and Conclusion 177

5.10 References 193

CHAPTER 6:CONCLUSION AND FURTHER RESEARCH 195

6.1 Conclusion 195

6.2 Further Research Work 202

List of Figures

Fig. 2.1 Buzz-net topology
Fig. 2.2 Buzz-net state diagram
Fig. 2.3 Topology of Expressnet
Fig. 2.4 Activity on Expressnet over a cycle on Inbound Channel
Fig. 2.5 Topology of Fasnet
Fig. 2.6 Format of a Slot in Fasnet
Fig. 2.7 Average Delay in Fasnet a=0.1
Fig. 2.8 Average Delay in Fasnet a=10
Fig. 2.9 System Topology of AMTRAC
Fig. 2.10 Channel assignment diagram for N=4 in AMTRAC Network
Fig. 2.11 AMTRAC Delay, a varies
Fig. 2.12 AMTRAC Delay, L varies
Fig. 2.13 FDDI Dual ring configuration
Fig. 2.14 FDDI Wrap back
Fig. 2.15.a FDDI Information Frame Format
Fig. 2.15.b FDDI Token Frame Format
Fig. 2.16 Queueing model for asynchronous data transmission in the FDDI system
Fig. 2.17 FDDI Mean Cycle Time
Fig. 2.18 FDDI Mean Waiting Time
Fig. 2.19 FDDI PS Throughput
Fig. 3.1 The DQDB network topology
Fig. 3.2 DQDB tagged station and its L_NET and R_NET
Fig. 3.3 DQDB PMF of Virtual Requests
Fig. 4.1 Distributed Queue Dual Bus Network
Fig. 4.2 Asynchronous slot
Fig. 4.3 ACF Format
Fig. 4.4 Distributed Queueing Topology
Fig. 4.5 Keeping Count of REQ
Fig. 4.6 Waiting for Turn to Transmit
Fig. 4.7 Narrowband Data Slots in a Video Slot
Fig. 4.8 State Diagram under case 1 of Priority Scheme
Fig. 4.9 Initial State Transition at P[(0,0)/k] under case 1 Priority Scheme
Fig. 4.10 State Matrix in Vector Form
Fig. 4.11 State Diagram under case 2 of Priority Scheme
Fig. 4.12 State Diagram under case 3 of Priority Scheme
Fig. 4.13 State Diagram under case 4 of Priority Scheme
Fig. 4.14 Narrowband Buffer Overflow (20% util.)
Fig. 4.15 Wideband Buffer Overflow (20% util.)
Fig. 4.16 Narrowband Queueing Delay (20% util.)
Fig. 4.17 Wideband Queueing Delay (20% util.)
Fig. 4.18 Narrowband Buffer Overflow (50% util.)
Fig. 4.19 Wideband Data Overflow (50% util.)
Fig. 4.20 Narrowband Queueing Delay (50% util.)
Fig. 4.21 Wideband Queueing Delay (50% util.)
Fig. 4.22 Narrowband Buffer Overflow (100% util.)
Fig. 4.23 Wideband Buffer Overflow (100% util.)

Fig. 4.24 Narrowband Queueing Delay (100% util.)
Fig. 4.25 Wideband Queueing Delay (100% util.)
Fig. 4.26 Video and Voice Results
Fig. 4.27 Two Dimensional Network of Queues
Fig. 4.28 Alpha3 with Beta=3% and TMR
Fig. 4.29 Alpha1 with Beta=3% and TMR
Fig. 4.30 Alpha3 with Beta=40% and TMR
Fig. 4.31 Alpha1 with Beta=40% and TMR
Fig. 4.32 Alpha2 TMR
Fig. 4.33 Alpha3 Beta=40% QMR
Fig. 4.34 Alpha1 with Beta=40% and QMR
Fig. 4.35 WEIBULL for Alpha1 M=0 Beta=3% TMR
Fig. 4.36 WEIBULL for Alpha3 M=0 Beta=3% TMR
Fig. 4.37 WEIBULL for Alpha1 M=1 Beta=3% TMR
Fig. 4.38 WEIBULL for Alpha3 M=1 Beta=3% TMR
Fig. 4.39 WEIBULL for Alpha1 M=0 Beta=40% TMR
Fig. 4.40 WEIBULL for Alpha3 M=0 Beta=40% TMR
Fig. 4.41 WEIBULL for Alpha1 M=1 Beta=40% TMR
Fig. 4.42 WEIBULL for Alpha3 M=1 Beta=40% TMR
Fig. 4.43 WEIBULL Alpha2 M=0 TMR
Fig. 4.44 WEIBULL on Alpha2 K=0.1, M=1
Fig. 5.1 ShuffleNet with P=2, k=2, and 5 hops
Fig. 5.2 Wavelength Diff. Util.
Fig. 5.3 Transmission Delay
Fig. 5.4 Mean Utilization
Fig. 5.5 Variance
Fig. 5.6 Queue Size Random VS Greedy
Fig. 5.7 Queue Size Random VS Optimal
Fig. 5.8 Buffer overflow Random VS Greedy
Fig. 5.9 Buffer overflow Random VS Optimal

List of Important Symbols

λ_w	mean arrival rate for video service
λ_v	mean arrival rate for voice service
λ_{d2}	mean arrival rate for wideband data service
λ_{d1}	mean arrival rate for narrowband data service
$1/\mu_w$	mean service time for video service
$1/\mu_v$	mean service time for voice service
$1/\mu_{d2}$	mean service time for wideband data service
$1/\mu_{d1}$	mean service time for narrowband data service
Δ_1	priority threshold 1
Δ_2	priority threshold 2
Pb_w	video blocking probability
N_w	maximum number of video slots in the video subframe
N_v	maximum number of voice slots in the voice subframe
Nd_1	maximum number of narrowband data slots in the narrowband data subframe
Nd_2	maximum number of wideband data slots in the wideband data subframe
ρ_w	video utilization
ρ_v	voice utilization
m	number of video slots in use
θ	ratio of video slot size to voice slot size
q'	voice distribution
Pb_v	voice blocking probability
T_w	average video call establishment time
k	number of available slots from the video and voice subframe expressed in terms of narrowband data slots.
z	equivalent narrowband slots in a video slot
y	equivalent narrowband slots in a voice slot
nd_1	narrowband data state
nd_2	wideband data state
$P[(nd_1,nd_2)/k]$	joint probability distribution for narrowband and wideband data.
S_n	vector representation of joint probability distribution for narrowband and wideband data.
M_1	maximum narrowband data buffer
M_2	maximum wideband data buffer
Q	state coefficient matrix
V	state vector
D_{Nd1}	delay for narrowband data
D_{Nd2}	delay for wideband data
P_{BNd1}	buffer overflow for narrowband data
P_{BNd2}	buffer overflow for wideband data
$P_{(N1)}$	marginal distribution for narrowband data
$P_{(N2)}$	marginal distribution for wideband data
F	frame size
α_1	probability that AU steals one slot
α_2	probability that AU loses one slot
α_3	probability that AU capacity does not change
β	probability that AU errors reduce frame F by one

	slot
μ_r	capacity of AU r
z(t)	hazard function
f(t)	density function
R(t)	reliability function
η	ShuffleNet channel efficiency
I	wavelength differential utilization
σ	variance

CHAPTER 1

INTRODUCTION

1.1 Introduction

Switched high speed data services are now more in demand than the dedicated lines because of their cost effectiveness and their accessibility to a larger population. Much larger areas have then to be covered with switched data networks. The protocols used for Local Area Networks (LANs) particularly the Media Access Control (MAC) protocols are not designed for networks with large geographical radii. The distance limitations are related to signal propagation times and the limits they impose on contention-based access methods such as carrier sense multiple access with collision detection (CSMA/CD) on a bus topology as described in [METC1].

The CSMA/CD method is inexpensive to realize and gives excellent performance to lightly loaded networks; however it has important drawbacks which make it impossible to use in wider area networks, namely:

- The CSMA/CD protocol is unstable for moderate and heavily loaded situations due to the random nature of the demands resulting in a large number of collisions.
- The insufficient adaptability of the simple contention scheme to various network functions required in newer local area networks; e.g. wide bandwidth capability, integrated

services, priority mechanism, virtual network etc.

Tobagi et al showed in [TOBA1] and [TOBA2] that the performance of CSMA/CD degrades significantly as the ratio $\tau W/B$ increases, where τ is the end-to-end propagation delay, W is the channel bandwidth, and B is the number of bits per packet (including the preamble for synchronization).

A technical alternative to the bus based network is the LAN that uses the ring topology. The most popular ring network is the token ring as specified in IEEE 802.5 standard. It has also been shown in [BUX 1] that the performance of the ring network does not make it suited for high speed networks. Its speed limit for a small number of stations is around 20 Mbps. This limit decreases with an increase of active stations.

Metropolitan Area Networks (MAN) are tailored to high speed wider area networks. MANs are an ideal mechanism to connect dispersed LANs with high cross-network traffic requirements; They can be interconnected to high speed networks such as Broadband Integrated Services Digital Network (BISDN). Throughput in MANs starts at 100Mbps and can grow to a few Gigabits per second range depending on the protocol and the architecture used. Both fiber and coaxial cable are suitable as the underlying media, with preference given to fiber which can accommodate higher data rates. The unidirectional bus concept has been widely adopted in MAN proposals and is being investigated here for various network architectures and protocols.

We will first investigate the technological trends in high speed MAN and survey their evolution as applied to some networks. The analysis and simulation of these contributions is covered to some extent sufficient to make comparisons. Then we will look at the metropolitan networks that have the potential of being high speed networks and develop the missing requirements to make them fulfil their potential.

The first system protocol which is the LAN BUZZ-NET protocol is considered only because it is one of the earliest attempts to make use of the dual bus adopted later for MAN, and demonstrates its superiority over CSMA/CD under heavy load.

Still using packet broadcasting as in CSMA/CD but overcoming some of the limitations, a newer approach based on unidirectional broadcast system has been proposed. This system uses a unidirectional transmission medium on which users contend according to some distributed conflict free round robin algorithm. Two such proposals are Expressnet [FRAT1] [TOBA2] [TOBA3] and Fasnet [LIMB1] [LIMB2] [TOBA3] protocols. They are analyzed and considered here with some of their known variations. It is also shown that these systems are superior tc BUZZ-NET and CSMA/CD under heavy load.

A high speed network must have the proper architecture that will enhance the performance but the most important is the protocol requirements. High speed network multiaccess protocols must have very fast responses and practically no retransmissions of packets. The transmission time must be assigned with no ambiguity as to whether the line is used by some other station in the network. The verification and response to protocol commands should be instantaneous. The protocol overhead should be minimal. The protocol should be robust enough to demonstrate a high reliability and performance. Error detection must be efficient and fast; a method such as bit voting rather then cycling redundancy check (CRC) which requires computation and thus time would be more appropriate. Equitable use of the lines by all stations in the network is another objective of this protocol. Effective use of the lines must also mean minimal waiting time. The protocol must also have an efficient mechanism to support integrated services such as televideo, video conferencing, telephone, computer communication, and facsimile.

An approach which attempts to achieve some of these features is a network using multibus such as in A Multibus Train Communication (AMTRAC) network [CHLA1] which combines the wavelength division multiplexing (WDM) and the frequency division multiplexing (FDM) technologies for high speed communication.

Network implementations for MAN are considered beginning with FDDI. The limitations in speed and in full integration of services makes it a low end candidate for the high speed MAN. The two proposals that are given serious considerations because they approach the best implementation of protocol for high speed networks are the Distributed Queue Dual Bus (DQDB) network protocol and the ShuffleNet network protocol. These two networks are capable of providing high speed communication and full service integration.

1.2 Scope of the thesis

The present study shows that full service integration for Distributed Queue Dual Bus is possible and proposes a protocol that can accommodate these various services. Performance and reliability analysis of this new proposal are also performed. Another contribution that is intended for improving the routing of ShuffleNet is made followed by a combined performance/reliability analysis.

The object is in fact to define a set of requirements for high speed MANs and fulfill these requirements for DQDB and for ShuffleNet. The DQDB MAN requires the full service integration and a reliable protocol, whereas ShuffleNet requires an efficient routing policy to distribute the traffic in the network.

In chapter 2 we give a description of the BUZZ-NET, Expressnet and Fasnet protocols and some analysis to show that although better than the conventional LAN systems they are still not suited for high speed networks. New techniques such as wavelength division multiplexing and time division multiple access could be applied to high speed networking. This is demonstrated in the analysis of the AMTRAC system. Wavelength division multiplexing is the technology applied to ShuffleNet which is considered to be one of the best systems for high speed networking. Chapter 2 concludes with a description and analysis of FDDI which is a metropolitan area network protocol that has been considered for the IEEE 802.6 standard but because of its speed limitations, was dropped in favour of DQDB.

Chapter 3 gives an analysis of DQDB as surveyed in the literature (and gives the high performance and the unfairness of the system under specific conditions).

Chapter 4 is our reseach contribution to high speed integrated services in metropolitan area networks. This contribution is a proposed protocol for DQDB that would integrate services such as data, bulk data, voice, and video.

The performance and reliability issues of the proposal are analyzed as well.

Chapter 5 makes an analysis of ShuffleNet and reports our contribution to its improved performance by considering a number of routing policies. This is done in order to fulfill the high speed network requirements.

The two high speed metropolitan networks, DQDB and ShuffleNet, that are considered to be the best performers have been studied and valuable contributions have been made in the present study in chapter 4 and 5 to further improve their performance.

Chapter 6 is the conclusion of the work and makes some suggestions on further work that can be undertaken in this fascinating field.

1.3 Research contributions

- New access scheme or a protocol for service integration in DQDB Metropolitan area network for video, voice, wideband data and narrowband data is proposed.

- Application of movable boundary type frames with a technique devised to assure their optimal utilization by the services is incorporated in the protocol and is analysed.

- Definition of priority schemes to have the data services make use of the additional capacity available from the unused frames and resulting from the movable boundaries are given.

- Detailed analysis of the priority schemes, and development of algorithms in a computational method designed to determine the best priority scheme that minimize delay and gives best access to narrowband data.

- Development of a model for analyzing reliability in DQDB when faults affect the protocol, and Analytical development and computation of the performance criteria of the DQDB model under faults and the application of modular redundancy in the analysis to determine the curing effect it has on the protocol.

- The throughput of ShuffleNet is analyzed under faults and nonuniform load for various routing policies. Results are used to make a comparative evaluation of each policy with respect to transmission delay, queue size, buffer overflow and balanced wavelength utilization.

1.4 References

[BUX 1] W. Bux, "Token-Ring Local-Area Networks and their Performance," Proc. IEEE. vol. 77, pp.238-256, Feb. 1989.

[CHLA1] I. Chlamtac and A. Ganz, "A Multibus Train Communication (AMTRAC) Architecture for High-Speed Fiber Optic Networks," IEEE Journal on Selected Area in Communications, Vol.6, No.6, July 1988.

[FRAT1] L. Fratta, F. Borgonovo, and F. A. Tobagi, "The Expressnet: A Local Area Communication Network Integrating voice and data," Proc. Int. Conf. Performance Data Commun. Syst. and Their Appl., Paris, France, Sept. 14-16, 1981.

[LIMB1] J. O. Limb and C. Flores, "Description of Fasnet, a Unidirectional Local Area Communications Network," Bell Syst. Tech. Journal, Sept.1982.

[LIMB2] J. O. Limb, "Fasnet: A Proposal for a High Speed Local Network," Proc. Office Inform. Syst. Workshop, St. Maximin, France, Oct. 1981.

[TOBA1] F. Tobagi and B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," in Proc. Local Area Comm. Network Symp., Boston, MA, May 1979.

[TOBA2] F. Tobagi, F. Borgonovo, and L. Fratta, "Expressnet: A High-Performance Integrated-Services Local Area Network," IEEE Journal on Selected Area in Communications, Vol. SAC-1, No.5, Nov. 1983, pp. 898-913.

[TOBA3] F. Tobagi, and M. Fine, "Performance of Unidirectional Broadcast Local Area Networks: Expressnet and Fasnet," IEEE Journal on Selected Area in Communications, Vol. SAC-1, No.5, Nov. 1983, pp.913-926.

CHAPTER 2

EVOLUTION TO SWITCHED HIGH SPEED INTEGRATED SERVICES

2.1 BUZZ-NET Protocol

Gerla et al [GERL1] have shown some of the advantages of the unidirectional multibus when they compare the performance of BUZZ-NET with that of CSMA/CD.

BUZZ-NET is a local area network using two unidirectional busses. The stations are connected to the two busses with a transmit and receive interface (figure 2.1). The access protocol is a hybrid random access/virtual token protocol.

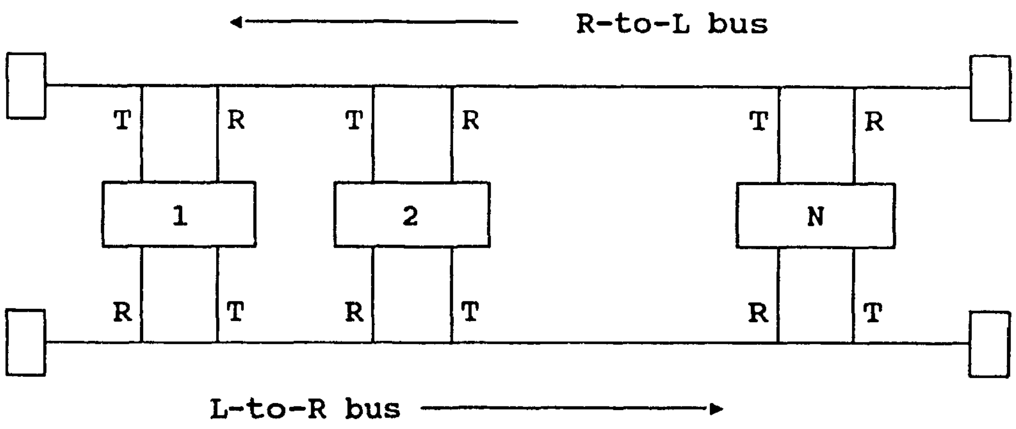


Figure 2.1 Buzz-net topology

The protocol has two phases depending on the traffic load. When the traffic load is light, BUZZ-NET behaves as a random access network and benefits from the good performance and access speed and simplicity of the random access protocol. When the traffic increases substantially namely at the point where the random access mode is less efficient, all stations switch to a controlled access mode. The protocol switchover is activated by a buzz pattern (a kind of jamming tone used in some CSMA/CD protocols) emitted on the bus (thus the name BUZZ-NET).

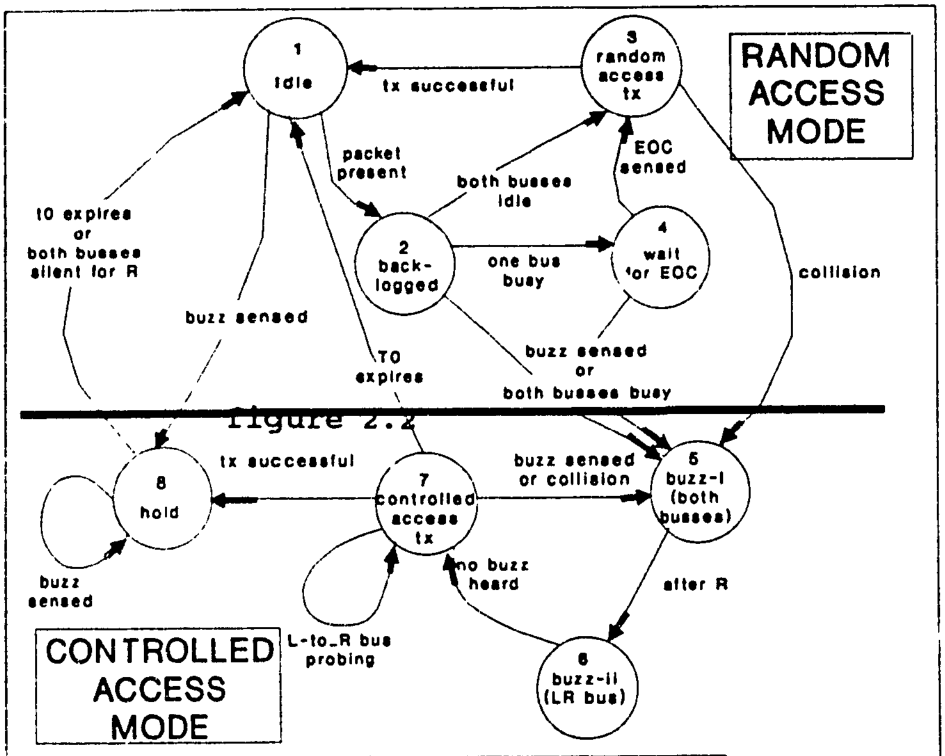


Figure 2.2 Buzz-net state diagram

The state diagram in figure 2.2 shows the operation of the BUZZ-NET protocol. At low traffic, the network is restricted to states 1 to 4 which is the random access mode.

Starting at state 1, the idle state, each station moves to state 2 when a packet is ready to be sent. Since the sending station in the random mode does not know the whereabout of the destination station, it must transmit the same packet on both busses in opposite directions. Both busses must therefore be free in order to transmit. When such is the case, the sending station goes to state 3 where the backlogged packets are transmitted. Upon successful transmission the station returns to the idle state (state 1). However, when only one bus is available and the other is still busy, the sending station goes from state 2 to state 4 where it waits for an end of carrier (EOC) the signal that indicates the bus is free. At that moment the sending station goes to state 3 to transmit. The transition to the controlled access mode at state 5 is made from either state 2, 3 or 4. This occurs when both busses are busy or if the buzz signal, which signals all the stations to switchover to the controlled access mode is detected or if there is a collision during a transmission. At state 5, a buzz pattern is transmitted on both busses to signal or confirm to all stations that a switchover to controlled access mode is occurring. After waiting R seconds which is equivalent to the round trip delay, the station goes to state 6 and transmits a buzz on the L-to-R bus (left to right bus direction which is one particular bus with respect to the station). The station then listens to both busses and goes to state 7 when no buzz is heard on either one. At state 7, the station waits for the

L-to-R bus to be free; it then probes this bus by starting to transmit on it a preamble. If no interference is sensed within a given time equal to a reaction time, it completes the transmission of the preamble followed by the data packet on the L-to-R bus as well as the R-to-L bus. If interference is sensed during this transmission, this later is aborted and the station loops back to state 7 to repeat the procedure. In order not to stay indefinitely in state 7, a timer runs for T_0 while the station is in state 7; after that time the station is forced to state 1 where it repeats the whole process. The station goes back from state 7 to state 5 when it senses the buzz or a collision. However a successful transmission allows the station to proceed to state 8 where it prepares the network to go back to the random access mode namely state 1.

Based on the fact that the activity in the network is a succession of cycles where active stations are served in a Round Robin way, lowest numbered station first, the bus utilization of BUZZ-NET at heavy load was evaluated. The cycle is described in [GERL1] as being cycle= NT + 3R +2a,
where N is the number of stations,

a = D/(N-1), D is the end-to-end delay,

T = packet transmission time, T>2a implying that no packets are successfully transmitted during random mode, and R is the round trip delay.

The heavy load bus utilization, defined as the net utilization when N stations are active and have infinite backlog is

$$S(N)=\frac{NT}{NT+3R+2a} \qquad \textbf{(2.1.1)}$$

When only i stations are active among N and disregarding the packets transmitted during random mode by the rightmost backlogged station, the utilization becomes,

$$S(i)=\frac{iT}{iT+3R+2(N-i+1)a} \qquad \textbf{(2.1.2)}$$

for i > 1.

If only one station is active, that is i=1, and no collision occurs, the station can transmit using the random access mode. In this case S(1)= 1.

Compared to CSMA/CD their performance is similar in very light load; as load increases, however, throughput degrades dramatically in CSMA/CD because of repeated collisions and as utilization increases the system exhibits an unstable behaviour where performance of stations depends randomly on the traffic pattern. In BUZZ-NET in contrast all stations are granted a fair share of the channel for any load.

In essence we see that BUZZ-NET relies on random access protocol but goes a step further in performance improvement by using two unidirectional busses. This makes its performance superior to CSMA/CD when traffic load increases. BUZZ-NET eliminates the problem of throughput degradation and unbounded delay encountered under heavy load with CSMA/CD by switching the protocol from random access to controlled access mode.

2.3 EXPRESSNET AND FASNET Network Protocols

In the same line of thought, Tobagi et al [TOBA3] have introduced the concept of unidirectional broadcast system describing in particular Expressnet and Fasnet network protocols.

The topology of Expressnet is shown in figure 2.3.

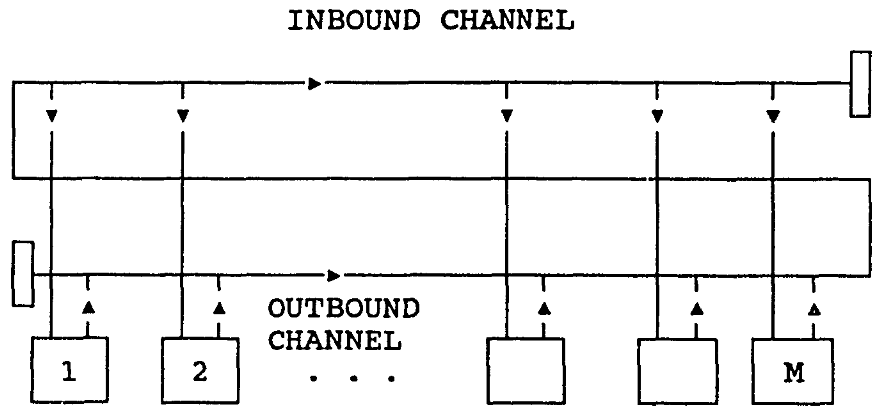


Figure 2.3 Topology of Expressnet

A station can only transmit on the outbound channel and only receive on the inbound channel. In order to transmit, a station first waits for an end of carrier [EOC(out)]. It then transmits the packet and at the same time senses the outbound channel for activity upstream of its transmission. When activity is detected, the station aborts the transmission. If there are still other packets to transmit, the cycle is repeated. A "train" which is a succession of transmission (figure 2.4) is thus generated on the outbound channel. This train is seen by all stations through the inbound channel; the destination (one or more) station can then read its packet.

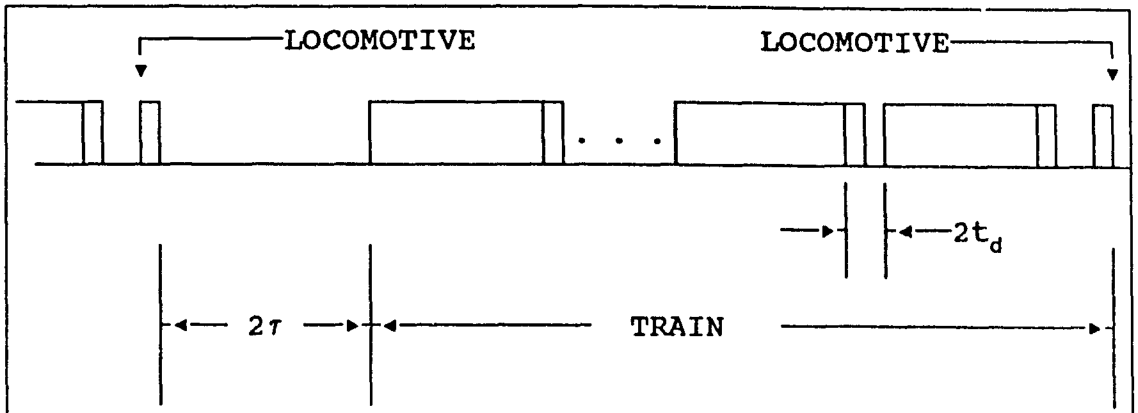


Figure 2.4 Activity on Expressnet over a cycle on Inbound Channel

The time taken to detect the presence or absence of carrier which is also the gap between two consecutive packets in the same round is denoted by t_d. So after 2 t_d, a station knows whether it has collided with the header of a packet and in such a case stops immediately its transmission. Since each station transmits a short burst of unmodulated carrier of duration t_d, the collision does not cause retransmissions for the most upstream station transmitting. Whenever either channel is idle for a time t_d, it is recognized as EOC(out) for the outbound channel. EOT(in) for the inbound channel signals the end of the train and also the start of a new round. This end of the train signal is used as a synchronization mechanism and is essential for reading the data packets on the inbound channel. Expressnet uses asynchronous transmission.

The topology of Fasnet is shown in figure 2.5 and the format of its slot in figure 2.6.

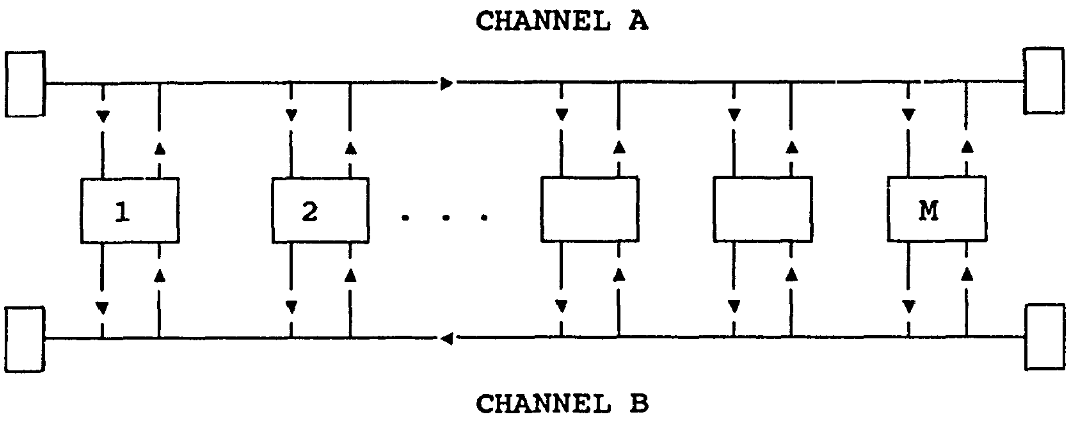


Figure 2.5 Topology of Fasnet

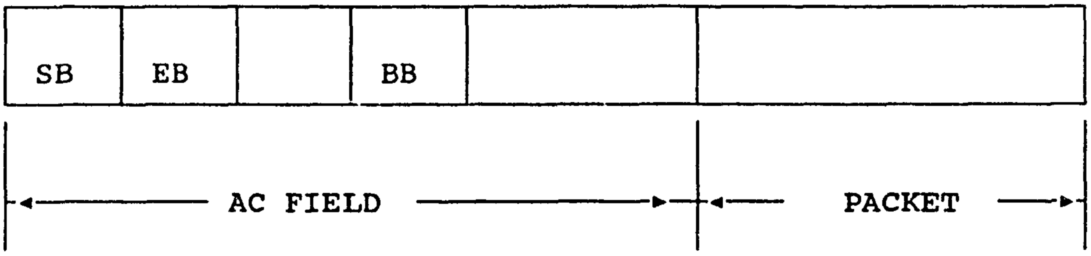


Figure 2.6 Format of a Slot in Fasnet

Fasnet uses dual unidirectional channels. Since the operation in one direction is identical to the other direction, we will only consider one of them which we denote as channel A. The destination station is always downstream to the sending station. The header station generates slots to keep the system synchronous. Each slot is made of an access control field (AC) and a data packet. The AC has a start bit (SB) used to indicate the start of a slot, a busy bit (BB) to indicate that the packet is being used and the end bit (EB) used by the end station which is the last station in the direction of transmission we are considering. The end station sets the EB on channel B to instruct the head station to initiate a new cycle.

Expressnet uses a typical round robin discipline. Each station is serviced one after the other; if a station has nothing to transmit, it transfers control to the next station. This is known as the nongated sequential service discipline (NGSS). Fasnet has two service disciplines the gated sequential service discipline (GSS) where only stations having something to transmit at the beginning of the cycle are serviced, and the most upstream first service discipline (MUFS) which is a nongated discipline; however the next to transmit is the most upstream station which has a packet to send and has not done so in the current cycle.

The model used for analysis assumes a system of M users each having a single packet buffer and being in one of two states namely idle or backlogged. An idle user generates a packet in a random time which is exponentially distributed with mean $1/\lambda$. A backlogged user does not generate any packets and becomes idle upon successful transmission of its buffer. The time required to transmit a packet is $T=B/W$ where B is the number of bits in a packet (assumed fixed) excluding the preamble in Expressnet and the AC in Fasnet; and where W is the bandwidth of the channel. The period required for the transmission of a packet of length T is $X = T + t_0 + t_p$. t_0 is the overhead before each transmission to determine which user gets access to the channel. In the case of Fasnet t_0 is given by the length of the AC field, in Expressnet it is equal to $2t_d$. t_p is the time to transmit the preamble in Expressnet; in Fasnet it is zero because Fasnet is synchronous. The interround overhead Y is the time that the channel becomes idle. In Expressnet $Y=2\tau$ where τ is the end-to-end propagation delay. In Fasnet $Y=\lceil 2\tau/X\rceil X+X$ which is an integer number of slots.

The performance measures derived from these analyses are the channel throughput, and the expected delay incurred by a packet.

In the case of a nongated sequential service discipline such as Expressnet the probability that there are n packet transmissions in a round is

$$p_n=p_0\binom{M}{n}\prod_{j=0}^{n-1}[e^{\lambda(jX+Y)}-1] \qquad (2.2.1)$$

for $0<n\leq M$.

Furthermore the average number in the system is

$$\overline{n}=\sum_{n=0}^{M} np_n \qquad (2.2.2)$$

note that p_0 can be derived from

$$\sum_{n=0}^{M} p_n=1 \qquad (2.2.3)$$

So for a given value of λ the average network throughput is

$$S=\frac{\overline{n}T}{\overline{n}X+Y} \qquad (2.2.4)$$

In Fasnet following the derivation in [LIMB1], and neglecting any overhead in the transmission slot, the throughput is given by:

$$S=\frac{\overline{n}T}{(\overline{n}+1)\,T+R} \qquad (2.2.5)$$

where R is the round trip delay.

The analysis showed that these systems can achieve close to 100% utilization even with high bandwidth and large propagation delay, which is not possible with BUZZ-NET or CSMA/CD.

Figure 2.7 shows the normalized average delay versus the throughput for number of stations M equal to 50. It is shown that the network performance for Expressnet as well as for Fasnet are superior to CSMA/CD when it comes to delay. CSMA/CD wastes twice the propagation delay between two stations for each of the transmitting station involved in the collision on top of other possible collisions when it is an n-persistent protocol. In the NGSS discipline for example, only the downstream station aborts its transmission upon detecting activity upstream. The upstream station transmits its packet. Furthermore, the downstream stations that had

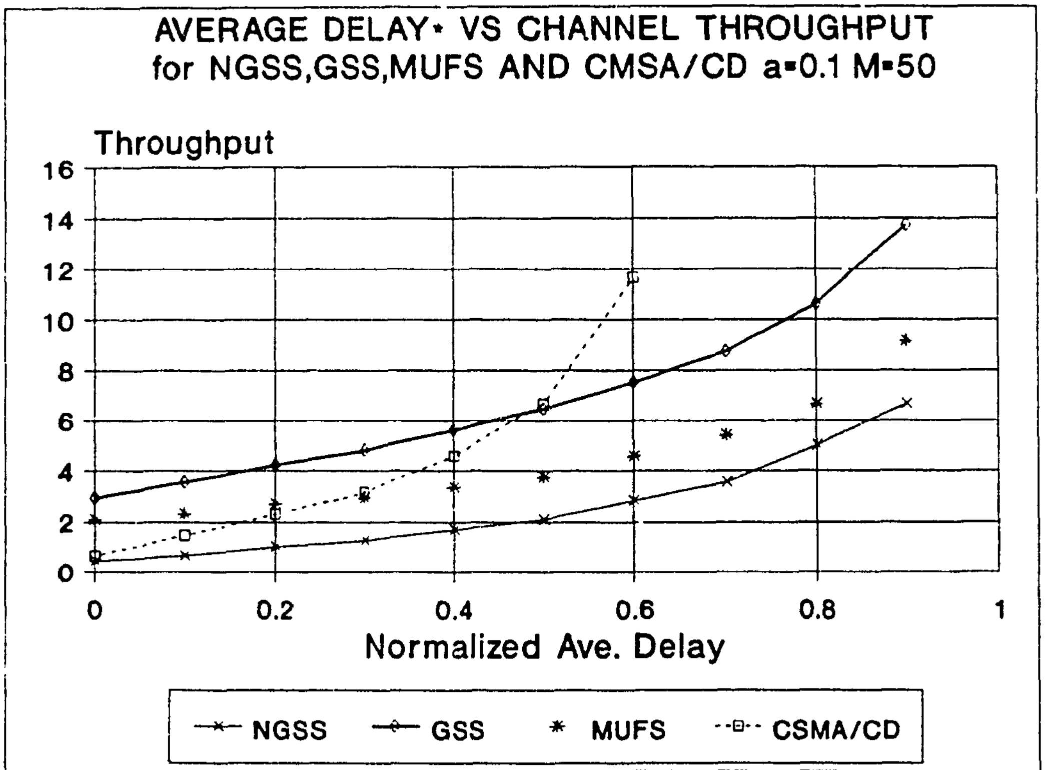


Figure 2.7 Average Delay in Fasnet a=0.1

to abort their transmission are assured a turn to transmit within the same round.

It is also shown in figure 2.8 that an increase in the factor **a** where **a=** τ/T (τ is the end to end propagation delay and **T** is the packet transmission time) makes the MUFS discipline better than the NGSS and thus the dual unidirectional bus better than the Expressnet concept.

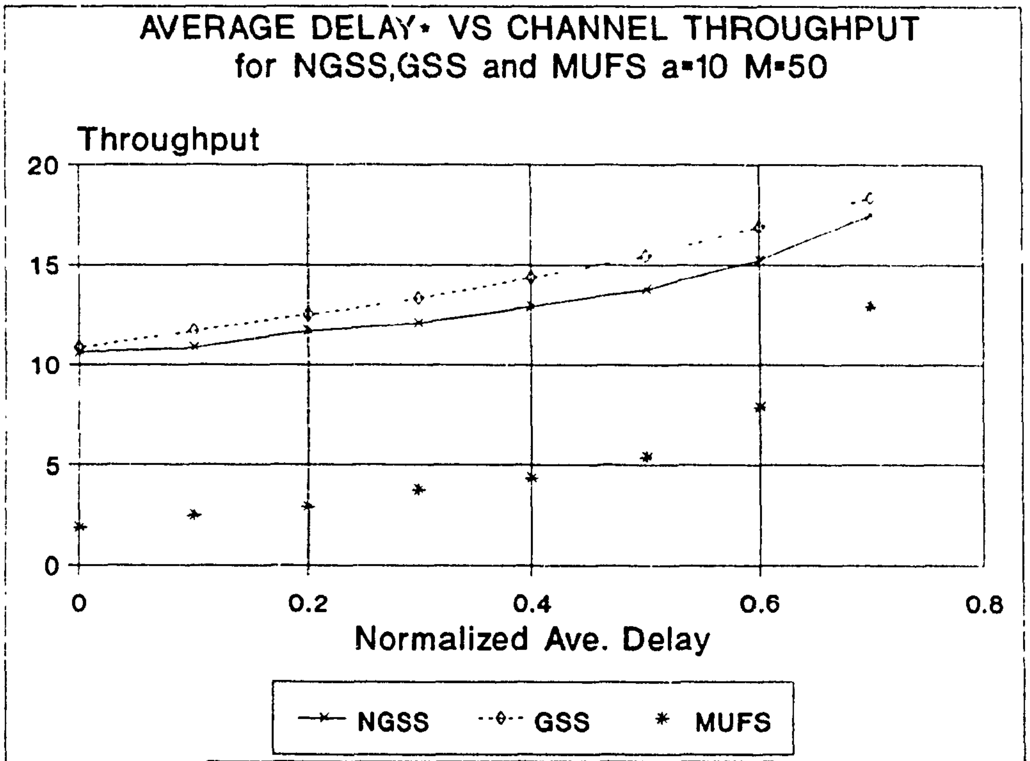

AVERAGE DELAY• VS CHANNEL THROUGHPUT
for NGSS,GSS and MUFS a•10 M•50

Figure 2.8 Average Delay in Fasnet a•10

2.3 AMTRAC Network

The unidirectional multichannel approach in the development of protocols for high speed systems is the subject of research using the frequency/time division multiple access (FTDMA) [CHLA2]. In parallel, wavelength division multiplexing (WDM) is being developed for high speed multiple-channel systems.

FTDMA allows transmission on multiple channels but the transmission on each channel obtained by frequency division is only made on preallocated time slots. This method allows the transmission of packets in parallel without creating collisions in transmission or conflicts in reception. The multiple channels are obtained by dividing the wide channel frequency spectrum into a number of smaller frequency bands.

The "A multibus train communication" (AMTRAC) solution proposed in [CHLA1] is applicable to multichannel fiber-optic systems based on frequency or wave division multiplexing. The multichannel principles of FTDMA are applied as well as the unidirectionality properties of the fiber optic medium. We shall describe the contribution by I. Chlamtac and A. Ganz entitled " A Multibus Train Communication (AMTRAC) Architecture for High-Speed Fiber Optic Networks" [CHLA1].

The network topology is shown in figure 2.9. There are N stations and b unidirectional channels in the network. Note that $1 \leq b \leq N$. All stations transmitters are connected to all channels; however each station is connected to only one particular channel for reception. The N stations are divided into b groups of N/b stations each. Each group is assigned to receive from one specific channel. Let μ=N/b; so if a source (s) station wants to transmit a packet to a destination (d) station, the channel selected (c) is given by $c=\lceil d/\mu \rceil$.

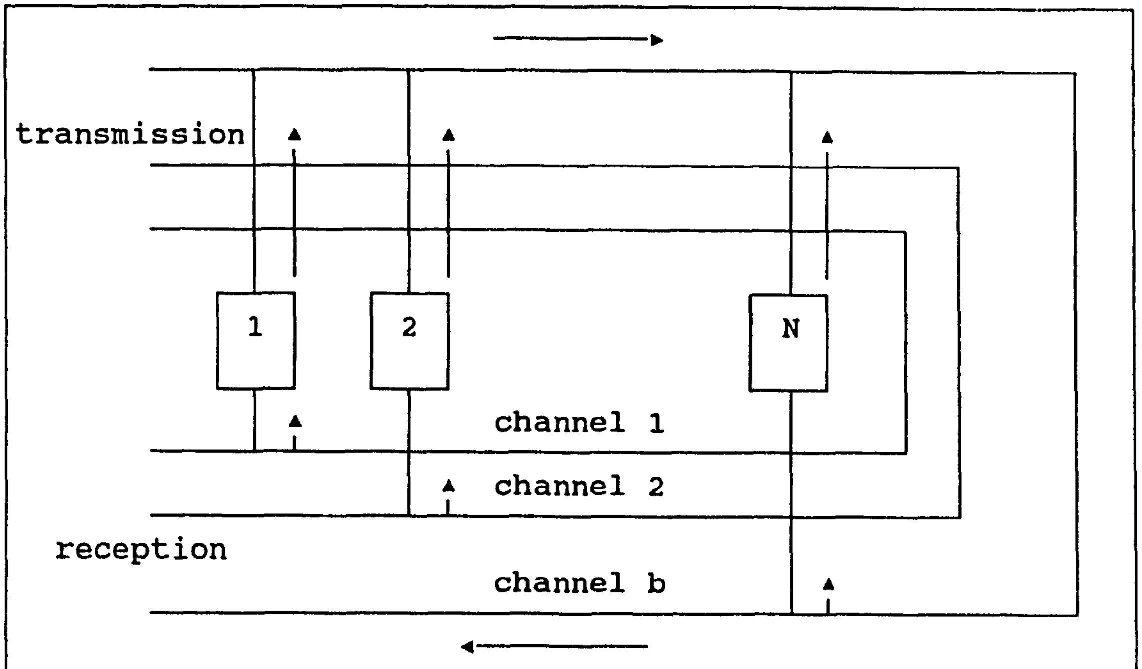


Figure 2.9 System Topology of AMTRAC

Since each channel is slotted, it is not enough to know which channel to use in order to transmit to a particular station, it is also necessary to determine the proper transmission time or slot within the channel selected. The slots are part of a cyclic scheduling structure whereby each cycle has 2N-2 slots each. Each station is assigned a synchronous time slot for transmission within each channel in the network. However, as shown in figure 2.10, the first time slot in a cycle of each channel is assigned to a different station so that a source station will not have to schedule a transmission on more than one channel at a time.

