

UNIVERSITY OF CALIFORNIA

Los Angeles

Access Protocols

for

High Speed Fiber Optics Local Networks

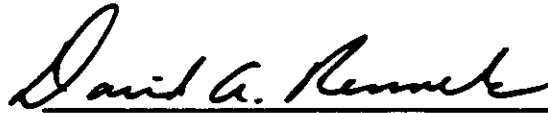
A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Computer Science

by

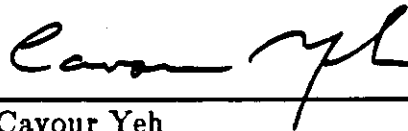
Paulo Henrique de Aguiar Rodrigues

1984

The dissertation of Paulo Henrique de Aguiar Rodrigues is approved.



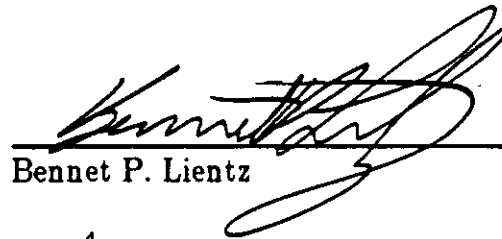
David A. Rennels



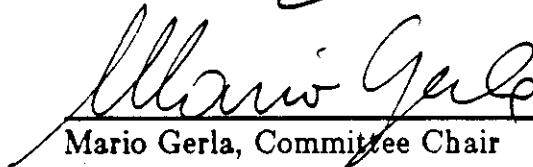
Cavour Yeh



Bruce Rothschild



Bennet P. Lientz



Mario Gerla, Committee Chair

University of California, Los Angeles

1984

To my mother, Nilza,
for her love and
rare example of understanding and perseverance.

TABLE OF CONTENTS

	page
TABLE OF CONTENTS	iv
LIST OF FIGURES	viii
LIST OF TABLES	xi
ACKNOWLEDGMENTS	xii
VITA	xiii
PUBLICATIONS	xiii
ABSTRACT OF THE DISSERTATION	xiv
1 INTRODUCTION	1
1.1 THE NEED FOR HIGH SPEED LANs	1
1.1.1 CURRENT BOTTLENECKS IN DATA TRANSFER	3
1.2 CHOICE OF DIRECTIONS	4
1.2.1 WHY FIBER OPTICS	4
1.2.1.1 LIMITATIONS OF ETHERNET-TYPE FIBER NETWORKS	6
1.2.2 WHY BUS TOPOLOGY	7
1.2.2.1 DESIGN GOALS	10
1.2.2.1.1 ROBUSTNESS	11
1.2.2.1.2 EFFICIENCY	11
1.2.2.1.3 FAIRNESS	11
1.2.2.1.4 EASE OF IMPLEMENTATION	12
1.2.2.1.5 EASE OF EXPANSION	12
1.2.2.1.6 GUARANTEED DELAY	12
1.2.2.2 PERFORMANCE MEASURES	13
1.3 EXISTING PROTOCOL/TOPOLOGIES FOR UNIDIRECTIONAL BUSES	16
1.3.1 SINGLE BUS TOPOLOGIES	16
1.3.1.1 Z TOPOLOGY	16
1.3.1.2 C TOPOLOGY	17
1.3.1.2.1 C-Net	17
1.3.1.3 DUAL BUS TOPOLOGY	18
1.3.1.3.1 DCR	19
1.3.1.3.2 Fasnet	19
1.4 DISSERTATION OUTLINE	20
2 TOKEN PROTOCOLS	23
2.1 INTRODUCTION	23
2.2 U-NET PROTOCOL	25
2.2.1 THE ACCESS PROCEDURE	26
2.2.2 END STATION ELECTION PROCEDURE	29
2.3 TDT-Net	34

2.3.1	PARAMETERS d_s AND d_r	35
2.4	PERFORMANCE ANALYSIS	38
2.5	U-NET RESULTS	38
2.5.1	DELAY PERFORMANCE	38
2.5.1.1	LIGHT LOAD	38
2.5.1.2	HEAVY LOAD	40
2.5.2	UTILIZATION	41
2.6	TDT-NET RESULTS	42
2.6.1	DELAY PERFORMANCE	43
2.6.1.1	LIGHT LOAD	43
2.6.1.2	HEAVY LOAD	44
2.6.2	UTILIZATION	45
3	BUZZ-NET	47
3.1	INTRODUCTION	47
3.2	PRINCIPLES OF OPERATION	49
3.3	THE ALGORITHM	50
3.4	BUZZ SIGNAL IMPLEMENTATIONS	54
3.5	NEW STATIONS JOINING THE NETWORK	57
3.6	PERFORMANCE ANALYSIS	60
3.6.1	UTILIZATION AT HEAVY LOAD	60
3.6.2	INSERTION DELAY	63
3.6.3	MAXIMUM INSERTION DELAY	64
Appendix 3.1	<i>R</i> GUARANTEES PROPER OPERATION FOR BUZZ-NET	67
Appendix 3.2	WORST CASE INSERTION DELAY FOR BUZZ-NET	69
4	RANDOM ACCESS WITH TIME-OUT CONTROL	75
4.1	INTRODUCTION	75
4.2	THE PROTOCOL	75
4.2.1	MINIMUM VALUE T_0 FOR FAIRNESS	76
4.3	PERFORMANCE ANALYSIS	78
4.3.1	UTILIZATION	78
4.3.2	DELAY PERFORMANCE	79
4.4	CONCLUSION	80
5	TOKEN-LESS PROTOCOLS	81
5.1	INTRODUCTION	81
5.2	PRINCIPLES OF OPERATION	83
5.3	THE PROTOCOL	85
5.3.1	BASIC TOKEN-LESS PROTOCOL	85
5.3.2	VARIOUS IMPLEMENTATIONS	87
5.3.2.1	TLP-1	89
5.3.2.2	TLP-2	93
5.3.2.3	TLP-3	96
5.3.2.4	TLP-4	99
5.4	RECOVERY AND JOINING	104
5.5	PERFORMANCE ANALYSIS	108
5.5.1	NETWORK UTILIZATION	109
5.5.2	DELAY PERFORMANCE	110

5.5.2.1	LIGHT LOAD	110
5.5.2.2	HEAVY LOAD	111
5.6	CONCLUSIONS	112
6	COMPARATIVE ANALYSIS AND SIMULATION RESULTS	113
6.1	INTRODUCTION	113
6.2	PERFORMANCE MEASURES FOR EXISTING PROTOCOLS	114
6.2.1	EXPRESS-NET, D-NET AND C-NET	114
6.2.2	FASNET	115
6.2.3	ETHERNET	116
6.3	UTILIZATION AND INSERTION DELAY COMPARISON	118
6.3.1	S vs α	118
6.3.2	S , IDL AND IDH	120
6.4	COMPARATIVE ANALYSIS THROUGH SIMULATION	126
6.4.1	DISCRETE EVENT SIMULATOR	126
6.4.2	A GENERAL INSERTION DELAY COMPARISON	130
6.4.3	TLP SIMULATION RESULTS	133
6.4.3.1	EXAMPLE 0: EQUALLY LOADED, SINGLE PACKET MESSAGE	133
6.4.3.2	EXAMPLE 1: SINGLE HEAVY LOADED STATION, SINGLE PACKET MESSAGE	135
6.4.3.3	EXAMPLE 2: SINGLE HEAVY LOADED STATION, MULTIPACKET MESSAGE	139
6.4.3.4	EXAMPLE 3: EQUALLY LOADED NETWORK, SMALLER ACTIVE SET	143
6.4.3.5	EXAMPLE 4: SINGLE HEAVY LOADED STATION, SMALLER ACTIVE SET	146
6.4.3.6	COMPARING TLP VERSIONS	148
7	APPROXIMATE ANALYSIS FOR OSCILLATING POLLING	150
7.1	INTRODUCTION	150
7.1.1	THE MODEL	151
7.1.1.1	DETERMINATION OF b_{i1} AND b_{iN}	154
7.1.2	AVERAGE QUEUEING DELAY W	160
7.1.3	RESULTS	162
7.2	CONCLUSIONS	164
8	BUILDING SYSTEMS WITH A LARGE NUMBER OF STATIONS	167
8.1	INTRODUCTION	167
8.2	DUAL BUS TOPOLOGY OPTIMIZATION	169
8.2.1	OPTIMIZATION WITH EQUAL COUPLERS	172
8.2.2	OPTIMIZATION WITH SYMMETRIC COUPLERS	173
8.2.3	HYBRID OPTIMIZATION	178
8.2.4	SINGLE TAP OPTIMIZATION	181
8.3	PASSIVE STAR/BUS CONFIGURATION	183
8.3.1	WIRE CONTROL INSIDE A GROUP	184
8.3.2	POWER BUDGET AND UTILIZATION IN A STAR/BUS NETWORK	188
8.4	LINEAR EXPANSION THROUGH ACTIVE REPEATERS	192
8.5	LINEAR EXPANSION THROUGH BRIDGES	193
8.5.1	COMPATIBILITY BETWEEN BRIDGES AND ACCESS	

8.6	PROTOCOLS	196
	HIERARCHICAL CONNECTIONS USING GATEWAYS	200
9	CONCLUSIONS	205
9.1	SUMMARY OF RESULTS	205
9.2	EXTENSIONS OF THIS RESEARCH	207
References	209



Fig. 5.8 - TLP-4 State Diagram.	101
Fig. 6.1 - Fasnet slot.	115
Fig. 6.2 - Utilization vs α for $N = 15$	119
Fig. 6.3 - Utilization vs α for $N = 30$	119
Fig. 6.4 - Utilization vs α for $N = 100$	120
Fig. 6.5 - State Diagram for Buzz-Net Simulation.	128
Fig. 6.6 - Insertion Delay vs Bus Utilization (span=1000m).	131
Fig. 6.7 - Ex.0: TLP-3,4 ID vs Bus Utilization (span=10,000m).	134
Fig. 6.8 - Ex.0: TLP-3,4 QD vs Bus Utilization (span=10,000m).	134
Fig. 6.9 - Ex.1: TLP-4 ID and QD vs Station 8 Load.	136
Fig. 6.10 - Ex.1: TLP-1,2,3 Station 8 Delays vs Station 8 Load.	136
Fig. 6.11 - Ex.1: TLP-1,2,3 Background Delays vs Station 8 Load.	137
Fig. 6.12 - Ex.2: TLP-4 ID and QD vs Station 8 Load.	139
Fig. 6.13 - Ex.2: TLP-1,2,3 Station 8 Delays vs Station 8 Load.	140
Fig. 6.14 - Ex.2: TLP-1,2,3 Background Delays vs Station 8 Load.	140
Fig. 6.15 - Ex.3: TLP-3,4 ID and QD vs Input Load.	143
Fig. 6.16 - Ex.3: TLP-1,2 ID and QD vs Input Load.	144
Fig. 6.17 - Ex.4: TLP-4 ID and QD vs Station 4 Load.	146
Fig. 6.18 - Ex.4: TLP-1,2,3 Station 4 Delays vs Station 4 Load.	147
Fig. 6.19 - Ex.4: TLP-1,2,3 Background Delays vs Station 4 Load.	147
Fig. 7.1 - Iterative Procedure To Calculate b_{i1}	159
Fig. 7.2 - Average Error (%) vs Normalized Utilization ($N=15$).	163
Fig. 7.3 - Station 8 Error (%) vs Normalized Utilization ($N=15$).	164
Fig. 7.4 - Station 1 Error (%) vs Normalized Utilization ($N=15$).	165
Fig. 7.5 - Station 1 Error (%) vs Normalized Utilization ($N=15$).	165
Fig. 8.1 - Optical Tap and Station Connections.	170

Fig. 8.2 - Configuration with N stations. 170

Fig. 8.3 - Two Tap Coupler. 174

Fig. 8.4 - 3-block network layout. 178

Fig. 8.5 - Passive Star/Bus Configuration. 183

Fig. 8.6 - Corruption by first transmission from a group. 186

Fig. 8.7 - Group transmission corrupted by another group. 186

Fig. 8.8 - Bridge Connections. 195

LIST OF TABLES

	page
Table 6.1 - Performance results for $N = 15$ and $l = 1$ km.	121
Table 6.2 - Performance results for $N = 15$ and $l = 5$ km.	122
Table 6.3 - Performance results for $N = 100$ and $l = 1$ km.	123
Table 6.4 - Performance results for $N = 100$ and $l = 5$ km.	125
Table 6.5 - Best choice of protocols.	126
Table 6.6 - Ex.1: TLP-1,2,3,4 Maximum Bus Utilization.	137
Table 6.7 - Ex.1: 95% Confidence Intervals for QD at Station 8.	138
Table 6.8 - Ex.2: TLP-1,2,3,4 Maximum Bus Utilization.	141
Table 6.9 - Ex.2: 95% Confidence Intervals for QD at Station 8.	142
Table 6.10 - Ex.3: TLP-1,2,3,4 Maximum Bus Utilization.	145
Table 6.11 - Ex.3: 95% Confidence Intervals for QD Averaged Over All Stations.	145
Table 6.12 - Ex.4: TLP-1,2,3,4 Maximum Bus Utilization.	148
Table 6.13 - Ex.4: 95% Confidence Intervals for QD at Station 4.	149
Table 7.1 - Example of values for b_{i1} and b_{iN}	161
Table 8.1 - Power Margin Required for N Equal Couplers.	173
Table 8.2 - N_{\max} for Equal Coupler Optimization.	174
Table 8.3 - N_{\max} for Symmetric Coupler Optimization.	177
Table 8.4 - N_{\max} for Hybrid Optimization.	180
Table 8.5 - N_{\max} for Single Tap Optimization.	182
Table 8.6 - N_{\max} for a Star/Bus example.	190
Table 8.7 - Max Utilization for a Star/Bus with 120 stations.	191

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my Chairman, Professor Mario Gerla, for the support and encouragement throughout this research. Sincere thanks are also extended to Professors David Rennels, Cavour Yeh, Bruce Rothschild and Bennet Lientz for serving on the doctoral Committee.

Many thanks are also expressed to all my friends who supported me through the most difficult periods of this research. Specially, I am deeply indebted to Baron O. Grey who generously shared his knowledge of the computing systems and provided our research environment with tools that made our work less painful and more exciting. I express my appreciation to Chet Lanctot for developing the initial version of the network simulator.

Financial support to this work was given by the Brazilian Research Council (CNPq)-Brazil, through fellowship 200.123/79; the Federal University of Rio de Janeiro (UFRJ)-Brazil; NSF under contract No. ECS-80-20300; and in part by Hewlett Packard and the State of California under a UC-MICRO grant. To all these entities I would like to express grateful appreciation.

Special thanks go to Kathleen O'Grady for her friendship and her numerous suggestions on matters of style and grammar.

VITA

- November 1, 1951 -- Born, Belo Horizonte, Minas Gerais, Brazil.
- 1974 -- B.S., Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, Brazil.
- 1975 - -- Computer Analyst, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.
- 1977 -- M.S., Coordenação dos Programas de Pós-Graduação de Engenharia (COPPE), Rio de Janeiro, Brazil.
- 1982 - 1984 -- Research Assistant, Computer Science Department, University of California, Los Angeles.

PUBLICATIONS

Rodrigues, P., L. Fratta, and M. Gerla, "Token-less Protocols for Fiber Optics Local Area Networks," in *Proceedings 1984 IEEE International Conference on Communications (ICC'84) - Vol 3*, Amsterdam, Netherlands: May 14-17, 1984, pp. 1150-1153.

Gerla, M., P. Rodrigues, and C. Yeh, "U-NET: A Unidirectional Fiber Bus Network," in *Proceedings Fiber Optic Communications/Local Area Networks (FOC/LAN 84)*, Las Vegas, NV: September 17-21, 1984.