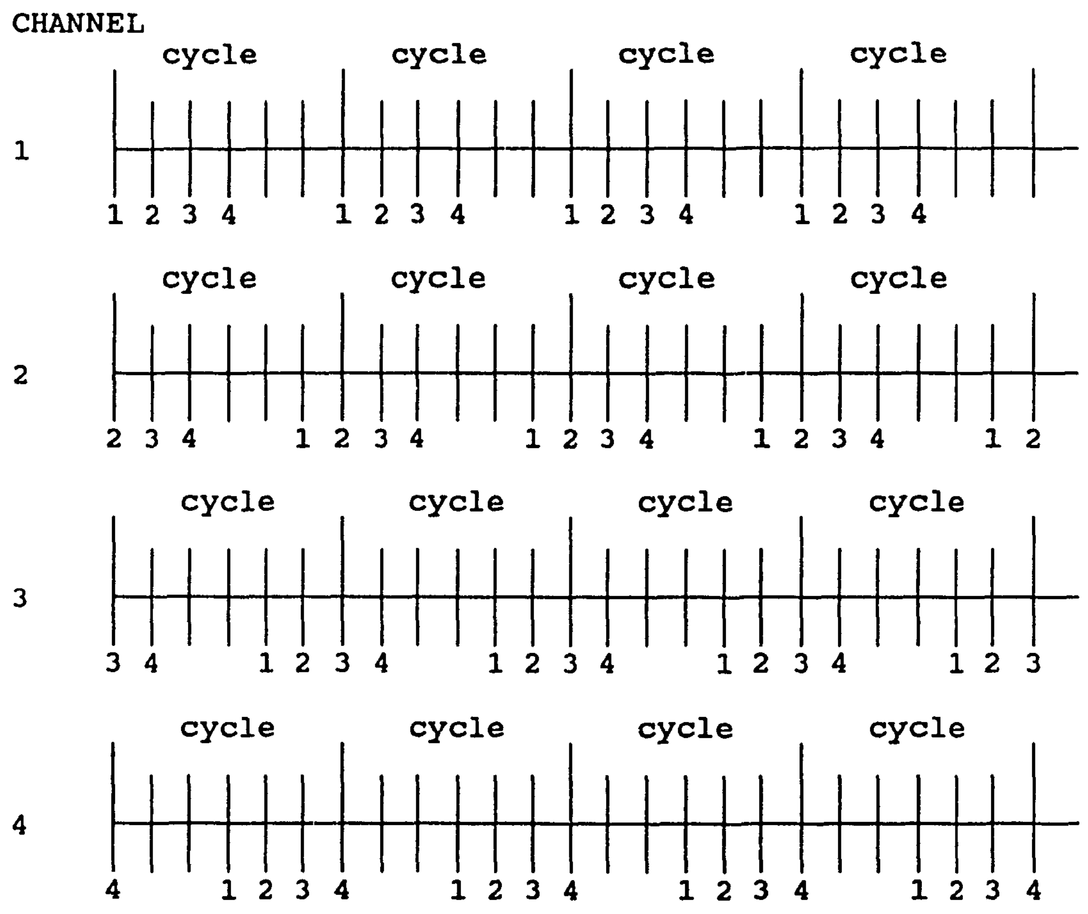


Figure 2.10 Channel assignment diagram for N=4 in AMTRAC Network

A station is thus scheduled for transmission on channel c and in slot (s-c+2N-2) mod (2N-2). The cycle length is 2(N-1) with N-1 slots dedicated to passing control among the N-1 stations and, due to propagation delay, N-1 slots needed to pass control back to the first station. We define $\tau_{i,j}$ to be the propagation delay between station i and j and τ or $\tau_{1,N}$ to be the end to end propagation delay. The time of each slot is $\tau_{i,i+1}$ or Δ and is also equal to the propagation delay between adjacent stations. If we let clock be a slot counter within the cycle, where $0 \leq \text{clock} \leq 2N-3$, then the algorithm governing the transmission protocol is

```
While packet (s,d) for transmission
  begin
    $c=\lceil d/\mu \rceil$   {choose channel for transmission}
    $m_{sc} = \Delta \cdot (s-c+2N-2) \bmod (2N-2)$   {choose slot for
                          start of transmission on channel c}
    wait until clock = $m_{sc}$
    if channel idle then transmit
  end.
```

The network is analyzed for its traffic behaviour. Each station i is assumed to have N buffers where N-1 buffers are dedicated for transmission, each to a different destination j, for $1 \leq j \leq N$, and $j \neq i$ and one buffer is dedicated to packet reception. Assuming a closed queueing system, the probability of a packet arrival at a slot for idle user (i,j) is r_{ij} . An idle user is one with an empty transmit buffer and a backlogged user is one with a packet for transmission.

The analysis is made for an exact model of the system behaviour. Because of the computational complexity of the exact model in the case of large networks an approximate model of the system behaviour was also developed.

2.3.1 Exact Model

The state of the system is defined by (X,s) where s is the beginning of the sth slot, $0 \le s \le 2N-3$ and X is an NxN matrix with

$$X(i,j)= \begin{cases} 0 & \textbf{no packet at user (i,j)} \\ -1 & \textbf{packet queued at user (i,j)} \\ m & \textbf{nber of slots remaining before transmission} \\ & \textbf{at user (i,j)} \end{cases}$$

where $1 \le m \le LEN$ (LEN is the per channel packet transmission time on each channel.)

We denote P_s as the transition probability matrix in slot s equal to $Pr((X,s) \rightarrow (X^*,(s+1) \bmod (2N-2)))$. The transition probability matrix is

$$P=\prod_{s=0}^{2N-3} P_s \qquad \textbf{(2.3.1)}$$

In order to find the steady state probability of X in slot s, we have

$$\Pi_0 = \Pi_0 \cdot P$$

$$\sum_X \Pi_0(X) = 1 \tag{2.3.2}$$

The steady state equation is then

$$\Pi_s = \Pi_0 \prod_{i=0}^{s-1} P_i \tag{2.3.3}$$

for s > 0.

The transition probability matrix P_s is

$$P_s = \prod_{(i,j)} q_s(i,j) \tag{2.3.4}$$

where $q_s(i,j) = \Pr\big((X(i,j),s) \rightarrow (X^*(i,j),(s+1) \bmod (2N-2)) \big)$ and

$$X^*(i,j) = \begin{cases} 0 & \text{if } [(X(i,j)=0) \text{ and (no arrival at user } (i,j))] \\ & \text{or } (X(i,j)=1) \text{ i.e. previous } m=1 \\ -1 & \text{if } [(X(i,j)=0) \text{ and (arrival at user } (i,j))] \\ & \text{or } [(X(i,j)=-1) \text{ and (user}(i,j) \text{ does not xsmit)}] \\ m+\text{LEN} & \text{if } (X(i,j)=m+1) \\ \text{LEN} & \text{if } [(X(i,j)=-1) \text{ and (user } (i,j) \text{ transmits)}] \end{cases}$$

Following this definition we can now calculate $q_s(i,j)$ by summing the probability of every possible value of $X^*(i,j)$ at state s+1, and for each of these values taking into consideration the possible values of X(i,j) at state s that may transmit to $X^*(i,j)$ at state s+1. However we have to give another definition namely TR(i,j,s) which is the probability that user (i,j) transmits in slot s and only takes the value 1 or 0.

User (i,j) transmits in slot s if 1) it has the permission to transmit i.e. the proper channel is selected and the time slot chosen, and 2) the chosen channel c is idle.

$$TR(i,j,s)=\delta(s=m_{ic})\cdot\prod_{(l,k)\in G}X(l,k)\,\delta\left(\left\lceil\frac{k}{\mu}\right\rceil\neq c\right) \qquad \textbf{(2.3.5)}$$

where G = { (1,k) | X(1,k) > 0 }

and δ (statement) equals 1 if statement is TRUE and 0 otherwise.

Now the term $q_s(i,j)$ is given by

$$\mathbf{q_s(i,j) = \delta(X^*(i,j)=0)\cdot[\ \delta(X(i,j)=0)\cdot(1-r_{ij}) + \delta(X(i,j)=1)] +}$$
$$\mathbf{\delta(X^*(i,j)=-1)\cdot[\delta(X(i,j)=0)\cdot r_{ij})+\delta(X(i,j)=-1)\cdot(1-TR(i,j,s))]+}$$
$$\mathbf{\delta(X^*(i,j)=m)\ \cdot\ \delta(0<m<LEN)\ \cdot\ \delta(X(i,j)=m+1)\ +}$$
$$\mathbf{\delta(X^*(i,j)=LEN)\cdot\ \delta(X(i,j)=-1)\ \cdot\ TR(i,j,s)\ .} \qquad \textbf{(2.3.6)}$$

Substituting this in equation 2.3.4 and solving equation 2.3.1, 2.3.2 and 2.3.3 we obtain the steady state equations $\Pi=\{\ \Pi_0\ \Pi_1\ \Pi_2\ \cdot\ \cdot\ \cdot\ \Pi_{2N-3}\ \}$.

The average system throughput is given by

$$S=\sum_{i}\sum_{j}S_{ij} \qquad \textbf{(2.3.7)}$$

where S_{ij}, the throughput of user (i,j) is

$$S_{ij}=\frac{1}{2N-2}\sum_{s=0}^{2N-3}\sum_{X}\delta(X(i,j)>0)\Pi_s(X)\,. \qquad (2.3.8)$$

where $\delta(X(i,j)>0)$ is the indication as we know that in state X user (i,j) transmits.

The average packet delay D is given by

$$D=\sum_i\sum_j\frac{S_{ij}}{S}\cdot D_{ij} \qquad (2.3.9)$$

where D_{ij} is the average delay for user (i,j). If we know the average number of packets at user (i,j) we can obtain D_{ij} using Little's result. The average number of packets at user (i,j) is given by

$$Q_{ij}=\frac{1}{2N-2}\sum_{s=0}^{2N-3}\delta(X(i,j)\neq 0)\Pi_s(X)\,. \qquad (2.3.10)$$

and using Little's result

$$D_{ij}=\frac{Q_{ij}}{S_{ij}} \qquad (2.3.11)$$

2.3.2 Approximate Model

The exact solution involves an exponential number of equations and therefore can only be used in networks with small number of stations. In the case of bigger networks an approximate model is developed for a symmetric system.

In this model, a user generates a packet at the slot with probability r. A Markov chain is embedded at the beginning of each cycle. The state at the embedded point is given by the vector (nt,nq) where nt is the packet transmission and nq the number of packets queued for transmission.

We let ψ be the steady-state probability vector where each of its terms $\psi_{nt,nq}$ is the steady-state probability that at the beginning of a cycle there are nt packets in transmission and nq packets queued. Let P be the transition probability matrix, then

$$\psi P=\psi$$

$$\sum_{nt,nq} \psi_{nt,nq}=1. \qquad (2.3.12)$$

To calculate the transition probability matrix P between consecutive embedded points, we consider the slotted time property of the model and compute P as the product of the slot transition probability matrices

$$P=\prod_{t=0}^{2N-3} ST(t) \qquad (2.3.13)$$

where ST(t) for $0 \le t \le 2N-3$, is the slot transition matrix whose entries are defined as Pr(nt,nq,t → nt1,nq1,t+1). At the beginning of the slot we can identify the following events that make up these entries:

1) j packets end transmission,

2) h packets arrive and

3) ℓ packets begin transmission.

Summing over all the possibilities of the number of packets ending the transmission we obtain

$$Pr(nt,nq,t\rightarrow nt1,nq1,t+1)=\sum_{j=0}^{\mathrm{mir}(nt,ns(t))}\begin{pmatrix}pend(j\,|nt,t,T_b)\\ \cdot parrive(nq1-(nq-(nt1-(nt-j)))\,|nt,nq,j)\\ \cdot pbegin(nt1-(nt-j)\,|nt,nq,j,t)\end{pmatrix}$$

where **(2.3.14)**

T$_b$ is the packet transmission time, integer multiple of a cycle.

pend term is the probability that j packets ended the transmission in slot t given nt packets are in transmission.

parrive term is the probability of h packets arrivals in a slot given nt packets are in transmission, nq packets are queued, and j packets ended the transmission.

pbegin term is the probability that ℓ packets begin the transmission in slot t given nt packets are in transmission, nq packets are queued and j packets ended the transmission.

We also define ns(t) which is used in the computation:

$$ns(t)=\begin{array}{ll}N-t & for\ 0\leq t\leq N-2\\ 2 & for\ t=N-1\\ t-N+2 & for\ N\leq t\leq 2\cdot N-3\end{array} \qquad \textbf{(2.3.15)}$$

We will now calculate every term in equation 2.3.13.

The term **pend** is found by taking the total number of different combinations of nt packets in transmission such that exactly j packets end transmission and dividing it by the total number of different combinations of nt packets in transmission.

$$pend(j|nt,t,T_b) = \sum_{k=j}^{\min(nt,ns(t))} (-1)^{k-j}\binom{k}{j}\binom{\min(nt,ns(t))}{k}$$
$$\binom{n-k}{nt-k}\binom{n-k}{nt-k}\cdot(nt-k)!\left(\frac{1}{T_b}\right)^{k}$$
$$/\left[\binom{n}{nt}\binom{n}{nt}\cdot nt!\right]$$

2.3.16

The term **parrive** involves the binomial distribution with the success parameter r (probability of a user generating a packet at the beginning of a slot).

$$parrive(h|nt,nq,j) = \binom{N^2-nq-nt+j}{h}\cdot r^{h}\cdot(1-r)^{N^2-nq-nt+j-h}$$

2.3.17

The term N^2-nq-nt+j represents the potential users that can have arrival.

The term **pbegin** is given as the ratio between the total number of different combinations of nq packets in queue such that exactly ℓ packets begin transmission in slot t, and the total number of different combinations of nq packets in queue.

$$pbegin(\ell|nt,nq,j,t) = \frac{\sum_{i=1}^{\min(nq,x)} (-1)^{i-\ell}\binom{i}{\ell}\binom{x}{i}\binom{N^2-nt+j-i}{nq-i}}{\binom{N^2-nt+j}{nq}}$$

2.3.18

where x is given by

$$\begin{aligned} x &= 0 && \text{if } \ell=0 \wedge ns(t)-(nt-j)<0 \\ &= ns(t)-(nt-j) && \text{otherwise} \end{aligned} \qquad \textbf{(2.3.19)}$$

The system performance can now be found by solving first equation 2.3.12 . The system throughput is

$$S=\sum_{nt=1}^{b} nt\cdot\sum_{nq=0}^{N^2-nt}\psi_{nt,nq}. \qquad (2.3.20)$$

The average packet delay is obtained by Little's result

$$D=\frac{Q}{S} \qquad (2.3.21)$$

where Q is the average number of packets in the system.

$$Q=\sum_{nt=0}^{b}\sum_{nq=0}^{N^2-nt}(nt+nq)\,\psi_{nt,nq}. \qquad (2.3.22)$$

These results have shown that at low loads each packet waits on the average half a cycle. At high throughput the maximum average packet delay is approximately NT_b. Using the normalized propagation delay **a**=τ/T which is the end to end propagation delay over the packet transmission time, and taking constant packet length, we found that as **a** increases so does the average packet delay as shown in figure 2.11. This also means as the channel speed increases so does the average packet delay. At throughput of 0.5 for instance the average delay doubles when the transmission rate goes from 1 to 8.

As the packet transmission time determines the normalized propagation delay penalty, the increased packet length leads to higher utilization levels.This is shown in figure 2.12.

It is to be noted that the control overhead of the protocol does not increase with the number of channels since each station has to transmit and receive on a single channel at a time.

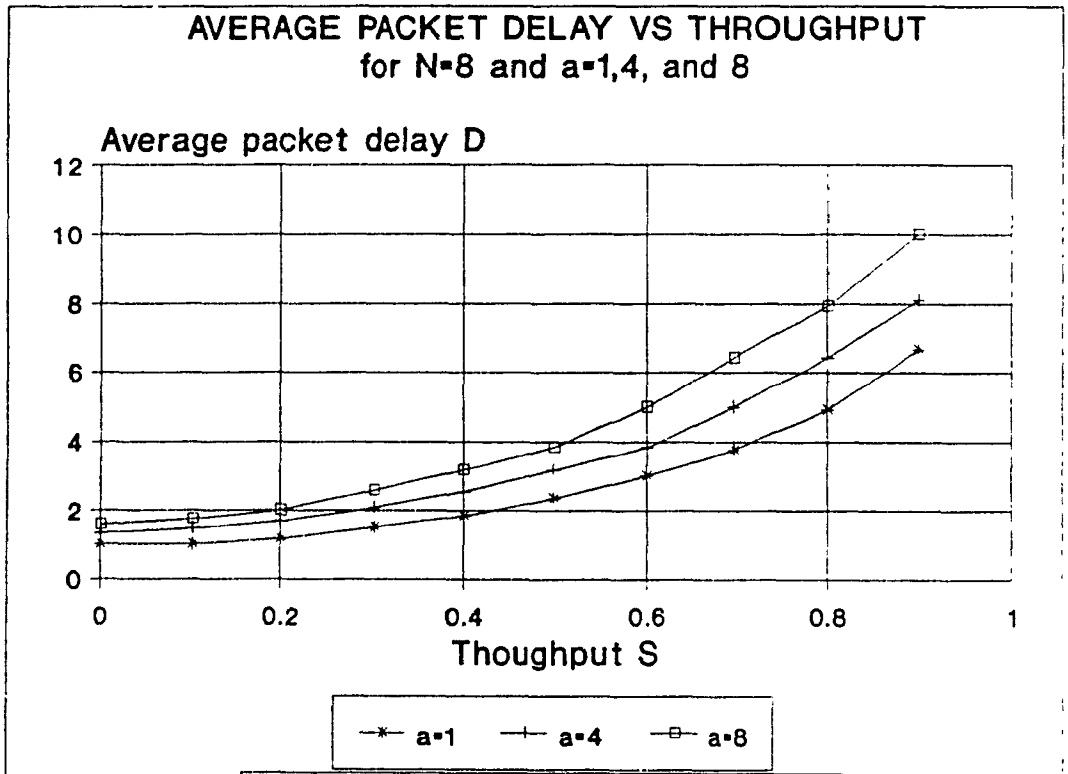


Figure 2.11 AMTRAC Delay, a varies

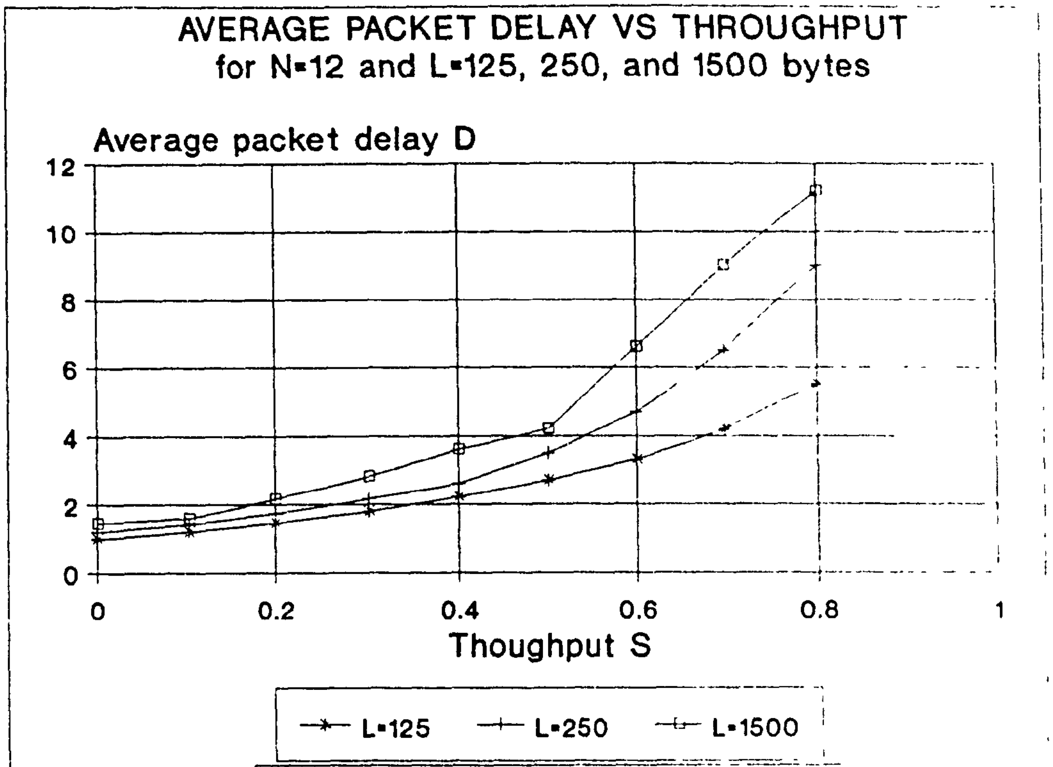


Figure 2.12 AMTRAC Delay, L varies

2.4 FDDI Network

The American National Standard Association (ANSI) X3T9 committee has developed the Fiber Distributed Data Interface (FDDI) which has been considered as a general purpose high speed local or metropolitan area network. The committee has also worked on a hybrid scheme, called FDDI-II, to provide FDDI with a circuit switched capability for voice in addition to its existing packet switching capability.

FDDI is composed of two counter-rotating token passing rings each of which runs at 100 Mbps (figure 2.13). Optical fiber is used in the rings for transmission between stations. The counter rotating rings can be used to provide redundant data paths for reliability. The stations in the network use optical transmitters and receivers.

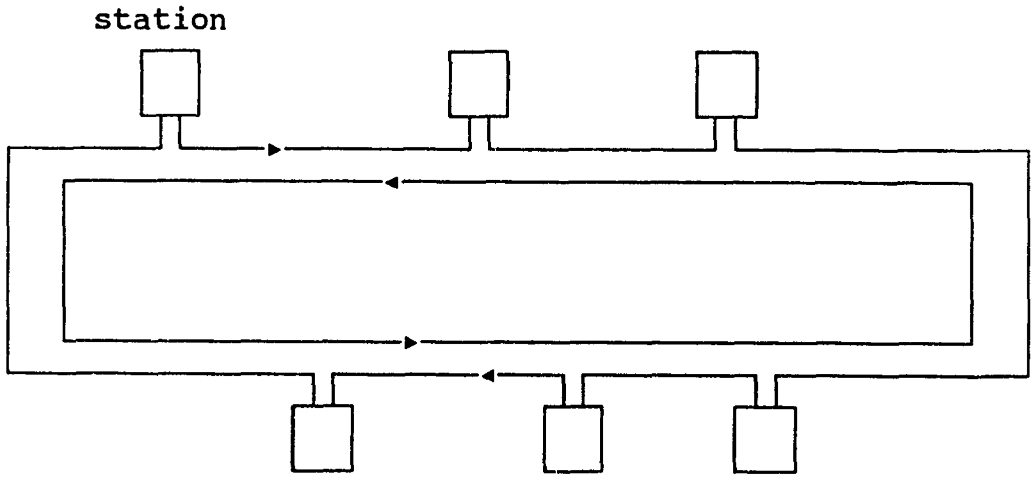


Figure 2.13 FDDI Dual ring configuration

In case of link failure or station failure the dual rings are reconfigured by wrapping back the wiring into a single ring as shown in figure 2.14. The failed link or station is thus eliminated from the network until it is repaired.

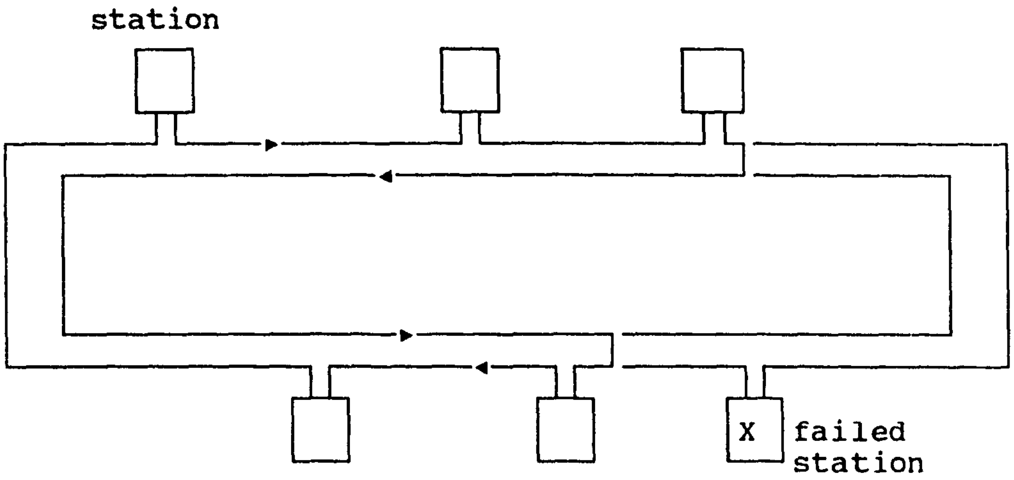


Figure 2.14 FDDI Wrap back

When both rings are connected to all stations and used for transmission at all time instead of saving one as a standby ring as shown above, then the transmission rate of the network becomes 200 Mbps. Transmission of information between stations is asynchronous. In the case of FDDI-II which is a superset of FDDI, the slots have channels dedicated to data packets and 16 wideband channels that can be used for voice or video. The wideband channels use isochronous transmission. A station is designated to act as a cycle master to create cycles or slots at an 8 KHz rate. It also marks any isochronous wideband channel as free to be used or assigned to a particular communication.

In FDDI a station can transmit only if it holds the token frame. The token frame is a control signal comprised of a unique signalling sequence which any station may capture. There is only one token frame in the ring so only one station can transmit at a time on the same ring. When the station holding the token finishes transmitting information frames, it releases the token and transmits it to the next station. When both rings are used for transmission, there is a token on each ring.

When an information frame is transmitted on a ring, all stations receiving the frame forward it on the ring without affecting it. The destination station copies the information as the frame passes. When the transmitted frame reaches the source station this latter destroys it. The token is passed when the station finishes its transmission or if its allotted time is over. The token is transmitted right after the last information frame by the source station. Unlike IEEE 802.5, the source station does not wait for the frame to come back from its trip around the ring before releasing the token.

2.4.1 Format of the Information and Token Frames

The frame format at the MAC level is described in terms of symbols where a symbol corresponds to 4 bits. The structure of the information and token frames generated by the FDDI protocol are shown in figure 2.15.

SFS		FCS Coverage					EFS	
PA	SD	FC	DA	SA	INFO	FCS	ED	FS

SFS- start of frame sequence
EFS- end of frame sequence

Figure 2.15 a) FDDI Information Frame Format

PA	SD	FC	ED

Figure 2.15 b) FDDI Token Frame Format

The Preamble (PA) is used to synchronize the frame with each station's clock. It consists of 16 idle symbols generated by the source station. Subsequent stations may change the length of the preamble to stay consistent with clocking requirements.

The Starting Delimiter (SD) indicates the start of the frame consisting of a J and K symbols both distinguishable from data symbols.

The Frame Control (FC) is composed of 2 symbols with the following bit format: CLFF ZZZZ where C is a bit that indicates whether the frame is synchronous or asynchronous; L is a bit that indicates the use of 16-bit or 48-bit addresses; FF are two bits indicating whether this is a LLC frame or a MAC frame; for a token FF is set to 00. When it is a MAC frame the symbol ZZZZ indicate the type of MAC frame; for a token ZZZZ is set to 0000.

The Destination Address (DA) is a physical address of a station, a group of station or all stations on the local network. The destination address as well as the Source Address (SA) are 4 or 12 symbols in length (16 bits or 48 bits) depending on the setting of the L bit in the Frame Control field.

The Information (INFO) field contains the higher layers data or information control.

The Frame Check Sequence (FCS) field uses 8 symbols for a 32-bit cyclic redundancy check based on the FC, DA, SA and the INFO fields.

The Ending Delimiter (ED) indicates the end of the frame; it is 4 bits long (1 symbol) for all frames except the token which is 8 bit long (2 symbols). The variation is required so that frames occupy an integral number of octets.

The Frame Status (FS) is the last field in an information frame and it contains the error detected (E), address recognized (A), and frame copied (C) indicators. Each indicator is represented by a symbol to give the status of the indicator (true or false). If an error is detected in the frame, the E indicator is set with the symbol for true. If a station detects its own address in the passing frame, it sets the A indicator. If it copies the frame it also sets the C indicator. With these indicators the source station is able to know if a station exists and if it read the frame sent to it. Errors are handled by higher layer protocols.

2.4.2 MAC Protocol

The network is initialized in order for all stations to negotiate a target token rotation time (TTRT). During the negotiation each station takes into consideration the percentage of time that will be used for synchronous transmission. The remaining time will be used for asynchronous transmission. At the end of the negotiation, the shortest TTRT is used to set a timer T_Opr identically in each station. The determination of TTRT results from a process of the MAC level known as the MAC claim process. This process also determines, during initialization, the station responsible for issuing the first token. The claim process can be started any time when the station's MAC detects the need for ring initialization. The MAC then continually transmits the claim frame which is composed of the following fields: the start delimiter, the FC coded as 1L00 0011 to uniquely identify it, the destination and source addresses both containing the same address that of the frame originator, and the information field with the first four bytes containing the station's bid for TTRT. Faster TTRT claims referred to as higher claims wins the bidding. A station receiving a claim frame with a lower claim than his disregards the frame and transmits its own. If the claim is higher it stops generating its own claim frame and passes the received one. When the originating station receives its own claim frame it has won the bidding and immediately generates the token. All stations know the new TTRT value since each has recorded the TTRT claim and passed it on.

The token rotation timer (TRT) is a counter used in each station to measure the time between successive arrivals of the token at that station. Normally the TRT is reset each time the token is received in order to time the next token rotation. If the TRT exceeds T_Opr before the token has arrived back at the station another counter Late_Ct initially zero is incremented and the TRT is reset to zero and continues timing. When the token arrives late at a station (Late_Ct=1) the TRT is not reset but continues timing. As a result of the accumulated lateness into the next token rotation time exceeding T_Opr, the asynchronous transmission is restricted until the lateness has been compensated by faster cycles of the token. The average token rotation is at most T_Opr. If Late_Ct ever exceeds one, error recovery is initiated. Late_Ct is reset to zero each time the token is received. When the token is received by each station, the token holding timer (THT) is used to control the amount of time the token is held for transmitting asynchronous frames.

At the reception of the token on time (Late_Ct=0) at the station, the THT is loaded with the current value of the TRT. The TRT is then reset and enabled. Synchronous frames are transmitted first then the THT is enabled to start asynchronous transmission. If the THT is less then T_Opr then there is still time left for asynchronous transmission.

Different asynchronous priorities can be implemented. Counters T_Pri(i) (i=1 to 8) define the token holding time threshold for asynchronous priority level 1 to 8. A large threshold value allows more time to elapse from the THT before passing the token. The maximum threshold value for a priority level is T_Opr.

The algorithm describing the protocol is:

```
If frame_received is a TOKEN
     THT counter = TRT counter
     If Late_Ct = 0
          TRT = 0
          TRT counter enabled
     endIf;
     If there is synchronous data to send
          While synchronous allocation valid
               transmit synchronous frame
          endWhile;
     endIf;
     If there is asynchronous data to send
          If TOKEN received is restricted
               If destination address is own
                    While asynchronous buffer not empty
                         transmit asynchronous frame
                    endWhile;
               endIF;
          Else
               THT counter enabled
               i = 7
               While THT < TTRT and i ≥ 0
                    While i buffer not empty and THT < T_Pr(i)
                         transmit frame
                    endWhile;
                    i = i-1
```

```
                endWhile;
            endIf;
        endIf;
        pass TOKEN forward on bus
endIf;
```

2.4.3 Analysis

The recent (Feb. 1991) queueing model taken for the FDDI MAC protocol analysis by Tangemann et al. [TANG1] deals with N stations that can be considered as a polling system. Each station is depicted as a single queue having certain unrestricted priority. Each of the queues may be of different priority. Queue i can buffer up to m_i packets. The packet arrival process at station i is Poison with rate λ_i. The server for the queueing model is the transmission channel, allocated cyclically to the station according to the FDDI MAC protocol. The service time of a packet T_{Hi} and the switchover time T_{Ui} from station i to station i+1 are random variables with general distributions. The model is shown in figure 2.16.

The imbedded Markov chain is used to analyze the model. The regeneration points are represented by the token arrival times at the stations. The token rotation time $T^{(n)}_{TRTi}$ measured by station i represents its system state at the nth token arrival along with the number of packets $A^{(n)}_i$ waiting at station i with the probability

$$a_i^{(n)}(j)=P[A_i^{(n)}=j] \qquad \textbf{(2.4.1)}$$

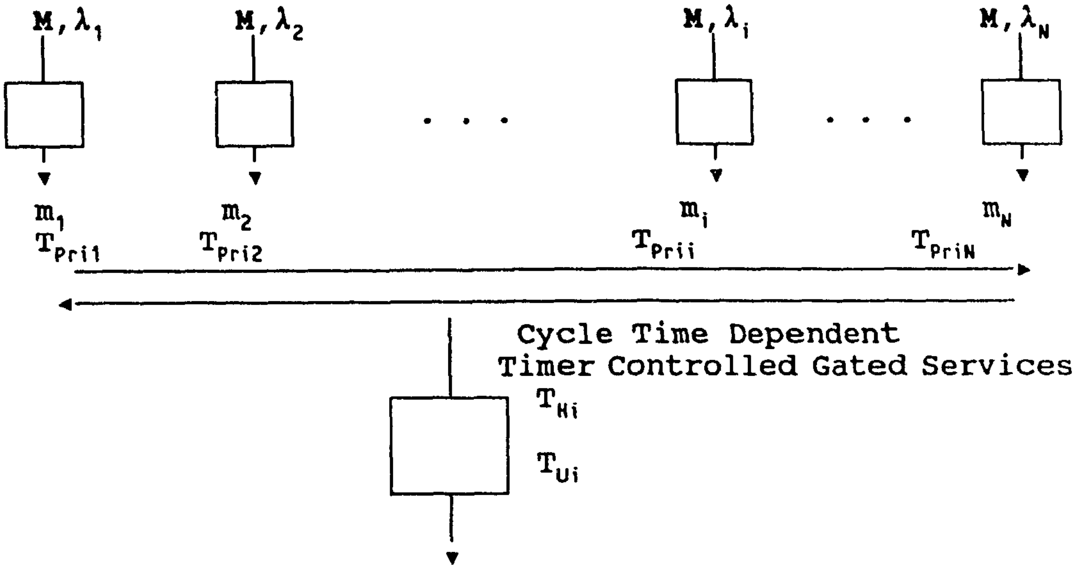


Figure 2.16 Queueing model for asynchronous data transmission in the FDDI system

$B^{(n)}_i$, representing the maximum number of packets that can be served at station i after the nth token arrival, depends on the token rotation time measured by station i. As described in the MAC protocol, the value of the token rotation timer TRT is copied into the token holding timer THT which is started counting upward when asynchronous transmission is initiated and expires when the level T_{Prii} of the station i priority packet transmission is finished. This model can be described in the Laplace domain as (characteristic function approach)

$$\Phi^{(n)}_{THT_{i,j}}(s) = \Phi^{(n)}_{TRT_i}(s)\,[\Phi_{H_i}(s)]^j \qquad (2.4.2)$$

where T_{THTij} and T_{TRTi} are random variables for j successive and independent packet service times T_{Hi}.

The maximum number of packets that can be served is expressed as

$$b_i^{(n)}=P[B_i^{(n)}=j] \qquad \textbf{(2.4.3)}$$

$$b_i^{(n)}=\begin{array}{ll} P[T_{TRT_i}^{(n)}>T_{Pri_i}] & j=0 \\ P[T_{TRT_i}^{(n)}+\sum_{k=1}^{j-1} T_{H_i}<T_{Pri_i}<T_{TRT_i}^{(n)}+\sum_{k=1}^{j} T_{H_i}] & 0<j<m_i \\ P[T_{TRT_i}^{(n)}+\sum_{k=1}^{m_i-1} T_{h_i}<T_{Pri_i}] & j=m_i \end{array}$$

or to put it in terms of the distribution functions we have

$$b_i^{(n)}=\begin{array}{ll} 1-F_{THT_{i,0}}^{(n)}(T_{Pri_i}) & j=0 \\ F_{THT_{i,j-1}}^{(n)}(T_{Pri_i})-F_{THT_{i,j}}^{(n)}(T_{Pri_i}) & 0<j<m_i \\ F_{THT_{i,m_i-1}}^{(n)}(T_{Pri_i}) & j=m_i \end{array}$$

Since the service is not fixed then the actual number of packets to be served can be determined by the discrete random variable

$$C_i^{(n)}=Min[A_i^{(n)},B_i^{(n)}] \qquad \textbf{(2.4.4)}$$

Assuming independence between $A^{(n)}_i$ and $B^{(n)}_i$, the distribution of $C^{(n)}_i$ is

$$c_i^{(n)}(j)=\begin{array}{ll} 1-(1-a_i^{(n)}(0))\cdot(1-b_i^{(n)}(0)) & j=0 \\ 1-\left(1-\sum_{k=0}^{j} a_i^{(n)}(k)\right)\left(1-\sum_{k=0}^{j} b_i^{(n)}(k)\right)-\sum_{k=0}^{j-1} c_i^{(n)}(k) & j=1,\cdots,m_i \end{array}$$

$$\textbf{(2.4.5)}$$

Since the token holding timer THT can expire before servicing all packets then the number of packets that cannot be served in this cycle is given by

$$D_i^{(n)} \begin{aligned} &= A_i^{(n)} - C_i^{(n)} = A_i^{(n)} - Min[A_i^{(n)}, B_i^{(n)}] \\ &= Max[A_i^{(n)} - B_i^{(n)}, 0] \end{aligned} \qquad (2.4.6)$$

with the distribution

$$d_i^{(n)}(j) \begin{aligned} &= \sum_{k=j}^{m_i} a_i^{(n)}(k) \cdot b_i^{(n)}(k-j) && 0 < j \le m_i \\ &= 1 - \sum_{k=1}^{m_i} d_i^{(n)}(k) && j = 0 \end{aligned} \qquad (2.4.7)$$

The station time $T^{(n)}_{Ei}$ is composed of $C^{(n)}_i$ independent packet service times T_{Hi} and the switchover time T_{Ui}, which yields in the Laplace transform domain

$$\Phi_{E_i}^{(n)}(s) = \Phi_{U_i}(s) \cdot G_{C_i}^{(n)}[\Phi_{H_i}(s)] \qquad (2.4.8)$$

where $G^{(n)}_{Ci}(z)$ is the generating function of $C^{(n)}_i$.

The cycle time $T^{(n+1)}_{Ci}$ measured by station i is the time between the nth and the (n+1)th token arrival at station i. It is given by

$$T_{C_i}^{(n+1)} = \sum_{k=i}^{N} T_{E_k}^{(n)} + \sum_{k=1}^{i-1} T_{E_k}^{(n+1)} \qquad (2.4.9)$$

Because of the interdependence of the station times, the cycle time can only be evaluated approximately by

$$\Phi_{C_i}^{(n+1)}(s) \approx \prod_{k=i}^{N} \Phi_{E_k}^{(n)}(s) \cdot \prod_{k=1}^{i-1} \Phi_{E_k}^{(n+1)}(s) \qquad (2.4.10)$$

As seen in the MAC protocol and in equation 2.4.2 and 2.4.3, the token rotation timer (TRT) is not always reset and restarted upon arrival of the token; therefore the token rotation time in equations 2.4.2 and 2.4.3 is not equal to the cycle time. However, since on the average the TRT is reset upon token reception, a good approximation would be to equate them without having to alter the basic cycle time properties of the protocol. This is represented in the Laplace domain as

$$\Phi_{TRT_i}^{(n+1)}(s) \approx \Phi_{C_i}^{(n+1)}(s) \qquad (2.4.11)$$

It is now apparent that the queue state after the next token arrival $A^{(n+1)}_i$ for i=1,2,..,N can be found from the cycle time by developing the state equations of the imbedded Markov chain:

$$a_i^{(n+1)}(j) = d_i^{(n)}(j) \otimes q_{C_i}^{(n+1)}(j) \qquad (2.4.12)$$

where $\otimes$ stands for the discrete convolution of a finite and infinite distribution and $q^{(n+1)}_{C_i}(j)$ is the probability that j packets arrive at station i during the (n+1)st cycle. then

$$d(j) \otimes q(j) \begin{array}{ll} = \sum_{k=0}^{i} d(k) \cdot q(j-k) & 0 \le j < m \\ = \sum_{k=0}^{m} \sum_{l=m-k}^{\infty} d(k) \cdot q(l) & j = m \end{array} \qquad (2.4.13)$$

and

$$q_{C_i}^{(n+1)} = \int_0^\infty \frac{(\lambda_i t)^j}{j!} e^{\lambda_i t} f_{C_i}^{(n+1)}(t)\,dt \qquad j=0,1,2,\cdots,\infty \quad \textbf{(2.4.14)}$$

where $f^{(n+1)}_{ci}(t)$ is the distribution density function of $T^{(n+1)}_{ci}$. The probability $q^{(n+1)}_{ci}(j)$ can then be evaluated by using a two moments approximation for $T^{(n+1)}_{ci}$. The above analysis is too complex to allow a closed form expressions for the steady state distributions of $A^{(n)}_i$ and $T^{(n)}_i$ to be found. An iterative algorithm is used. As a start, the initial values for an empty system $A^{(0)}_i$ and $T^{(0)}_i$ are used. Then equations 2.4.1 to 2.4.12 yield an improved $T^{(1)}_i$, and equation 2.4.13 to 2.4.14 in turn yield a new $A^{(1)}_i$. These calculations are repeated until stability is reached, namely when

$$\sum_{i=1}^{N} \frac{E[A_i^{(n)}] - E[A_i^{(n-1)}]}{E[A_i^{(n-1)}]} < \epsilon \qquad \textbf{(2.4.15)}$$

The parameter ϵ is selected to determine the accuracy of the results. After the iteration, the steady state distributions of the queue lengths A_i at token arrival times and the cycle time T_C are known approximately.

FDDI II cycle structure and the circuit switch traffic is also being considered with the following assumptions. The number of circuit switch (CS) wideband channels remains fixed. Circuit switch connections are assumed to be static compared to packet switch (PS). The burstiness of the packet switch bandwidth is neglected.

This implies the following calculation of the packet service times:

$$T_{H_i}=T_0+T_{I_i} \tag{2.4.16}$$

where T_0 is the random variable of the transmission time of the packet overhead ℓ_0 with

$$E[T_0]=E[\ell_0]/B_{PS} \qquad c[T_0]=0 \tag{2.4.17}$$

where c is the coefficient of variation for the random variable T_0, and T_{Ii} denotes the transmission time of the information part of the packets with

$$E[T_{I_i}]=E[\ell_{I_i}]/B_{PS}, \qquad c[T_{I_i}]=c[\ell_{I_i}]\,. \tag{2.4.18}$$

The PS bandwidth B_{PS} depends on the amount of CS traffic. In the FDDI system it is

$$B_{PS}=B_{Sys}, \tag{2.4.19}$$

whereas with FDDI II it is

$$B_{PS}=B_{Sys}-B_0-n_{CS}\cdot B_{WBC} \tag{2.4.20}$$

where B_0 is the bandwidth loss due to cycle overhead, n_{CS} is the number of CS wideband channels (WBC) and B_{WBC} is the WBC bandwidth.

2.4.4 Results

A model of 25 stations connected to a ring with 100 Mbps system bandwidth and 100 km ring length is selected for the analysis. Every queue of figure 2.16 has a limited buffer space of $m_i = 5$, and all stations have the same priority threshold $T_{Prii} = T_{TTRT}$ with symmetric traffic. The packet information part length of 1 kbyte is negative exponentially distributed. The total offered traffic is varied by increasing the packet arrival rate. Four cases have been examined namely:

a) FDDI with $T_{TTRT} = 5$ ms

b) FDDI with $T_{TTRT} = 10$ ms

c) FDDI-II with $T_{TTRT} = 10$ ms and 25 % CS traffic

d) FDDI-II with $T_{TTRT} = 10$ ms and 50 % CS traffic

The results of the analysis for the above cases are reported by [TANG1] to have come comparable to simulation results obtained by a detailed discrete event simulation.

Figure 2.17 shows the mean cycle time $E[T_C]$ versus the normalized total offered traffic ρ which is calculated as

$$\rho=\sum_{i=1}^{N}\lambda_i\cdot E[\ell_{I_i}]/B_{Sys}. \qquad \textbf{(2.4.21)}$$

We can see that the top limit in figure 2.17 is that of TTRT used. The influence of the service discipline is negligible in the analysis. In cases c and d the mean cycle times increase to the maximum faster than the other cases because the residual maximum PS load is reduced according to equation 2.4.20 for FDDI-II.

Figure 2.18 shows the mean waiting time $E[T_W]$ versus ρ. It is observed that under heavy load, the waiting times gradually reach a saturation level determined by queue size and throughput of the stations.

Figure 2.19 shows the PS information throughput Γ_{PS} measured in Mbps given by

$$\Gamma_{PS}=\sum_{i=1}^{N}\Gamma_i=\sum_{i=1}^{N}(1-p_{L_i})\lambda_i E[\ell_{I_i}] \qquad (2.4.22)$$

At light load the throughput increases linearly with ρ. At higher load the information throughput reaches saturation. The PS throughput is reduced when the target rotation time is decreased and the amount of CS traffic is increased.

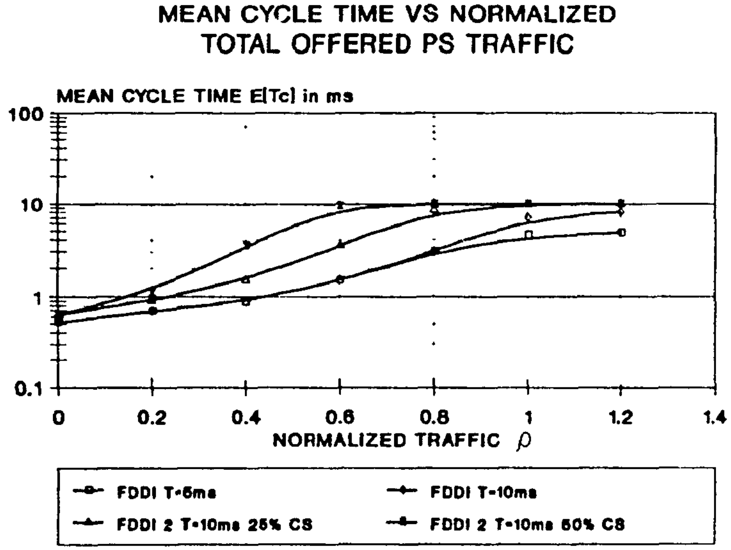


Figure 2.17 FDDI Mean Cycle Time

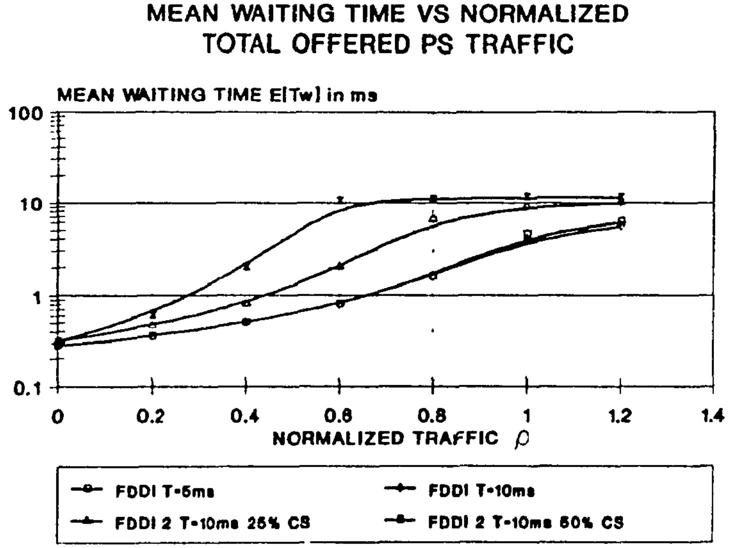


Figure 2.18 FDDI Mean Waiting Time

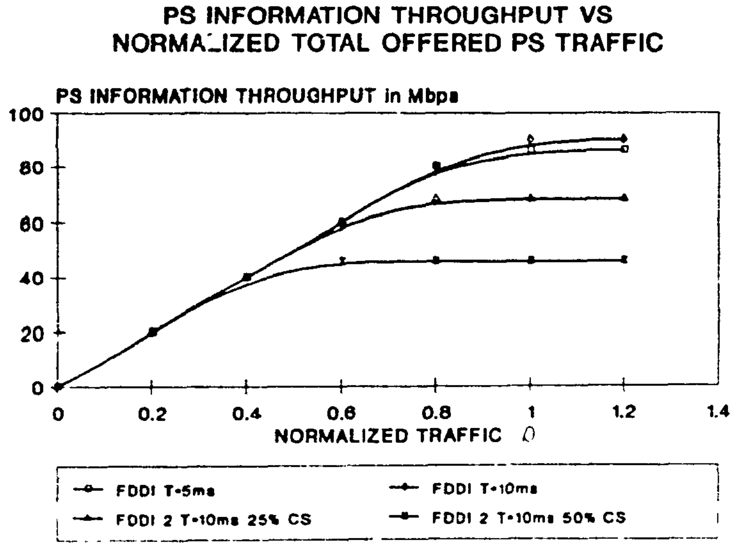


Figure 2.19 FDDI PS Throughput

2.5 References

[ANSI1] ANSI Standard X3T9.5, "FDDI Token Ring Media Access Control", 1988.

[BUX 1] W. Bux, "Token-Ring Local-Area Networks and their Performance," Proc. IEEE, vol. 77, pp.238-256, Feb. 1989.

[CHLA1] I. Chlamtac and A. Ganz, "A Multibus Train Communication (AMTRAC) Architecture for High-Speed Fiber Optic Networks," IEEE Journal on Selected Area in Communications, Vol.6, No.6, July 1988.

[CHLA2] I. Chlamtac and A. Gaz, "Design and Analysis of Very High Speed Network Architectures," IEEE Trans. Commun., Vol.36, pp.252-262, Mar. 1988.

[DYKE1] D. Dykeman and W. Bux "Analysis and Tuning of the FDDI Media Access Control Protocol," IEEE Journal on Selected Area in Communications, Vol.6, No.6, July 1988.

[FRAT1] L. Fratta, F. Borgonovo, and F. A. Tobagi, "The Expressnet: A Local Area Communication Network Integrating voice and data," Proc. Int. Conf. Performance Data Commun. Syst. and Their Appl., Paris. France, Sept. 14-16, 1981.

[KARO1] M. J. Karol and Gitlin, "High-Performance Optical Local and Metropolitan Area Networks: Enhancement of FDDI and IEEE 802.6 DQDB", IEEE Journal on Selected Areas in Communications, Vol. 8, No.8, Nov. 1990.

[LIMB1] J. O. Limb and C. Flores, "Description of Fasnet, a Unidirectional Local Area Communications Network," Bell Syst. Tech. Journal, Sept.1982.

[LIMB2] J. O. Limb, "Fasnet: A Proposal for a High Speed Local Network," Proc. Office Inform. Syst. Workshop, St. Maximin, France, Oct. 1981.

[METC1] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," Commun. Ass. Comput. Mach., Vol. 19, pp. 395-404, 1976.

[MUKH1] B. Mukherjee and J. S. Meditch, "Tne pi-Persistent Protocol for Unidirectional Broadcast Bus Networks," IEEE Transactions on Communications, Vol. 36, No. 12, Dec. 1988.

[MUKH2] B. Mukherjee and J. S. Meditch, "Integrating Voice with the pi-Persistent Protocol for Unidirectional Broadcast Bus Network," IEEE Transactions on Communications, Vol.36, No.12, Dec. 1988.

[PATI1] A. Patir, T. Takahashi, Y. Tamura, M. El Zarki, and A. Lazar, "An Optical Fiber-Based Integrated LAN for MAGNET"s Testbed Environment", IEEE Journal on Selected Areas in Communications, Vol. SAC-3, No.6, Nov. 1985.

[REED1] J. W. Reedy, "The TDM Ring- A Fiber Optic Transport System for Campus or Metropolitan Networks," IEEE Journal on Selected Areas in Communications, Vol. SAC-4, No.9, Dec. 1986.

[ROSS1] F. E. Ross, "FDDI - A Tutorial," IEEE Commun. Mag., Vol.24, pp.10-17, May 1986.

[STAL1] W. Stallings, "Handbook of COMPUTER COMMUNICATIONS Standards: Local Area Network Standards", Vol. 2, Macmillan Inc., 1988.

[TANG1] H. Tangemann and K. Sauer, "Performance Analysis of the Timed Token Protocol of FDDI and FDDI II", IEEE Journal on Selected Area in Communications, Vol. 9, No. 2, Feb. 1991.

[TOBA1] F. Tobagi and B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," in Proc. Local Area Comm. Network Symp., Boston, MA, May 1979.

[TOBA2] F. Tobagi, F. Borgonovo, and L. Fratta, "Expressnet: A High-Performance Integrated-Services Local Area Network," IEEE Journal on Selelected Area in Commun., Vol.SAC-1,No.5,Nov. 1983, pp.898-913.

[TOBA3] F. Tobagi, and M. Fine, "Performance of Unidirectional Broadcast Local Area Networks: Expressnet and Fasnet," IEEE Journal on Selected Area in Communications, Vol. SAC-1, No.5, Nov. 1983, pp.913-926.

CHAPTER 3

WAITING TIME ANALYSIS IN A SINGLE BUFFER DQDB NETWORK

3.1 Introduction

The DQDB MAN network consists of two high speed unidirectional buses carrying information in opposite directions and serving stations connected to both buses. Simulation studies [CONT1] [DAVI1] have revealed the unfairness of DQDB in accessing the network from stations at different positions on the buses. It is shown that the stations most upstream to the sender and nearer the network controller which generates the transmission slots, have more chances than the downstream stations when the number of transmission requests increases. In order to look at this problem in an analytical way, it is necessary to formulate the waiting time of a station and derive from it the unfairness problem with respect to the position of the station.

Bisdikian in "Waiting Time Analysis in a Single Buffer (802.6) Network" [BISD1] has calculated the data unit waiting time characteristics of a specific station in the network under the assumption that the station cannot queue more than one data unit at a time. Analysis was used to determine the unfairness of DQDB and to estimate to what extend the "destination release of slots" can improve the network performance. We will describe, in what follows, Bisdikian approach.

3.2 DQDB Analysis

In the DQDB network each station is in one of two states. It is idle when there is nothing to transmit and active otherwise. In the idle state, the request counter (RQ_CTR) is incremented for each request it receives on the reverse channel and it is decreased by one for every empty slot that passes the station in the forward direction (downstream) as shown in figure 3.1.

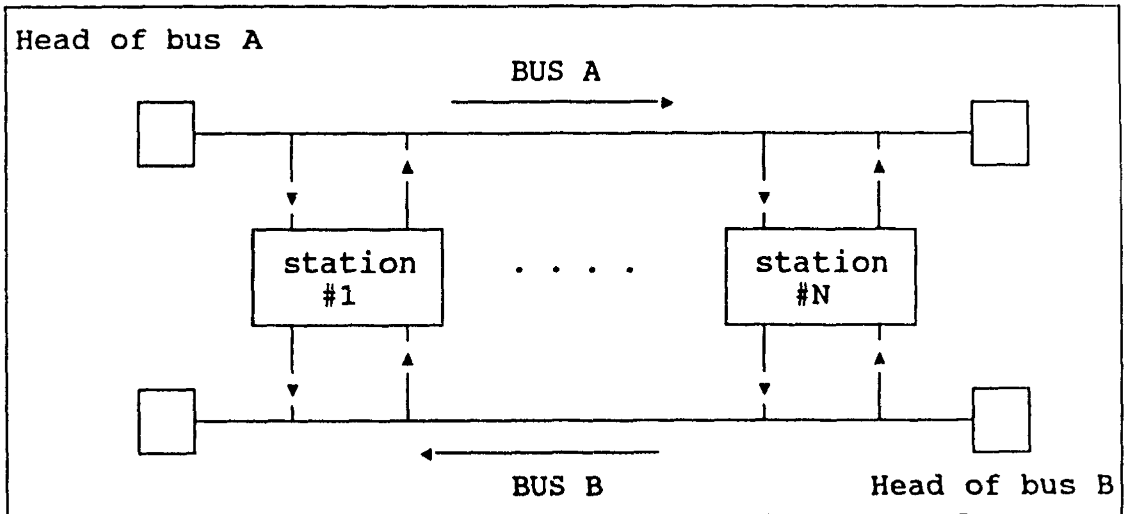


Figure 3.1 The DQDB network topology.

In the active state, when the station has something to transmit, it transfers the content of the RQ_CTR to the countdown counter (CD_CTR) and resets the RQ_CTR to 0[1]. It then sends a request upstream in the reverse channel (REQ set to 1). The CD_CTR is decremented by one for every empty slot that passes the station in the forward channel (downstream) until it reaches 0. At this instant, the station transmits a data unit waiting in the queue into the first empty slot of the forward channel. Meanwhile, the RQ_CTR continues to be incremented for all requests received on the reverse channel. In order to determine the waiting time of data units we note that the number of requests in the RQ_CTR when a data unit arrives for transmission is in direct relationships with the waiting time of the data unit. So we analyze the statistics of the number of requests at the RQ_CTR when the data unit arrives.

A single station is considered for the analysis with the network partitioned into an L_NET and an R_NET by aggregating together all stations lying to the left and right of the tagged station (figure 3.2).

[1] **The RQ_CTR reset to zero is one of the implementations that is used in DQDB protocol. As far as this analysis is concerned there is no difference which method is used since the result is the same.**

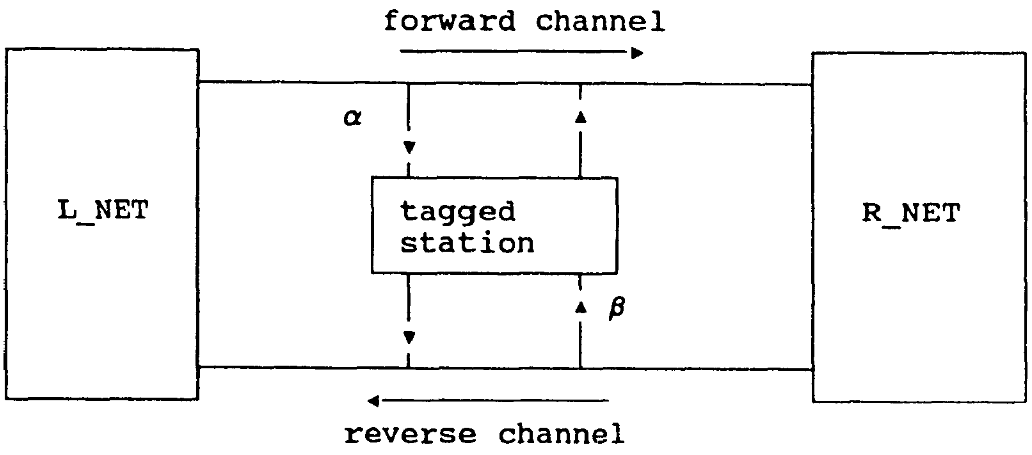


Figure 3.2 DQDB tagged station and its L_NET and R_NET.

It is assumed that each slot coming from the L_NET in the forward channel and passing in front of the tagged station is busy with probability α, independently per slot. Furthermore, each slot coming from the R_NET and passing in front of the tagged station in the reverse channel contains a request with probability β, independently per slot and independently from the busy slot of the forward channel. It should be noted that the L_NET and R_NET generate data units (not slots) at rates per slot equal to probabilities α and β respectively.

The tagged station is assumed to have a data unit transmission buffer of size one. Thus whenever the user is active, only one data unit awaits transmission. The arrival rate is λ data units per seconds. The number of slots between the transmission of a data unit and the arrival of the next one is a number T of slots, that is independent of any process in the network and with a probability mass function:

$$p_n = Pr(T=n) = e^{-n\lambda}(1-e^{-\lambda}) \qquad (3.1)$$

for $\lambda>0$, n=0, 1, 2, ...
The first term is the probability that the data unit arrives at the nth slot and the second term is the probability that no data unit is in the system.

The virtual request counter (VRQ_CTR) is useful in describing the status of the station at any moment in time. So

$$\begin{aligned} \text{VRQ\_CTR} &= \text{CT\_CTR} + 1, && \text{when the user is active} \\ &= \text{RQ\_CTR} && \text{when the user is idle} \end{aligned} \qquad (3.2)$$

Now when the RQ_CTR is 0 and a data unit arrives in the station, the VRQ_CTR is 0 if the slot that passes on the forward channel is not busy, on the other hand if the slot is set busy then the VRQ_CTR is 1. So the waiting time of a packet is the time it takes for the VRQ_CTR to be reduced to 0, following the arrival of a data unit in the station.

The data unit waiting time is now analyzed. We let F_n denote the value of the VRQ_CTR at the instant of the nth data unit arrival. Since the busy and request slots are memoryless in other words not dependent on previous states, the sequence $\{F_n;\ n\geq1\}$ forms a Markov chain embedded at the instances of data unit arrivals at the tagged station. During the subsequent development we prove that the chain is homogenous, irreducible, and aperiodic with the state space being the set N_0 of nonnegative integers. The transition matrix of the Markov chain $\{F_n;\ n\geq 1\}$ is defined as

$$\Psi=[\psi_{ij}]=[Pr(F_{n+1}=j|F_n=i)] \tag{3.3}$$

where $i,j \in N_0$

We let K_n denote the value of the RQ_CTR at the first slot following the transmission of the nth data unit. Again because of the memoryless property of the busy and request slots and given that $F_n = i$, we can say that K_n depends only on i and is independent of n. The next arrival of the data unit F_{n+1} likewise depends only on the last event being k if $K_n = k$ and is independent of n.

We then define

$$\begin{aligned}\sigma_{ik}&=Pr(K_n=k|F_n=i)\\ \tau_{kj}&=Pr(F_{n+1}=j|K_n=k)\end{aligned} \tag{3.4}$$

for $i,\ j,\ k \in N_0$

We can then write each element of the state transition matrix of the Markov chain $\{F_n;\ n\geq 1\}$ as

$$\psi_{ij}=\sum_{k=0}^{\infty}\sigma_{ik}\tau_{kj}. \tag{3.5}$$

Its generating function is

$$G_{\psi}(z;i)=\sum_{j=0}^{\infty}\psi_{ij}z^{j}, \qquad (3.6)$$

for i=0, 1, 2, ... and $|z|\le 1$.

In order to obtain this function we will determine the conditional waiting time when the station is active and a random walk analysis when the station is idle. With these elements we will be able to determine the steady state of the Markov Chain $\{F_n;n\ge 1\}$.

A data unit that arrives at a station in a slot such that the VRQ_CTR = i has a waiting time W(i). The VRQ_CTR will either decrease by one with probability $1-\alpha$, or remain at the same level with probability α.

The generating function for W(i) is $G_W(z;i)$ and it is equal to

$$G_W(z;i)=1 \text{ if } i=0 \text{ and}$$

$$G_W(z;i)=(1-\alpha)zG_W(z;i-1)+\alpha zG_W(z;i), \qquad (3.7)$$

if $i\ge 1$.

Solving the recursion yields the following:

$$G_W(z;i)=\left(\frac{(1-\alpha)z}{1-\alpha z}\right)^{i}, \qquad (3.8)$$

for $|z|\le 1$, i= 0, 1, 2, ...

From the properties of the z-transform this equation implies that W(i) is distributed as the sum of i independent geometrically distributed random variables with parameter $1-\alpha$. Instead of inverting $G_W(z;i)$ which can be quite tedious, we can derive directly W(i) by noting that it has a negative binomial distribution with

$$Pr(W(i)=n)=\binom{n-1}{i-1}(1-\alpha)\alpha^{n-i} \tag{3.9}$$

with $1\le i\le n$. Noting that $Pr(W(0)=0)=1$.

The generating function $G_W(z;i)$ is useful in deriving easily the

mean conditional waiting time and its variance namely

$$E[W(i)]=\frac{i}{1-\alpha}$$

$$var[W(i)]=\frac{i\alpha}{(1-\alpha)^2} \tag{3.10}$$

In order to calculate σ_{ik} which is defined above, it is necessary to first find the probability of registering n new requests in t slots namely $s_t(n)$. Again due to the memoryless property of the request occurrence per slot, we can state

$$s_t(n)=\binom{t}{n}\beta^n(1-\beta)^{t-n} \tag{3.11}$$

for $0\le n\le t$,

The generating function is

$$S(z;t)=\sum_{n=0}^{t} s_t(n)\, z^n=(1-\beta+\beta z)^{t}, \qquad (3.12)$$

for $|z| \leq 1$.

Now σ_{ik} which is defined as $Pr(K_n=k \mid F_n=i)$ can be derived as

$$\sigma_{ik}=\sum_{t=\max(i,k-1)}^{\infty} s_{t+1}(k)\, Pr(W(i)=t) \qquad (3.13)$$

This is the sum of probabilities that there are k new requests in t+1 slots with conditional waiting time in t slots for all possible values of t. Closed form solution is not possible for this expression. So we define a random variable conditioned on i Y(i) such that $(Y(i)=k)=\sigma_{ik}$ and the generating function is

$$G_Y(z;i)=\sum_{t=i}^{\infty} (1-\beta+\beta z)^{t+1}\binom{t-1}{i-1}(1-\alpha)^{i}\alpha^{t-i} \qquad (3.14)$$

for $|z| \leq 1$,

which after manipulation becomes

$$G_Y(z;i)=(1-\beta+\beta z)\left(\frac{(1-\alpha)\,(1-\beta+\beta z)}{1-\alpha\,(1-\beta+\beta z)}\right)^{i} \qquad (3.15)$$

for $|z| \leq 1$.

This expression will be used later on to express the steady state of the markov chain.

We shall now consider an expression which describes the state of the station when it is idle and no transmission of data unit is waiting. The VRQ_CTR is a representation of the variations of the RQ_CTR. Let us assume that right after the transmission of its last data unit the VRQ_CTR of the station is equal to k and no new data unit arrives. Let $Z_n(k)$ be the VRQ_CTR indication at the nth slot, starting from k. Then, $Z_n(k)$ evolves according to the following random walk equations:

When it is right after the transmission in other words n is zero then $Z_0(k)=k$ with probability 1 for $k\epsilon N_0$ as defined above. i.e.

$$Z_0(k)=k \qquad k\epsilon N_0$$

In a general case

$$Z_{n+1}(k)=\max(0,Z_n(k)+I_\epsilon^\delta(n)) \qquad (3.16)$$

for n=0, 1, 2, ...,

and where I_ϵ^δ is a ternary, independent identically distributed sequence, independent of any other process and assumes the following values

$$I_\epsilon^\delta(n)=\begin{array}{rl} 1 & \to w.p.\ \delta=\alpha\beta \\ 0 & \to w.p.\ 1-\delta-\epsilon \\ -1 & \to w.p.\ \epsilon=(1-\alpha)(1-\beta) \end{array} \qquad (3.17)$$

Some definition are necessary in order to give a solution to this equation.

Let $\pi_j(n;k)$ = Pr $(Z_n(k)=j)$, n, j, k ϵ N_0 then from the above expressions

$$\pi_j(0;k)=\begin{array}{ll} 1, & \to j=k \\ 0, & \to otherwise \end{array}$$

$$\pi_j(n+1;k)=\begin{cases}(1-\delta)\pi_0(n;k)+\epsilon\pi_1(n;k), & j=0\\ \delta\pi_{j-1}(n;k)+(1-\delta-\epsilon)\pi_j(n;k)+\epsilon\pi_{j+1}(n;k), & j\geq 1\end{cases}$$

(3.18)

and let the generating function that corresponds to $Z_n(k)$ be

$$G_k(z;n)=\sum_{i=0}^{\infty}\pi_i(n;k)z^i, \tag{3.19}$$

for $|z|\leq 1$ and k= 0, 1, 2, ..., then

$$\begin{aligned}G_k(z;n+1)&=\sum_{i=0}^{\infty}\pi_i(n+1;k)z^i\\ &=[\pi_0(n;k)(1-\delta)+\pi_1(n;k)\epsilon]\\ &+[\pi_0(n;k)\delta+\pi_1(n;k)(1-\delta-\epsilon)+\pi_2(n;k)\epsilon]z\\ &+\cdots\\ &+[\pi_{i-1}(n;k)\delta+\pi_i(n;k)(1-\delta-\epsilon)+\pi_{i+1}(n;k)\epsilon]z^i\\ &+\cdots.\end{aligned}$$

(3.20)

This further resolves into the following recursion

$$G_k(z;n+1)=(\delta z+1-\delta-\epsilon+\epsilon z^{-1})G_k(z;n)+\pi_0(n;k)(1-z^{-1})\epsilon.$$

(3.21)

This linear difference equation has the following solution

$$\begin{aligned}G_k(z;n)=(\delta z+1-\delta-\epsilon+\epsilon z^{-1})^n z^k+\epsilon(1-z^{-1})\\ \cdot\sum_{i=0}^{n-1}(\delta z+1-\delta-\epsilon+\epsilon z^{-1})^{n-1-i}\pi_0(i;k),\end{aligned}$$

for n= 0, 1, 2,... and $G_k(z;0)=z^k$. (3.22)

Note that the sequence $\pi_0(n;k)$ has no closed expression but it can be calculated numerically.

Now suppose that the sequence $\{Z_n(k);\ n\geq 0\}$ stops its progress at the nth step (when a data unit arrives) with probability p_n defined above as $e^{-n\lambda}(1-e^{-\lambda})$. For R(k) being the value of $Z_n(k)$ when the sequence stops

$$R(k) = \sum_{n=0}^{\infty} Z_n(k)\,(1-e^{-\lambda})\,e^{-n\lambda} \tag{3.23}$$

Since $\pi_j(n;k)=\Pr(Z_n(k)=j)$ then the pmf of R(k) is

$$\overline{\pi_j}(k) = (1-e^{-\lambda})\sum_{n=0}^{\infty} \pi_j(n;k)\,e^{-n\lambda}, \tag{3.24}$$

and the generating function is

$$G_R(z;k) = E[z^{R(k)}] = \sum_{n=0}^{\infty} G_k(z;n)\,(1-e^{-\lambda})\,e^{-n\lambda} \tag{3.25}$$

This equation is also equal to

$$G_R(z;k) = \sum_{n=0}^{\infty} \overline{\pi_n}(k)\,z^n \tag{3.26}$$

Combining the solution of $G_k(z;n)$ with this equation we get

$$G_R(z;k) = \frac{(1-e^{-\lambda})\,z^k + \epsilon(1-z^{-1})\,e^{-\lambda}\overline{\pi_0}(k)}{1-(\delta z+1-\delta-\epsilon+\epsilon z^{-1})\,e^{-\lambda}} \tag{3.27}$$

Using the above expressions we can formulate the probability τ_{kj} by the following

$$\tau_{kj} = \begin{matrix} \overline{\pi_{j-1}}(k), & \to j\ge 2, \\ \alpha\overline{\pi_0}(k), & \to j=1, \\ (1-\alpha)\overline{\pi_0}(k), & \to j=0. \end{matrix} \tag{3.28}$$

We can now proceed with the analysis of the Markov chain $(F_n;n\ge 1)$ and derive its steady state probabilities.

We carry on from equation 3.6 now that we have developed the terms that make it up. Therefore

$$G_{\psi}(z;i)=\sum_{j=0}^{\infty}\left[\sum_{k=0}^{\infty}\sigma_{ik}\tau_{kj}\right]z^{j}$$
$$=\sum_{k=0}^{\infty}\sigma_{ik}\left[z\left(\sum_{j=0}^{\infty}\overline{\pi_{j}}(k)\,z^{j}\right)+\overline{\pi_{0}}(k)\,(i-\alpha)\,(1-z)\right],\ \rightarrow|z|\leq 1.$$

(3.29)

The inner summation is in fact $G_R(z;k)$; we thus have

$$G_{\psi}(z;i)=\sum_{k=0}^{\infty}\sigma_{ik}\left[z\left(\frac{(1-e^{-\lambda})\,z^{k}+\epsilon(1-z^{-1})\,e^{-\lambda}\overline{\pi_{0}}(k)}{1-(\delta z+1-\delta-\epsilon+\epsilon z^{-1})\,e^{-\lambda}}\right)+\overline{\pi_{0}}(k)\,(i-\alpha)\,(1-z)\right],$$

for $|z| \leq 1$ **(3.30)**

After manipulating equation 3.30, equation 3.15 was recognized as a term in it. The result of this manipulation is

$$G_{\psi}(k;i)=\frac{[(1-e^{-\lambda})\,G_{Y}(z;i)+(1-\alpha)\,(1-\beta)\,(1-z^{-1})\,e^{-\lambda}\gamma_{i}]\,z+\gamma_{i}(1-\alpha)\,(1-z)}{1-[\alpha\beta z+\alpha+\beta-2\alpha\beta+(1-\alpha)\,(1-\beta)\,z^{-1}]\,e^{-\lambda}}$$

where **(3.31)**

$$\gamma_{i}=\sum_{k=0}^{\infty}\sigma_{ik}\overline{\pi_{0}}(k)$$

which is the probability that starting at state i, the next arrival finds no outstanding requests. This probability is easily obtained numerically. Equation 3.31 gives the transition probability of the Markov chain $\{F_n;n\geq 1\}$. This equation can also be used to prove the ergodicity of the Markov chain provided that

i.e. the average number of requests per slot in the reverse

$$\beta<1-\alpha \tag{3.32}$$

channel is less than the average number of empty slots in the forward channel.

We are now able to find $\underline{\phi}$ the steady state probability vector of the Markov chain $\{F_n;n\geq 1\}$, not conditioned on any parameter. So the terms of the vector are

$$\phi_i=\sum_{j=0}^{\infty}\phi_j\psi_{ji}, \quad \neg i=0,1,2,\cdots, \tag{3.33}$$

and the associated generating function is

$$\begin{aligned} G_F(z) &=\sum_{i=0}^{\infty}\phi_i z^i, \neg|z|\leq 1. \\ &=\sum_{j=0}^{\infty}\phi_j\sum_{i=0}^{\infty}\psi_{ji}z^i \\ &=\sum_{j=0}^{\infty}\phi_j G_{\psi}(z;j) \\ &=\frac{\left[(1-e^{-\lambda})(1-\beta+\beta z)\cdot G_F\left(\frac{(1-\alpha)(1-\beta+\beta z)}{1-\alpha(1-\beta+\beta z)}\right)+(1-\alpha)(1-\beta)(1-z^{-1})e^{-\lambda}\theta\right]z+\chi}{1-[\alpha\beta z+\alpha+\beta-2\alpha\beta+(1-\alpha)(1-\beta)z^{-1}]e^{-\lambda}} \end{aligned}$$

where (3.34)

$$\chi=\theta(1-\alpha)(1-z)$$
$$\theta=\frac{\phi_0}{(1-\alpha)}$$

where θ is the probability that an arriving data unit finds no outstanding requests. The moments of F may be obtained by differentiating equation 3.34.

The mean value of the VRQ_CTR at the arrival of a data unit is

$$E[F]=\frac{1-\alpha}{1-\alpha-\beta}\left[\beta+\frac{e^{-\lambda}}{1-e^{-\lambda}}[\alpha\beta-(1-\alpha)(1-\beta)(1-\theta)]+1-\theta(1-\alpha)\right]$$

(3.35)

which means the number of requests that were made during the waiting time plus transmission time of the previous data unit, plus the number of requests not satisfied that were made during the idle period plus one, unless the data unit arrives at an empty slot and CD_CTR=0.

If we let $\{I(i);i\geq 0\}$ be a sequence of mutually exclusive indicator random variables with $Pr(I(i)=1)=\phi_i$. Then the conditional waiting time is

$$W=\sum_{i=0}^{\infty} W(i)\,I(i), \tag{3.36}$$

and the generating function is

$$\begin{aligned} G_W(z)=E[z^W]&=\sum_{i=0}^{\infty}\phi_i\left(\frac{1-\alpha)z}{1-\alpha z}\right)^i \\ &=G_F\left(\frac{(1-\alpha)z}{1-\alpha z}\right) \end{aligned} \tag{3.37}$$

where $|z|\leq 1$.

Note that $G_W(0)=G_F(0)$ and ϕ_0 is the probability that a data unit has no waiting time. Taking the derivative of equation 3.37 we obtain the mean waiting time

$$E[W]=\frac{E[F]}{1-\alpha}, \tag{3.38}$$

The probability mass function (pmf) of the random variable F, for various values of the parameter α, β and λ are shown in figure 3.3 with the mean waiting time E[W] for each case.

It is observed that as busy or request probabilities are increased there is a shift to the right of the pmf of the virtual requests, which reflects also an increase of the waiting time of the data unit. This is caused by an increase of requests queued ahead of an arriving data unit (increase in β) or a decrease of the effective capacity of the forward channel (increase in α) or a combination of the two. The condition for system stability whereby $\beta \geq 1-\alpha$ simply means that the L_NET must allow enough empty slots to satisfy the traffic generated by the R_NET. It should be noted that an increase in α or β is not only caused by an increase of traffic generated by each station in the L_NET or R_NET respectively but also when the tagged station places itself nearer the head of the reverse or forward channel respectively. Since $\beta \geq 1-\alpha$ then an increase in α requires a decrease in β. As a consequence a single station in the L_NET can shut down the entire R_NET, not even allowing requests to be sent by any R_NET stations. Thus the unfairness of the DQDB network is once again shown but this time analytically.

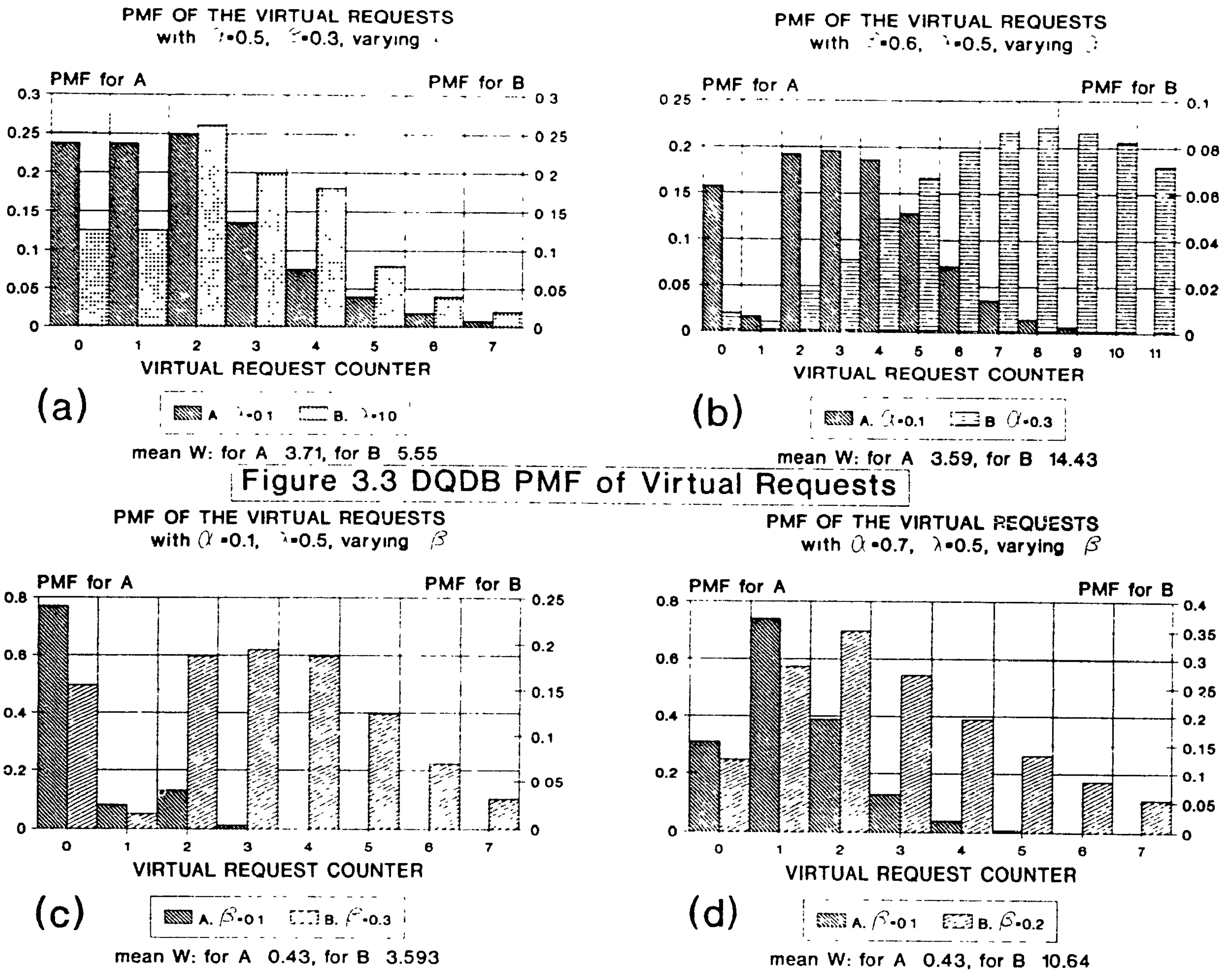


Figure 3.3 DQDB PMF of Virtual Requests

3.3 References

[BISD1] C. C. Bisdikian,"Waiting Time Analysis in a Single Buffer DQDB (802.6) Network" IEEE Journal on Selected Areas in Communications, Vol.8. No.8, October 1990.

[BUDR1] Z. L. Brudrikis, J. L. Hullett, R. M. Newman, D. Economou, F. M. Fozdar, and R. D. Jeffery, "QPSX: A Queue Packet and Synchronous Circuit Exchange," in Proc. 8th Int. Conf. Comput. Commun. ICCC'86, Munich, West Germany, Sept. 15-19, 1986, pp. 288-293.

[CONT1] M. Conti, E. Gregory, and L. Lenzini,"DQDB media access control protocol: Performance Evaluation and unfairness analysis," in Proc. 3rd IEEE Workshop on Metropolitan Area Networks, San Diego, CA, Mar. 28-30 1989, pp. 7.3.1-7.3.33.

[DAVI1] P. Davids and T. Welzel,"Performance analysis of DQDB based on simulation," in Proc. 3rd IEEE Workshop on Metropolitan Area Networks, San Diego, CA, Mar. 28-30 1989, pp. 7.6.1-7.6.15.

[HUBE1] M. N. Huber, K. Sauer, and W. Schodl, "QPSX and FDDI-II Performance Study of High Speed LAN's," in Proc. 6th Europ. Fibre Optic Commun. Local Area Networks Expo. EFOC/LAN 88, Amsterdam, The Netherlands, June 29-July 1, 1988, pp. 316-321.

[IEEE1] IEEE Standard P802.6 Working Group, "Proposed Standard: Distributed Queue Dual Bus (DQDB)- Metropolitan Area Network," Unapproved Draft D11, Dec.1989.

[NEWM1] R. M. Newman and J. L. Hullet, "Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX," in Proc. 8th Int. Conf. Comput. Commun. ICCC'86, Munich, West Germany, Sept. 15-19, 1986, pp. 288-293.

[NEWM2] R. M. Newman, Z.L. Budrikis, and J. L. Hullet, "The QPSX MAN," IEEE Commun. Mag., Vol 26, No.4, pp.20-28, Apr. 1988.

[WAIN1] N. Wainwright and A. Myles, " A Comparison of the Delay Characteristics of the FDDI and IEEE 802.6 MAC Layer Protocols," in Proc. 3th IEEE Workshop on Metropolitan Area Networks, San Diego, CA. Mar. 28-30, 1989, pp.7.5.1-7.5.9.

[ZUKE1] M. Zukerman, "On Packet Switching Capacity in Qpsx," in Proc. IEEE/IECE Global Telecommun. Conf. GLOBECOM'87, Tokyo, Japan, Nov. 15-18, 1987, pp. 45.2.1-45.2.5.

[ZUKE2] M. Zukerman, "Queueing Performance of QPSX," in Proc. 12th Int. Teletraffic Congress ITC 12, Torino, Italy, June 1988, pp. 2.2B.6.1-2.2B.6.7.

[ZUKE3] M. Zukerman, "Overload Control of Isochronous Traffic in QPSX," in Proc. IEEE Global Telecommun. Conf. Exhibition GLOBECOM'88, Holywood, FL, Nov.28-Dec.1, 1988, pp.38.3.1-38.3.5.

CHAPTER 4

DQDB REQUIREMENTS: SERVICE INTEGRATION AND RELIABILITY ISSUES FOR DUAL BUS METROPOLITAN NETWORKS

4.1 Introduction

The Metropolitan Area Network (MAN) being adopted in the IEEE 802.6 Standard is based on the concept of distributed queueing. The distributed queueing technique under consideration, the distributed queue dual bus (DQDB) was first proposed for inclusion in the IEEE 802 standard by Telecom Australia and its subsidiary QPSX Communications [NEWM1].

The DQDB system can accommodate both voice and data traffic. Current DQDB protocols do not however, include complete specifications for broadband services. Broadband services, those requiring a data rate of 1.544 Mbps or more typically include videotelephony, videoconferencing, high-fidelity sound broadcasting, television, high-speed high-definition facsimile and high-speed data transmission. In this chapter, the first four examples are referred to as "video services", the last two as "broadband data services". Data transmission at rates of 9.6 Kbps and 64 Kbps is classified as "narrowband data service". The DQDB system uses the unidirectional bus structure [NEWM2] shown in figure 4.1. Distributed queueing is carried out by the media access control (MAC) protocol implemented in the access units (AUs).

The dual bus consists of two parallel unidirectional buses connected to every access unit in the network. Frames in each bus flow in the opposite directions. Both ends of both buses terminate on the same network controller (NC). Although the resulting topology resembles a ring, the dual bus network has nothing more in common with a ring network. The frames generated by the NC at one end flow on the bus and are removed by the NC when they reach the other end. Following a specific protocol, AUs can read or write to frames flowing on either bus without delaying the flow.

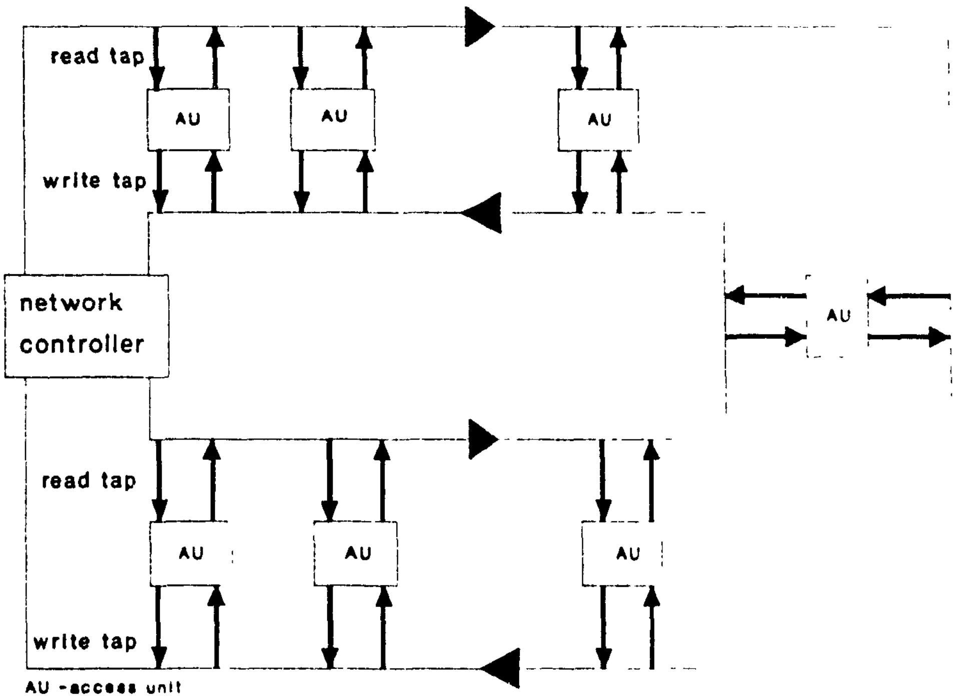


Figure 4.1 A Distributed Queue Dual Bus Network

The present study analyzes a new and expanded DQDB scheme and proposes a protocol that can handle service integration compatible with the asynchronous transfer mode (ATM) used in broadband integrated services digital networks [CCIT1]. This new scheme guarantees that a virtual circuit for each voice and video call is available on demand and for the duration of the call [KADO1]. Quality of voice and video services are also guaranteed. Delay jitter, for instance, is minimal. Each voice or video call occupies the same slot in successive frames for the duration of the call. Call establishment time for voice and video , a function of the frame size, is also minimal. This system does not, however, use the silent intervals in voice calls for other calls; voice slots are dedicated for the duration of the call. To do so, the network controller would have to be programmed with an algorithm implementing, say, the time assignment speech interpolation (TASI) technique [WILL1]. In addition to circuit switching for video and voice services, packet switching is provided for services such as broadband data and narrowband data.

In the DQDB network, the network controller generates frames transmitted in opposite directions on each bus. Under the proposed scheme, each frame is divided into four subframes, one for each of the four types of service: video, voice, wideband data and narrowband data. The subframes have slots of a specific size for each service. Regardless of the service, however, slot size is a multiple of the narrowband data slot. All slots have the same header format. Furthermore, each video slot is structured in such a way that it can include a whole number of slots for either voice, wideband data or narrowband data but not a mix of services. Similarly, each voice slot is structured in such a way that it can include a whole number of slots for either wideband data or narrowband data but not a mix of both. This utilization of voice and video capacity depends on availability; voice and video services have priority in the use of their respective subframes. Such sharing of subframes is known as the "frame movable boundary" technique [KRAI1]. Wideband data and narrowband data subframes, it should be noted, have fixed boundaries, i.e., can only be used for their assigned service.

Since both broadband data and narrowband data can use video or voice slots, contention may occur between the two data services. Various priority schemes are studied with the object of minimizing delay. Voice service, it must be stressed, has priority over the data services for use of any available video slots.

The reliability and performance of the expanded DQDB protocol are also studied. The type of failure considered occurs at the AU-bus interface and are caused by intermittent and/or stuck-at bit errors when reading or writing the control field in the frame header. In the model used to analyze the performance reliability of the DQDB system, each AU causing protocol errors is represented by a node and all healthy AUs are grouped together and represented by a single node as far as traffic flow and fault types are concerned [KAD02].

Each AU services slots on both buses in accordance with the DQDB protocol. A faulty AU will exhibit an abnormal service. The presence of faulty AUs affects the healthy AU since an AU stealing a slot due to its error reduces the number available from the fixed total number of slots. This model can be analyzed as a network of M/M/1 queues. Nodes are described analytically within each dimension in the network of queues. If there is only one faulty AU then the network of queues is two dimensional, one dimension for the faulty AU and the other for the healthy AUs. The dimension of the network of queues is therefore equal to the number of faulty AUs plus one.

4.2 Frame Structure

Each frame generated by the network controller comprises four subframes, one for each type of service (video, voice, wideband data and narrowband data). There is no reason why more subframe cannot exist. In case of a new service implementation, a corresponding subframe can be added with all its required features but following the constraints stated further on. The Network Controller and the Access Units handling this new service will be updated with the new changes. The other AUs may ignore this subframe since the code describing the service in the slot does not correspond to the service being supported by them. Since there are two types of services, synchronous for video and voice and asynchronous for data, the software that is added will simply be one of the two fonctions. The other change will be in the subframe size for this additional service.

Each subframe is divided into a number of slots, whose size depends on the service type. Slots of all types, however, have the same header structure, called the "access control field" (ACF).

Slots for the data services and for establishing video and voice services contain (figure 3.2):

- an access control field (ACF),
- a calling address field,
- a called address field,
- a parameter field, and
- a data field.

ACF	CALLING ADDRESS	CALLED ADDRESS	PARAMETERS	DATA

Figure 4.2 Asynchronous slot

The coding of these fields is beyond the scope of this study, whose intent is to design an efficient DQDB protocol for improved performance and reliability of a multiservice MAN.

The address fields contain the addresses of the calling and called AUs. The length of these fields is fixed and depends on the network size.

The ACF contains essential information used by the DQDB protocol. The ACF of the proposed approach has a length of 32 bits distributed as in figure 4.3.

	F1	F2	BUSY	CONTROL	REQ1	REQ2	REQ3	REQ4	
b position	1	2	3	8	13	18	23	28	32

Figure 4.3 ACF Format

Where

F1, F2 two bits for frame synchronization,

BUSY - five bits indicating that the slot carries information,

CONTROL - five bits identifying the slot type and function,

REQ1 - five bits for the request of the narrowband data service,

REQ2 - five bits for the request of the wideband data service,

REQ3 - five bits for the request of the voice service (used in call establishment), and

REQ4 - five bits for the request of the video service (used in call establishment)

Note: Five bits are used for the BUSY field and each REQ field in the case of the quintuple modular redundancy (see section 4.8). In the case of triple modular redundancy, the BUSY and REQ fields are only three bits long giving an ACF 22 bits long.

The frame synchronization bits are generated by the network controller. When an AU uses a slot to carry information, it sets the BUSY bits to indicate that the slot is no longer available. An AU with information to transmit must request a free slot before it is allowed to use one. The request is made by setting the REQ bits corresponding to the service that the AU wants to use, i.e., REQ1, REQ2, REQ3 or REQ4. The REQ bits can belong to the ACF of any type of slot. For example, a wideband request for a free slot can be made by setting the REQ2 bits in the ACF of a video slot, voice slot, wideband slot or narrowband slot. Which bus is used to make a request is determined by the distributed queueing protocol [BUDR1] and will be described later. The BUSY bits and REQ bits are inherent to the distributed queueing protocol. The CONTROL field in the ACF gives the state of the slot. Since the CONTROL field has five bits, it can represent 32 different states. The different states that are possible depend on the slot category. The data slot in the narrowband data service subframe can have only one state. The coding in the CONTROL field represents that state, i.e., NARROWBAND DATA SERVICE. Similarly, the data slot in the wideband data service subframe can only have one state, the CONTROL field signifying WIDEBAND DATA SERVICE. Both data services may have different types and accordingly DATA field sizes, throughput, and other parameters specific to the service. This information is conveyed in a PARAMETER field between the CONTROL and DATA fields.

When not needed by their intended service, voice slots can be used by either data service and video slots by the data or voice services. The coding of the CONTROL field is hence more elaborate. All possible significances of the CONTROL field for each category of slot are listed in table 3.1.