Gerla, M., P. Rodrigues, and C. Yeh, "BUZZ-NET: a hybrid random access/virtual token local network," in *Proceedings 1983 IEEE Global Telecommunications Conference (Globecom'83) - Vol 3*, San Diego, CA: November 28 - December 1, 1983, pp. 1509-1513. Local

ABSTRACT OF THE DISSERTATION

Access Protocols

for

High Speed Fiber Optics Local Networks

by

Paulo Henrique de Aguiar Rodrigues

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 1984

Professor Mario Gerla, Chair

Emergence of new applications requiring high data traffic necessitate the development of high speed local area networks. Optical fiber is selected as the transmission medium due to its inherent advantages over other possible media and the dual optical bus architecture is shown to be the most suitable topology. Asynchronous access protocols, including token, random, hybrid random/token, and virtual token schemes, are developed and analyzed. Exact expressions for insertion delay and utilization at light and heavy load are derived, and intermediate load behavior is investigated by simulation. A new tokenless adaptive scheme whose control depends only on the detection of activity on the channel is shown to outperform round-robin schemes under uneven loads and multipacket traffic and to perform optimally at light load. An approximate solution to the queueing delay for an oscillating polling scheme under chaining is obtained and results are compared with simulation. Solutions to the problem of building systems with a large number of stations are presented, including maximization of the number of optical couplers, and the use of passive star/bus topologies, bridges and gateways.

CHAPTER 1

INTRODUCTION

1.1 THE NEED FOR HIGH SPEED LANs

Local area networks (LANs) are essentially switching technologies designed to provide reliable digital data transmission in a limited geographic area (e.g., within a single facility or campus of facilities), to serve hosts, minis, micros, work stations, and other digital devices [Liss83].

LANs provide a direct, short (most of the times one hop), usually broadcast, and relatively noise-free path for a given pair of users. LANs broadcast capability eliminates buffer requirements in intermediate nodes. Error recovery can be simply implemented through retransmission or end-to-end acknowledgement since the interaction between sender and receiver are almost instantaneous compared to long haul networks.

These unique features have helped LANs become increasingly popular. Most existing LANs serve environments where host-to-host and interactive terminal traffic are the only load sources. Traffic measurements in an operational Ethernet running at 3Mbps have shown an average line utilization of only 0.60% to 0.84%, with a peak utilization of 40% during rush hours [Shoc80].

Recent years have witnessed a rapid growth in local area communication needs corresponding to rapidly increasing user sophistication and the emergence

of new applications, especially those addressing the automated and distributed office environment. Block transfer and video applications are among those that require bandwidth not yet provided by actual LAN implementations (bit rate < 50Mbps). Listed below is a set of applications and their peak data rates [IEEE82].

<u>TYPE OF SOURCE</u>	<u>PEAK DATA RATE (kbps)</u>
File transfer/Block transfer	20,000
Video (uncompressed)	30,000
Voice (immediate)	64
Laser printer	256
Graphics (uncompressed)	256

To support these high data traffic requirements, local area networks with larger bandwidths must be designed. Although other switching technologies, such as CBXs, can provide some communication to the local environment, LANs may be the only available option when high bandwidths are required [Pfis82].

Very high speed LAN design requires an integrated choice of transmission medium, topology, and access protocol. Access protocols are dependent upon the underlying topology, and medium selection may restrict the range of feasible topologies. Unfortunately, existing LANs cannot upgrade to very high data rates due to medium, topology, or access protocol limitations.

One purpose of this research is to identify a medium and topology appropriate for the development of a very high speed LAN. In this selection, we are concerned about issues of robustness, efficiency, fairness, ease of implementation, ease of expandability, and delay. For the chosen combination of medium and topology we developed and evaluated protocols which satisfied the above issues.

1.1.1 CURRENT BOTTLENECKS IN DATA TRANSFER

At present, most network interface units (NIU) consist of two basic parts: the transceiver and the controller. The transceiver couples directly to the line and performs basic frame functions: error checking, packet delimiting and address recognition. The controller usually contains both a CPU and memory, and performs DMA functions during transmission or reception of a packet. The controller must also support the link level protocol and the protocol which controls transfers from attached devices (terminals, host, diskpacks, file servers, etc.). The NIU is frequently used as a multiplexer, and as such it concentrates many single sources into one access point. LANs usually provide a layered address structure which allows direct addressing to the physical ports or attached devices. Occasionally the NIU is part of a gateway which allows communications with other local nets.

Simulation results show that the chief limitation of LAN performance is due to the switching functions of the interface for buffer management and protocol processing [Yeh79, Magl82]. In those experiments the processing capability of the micro-processor based controller unit becomes the bottleneck of the system under heavy load. The existence of a very high speed communication medium will call for innovative transfer operations between devices and simpler protocols to capitalize on the available capacity. For example, if a local network can transmit at 1Gbps, then remote memory to memory transfers might be feasible. In reality, a network that fast would work transparently, and the entire transfer would behave as a local DMA transfer. Because we are using a very reliable and high speed medium, segmented messages are not necessarily required for efficient line utilization (in contrast to requirements when lines are

unreliable) and the ability to transmit long messages minimizes the overhead of packet assembly and disassembly at communication nodes. Previous work in this area has shown that implementing simple protocols directly on hardware makes a very high speed controller unit feasible [Blau79].

We emphasize that the development of very high speed interconnection media must be complemented by new hardware/software designs to allow complete utilization of that technology. We will not pursue this issue further, but we believe that new ideas in the high-level-protocol/OS/architecture fields will match the needs defined.

1.2 CHOICE OF DIRECTIONS

1.2.1 WHY FIBER OPTICS

Among possible choices of a medium for the implementation of a very high speed LAN (namely, coaxial cable, microwave, waveguide and fiber) fiber is the most cost-effective and promising technology. Waveguide is rejected because of cost and difficulty of practical installation. Microwave is inherently point-to-point, expensive, susceptible to interference, and not adequate for the local environment. Microwave links between buildings are feasible but only justified as gateway implementations. Therefore, only fiber and coaxial cable remain under consideration. Although fiber is a recent developed technology, it has inherent advantages over coaxial cable, as follows:

- a. fiber has larger bandwidth/km.
- b. fiber has typical cost of \$0.05 Mhz/Km compared to coaxial cable cost of \$3 Mhz/Km [Lute82].

- c. fiber (single-mode) has dispersion of less than 0.01 ns/Km compared to a coaxial cable dispersion of 20 ns/Km [Lute82].
- d. fiber has immunity against electrical and magnetic interference (EMI, crosstalk, noise, short circuiting, explosions, sparks, radiated signals, etc.) [Jone76, Mull77, Epwo77].
- e. optical fiber links offer secure transmission because they are difficult to tap without noticeable signal loss.
- f. fiber is much lighter and smaller.
- g. fiber has a typical loss of -.16 db/Km compared to a coaxial cable loss of -13 db/Km.
- h. fiber can be easily and extensively multiplexed.

Fibers can be multimode or single-mode. In a multimode fiber the light propagates in different modes which follow different optical paths. Because modes are delayed differently, a light pulse deforms and expands as it propagates along the fiber. This deformation called modal dispersion reduces the available bandwidth/km for multimode fibers. Modal dispersion can be reduced by using multimode graded-index fiber which forces the light to travel slower along the longer paths, thus minimizing the difference in propagation delay among the modes. However, complete modal dispersion elimination only occurs with single-mode fibers, where only one mode is allowed to propagate. Because the fiber functions as a wave guide, single-mode propagation is achieved by using a very small core diameter.

For very high data rates single-mode transmission must be used. The small size of the core requires precision manufacturing techniques for fiber fabrication and the use of lasers as light sources. Nowadays single-mode technology has sufficiently matured and high performance reliable components have been fabricated.

At this writing, coaxial cable still has the advantage that off-the-shelf components of the CATV industry are readily available and cheaper than corresponding components for the optical technology. At present, performance and price of connectors and couplers for use on taps are the main obstacles for widespread use of fiber optics. The major loss of signal in fiber is due to coupler insertion loss. Values of the order of -2db (one transmitter tap and one receiver tap per coupler) are industry achievable, and progress in this area is expected in coming years. In practical implementations each coupler may require two optical connectors or splices for its connection. Low-loss lens connectors have been fabricated providing an average loss of -0.54dB [Masu82]. Single-mode splicing techniques are well developed and they provide connections with minimum loss ($< -0.05\text{dB}$) under field conditions. Because of the above losses an acceptable number of stations in an optical fiber LAN is only achieved if the number of taps per station per bus is minimized.

1.2.1.1 LIMITATIONS OF ETHERNET-TYPE FIBER NETWORKS

Ethernet-type optical fiber networks use non-persistent CSMA-CD as access method: if the bus is idle, transmit; if the bus is busy or a collision occurs, reschedule retransmission for some time in future, with random exponential backoff. Therefore, the average retransmission time increases exponentially with number of collisions. Packets are discarded after a maximum number of retransmissions are unsuccessful. Discarded packets are eventually retransmitted due to the action of higher level protocols. The underlying topology may vary. The Mitrenet facility in Bedford, Massachusetts, uses a dual unidirectional optical bus topology [Ping82, Hopk80]. The Novanet, at Lawrence Livermore National Laboratories, uses an active star configuration [Ping82]. Xerox has

proposed two optical fiber architectures. Fibernet I [Raws78] uses a passive star configuration and Fibernet II [Raws82] uses an active star. Specifications for Ethernet compatible implementations can be found in [DEC80].

A general problem with Ethernet is that no bounded delay can be guaranteed for any transmission. A second problem is low efficiency for high transmission rates. Collision detection in CSMA-CD requires that transmission time $>$ round trip delay. In a very high speed environment transmission times become smaller than the round trip propagation delay, and carrier sense becomes ineffective. Under those conditions, CSMA starts performing as an Aloha channel, and the maximum achievable throughput is 18% [Abra73]. Aloha channels are unstable if no control is exercised on the channel [Lam75]. Collision detection coupled with randomization of retransmission brings control over the channel. However, if bit padding is used for short packets so that transmission time = round trip delay, throughput decreases to zero with decreasing packet lengths. Because propagation delay in the network is independent of the data rate and CSMA delays are not bounded, we conclude that Ethernet-type networks are not suitable for very high speed transmission.

1.2.2 WHY BUS TOPOLOGY

Fiber optics topologies can be configured in three basic ways: star, bus, and ring. Independent of the specific topology, the major difficulty in connecting to a very high speed optical medium is the electronic circuitry, which must function at the line speed. For switching speeds lower than 250 Mhz, 100K ECL circuits are available and can perform the digital functions. For higher switching speeds, either optical logics or circuits using discrete microwave electronic

components are necessary. To date, optical technology has been unable to provide logical elements that are as fast and efficient as their electronic counterparts, and discrete microwave components are expensive and unavailable off-the-shelf. Feasibility and cost-effectiveness of a very high speed LAN implementation requires that the electronic logic functioning at the line speed be kept to a minimum. In a more general sense, reliability of the transmission medium may be enhanced by keeping active electronics to a minimum.

A star topology provides a point-to-point communication link between any pair of stations with an end-to-end propagation delay being suffered by any transmission. Furthermore, simultaneous transmissions always collide at the central node. At high speeds packet transmission time becomes smaller than the end-to-end propagation delay and the star behaves as a satellite link. As explained in Section 1.2.1.1, CSMA/CD performs poorly under the above conditions. Because optical star implementations [Raws82, Raws78, Ping82] use CSMA/CD as the underlying protocol, they perform poorly at high speed. Reservation schemes as adopted for satellite links are too complex to be considered for a LAN implementation.

Optical bus architectures use passive taps without active electronics interfering directly with the medium. Also, address and flag recognition can be done at a speed much slower than the line, and the only electronics required to run at line speed are the clock recovery circuit, the carrier sense circuit and the line buffer circuitry (which can be kept to a minimum, the amount necessary to provide byte demultiplexing and transfer to a slower speed logic).

Ring topology requires that certain amount of active electronics be inserted into the data path for each station joining the network. Therefore reliability degrades as the number of stations increases.

Point-to-point low speed links can be effortlessly converted to optical links, if transmission speed is maintained. Thus, rings using coaxial cable can be easily upgraded by substituting fiber for copper, a conversion which has been successfully accomplished in many places [Ping82].

Nevertheless, when very high speed links are needed, ring topology presents some serious drawbacks. A crucial problem in ring implementation is the necessity to perform address recognition and flag setting at line speed and, usually, depending on ring implementation, some buffering must also be provided. For example, Pierce ring [Pier72] is a slotted ring where the destination, upon matching its address with the destination address in the slot, sets the empty bit in the slot header. The used slot is then removed by a central controller upon detection of the empty condition. In the Loomis ring [Loom73], the destination sets the accept bit in the header of the packet addressed to it, and the sender or any other station is responsible for removing the packet from the ring upon detection of the accept condition. In the buffer insertion ring [Liu75] the destination is responsible for removing a packet addressed to it. In the Farmer and Newhall ring [Farm69], though no address recognition is required for packet removal, the sender is responsible for estimating total ring delay and message removal is by shutting off the receiver shortly before the message is expected to return. This removal technique is infeasible in a dynamic and very high speed environment. Furthermore, address recognition is still necessary for packet acceptance. All present ring protocols require address and flag setting

hardware working at the line speed. Therefore, high cost ring implementation is expected in high speed, and reliability is at risk.

Considering the above requirements, bus topology seems very promising for high speed local network architecture. We further compare the single unidirectional bus topologies in Figs. 1.1, 1.2 and 1.3 with the dual unidirectional bus topology shown in Fig. 1.4. In the dual bus topology there are only two connecting points per station per bus and expansion is easily done at both ends. The Z topology in Fig. 1.1 has the disadvantage of requiring three connecting points per station on the same bus. This further limits the maximum number of stations that can be supported. Bus folding restricts practical implementations, and future expansion requires cutting the cable. The C topology in Figs. 1.2 and 1.3 has the same disadvantage of three connecting points per station as the Z topology. Expansion is also difficult because of bus folding. From the above, we realize that the dual bus topology suffers only half the insertion loss per station per bus, and offers easy expansion. Therefore, our research has concentrated on developing protocols for the high speed dual unidirectional optical bus topology.

1.2.2.1 DESIGN GOALS

The goals for our protocol/topology integrated design are : robustness, efficiency, fairness, ease of implementation, ease of expandability and guaranteed delay. These six points define the optimal guidelines for designing the protocols.

1.2.2.1.1 ROBUSTNESS

Robustness here means reliable operation and automatic recovery following station insertions and deletions. Improvements in reliability can be achieved by avoiding sophisticated hardware requirements (i.e. phase synchronization of all stations in the net, generation and detection of special packets, etc.). Our protocols should allow simple and reliable engineering solutions to necessary control procedures. Insertion and deletion of stations in the network should be transparently done, and only transitory interference should be observed. In the ideal protocol, deletion should have no effect on the network performance and insertions should only cause a small transitory degradation of performance. Automatic recovery following network failures should be built in the access protocol. No higher level intervention should be required.

1.2.2.1.2 EFFICIENCY

Efficiency means that the protocol should provide high throughput and low delay, especially when only a fraction of the stations are actively using the network. Ideally, when considering any set of active stations in a fixed topology, performance achieved by this set of stations should be independent of the network length.

1.2.2.1.3 FAIRNESS

Fairness implies that active stations should be served in a round robin fashion if all transmissions have equal priority. The high available bandwidth and, consequently, the low delays encountered in the network make the need for

priority schemes a secondary issue. In the exceptional case, a bridge or a gateway may need instantaneous high priority to insert external traffic and avoid the need for huge interface buffers.

1.2.2.1.4 EASE OF IMPLEMENTATION

Ease of implementation implies that the protocols should be simple enough to allow complete hardware implementation. It is inevitable that high bandwidth will require that control logic be implemented with technology like GaAs which allows gate delays of the order of picoseconds. Detailed hardware implementations are not the subject here. However, limitations of reliable detection and feasibility of implementation must be addressed when mechanisms that allow special patterns to be generated or detected are described.