TABLE 4.1 CONTROL Field Significance

Category of Slot	CONTROL Field Signification
narrowband data	- NARROWBAND DATA SERVICE
wideband data	- WIDEBAND DATA SERVICE
voice	- NARROWBAND DATA SERVICE - NARROWBAND DATA FULL - WIDEBAND DATA SERVICE - WIDEBAND DATA FULL - VOICE SERVICE - VOICE CALL ESTABLISHMENT - VOICE CALL TERMINATION
video	- NARROWBAND DATA SERVICE - NARROWBAND DATA FULL - WIDEBAND DATA SERVICE - WIDEBAND DATA FULL - VOICE SERVICE - VOICE FULL - VIDEO SERVICE - VIDEO CALL ESTABLISHMENT - VIDEO CALL TERMINATION

The use of the CONTROL and other ACF fields will be described in subsequent sections dealing with the protocol.

4.3 Transfer of Narrowband and Wideband Data

An AU that wants to send information to another AU first requests a free slot on the bus towards the destination AU. This direction is known as the "downstream direction" with respect to the sender. The request is sent on the other bus, which accesses AUs located away from the destination AU, i.e. in the "upstream direction" with respect to the sender [BUDR1]. figure 4.4 shows AU_2 transmit to AU_i. AU_2 sends a REQuest upstream (on line B) to AU_1. When its assigned free slot arrives (on line A), AU_2 uses the slot to transmit downstream to AU_i.

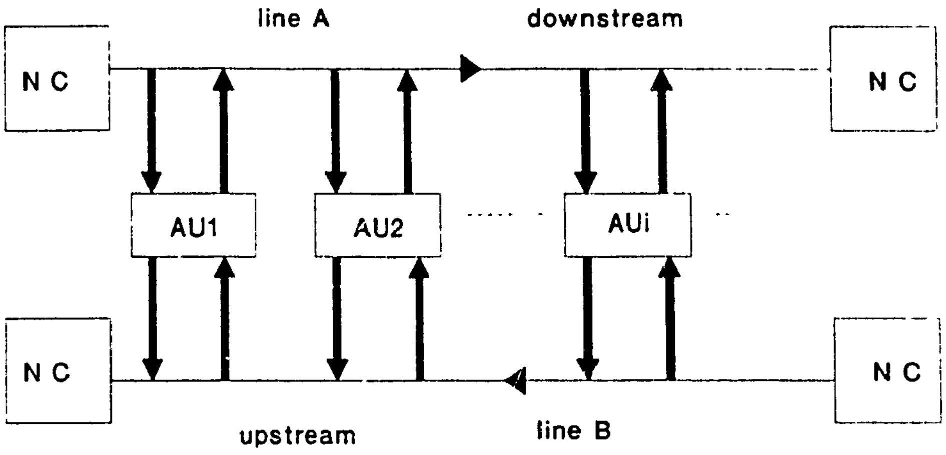


Figure 4.4 Distributed Queueing Topology

When requesting free slots, the sending AU advises upstream AUs only. Downstream AUs need not be aware of this request. Whenever a REQ is received from a downstream AU, all the AUs that read it increment their request counter (figure 4.5) to keep track of all downstream requests for a free slot. This request counter indicates how many data units are queued for transmission downstream. Since the data units waiting for a free slot are spread among the downstream AUs, the process is known as "distributed queueing".

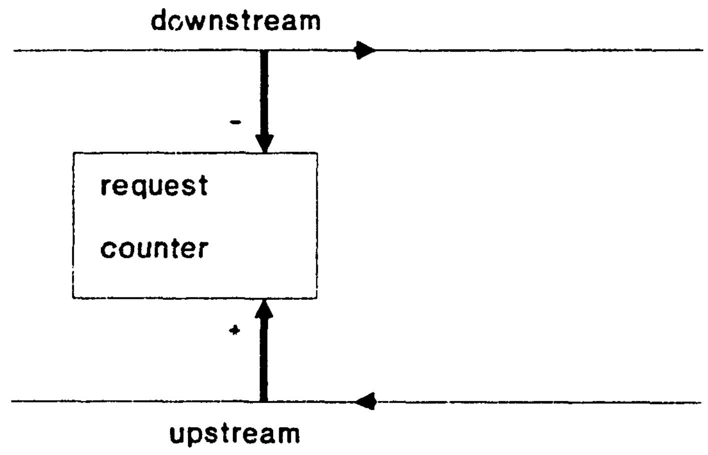


Figure 4.5 Keeping Count of REQ

The sending AU, after transmitting its REQ, copies the content of the request counter to a countdown counter (figure 4.6). Both the countdown counter and the request counter are decremented whenever a free slot goes by downstream. However, only the request counter (not the countdown counter) is incremented whenever a REQ goes by upstream direction [1]. When

[1] There is another implementation which has the same effect. The request counter is transferred to the countdown counter and reset to zero; then only the countdown counter is decremented. This is the method described by [NEWM1]. The first method is adopted because it is less complex. The request counter does not change functions when a data unit is to be sent. Under the proposed approach, one countdown counter is required for each service.

the countdown counter reaches zero, the AU takes the first empty slot going downstream and inserts its data unit. Each AU has a countdown counter for data units of each type of service it provides. This is the basic DQDB protocol [MOLL1], which is straightforward when the slots used are in the wideband or narrowband data subframes. However, when data units can be transmitted on free slots in the voice or video subframes the proposed expanded version of the DQDB protocol must be used.

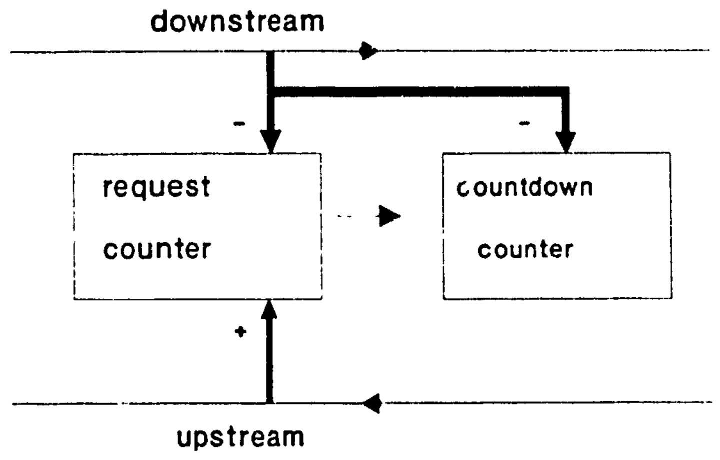


Figure 4.6 Waiting for Turn to Transmit

When an AU's countdown counter for data transmission becomes zero and a slot in a voice or video subframe passes downstream, the AU can use this slot. If the BUSY bit is not set, the AU sets it and codes the CONTROL subfield to NARROWBAND DATA SERVICE or WIDEBAND DATA SERVICE depending on the type of data to be transmitted. The video or voice subframe reallocated to data transmission is then partitioned into multiple data slots (both video and voice slots are multiples of NARROWBAND and WIDEBAND data slots), all available for either NARROWBAND or WIDEBAND data but not both (figure 3.6). The calling AU then takes the first of these new data slots, sets the BUSY bit, codes the CONTROL subfield to NARROWBAND DATA SERVICE or WIDEBAND DATA SERVICE, and transmits a data unit.

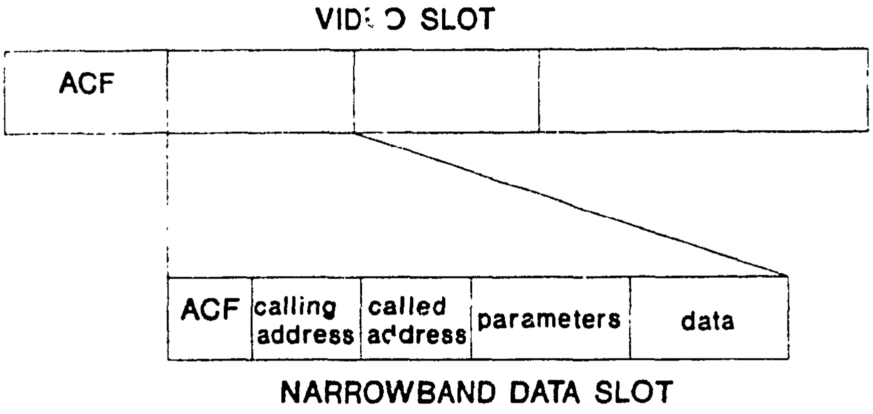


Figure 4.7 Narrowband Data Slots in a Video Slot.

On the other hand, if the BUSY bit of the voice or video slot is set, the AU looks at the slot's CONTROL subfield. If the code is NARROWBAND DATA SERVICE or WIDEBAND DATA SERVICE, the AU can use the slot to transmit data of the same type only. The AU looks for the first empty data slot in the video or voice slot. If the empty data slot is the last available, then the AU changes the CONTROL subfield of the voice or video slot to NARROWBAND DATA FULL or WIDEBAND DATA FULL. This code simply indicates that the voice or video slot has been filled with data of the stated type.

4.4 Priority Schemes

In the proposed system, priority only pertains to data transmitted in empty (BUSY not set) video or voice slots. Priority has no relevance otherwise because every service has its own subframe and there is no contention for capacity in voice or video slots reserved for data of a particular type.

A number of priority schemes are considered. The selected priority scheme should minimize delay in the data services involved. The priority schemes that have been studied are:

Case 1. Restricted priority - Narrowband and wideband data have equal priority in accessing voice and video subframes. However, narrowband data loses access when the wideband data queue reaches a prespecified size Δ_2.

Case 2. All AUs consider wideband data of higher priority than narrowband data. As free voice or video slots pass by, the wideband data countdown counter is decremented but not the narrowband countdown counter. The narrowband data countdown counter starts being decremented by free slots only wher there are no more requests for wideband data.

Case 3. Wideband data has priority over narrowband data until the narrowband data queue reaches a prespecified size Δ_1. Then both wideband and narrowband data have the same priority.

Case 4. As in case 3, narrowband and wideband data have the same priority after narrowband data reaches state Δ_1 but narrowband data loses this priority once wideband data reaches state Δ_2.

4.5 Synchronous Transmission for Voice and Video

Voice and video calls are established in asynchronous mode. The call request is sent by the AU initiating the call in the same fashion as described above for data transmission. Once the call is accepted and acknowledged by the destination AU, the NC reserves a slot for the active voice or video session; there is no detectable difference between AUs functioning this way and AUs functioning entirely in the asynchronous mode; the voice or video traffic, though occupying the same slot in each frame throughout the session, could be regarded by other AUs as asynchronous traffic coming from any upstream AU. The synchronous nature of the communication is only important to the sending and receiving AUs.

An AU wishing to establish a voice or video call first has to make a reservation request either in the voice slot (for voice service) or video slot (for either service). The NC, which generates the frames, must be made aware of and validate the call request. After call establishment, however, NC involvement must be minimal. The destination AU must synchronize itself with the slot position for the call it is to receive. Since the slots are synchronized with respect to the frames, the time between slots is equal to the frame time.

A new protocol is now proposed for voice and video communication over a dual bus network. The starting point is the establishment of a voice or video call. The calling AU initially follows the procedure for DQDB asynchronous transmission. When the calling AU gets hold of a free slot, it sets the BUSY bit and codes the CONTROL subfield for VOICE CALL ESTABLISHMENT or VIDEO CALL ESTABLISHMENT. The same slot also includes the destination and source addresses, call identifier (needed to single out one of the many calls that could exist simultaneously between source and destination) and call parameters. In response to this call request, the destination AU makes a return call on the other bus to the calling AU. It sets the CONTROL subfield of the slot to VOICE CALL ESTABLISHMENT or VIDEO CALL ESTABLISHMENT, puts the calling AU address as the destination address and its own address as the source address, and includes the same call identifier and call parameters. This tells the calling AU that the call is a positive acknowledgement.

These two calls are independent of one another; it is only the AUs involved that need to identify the first as a call request and the second as positive acknowledgement establishing full duplex communication. The destination AU may, of course, refuse the call. The return call will then have the CONTROL subfield coded for VOICE CALL TERMINATION or VIDEO CALL TERMINATION, the proper source and destination addresses and the same call identifier as that found in the call request.

The NC is also involved in the isochronous call establishment. Upon detecting a VOICE CALL ESTABLISHMENT in the CONTROL subfield of a video or voice slot, or VIDEO CALL ESTABLISHMENT in the CONTROL subfield of a video slot, the NC reserves a slot position in the frames it generates by setting the BUSY bit and coding the CONTROL subfield to VOICE SERVICE or VIDEO SERVICE. Call request and acknowledgments, it must be emphasized, are handled independently by the NC. Furthermore, the slots reserved on one bus through the call request and on the other bus through the call acknowledgment need not be in the same position. To terminate a call, each AU codes the CONTROL subfield to VOICE CALL TERMINATION or VIDEO CALL TERMINATION. Upon detecting this code, the NC stops reserving the slots. It can be seen that NC involvement is minimal.

A call request may not always receive a response. The calling AU must therefore timeout the response. After the timeout expires, the calling AU must clear the call request by setting the CONTROL subfield in the reserved slot to VOICE CALL TERMINATION or VIDEO CALL TERMINATION.

Voice service can use video slots. When the distributed queueing protocol allows an AU to take a slot for sending a voice call request, the AU can grab a video slot passing upstream. The AU first checks the BUSY bit of the video slot. If the BUSY bit is not set, the AU sets it and codes the CONTROL subfield to VOICE SERVICE. The video reallocated to voice transmission is then partitioned into multiple voice slots (video slots are some multiple of the voice slot). The calling AU then takes the first of these new voice slots, sets the BUSY bit and codes the CONTROL subfield to VOICE SERVICE.

On the other hand, if the BUSY bit of the video slot is set, the AU looks at the slot's CONTROL subfield. If the code is VOICE SERVICE, the AU can use the slot to transmit voice. The AU looks for the first empty voice slot in the video slot. If the empty voice slot is the last available, then the AU changes the CONTROL subfield of the video slot to VOICE FULL. This code simply indicates that the video slot has been used by voice and is now full.

When an AU wants to send a REQ for a particular service, it must check the appropriate REQ field in successive slots until it detects a free REQ for that service. This creates delay. In this regard, downstream AUs have a slight advantage when requests are simultaneous since they can reserve slots before upstream AUs.

Since there is a REQ for every slot, the AU has to have a countdown counter (CD) for every data unit queued. As slots available for a requested service pass by the AU, all the CDs for that service are decremented.

4.6 A Model of the Proposed Protocol

In order to analyze the proposed protocol, a model is configured as a network of queues. The model assumes that no errors affecting the protocol occur (a model that takes protocol errors into consideration is studied in section 4.8). We will call this the "model with no faults".

Frames available for video, voice, wideband data and narrowband data services are assumed to arrive according to a Poisson process with mean arrival rates of λ_w, λ_v, λ_{d2}, and λ_{d1} respectively. Frame service time is assumed to be exponentially distributed with a mean of $1/\mu_w$, $1/\mu_v$, $1/\mu_{d2}$ and

$1/\mu_{d1}$ respectively. The number of channels for each service, the capacity for each service in terms of narrowband data slot size, and the priority thresholds Δ_1 and Δ_2 are other parameters fed into the model.

Since video traffic is not affected by any other service, i.e., is independent of all other traffic, its steady-state probability distribution is given by the common Erlang distribution formula.

$$P'_m = \frac{\rho_W^m/m!}{\sum_{j=0}^{N_W} \rho_W^j/j!} \qquad (4.1\ a)$$

where m= 0, 1, ... , N_W

N_W is the maximum number of video slots in the video subframe, $\rho_W = \lambda_W / \mu_W$.

Video calls are not buffered; if service is not available calls are blocked cleared. The video blocking probability Pb_W is given by the Erlang B formula because of the independence asumption with respect to other services [KLEI1].

$$Pb_W = \frac{\frac{\rho_W^{N_W}}{N_W!}}{\sum_{j=0}^{N_W} \rho_W^j/j!} \qquad (4.1\ b)$$

Voice traffic is not affected neither by narrowband nor by wideband data traffic. It is also assumed that video and voice traffic do not affect each other eventhough voice slots in video subframes are not preempted by video call requests. This is an acceptable simplification considering that the average video call establishment time is longer than the average voice call duration.

The number of voice slots available at any time is

$$N_V + (N_W - m) \theta$$

where N_V is the maximum number of voice slots in the voice subframe, $\theta = z/y$, the ratio of video slot size to voice slot size, and m is the number of video slots in use.

Averaging over all video states we get the following voice service probability distribution

$$q_\ell = \sum_{m=0}^{N_W} \frac{\rho_V^\ell/\ell!}{\sum_{j=0}^{N_V+N_W\theta-m\theta} \rho_V^j/j!} \cdot \frac{\rho_W^m/m!}{\sum_{j=0}^{N_W} \rho_W^j/j!} \tag{4.2}$$

with $\ell = 0,1,\ldots, (N_W \theta + N_V)$, and $\rho_V = \lambda_V / \mu_V$.

As with video, voice calls are not buffered. They are served by ℓ slots available for voice, which are considered as servers. When all $(N_V + (N_W - m)\theta)$ possible servers for all m video calls are busy, then any incoming voice call will be blocked clear.

The voice blocking probability is thus

$$Pb_V=\sum_{m=0}^{N_W} \frac{\rho_V^{N_V+N_W\theta-m\theta}/(N_V+N_W\theta-m\theta)\,!}{\sum_{j=0}^{N_V+N_W\theta-m\theta} \rho_V^j/j\,!} \cdot \frac{\rho_W^m/m!}{\sum_{j=0}^{N_W} \rho_W^j/j\,!} \qquad (4.3)$$

The average video call establishment time is longer than the average voice call duration. The average videc call establishment time can therefore, in the worst case, be equated to the holding time for voice. The video call establishment time is the time, after having sent a REQ, that a given AU needs to prepare a video call before using a free slot that may become available.

The average video call establishment time is therefore

$$T_W=\frac{1}{2}\sum_{\ell=N_V}^{N_W\theta+N_V} q_\ell' \cdot (\ell-N_V)/\mu_V \qquad (4.4)$$

This is a function of the probability distribution of voice traffic using the video subframe.

The narrowband and wideband data services can compete for empty voice or video slots. Data, whether narrowband or wideband, can be queued when no slot is available; however, there is a limited buffer capacity for each. When the buffer is full, data call requests are blocked cleared. The narrowband and wideband data are assumed to be bursty i.e. hold time is assumed to be very short compared to that of video and voice traffic. Video and voice calls thus do not have to preempt data calls.

Each video slot can accommodate z narrowband data slots z/x wideband data slots or z/y voice slots. Each voice slot can accommodate y narrowband data slots or y/x wideband data slots. The probable number of available slots from the video and voice subframe is therefore

$$k = N_w z + N_v y - m z - \ell y$$

where

k is expressed in terms of equivalent narrowband data slots,

N_w is the number of video slots in the video subframe,

N_v is the number of voice slots in the voice subframe,

m is the number of video slots used for video service,

m z is the number of equivalent narrowband slots used for video service (m video slots used, each equivalent to z narrowband slots) and

ℓ y is the number of equivalent narrowband slots used for voice service (ℓ voice slots used, each equivalent to y narrowband slots).

k can also be represented in the following form

$$k = \xi - \gamma y$$

where $\xi = N_w z + N_v y$, and

$\gamma y = m z + \ell y$.

The probability distribution of available video and voice slots where voice can occupy video slots, is thus the probability of the sum of occupied voice and video slots

($S=\ell+m\theta$). This probability is

$$P(S)=\sum_{m=0}^{N_W} q'_{((S-m\theta)/m\theta)}P'_m \qquad (4.5)$$

for $S = 0, \ldots , (N_V + N_W\,\theta)$

The probability distribution of available video and voice slots for data is transformed as follows

$$P_k(k=\xi-\gamma y)=P(S=N_V+N_W\theta-k/y) \qquad (4.6)$$

for $\gamma = 0,1, \ldots ,\xi/y$.

The Narrowband data arrival rate is λ_{d1}. The total service rate for the narrowband subframe is $N_{d1}\mu_{d1}$, where N_{d1} is the number of narrowband data slots and μ_{d1} is the unit service rate.

The wideband data arrival rate expected of equivalent narrowband slots, is λ_{d2}. The total service rate for the wideband subframe is $N_{d2}X\mu_{d2}$, where N_{d2} is the number of wideband data slots and $X\mu_{c2}$ is the unit service rate in equivalent narrowband slots.

When narrowband data uses empty voice or video slots, its total service rate becomes $N_{d1}\mu_{d1} + k\mu_{d1}$, where k, as shown above, is the probable number of other available video and voice slots. When wideband data uses available video or voice slots, its total service rate becomes $N_{d2}X\mu_{d2} + kX\mu_{d2}$.

Wideband data can have priority over narrowband data, although this priority can be restrictive (Case 1 priority scheme in section 4.4). In this analysis, priority is represented as two subsets as follows. Wideband data has, starting at state 0 of the queueing network as shown in figure 4.8, a service rate $N_{d2}X\mu_{d2}$ + $kX\mu_{d2}$. Narrowband data has starting at state 0 a service rate of $N_{d1}\mu_{d1}$ + $k\mu_{d1}$ until wideband data reaches the state Δ_2, at this point on the narrowband data service rate becomes $N_{d1}\mu_{d1}$.

The probability distribution for narrowband and wideband data can be represented by a joint probability $P[(n_{d1},n_{d2})/k]$ where n_{d1} represents the narrowband state and n_{d2} the wideband state. The state diagram of figure 4.8, the dimensions of the matrix depend on the size of the narrowband data and wideband data buffers. If the maximum narrowband data buffer is M_1 and the wideband data buffer M_2, then the state matrix represents an $(M_1+1)(M_2+1)$ network of queues and there are as many states in the system.

For analysis purposes, the states for specific values of k are represented in vector form. Thus :

$S_0=P[(0,0)/k]$, $S_1=P[(0,1)/k]$, ... , $S_n=P[(0,n)/k]$ for $n<M_2$
and $S_{M2}=P[(0,M_2)/k]$, $S_{M2+1}=P[(1,1)/k]$, ...,
till $S_{M1\ M2+M1+M2}=P[(M_1,M_2)/k]$.

From any state $P[(i,j)/k]$ which is an element of the state matrix, we can derive the equivalent term S_n belonging to the corresponding state vector, using the following algorithm:

Transformation Algorithm -

given $0 \le i \le M_1$ and $0 \le j \le M_2$

then $S_n = P[(i,j)/k]$ for $n = i\,M_2 + i + j$

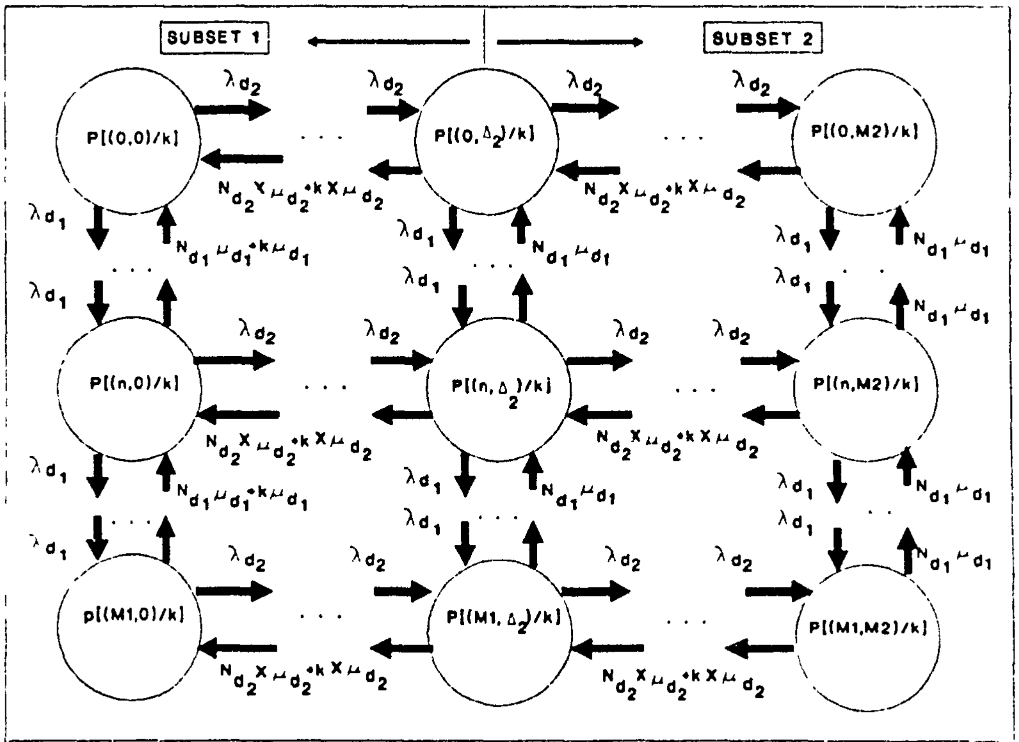


Figure 4.8 State Diagram under Case 1 of Priority Scheme

The state equations can then be derived.

For the restricted priority scheme (case 1) the state equations are as follows:

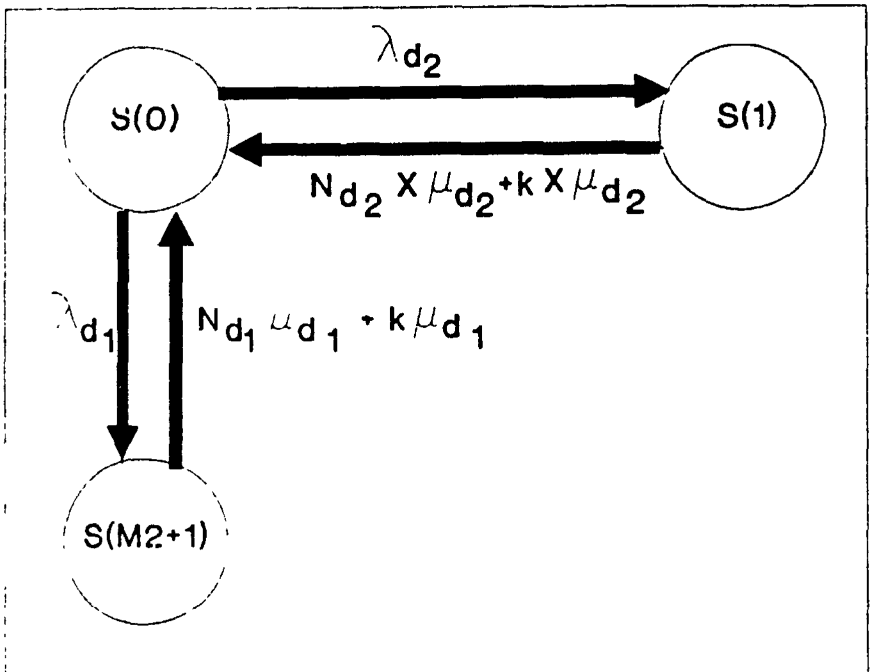

λd_2
S(0)
S(1)
$N_{d_2} X \mu_{d_2} + k X \mu_{d_2}$
λd_1
$N_{d_1} \mu_{d_1} + k \mu_{d_1}$
S(M2+1)

Figure 4.9. Initial State Transition at P[(0,0)/k] under Case 1 Priority Scheme.

At the null state P[(0,0)/k] where $\Delta_2 > 0$ and according to figure 4.9

$$(\lambda_{d_2}+\lambda_{d_1}) S_0 = (N_{d_2}X\mu_{d_2}+kX\mu_{d_2}) S_1 + (N_{d_1}\mu_{d_1}+k\mu_{d_1}) S_{(M_2+1)} \qquad (4.7)$$

At the top state boundary P[(0,ℓ)/k] where $0 < \ell < M_2$

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1}) S_n = \lambda_{d_2}S_{(\ell-1)} + (N_{d_2}X\mu_{d_2}+kX\mu_{d_2}) S_{(\ell+1)} + (Nd_1\mu_{d_1}+\Theta) S_{(M_2+\ell+1)} \qquad (4.8)$$

where $\Theta = k\mu_{d1}$ if $\Delta_2 < \ell < M_2$, and $\Theta = 0$ otherwise.

In general P[(n,ℓ)/k] where $0< n < M_1$ and $0 < \ell < M_2$

$$\begin{aligned}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1}+N_{d_1}\mu_{d_1}+\Theta)\,S_{(nM_2+n+\ell)} &= \\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+\ell)}+\lambda_{d_2}S_{(nM_2+n+\ell-1)} & \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+\ell+1)} & \\ +(N_{d_1}\mu_{d_1}+\Theta)\,S_{((n+1)M_2+n+1+\ell)} &\end{aligned} \tag{4.9}$$

where $\theta = k\mu_{d1}$ if $1 \le \ell < \Delta_2$, and $\theta = 0$ if $\Delta_2 \le \ell < M_2$.

At the state P[(0,M_2)/k] (S_{M2})

$$(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_{M_2}=\lambda_{d_2}S_{(M_2-1)}+N_{d_1}\mu_{d_1}S_{(2M_2+1)} \tag{4.10}$$

At the state P[(M_1,0)/k] ($S_{M1M2+M1}$)

$$\begin{aligned}(\lambda_{d_2}+n_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_1M_2+M_1)} &=\lambda_{d_1}S_{((M_1-1)M_2+(M_1-1))} \\ &+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+1)}\end{aligned} \tag{4.11}$$

At the last state P[(M1,M2)/k] ($S_{M1M2+M1+M2}$)

$$\begin{aligned}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1})\,S_{(M_1M_2+M_1+M_2)} &= \\ \lambda_{d_1}S_{((M_1-1)M_2+(M_1-1)+M_2)} & \\ +\lambda_{d_2}S_{(M_1M_2+M_1+(M_2-1))} &\end{aligned} \tag{4.12}$$

At the lower boundary P[(M_1,ℓ)/k] where $0 < \ell < M_2$

$$\begin{aligned}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\Theta)\,S_{(M_1M_2+M_1+\ell)} &=\lambda_{d_2}S_{(M_1M_2+M_1+\ell-1)} \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+\ell+1)}+(N_{d_1}\mu_{d_1}+\Theta)\,S_{((M_1-1)M_2+(M_1-1)+\ell)} &\end{aligned} \tag{4.13}$$

where $\theta = k\mu_{d1}$ if $1 \le \ell < \Delta_2$, and $\theta = 0$ if $\Delta_2 \le \ell < M_2$.

At the left boundary P[(n,0)/k] where $0 < n < M_1$

$$\begin{aligned}(\lambda_{d_2}+N_{d_1}\mu_{d_1}{:}k\mu_{d_1}+\lambda_{d_1})\,S_{(nM_2+n)} &=\lambda_{d_1}S_{((n-1)M_2+n-1)} \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+1)}+(N_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{((n+1)M_2+n+1)} &\end{aligned} \tag{4.14}$$

At the right boundary P[(n,M_2)/k] where 0 < n < M_1

$$\begin{gathered}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\lambda_{d_1})\,S_{(nM_2+n+M_2)}=\\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+M_2)}+\lambda_{d_2}S_{(nM_2+n+M_2-1)}+N_{d_1}\mu_{d_1}S_{((n+1)M_2+(n+1)+M_2)}\end{gathered} \qquad (4.15)$$

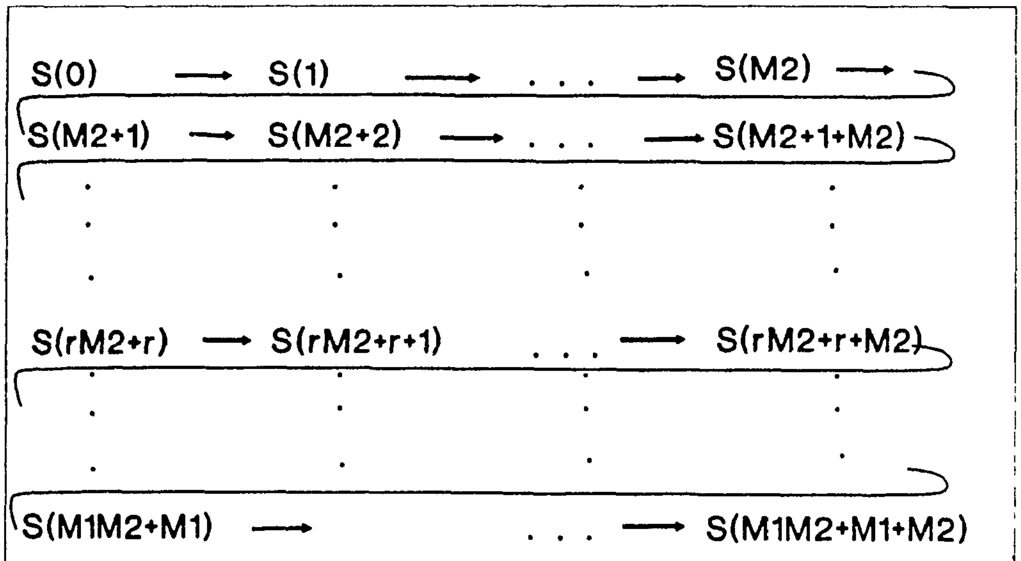


Figure 4.10 State Vector [S(0) to S(M1M2+M1+M2)]

The state equations for the case 2 priority scheme according to figure 4.11 are as follows:

At the null state P[(0,0)/k]

$$(\lambda_{d_2}+\lambda_{d_1})\,S_0=(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_1+N_{d_1}\mu_{d_1}S_{(M_2+1)} \qquad (4.16)$$

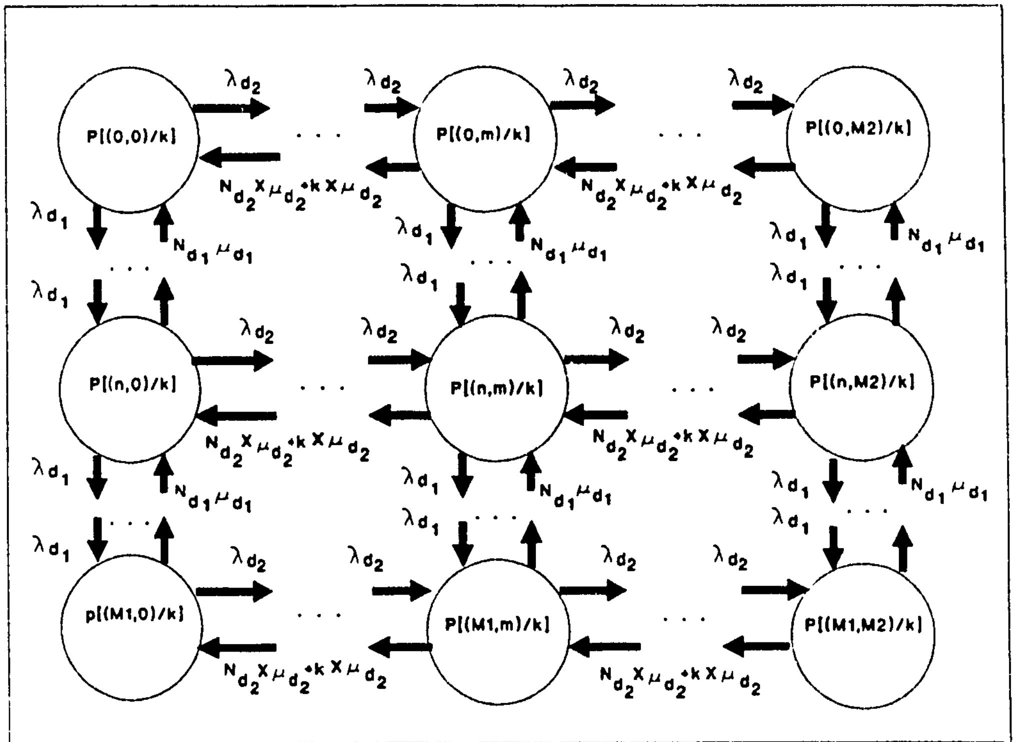


Figure 4.11 State Diagram under Case 2 of Priority Scheme

At the top state boundary P[(0,ℓ)/k] where 0 < ℓ < M_2

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_n=\lambda_{d_2}S_{(\ell-1)}+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(\ell+1)} \\ +Nd_1\mu_{d_1}S_{(M_2+\ell+1)} \tag{4.17}$$

In general P[(n,ℓ)/k] where 0< n < M_1 and 0 < ℓ < M_2

$$\begin{gathered}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1}+N_{d_1}\mu_{d_1})\,S_{(nM_2+n+\ell)}= \\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+\ell)}+\lambda_{d_2}S_{(nM_2+n+\ell-1)} \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+\ell+1)} \\ +N_{d_1}\mu_{d_1}S_{((n+1)M_2+n+1+\ell)}\end{gathered} \tag{4.18}$$

At the state P[(0,M_2)/k] (S_{M2})

$$(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_{M_2}=\lambda_{d_2}S_{(M_2-1)}+N_{d_1}\mu_{d_1}S_{(2M_2+1)} \tag{4.19}$$

At the state P[(M$_1$,0)/k] (S$_{M1M2+M1}$)

$$(\lambda_{d_2}+n_{d_1}\mu_{d_1})\,S_{(M_1M_2+M_1)}=\lambda_{d_1}S_{((M_1-1)M_2+(M_1-1))}+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+1)} \tag{4.20}$$

At the last state P[(M1,M2)/k] (S$_{M1M2+M1+M2}$)

$$\begin{gathered}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1})\,S_{(M_1M_2+M_1+M_2)}=\\ \lambda_{d_1}S_{((M_1-1)M_2+(M_1-1)+M_2)}\\ +\lambda_{d_2}S_{(M_1M_2+M_1+(M_2-1))}\end{gathered} \tag{4.21}$$

At the lower boundary P[(M$_1$,ℓ)/k] where 0 < ℓ < M$_2$

$$\begin{gathered}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1})\,S_{(M_1M_2+M_1+\ell)}=\lambda_{d_2}S_{(M_1M_2+M_1+\ell-1)}\\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+\ell+1)}+N_{d_1}\mu_{d_1}S_{((M_1-1)M_2+(M_1-1)+\ell)}\end{gathered} \tag{4.22}$$

At the left boundary P[(n,0)/k] where 0 < n < M$_1$

$$\begin{gathered}(\lambda_{d_2}+N_{d_1}\mu_{d_1}+\lambda_{d_1})\,S_{(nM_2+n)}=\lambda_{d_1}S_{((n-1)M_2+n-1)}\\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+1)}+N_{d_1}\mu_{d_1}S_{((n+1)M_2+n+1)}\end{gathered} \tag{4.23}$$

At the right boundary P[(n,M$_2$)/k] where 0 < n < M$_1$

$$\begin{gathered}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\lambda_{d_1})\,S_{(nM_2+n+M_2)}=\\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+M_2)}+\lambda_{d_2}S_{(nM_2+n+M_2-1)}+N_{d_1}\mu_{d_1}S_{((n+1)M_2+(n+1)+M_2)}\end{gathered} \tag{4.24}$$

The state equations for the case 3 priority scheme according to figure 4.12 are as follows:

At the null state P[(0,0)/k] where Δ_1 > 0

$$(\lambda_{d_2}+\lambda_{d_1})\,S_0=(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_1+N_{d_1}\mu_{d_1}S_{(M_2+1)} \tag{4.25}$$

At the top state boundary P[(0,ℓ)/k] where 0 < ℓ < M_2

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_n=\lambda_{d_2}S_{(\ell-1)}+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(\ell+1)} \\ +Nd_1\mu_{d_1}S_{(M_2+\ell+1)}\tag{4.26}$$

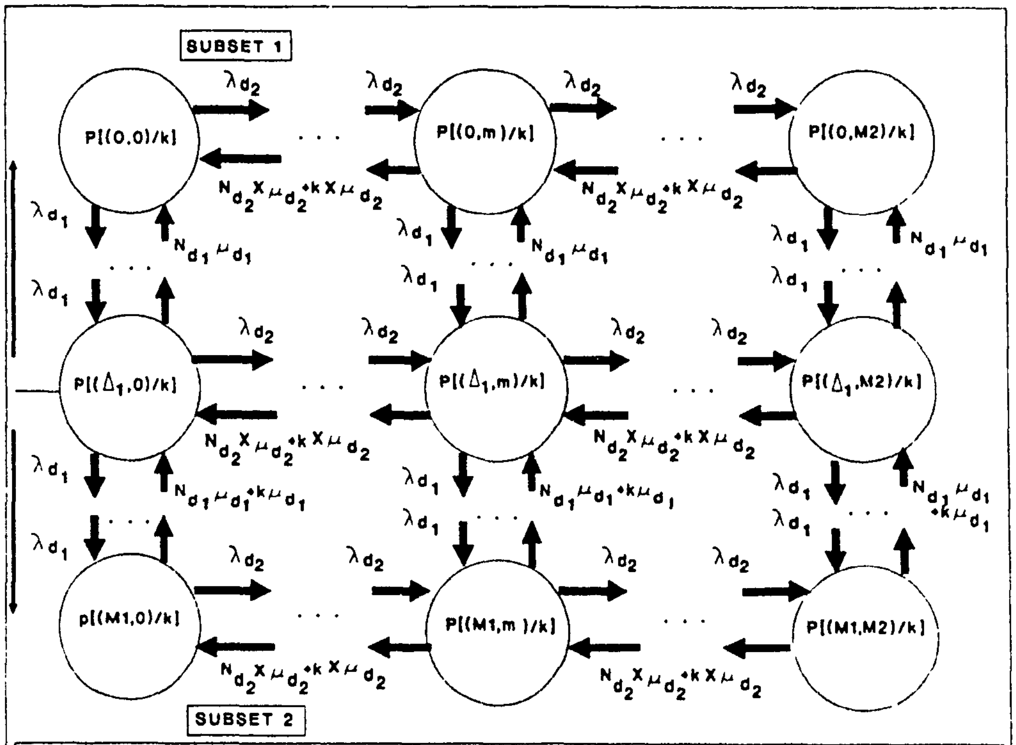


Figure 4.12 State Diagram under Case 3 of Priority Scheme

In general P[(n,ℓ)/k] where 0< n < M_1 and 0 < ℓ < M_2

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1}+N_{d_1}\mu_{d_1}+\Theta_D)\,S_{(nM_2+n+\ell)}= \\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+\ell)}+\lambda_{d_2}S_{(nM_2+n+\ell-1)} \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+\ell+1)} \\ +(N_{d_1}\mu_{d_1}+\Theta_N)\,S_{((n+1)M_2+n+1+\ell)}\tag{4.27}$$

where $\Theta_D = k\mu_{d1}$ if $\Delta_1 < n < M_1$, and $\Theta_D = 0$ if $1 \le n \le \Delta_1$

$\Theta_N = k\mu_{d1}$ if $\Delta_1 \le n < M_1$, and $\Theta_D = 0$ if $1 \le n < \Delta_1$

At the state P[(0,M_2)/k] (S_{M2})

$$(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_{M_2}=\lambda_{d_2}S_{M_2-1}+N_{d_1}\mu_{d_1}S_{2M_2+1} \qquad (4.28)$$

At the state P[(M_1,0)/k] ($S_{M1M2+M1}$)

$$\begin{aligned}(\lambda_{d_2}+n_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_1M_2+M_1)}&=\lambda_{d_1}S_{((M_1-1)M_2+(M_1-1))}\\&+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+1)}\end{aligned} \qquad (4.29)$$

At the last state P[(M1,M2)/k] ($S_{M1M2+M1+M2}$)

$$\begin{aligned}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_1M_2+M_1+M_2)}&=\\\lambda_{d_1}S_{((M_1-1)M_2+(M_1-1)+M_2)}&\\+\lambda_{d_2}S_{(M_1M_2+M_1+(M_2-1))}&\end{aligned} \qquad (4.30)$$

At the lower boundary P[(M_1,ℓ)/k] where $0 < \ell < M_2$

$$\begin{aligned}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_1M_2+M_1+\ell)}&=\lambda_{d_2}S_{(M_1M_2+M_1+\ell-1)}\\+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+\ell+1)}&+(N_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{((M_1-1)M_2+(M_1-1)+\ell)}\end{aligned} \qquad (4.31)$$

At the left boundary P[(n,0)/k] where $0 < n < M_1$

$$\begin{aligned}(\lambda_{d_2}+N_{d_1}\mu_{d_1}+\theta_D+\lambda_{d_1})\,S_{(nM_2+n)}&=\lambda_{d_1}S_{((n-1)M_2+n-1)}\\+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+1)}&+(N_{d_1}\mu_{d_1}+\theta_N)\,S_{((n+1)M_2+n+1)}\end{aligned} \qquad (4.32)$$

where $\Theta_D = k\mu_{d1}$ if $\Delta_1 < n < M_1$, and $\Theta_D = 0$ if $1 \le n \le \Delta_1$

$\Theta_N = k\mu_{d1}$ if $\Delta_1 \le n < M_1$, and $\Theta_D = 0$ if $1 \le n < \Delta_1$

At the right boundary P[(n,M_2)/k] where $0 < n < M_1$

$$\begin{aligned}&(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\theta_D+\lambda_{d_1})\,S_{(nM_2+n+M_2)}=\\&\lambda_{d_1}S_{((n-1)M_2+(n-1)+M_2)}+\lambda_{d_2}S_{(nM_2+n+M_2-1)}+(N_{d_1}\mu_{d_1}+\theta_N)\,S_{((n+1)M_2+(n+1)+M_2)}\end{aligned}$$

(4.33)

where $\Theta_D = k\mu_{d1}$ if $\Delta_1 < n < M_1$, and $\Theta_D = 0$ if $1 \le n \le \Delta_1$

$\Theta_N = k\mu_{d1}$ if $\Delta_1 \le n < M_1$, and $\Theta_D = 0$ if $1 \le n < \Delta_1$

The state equations for the case 4 priority scheme according to figure 4.13 are as follows:

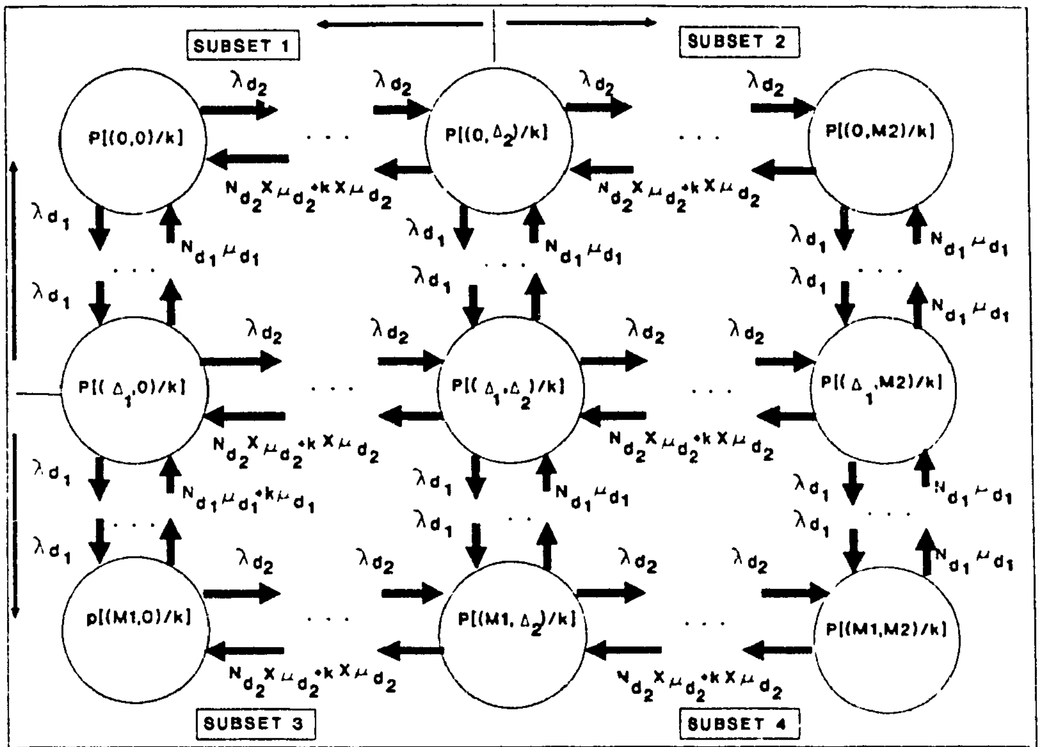


Figure 4.13 State Diagram under Case 4 of Priority Scheme

At the null state P[(0,0)/k] where $\Delta_2 > 0$ and $\Delta_1 > 0$

$$(\lambda_{d_2}+\lambda_{d_1})\,S_0=(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_1+(N_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_2+1)} \qquad \textbf{(4.34)}$$

At the top state boundary P[(0,ℓ)/k] where $0 < \ell < M_2$

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_n=\lambda_{d_2}S_{(\ell-1)}+(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(\ell+1)} \\ +Nd_1\mu_{d_1}S_{(M_2+\ell+1)} \qquad \textbf{(4.35)}$$

In general P[(n,ℓ)/k] where $0< n < M_1$ and $0 < \ell < M_2$

$$(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1}+N_{d_1}\mu_{d_1}+\Theta_D)\,S_{(nM_2+n+\ell)}= \\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+\ell)}+\lambda_{d_2}S_{(nM_2+n+\ell-1)} \\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+\ell+1)} \\ +(N_{d_1}\mu_{d_1}+\Theta_N)\,S_{((n+1)M_2+n+1+\ell)} \qquad \textbf{(4.36)}$$

where $\Theta_D = k\mu_{d1}$ if $1 \le \ell < \Delta_2$, and $\Delta_1 < n < M_1$,

$\Theta_D = 0$ if $\Delta_2 \le \ell < M_2$ and $1 \le n \le \Delta_1$

$\Theta_N = k\mu_{d1}$ if $1 \le \ell < \Delta_2$, and $\Delta_1 \le n < M_1$,

$\Theta_N = 0$ if $\Delta_2 \le \ell < M_2$ and $1 \le n < \Delta_1$

At the state $P[(0,M_2)/k]$ (S_{M2})

$$(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+\lambda_{d_1})\,S_{M_2}=\lambda_{d_2}S_{(M_2-1)}+N_{d_1}\mu_{d_1}S_{(2M_2+1)} \tag{4.37}$$

At the state $P[(M_1,0)/k]$ $(S_{M1M2+M1})$

$$\begin{gathered}(\lambda_{d_2}+n_{d_1}\mu_{d_1}+k\mu_{d_1})\,S_{(M_1M_2+M_1)}=\lambda_{d_1}S_{((M_1-1)M_2+(M_1-1))}\\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+1)}\end{gathered} \tag{4.38}$$

At the last state $P[(M1,M2)/k]$ $(S_{M1M2+M1+M2})$

$$\begin{gathered}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1})\,S_{(M_1M_2+M_1+M_2)}=\\ \lambda_{d_1}S_{((M_1-1)M_2+(M_1-1)+M_2)}\\ +\lambda_{d_2}S_{(M_1M_2+M_1+(M_2-1))}\end{gathered} \tag{4.39}$$

At the lower boundary $P[(M_1,\ell)/k]$ where $0 < \ell < M_2$

$$\begin{gathered}(\lambda_{d_2}+N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\Theta)\,S_{(M_1M_2+M_1+\ell)}=\lambda_{d_2}S_{(M_1M_2+M_1+\ell-1}\\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(M_1M_2+M_1+\ell+1)}+(N_{d_1}\mu_{d_1}+\Theta)\,S_{((M_1-1)M_2+(M_1-1)+\ell)}\end{gathered}$$

(4.40)

where $\Theta = k\mu_{d1}$ if $1 \le \ell < \Delta_2$, and $\Theta = 0$ if $\Delta_2 \le \ell < M_2$.

note that $0 < \Delta_1 < M_1$.