1.2.2.1.5 EASE OF EXPANSION

Ease of expansion depends more on network topology and less on the protocol itself, if the above issues are resolved. As seen, the dual bus topology satisfies this requirement.

1.2.2.1.6 GUARANTEED DELAY

Guaranteed delay is of concern because the local network must be prepared to carry different kinds of traffic such as: low throughput-high delay traffic (i.e., interactive), low throughput-low delay traffic (i.e., OS to OS calls, real time control), high throughput-high delay (i.e., file transfer), low throughput-bounded delay (i.e., voice), high throughput-bounded delay (i.e., video), and others. It is important to be able to allocate bandwidth to stations

with different traffic requirements in such a way that those requirements are satisfied and fairness and performance are maintained. If the access protocol offers a bounded delay, the maximum number of sessions allocated to each kind of traffic is easily evaluated and higher protocols in charge of flow control and bandwidth allocation are greatly simplified.

1.2.2.2 PERFORMANCE MEASURES

In this section we introduce the performance measures and basic assumptions used in evaluating the new protocols proposed in this dissertation. The measures are also essential for comparison with other existing unidirectional schemes and are the following:

- (1) Queueing delay, defined as the total time spent in queue.
- (2) Average insertion delay ID , defined as the interval between the time when the packet moves to the head of the transmitting queue and the time when successful transmission begins. Note that insertion delay is equivalent to queueing delay when there is only one buffer per station. Insertion delay is evaluated as a function of the number of active stations and the offered load. The average is over all stations and over time. IDL and IDH designate ID at light and heavy load, respectively.
- (3) Maximum insertion delay MID , over all stations and over time. MID is a function of the number of stations and the offered load.
- (4) Heavy load bus utilization $S(i)$, defined as the net bus utilization when i stations are active and have infinite backlog.

The above measures, albeit simple, provide useful criteria to determine whether or not a bus protocol is suitable for a given application. For example, interactive and real time applications are particularly sensitive to average and maximum insertion delays. Batch data transfer is most affected by bus throughput efficiency. Queueing delay is used in the simulation results and approximate analysis in Chapters 6 and 7.

Several assumptions are made to render the models tractable. In the sequel we introduce some assumptions which apply to all models used in our study, along with some general definitions.

- (a) Performance is evaluated at steady state. Transient conditions are not investigated.
- (b) Whenever i stations are selected among N , we assume that all stations are equally likely. A subscript to a performance symbol indicates the index of the station where the performance is evaluated.
- (c) Information is transmitted in packets. A packet has a data field and a preamble. The data field includes headers (sender and destination addresses, data field length, etc), CRC fields and higher level information. We are not concerned with internal overhead in a data packet, so the preamble is the only overhead considered. In all analytical derivations data and preamble transmission times are assumed constant and equal to T_r and T_p , respectively. Thus, total transmission time $T = T_r + T_p$.
- (d) A token is a special sequence of bits or a well defined burst of carrier with transmission time equal to T_k .

- (e) The propagation delay between stations S_i and S_j is assumed to be the same in both busses, and is indicated by τ_{ij} . τ is the end-to-end propagation delay. Our analysis is restricted to the case of equally spaced stations. Hence $a = \tau_{i,i+1} = \tau/(N-1)$ and $\tau_{ij} = a(i-j)$ for $i \geq j$, where N is the total number of stations.
- (f) Consider the time instants:
- $EOC(b)$ as the time when END OF CARRIER occurs in the bus.
 - $EOC(s)$ as the time when END OF CARRIER is sensed at the station.
 - $SOT(b)$ as the time when a transmission starts in the bus.
 - $SOT(s)$ as the time when a transmission is initiated at the station.
- (g) If a station has a packet to transmit, the interval of time between the occurrence of $EOC(s)$ and $SOT(s)$ is assumed negligible. Thus, $EOC(s) - SOT(s) = 0$.
- (h) d is the reaction time of a station. To simplify the analytic expressions and without loss of generality we assume $d/2 = EOC(s) - EOC(b) = SOT(b) - SOT(s)$. Thus, the reaction time for a station, defined as the elapsed time between the END OF CARRIER in the bus and the start of the station transmission in the same bus, is equal to $SOT(b) - EOC(b) = d$. Similarly, there is a d second delay between sensing carrier from an upstream station and the interruption of an ongoing transmission. We assume stations have equal reaction time d .
- (i) The initial d seconds of the preamble may be corrupted by collisions. In fact, if a packet collides with p other downstream transmissions, the first K bits of its preamble (where $K = dG$, and $G =$ transmission rate

(bits/s)) correspond to the superimposition of $p+1$ transmissions. The preamble, therefore, should be large enough so that clock synchronization can be acquired despite initial garbage.

1.3 EXISTING PROTOCOL/TOPOLOGIES FOR UNIDIRECTIONAL BUSSES

1.3.1 SINGLE BUS TOPOLOGIES

1.3.1.1 Z TOPOLOGY

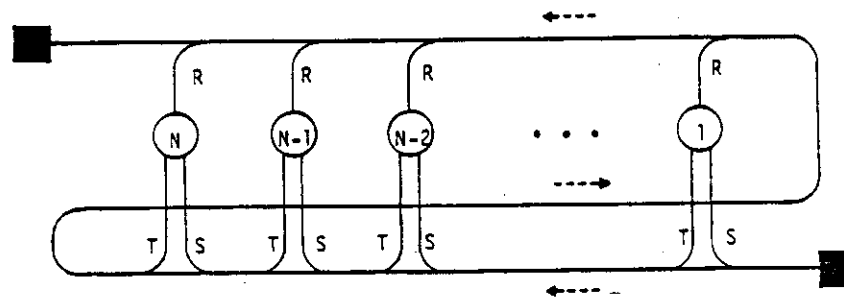


Fig. 1.1 - Express-Net.

A local network called Express-Net was proposed in [Frat81]. The topology is a single unidirectional bus connecting the stations as shown in Fig. 1.1. Tap S is able to sense incoming upstream transmissions. Tap T performs the transmission function and is able to abort ongoing transmission if tap S senses any incoming line activity. Tap R is the receiver. Tap R is able to receive a packet and detect end-of-train (EOT). A train is a succession of consecutive transmissions. In normal conditions the protocol works as follows.

Station 1 starts a train of messages by transmitting a locomotive (burst of energy) each time it senses EOT at tap R. If station 1 has a packet to transmit, it appends the packet to the locomotive. The other stations sense EOT at tap S

and append their own ready packet. A station can only transmit one packet per train.

1.3.1.2 C TOPOLOGY

1.3.1.2.1 C-Net

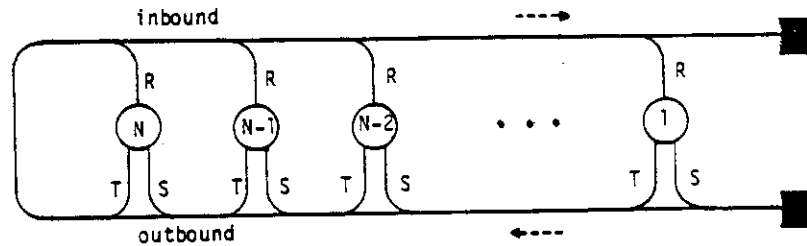


Fig. 1.2 - C-net.

C-net was proposed in [Mars81]. A station with a packet ready for transmission senses the outbound channel. If no activity is detected it starts transmission. If during transmission a packet transmitted by an upstream station is sensed, the station aborts its own transmission and waits for EOT at tap S to append its packet to the current train of messages. After a station has successfully transmitted a packet, a new packet can only be transmitted after the station hears its own packet at tap R and the end of train to which its packet belongs is also detected at tap R.

D-net was proposed in [Tsen82]. There is a locomotive generator whose function is to maintain a locomotive circulating through the network to allow stations to synchronize their transmissions. The locomotive can be a burst of carrier. A station senses the locomotive at tap S and, at the end of the train, it appends its own packet, if one is ready. The locomotive generator generates a new locomotive as soon as it detects EOT at its tap R. A station can only

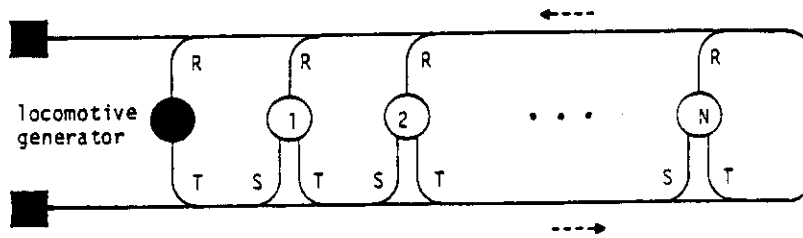


Fig. 1.3 - D-Net.

transmit one packet per train.

The locomotive generator is a single-point failure. Expansion to the left requires physically moving the locomotive generator.

1.3.1.3 DUAL BUS TOPOLOGY

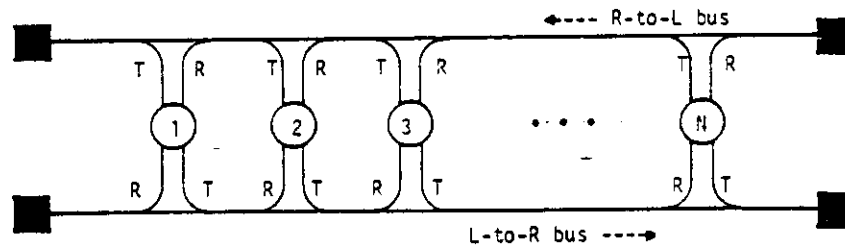


Fig. 1.4 - Dual Unidirectional Bus Topology.

In this topology protocols have the option to implement unidirectional or bidirectional transmissions. Unidirectional transmission allows increases in network throughput because stations only use bandwidth in the desired direction. However, knowledge of the desired direction implies that a session set-up phase must be performed to discover the physical location of the destination. Although bidirectionality wastes some useful bandwidth, it avoids the set-up phase and allows direct addressing by name. Addressing by name allows stations to send data to processes without knowing their physical location on the bus. Processes can be moved about in the network without the knowledge of

other processes. Addressing by name can be achieved by using a word associative buffer in each bus interface to hold the names of the processes resident in the attached computers. Session connections can be established in hardware without intervention from higher level protocols.

1.3.1.3.1 DCR

A recent protocol proposal for the dual bus topology, DCR-Net [Taka83], employs a deterministic resolution scheme on top of CSMA-CD to resolve collisions in a finite time. The normal mode of operation is random access CSMA-CD. Once a collision occurs, it is resolved using an implicit token passing scheme. This protocol, however, is not suitable to very high speed busses because it requires transmission times larger than the round trip delay. Buzz-Net, to be introduced in Chapter 3, was developed independently during this research and proposes a similar hybrid mode of operation (random and token passing). Nevertheless, Buzz-Net does not impose any restriction on the minimum packet length and, therefore, it is suitable to the high speed environment.

1.3.1.3.2 Fasnet

To date, Fasnet [Limb82] was the only local network proposed for the high speed dual bus topology, excepting the current research. Fasnet uses slotted busses and two independent implicit tokens for access control. A token identifies the beginning of a cycle in a bus and is recognized as a bit set in the slot header. Stations are synchronized at the bit level, and end stations are responsible for generating tokens and empty slots on the busses. Slots carry an

empty/busy bit and a token return bit in their headers. A station with a ready packet acquires the first empty slot following the detection of a token. Busy slots form a train on the bus, and an end station, upon detecting the end-of-train (EOT), sets the return bit on the first slot generated on the other bus. The return bit propagates to the other end station which then regenerates the token on the first bus. This scheme works independently for each bus, and a station can only transmit one packet per cycle per bus.

Other variations of this design have also been proposed in [Limb82], but all assume fixed size slots and same synchronization scheme. We believe that the requirement of modifying control bits "on the fly" inside synchronous slots may be a serious limitation to the use of Fasnet at gigabit rate.

1.4 DISSERTATION OUTLINE

Our research focuses on the development and performance evaluation of protocols for the high speed dual unidirectional bus topology. As seen in the previous section, from the two existing protocols for the dual bus topology only Fasnet is suitable to high data rates. Fasnet, however, is intended for networks of reduced length and small packets of fixed size [Limb82]. We consider this environment very restrictive to the development of applications intended to take full advantage of the high bandwidth available. One objective of this dissertation is to produce protocols able to adapt to a variety of traffic and network conditions.

For our protocols we assume that transmissions are asynchronous and packet size is variable. Analytical expressions for insertion delay and utilization are derived. Results for intermediate load are obtained by discrete event

simulation.

In Chapter 2 we describe and analyze two token-based protocols: U-Net and TDT-Net. U-Net circulates a token between end stations. To improve reliability a dynamic end station election mechanism is incorporated to the protocol, providing automatic recovery in case of end station failure. Transmissions are bidirectional and stations transmit synchronized by the token. Many implementations for the token are suggested. Because stations have finite reaction time, packets may be corrupted in the beginning and a preamble is required in each packet. TDT-Net uses the infra-structure of U-Net but transmissions are corruption free because stations are synchronized by minislots. We show that the minislot can be as small as the maximum reaction time among the stations. We show that these protocols achieve optimal performance for equally loaded traffic.

In Chapter 3 we discuss Buzz-Net which uses a hybrid random/token scheme to achieve utilization near 1 for a single transmitting station while performing optimally at light load. Buzz-Net utilizations are lower than those for U-Net or TDT-Net under equally loaded traffic. Transmissions are also bidirectional and a special buzz pattern is needed to control the channel.

In Chapter 4 we describe a pure random scheme called Rato. Rato uses a simple time-out delay to control the channel and transmissions are unidirectional. Because the control is so simple, hardware requirements for Rato implementation are minimum. Utilization has an asymptotic value of 0.50 and is independent of network span. Delay, however, is a function of the number of stations and the maximum packet transmission time.

The Token-Less family is introduced in Chapter 5. The beauty of these protocols is the simple channel control based solely on activity detection. There is no need for special patterns or packets. Four versions of different complexities are proposed: TLP-1,2,3 and 4. In particular, TLP-3 is shown to perform identically to U-Net. Best adaptive performances are obtained with TLP-4. TLP-4 employs a dynamic end station selection which permits adaptability to multipacket and unbalanced traffic. The protocol also behaves as a random scheme at light load. TLP-4 provides the best overall performance of all protocols and comes very close to optimal.

Chapter 6 is dedicated to a comprehensive comparative analysis among all proposed and existing protocols. Analytical expressions are used in utilization and insertion delay comparison at light and heavy load. Simulation provide results for intermediate load, and unbalanced and multipacket traffic.

An analytical approximation to the queueing delay in oscillating polling under chaining is obtained in Chapter 7. Oscillating polling models protocols such as TLP-3 and U-Net. Our analysis is restricted to the case of equally loaded and single packet traffic. Simulation results show that the approximation is very good when $\alpha = \tau/T \geq 5$. No previous results were available for oscillating polling under chaining.

To finalize our investigation of high speed LANs, we address the problem of building systems with large number of stations in Chapter 8. We propose solutions that include maximization of the number of optical couplers in the dual bus topology, use of hybrid passive star/bus topologies, and use of bridges and gateways. Part of this research is original.

CHAPTER 2

TOKEN PROTOCOLS

2.1 INTRODUCTION

In this chapter we propose and analyze two asynchronous token based protocols. The first protocol, U-Net, uses a new concept of bidirectional transmission coupled with bidirectional token synchronization on the dual unidirectional bus architecture. In U-Net collisions are nonexistent because stations transmit only synchronized by token detection. The reliability issue of having token regeneration attached to physical end stations is eliminated by performing end station election at initialization (or after a configuration change: stations leaving or joining the network.), and by requiring the end stations to remember their state and exchange a token.

Our second asynchronous token protocol, TDT-Net (Time Division Token Network), uses the same token regeneration and end station election procedures of U-Net. The difference resides in the way a station appends its packet. After a token detection stations wait for a corresponding synchronizing slot before transmitting. TDT-Net is an extension of the concept of minislots [Klei77] to dual unidirectional high speed bus architecture. This protocol has the advantage of avoiding the initial corruption of packets observed in U-Net, eliminating the need for an extensive packet preamble. Extra overhead, however, is necessary to control the transmissions. The maximum utilizations and delays

observed in TDT-Net and U-Net are approximately the same.

Of the two protocols that have been proposed for the dual unidirectional bus architecture, Fasnet [Limb82] is the only one suitable to high data rates (see Section 1.3.1.3). Fasnet is a synchronous slotted network with end stations (the right-most and the left-most stations) responsible for slot generation and cycle regeneration. The bit synchronization required at each station and the centralized control allocated to physical end stations degrade reliability and compromise robustness (see discussion in Chapter 1).

The latter two schemes have an advantage over Fasnet because they are able to establish a token passing round (after collision) without prior knowledge of the end stations. End stations are dynamically elected before each token passing round, clearly improving robustness and ability to withstand station failure.

Previous token protocols for a single bus [Frat81, Tsen82] used unidirectional token synchronization. These protocols can be modelled as a cyclic polling system. Cyclic polling has been studied by many authors [Konh74, Rubi81, Toba83]. Bidirectional token synchronization, however, produces an oscillating polling unsuitable for exact mathematical analysis when packets are transmitted one per polling instant [Ulug81]. In Chapter 7 we discuss the difficulties in analyzing the oscillating polling and present an approximate solution that can be used in evaluating protocols using bidirectional token synchronization (i.e. TLP-3, U-Net, TDT-Net, etc.) under equally loaded and single packet traffic.

In this chapter we derive performance expressions for light and heavy load. Simulation results are used in Chapter 6 for comparative analysis in mid-

dle range load.

2.2 U-NET PROTOCOL

U-Net (Unidirectional Network) is a local network designed for dual bus architecture. Briefly, we recall below some of the features of the dual bus architecture essential for the implementation of U-Net. Stations are connected to the busses via passive taps (see Fig. 1.4), each tap including a receiver and a transmitter. The receiver detects presence/absence of carrier. When carrier is present, the receiver attempts to acquire bit synchronization from the preamble. After acquisition, the receiver copies bus data into private memory. The transmitter sends a preamble followed by the data packet after it has received the go-ahead by the access protocol. If the station senses carrier coming from upstream while transmitting, it aborts its own transmission and tries again following the incoming data.

We assume a reaction delay of d seconds between the time a station senses end of carrier on the bus and the time it can start transmission on the same bus. Likewise, there is a d second delay between the sensing of carrier coming from an upstream station and the interruption of an ongoing transmission.

The above functions are common to all UBS interfaces. Actual UBS protocols differ from each other in the way they use these basic functions to provide access scheduling and synchronization.

2.2.1 THE ACCESS PROCEDURE

The U-Net protocol consists of two procedures. The first procedure, described in this section, defines access to the bus after the end stations have been elected and the token mode has been established. The second procedure, introduced in the next section, defines the election of end stations at network initialization and/or network configuration change.

The following describes the token mode of operation used in U-Net. The two end stations are defined as L (left) and R (right). Protocol operation can be viewed as a sequence of cycles. Each cycle is initiated by one end station, for example, R station. R sends a special bit pattern, called token, on the R-to-L bus. This token is followed by a data packet from R (if R has data to send).

Each station continuously monitors both busses for a token. Once the token is heard on a bus (henceforth referred to as the token bus), the station is allowed to transmit one packet on both busses. More precisely, immediately after hearing the token, the station begins transmitting the preamble on the token bus. If, after an interval d from the beginning of its transmission, the station does not hear conflict on the bus (conflict may occur if an upstream station on the token bus is also attempting to transmit), it proceeds transmitting the preamble on the token bus as well as on the reverse bus (i.e., the bus in the opposite direction). If conflict is detected (i.e., the station hears another preamble coming in from upstream while it is transmitting its own), the station aborts its transmission on the token bus and does not attempt to transmit on the reverse bus. The station restarts transmission after the oncoming packet has passed. This procedure is called *probing* the token bus.

On the token bus, packets are appended to the token in the same way that cars in a train are appended to a locomotive. Each station has the chance to transmit on the train, and can transmit at most one packet. Packets on the bus are separated by gaps of size d . On the reverse bus, a similar train is formed. However, packets are not preceded by a token; rather they are separated by larger gaps than the packets on the token bus. The size of the gap between two packets on the reverse bus is equal to twice the propagation delay between the two sending stations, plus $2d$. Fig. 2.1 shows the space-time diagram for a possible sequence of packets on the token bus and on the reverse bus. A snapshot of the system is also shown.

Another difference between the token bus and the reverse bus is that on the token bus the initial d seconds of the preamble may be damaged by conflicts. In fact, if the train carries N packets, the first K bits of the preamble (where $K = dC$, and $C =$ bus speed) in the first packet correspond to the superimposition of $N-1$ preambles. The preamble must be large enough to allow bit sync to be acquired despite initial garbage.

It is important to note that each packet transmission is heard by all stations exactly once. Assuming the R-to-L bus is the token bus (see Fig. 1.4), the packet transmitted by station i is received by station $i+1$, $i+2$, ... , and N on the token bus, and by station $i-1$, $i-2$, ..., 1 on the reverse bus. The transmission mode is implicitly a broadcast mode; specific knowledge of the destination station is unnecessary to properly route the packet.

The cycle terminates when the train terminates (i.e., when all the stations, including L and R, have had the opportunity to send their packets). The L station detects the end of the train from the absence of carrier for more than

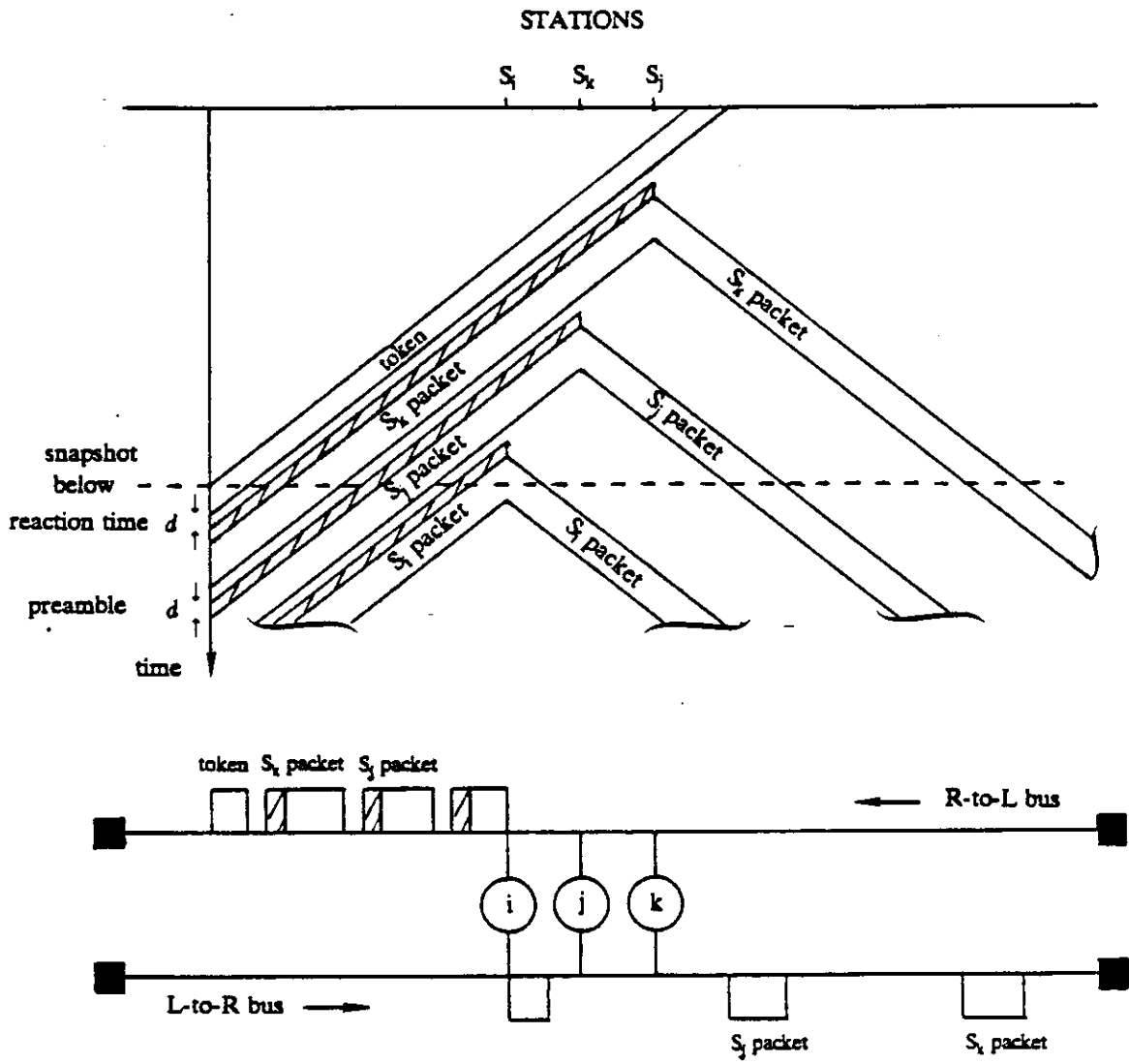


Fig. 2.1 - Space-Time Diagram and Snapshot.

2d seconds at the end of a packet (or token). After detecting the end of the train and (possibly) transmitting its own packet, the L station declares the cycle closed and starts a new cycle in the reverse direction by injecting a token in the reverse bus, which becomes the new token bus. The operation is the same with the roles of token bus and reverse bus interchanged.

Tokens can be implemented as bursts of carrier smaller than the minimum packet transmission time but large enough to be reliably detected ($\geq d$). A carrier burst is simple to generate, easy to be detected, and cannot be corrupted by errors on the channel. If the hardware is carefully designed, burst counting can be reliably implemented and a sequence of n bursts can be recognized. If this sequence represents an n token, priority traffic can be directly implemented in the low level access protocol by generating cycles with different tokens and allowing only the traffic corresponding to the token type to be transmitted in the cycle. Because stations always defer to incoming traffic, the bursts are not destroyed and maintain their integrity along the cable. Tokens may also be implemented as special packets. This implementation requires more sophisticated generation and detection, and is prone to eventual errors in the channel. Because the token is a packet that may contain generic information, this implementation leaves margin to future developments.

2.2.2 END STATION ELECTION PROCEDURE

U-Net is equipped with a dynamic procedure for electing end-stations. This procedure provides automatic recovery from station failure and from token loss, without operator intervention. It also permits smooth insertion of new stations in the system.

R is defined as the round trip propagation delay on the fiber cable plus twice the station reaction time. T_{MAX} is the maximum size packet transmission time. t_0 is the time required by an end station to "turn around" the token (read it from one bus and inject it onto the other bus).

Next, some observations. During normal token mode operation there are short gaps between packets within each train, and larger gaps between trains. The distance between gaps is $\leq T_{MAX}$, by definition. If a continuous data stream of duration $> T_{MAX}$ is detected, it is interpreted as an anomaly. This property is exploited in the election procedure. As a second observation, the maximum duration of a silence gap at a station (the time during which both busses are sensed idle) during token mode operation is $R + t_0$. This worst case silence happens when the token is circulating with no packets being appended. If the train is not empty then the end station may process the token regeneration while packets are being received. Therefore, the token regeneration takes less than t_0 and the silence gap is less than $R + t_0$. A larger silence gap denotes an abnormal situation (e.g., a failure).

The following describes the end station election procedure. During this procedure each station moves through the states shown in Fig. 2.2.

During normal operation each established (as opposed to new entering) station is found in the **token mode** state. Operation in this state was described in Section 2.2.1. From this state a station moves to the **buzz mode** state if it observes a silence gap $\geq R + t_0$, or senses continuous signal for an interval $> T_{MAX}$.

STATE DIAGRAM

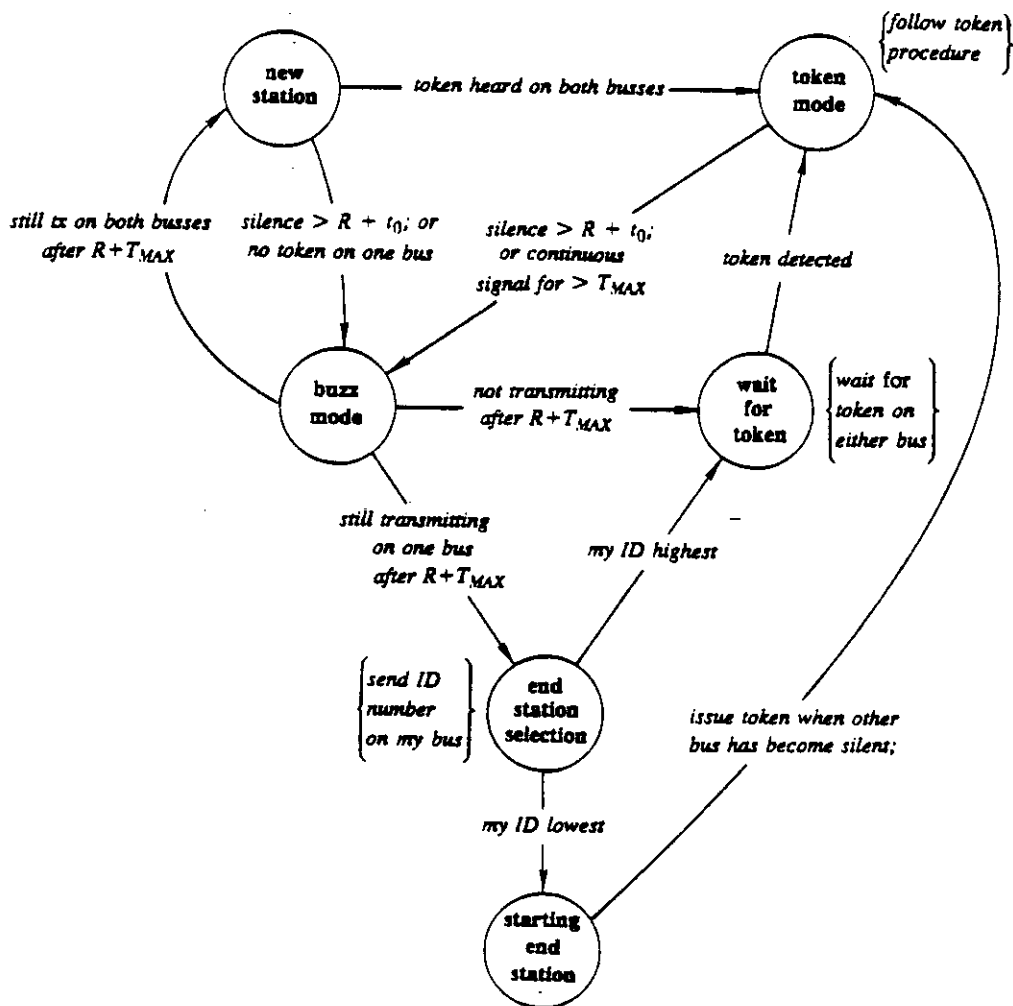


Fig. 2.2 - U-Net End Station Election State Diagram.

In **buzz mode** a station issues a buzz tone on both busses. As a possible implementation, this buzz tone could consist of a preamble repeated continuously without gaps. During **buzz mode** a station defers to upstream stations by aborting its buzz tone when a buzz tone arrives from upstream.