At the left boundary $P[(n,0)/k]$ where $0 < n < M_1$

$$\begin{gathered}(\lambda_{d_2}+N_{d_1}\mu_{d_1}+\theta_D+\lambda_{d_1})\,S_{(nM_2+n)}=\lambda_{d_1}S_{((n-1)M_2+n-1)}\\ +(N_{d_2}X\mu_{d_2}+kX\mu_{d_2})\,S_{(nM_2+n+1)}+(N_{d_1}\mu_{d_1}+\theta_N)\,S_{((n+1)M_2+n+1)}\end{gathered} \tag{4.41}$$

where $\Theta_D = k\mu_{d1}$ if $\Delta_1 < n < M_1$, and $\Theta_D = 0$ if $1 \leq n \leq \Delta_1$

$\Theta_N = k\mu_{d1}$ if $\Delta_1 \leq n < M_1$, and $\Theta_N = 0$ if $1 \leq n < \Delta_1$

At the right boundary P[(n,M_2)/k] where 0 < n < M_1

$$\begin{gathered}(N_{d_2}X\mu_{d_2}+kX\mu_{d_2}+N_{d_1}\mu_{d_1}+\lambda_{d_1})\,S_{(nM_2+n+M_2)}=\\ \lambda_{d_1}S_{((n-1)M_2+(n-1)+M_2)}+\lambda_{d_2}S_{(nM_2+n+M_2-1)}+N_{d_1}\mu_{d_1}S_{((n+1)M_2+(n+1)+M_2)}\end{gathered} \qquad (4.42)$$

The state equations can be derived either from figure 4.8, 4.9 and 4.10 for the case 1 priority scheme, from figure 4.11 for the case 2 priority scheme, from figure 4.12 for the case 3 priority scheme, from figure 4.13 for the case 4 priority scheme or with the use of an algorithm described below for any of the priority schemes. The state equations so derived form a set of simultaneous equations that can be solved using the Gauss-Seidel iterative method.

The simultaneous equations are of the form

$$S=QS+V \qquad (4.43)$$

where V is a vector, S is the state vector as shown in figure 4.10 and Q is a matrix of dimension ((M_1+1)(M_2+1)) x ((M_1+1)(M_2+1)) representing the coefficients in the state equations.

The algorithm to derive the matrix Q and thus the state equations is as follows:

State Coefficient Algorithm

```
While $i \le M_1$
begin
    While $j \le M_2$
    begin
        $\theta_D = 0;\ \theta_N = 0;$
        if $\{\ j < \Delta_2\}$ then
            if$\{\ i > \Delta_1\}\ \theta_D = k\mu_{d1};\theta_N = k\mu_{d1};$
            if$\{\ i = \Delta_1\}\ \theta_N = k\mu_{d1};$
        endif;
        $d_1=\lambda_{d1};\ n_1=\lambda_{d1};\ d_2=\lambda_{d2};\ n_2=\lambda_{d2};$
        $d_3=N_{d1}\mu_{d1} + \theta_D;\ n_3=N_{d1}\mu_{d1} + \theta_N;$
        $d_4=N_{d2}X\mu_{d2} + kX\mu_{d2};\ n_4=N_{d2}X\mu_{d2} + kX\mu_{d2};$
        if $\{\ i = 0\}\ n_1=0;\ d_3=0;$
        if $\{\ j = 0\}\ n_2=0;\ d_4=0;$
        if $\{\ i = M_1\}\ n_3=0;\ d_1=0;$
        if $\{\ j = M_2\}\ n_4=0;\ d_2=0;$
        denom= $d_1+d_2+d_3+d_4;$
        up=$n_1$/denom; left=$n_2$/denom;
        down=$n_3$/denom; right=$n_4$/denom;
        $\alpha=iM_2+j+i;$
        $\beta_{up}=(i-1)M_2+i+(j-1);\ \beta_{left}=iM_2+i+(j-1);$
        $\beta_{right}=iM_2+i+(j+1);\qquad \beta_{down}=(i+1)M_2+i+(j+1);$
        if $\{\ i \ne 0\ \}\ Q(\alpha,\beta_{up})=-up;$
        if $\{\ j \ne 0\ \}\ Q(\alpha,\beta_{left})=-left;$
```

```
            if ( i $\neq$ $M_1$ ) $Q(\alpha,\beta_{down})$=-down;
            if ( j $\neq$ $M_2$ ) $Q(\alpha,\beta_{right})$=-right;
            if ( i = j ) $Q(\alpha,\alpha)$=1;
            i=i+1;
        end;
        j=j+1;
    end;
```

Note that this algorithm is valid for all cases of priority schemes. The settings of the threshold Δ_1 and threshold Δ_2 determine which case is used. So for case 1, the restricted priority scheme, threshold Δ_1 is set to zero and threshold Δ_2 is set to the state position where the narrowband data service is to relinquish the extra capacity in favour of the wideband data. For case 2, threshold Δ_2 is set to zero since the narrowband data service does not use the extra capacity. In this case threshold Δ_1 has no effect. For case 3, threshold Δ_1 is set to the state where narrowband data service is to start sharing the extra capacity with wideband data service provided that the threshold Δ_2 is set to M_2. Finally for case 4, threshold Δ_1 and threshold Δ_2 are set each to a state dimension that would limit narrowband data service in benefiting of the extra capacity.

The identity equation must also be considered

$$\sum_{i=0}^{(M_1+1)(M_2+1)} S(i)=1 \qquad (4.44)$$

This equation can replace any of the simultaneous equations.

Equation (4.43) can be rewritten as

$$(I-Q)S=V \qquad (4.45)$$

where V is a vector with all terms equal to zero except for one term which has the value of one. This term is at the position where the identity equation (4.44) is placed within equation (4.43) or (4.45).

The Gauss-Seidel iterative method uses equation (4.45) and gives a solution for S. The constraints to the resolution of this equation are: $0 \leq S \leq 1$ and the identity equation (4.44).

The solution to the vector S is then transformed back into a matrix that represents the two dimensional states of both data services. Delays for both narrowband and wideband data as well as their buffer overflow probabilities can now be calculated.

It is now possible to find the delays for narrowband and wideband data as well as their waiting time, buffer overflow probabilities and marginal distributions.

The delay for narrowband data is

$$D_{N_{d_1}}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_2=0}^{M_2}\sum_{n_1=0}^{M_1}n_1[P(n_1,n_2)/k]\cdot P_k(k=\xi-\gamma y) \qquad (4.46)$$

where the term P_k is as defined in equation 4.6.

The delay for wideband data is

$$D_{N_{d_2}}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_1=0}^{M_1}\sum_{n_2=0}^{M_2}n_2[P(n_1,n_2)/k]\cdot p_k(k=\xi-\gamma y) \qquad (4.47)$$

The waiting time for narrowband data is

$$W_{N_{d_1}}=D_{N_{d_1}}/\lambda_{d_1} \qquad (4.48)$$

The waiting time for wideband data is

$$W_{N_{d_2}}=D_{N_{d_2}}/\lambda_{d_2} \qquad (4.49)$$

Both waiting times are in frame time units. In order to find the waiting time in seconds one has to multiply it with the frame time T_{frame} where

$$T_{frame}=S_L\cdot N\cdot\frac{1}{C} \qquad (4.50)$$

where

S_L = number of slots per frame

N = number of bits per slot

C = line capacity in bits per seconds.

The buffer overflow for narrowband data is

$$P_{B_{N_{d_1}}}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_2=0}^{M_2}[P(M_1,n_2)/k]\cdot P_k(k=\xi-\gamma y) \qquad (4.51)$$

The buffer overflow for wideband data is

$$P_{B_{N_{d_2}}}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_1=0}^{M_1}[P(n_1,M_2)/k]\cdot P_k(k=\xi-\gamma y) \qquad (4.52)$$

The marginal distributions are:

$$P_{(n_1)}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_2=0}^{M_2}[P(n_1,n_2)/k]\cdot P_k(k=\xi-\gamma y) \qquad (4.53)$$

for narrowband

$$P_{(n_2)}=\sum_{\gamma=0}^{[\xi/y]}\sum_{n_1=0}^{M_1}[P(n_1,n_2)/k]\cdot P_k(k=\xi-\gamma y) \qquad (4.54)$$

for wideband

4.7 Computational Results

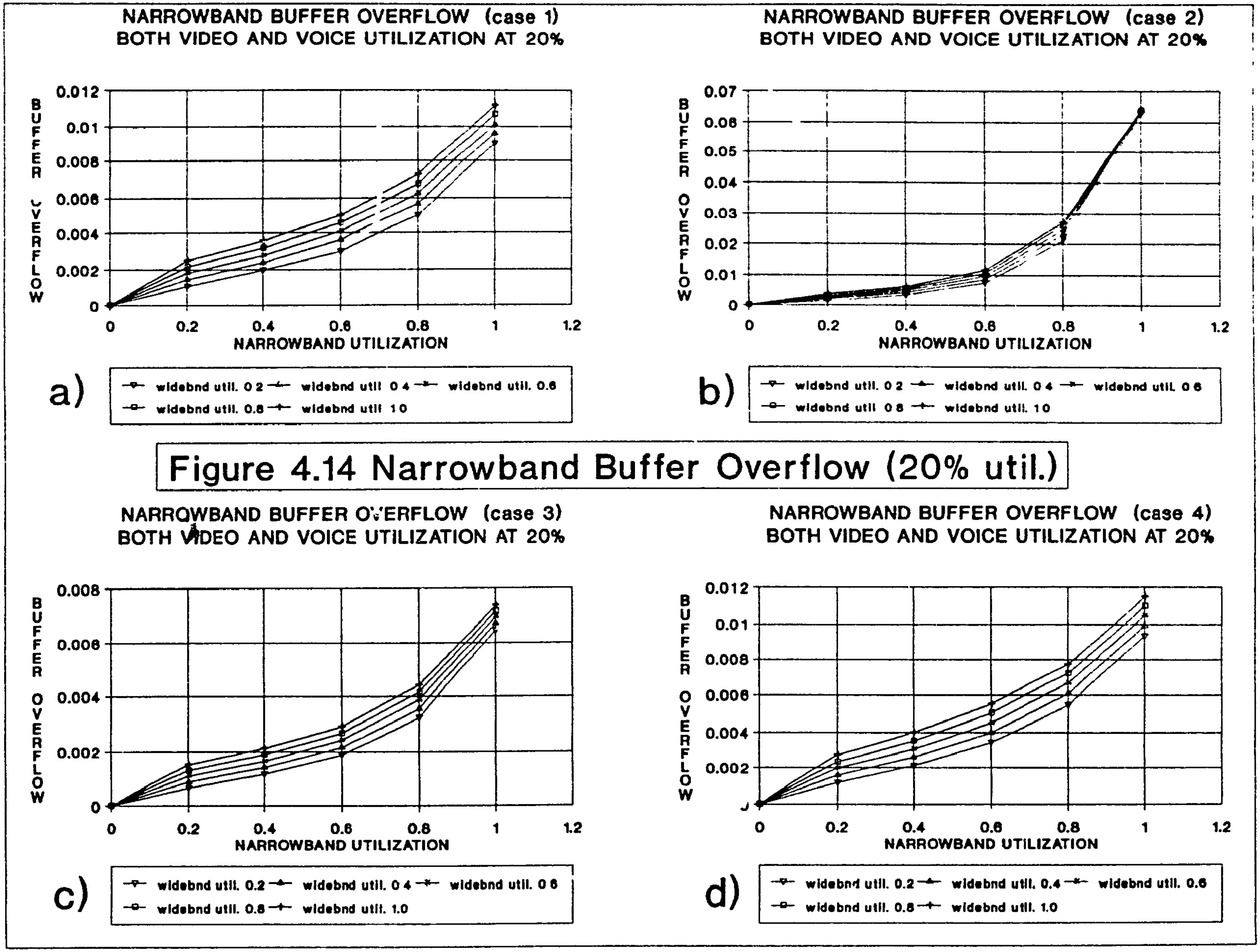
In order to analyze the proposed model, a program has been written to take all the above analysis into consideration. The input to the program are the system parameters for all four services. The program is run for all four priority schemes described above. The input parameters are: mean arrival λ, mean service time μ, number of channels each service is provided with and the capacity equivalent for each service with respect to narrowband data packets. The states Δ_1 and Δ_2 are also entered for each case. The program computes the matrix $P[(N_1,N_2)/k]$ for each value of k by first solving for the state vector S for each case requested by the input, and uses the results to compute the buffer overflow probabilities and queueing delays for both narrowband data and wideband data. The blocking probabilities for voice and video are also computed and the video call establishment estimated.

The model analyzed in the following discussion uses the values in table 4.2.

TABLE 4.2

	<u>video</u>	<u>voice</u>	<u>wideband data</u>	<u>narrowband data</u>
Slots/frame	N_w= 2	N_v= 4	N_{d1}= 3	N_{d2} = 5
Capacity in equiv narrbnd slots	z= 4	y= 2	x= 2	1

The model is tested for different values of voice and video utilization (λ/μ). In each case, and for each priority scheme the narrowband and wideband buffer overflow and queueing delay are computed for various combinations of narrowband and wideband utilizations. The results are plotted in figures 4.14 to 4.17.



NARROWBAND BUFFER OVERFLOW (case 1)
BOTH VIDEO AND VOICE UTILIZATION AT 20%

NARROWBAND BUFFER OVERFLOW (case 2)
BOTH VIDEO AND VOICE UTILIZATION AT 20%

Figure 4.14 Narrowband Buffer Overflow (20% util.)

NARROWBAND BUFFER OVERFLOW (case 3)
BOTH VIDEO AND VOICE UTILIZATION AT 20%

NARROWBAND BUFFER OVERFLOW (case 4)
BOTH VIDEO AND VOICE UTILIZATION AT 20%

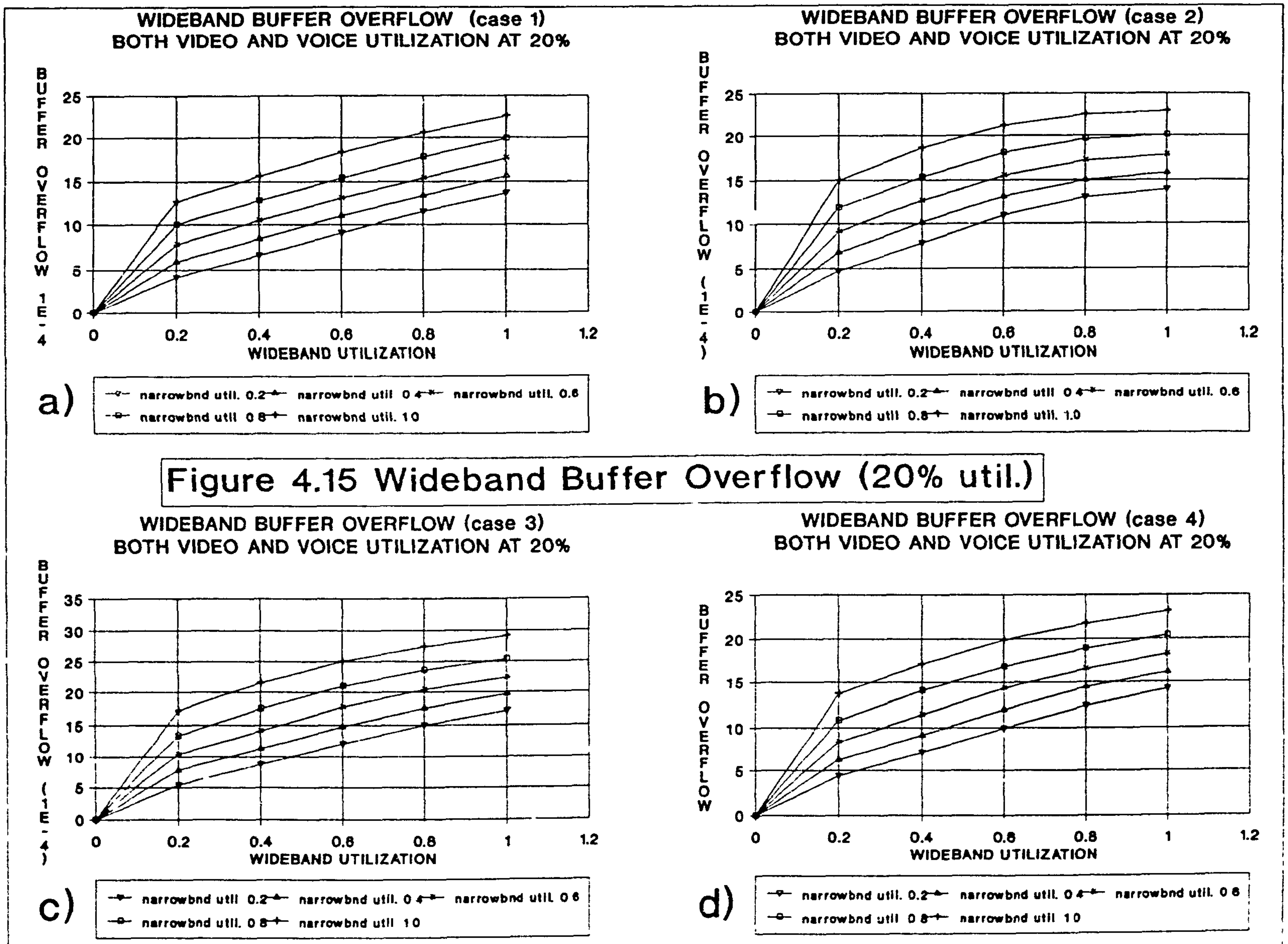


Figure 4.15 Wideband Buffer Overflow (20% util.)

When video and voice utilization are at 20%, wideband data buffer overflow is practically the same for all four priority schemes (figure 4.15). Only under case 3 priority scheme is buffer overflow slightly higher. This is to be expected since case 3 gives equal priority to narrowband data when the narrowband load in high. The same observations can be made for wideband queueing delay (figure 4.17), which ranges on the average from 0.1 to 0.42 seconds when video and the voice utilization are 20%. In case 3 priority scheme the average maximum queueing delay is 0.45 seconds. Under the given conditions, however, **wideband data is not largely affected by the priority scheme.** This is certainly not true for narrowband data. When both video and voice utilization are at 20%, the probability of narrowband data buffer overflow (figure 4.14) is lowest for case 3 priority scheme and highest for case 2 priority scheme (wideband data has full priority). A drastic difference appears, however, in narrowband queueing delay (figure 4.16) with both video and voice utilization at 20%. Case 1 gives an average queueing delays below 2.5 seconds. Case 2 gives the worst average queueing delays, above 11 seconds. Cases 3 and 4 give average queueing delays going up to 7 seconds. These results clearly show **case 1 to be the best choice with respect to narrowband data queueing delay.** The narrowband data buffer overflow in case 1 is near the lowest values found in case 3. The lower queueing delay in case 1, however, makes this priority scheme the obvious choice.

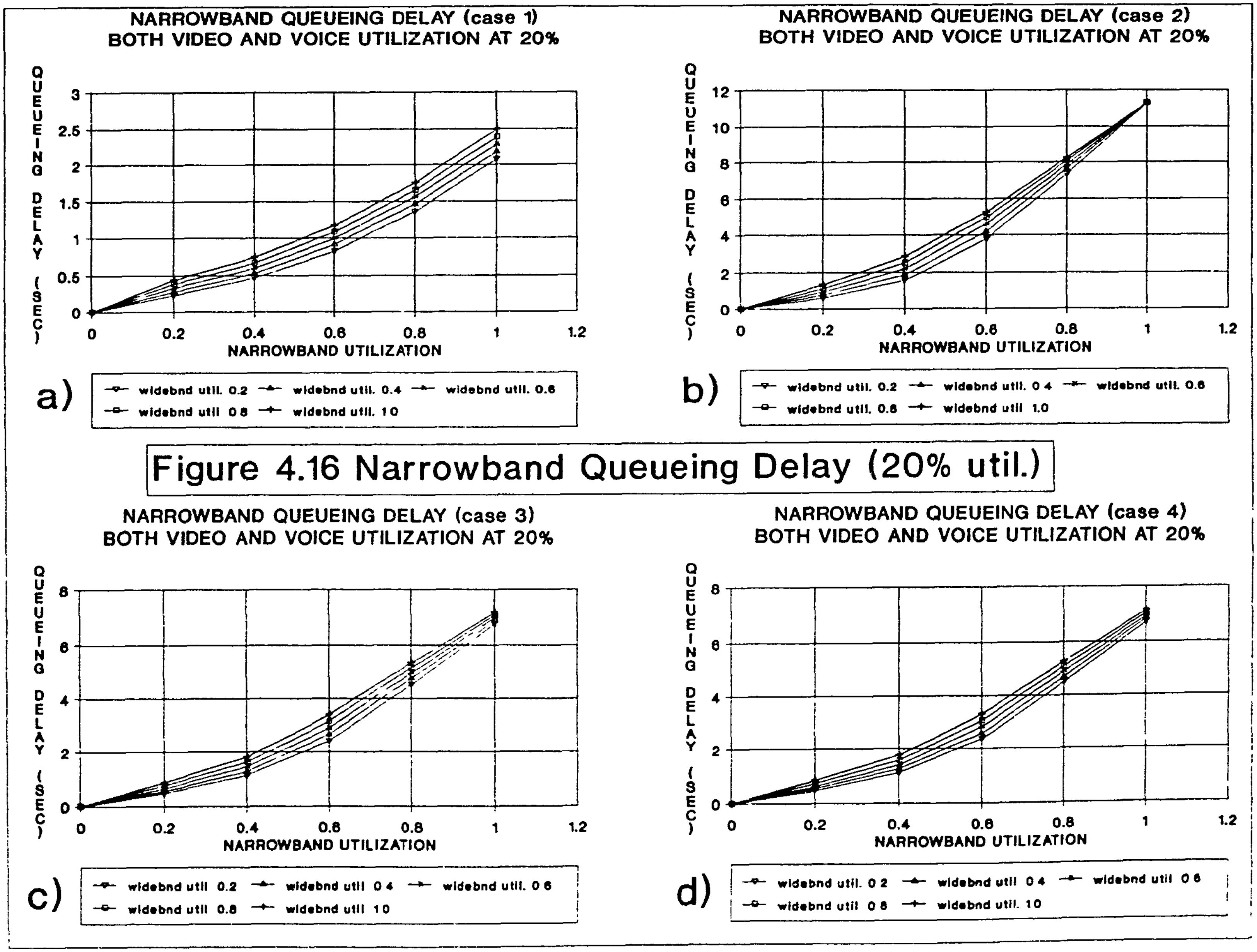


Figure 4.16 Narrowband Queueing Delay (20% util.)

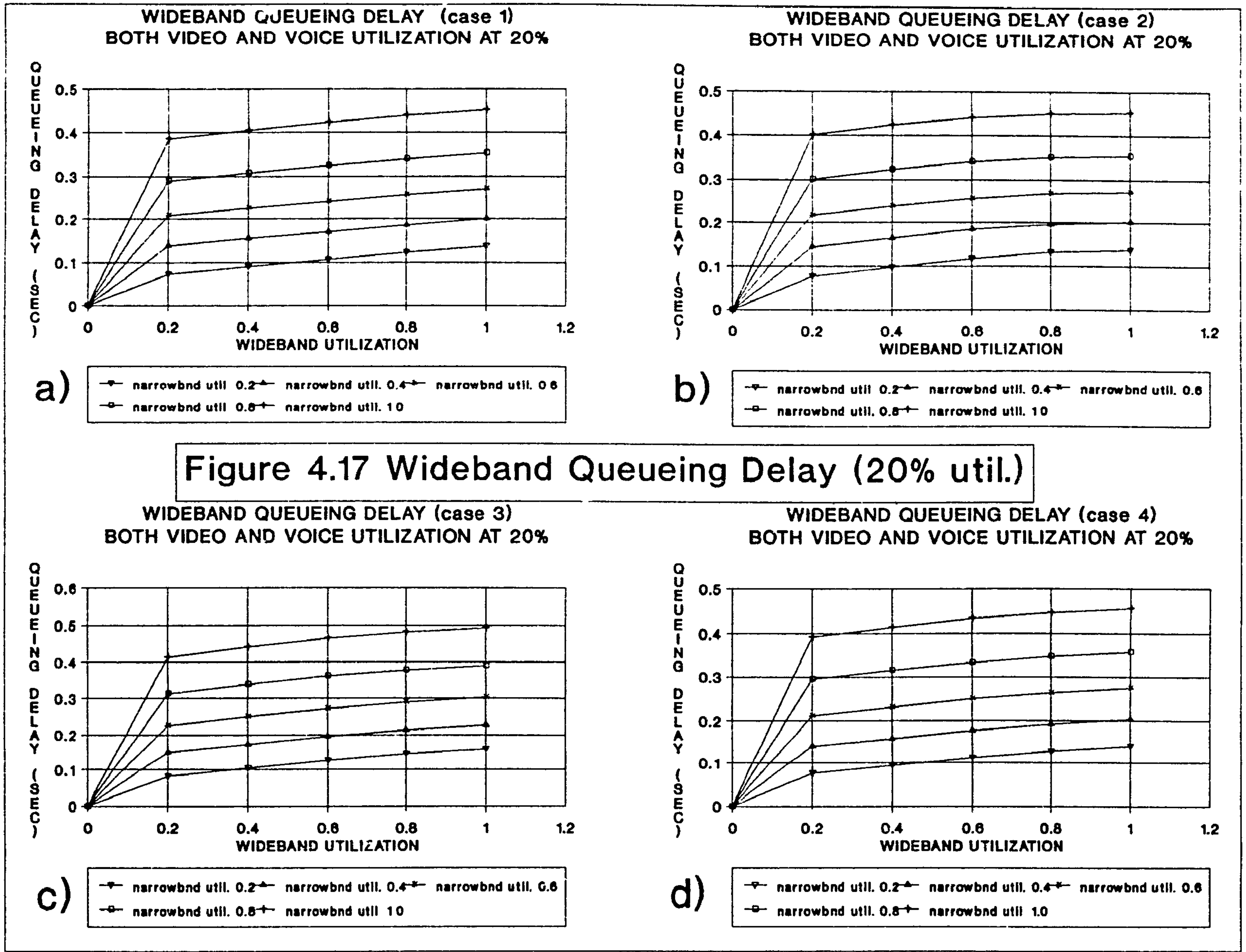
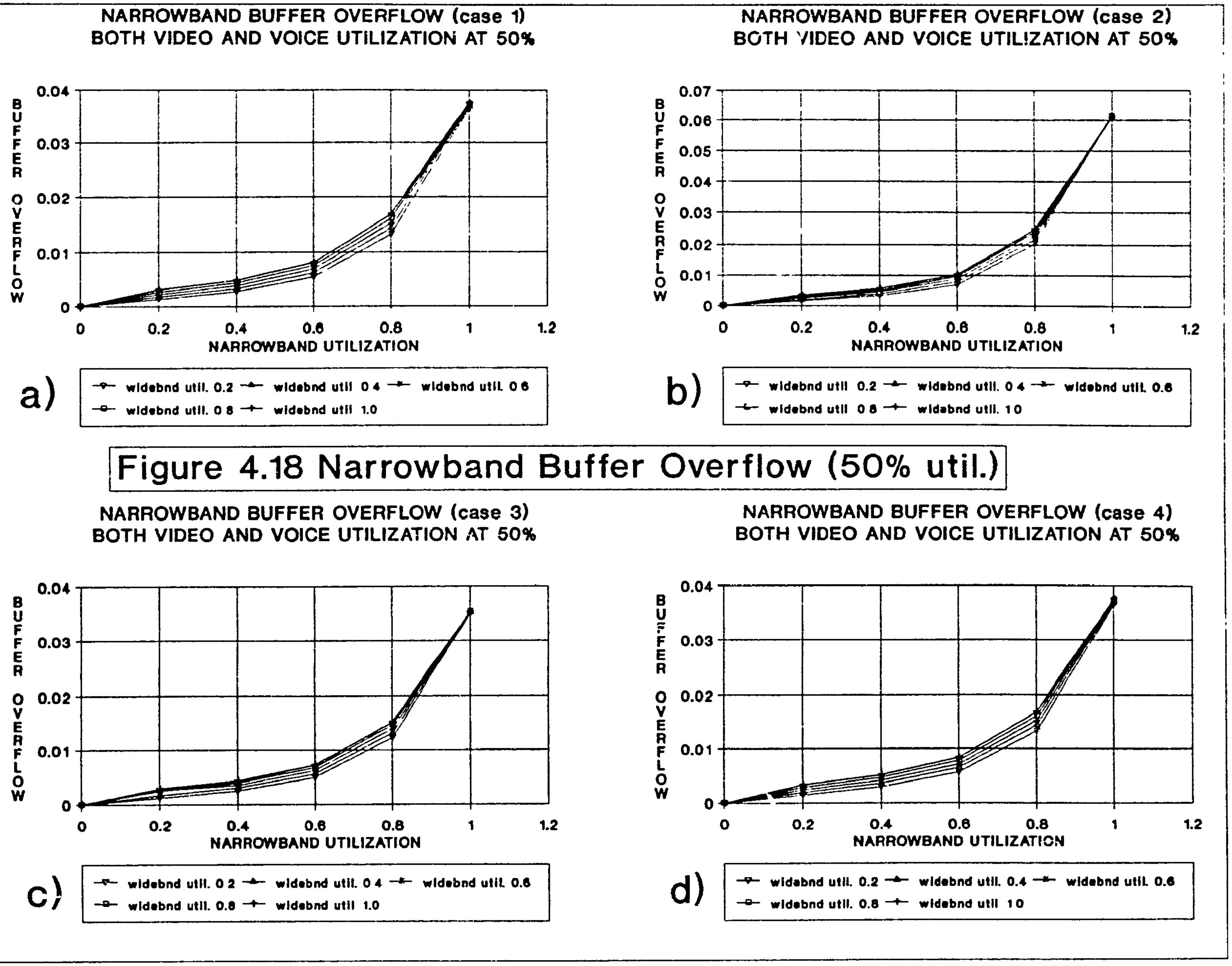


Figure 4.17 Wideband Queueing Delay (20% util.)

The high performance of the case 1 priority scheme is again found when both video and voice utilization are at either 50% or 100% (figure 4.18 to 4.25). At both levels, narrowband buffer overflow increases above that for 20% utilization but in the same proportion for all priority schemes. Unsurprisingly, queueing delay is also increased. For case 1, still the best priority scheme, narrowband queueing delay can reach 8 seconds when both video and voice utilization are 50%, and 10 seconds when utilization is 100%.

This analysis shows that for the case 1 priority scheme, even under a heavy load, data buffer overflow for both narrowband and wideband data is low and queueing delays very acceptable.



NARROWBAND BUFFER OVERFLOW (case 1)
BOTH VIDEO AND VOICE UTILIZATION AT 50%

NARROWBAND BUFFER OVERFLOW (case 2)
BOTH VIDEO AND VOICE UTILIZATION AT 50%

Figure 4.18 Narrowband Buffer Overflow (50% util.)

NARROWBAND BUFFER OVERFLOW (case 3)
BOTH VIDEO AND VOICE UTILIZATION AT 50%

NARROWBAND BUFFER OVERFLOW (case 4)
BOTH VIDEO AND VOICE UTILIZATION AT 50%

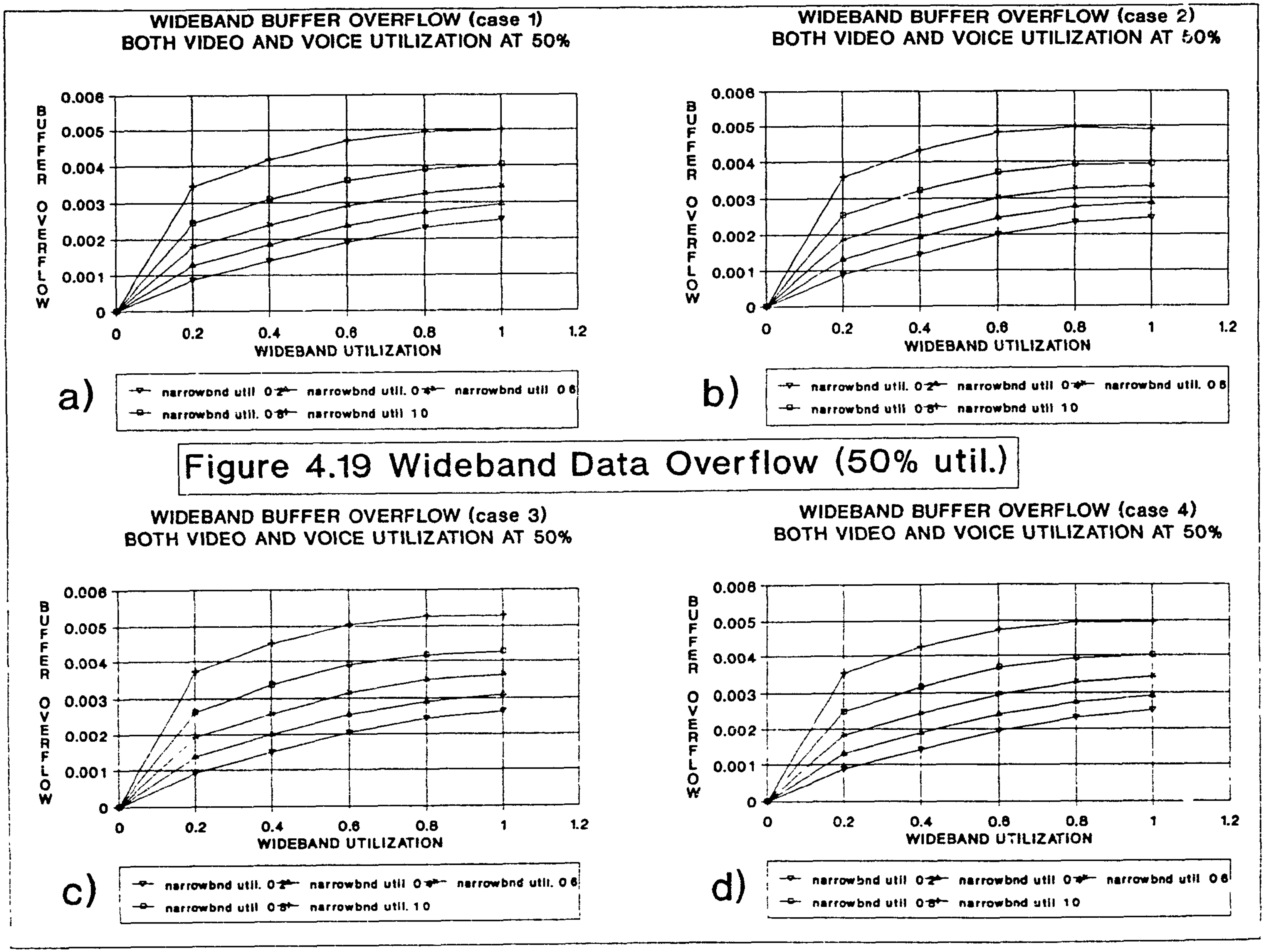


Figure 4.19 Wideband Data Overflow (50% util.)