After an interval $R + T_{MAX}$ from the time the first station entered **buzz mode**, all stations are necessarily in **buzz mode** (a similar fact is proven in Appendix 5.1). At this point, a station can be in one of three possible conditions:

- (1) Deferred on both busses. In this case, the station is an intermediate station (i.e. not an end station.) It moves to the **wait for token** state. In this state, the station remains silent, awaiting for the token.
- (2) Still transmitting on one bus (and has deferred on the other because a busy tone was detected or the bus is busy). The station is an end station and moves to the **end station selection** state, where one of the two end stations is selected to start the token cycle.
- (3) Transmitting on both busses. This implies that there is only one station on the bus! The station moves to the **new station** state (to be defined later).

In **end station selection** the newly elected end stations must decide which has the lowest ID and thus starts first. This decision can be fixed based on the topology (physical ID), or can be made by the stations based on some logical ID. In a logical decision, each station replaces the buzz tone with a pattern consisting of its ID number repeated over and over. The elected end stations compare ID numbers. The high ID number station moves to **wait for token**;

the low ID number station moves to the **starting end station** state, waiting for the reverse bus to become idle. It then issues a token and moves to **token mode**. In a fixed decision, one of the end stations is selected a priori as the token regenerator (the station still transmitting on the R-to-L bus is the rightmost whereas the one still transmitting on the L-to-R bus is the leftmost). Upon entering **end station selection**, the selected station assumes it has the lowest ID number while the other end station behaves as having the highest ID. Thereafter, both end stations perform as above.

Upon hearing the token, all other stations move from **wait for token** to **token mode**.

A new entering station finds itself initially in the **new station** state. From this state, it must detect the token on both busses before moving to **token mode**. If a token is heard twice on the same bus, but not on the other bus, the station is the new end station. Thus, the station moves to **buzz mode** to trigger a new election. Likewise, the station moves to **buzz mode** if a silence gap $> R + t_0$ is detected. This gap may occur at system initialization.

The election procedure may appear somewhat elaborate, but it is quite efficient. The whole procedure requires approximately $3R + T_{MAX} + t_0$ to recover from failures. Typically, this is in the order of fractions of a millisecond for channel speeds over 100 Mbps. The procedure is robust to any sort of failure (the system can even detect and recover from failures occurring during the recovery procedure). even failures occurring during the recovery procedure are detected and recovered from.

2.3 TDT-NET

In TDT-Net, token regeneration and initialization procedures are performed as in U-Net. Similarly, stations synchronize with tokens on both channels, and transmissions are bidirectional and of variable length. Stations are assigned numbers according to their physical location in the network. In this way, station $N-1$ knows that it is the second to transmit on the R-to-L bus and the $N-1$ th to transmit on the L-to-R bus.

The token synchronizes the start of a transmission round. Each station, upon detection of end of token, starts its own slot schedule in a completely distributed fashion. The d_s seconds following *EOC* constitute the first slot. If the station assigned to that slot (the end station itself) does not have a packet to transmit, the station leaves the slot intact (silence for d_s seconds). All other stations detect this empty slot and realize that no transmission from the slot owner occurred in that round. If no packets are transmitted, each succeeding d_s silent period is considered a slot and assigned correspondingly to succeeding downstream stations. An empty round corresponds to a token followed by N empty d_s slots.

However, if a station has a backlogged packet, the station transmits the packet starting d_e from the beginning of the corresponding synchronizing (or reservation) slot. In the next section we discuss the setting of parameters d_e and d_s and show that in fact d_e should be set to 0 to improve performance, if some extra precaution is taken.

If a station detects a transmission on a synchronizing slot, the slot schedule is restarted only after *EOC* is detected. In this way, each transmission enables slot resynchronization for every downstream station, allowing a great safety margin in the design of interface clocks. Furthermore, collisions are avoided and the preamble has to account only for clock synchronization.

If we observe events on the bus, the first slot logically occurs d seconds after the end of token. After a transmission, synchronizing slots logically restart d seconds after *EOC* occurs on the bus. The gaps of d seconds occur because of the reaction time of stations.

2.3.1 PARAMETERS d_s AND d_e

Network utilization is improved by minimizing the overhead caused by synchronizing slots d_s . In this section we calculate lower bounds for d_s and d_e to tolerate deviations in the reaction time of stations and drift of clock frequency.

There is no central control and stations identify slot boundaries independently based on the detection of *EOC* and on measures of its internal clock. Therefore, each station carries its own view of the slotted schedule. The network is out-of-sync when a station detects a transmission from another station in the synchronizing slot (as computed by itself) reserved for a third station. Improper selection of parameters d_s and d_e may cause out-of-sync situations when deviations in clock frequency and delays in logical circuits accumulate unfavorably. However, if the maximum deviations are known, d_s and d_e can be set properly to avoid loss of synchronism under the worst case condition.

To analyze the effects of time dependencies, we consider an absolute time reference starting with token *EOC* on the bus. Because stations have clocks and internal circuit delays slightly different from each other, slots are indexed by their generating station index, when necessary. Assume $d_{\max} = \max \{d_i \mid i=1, N\}$ and $d_{\min} = \min \{d_i \mid i=1, N\}$, where d_i is the reaction time of station i . Let $\Delta d = d_{\max} - d_{\min}$. Similarly, define $d_{s,\max}$, $d_{s,\min}$, and Δd_s .

Let us define:

boc_i - time when *BOC* due to station i occurs in the channel.

$boc_i(j)$ - time when *BOC* due to station i is sensed at station j .

$BOS_i(j)$ - beginning of synchronizing slot i at station j .

$EOS_i(j)$ - end of synchronizing slot i at station j .

The conditions for synchronism are:

$$BOS_i(j) = < boc_i(j), \text{ and } boc_i(j) = < EOS_i(j), \text{ for all possible pairs } i, j .$$

The conditions above guarantee that the beginning of any transmission is detected at another downstream station during the proper synchronization slot.

Without loss of generality, assume that station i is the first station to transmit after the token is generated at station 1. Consider $1 = < i < j = < N$. Thus:

$$boc_i = (i-1)d_{s,i} + d_i + d_e ,$$

$$boc_i(j) = (i-1)d_{s,j} + d_i + d_e + d_j/2 ,$$

$$BOS_i(j) = (i-1)d_{s,j} + d_j/2 ,$$

$$EOS_i(j) = id_{s,j} + d_j/2 .$$

Applying the first inequality condition for synchronism we obtain:

$$d_c \Rightarrow (i-1)(d_{s,j} - d_{s,i}) - d_i , d_c > 0 .$$

This inequality gives us a lower bound on d_c . The worst case for d_c occurs for $i = N-1$, $j = N$ and adequate combination of deviations. The lower bound d_{cmin} is given by:

$$d_{cmin} = (N-2)\Delta d_s - d_{min} , d_{cmin} \Rightarrow 0 .$$

If the minimum reaction time is greater than the total clock drift, d_c can be set to 0.

The second inequality becomes:

$$d_{s,j} \Rightarrow d_c + d_i + (i-1)(d_{s,i} - d_{s,j}) .$$

Therefore, the lower bound $d_{s,min}$ is:

$$d_{s,min} = d_{max} + d_c + (N-2)\Delta d_s .$$

Under most conditions, clock frequencies are very stable making clock deviations negligible. However, circuit delays always exist and are necessary in practical implementations. Therefore, we expect $d_{min} \gg N\Delta d_s$. Consequently, we consider $d_c = 0$ and $d_s \cong d_{max}$.

2.4 PERFORMANCE ANALYSIS

2.5 U-NET RESULTS

For U-Net, in addition to the assumptions in Section 1.2.2.2, we further assume that token regeneration is hardware implemented and occurs within a negligible delay after the end of train is detected. Therefore, an end station regenerates the token $2d$ seconds after the *EOC* of the last packet in the train is sensed in its taps. Without loss of generality, we assume that the same delay in token regeneration occurs when the end station is the last to transmit at the end of round. This assumption guarantees a minimum $2d$ interval between the token and preceding packet on the same bus. We also assume that token and packet generated by an end station in the same bus are separated by an interval d . This assumption guarantees token integrity when a burst of carrier is used as a token implementation.

2.5.1 DELAY PERFORMANCE

2.5.1.1 LIGHT LOAD

Before deriving the expressions for *IDL*, first note the following. Assume that at light load a packet is generated at some random point in time. Also, assume that the possible transmission instants are separated by x and y seconds, as shown in the following diagram:

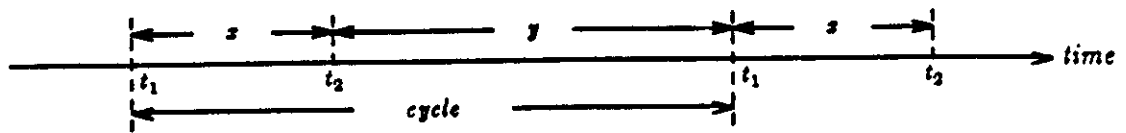


Fig. 2.3 - Transmission instants in a cycle.

In a cycle:

$$E[\text{delay for an arrival in } x] = x/2$$

$$E[\text{delay for an arrival in } y] = y/2$$

$$P\{\text{arrival occurs in } x\} = x/(x+y)$$

$$P\{\text{arrival occurs in } y\} = y/(x+y)$$

Hence:

$$E[\text{delay}] = \frac{x^2 + y^2}{2(x+y)}$$

Consider station i . The idle cycles seen by i are as follows:

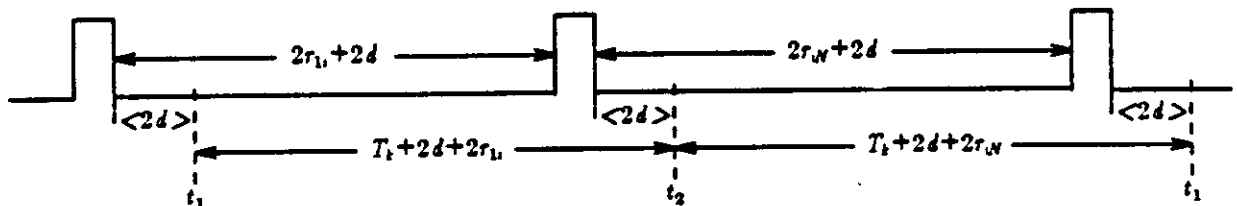


Fig. 2.4 - Idle cycles seen by station i in U-Net.

Consequently:

$$IDL_i = \frac{(2\tau_{1i} + 2d + T_k)^2 + (2\tau_{iN} + 2d + T_k)^2}{2(2\tau + 4d + 2T_k)}$$

Assuming a symmetric topology we have $\tau_{1i} = (i-1)a$ and $\tau_{iN} = (N-i)a$.

Averaging over all stations we get:

$$IDL = \frac{1}{N} \sum_{i=1}^N IDL_i = \frac{1}{\tau + b} \left[\frac{\tau^2}{3} \left(2 + \frac{1}{N-1} \right) + \tau b + \frac{b^2}{2} \right],$$

where $b = T_k + 2d$ can be interpreted as the minimum delay in inverting rounds. For the usual case in high speed busses where $\tau \gg b$ we simplify the above expression to yield:

$$IDL = \frac{\tau}{3} \left[2 + \frac{1}{N-1} \right].$$

When $N \gg 1$, IDL is minimum and equal to $2\tau/3$. The worst case for IDL occurs for $N = 2$ where, under the simplifying assumptions, we get $IDL = \tau$.

2.5.1.2 HEAVY LOAD

At heavy load active stations transmit twice every cycle. Due to the network topology, the closer a station is to the end stations, the closer the transmission instants are. Considering a station i (as in the case of light load), the transmission instants are located in a cycle as follows:

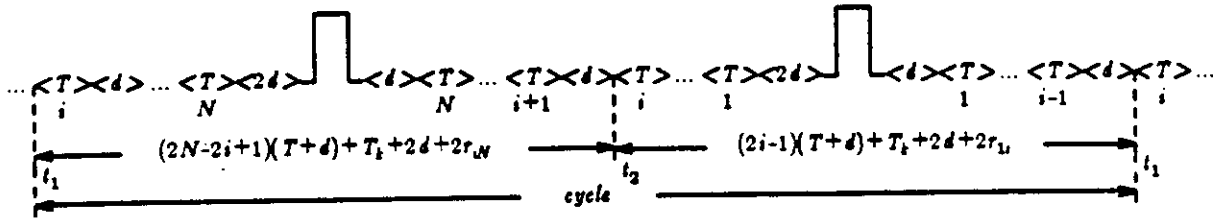


Fig. 2.5 - Transmission instants for U-Net at heavy load.

Averaging over all packets, IDH_i is clearly equal to $cycle/2$. Consequently, $IDH = IDH_i$ and, when i stations are active, $IDH(i)$ is given by:

$$IDH(i) = \tau + T_k + 2d + i(T+d) - T.$$

$IDH(i)$ is bound and increases linearly with i . It is also clear that the maximum insertion delay (MID) is given by $IDH(N)$.

2.5.2 UTILIZATION

Knowing the expression for the cycle at heavy load, the bus utilization when i stations are active is immediately calculated as follows:

$$S(i) = \frac{2i T_r}{cycle} = \frac{i T_r}{\tau + 2d + T_k + i(T+d)}.$$

The maximum utilization S is given by $S(N)$. For the usual case where $\tau \gg 2d + T_k$ and $T \gg d$ we get:

$$S(i) = \frac{i T_r}{\tau + iT}, \text{ and}$$

$$S = \frac{NT_r}{\tau + NT} = \frac{NT_r}{(\tau + NT_p) + NT_r} \quad (2.2)$$

As we see, even for small T_r , we may still have considerable capacity, especially when $NT_r \gg \tau + NT_p$. In that case, the performance of the system depends upon the percentage of preamble needed to handle collision and locking of the receiver clock. For $T_r \gg T_p$, and assuming $\alpha = \tau/T$, we have:

$$S(N) = \frac{1}{1 + \frac{\alpha}{N}} \quad (2.1)$$

Equation (2.1) will be used to calculate U-Net maximum utilization in the comparative analysis in Chapter 6.

2.6 TDT-NET RESULTS

For TDT-Net, in addition to the assumptions in Section 1.2.2.2 we assume that end stations regenerate tokens d seconds after the end of their transmission or synchronizing slot in the past round. In the new round, the end station that has originated the token observes a delay d before transmitting a backlog packet. If we look at the events on the bus, a token is, in the worst case, surrounded by silence intervals of size d . Consequently, token integrity is preserved and reliable token detection occurs even when the token is a simple burst of carrier.

2.6.1 DELAY PERFORMANCE

Similar to U-Net, delay performance in this section is measured in terms of insertion delay (ID), defined in Session 1.2.2.2. Analytical expressions for ID at light (IDL) and heavy load (IDH) are derived.

2.6.1.1 LIGHT LOAD

In order to evaluate IDL , the transmission instants for station i at light load are obtained from the following diagram:

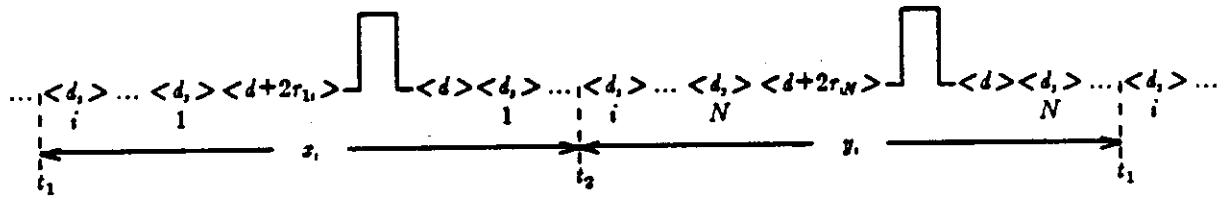


Fig. 2.6 - Transmission instants for TDT-Net at light load.