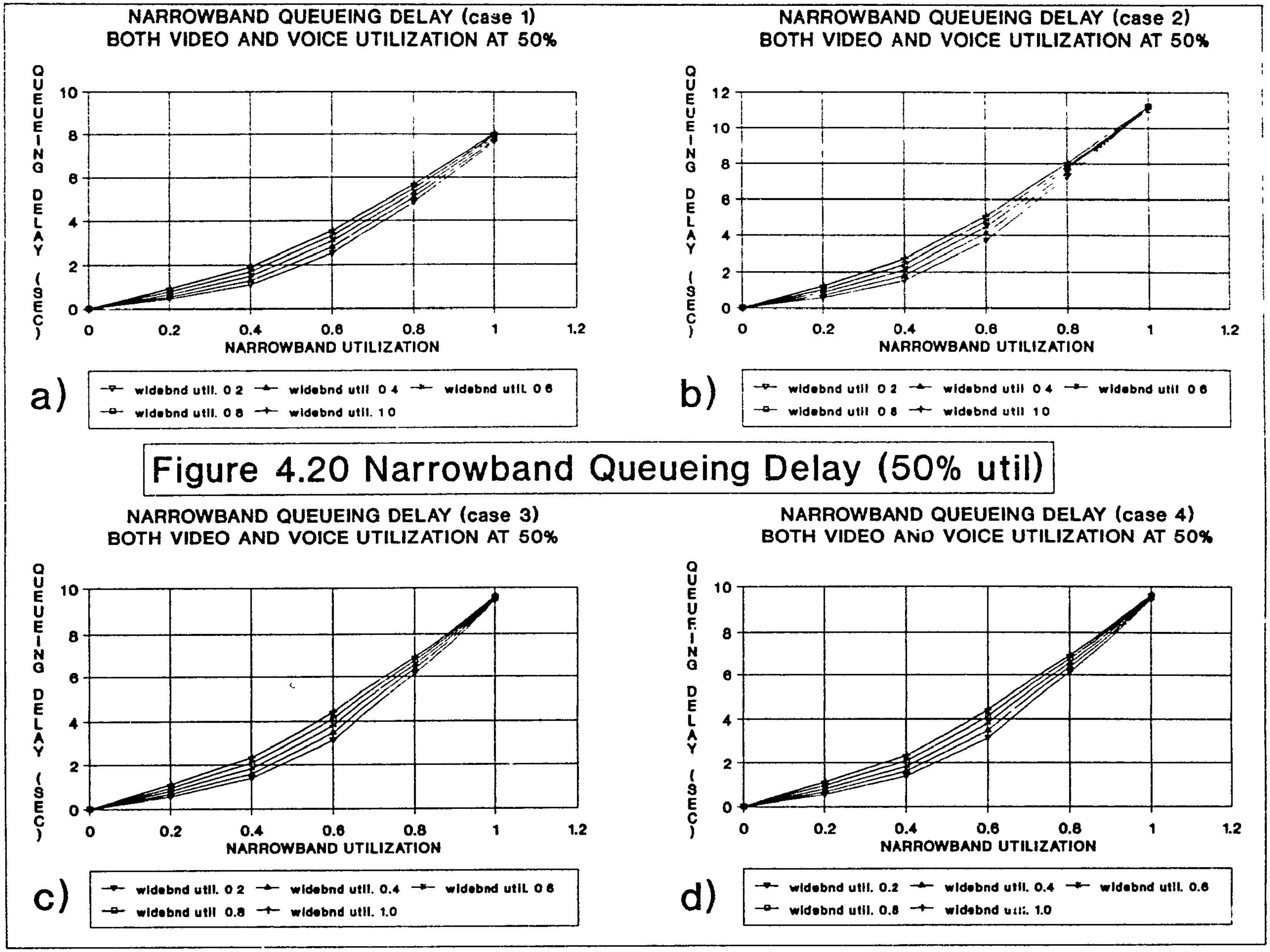


Figure 4.20 Narrowband Queueing Delay (50% util)

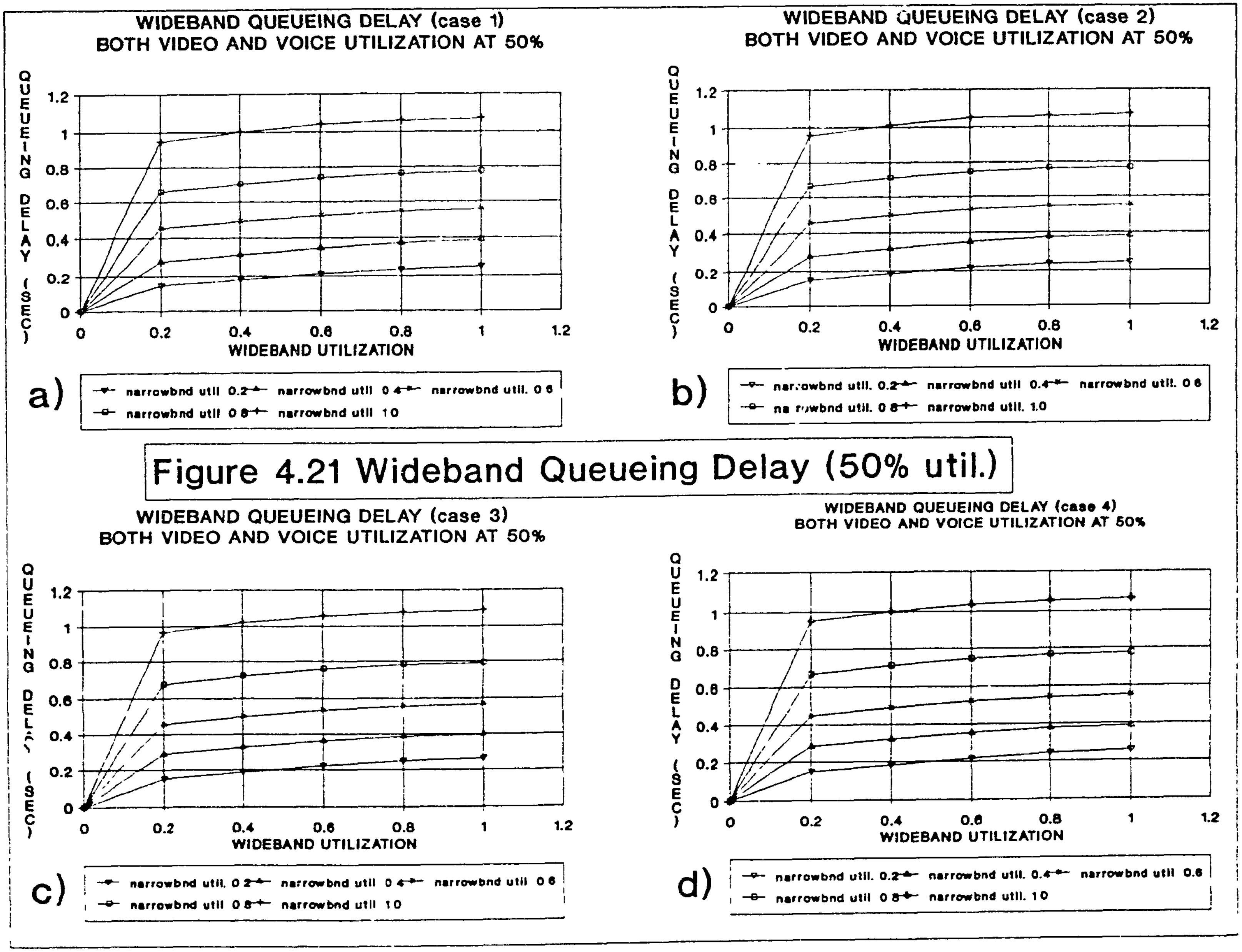
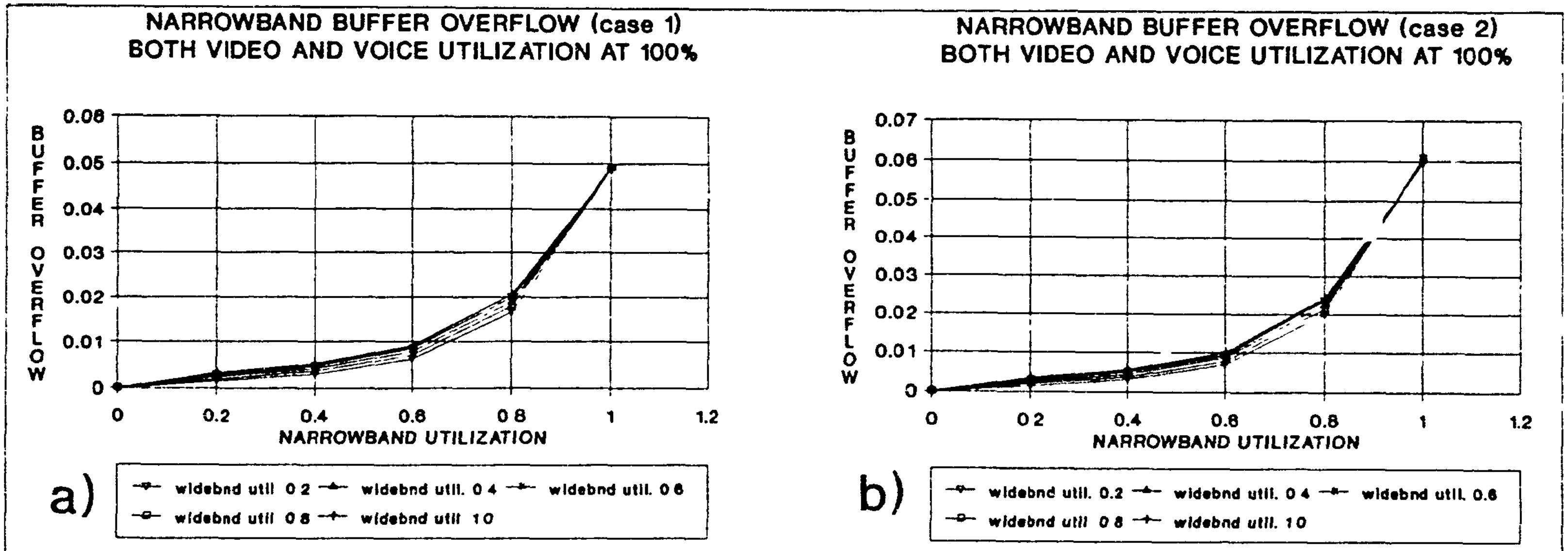


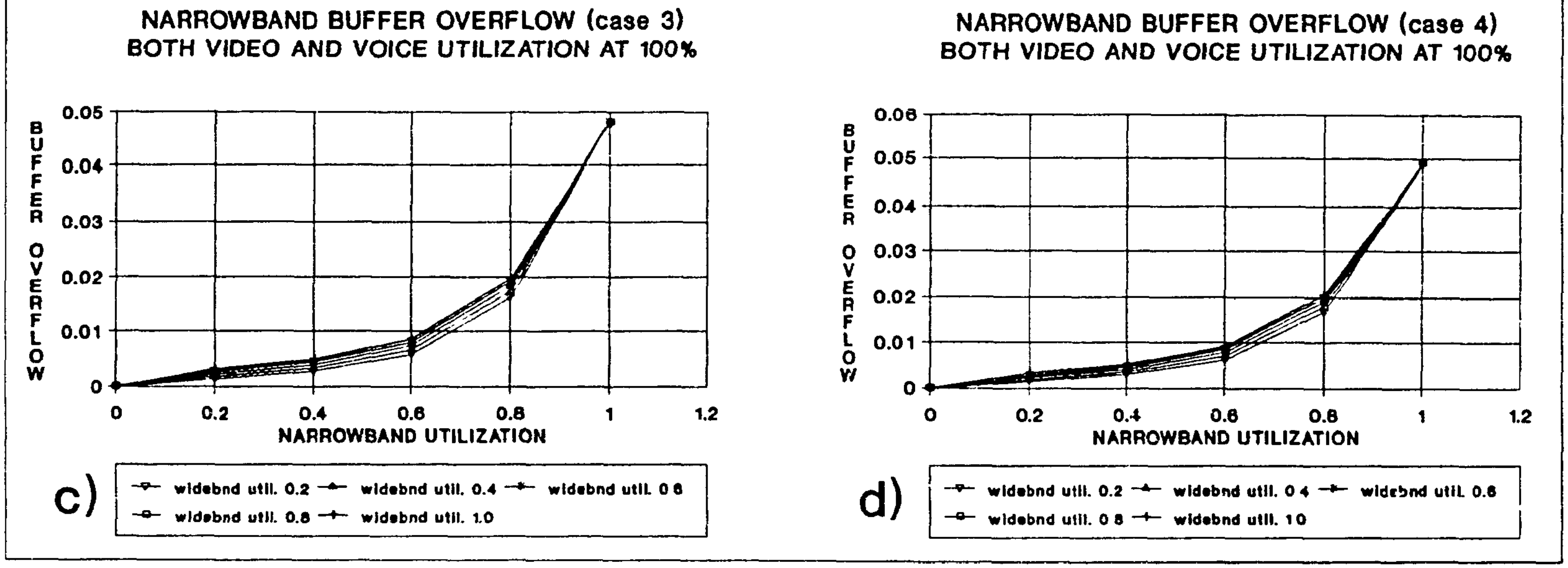
Figure 4.21 Wideband Queueing Delay (50% util.)



NARROWBAND BUFFER OVERFLOW (case 1)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

NARROWBAND BUFFER OVERFLOW (case 2)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

Figure 4.22 Narrowband Buffer Overflow (100% util.)



NARROWBAND BUFFER OVERFLOW (case 3)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

NARROWBAND BUFFER OVERFLOW (case 4)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

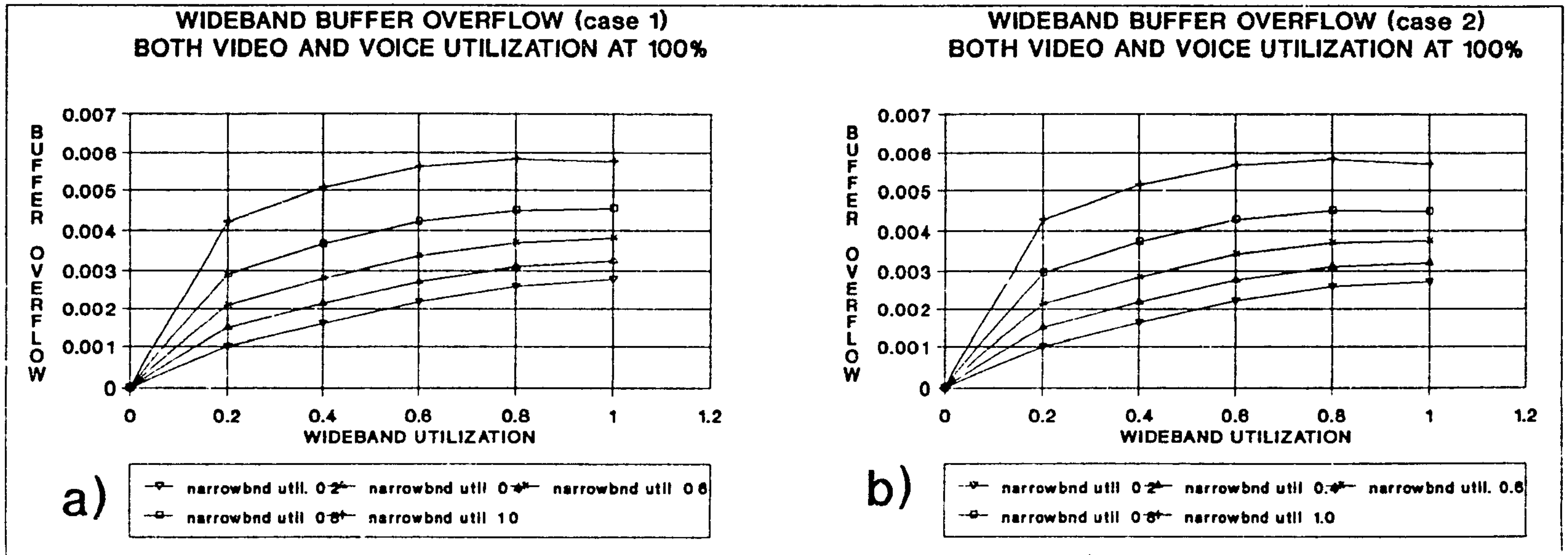
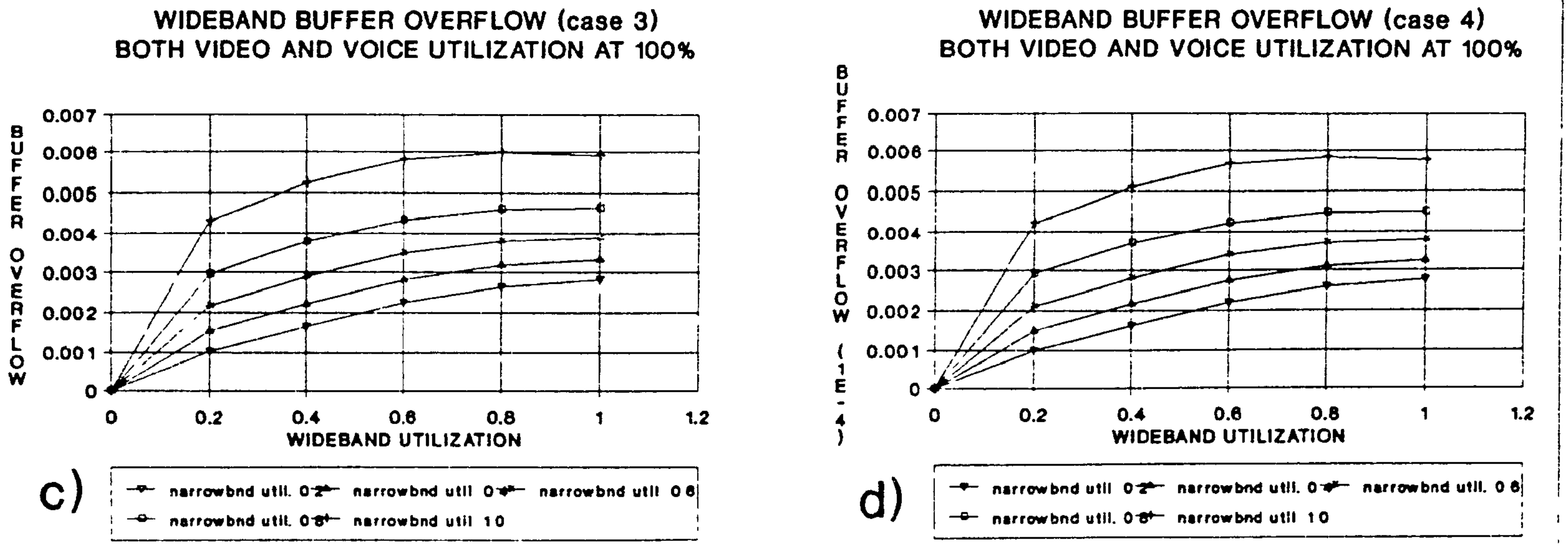
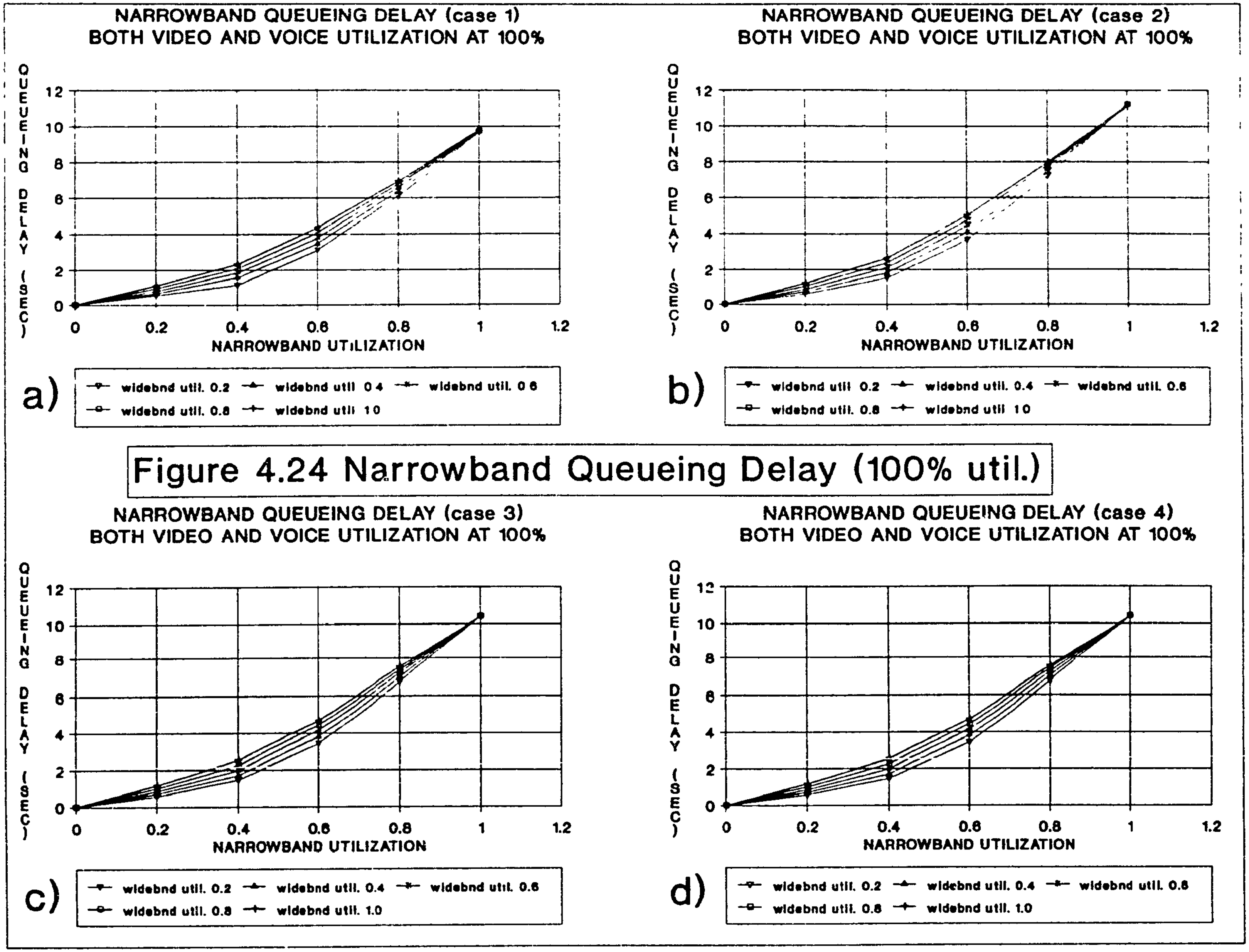


Figure 4.23 Wideband Buffer Overflow (100% Util.)





NARROWBAND QUEUEING DELAY (case 1)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

NARROWBAND QUEUEING DELAY (case 2)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

NARROWBAND QUEUEING DELAY (case 3)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

NARROWBAND QUEUEING DELAY (case 4)
BOTH VIDEO AND VOICE UTILIZATION AT 100%

Figure 4.24 Narrowband Queueing Delay (100% util.)

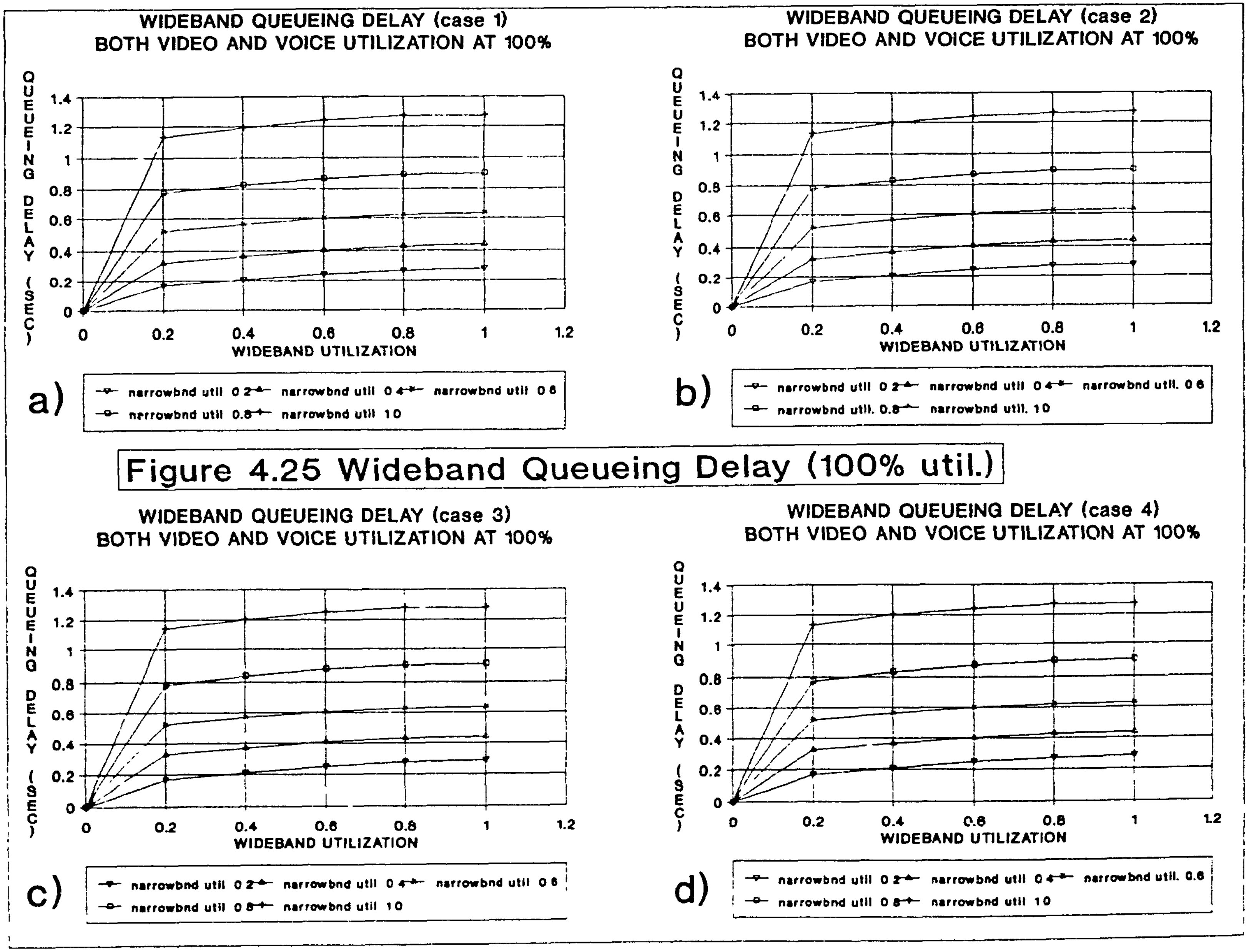


Figure 4.25 Wideband Queueing Delay (100% util.)

The blocking probability for video and voice (figure 4.26) are plotted against the traffic rate ($\lambda/m\mu$) and not the utilization (λ/μ). Note that m is the number of video slots used for video service. Here again, the statistics describe the behaviour of the video and voice service without faults.

The video call establishment time is expressed in frame times. For example if video traffic rate is 60% and voice traffic rate 80%, it takes about 8000 times the frame time for a video call to be established. The important point to note is that voice traffic is independent of video traffic because the estimated minimum time for video call establishment is, on the average, equal to the voice holding time. This independence of voice and video traffic is important when the voice service uses video slots. These slots must be used without the video service being impaired. Thus when a video call arrives, a slot should be available by the time the AU prepares the call and is ready to transmit.

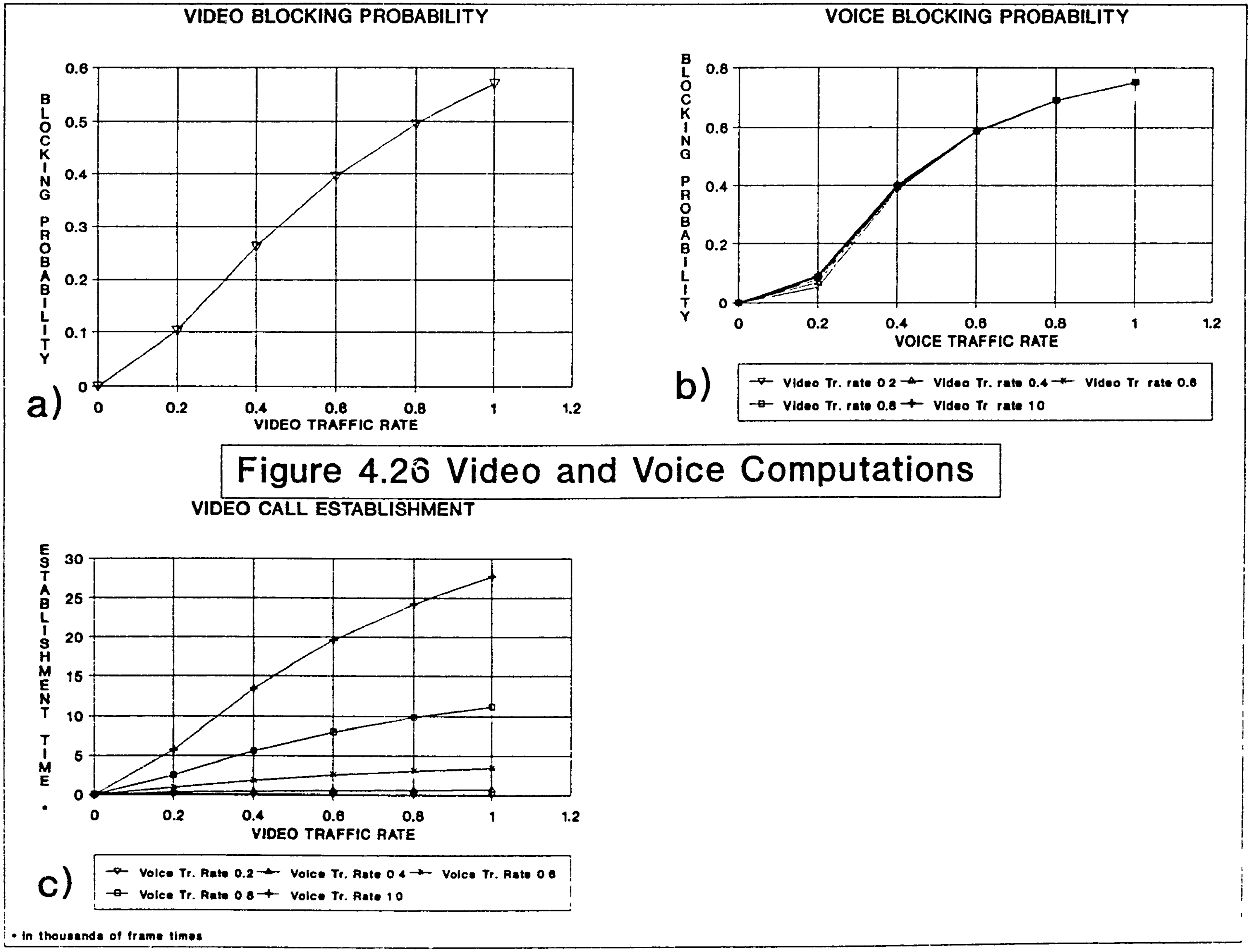


Figure 4.26 Video and Voice Computations

• In thousands of frame times

4.8 Fault Probabilities and their Effect on Service Rate.

In this section, we evaluate the effects that intermittent errors and stuck-at fault errors appearing at the various interfaces implementing the modified DQDB protocol have on service rate and then on overall performance. Although only a single faulty access unit (AU) with different possible faults is considered, the analysis could easily be extended to cover simultaneous errors in many AUs. The service offered for each state in the queueing network, where a state is defined by the number in an AU buffer, must first be determined. The service rate of the healthy AU is the effective frame size (F') [2] divided by the total number of AUs.

$$\mu = F'/number\ of\ AUs \tag{4.55}$$

Determining the service rate of the faulty station, however, is not as simple. Knowing the types of protocol error and the probability of each is essential for defining the service rate.

The DQDB protocol is mainly based on the manipulation of the REQ and BUSY bits in each slot; distributed queueing operates according to the state of these bits. Any error in these bits will affect the protocol either by delaying a

[2] **Effective frame size is the number of slots in the frame minus the number destroyed by faulty AUs.**

request from an AU, by delaying the availability of slots to an AU with higher priority, or even by allowing an AU to grab a slot ahead of its turn.

The following probabilities cover all possible protocol errors:

For any AU,

P_1 <u>probability that REQ write zero</u>

(defined as state in interface: reading REQ = 0, should write REQ = 1, but 0 written)

This results in a loss of one slot for the faulty AU; however there is no change in the effective frame size since other AUs in the DQDB system can still grab the slot for transmission.

P_2 <u>probability that REQ write one</u>

(defined as state in interface: reading REQ = 0, should write REQ = 0, but 1 written)

This results in no capacity loss or gain for the AU making the error; however the effective frame size is decreased by one slot. Although the AU did not intend to reserve a slot, other AUs lost out in the process.

P_3 <u>probability that REQ read zero</u>

(defined as state in interface: REQ = 1, but 0 reads)

This results in a gain of one slot for the faulty AU. The effective frame size decreases by one. The effect of the AU gaining a slot is ignored to simplify analysis.

P_4 probability that REQ read one

(defined as state in interface: REQ = 0, but 1 read)

This results in a loss of one slot for the faulty AU. The effective frame size is unchanged.

P_5 probability that BUSY write zero

(defined as state in interface: reading BUSY = 0, should write BUSY = 1 but 0 written)

This results in a loss of one slot for the faulty AU. The effective frame size is not affected. AUs downstream, however, capitalize on the faulty AU's loss.

P_6 probability that BUSY write one

(defined as state in interface: reading BUSY = 0, should write BUSY = 0 but 1 written)

This results in no change for the AU making the error. The effective frame size is decreased by one slot since the AU did not intend to transmit.

P_7 probability that BUSY read zero

(defined as state in interface: BUSY = 1, but 1 read)

This results in a gain of one slot for the faulty AU. The effective frame size is decreased by one slot.

P_8 probability that BUSY read one

(defined as state in interface: BUSY = 0, but 1 read) This results in a loss of one slot for the faulty AU. The effective frame size is not affected.

P_9 probability that no error whatsoever.

The probabilities just defined are assumed to be mutually exclusive and are based on one time slot in a frame of F slots. So $\sum P_{ji} = 1$ for any AU i (j is from 1 to 9).

The foregoing is a dependent model, i.e., when a typical AU steals or loses a slot it affects all other AUs in the network by changing the number of slots remaining in the frame. The model, however, provides no insight; the various events remain extremely difficult to follow. A straightforward independent model simplifying through worst-case interpretations of events is required.

What must now be found is the combined degrading effect on the frame size of all the above errors from all AUs,i.e. the reduction of frame size F to the effective frame size F' $\leq$ F due to all errors. Each AU's capacity or service rate is calculated as a function of F' and probabilities P_1 to P_9 for that AU.

Probabilities P_1 to P_9 can be grouped according to the effect they have on the AU i and on the frame. β_i is defined as the

probability that AU i errors reduce frame F by one slot.

$$\beta_i=P_{2i}+P_{3i}+P_{6i}+P_{7i} \tag{4.56}$$

α_{1i} is the probability that AU i steals one slot.

$$\alpha_{1i}=P_{3i}+P_{7i} \tag{4.57}$$

α_{2i} is the probability that AU i loses one slot.

$$\alpha_{2i}=P_{1i}+P_{4i}+P_{5i}+P_{8i} \tag{4.58}$$

and α_{3i} is the probabilıty that AU i capacity does not change.

$$\alpha_{3i}=P_{2i}+P_{6i}+P_{9i} \tag{4.59}$$

Note that $\alpha_{1i} + \alpha_{2i} + \alpha_{3i} = 1$.

It is also worth noting that none of the probabilities making up β_i are found in α_{2i}.

In fact $\beta_i = \alpha_{1i} + \alpha_{3i} - P_{9i}$.

The probability that AU i destroys j slots of the frame is

$$\binom{F}{j}\beta_i^j(1-\beta_i)^{F-j} \tag{4.60}$$

Note that $\beta_i = 0$ when j = 0, and $\beta_i = 1$ when j = 1. The average number of slots destroyed is clearly found as $F\beta_i$.

Assuming all AU errors are independent and no AU can gain from the errors of other AUs through substitution; i.e., the worst-case hypothesis, then the probability $P(\ell)$, that ℓ slots of the F slots per frame are destroyed because of all u AUs is

$$P(\ell)=\sum_{j_1=0}^{F}\sum_{j_2=0}^{F-j_1}\cdots\sum_{j_u=0}^{F-\sum_{n=1}^{u-1}j_n}\prod_{i=0}^{u}\binom{F-\sum_{n=1}^{i-1}j_n}{j_i}\beta_i^{j_i}(1-\beta_i)^{\left(F-\sum_{n=1}^{i-1}j_n\right)-j_i} \quad (4.61)$$

where $j_1 + j_2 + \ldots + j_u = \ell$; this reduces the effective frame F size to F', where

$$F'=\sum_{\ell=0}^{F}(F-\ell)\,P(\ell) \quad (4.62)$$

Having found the effective frame size F' as reduced by errors, it is possible to find the capacity of each AU, i.e., the service rate μ_r of AU r. Note that in the symmetric case (all AUs having the same probabilities P_1 to P_9) the capacity of each AU becomes μ=F'/u. In the general case of asymmetric service, however, the capacity of each AU r is

$$\mu_r=\sum_{j_1=0}^{(F'-F'/u)}\sum_{j_2=0}^{(F'/u)}(F'/u+j_1-j_2)\binom{F'-F'/u}{j_1}\binom{F'/u}{j_2}\alpha_{1r}^{j_1}\alpha_{2r}^{j_2}\alpha_{3r}^{(F'-j_1-j_2)}$$

(4.63)

where j_1 = number of slots that user r can steal (F'-F'/u). and

j_2 = number of slots that user r can lose (F'/u).

It is assumed that traffic is symmetric, i.e., all AUs have the same arrival rate λ_i. If the rate of arrival is not the same for all AUs, then the term F'/u is replaced by (F'/u)

(λ_1/λ_{av}) (where λ_i is the arrival rate for AU i and λ_{av} is the average arrival rate in the network). The term is multiplied by the factor λ_i/λ_{av}, representing the proportional arrival rate for AU i. The capacity for all AUs is then normalized with respect to F'.

4.9 Computation of the Performance Criteria under Faults

A MAN with four AUs, one faulty and three healthy, is taken as a model for analyzing reliability when AU faults affect the DQDB protocol. The model is defined as a two-dimensional network of queues (figure 4.27). One dimension shows the states (number in the buffer) of the faulty AU, the other the states of all healthy AUs.

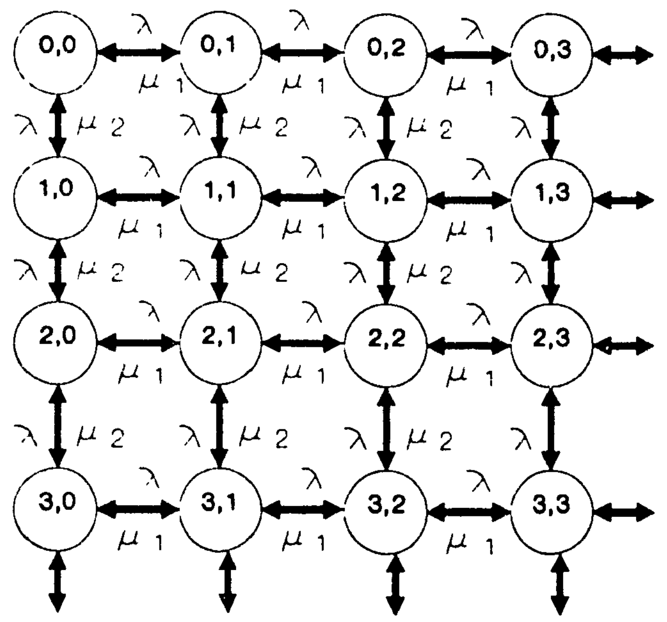


Figure 4.27.
Two Dimensional Network of Queues

The maximum service rate μ_{max} is computed for various scenarios where one of the fault parameters α_1, α_2 or α_3 is varied. The highest μ_{max} is the top service rate possible in a stable network. Delays are also calculated for the various arrival rates. These computations are made for both the faulty AU and the other AUs. Two error correction techniques, triple modular redundancy (TMR) and compared to the quintuple modular redundancy (QMR), are applied to the various interfaces and the effect on μ_{max} compared to the uncorrected system. Modular redundancy is a method used to correct errors by bit voting. With TMR, three-bit REQ and BUSY fields are used; with QMR, five-bit fields are used. If an intermittent error occurs in either field, a bit vote is taken and the majority determines the status of the field. The number of bits used in modular redundancy is always odd.

A function of α_1 and $\alpha_{3,}$ β reflects network stability. Low values of β imply that the faulty AU has little effect on the network, high values imply a great effect. It is therefore important to analyze the effect of either the parameter α_1 or α_3 over the network while keeping β constant. Naturally, error correction will cause β to vary and, in fact, decrease.

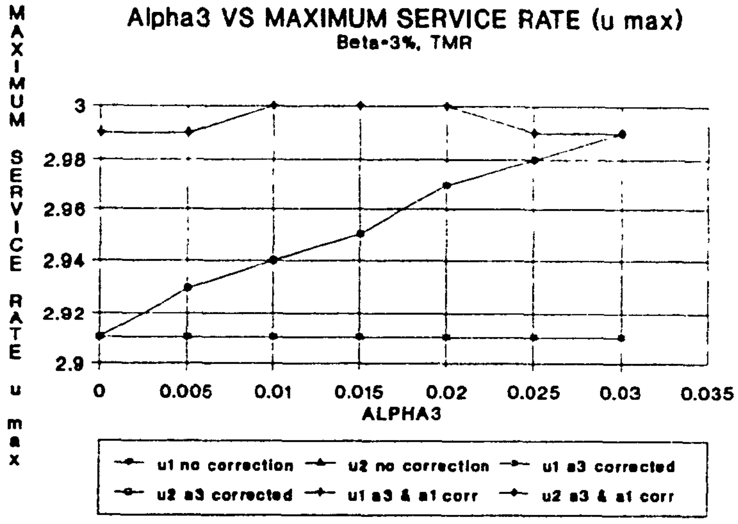


Figure 4.28 Alpha3 with Beta-3% and TMR

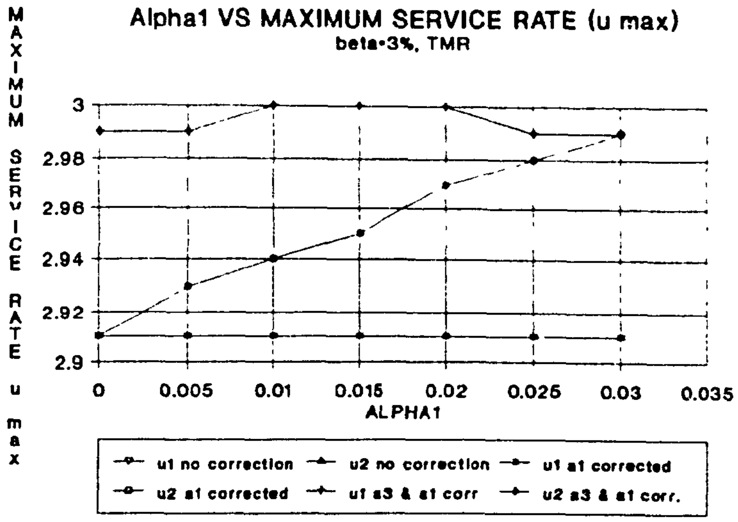


Figure 4.29 Alpha1 with Beta-3% and TMR

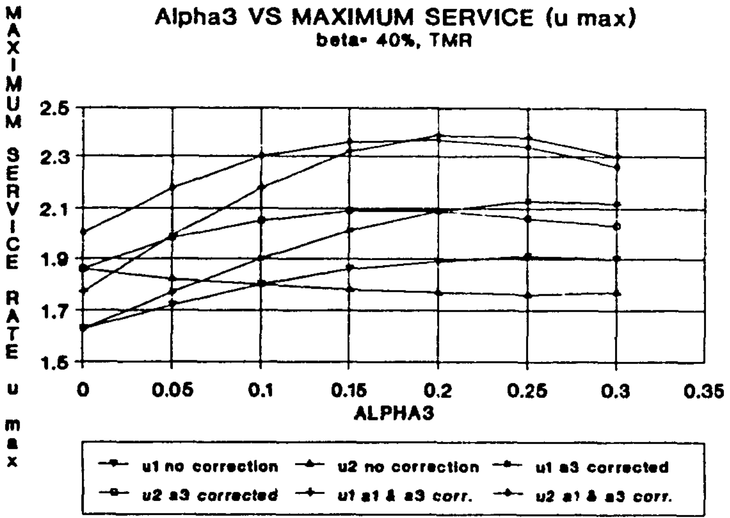


Figure 4.30 Alpha3 with Beta-40% and TMR

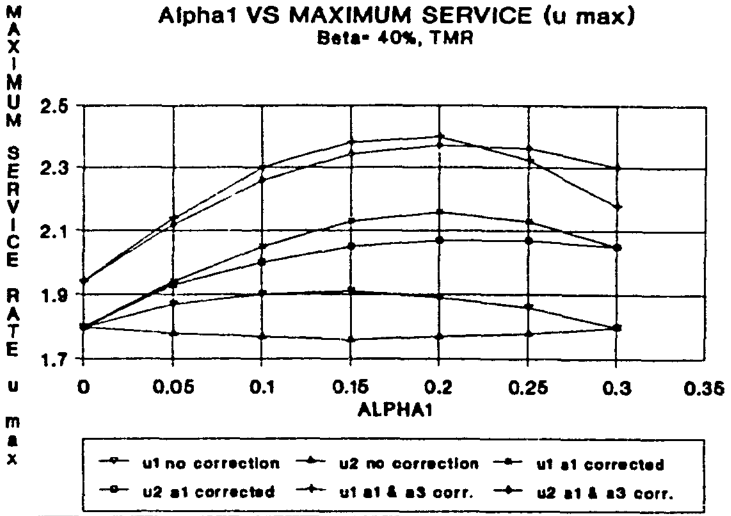


Figure 4.31 Alpha1 with Beta-40% and TMR

In the four AU model, the network error-free service rate per AU is 3 slots/sec. When β is kept constant at 0.03 (3%) and α_3 is varied (figure 4.28), the maximum value of μ_{max} for both faulty and healthy AUs is 2.91 slots/sec (97% of the error-free value (of 3 slots/sec) for all values of α_3. When TMR is applied to α_3, μ_{max} increases as α_3 increases to reach 2.99 slots/sec at α_3 = 0.03. Consequently, β decreases (because of error correction) and the effective frame size is around 12. Even without error correction, the effect of errors on the network is small. The effect is even smaller when β is lower than 3%. These results are comparable to those obtained when varying α_1 (figure 4.29) while keeping β constant.

The model is also tested with β = 0.4 (40%) (error practically a stuck-at-fault) and varying α_3 (figure 4.30). Under this scenario, a noticeable change and disruption to service is observed. The healthy AUs have a maximum service rate μ_{max} of 1.8 slots/sec (60% of the error free value). The faulty AU, on the other hand, experiences an increase in μ_{max} as α_3 increases, going from 1.65 slots/sec (55%) to 1.9 slots/sec (63%). This occurs because the faulty AU is stealing slots and α_1 is varied to keep β constant. When TMR is applied to α_3, the maximum service rate both (faulty and healthy AUs) improves to about 2.1 slots/sec (70%).

Varying α_1 while keeping β constant at first 3% and then 40% and applying TMR to α_1 gives results (figures 4.29, 4.31) very close to those obtained with α_3.

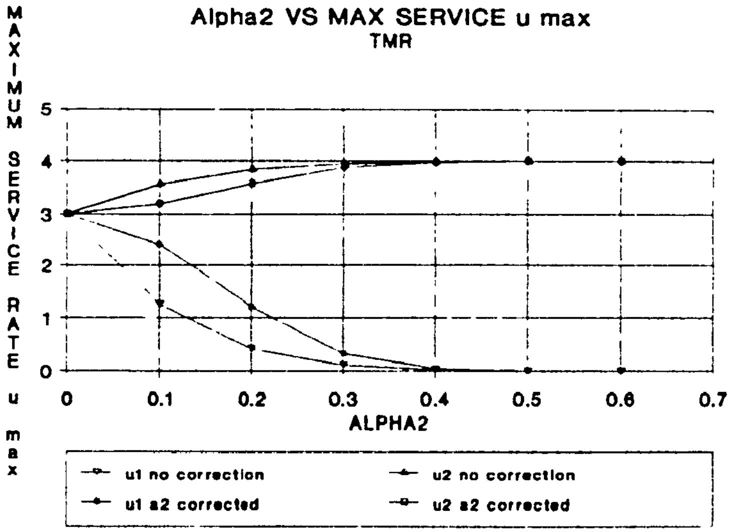


Figure 4.32 Alpha2 TMR

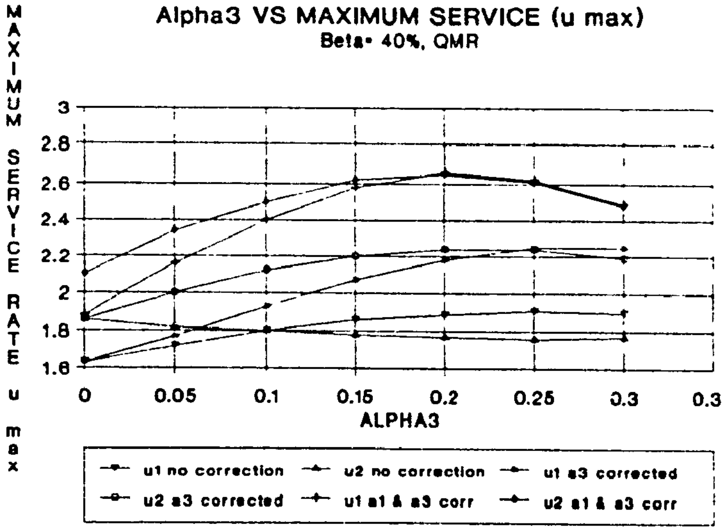


Figure 4.33 Alpha3 Beta=40% QMR

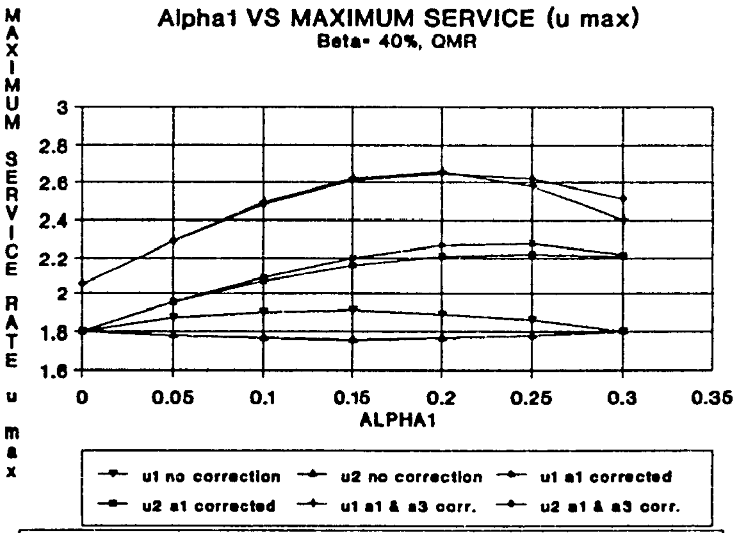


Figure 4.34 Alpha1 with Beta=40% and QMR

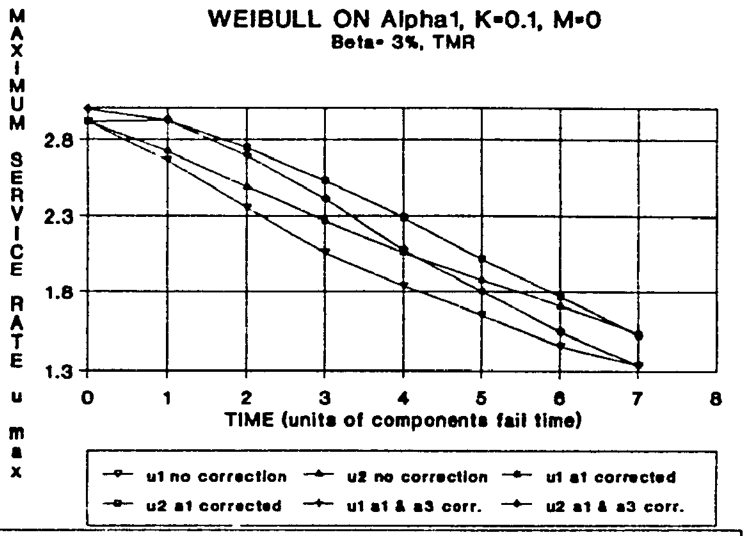


Figure 4.35 WEIBULL for Alpha1 M=0 Beta=3% TMR

When QMR is applied to the circuits underlying α_1 and α_3 the results (figures 4.33, 4.34) show a slight improvement over TMR. The maximum service rate in both cases is about 2.6 slots/sec (73%).

Applying error correction to both α_1 and α_3, (figures 4.30, 4.31, 4.33, 4.34) gave maximum service rates of about 2.3 slots/sec (76%) with TMR and about 2.5 slots/sec (83%) with QMR.

These analyses show that modular redundancy over either α_1 or α_3 but not both results in some improvement but not enough to be economically justified. Similarly the decision whether to apply modular redundancy to α_1 or α_3 is also an economic decision. It was shown that QMR provides only a small improvement over TMR. Implementing QMR is only justified if the cost of QMR is not much greater than that of TMR.

The effect of α_2 the probability of losing one slot, on the maximum service rate (figure 4.32) was next. The results show that only the faulty AU suffers from this condition; the other AUs gain service time by sharing the slots lost by the faulty AU. In the example with four AUs, when the faulty AU has a service rate of zero at $\alpha_2 = 0.5$, the other three AUs each have 4 slots/sec (a 33% increase over normal service). Applying TMR or QMR to α_2, does not bring any improvement since the faulty AU starts at a low service rate, and, in fact, only reduces the gain of the other AUs. There is thus no benefit in applying modular redundancy to α_2.

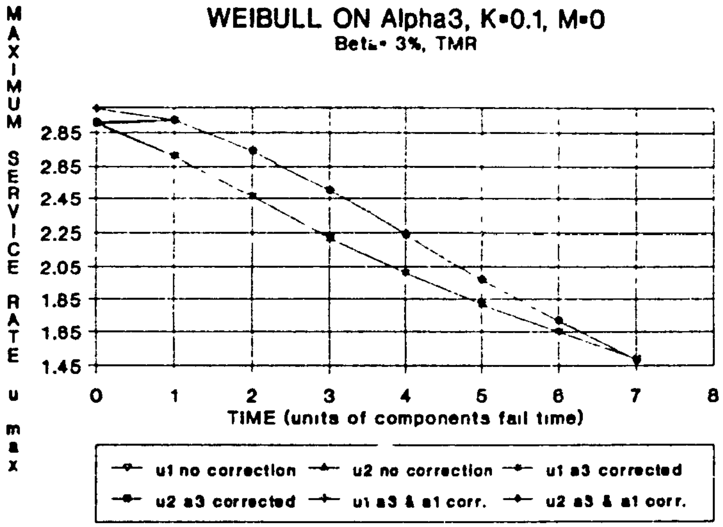


Figure 4.36 WEIBULL for Alpha3 M=0 Beta=3% TMR

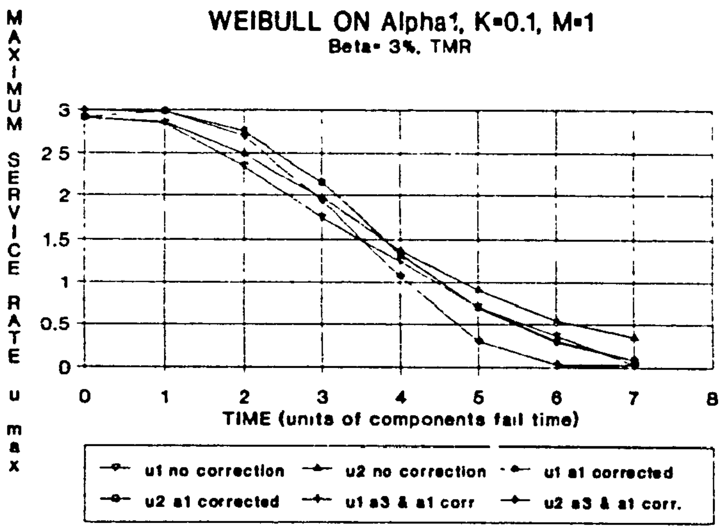


Figure 4.37 WEIBULL Alpha1 M=1 Beta=3% TMR

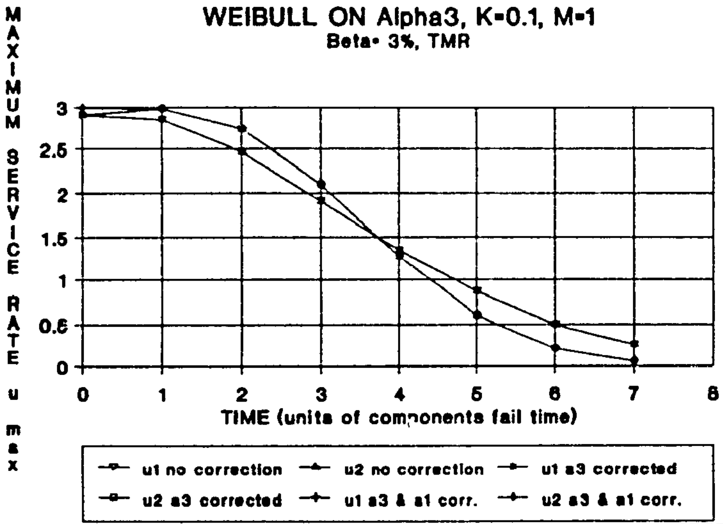


Figure 4.38 WEIBULL Alpha3 M=1 Beta=3% TMR

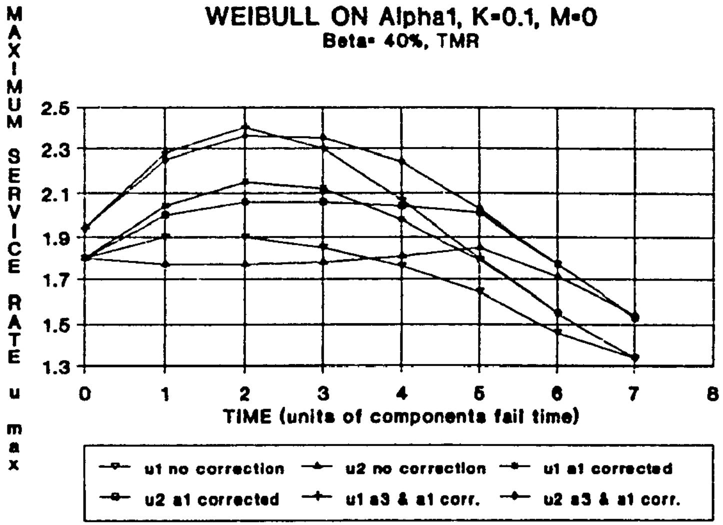


Figure 4.39 WEIBULL Alpha1 M=0 Beta=40% TMR

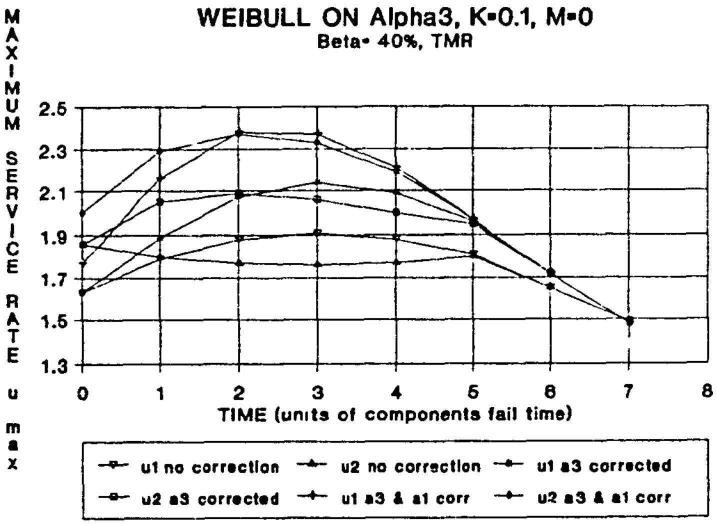


Figure 4.40 WEIBULL Alpha3 M=0 Beta=40% TMR

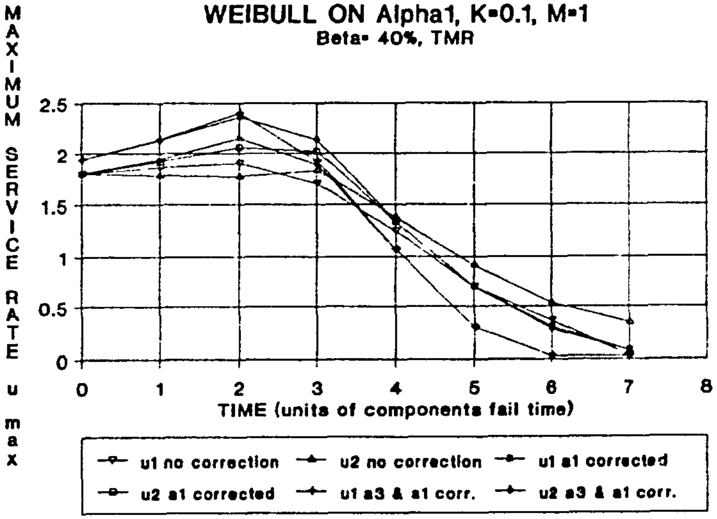


Figure 4.41 WEIBULL Alpha1 M=1 Beta=40% TMR

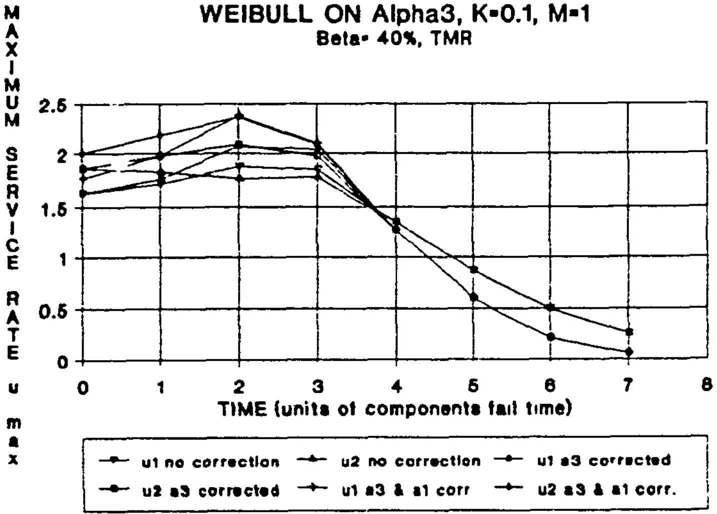


Figure 4.42 WEIBULL Alpha3 M=1 Beta=40% TMR

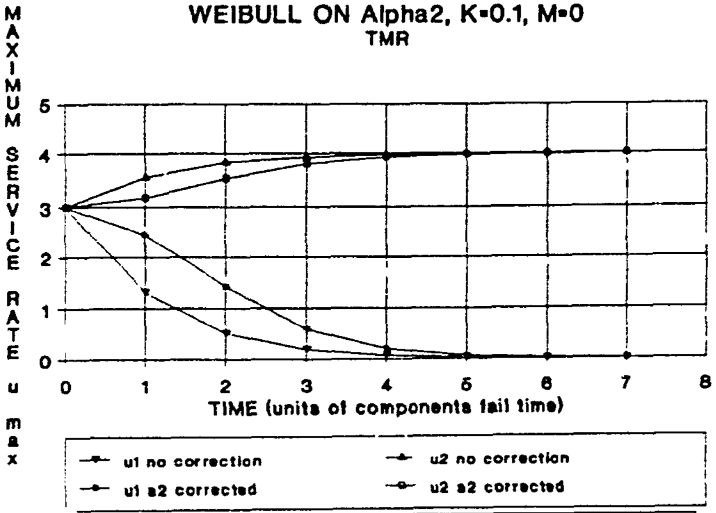


Figure 4.43 WEIBULL Alpha2 M=0 TMR

The Weibull model (appendix 4B) is applied to the various probabilities to see the effect errors, as well as component redundancy, have over the life of the supporting circuitry. The Weibull model is applied to α_1 with a constant failure rate (M=0) and linearly increasing failure rate (M=1) for β=3% (figures 4.35, 4.37) and β = 40% (figures 4.39, 4.41). The same analysis is performed on α_3 (figures 4.36, 4.38, 4.40, 4.42). The results show that whether a constant failure rate or linearly increasing failure rate is assumed, modular redundancy on either α_1 or α_3 improves service rate improved during system degradation but does not substantially affect circuit life span. Again, only one of the circuit components need be corrected since only a small improvement is derived from the correction of both α_1 and α_3. The Weibull model applied to α_2 (figures 4.43, 4.44) shows rapid degradation of the faulty AU and a corresponding improvement of the service rate of healthy AUs. Again, error correction on α_2 is not desirable.

One argument for applying modular redundancy to both α_1 and α_3 is to improve circuit reliability since this provides a form of circuit backup.

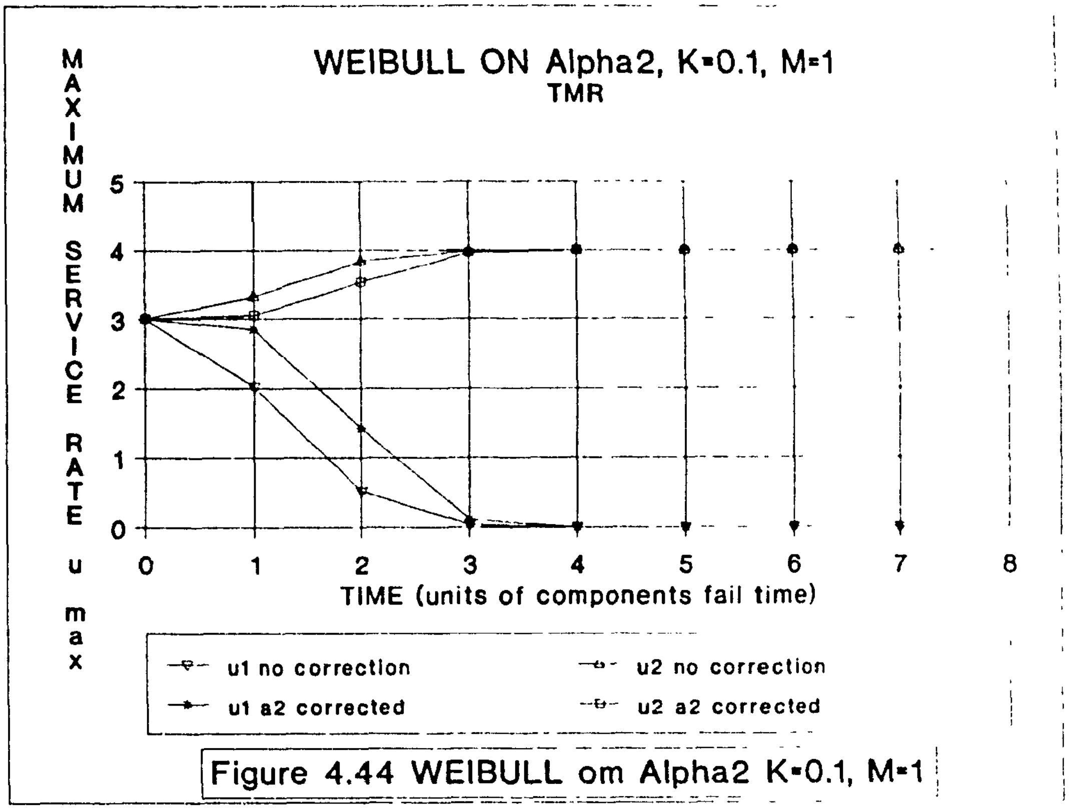


Figure 4.44 WEIBULL om Alpha2 K=0.1, M=1

4.10 Conclusion

A distributed queue dual bus (DQDB) metropolitan area network for video, voice and both wideband and narrowband data services was examined and an access technique for flexibly sharing network bandwidth among the different services was proposed.

The high potential capacity and flexibility of such a network was demonstrated. The performance results brought out the relative merits of the different priority schemes proposed. It was found that the best priority scheme is "restricted priority".

As was seen, the DQDB protocol is mainly based on the manipulation of the request and BUSY bits in the access control field of each slot. Because the DQDB protocol depends heavily on such bit manipulation, the sensitivity of the network to intermittent yet frequent bit errors in the crucial read and write operations was investigated, as were the effects of redundancy.

Errors in request or BUSY bits affect the protocol either by delaying a request from an AU, by delaying the availability of slots to an AU with higher priority or even by allowing an AU to grab a slot ahead of its turn.

A DQDB reliability model of three healthy AUs with one faulty AU was analyzed. The results illustrated the error behaviour of the DQDB system. The approach in this paper demonstrated a method for maintaining a high grade of service while isolating faulty circuitry causing protocol errors.

4.11 References

[AGRA1] S. C. Agrawal "Metamodeling. A Study of Approximations in Queueing Models," The MIT Press, Cambridge Mass, 1985.

[BASK1] F. Baskett, K.N. Chaudy, R.R. Muntz, and F.G. Palacios, "Open, Closes and Mixed Network of Queues with Different Classes of Customers," Journal of the ACM, Vol. 22, No. 2,April 1975.

[BRUE1] S.C. Bruel and G. Balbo "Computational Algorithms for closed Queueing Networks," North Holland, New York 1980.

[BUDR1] Z.L. Budrikis, J.L. Hullet, R.M. Newman, D. Economou, F.M. Fozdar, and R.D. Jeffrey, "QPSX: A Queue Packet and Synchronous Circuit Exchange", ICCC 1986, pp. 288-293.

[BUZE1] J. P. Buzen "Queueing Network Models of Multiprogramming," Garland Publishing Inc., New York, 1980.

[CCIT1] CCITT Blue Book, Volume III, Fascicle III.7, Rec. I.121, "Broadband Aspect of ISDN", Geneva, Switzerland, 1988.

[DESE1] Y. De Serres and L. G. Mason, "A Multiserver Queue with Narrow- and Wide-Band Customers and Wide-Band Restricted Access," IEEE Transaction on Communications, Vol.36,No. 6, June 1988.

[GELE1] E. Gelenbe and I. Mitrani "Analysis and Synthesis of Computer Systems," Academic Press, New York 1980.

[GRUB1] J. G. Gruber and N. H. Le, "Performance Requirements for Integrated Voice/Data Networks," IEEE Journal on Selected Area in Communications, Vol. SAC-1, No. 6, Dec. 1983.

[HOBE1] W. L. Hoberecht, "A Layared Network Protocol for Packet Voice and Data Integration," IEEE Journal on Selected Area in Communications, Vol. SAC-1, NO.6, Dec. 1983.

[HOWA1] J. H. Howard Jr. and D Towsley, "Product Form and Local Balance in Queueing Networks," Journal of the Association for Computing Machinery, Vol.24, No.2, April 1977.

[KADO1] M. Kadoch, A.K. Elhakeem, "A Distributed Metropolitan Area Network for Combined Video, Voice, Wideband data and Narrowband Data", GLOBECOM 1989, pp. 152-156.

[KADO2] M. Kadoch, A.K. Elhakeem, "Performance and reliability of DQDB under faults", GLOBECOM 1990, pp. 1758-1762.

[KAUF1] J.S. Kaufman, "Blocking in a Shared Resource Environment", IEEE Transactions on Communications, Vol. COM-

29, No. 10, October 1981.

[KRAM1] B. Kraimeche, and M. Schwartz, "Analysis of Traffic Access Control Strategies in Integrated Service Networks", IEEE Transactions on Communications, Vol. COM-33, No.10, October 1985.

[KONH1] A. G. Komheim and R. L. Pickholtz, "Analysis of Integrated Voice/Data Multiplexing," IEEE Transaction on Communications, Vol. COM-32, No.2, Feb. 1984.

[SLAM1] S. S. Lam, "Queueing Networks with Population Size Constraints", IBM Journal Res. Develop., pp. 370-378, July 1977.

[SLAM2] S. S. Lamm, "Store and Forward Buffer Requirement in a Packet Switching Network," IEEE Transactions on Communications, Vol. COM-24, No.4, April 1976.

[SLAM3] S. S. Lam and M. Reiser, "Congestion Control of Store and Forward Networks by Input Buffer Limits- An Analysis," IEEE Transaction on Communications, Vol. COM-27, No.1, Jan. 1979.

[SLAM4] S. S. Lam and J. W. Wong, "Queueing Network Models of Packet Switching Networks. Part2: Networks with Population

Size Constraints," Performance Evaluation 2, North Holland Publishing Co., New York, 1982.

[LAMyl] Y. F. Lam and V. O. K. Li, "On Network Reliability Calculations with Dependent Failures," IEEE Proc. 1983.

[MONT1] W. A. Montgomery, "Techniques for Packet Voice Synchronization," IEEE Journal on Selected Area in Communications, Vol. SAC-1, No.6, Dec. 1983.

[MOLL1] James F. Mollenauer, "Standards for Metropolitan Area Network", IEEE Communication Magazine, Vol. 26, No. 4, pp.15-19, April 1988.

[NEWM1] R.M. Newman, Z.L. Budrikis, J.L. Hullet, "The QPSX MAN", IEEE Communication Magazine, Vol. 26, No.4, pp.20-28, April 1988.

[NEWM2] R.M. Newman, J.L. Hullet, "Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX", ICCC 1986, pp. 294-299.

[NGSW1] S. W. Ng, "Reliability and Availability of Duplex Systems: Some Simple Models," IEEE Transactions on Reliability, Vol. R-35, No. 3, Aug. 1986.

[ROM 1] R. Rom and N. Shacham, "A Reconfiguration Algorithm for a Double-Loop Token-Ring Local Area Network," IEEE Transaction on Computers, Vol.37, No.2, Feb. 1988.

[WILL1] G.F. Williams, A. Leon-Garcia, "Performance Analysis of Integrated Voice and Data Hybrid-Switched Links", IEEE Transactions on Communications Vol. COM-32, No. 6, June 1984.

[ZUKE1] M. Zukerman, "On Packet Switching Capacity in QPSX," Proc. IEEE/IECE Global Telecommun. Conf. GLOBECOM'87, Tokyo, Japan, Nov. 15-18, 1987, pp. 45.2.1-45.2.5.

[ZUKE2] M. Zukerman, "Queueing Performance of QPSX," Poc. 12th Int. Teletraffic Congress ITC 12, Torino, Italy, June 1988, pp. 2.2N.6.1-2.2B.6.7.

[ZUKE3] M. Zukerman, "Overload Control of Isochronous Traffic in QPSX," Proc. IEEE Global Telecommun. Conf. Exhibition Globecom'88, Holywood, FL, Nov. 28-Dec. 1, 1988, pp. 38.3.1-38.3.5.

[ZUKE4] M. Zukerman, "QPSX- The Effect of Circuit Allocation on Packet Capacity under Bursty Traffic," IEEE International Conference on Communications 1988.

APPENDIX 4A - DELAY IN A NETWORK OF QUEUES

In the case of one AU having intermittent and/or stuck at-fault errors, the reliability model is a two-dimensional queueing system with one dimension representing the faulty AU and the other dimension representing all healthy AUs. Since the model is uniform, the system has a closed form solution [BASK1],[SLAM1].

$$P_{n,m}=\frac{\lambda^{n+m}}{\mu_1^n\mu_2^m}P_{00} \qquad \textbf{(A1)}$$

The initial state is

$$P_{00}=1-\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}P_{kl}=\frac{1}{1+\left[\frac{\lambda/\mu_2-\lambda^2/\mu_1\mu_2+\lambda/\mu_1}{(1-\lambda/\mu_1)(1-\lambda/\mu_2)}\right]} \qquad \textbf{(A2)}$$

The delay and probability of blocking [KAUF1] for the faulty AU are represented respectively by

$$D_1=\sum_{m_2=0}^{M_2}\sum_{m_1=0}^{M_1}m_1P_{m_1m_2} \qquad \textbf{(A3)}$$

and

$$P_{blocking1}=\sum_{m_2=0}^{M_2}P_{M_1m_2} \qquad \textbf{(A4)}$$

The delay and probability of blocking for the normal AU are represented respectively by

$$D_2 = \sum_{m_2=0}^{M_2} \sum_{m_1=0}^{M_1} m_2 P_{m_1 m_2} \tag{A5}$$

and

$$P_{blocking2} = \sum_{m_1=0}^{M_1} P_{m_1 M_2} \tag{A6}$$

The same derivation can be performed for a network of queues with three dimensions or more.

APPENDIX 4B - THE WEIBULL MODEL

The Weibull model is used to represent most situations of the hazard curve z(t).

When $z(t) = Kt^{M}$ for $M > -1$,

the associated density and reliability functions are

$$f(t)=Kt^{M}e^{-Kt^{M+1}}/(M+1) \tag{B1}$$

and

$$R(t)=e^{-Kt^{M+1}}/(M+1) \tag{B2}$$

By a choice of the two parameters K and M, a wide range of hazard curves can be approximated. For M=0 we obtain a constant hazard model. For M=1 we obtain a linearly increasing hazard model.

CHAPTER 5

SHUFFLENET REQUIREMENTS: ROUTING TECHNIQUES FOR LIGHTWAVE SHUFFLENET UNDER FAULTS AND UNBALANCED LOADS

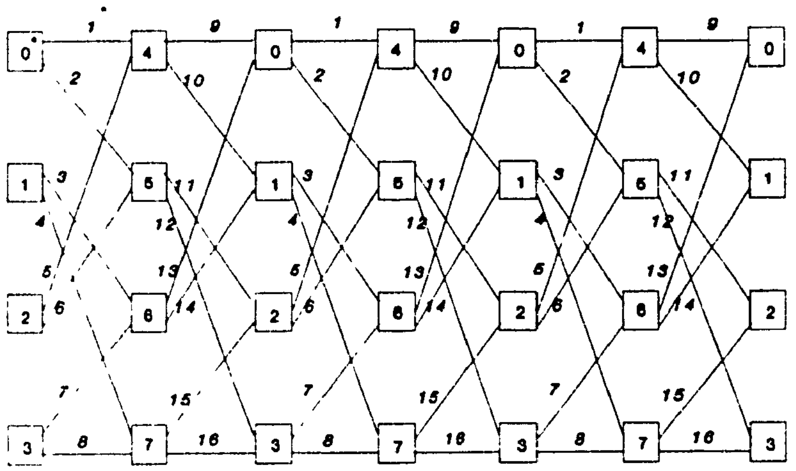
5.1 Introduction

Rapidly tunable wavelength-agile optical transmitters and receivers have not yet emerged from research laboratories. In wave division multiplexing (WDM) networks each user is assigned on fixed or momentary basis one or more transmission and reception frequencies thus eliminating the frequency coordination problem [BANI1]. However, routing of packets through intermediate users each repeating the packet en route to destination on a new wavelength is necessary, i.e. multiple hops and multiple buffering (at least for demodulation and remodulation purposes) may be necessary [HLUC1].

Our main objective in this research is to device and evaluate efficient routing techniques for the ShuffleNet which are robust even when under faults and nonuniform load. The routing techniques that are taken into consideration are those which on top of minimizing the transmission delay will also attempt to balance utilization of the wavelengths [KADO3]. The average queueing size as well as the buffer overflow for each of these techniques are considered and compared.

5.2 Network Description

To construct the ShuffleNet connectivity graph we arrange $N=kp^k$ (for k=1,2,... and p=1,2,...) users in k columns of p^k users each (figure 5.1). Moving from left to right successive columns are connected by p^{k+1} directed arcs arranged in a fixed shuffle pattern, with the last column connected to the first as if the entire graph was wrapped around a cylinder. Each of the p^k users in a column has p arcs directed to p different users in the next column. Numbering the users in a column from 0 to p^{k-1}, user i has arcs directed to user j, j+1, up to j+p-1 in the next column, where $j=(i \bmod p^{k-1}).p$. The resulting



• wavelength number; • node number.

Figure 5.1. ShuffleNet with P=2, k=2, and 5 hops

patterns of arcs between adjacent columns is referred to as a p-Shuffle by Patel [PATE1], being a generalization of the (p=2) perfect shuffle. Figure 5.1 shows the connectivity pattern for eight users (p=2, k=2) (the subject network for the routing techniques in this paper). There are a total of kp^{k+1} arcs in a ShuffleNet graph: p arcs outgoing from and p arcs incoming to each user. In other words there are kp^{k+1} optical channels and each user requires p optical transmitters and p optical receivers.

Under no faults, uniform flow matrix (between nodes) the expected number of hops between the randomly selected users is given by [ACAM1]

$$E(NumberOfHops) = \frac{kp^{k}(p-1)(3k-1)-2k(p^{k}-1)}{2(p-1)(kp^{k}-1)} \qquad \textbf{(5.1)}$$

with the maximum number of hops = 2k-1 under no fault condition.

For a routing strategy that puts a uniform traffic load on all channels, the channel efficiency η of the ShuffleNet is given by [EISE1] :

$$\eta = \frac{1}{E(NumberOfHops)} = \frac{2(p-1)(kp^{k}-1)}{kp^{k}(p-1)(3k-1)-2k(p^{k}-1)} \qquad \textbf{(5.2)}$$

for large p equation 5.2 becomes

$$\eta=\frac{2}{3k-1} \tag{5.3}$$

5.3 The Optimum Routing Policy

The optimum routing policy tries to equalize the traffic carried on each wavelength (to avoid possible bottlenecks), (i.e.) it minimizes the sum of squares of the differences between the traffic carried by each wavelength by routing each traffic to less congested nodes. This necessitates the presence of a control processor (one of the nodes) which has the information regarding the traffic conditions including fault occurrences if any.

The routing (or control) parameter is β_{ijk} (i.e.) the probability of selecting route k for the traffic from node i to node j. The central controller will find the optimal values of β_{ijk} for all i,j, and k and relays their values to the appropriate node (possibly through inband signalling) which will in turn use random number generators to forward the packet through path (k) with the probability β_{ijk} given by the controller.

The wavelength differential utilization I, reflects the overall use of all the wavelengths in the ShuffleNet network. A low value of I indicates that the traffic policy used to route the calls is making a better utilization of all the wavelengths availakle. The value I reflecting the differential utilization of all the wavelengths in the network is given by

$$I=\sum_{\ell_1=1}^{a}\sum_{\ell_2=1}^{a}\left[\left(\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{R_{ij}}\beta_{ijk}P_{ij}\gamma_{ijk\ell_2}\right)-\left(\sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{R_{ij}}\beta_{ijk}P_{ij}\gamma_{ijk\ell_1}\right)\right]^2 \quad (5.4)$$

for $\ell_1 \neq \ell_2$ and $i \neq j$.

ℓ_1 and ℓ_2 represent the wavelength numbers, a is the total number of wavelengths, n is the total number of nodes, P_{ij} is the traffic flow probability between node i and node j, k is the alternate route number and R_{ij} is the total number of alternate routes for the traffic from node i to node j. $\gamma_{ijk\ell}$ is an indicator that is set to 1 if the wavelength ℓ is used in route k for the traffic going from source i to destination j; otherwise it is set to zero.

Before attempting to optimize I, it is desirable for computational convenience to reduce the dimensionality of the parameters of I by mapping the two indices i and j to one variable α. Reducing the parameters of I by one dimension we obtain $\hat{I}$

$$\hat{I}=\sum_{\ell_1=1}^{a}\sum_{\ell_2=1}^{a}\left[\left(\sum_{\alpha=1}^{n(n+1)}\sum_{k=1}^{R_\alpha}\beta_{\alpha k},\hat{P}_\alpha,\hat{\gamma}_{\alpha k\ell_2},\right)-\left(\sum_{\alpha=1}^{n(n+1)}\sum_{k=1}^{R_\alpha}\beta_{\alpha k},\hat{P}_\alpha,\hat{\gamma}_{\alpha k\ell_1},\right)\right]^2 \quad (5.5)$$

for $\ell_1 \neq \ell_2$ and $\beta_{\alpha k}=\beta_{ijk},\hat{P}_\alpha=P_{ij},\hat{\gamma}_{\alpha k\ell}=\gamma_{ijk\ell}$

For uniformity and computational convenience we express $\hat{P}_\alpha$ as a two dimensional variable where the second dimension is now the route k (i.e.) we replace in equation 5.5 $\hat{P}_\alpha$ with $\hat{P}_{\alpha k}$ so equation 5.5 becomes

$$\hat{I}=\sum_{\ell_1=1}^{a}\sum_{\ell_2=1}^{a}\left[\left(\sum_{\alpha=1}^{n(n+1)}\sum_{k=1}^{R_\alpha}\beta_{\alpha k},\hat{P}_{\alpha k},\hat{\gamma}_{\alpha k\ell_2},\right)-\left(\sum_{\alpha=1}^{n(n+1)}\sum_{k=1}^{R_\alpha}\beta_{\alpha k},\hat{P}_{\alpha k},\hat{\gamma}_{\alpha k\ell_1},\right)\right]^2 \tag{5.6}$$

for $\ell_1 \neq \ell_2$

Finally, parameters of $\hat{I}$ are further reduced by one dimension

$$\overline{I}=\sum_{\ell_1=1}^{a}\sum_{\ell_2=1}^{a}\left[\left(\sum_{\xi=1}^{\eta}\overline{\beta}_\xi\overline{P}_\xi\overline{\gamma}_{\xi\ell_2}\right)-\left(\sum_{\xi=1}^{\eta}\overline{\beta}_\xi\overline{P}_\xi\overline{\gamma}_{\xi\ell_1}\right)\right]^2 \tag{5.7}$$

for $\ell_1 \neq \ell_2$

where

$$\eta=\sum_{\nabla=0}^{n(n+1)-1} M(\nabla) \tag{5.8}$$

M(∇) is the number of alternate routes for source destination ∇.

Note that M(0)=0.

It is now easy to optimize $\overline{I}$ with respect to $\overline{\beta}_x$ where x is any of the possible alternate routes in the network.

$$\frac{\partial\overline{I}}{\partial\overline{\beta}_x}=\sum_{\ell_1=1}^{a}\sum_{\ell_2=1}^{a}2\cdot\left[\left(\sum_{\xi=1}^{\eta}\overline{\beta}_\xi\overline{P}_\xi\overline{\gamma}_{\xi\ell_2}\right)-\left(\sum_{\xi=1}^{\eta}\overline{\beta}_\xi\overline{P}_\xi\overline{\gamma}_{\xi\ell_1}\right)\right]\cdot\left(\overline{P}_x\overline{\gamma}_{x\ell_2}-\overline{P}_x\overline{\gamma}_{x\ell_1}\right)=0 \tag{5.9}$$

for $\ell_1 \neq \ell_2$

This optimization is subject to the following constraints

$$\sum_{k=1}^{M(\alpha)}\beta_{\alpha k}=1 \tag{5.10}$$

for each source destination α, or

$$\sum_{k=1}^{M(\alpha)} \overline{\beta}_z=1 \qquad (5.11)$$

where

$$z=\sum_{\nabla=1}^{\alpha-1} M(\nabla-1)+k \qquad (5.12)$$

in other words the sum of the probabilities of all possible routes for any source destination pair is equal to one. Furthermore

$$\overline{\beta}_x\geq 0 \qquad (5.13)$$

Note that the first constraint implies that $\overline{\beta}_x\leq 1$.

In order to compute all the optimized values of $\overline{\beta}$'s, the normalized gradients are found

$$\frac{\nabla\overline{I}^p}{\|\nabla\overline{I}^p\|}=\left[\frac{\partial\overline{I}^p}{\partial\overline{\beta}_j}\right]\cdot\left[\sum_{j=1}^{n}\left(\frac{\partial\overline{I}^p}{\partial\overline{\beta}_j}\right)^2\right]^{-1/2} \qquad (5.14)$$

where p is the iteration number.

The algorithm used to determine the optimum values of $\overline{\beta}$'s is a constrained gradient technique [WISM1] and uses a method of multiple gradient summation defined by the algorithm

$$\overline{\beta}_j^{p+1}=\overline{\beta}_j^{p}+k_p d^p \qquad (5.15)$$

where

$$d^p=\frac{\nabla f(\overline{\beta}^p)}{|\nabla f(\overline{\beta}^p)|}=\frac{\nabla\overline{I}^p}{\|\nabla\overline{I}^p\|} \qquad (5.16)$$

if $\overline{\beta}^p$ is inside the feasible region.

However if $\overline{\beta}_j^p$ is outside the feasible region then d^p is computed as follows

$$d^p = \frac{\nabla \overline{I}^p}{\|\nabla \overline{I}^p\|} + \sum_{i=1}^{\eta} \frac{\nabla g_i(\overline{\beta}^p)}{\|\nabla g_i(\overline{\beta}^p)\|} \tag{5.17}$$

The feasible region ir described by the vector $g_i(\overline{\beta}) \leq 0$ for $i=1,2,\ldots,\eta$ and $k_p < 0$. The value of ∇g_i is used to adjust the β being computed. ∇g_i can be decomposed into an equation comprising a scalar δ multiplied to a vector with all of its terms equal to zero except for the i term which has the value of one. The polarity of δ depends on whether the sum of βs is less than one (positive polarity) or greater than one (negative polarity). The term d^p in equation 5.17 can thus be represented as

$$d^p = \frac{\nabla \overline{I}^p}{|\nabla \overline{I}^p|} + \delta_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ . \\ . \end{bmatrix} + \delta_2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ . \\ . \end{bmatrix} + \ldots + \delta_\eta \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 1 \end{bmatrix} \tag{5.18}$$

for $\delta \leq 1$.

The βs are computed iteratively until the stopping criteria is satisfied. (i.e.)

$$\sum_{j=1}^{\eta} (\overline{\beta}_j^{p+1} - \overline{\beta}_j^p)^2 < \epsilon \tag{5.19}$$

where ϵ is a small number.

Once the optimum βs are found, the mean transmission delay in average number of hops is computed as follows :

$$E=\sum_{\xi=1}^{n} \frac{\overline{\beta}_{\xi}\overline{P}_{\xi}H_{\xi}}{n(n+1)} \tag{5.20}$$

where H_{ξ} is the number of hops spanned by route ξ and n(n+1) is the total number of source-destination. This represents the transmission delay for ShuffleNet expressed in terms of number of hops times the traffic flow to the destination node.

5.4 The Random Routing Policy

The random routing policy is used to define an initial guess for the values of the parameter β_{ξ}. To each source destination route there are a number of alternate paths. The alternate path with the least number of hops is selected with probability χ, a design parameter; the other r alternate paths in the same route have each equal probability $(1 - \chi)/r$.

The gradient algorithm for the optimum policy uses these values of βs as initial values.

5.5 The Greedy Routing Policy

The greedy routing policy is a specific case of the random policy. Under this policy, only one path for any given source destination route is selected. The path selected is always the one with the least number of hops. If for any reason the selected path is not possible (failure of a link) than the next path with the least number of hops is selected.

The βs of the selected paths are equal to one and the other βs are zero.

Each node keeps a table of its shortest distance (minimum number of hops) to all destinations. This table has fixed values which are changed only in the case of link failures.

5.6 Buffering Delay and Reliability under Faults

The object of this section is to compute the average traffic on each link (i.e. on each wavelength) under the three routing policies outlined and under certain fault conditions, after which the buffering delay due to demodulation and remodulation and/or faults at a different wavelength is found. This buffering is at least necessary for the purpose of processing the information; it is also useful for temporary store and forward to control congestion. The buffering may also be needed to withstand temporary and/or permanent failures in one or more links (wavelengths) especially if the routing strategy does not appropriately cope with the fault situation. Also using different wavelength on the fibres means different signal dispersions and/or possible changes in index of refraction with frequencies (wavelengths) and different bit errors from demodulation. Each wave may give rise to one or many different modes on the fibre channel. Error detection and or correction codes might be necessary. Packetizing and ARQ might be used. In all, certain amount of buffering might be needed because of bit error and correction discrepancies at different wavelengths.

In the remaining part of this chapter, we pursue a combined analysis-simulation routine to compute the effects of the three routing strategies on buffer requirements under faults. The case under consideration is the permanent single link (wavelength) fault. We assume an M/M/1/K finite buffer type at the receiver of each node and we associate it with a certain optical wavelength. The differential wavelength utilization I, the expected transmission delay and queue size and buffer overflow probabilities are computed for each link (wavelength) fault and for each routing strategy.

Based on the simulation of the object ShuffleNet, and using the routing vector $\overline{\beta}$, the average traffic loaded on each wavelength are computed as:

$$m_1=\sum_{\xi=1}^{\eta}\frac{\overline{\beta}_\xi \overline{P}_\xi \overline{\gamma}_\xi f_1}{\eta}=\rho_1 \tag{5.21}$$

$$m_2=\sum_{\xi=1}^{\eta}\frac{\overline{\beta}_\xi \overline{P}_\xi \overline{\gamma}_\xi f_2}{\eta}=\rho_2 \tag{5.22}$$

up to the last wavelength y

$$m_y=\sum_{\xi=1}^{\eta}\frac{\overline{\beta}_\xi \overline{P}_\xi \overline{\gamma}_\xi f_y}{\eta}=\rho_y \tag{5.23}$$

where f_1 = 0 if wavelength 1 fails and equals to one otherwise.

The mean wavelength utilization (averaged over all wavelengths) is given by

$$\overline{m}=\frac{\rho_1+\rho_2+\ldots\ldots+\rho_y}{y} \tag{5.24}$$

Another measure for expressing the different usage of the various wavelengths is the variance which is identified as

$$\sigma^2=\frac{(\rho_1-\overline{m})^2+(\rho_2-\overline{m})^2\ldots+(\rho_y-\overline{m})^2}{y} \tag{5.25}$$

5.7 Average Queue Size and Buffer Overflow

In ShuffleNet, each node receives traffic on one wavelength, detects the frame, stores it into a temporary buffer and then sends it through one of its transmission wavelengths en route to its destination. Buffering of incoming traffic as well as buffer overflow are indications of the effectiveness of the system.

A measure which is used is the mean queue size for an M/M/1/K Markovian queueing model with one server where K is the maximum number of users allowed at one time in the system. When there are K frames in the system, the new arriving frames are blocked. The basic formula for the average number in the queue is [MURD1]:

$$\overline{Q}=\rho^2\left[\frac{1-K\rho^{K-1}+(K-1)\rho^K}{(1-\rho)(1-\rho^{K+1})}\right] \tag{5.26}$$

The buffer overflow or the proportion of frames not served is:

$$P_K=\frac{\rho^K(1-\rho)}{(1-\rho^{K+1})} \qquad \textbf{(5.27)}$$

For each of the three routing policies, the mean queue sizes and buffer overflows for every station are computed under no fault as well as under single wavelength fault conditions.

5.8 Traffic Flow Distribution and Traffic Intensity

The traffic flow from one source to a destination and the traffic intensity which is the total traffic flow from one source to all destinations are essential elements in the analysis of network congestion.

The network model is analyzed with a balanced traffic intensity. In other words the flow from a source to each destination is equal and uniform. The wavelength differential utilization, the transmission delay, queue size and buffer overflow for the three routing policies are computed and compared. The computation is performed for a network under balanced traffic intensity with no fault as well as with a single wavelength fault.

The analysis is repeated but with an unbalanced traffic flow matrix. The traffic flow from a source node to some particular node is higher than from the same source node to other nodes (the preferred node case). The traffic flow from any source node to the high traffic node is 80% of the traffic intensity; the remaining 20% is split evenly for the flows from the same source node to other destination nodes. Furthermore, in the analysis for a single wavelength failure, the wavelength that fails is always taken as one of the wavelengths that feed the high traffic node. This makes the congestion analysis more critical. These scenarios have been computed for each of the three routing policies for different values of traffic intensities and the results have been compared.

5.9 Results and Conclusion

The wavelength differential utilizations for the random policy $\mathbf{I}_i$, for the greedy policy $\mathbf{I}_g$, and for the optimal policy $\mathbf{I}_0$ are computed in the case of no link failure, as well as in the case of a single link failure for conditions of uniform and non uniform traffic flow. Results are plotted against the traffic intensity (figure 4.2). Since the random policy and associated optimal policy are base on routing probabilities, a condition has been investigated namely when β_1 which is the route with the least number of hops is selected 50% of the time. Figure 5.2 show that the optimal policy and the random policy give best results in the case of wavelength differential utilization. It is seen in figure 5.2b and d that the wavelength differential utilization for the non uniform traffic flow shows a tremendous increase over the uniform traffic flow in figure 5.2 a and c for the same cases. The increase is not as dramatic when the graphs for no fault network is compared to the single wavelength fault. The non uniform traffic flow makes as expected the least efficient utilization of the wavelength across the network.

For the wavelength differential utilization, the greedy policy under uniform traffic flow experiences a small degradation from the no fault situation (figure 5.2a) which shows a value of 15900 at traffic intensity of 0.91 compared to the single wavelength fault situation (figure 5.2c) which shows a value of 19400 at traffic intensity of 0.91. The random policy and the optimal policy, although better than the greedy policy show a more significant change from the no fault situation (figure 5.2a) which has values of 6144 and 6445 respectively at traffic intensity of 0.91 compared to the single wavelength situation (figure 5.2c) which show values of 16631 and 16342 respectively at traffic intensity of 0.91. It is to be noted that although the optimal policy is better than the random policy the difference is insignificant. Under non uniform traffic flow the three policies have the same behaviour pattern for the wavelength differential utilization but at a much higher scale. For example, under no fault (figure 5.2b) the greedy policy shows a value of 111300 at 0.91 traffic intensity whereas the random and optimal policies show values of 60380 and 59300 at 0.91 traffic intensity. Under fault (figure 5.2d) these values become 151200 79100 and 78300 respectively for the greedy, random and optimal policies at 0.91 traffic intensity.