We have:

$$x_i = T_k + 2d + (2i-1)d_s + 2\tau_{1i} ,$$

$$y_i = T_k + 2d + (2N-2i+1)ds + 2\tau_{iN} .$$

Proceeding as in U-net and using $A = d_s + a$ and $B = 2d + d_s + T_k$, we get:

$$IDL = \frac{1}{A(N-1)} + B \left[\frac{(N-1)(2N-1)A^2}{3} + (N-1)AB + \frac{B^2}{2} \right] .$$

For very fast logic implementation of the station interface, especially when integrated optics is used, it is reasonable to assume $\tau \gg (N-1)d_s$. Therefore, $a \gg d_s$ and $A \cong \tau$. Under the above assumption, we can rewrite IDL as:

$$IDL = \frac{1}{\tau + b} \left[\frac{\tau^2}{3} \left(2 + \frac{1}{N-1} \right) + \tau b + \frac{b^2}{2} \right],$$

If we compare this expression with the one derived for U-Net we observe that they are identical if we exchange B for b . The reason is that in the latter expression we are ignoring the synchronization slots what leads both systems to very similar performance.

Following the same steps performed for U-Net, when $\tau \gg B$, we simplify the above expression to yield:

$$IDL = \frac{\tau}{3} \left[2 + \frac{1}{N-1} \right].$$

Repeating the observations in U-Net, the worst case for IDL , assuming a constant D , occurs for $N=2$. When $N \gg 1$, IDL is minimum and equal to $2\tau/3$. The worst case for IDL occurs for $N = 2$ where, under the simplifying assumptions, we get $IDL = \tau$.

2.6.1.2 HEAVY LOAD

At heavy load, the transmission instants in TDT-Net are given by the diagram in Fig. 2.7. From Fig.2.7 we get:

$$cycle = 2T_k + 2(\tau + d) + 2N(T_r + d).$$

Hence,

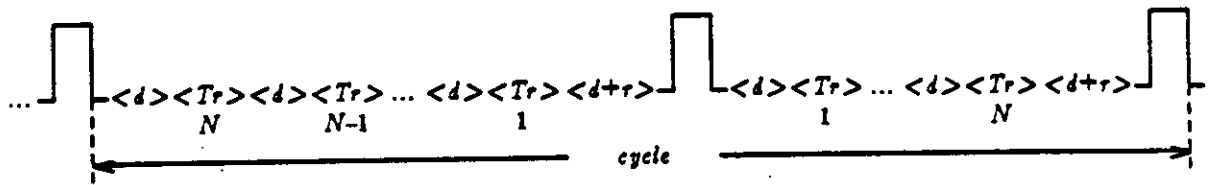


Fig. 2.7 - Transmission instants at heavy load in TDT-Net.

$$IDH(N) = MID = \tau + T_h + d + N(T_r + d) - T_r.$$

Comparing this result with the one obtained for U-NET, we can identify a slight improvement on IDH due to the absence of collisions and preamble overhead.

2.6.2 UTILIZATION

For throughput derivation in TDT-Net, we consider the following diagram which describes a train when stations i and j transmit:

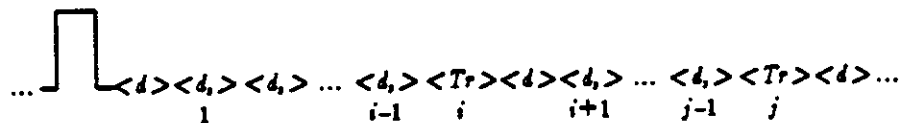


Fig. 2.8 - Train of transmissions in TDT-Net.

As we see, if a station does not transmit it contributes with a slot of size d_i for the cycle, otherwise it transmits and contributes with $Tr+d$ seconds for the cycle. Thus:

$$S(i) = \frac{iT_r}{\tau + 2d + T_k + i(T_r + d) + (N-i)d_s}.$$

The maximum utilization S is given by $S(N)$. For the usual case where $\tau \gg d + T_k$ and $T_r \gg d$ we get:

$$S(i) = \frac{iT_r}{\tau + iT_r + (N-i)d_s}, \text{ and}$$

$$S = \frac{NT_r}{\tau + NT_r}. \quad (2.2)$$

As we see, even for small T_r , we may still have considerable capacity, especially when $NT_r \gg \tau$. In this case, the capacity approaches 1 as NT_r increases. We observe that (2.2) and (2.1) are equal for $T \cong T_r$.

CHAPTER 3

BUZZ-NET

3.1 INTRODUCTION

In this chapter, we describe and analyze Buzz-net, a hybrid random access/virtual token protocol for the dual bus topology. In principle, Buzz-net behaves as a random access network at light load. If there is an upsurge of traffic, all stations switch from random access to controlled access mode. The synchronizing event for this transition is a special "buzz" pattern emitted on the bus (hence the name of Buzz-net). In the controlled mode all backlogged stations alternate in transmitting one packet. When the controlled cycle is completed, random access mode is resumed.

The main goal in the design of Buzz-net was to develop a local network that could yield high throughput efficiency, provide bounded insertion delay, operate in fiber optic environment, run under totally distributed control, survive to processor failures, and allow automatic station insertions/removals.

Within the family of unidirectional bus architectures we can distinguish two classes: token (or virtual token) schemes, and random access schemes. In the first class are Express-net [Frat81], D-net [Tsen82], Fasnet [Limb82], and U-net, described in Chapter 2. All of these token schemes can provide good performance in a local fiber optics network environment. However, each has some drawbacks. For example, the "folded" topology in Express-net and D-net causes

higher attenuation than the dual bus topology since the signal must traverse twice as many taps. In D-net the network fails if the token generator(s) fails. Fasnets has similar problems with failures of end stations. In all schemes a token latency proportional to the end to end propagation delay is suffered at packet insertion. This delay translates into throughput degradation if only one station has data to send and can transmit only one packet per token.

In the random access family the most popular scheme is CSMA-CD [Metc76]. Although this scheme was initially developed for bidirectional busses, it can be extended to dual unidirectional busses. CSMA-CD eliminates token latency and provides high throughput to a single sending station. However, it shows throughput degradation, unbounded delays, and capture problems in heavy load multistation situations.

Because of the above trade-offs, the "best of all worlds" appears to be a hybrid random access/token architecture. One such architecture, MAP, was proposed in [Mars81]. That architecture eliminated the latency problem, but did not resolve the single station throughput problem. Furthermore, the folded topology still caused an undesirable extra attenuation in the signal. Recently, a similar approach to Buzz-net has been proposed but it requires messages greater than the end-to-end propagation delay for reliable collision detection [Taka83]. Meeting this requirement leads to performance degradation as packet padding becomes necessary when the transmission speed increases.

Buzz-net, described below, appears to be a more viable hybrid architecture in that it combines many of the advantages of token and random access schemes without suffering of their limitations.

3.2 PRINCIPLES OF OPERATION

The network can operate in either of two states: random access or controlled access. In the *random access* mode each station transmits ready packets on both busses as soon as it senses them free. When a backlog builds up (and, therefore, interference starts occurring) one or more stations start *buzzing* the busses. The buzz causes the mode to switch from random access to *controlled access*. In controlled access mode all the backlogged stations are allowed to transmit one packet each without collisions. After the controlled access cycle terminates, random access mode is resumed.

The following general assumptions are made:

- (1) Once a station has completed the transmission of a packet on a bus, this packet will be heard correctly by all downstream stations on that bus. That is, a station engaged in transmission always *defers* to an upstream transmission by aborting its packet. The upstream transmission is allowed to proceed intact. The underlying assumption is that the beginning-of-packet flag cannot be replicated within the packet data. This way, a new packet can be detected even when this packet is immediately preceded by another (truncated) packet. Flags can be implemented as reserved bit patterns (in which case bit stuffing is needed to preserve data transparency) or as code violations during transmission. This assumption, although not strictly required for the proper operation of the Buzz-net protocol, is introduced here to simplify the presentation.
- (2) The buzz signal is a signal (or event) clearly distinguishable from regular packet flow. For example, the buzz could consist of a preamble string

longer than the standard preamble used for data packets. Other possibilities include the use of short bursts (shorter than the minimum packet length) or the use of interpacket gap fillers. As we shall see, the protocol can be defined independently of the buzz implementation: only a few timing parameters are affected. In Section 3.4 we compare the various buzz implementations.

3.3 THE ALGORITHM

Initially, a station starts in the **Idle** state of the random access mode (see Fig. 3.1). When a packet arrives, the station moves to the **Backlogged** state. From this state, transmission of the packet is attempted in random access mode as follows:

- (a) If both busses are sensed idle, the station moves to **Random Access Transmission** state. In this state, packet transmission immediately begins on both busses (it is assumed that the sender is unaware of the relative position of the destination on the bus).
- (b) If one bus is idle and the other is busy, the station moves to **Wait for EOC** state, where it waits for *EOC* (End-of-Carrier) on the busy bus.
- (c) If both busses are sensed busy or a buzz pattern is sensed, the station moves to the **Buzz-I** state, which is part of the controlled access procedure.

In **Random Access Transmission** the station proceeds to transmit on both busses. If, while transmitting, the station is interfered by an upstream station (that is, it hears a *BOC*, Beginning-of-Carrier, on one of the busses) it

aborts its transmission and moves to **Buzz-I**. The upstream transmission is allowed to proceed intact. If the transmission is successfully completed, the station moves to **Idle**.

In **Wait for EOC**, when *EOC* is sensed, the station moves to **Random Access Transmission**. If, while in **Wait for EOC**, the station senses a buzz pattern or it senses both busses busy, it moves to **Buzz-I**.

While in the random access mode a station with several packets ready for transmission may attempt to send them all in a single train, cycling between **Backlogged** and **Random Access Transmission** states, thus capturing the channel and locking out the other stations. To avoid capture, a minimum inter-packet gap must be observed between any consecutive packet transmissions. This minimum gap, on the order of a station reaction time interval (the delay between detection of *EOC* on the bus and the issue of *BOC* by the station), allows downstream stations in **Wait for EOC** to detect *EOC* inside a train and, upon collision, force the system to controlled access mode, thus breaking capture.

If the network is lightly loaded, stations tend to remain in the random access mode, cycling between **Idle**, **Backlogged**, and **Random Access Transmission** states. When the load builds up and interference occurs, all backlogged stations move to the controlled access mode of operation through the **Buzz-I** (see Fig. 3.1).

In **Buzz-I** a station transmits the buzz pattern on both busses (deferring, of course, to upstream transmissions) for R seconds, where $R =$ round trip propagation delay (2τ) plus twice the station reaction time ($2d$) plus twice the time

to recognize a buzz (2φ). Because of deferrals, a station in the buzz state may actually buzz the busses only intermittently, or it may not buzz them at all. After R seconds, the station moves to **Buzz-II** state.

At the end of the **Buzz-I** phase a station either senses a buzz or it senses silence on the Left-to-Right bus (see Appendix 1). In the latter case, the station knows that it is the leftmost backlogged station. As such, the station is responsible for initiating the controlled access cycle described below. It cannot initiate the cycle, however, until buzzing has ceased also on the Right-to-Left bus. In **Buzz-II** the station buzzes only the Left-to-Right bus, deferring as usual to upstream stations, until it hears no more buzzing on either bus. At this point, the station moves to **Controlled Access Transmission**. The intermediate **Buzz-II** state guarantees that the leftmost (and *only* the leftmost) station starts the controlled access cycle when all the Right-to-Left buzzing has ceased.

In **Controlled Access Transmission** each station is allowed to transmit its backlogged packet and move to **Hold** state thereafter. Controlled mode transmission is carried out much the same as in token networks, except that the Left-to-Right bus must be probed before transmission. A station waits for the Left-to-Right bus to become free, then probes this bus by starting transmission of the preamble. If the station does not hear upstream interference within a reaction time interval, it then proceeds to transmit the remaining preamble also on the Right-to-Left bus, followed by the data packet. If interference is sensed, the station aborts transmission and retries when the Left-to-Right bus is free again (after EOC is sensed).

Clearly, in the controlled access mode, a "train" of packets is formed from left to right, and backlogged stations are allowed to append their packets to the train in a left to right order. At the conclusion, all stations are in the **Hold** state.

A station remains in **Hold** until it detects a silence interval of $R1$ seconds on both busses, where $R1$ is equal to $2\tau + 2d$. $R1$ guarantees that a station leaves **Hold** to move to **Idle** only after the controlled access round has been completed but before any new transmission can occur. This property is important for fairness. If a station were allowed to enter **Idle** before all backlogged stations had transmitted their packets, it could attempt to transmit a new packet in random mode, causing interference and forcing the network into buzz mode, thus getting a second chance in the ensuing controlled cycle. The longest gap between two subsequent packets clearly occurs when the two backlogged stations engaged in the cycle are at the left and right end of the bus, respectively. First, the left station transmits its packet, then $2\tau + 2d$ seconds must elapse before the station detects (on the R-to-L bus) the packet from the right end station. Therefore, it is impossible for a station to move to **Idle** before the cycle is completed.

Small inaccuracies in measuring $R1$ may permit one station to resume random access mode early and keep the other stations in **Hold**. Although the stations in **Hold** eventually time out, this unfair behavior may be undesirable in practical implementations. To compensate for clock deviations, stations in **Idle** should wait Δt seconds before resuming the random access mode. The Δt safety interval should be larger than the maximum deviation in measuring $R1$ among all stations. If it is necessary, an additional state may be included between

Hold and **Idle** to enforce delay Δt .

A new entering station may attempt to transmit in random access mode although the bus is operating in controlled mode. If a collision occurs, the new station starts buzzing. To avoid deadlocks we require each station in **Controlled Access Transmission** to move back to **Buzz-I** upon hearing a buzz. However, the new entering station may capture the channel preventing the remaining stations from leaving the controlled access mode. Time-out T_0 from **Controlled Access Transmission** and **Hold** to **Idle** avoids the lock-up effect. A more detailed description of the recovery procedures when a new station joins the network is given in Section 3.5.

3.4 BUZZ SIGNAL IMPLEMENTATIONS

The buzz signal is a signal (or event) clearly distinguishable from regular packet flow. If the preamble pattern is uniquely distinguishable even when embedded in other data, then a simple buzz implementation consists of sending a prolonged preamble pattern. The uniqueness of the preamble pattern precludes its use within the data field of a packet. To maintain data transparency (and allow transmission of random data) bit stuffing must be used. If the preamble is a $\{0,1,0,1,\dots\}$ sequence N bit long, a 1 must be inserted at the transmitter after each $\{0,1,0,1,\dots\}$ sequence N bit long which occurs in the body of the packet. The extra 1 is later removed by the receiver.