Transmission delay computation of the same three policies under the same conditions are shown in figure 5.3. Delay is expressed in terms of number of hops times traffic flow per route which we will denote as HFR (hops traffic flow per route). For the uniform traffic flow, the greedy policy has a transmission delay of 0.26 HFR when traffic intensity is 0.91 for the no fault situation (figure 5.3a) and 0.28 HFR when traffic intensity is 0.91 for the single wavelength fault situation (figure 5.3c). This is a small degradation in transmission which is insignificant. The random and optimal policies have respectively transmission delays of 0.41 HFR and 0.40 HFR at 0.91 traffic intensity for the no fault situation (figure 5.3a) and 0.36 HFR and 0.37 HFR at 0.91 traffic intensity for the single wavelength fault situation (figure 5.3c). The random and optimal policies have consistently close values. The very small apparent improvement in the average delay from the no fault to the single wavelength fault can be explained by the fact that the averages are used and the additional diverted load on the other wavelengths when a fault in one of the wavelength occurs is very small; furthermore, since the fault is on the wavelength that carries heavy traffic, the diverted traffic is smoothed out among other wavelengths. For the non uniform traffic flow, the greedy policy has a transmission delay of 0.22 HFR at 0.91 traffic intensity for the no fault situation and 0.30 HFR at 0.91 traffic intensity for the single wavelength fault situation. The random and optimal policies have respectively transmission delays of 0.35 HFR and 0.34 HFR at 0.91 traffic intensity for the no fault situation and 0.28 HFR for both at 0.91 traffic intensity in the single wavelength fault situation. The same behaviour as in the uniform traffic flow is seen for the random and optimal policies. However we see that these two later policies have a slightly better delay than the greedy policy for single wavelength fault. This is explained by the fact that for an average single wavelength fault under a non uniform situation, the random and optimal policies balance the traffic among the wavelengths in the network whereas the greedy policy always uses only one route and therefore concentrate the traffic on some wavelengths.

The mean wavelength utilization in figure 5.4 which is derived from equation 5.24 is also equal to the average queue size as depicted in equation 5.26 when the buffer size is one. The three policies in this situation behave in the same fashion as in the transmission delay situation in figure 5.3. The average queue size for buffer size of one is 0.0084 packets for the random as well as for the optimal policy at

0.91 traffic intensity for uniform traffic and in the case of no fault (figure 5.4a). The greedy policy in the same circumstances is 0.0054 packets (figure 5.4a). Again the only case when the average queue size is higher for the greedy policy (0.0063 packets) than for the random or optimal policy (0.0058 packets) is when the traffic flow is non uniform and in the case of the average single wavelength fault (figure 5.4d).

The variance in figure 5.5 as well as the wavelength differential utilization in figure 5.2 are measures of the frequency of use of wavelength or in other words the usage concentration on certain wavelength which may result in bottlenecks which again makes it a measure of reliability. Figure 5.5 shows that the variances for all policies are very low. It is therefore a comparative analysis between the policies that is in effect here. This analysis is actually the one that is made under the wavelength differential utilisation which would apply here replacing of course values relevant to variances.

The analysis for the average queue size was done for many values of buffer sizes. It was found that for buffer size higher than 2 the difference in the results with those obtained with buffer size equal to two are extremely small. However, the average buffer overflow, although very small for buffer size of 2 (order of 10^{-6}) gets practically null for buffer size of 3 (order of 10^{-9}) and 4 (order of 10^{-12}). The analysis for the average queue size and average buffer overflow is thus a comparative one among the three policies. Figure 5.7 and 5.9 compare the random policy with the optimal policy. Figure 5.7 is for the average queue size and figure 5.9 is for the average buffer overflow. For all practical purposes the two policies generate equal results. A comparative analysis between the random policy and the greedy policy for buffer size of two is thus sufficient.

The average queue size and buffer overflow in figures 5.6 to 5.9 show the computed values under no wavelength fault, the highest computed value when a single wavelength fails as well as the lowest computed value for a failed wavelength and the average for the single wavelength failure. The spread between the highest and lowest values is an indication of how the average value represents the queue size and buffer overflow for any single wavelength failure.

For the random policy in figure 5.6, the average queue size under uniform traffic flow (figure 5.6a) with no wavelength fault is 0.71 x 10^{-6} packets for a traffic intensity of 0.91. With a single wavelength fault the highest average queue size is 0.67 x 10^{-6} packets, the lowest average queue size is 0.48 x 10^{-6} packets and the mean average queue size is

0.58 x 10^{-6} packets all for a traffic intensity of 0.91. The no fault case shows the highest average queue size. In the case of the greedy policy in figure 5.6c it is the no fault situation that shows the lowest average queue size. We see that the random and greedy policies have the same trends whether under uniform or under non uniform traffic flow.

The average buffer overflow graphs in figure 5.8 compare the random policy with the greedy policy. Average buffer overflow graphs are the same as those of the average queue size of figure 5.6 for buffer size equal t , two. For higher buffer sizes the average buffer overflow is extremely small.

We have seen that throughout the analysis, the greedy policy was better when no wavelength faults occurred; however when a single wavelength fault occurred the random and optimal policies showed better results.

Our results show the performance of the optimal policy corresponding to one initial selection of the β vector (i.e.) finally leads to a local minimum. To obtain a global minimum we may have to run in real time the gradient routine which may not be visible thus indicating the danger of applying the optimal policy in this application. Specially in case of applications where large burst of high rate data are the norm rather than the exception. On the other hand the random policy is more cost effective due to the visibility of computation and performance obtained.

ShuffleNet shows robustness to single wavelength failure (comparing the transmission delay and wavelength utilisation in the two cases of no failure and one wavelength failure). Also simple routing policies such as random and greedy policies yield excellent transmission delays, queueing buffer delays, buffer overflow delays, and wavelength utilizations. This means that centralized control and exchange of extensive routing tables required for the optimal policy are not required in these cases.

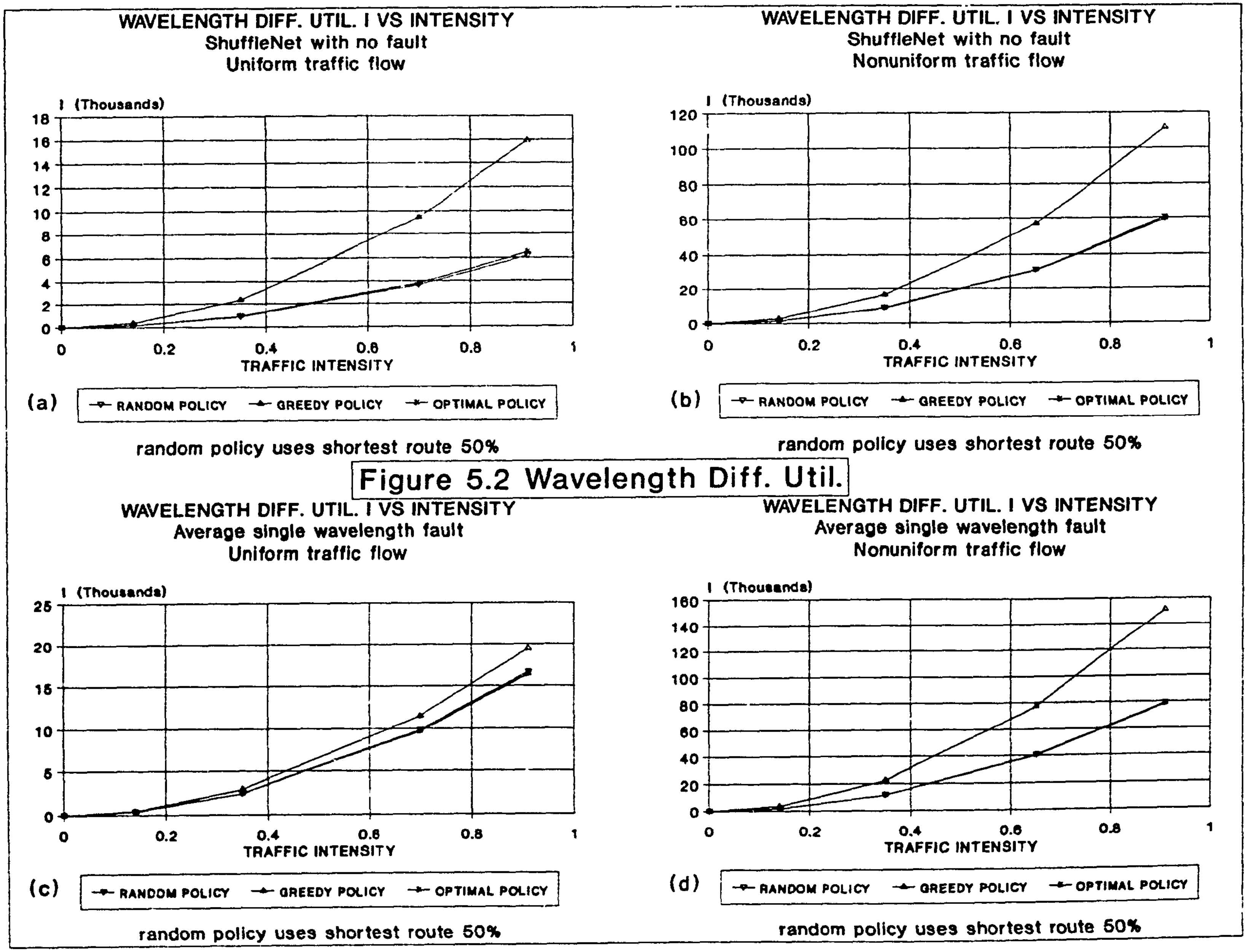
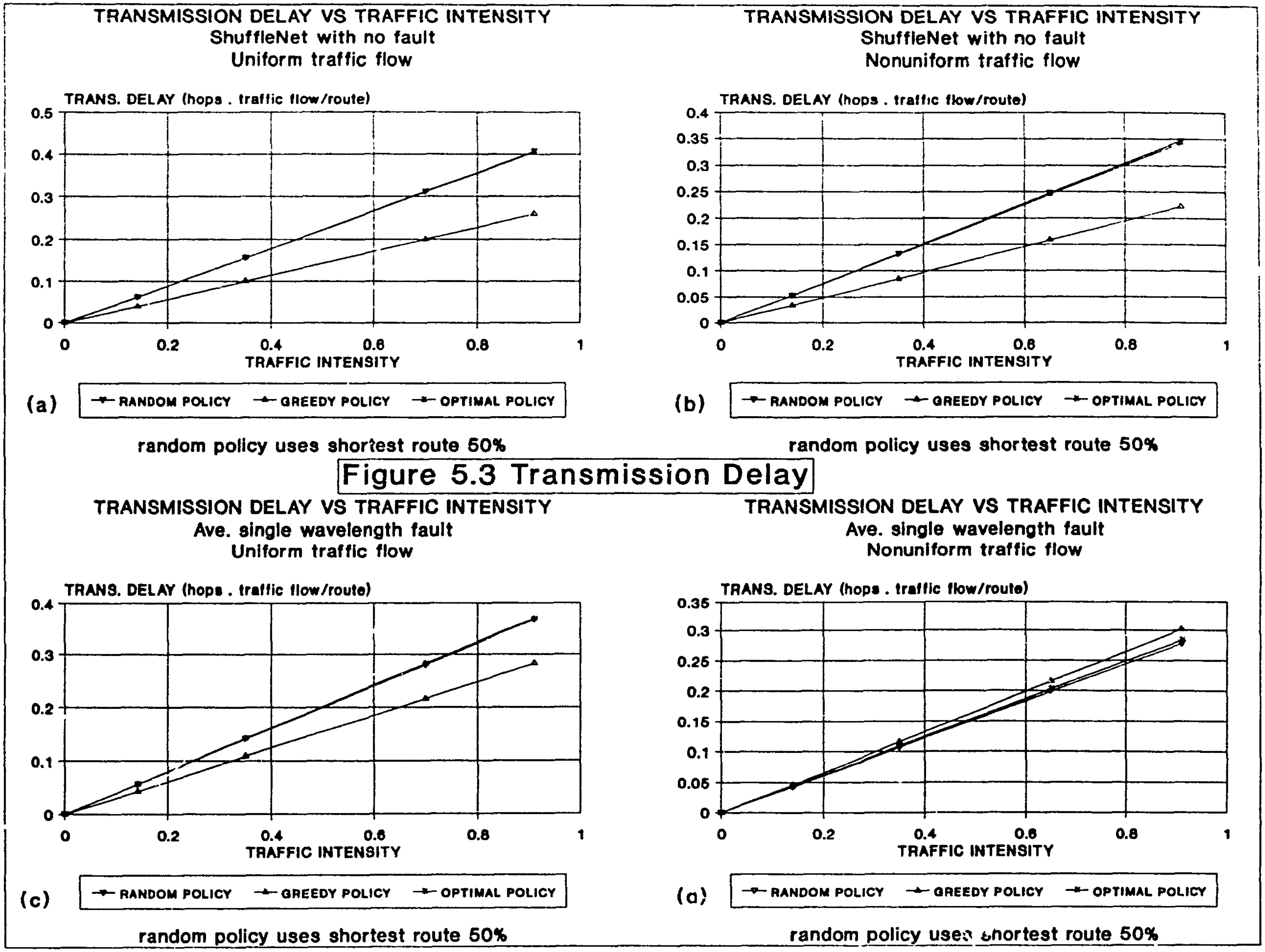


Figure 5.2 Wavelength Diff. Util.



(a) random policy uses shortest route 50%

(b) random policy uses shortest route 50%

(c) random policy uses shortest route 50%

(d) random policy uses shortest route 50%

Figure 5.3 Transmission Delay

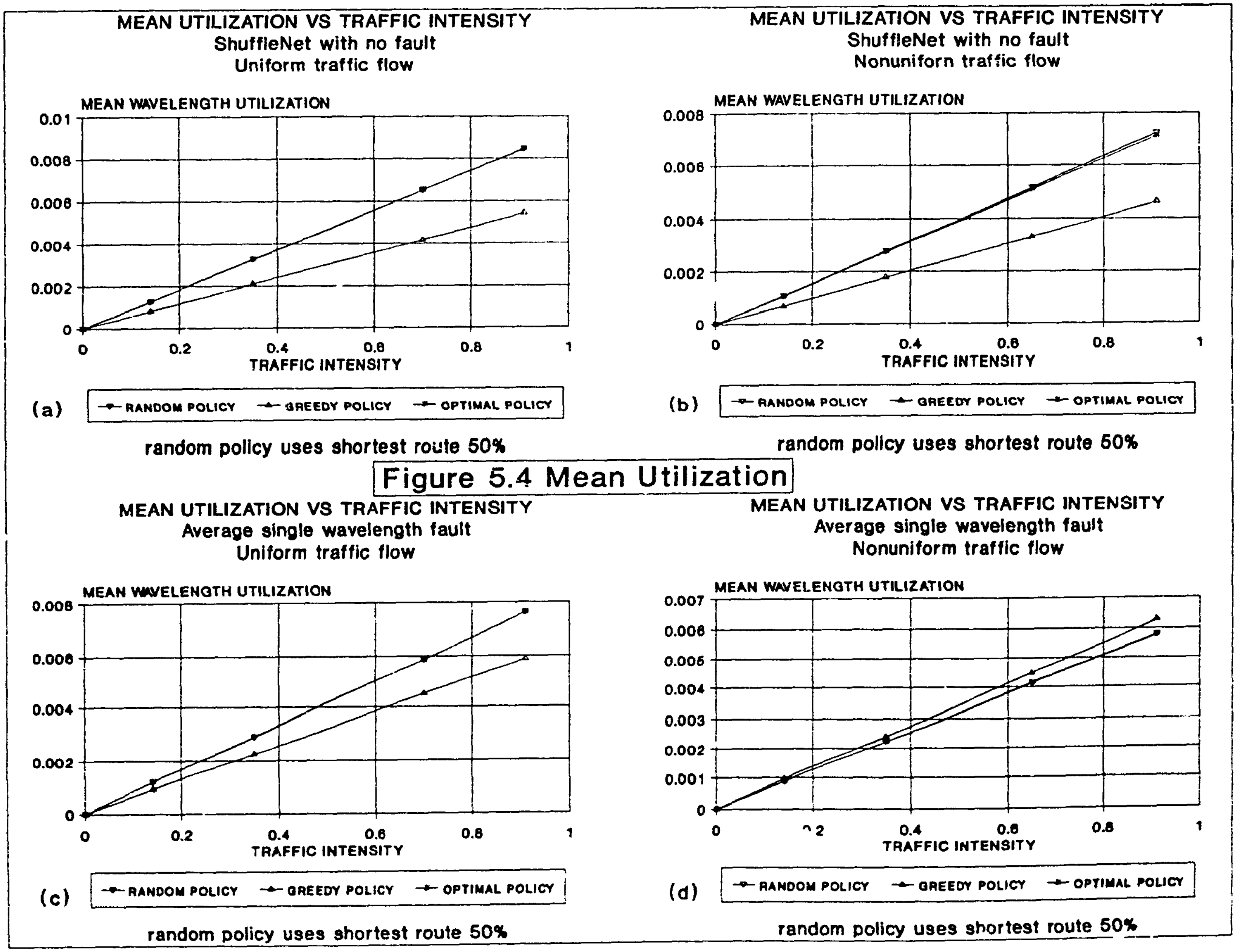
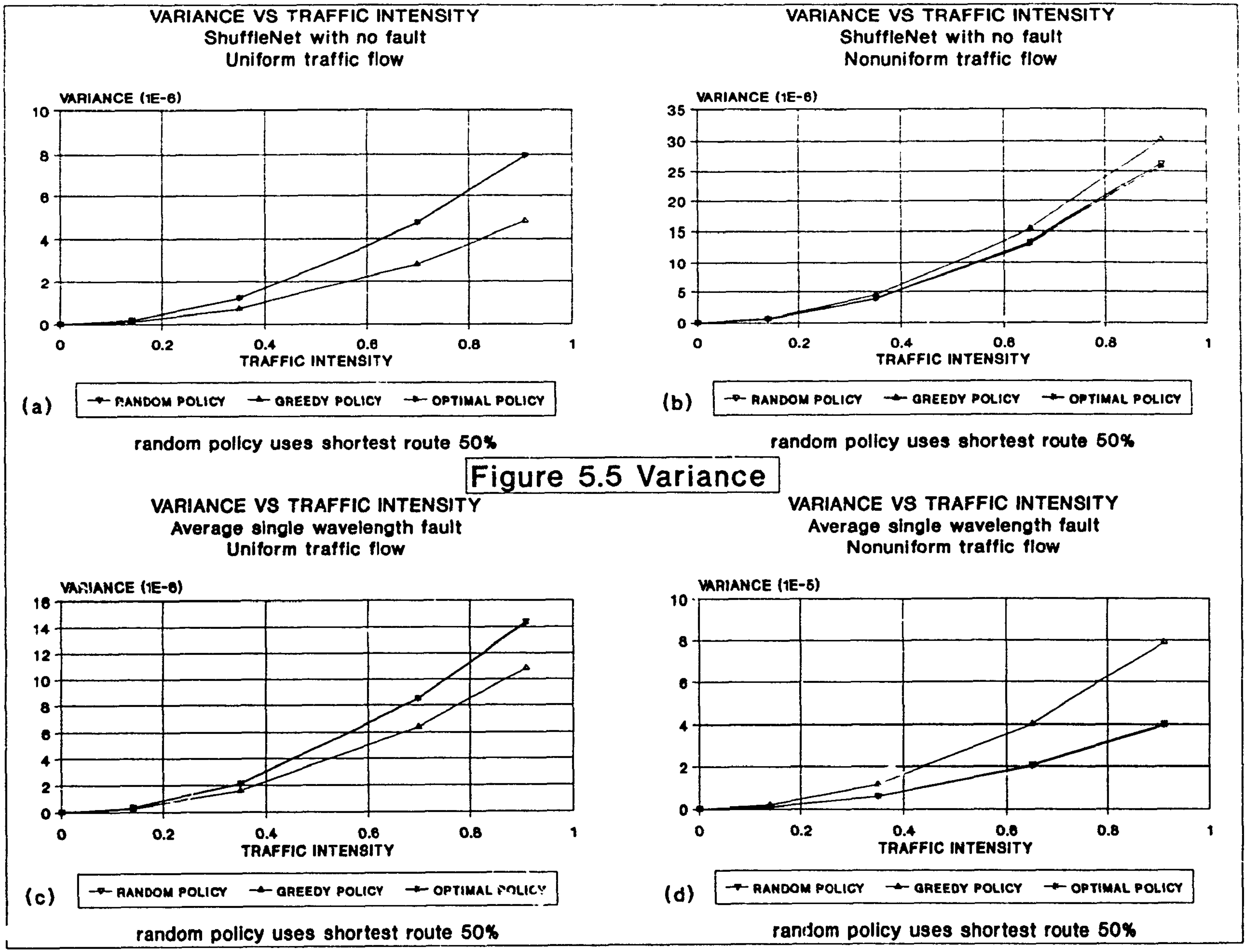


Figure 5.4 Mean Utilization



VARIANCE VS TRAFFIC INTENSITY
ShuffleNet with no fault
Uniform traffic flow

random policy uses shortest route 50%

VARIANCE VS TRAFFIC INTENSITY
ShuffleNet with no fault
Nonuniform traffic flow

random policy uses shortest route 50%

Figure 5.5 Variance

VARIANCE VS TRAFFIC INTENSITY
Average single wavelength fault
Uniform traffic flow

random policy uses shortest route 50%

VARIANCE VS TRAFFIC INTENSITY
Average single wavelength fault
Nonuniform traffic flow

random policy uses shortest route 50%

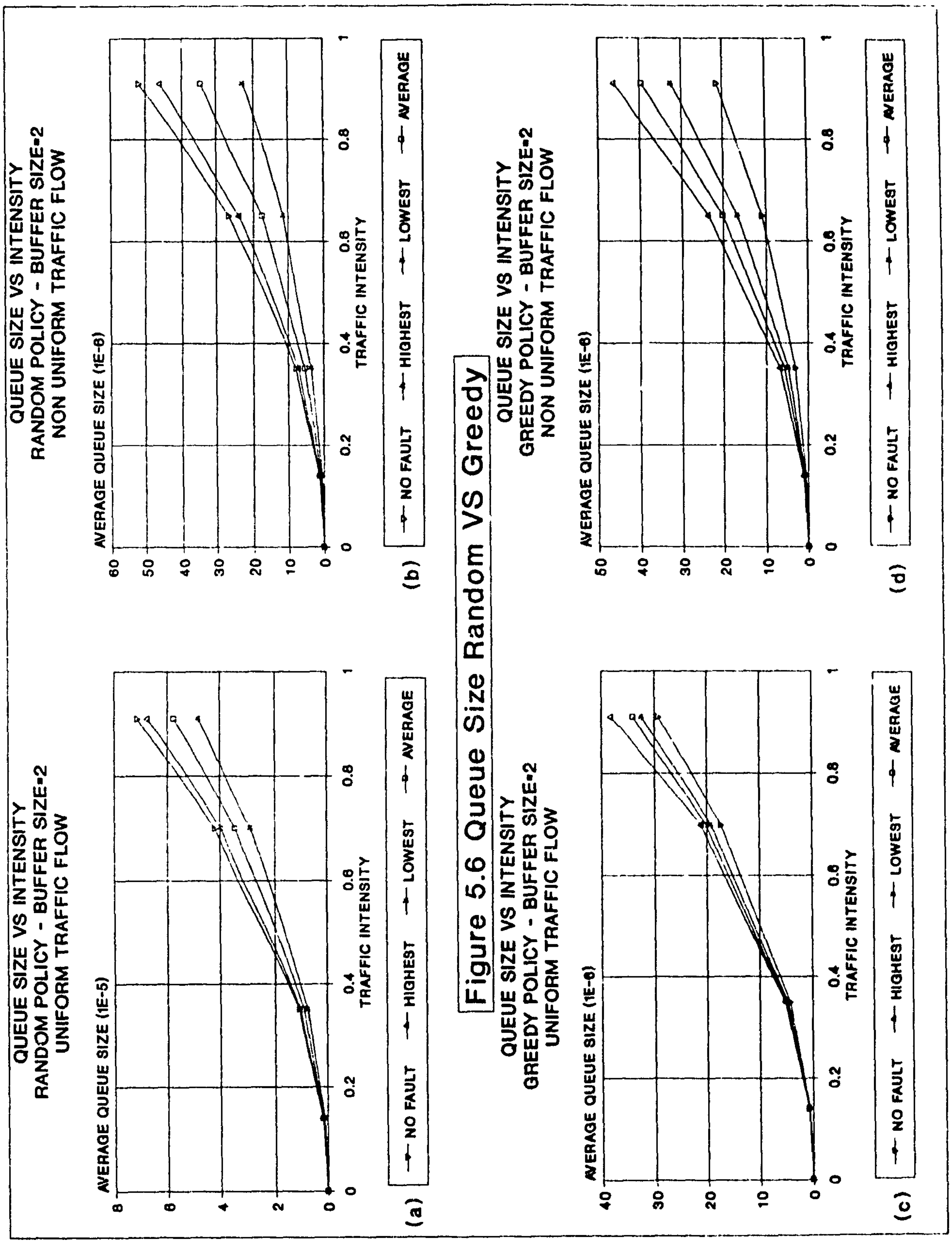


Figure 5.6 Queue Size Random VS Greedy

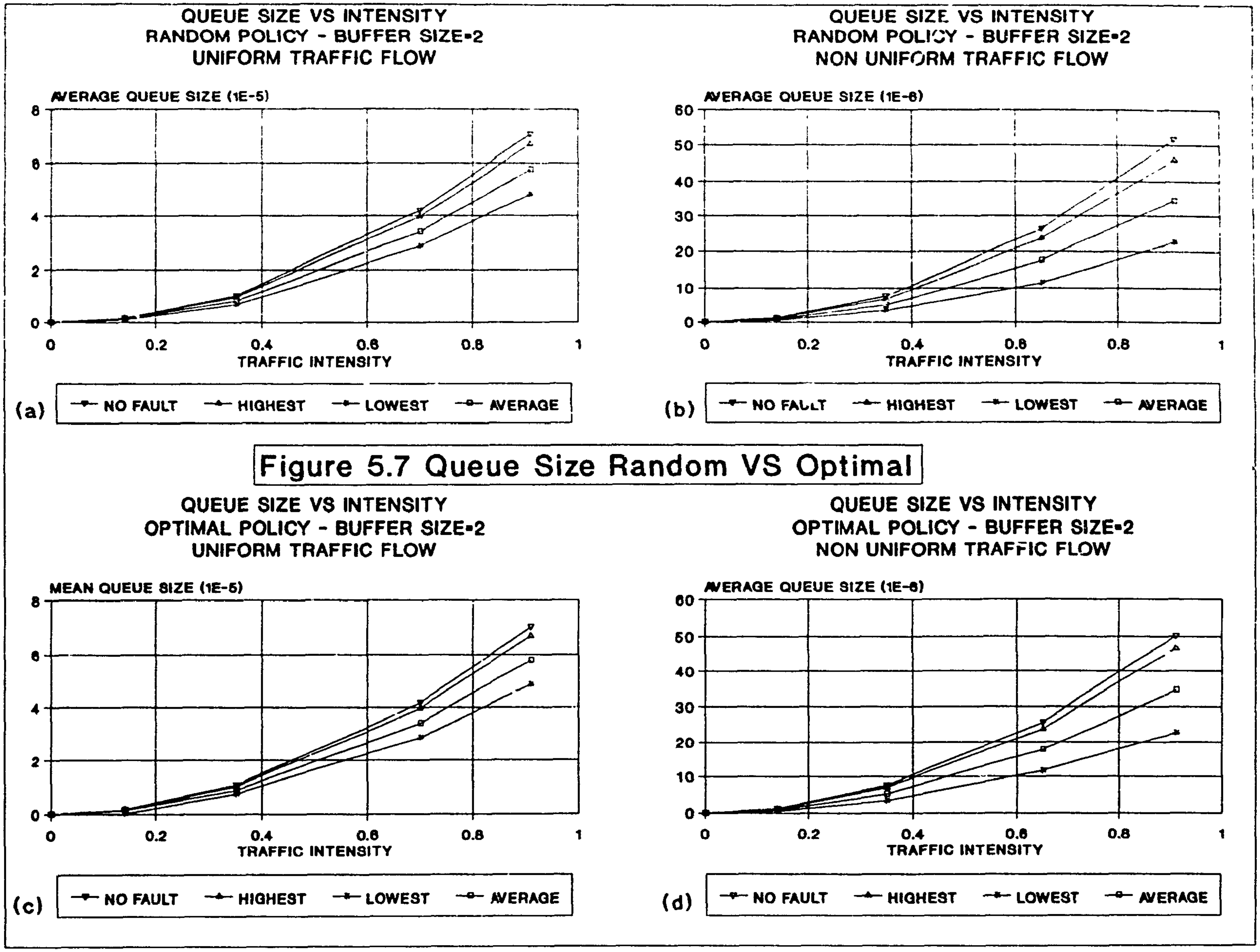


Figure 5.7 Queue Size Random VS Optimal

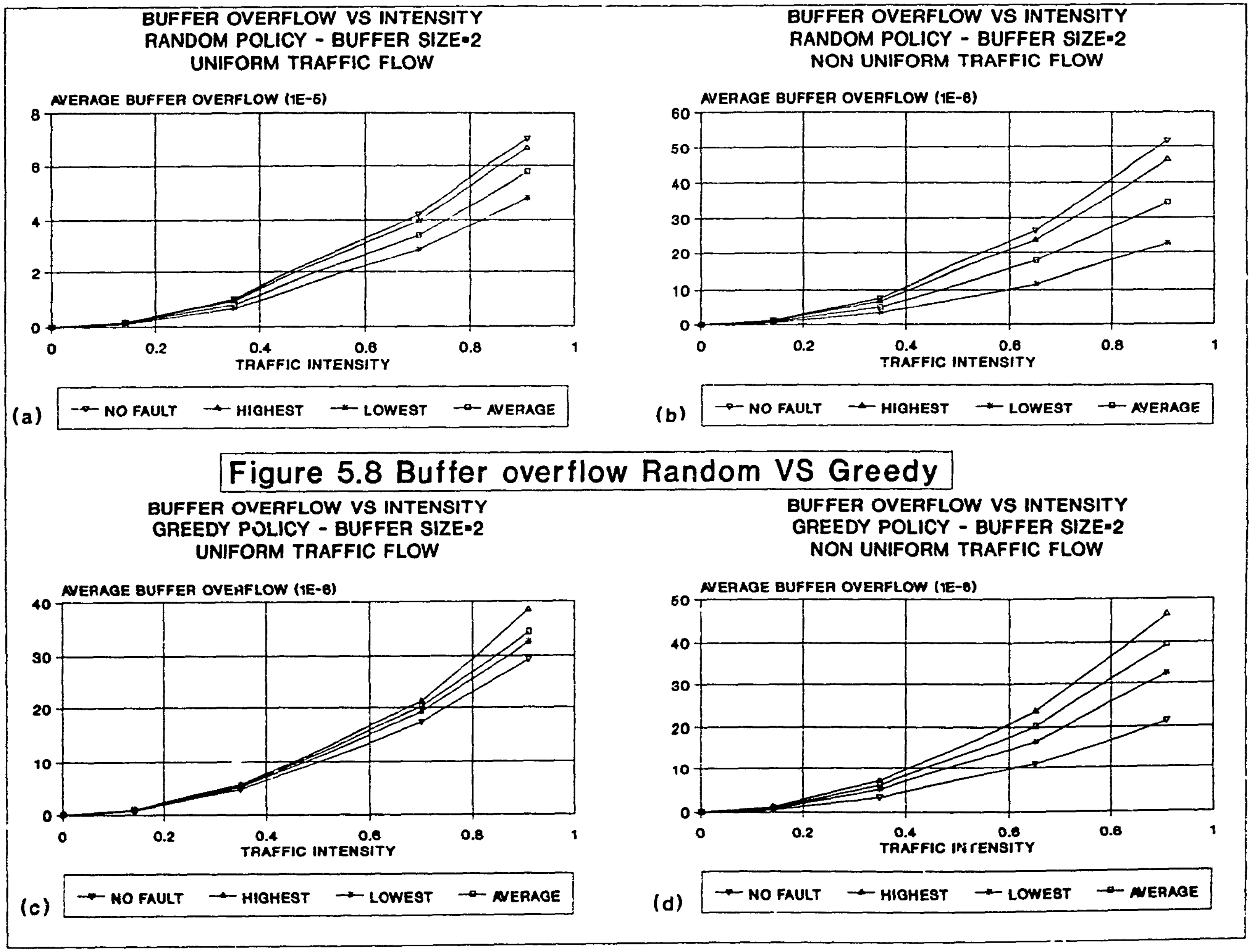
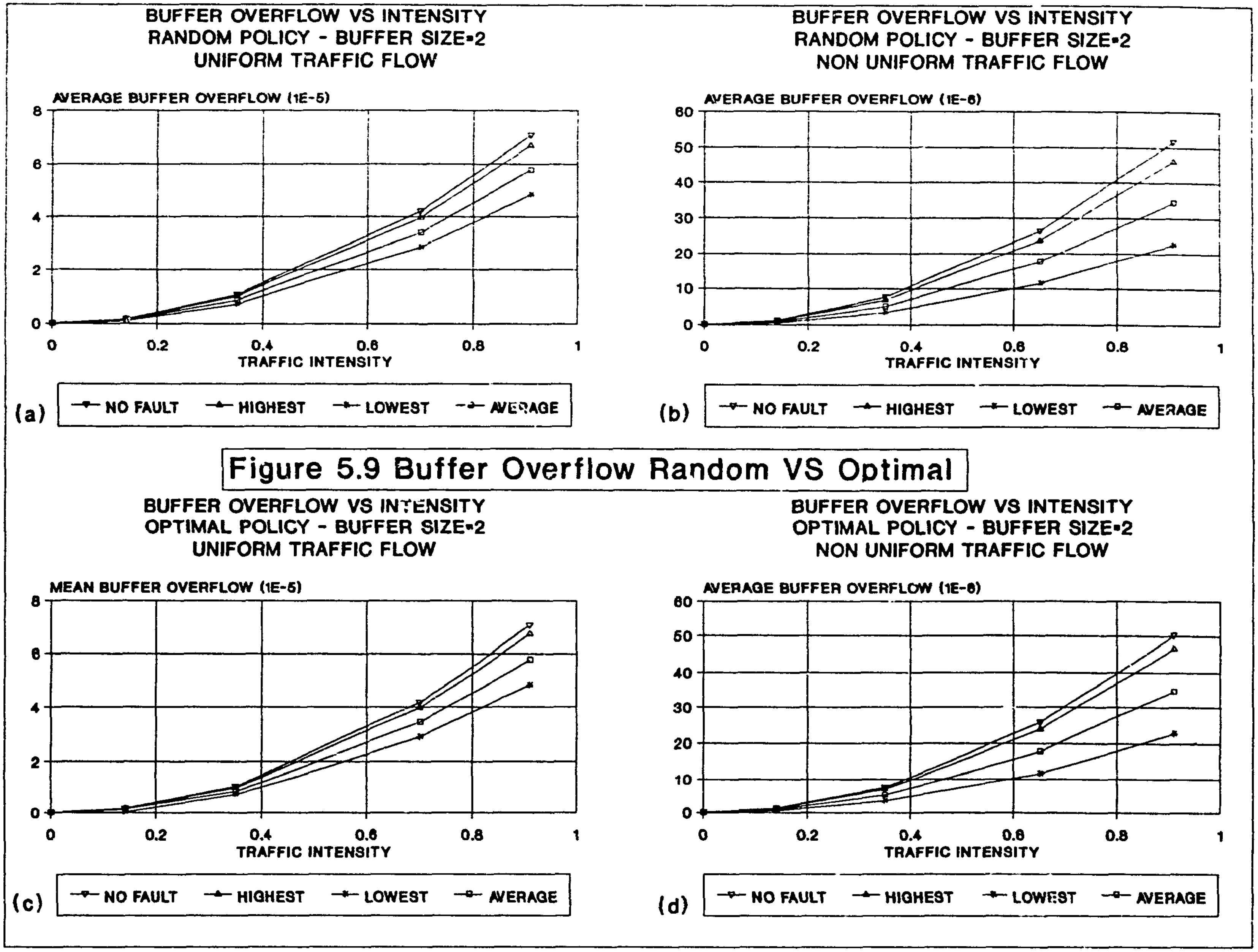


Figure 5.8 Buffer overflow Random VS Greedy



BUFFER OVERFLOW VS INTENSITY
RANDOM POLICY - BUFFER SIZE=2
UNIFORM TRAFFIC FLOW

BUFFER OVERFLOW VS INTENSITY
RANDOM POLICY - BUFFER SIZE=2
NON UNIFORM TRAFFIC FLOW

Figure 5.9 Buffer Overflow Random VS Optimal

BUFFER OVERFLOW VS INTENSITY
OPTIMAL POLICY - BUFFER SIZE=2
UNIFORM TRAFFIC FLOW

BUFFER OVERFLOW VS INTENSITY
OPTIMAL POLICY - BUFFER SIZE=2
NON UNIFORM TRAFFIC FLOW

5.10 References

[ACAM1] A. S. Acampora, M. J. Karol, and M. G. Hluchyj, " Terabit Lightwave Networks: The Multihop Approach", ATT Technical Journal, Volume 66, Issue 6, Nov/Dec 87.

[ACAM2] A. S. Acompora and M. J. Karol, "An Overview of Lightwave Packet Networks," IEEE Network Mag., Vol. 3, pp.29-41, Jan. 1989.

[ACAM3] A. S. Acompora, "A Multichannel Multihop Local Lightwave Network," GLOBECOM'87 Conf. Rec., pp. 1459-1467, Nov. 1987.

[BANI1] J. A. Bannister, M. Gerla, "Design of the Wavelength-Division Optical Network", ICC 90, pages 962-967, April 1990

[EISE1] M. Eisenberg, N. Mehravari, "Performance of the Multichannel Multihop Lightwave Network Under Nonuniform Traffic", IEEE Journal on Selected Areas in Communications, Vol. 6, No. 7, August 1988.

[HLUC1] M. G. Hluchyj, M. J. Karol, "ShuffleNet: An Application of Generalized Perfect Shuffles to Multihop Lightwave Networks", In Proceedings of IEEE INFOCOM '88, pages 4B.4.1-4B.4.12, New Orleans, Louisiana, March 1988.

[KADO3] M. Kadoch and A. K. Elhakeem, "Adaptive Routing for Lightwave ShuffleNet under Unbalanced Load," GLOBECOM'91, Phoenix, AZ, Dec. 1991.

[KARO1] M. J. Karol, S. Shaikh, "A Simple Adaptive Routing Scheme for ShuffleNet Multihop Lightwave Networks", Globecom 88, pages 1640 - 1647, Nov 1988.

[KARO2] M. J. Karol, R. D. Gitlin, "High-Performance Optical Local and Metropolitan Area Networks: Enhancements of FDDI and IEEE 802.6 DQDB", IEEE Journal on Selected Areas in Communications, Vol. 8, No. 8, October 1990.

[KARO3] M. J. Karol, "Optical Interconnection Using ShuffleNet Multihop Networks in Multi-Connected Ring Topologies," Proc. ACM SIGCOMM'88 Symposium, pp. 25-34, Aug.1988.

[PATE1] J. H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors", IEEE Trans. Comput., Vol. C-30, pp. 771-780, Oct. 1981.

[WISM1] D. A. Wismer and R. Chattergy, "Introduction to Nonlinear Optimization", North Holland, New York, 1978.

CHAPTER 6

CONCLUSION AND FURTHER RESEARCH

6.1 Conclusion

We have established an improved means for the integration of multiservices in a high speed metropolitan area network (MAN). The performance for such an integration has been analyzed and the network behaviour has been analytically studied under various conditions.

The metropolitan area networks considered are the Distributed Queue Dual Bus (DQDB) and the ShuffleNet because they both meet the requirements for high speed networks that can integrate multiservices. The DQDB MAN as it is currently considered by IEEE project 802 has limited capabilities and speed. The present study presents an improvement to the DQDB protocol. With the proposed modified protocol and hardware based on high speed material such as GaAs, the DQDB MAN can handle a wide range of services such as video, voice, wideband data and narrowband data at much higher speed (1.5 Gbps). ShuffleNet is also a MAN that can handle multiservices in a high speed environment. The contribution to ShuffleNet in this work is the improvement on its routing policies.

Both of these metropolitan area networks are high speed networks that can perform in the gigabits per second range.

They both have protocols that have fast responses and are not based on packet retransmissions when errors are detected. Errors are in fact handled by higher level protocols such as the LLC. The transmission of frames is controlled; every station when transmitting knows that it is the only one using the facilities at any particular instant and no collision of packets is possible. Commands received at a station are instantaneously verified and responded to. The protocol overhead is mostly found at the frame level and is reduced to the strict essentials for the protocol interactions. The frame header is used to identify the service, control the transmission and detect errors. In the new DQDB the proposed error detection is bit voting in the protocol control elements resulting into modular redundancy which also performs some error correction without any loss of time. Equitable use of the lines by all stations is perhaps the only aspect that although acceptable could be improved in DQDB. In shuffleNet the contribution of the present work is on the routing strategy that would distribute the load equitably in case of unbalanced load.

The unfairness of the DQDB protocol has been studied [CONT1] and propositions presented to minimize its effect under heavy load. Furthermore, the proposed DQDB protocol makes the most efficient use of the facilities by making the asynchronous transmissions use the subframes reserved for synchronous transmissions when these latter are not utilized. The protocol is also geared in serving multiservices by allocating specialized subframes to each of the services. In this case no interference is possible between the services. Indeed the video service is assigned a specific subframe that is reserved for video transmissions. Similarly the voice service is assigned a subframe reserved for the voice transmissions and wideband data as well as narrowband data are assigned their specific subframes for their respective transmissions. Each of these transmissions is using the new DQDB protocol independently from each other. A particular aspect of the proposed protocol is that the video subframe when empty can be used by the voice service and furthermore the video subframe and voice subframe when not used, can be taken by either wideband or narrowband data. This is known as movable boundary and optimize the utilisation of the available facilities. It should be noted that voice service has priority over wideband or narrowband data in the utilization of the video subframe.

Voice service can utilize the extra bandwidth offered by the video subframe even though it is an isochronous communication service. The concern is that in isochronous communication the same slot within all the video subframes is reserved for the duration of the voice call. If a video call comes in, it would have higher priority for the utilization of that subframe than the voice service. What we want to avoid is the sudden preemption and termination of the voice call. Fortunately it has been established within this study that voice traffic is independent of video traffic because the estimated minimum time for video call establishment is, on the average, equal to the voice holding time. The video slots can then be used by the voice service without the video service being impaired. Thus when a video call arrives, a video slot will be available by the time the access unit (AU) prepares the call and is ready to transmit.

Priority schemes that would efficiently resolve the contention for the use of the video or voice subframe by the wideband or narrowband data have been studied. Four priority schemes are considered. A model of the proposed protocol has been designed and a thorough analysis performed to determine the best scheme that would minimize the delay in the data services. The restricted priority was found to be the best. In this priority scheme the narrowband and wideband data have equal priority in accessing video and voice subframes; however narrowband data loses access when the wideband data queue reaches a given threshold. The exact nature of this threshold or the algorithm to determine it is for further studies. An analysis was conducted for various video and voice utilization conditions. It was observed that wideband data is not largely affected by any of the priority schemes. The effect of the priority schemes is on narrowband data and results clearly show that restricted priority is the best choice with respect to narrowband data queueing delay.

The fault probabilities and their effects on the service rate are analyzed for the modified DQDB protocol. Intermittent errors and stuck-at fault errors that would appear an the AU interfaces are investigated for the effects they have on the service rate and on the overall performance. A reliability model with faults affecting the protocol is defined to perform the analysis. In the model the error corrections such as triple modular redundancy (TMR) and quintuple modular redundancy (QMR) are used for intermittent as well as stuck-at faults.

In the case of intermittent errors it was found that when applying TMR the service rate improves to near the maximum because of the error correction. When the error is stuck-at fault a noticeable change and disruption to service in the network is observed. Even the healthy AU has a drastic drop in service rate. The faulty AU on the other hand experiences an increase in service to equal practically the healthy AU. This is occurring because the faulty AU is stealing slots. When TMR is applied to this case, the service rate of both healthy and faulty AUs improves.

When QMR is applied to the model with intermittent error there is a slight improvement over the results obtained with TMR. In the case of the stuck-at fault, QMR has about 7% improvement over TMR. The decision of applying one or the other modular redundancy is an economic decision since QMR provides a small improvement over TMR. Implementing QMR is only justified if the cost of QMR is not much greater than that of TMR.

The Weibull model was then applied to the various probabilities to see the effect errors as well as component redundancy have over the life of the supporting circuitry. Weibull model with constant failure rate as well as linear increasing failure rate are analyzed. The results show that in either cases modular redundancy improves service rate during system degradation but the circuit life span is not substantially affected.

The Contribution that was made with respect to ShuffleNet is the evaluation of efficient routing techniques that are robust even under faults and nonuniform load. On top of minimizing the transmission delay these routing techniques attempt to balance utilization of the wavelengths.

Three routing policies were considered. The optimum routing policy tries to equalize the traffic carried on each wavelength by minimizing the wavelength differential utilization. The random routing policy starts with an initial guess on the alternate routing probabilities. The alternate path with the least number of hops is selected with a given probability. The greedy routing policy is a specific case of the random policy. Only one path for any given source-destination route is selected with the least number of hops.

Since the optimum routing policy optimizes the wavelength differential utilization, it was expected to give best results in this respect when compared to the other policies. However the results did not come significantly better than those of the random policy. Under non uniform traffic flow, the three policies have similar behaviour than their respective uniform traffic flow results but at a much higher scale.

Results show that the performance of the optimal policy leads to a local minimum. To obtain a global minimum, we have to run in real time the gradient routine which may not be visible thus indicating the danger of applying the optimal policy in this application. Specially in case of applications where large burst of high rate data are the norm rather than the exception. On the other hand, the random policy is more cost effective due to the visibility of computation and performance obtained.

6.2 Further Research Work

The above work although complete can still be expanded by further research. Some of the works that could be developed directly related to this research are:

1. In the case of the restricted priority scheme, analysis and computation of the optimal threshold to be reached by wideband data queue when given higher priority over the narrowband data queue in using the additional bandwidth.

2. Computer simulation of the DQDB restricted priority scheme for a much larger network in real time.

3. Computation of a 3 dimensional model for DQDB as outlined in this work, namely the analysis of the case of two faulty stations.

4. Reflection of priority, errors, and fairness problems on the performance of the different integrated services.

5. Implementation of the proposed protocol for DQDB.