An alternative buzz implementation which does not require bit stuffing consists of enforcing a minimum gap ΔT between any two consecutive packets on the bus. ΔT is large enough to allow a station in buzz mode to fill the gap with a burst of (arbitrary) data, which downstream stations can later detect

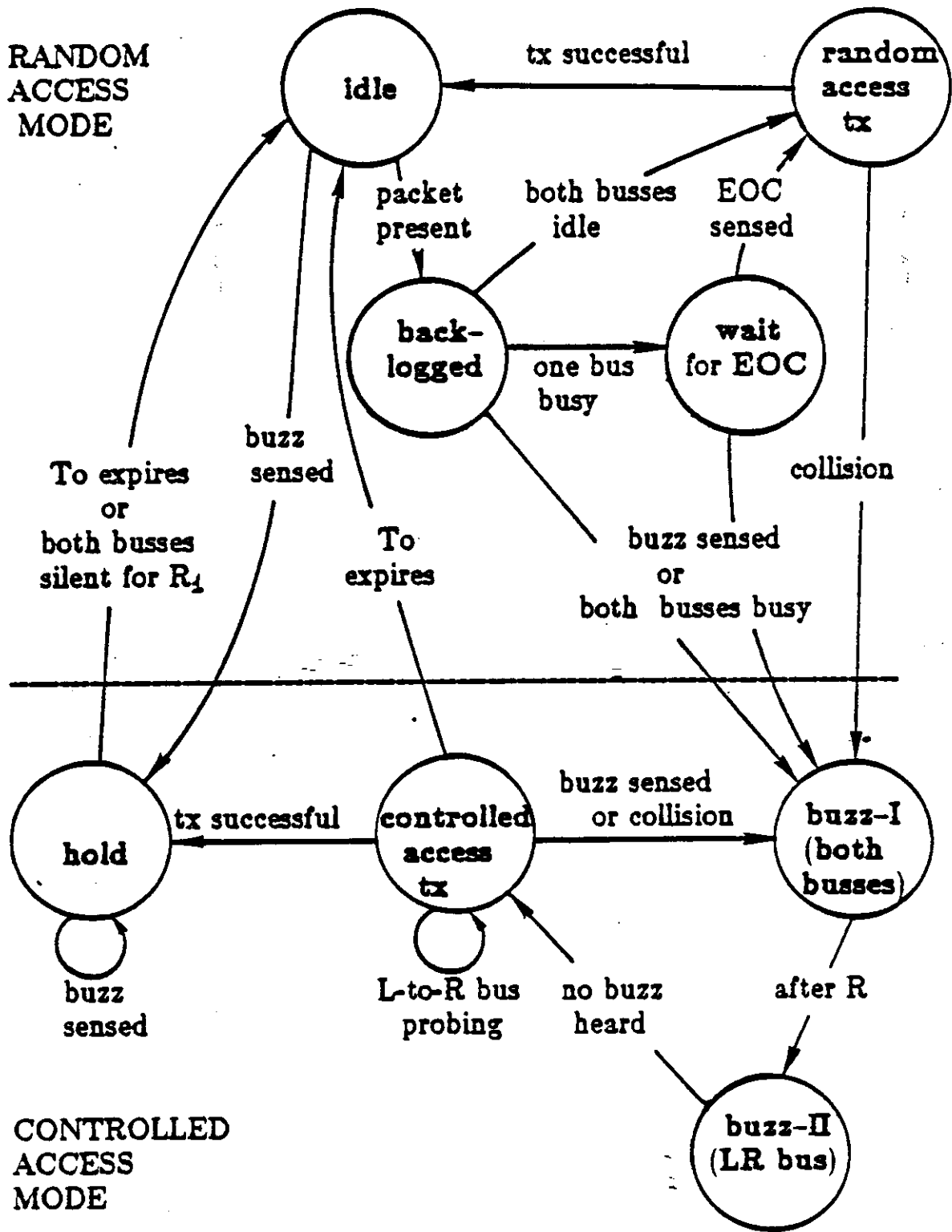


Fig. 3.1 - Buzz-net state diagram.

(i.e., ΔT must be larger than station reaction time). Furthermore, it is assumed that there is a maximum packet transmission time $\leq T_{max}$. Under these assumptions a station may buzz the network by filling interpacket gaps, and, when the bus becomes free, by sending an uninterrupted arbitrary data pattern lasting longer than T_{max} . A station recognizes the bus condition when it measures more than T_{max} seconds between two gaps $\geq \Delta T$ on any of the busses.

Yet another buzz implementation consists of sending short bursts of unmodulated carrier where the length of a burst is less than the smallest packet size, but large enough to be safely detected by a station. More precisely, a station buzzes the network by sending one (or more, for reliability purposes) burst(s) on both busses. A station recognizes the buzz condition when it detects on either bus the presence of a burst shorter than the minimum packet length. This scheme provides a faster detection of the buzz signal than previous schemes.

In general any method which permits some form of out-of-band signalling is a feasible buzzing method. The best method will most likely depend on interface implementation considerations, and may vary from application to application.

If the preamble cannot be distinguished when it is embedded in a packet, the first buzz implementation cannot be used. Furthermore, some minor changes are required in the basic Buzz-net protocol described in Section 3.3, since we can no longer assume that a packet transmitted without interference is successfully received by all downstream stations. After a successful transmission (either in random access or in controlled access mode) a copy of the packet must be saved for a round trip time R . If no buzz signal is heard within R seconds,

the copy is discarded. Otherwise, it is scheduled for retransmission. Duplication and out of sequencing are possible in this mode of operation, and must be eliminated by higher level protocols.

3.5 NEW STATIONS JOINING THE NETWORK

A newly active station may join the network at any time. No extra precautions are necessary if the new station starts the access algorithm from Idle. In some cases the joining process occurs transparently. In other situations, activity of the new station forces a transient phase which adds delay to the transmissions in progress. Whatever the case, the new station does not cause any permanent disruption of the network traffic, and the access algorithm automatically absorbs the external interference.

To facilitate our discussion, we name the "present" leftmost colliding station L and the newly active station A . The actual identity of L may vary depending on which event we consider to be the timing event. For example, it is possible that when A comes alive the present identity of L is station i , but by the time A transmission hits L , L may have changed to station j , with $j > i$. This change is likely because station i has finished its transmission and moved to Hold when the transmission from A hits its taps. In another situation, if L has not initiated the controlled access cycle, then we are actually considering the initial leftmost station. In this case no colliding station has transmitted any successful packet.

We define *controlled access mode delay* as the time during controlled access mode in which the busses are not used for packet transmission.

To better understand the joining process, we analyze the following possible states which a newly active station may find in the network when it comes alive:

- (a) *The network is operating in random access mode.*

In this situation, A is absorbed transparently.

- (b) *The network is operating under controlled access mode and*

*A detects a buzz signal. A , upon detecting the buzz, moves to **Buzz-I**.*

We identify two subcases:

- (b1) *A buzz is detected by L before L starts the controlled access cycle.*

No forced transitions occur due to A buzz. Only a maximum extra delay of R seconds is added to the controlled access mode delay, in the worst case (L is station 1 and A is station N).

- (b2) *A buzz collides with transmission by L .*

L moves to **Buzz-I**. Stations located between A and L stay in **Buzz-II** until the end of the new buzzing phase. Stations located on the right of $\max(A, L)$, upon sensing the buzz that follows the interrupted transmission by L , move back to **Buzz-II** and participate in the new buzzing phase. The new buzzing phase takes at most another $2R$ seconds in the worst case (stations 1 and N participating in the new buzzing phase). This delay is added to the controlled access mode delay. At the completion of the new buzzing phase, the new access controlled cycle resumes the transmission of interfered stations.

- (c) *The network is operating under controlled access mode and A starts a*

transmission without detecting any buzz signal.

If A collides with some transmission, it moves to **Buzz-I** and, if A buzz hits some station not yet in **Hold**, we are back to case (b). If all stations have already moved to **Hold**, after R seconds A is allowed to transmit (the transition from **Buzz-II** to **Controlled Access Transmission** is instantaneous) and, at the end of its transmission, A moves to **Hold** together with the other stations. Subsequently, the network operates normally. The extra R seconds are added to the controlled access mode delay.

If A appends its transmission to L transmission, then A moves back to **Idle** and the other stations may append their packets to the train at the right moment. It is possible that A may continue to transmit in random mode after reaching **Idle**. If that occurs, any station which has previously moved to **Hold** eventually times out and moves back to **Idle**. Nevertheless, if these stations do not have any packets to transmit, A may continue to lock the remaining stations in controlled access mode. Time-out T_0 in **Controlled Access Transmission** prevents this capture effect. There is, of course, the case of A joining the network and all other stations moving to **Hold** without any interference with A . If A keeps transmitting vigorously, time-out from **Hold** will save the day.

The above cases (a), (b) and (c) cover all possible states that a joining station can encounter in the network.

The new station joins the active stations gracefully. The extra delay added to controlled access mode delay is at best 0, between 0 and $2R$ in most cases, and of the order of T_0 in the very unlikely worst case situations.

3.6 PERFORMANCE ANALYSIS

We use the performance measures defined in Section 1.2.2.2. For this analysis, we assume that data and preamble transmission times are constant and equal to T_r and T_p , respectively. $T = T_r + T_p$. We also assume that stations have an equal reaction time d and that the buzzing scheme is implemented by transmitting short bursts (d seconds) of unmodulated carrier. The d burst is the minimum transmission time that can be reliably detected by the hardware (see Section 3.4). Under this buzzing scheme the time to detect a buzz signal is of the order of d and, for simplicity, henceforth we assume that $R1 = R$.

For the event diagrams in this section, we assume the following naming conventions for the main events occurring on the tap of a station:

<i>ecoc</i>	= end-of-carrier detected.
<i>boc</i>	= beginning-of-carrier detected.
<i>dob</i>	= detection of buzz at tap.
<i>sob</i>	= start of buzzing at tap.
<i>cobr</i>	= tap stops buzzing on the R-to-L bus.
<i>eobl</i>	= tap stops buzzing on the L-to-R bus.
<i>bop</i>	= beginning-of-packet transmission.
<i>epo</i>	= end-of-packet transmission.
<i>cd</i>	= collision detected at tap.

3.6.1 UTILIZATION AT HEAVY LOAD

Fig. 3.2 portrays the cyclic pattern when all N stations are at heavy load, assuming that packet transmission time is greater than propagation delay between adjacent stations ($T > 2a$). This inequality implies that no packets are successfully transmitted during random mode. Later we will discuss the nuances of allowing $T < 2a$. We observe that stations always conflict at the end of a controlled phase, thus moving back to the buzzing phase. Therefore, the activity in the network is a succession of cycles where active stations are served

round robin, lowest numbered stations first. From Fig. 3.2 we see that:

$$\text{cycle} = 2R + 2\tau + N(T+d) + 2a + 2d + \Delta t .$$

For the usual case where $T \gg d$, $R \gg d$, and Δt is negligible, the utilization is:

$$S(N) = \frac{NT_r}{NT + 2R + 2\tau + 2a} .$$

For $N \gg 1$, $\tau \gg \varphi$, $T_r \gg T_p$, and assuming $\alpha = \pi T$, we have:

$$S(N) = \frac{1}{1 + \frac{6\alpha}{N}} . \quad (3.1)$$

Equation (3.1) will be used to calculate the maximum utilization when we compare Buzz-net to other schemes in Chapter 8.

Now we consider $T < a$. At heavy load, if the rightmost active station is S_j , the only station able to transmit packets during random mode is S_j itself. In fact, if packet transmission times satisfy the following inequality:

$$kT + (k-i)3d < 2\tau_{i-1,j} + 2d ,$$

where k is an integer, then $\max\{k\}$ would be the maximum number of packets that S_j could transmit in random mode. In the formula above, S_{i-1} is the closest backlogged station to S_j , and $3d$ is the interconsecutive packet gap.

This unusual asymmetric behavior gives the rightmost backlogged station a better throughput than the other stations. As we are interested in calculating the throughput over all stations, we disregard the packets transmitted during

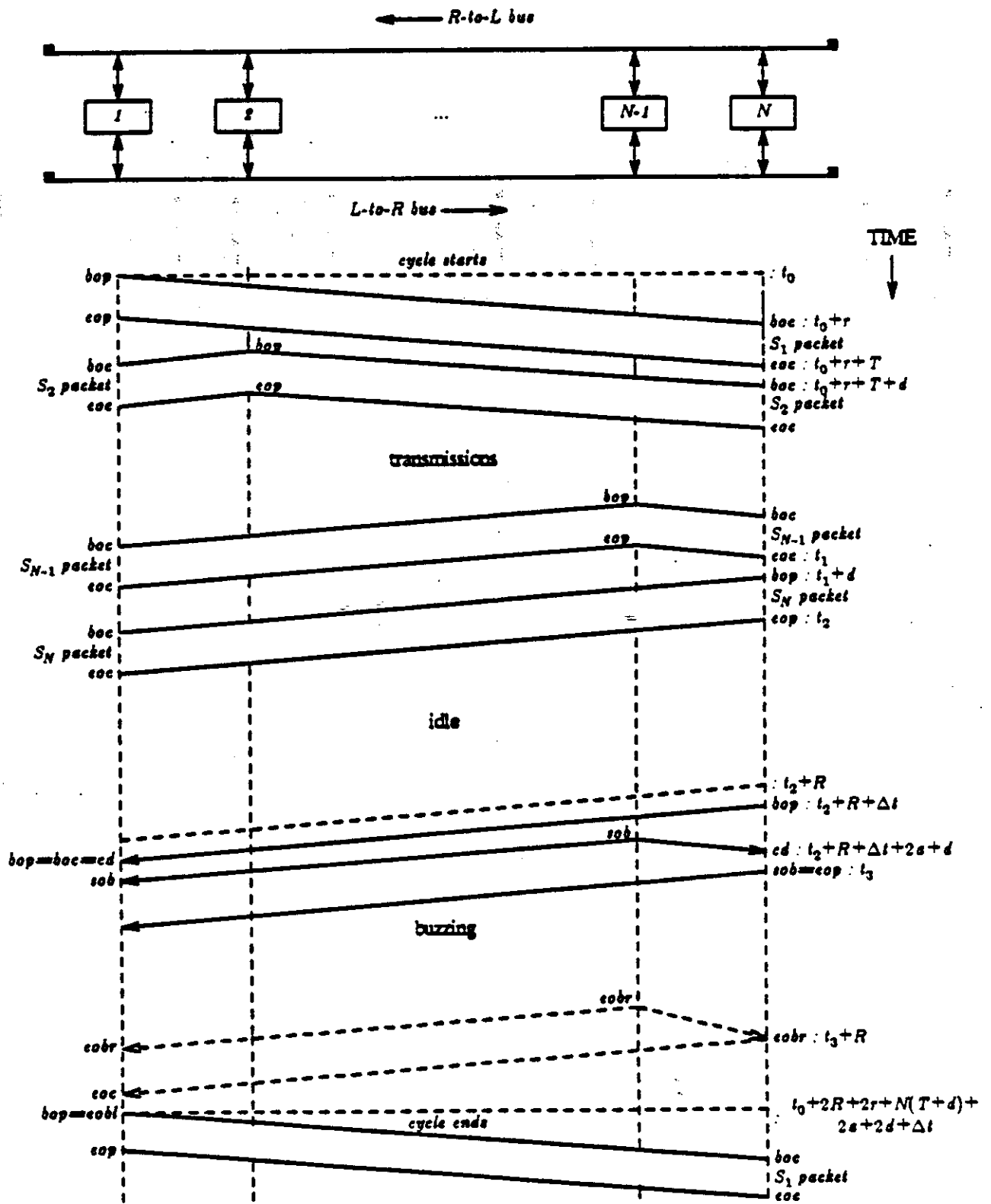


Fig. 3.2 - Cycle at heavy load

random mode by the righthmost backlogged station and a lower bound in total average throughput is achieved.

The worst case situation for utilization occurs when the propagation delay between the right most colliding station and the immediate precedent colliding station is maximized. When only i stations are active, the worst case is achieved for set $1, 2, \dots, i-1, N$ of active stations. For this situation, the cycle is expressed as:

$$\text{cycle}(i) = 2R + 2\tau + 2\tau_{i-1, N} + i(T+d) + 2d + \Delta t .$$

For $T \gg d$, $R \gg d$, and Δt is negligible, we obtain:

$$\text{cycle}(i) = 2R + 2\tau + 2\tau_{i-1, N} + iT .$$

As $\tau_{i-1, N} = (N-i+1)a$, we get, under the fair assumption,:

$$S(i) = \frac{iT_r}{iT + 2R + 2\tau + 2(N-i+1)a}, \quad i > 1 .$$

The worst case for $S(i)$ occurs for $i=2$. If only one station is active, that is $i = 1$, the station can transmit in random access mode since no collision occurs. Thus we have:

$$S(1) = \frac{T_r}{T + d} .$$

3.6.2 INSERTION DELAY

At light load a station can transmit immediately with negligible probability of collision. Therefore, average insertion delay tends to zero as the offered

load goes to zero. At heavy load average insertion delay is closely related to utilization S . Namely, if i is the number of active stations:

$$IDH(i) = iT/S(i) - T$$

For intermediate load values, the average insertion delay cannot be evaluated analytically since the lengths of random access and controlled access cycles are random variables very difficult to characterize. Simulation was used to obtain intermediate load values.

Fig. 3.3 shows simulation results for a network with 15 stations and three different combinations of packet transmission time and round-trip propagation delay. The traffic was uniform (equally distributed among all stations) and Poisson (exponential interarrival time), with single packet messages of fixed size. The preamble transmission time was set to 100 ns. Buzz detection time was null. Since our interests were concentrated in measuring the insertion delay, single buffer stations were used in the simulation. The use of a single buffer avoided excessively increasing the simulation time excessively for high utilizations.

95% confidence intervals were collected and shown if they were over 5% of plotted mean point values.

3.6.3 MAXIMUM INSERTION DELAY

If i stations are active and at heavy load, the maximum insertion delay is:

$$MID(i) = cycle(i) - T, \text{ at heavy load.}$$

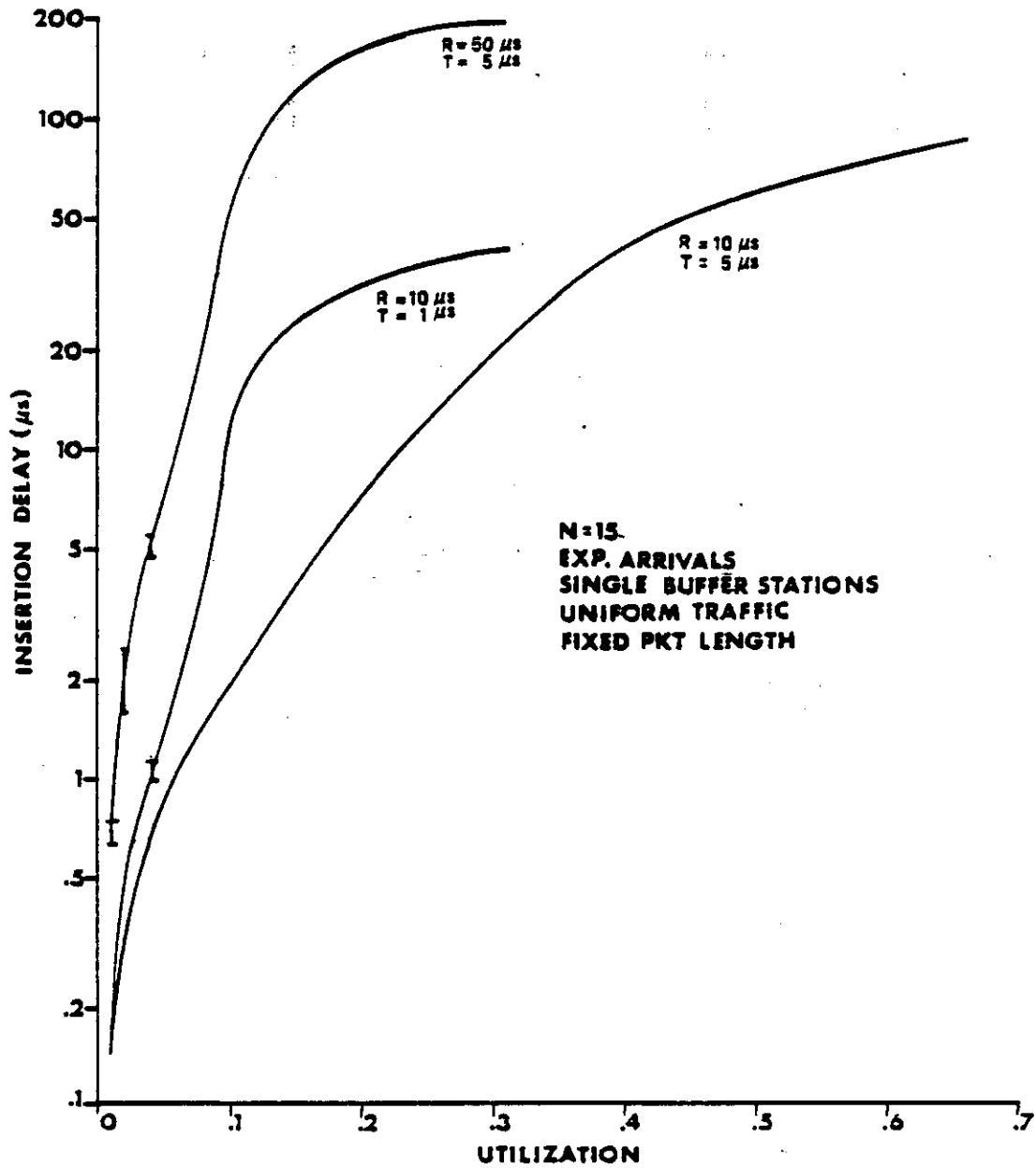


Fig. 3.3 - Insertion delay vs Utilization by simulation.
 95% confidence intervals are within 5% of plotted
 value, unless shown in figure.

In Appendix 3.2 we show that under conditions of intermediate load and very unlikely events, MID can reach the approximate maximum value of $12\tau + (2N-2)T + 5\varphi$.

APPENDIX 3.1

R GUARANTEES PROPER OPERATION FOR BUZZ-NET

Fact: R seconds after entering the *Buzz* mode, a station either detects a buzz on the L-to-R bus, or it senses the L-to-R bus idle, in which case the station knows it is the leftmost backlogged station. Furthermore, the other stations are in **Hold** or **Buzz** mode.

Proof: In this proof we assume that the time required to recognize a buzz pattern is φ . Assume that station S_i enters the **Buzz-I** state at time 0. Soon after S_i enters this state, a buzz pattern will propagate on the R-to-L bus from S_i to the left end of the bus.

First we prove that at most the buzz will reach the left end station $2\tau_{iN} + 2\tau_{i1}$ sec after S_i enters the **Buzz**. This worst case occurs as follows. A station at the right end of the bus is engaged in the transmission of a long packet (or a long sequence of packets). Thus, S_i must defer to the ongoing transmission and cannot inject the buzz on the R-to-L bus. However, if S_i enters the **Buzz-I** at time 0, then by time τ_{iN} , the right end station either senses the buzz from S_i or senses the transmission which prevented S_i from buzzing. In either case, if the right end station is still transmitting, it reacts to the collision d seconds later. At this point, according to our protocol, the right end station interrupts its transmission and emits the buzz pattern, which will reach the left end of

the R-to-L bus by time $2\tau_{iN} + \tau_{i1} + d$.

If the right end station is in **Idle** when the transmission arrives, then it will be prevented from starting any transmission (and will eventually be forced to move to **Hold**) because any possible idle period on the L-to-R bus will be filled with buzz from S_i . Therefore, by time $2\tau_{iN}$, S_i senses the R-to-L bus idle and it starts buzzing that bus d seconds later. The buzz signal reaches the end of the bus after τ_{i1} seconds, by time $2\tau_{iN} + \tau_{i1} + d$. Any upstream transmission hitting S_i on the R-to-L bus after it starts buzzing is necessarily a buzz signal. We have thus proved that the buzz reaches the left end station within $\tau + \tau_{iN} + d$ and all stations on the right of S_i are in **Hold** or **Buzz-I**.

φ seconds after the buzz pattern has reached the left end of the bus, the stations at the left of S_i can be either in **Hold** or in **Buzz-I**. Consider the leftmost of the stations in **Buzz-I**. This station has entered the **Buzz-I** at the latest at time $\tau + \tau_{iN} + d + \varphi$. In any case, a buzz pattern from this station is present at S_i from the left at time $2\tau + 2d + \varphi$. The extra d accounts for the reaction time of S_i . If no buzz is heard by station i by $2\tau + 2d + 2\varphi$, then the set of stations in **Buzz-I** at the left of i is empty. Thus, S_i is the leftmost station with a non-zero backlog (i.e, in **Buzz-I** state). Because the buzzing parameter R is defined to be greater than or equal to $2\tau + 2d + 2\varphi$, the protocol works properly.

Q.E.D. ■

APPENDIX 3.2

WORST CASE INSERTION DELAY FOR BUZZ-NET

A long insertion delay in Buzz-net eventually occurs because a packet arriving during **Hold** must wait for the current controlled mode to terminate before its transmission can start. A tentative transmission of the backlogged packet occurs when the station enters **Idle**. However, as shown in Appendix 1, all stations at the end of the controlled phase move synchronously to **Idle**. If more than one station has a backlog, the tentative transmission may be corrupted by a collision, and a new controlled mode will add overhead to the insertion delay. The backlog packet is only successfully transmitted at the end of the new controlled phase.

The worst case for Maximum Insertion Delay (MID) is determined by the topology, which station starts buzzing first, and on the relationship between τ and T . We consider φ to be the time needed to detect the buzz and T_{\min} to be the minimum packet transmission time. Other parameters have been introduced previously in the chapter. The evolution of events in the network is described in a sequential time table for concise and ease description.

The station under observation is called the tagged station. Two worst case situations are considered:

- (I) The tagged station is in a group physically very closely located to S_1 .

The tagged packet arrives at time t_0 and finds the tagged station in Hold because of a buzz originated in the group at $t_0 - \rho - d$. We assume the group buzzing is originally the first and only one in the net. The following sequence of events occur:

- e1. S_N detects the buzz at $t_0 + \tau$.
- e2. S_1 stops buzzing (assuming S_1 did not originate the initial buzz) at $t_0 + R$.
- e3. End of S_N buzzing is detected by S_1 at $t_0 + R + 2\tau + d$.
- e4. All stations except the tagged station are participating in this controlled phase (they have collided at the beginning of the random phase) and they transmit. S_N ends transmission at $t_0 + R + 3\tau + (N-1)T + (N-1)d$.
- e5. Group senses R-to-L bus idle at $t_0 + R + 4\tau + (N-1)T + Nd$.
- e6. Both busses sensed idle for $R1$ by group at $t_{wait} = t_0 + R + R1 + 4\tau + (N-1)T + Nd$.

At this point the net has returned to random mode. The tagged station tries to transmit its packet. If $T > 2\tau + d - T_{min}$ the following sequence of events may occur:

- e7. The packet propagates and reaches S_N at $t_{wait} + \tau + d$.
- e8. The worst case occurs when S_N had finished a T_{min} transmission d seconds earlier, not colliding with incoming

packet. S_N packet, however, hits S_1 while it is still transmitting and collision is detected by S_1 at $t_{wait} + 2r + d - T_{min}$.

e9. S_1 starts buzzing and buzz is detected by S_N at $t_{wait} + 3r + \varphi + 2d - T_{min}$.

e10. S_N starts buzzing and R-to-L bus is sensed idle by S_1 at $t_{wait} + R + 4r + \varphi + 3d - T_{min}$.

e11. Now, if the tagged station is the right most in the group of $N-1$ stations, it could be forced to wait for $N-2$ transmissions. Thus, the tagged station would only be allowed to transmit at $t_{wait} + R + 4r + (N-2)T + (N+1)d + \varphi - T_{min}$.

If $T < 2r + d - T_{min}$ the following sequence of events may occur:

e7. Collision occurs at tagged station at the end of transmission at $t_{wait} + T$.

e8. Buzz is detected by S_N at $t_{wait} + r + T + \varphi + d$.

e9. R-to-L bus is detected idle by S_1 at $t_{wait} + R + 2r + T + \varphi + 2d$.

e10. As before, the worst case occurs with the tagged station having to wait $(N-2)(T+d)$ before appending its packet at $t_{wait} + R + 2r + (N-1)T + Nd + \varphi$.

Combining the previous two cases we get:

$$\begin{aligned}
MID &= 2R + R1 + 6\tau + (2N-3)T + 2Nd + \varphi + \min(T, 2\tau + d - T_{\min}) \\
&= 12\tau + (2N-3)T + (2N+6)d + 5\varphi + \min(T, 2\tau + d - T_{\min}). \quad (1)
\end{aligned}$$

For the usual case where $\tau > T \gg d$, and $T_{\min} \ll 2\tau$, we obtain:

$$MID = 12\tau + (2N-2)T + 5\varphi. \quad (2)$$

(II) S_N is the tagged station, and we assume that S_{N-1} is closely located to S_N . The tagged packet arrives at time t_0 and finds the tagged station in Hold because of a buzz originated by S_{N-1} at $t_0 - \varphi - d$. We assume S_{N-1} buzzing is originally the first and only one in the net. The following sequence of events may occur:

- e1. S_1 detects the S_{N-1} buzz at $t_0 + \tau$.
- e2. S_{N-1} stops buzzing at $t_0 + R - \varphi - d$.
- e3. End of S_{N-1} buzzing is detected by S_1 at $t_0 + R + \tau - \varphi$.
- e4. S_1 stops buzzing and starts transmitting at $t_0 + R + \tau$.
- e5. All stations except the tagged station are participating in this controlled phase (they have collided at the beginning of the random phase) and they append transmissions to S_1 packet. S_N starts counting silence at $t_0 + R + 2\tau + (N-1)T + (N-1)d$.
- e6. S_N senses both busses idle for $R1$ at $t_{wait} = t_0 + R + R1 + 2\tau + (N-1)T + (N-1)d$.

At this point the net has returned to random mode. The tagged station tries to transmit its packet. If $T > 2r + d - T_{\min}$ the following sequence of events may occur:

- e7. The packet propagates and reaches S_1 at $t_{\text{wait}} + r + d$.
- e8. The worst case occurs when S_1 had finished a T_{\min} transmission d seconds earlier, not colliding with incoming packet. S_1 packet, however, hits S_N while it is still transmitting and collision is detected by S_N at $t_{\text{wait}} + 2r + d - T_{\min}$.
- e9. S_N starts buzzing and buzz is detected by S_1 at $t_{\text{wait}} + 3r + \varphi + 2d - T_{\min}$.
- e10. R-to-L bus is sensed idle by S_1 at $t_{\text{wait}} + R + 3r + 2d - T_{\min}$.
- e11. S_1 stops buzzing at $t_{\text{wait}} + R + 3r + \varphi + 2d - T_{\min}$.
- e12. S_N appends its packet to the train formed by transmissions from stations 1 to $N-1$ at $t_{\text{wait}} + R + 4r + (N-1)T + (N+1)d + \varphi - T_{\min}$.

If $T < 2r + d - T_{\min}$ the following sequence of events may occur:

- e7. Collision occurs at the tagged station at the end of transmission at $t_{\text{wait}} + T$.
- e8. Buzz is detected by S_1 at $t_{\text{wait}} + r + T + \varphi + d$.

e9. R-to-L bus is detected idle by S_1 at $t_{wait} + R + \tau + T + d$.

e10. S_1 stops buzzing and starts transmitting at $t_{wait} + R + \tau + T + \varphi + d$.

e11. S_N appends its packet to the train formed by transmissions from stations 1 to $N-1$ at $t_{wait} + R + 2\tau + NT + Nd + \varphi$.

Combining the previous two cases we get:

$$\begin{aligned} MID &= 2R + R1 + 4\tau + (2N-2)T + (2N-1)d + \varphi + \min(T, 2\tau + d - T_{min}) \\ &= 10\tau + (2N-2)T + (2N+5)d + 5\varphi + \min(T, 2\tau + d - T_{min}). \end{aligned} \quad (3)$$

For the usual case where $\tau > T \gg d$, and $T_{min} \ll 2\tau$, we obtain:

$$MID = 10\tau + (2N-1)T + 5\varphi. \quad (4)$$

Comparing the results of (I) and (II), we observe that (2) is likely to be greater than (4) because usually $T < \tau$ for the high transmission speeds with which we are dealing. However, the worst case observed in (I) only occurs when all stations but one are grouped together, which is highly improbable. Moreover, the sequence of required events for those two cases has a very low probability of occurrence, so the ID will be much lower than MID on the average.

CHAPTER 4

RANDOM ACCESS WITH TIME-OUT CONTROL

4.1 INTRODUCTION

In this chapter we describe RATO, a random access protocol with time-out control. RATO is a very simple scheme that uses the minimum hardware necessary for a protocol implementation in the dual bus topology. The only control requirements are the sensing of activity in the bus, and a fixed time delay between consecutive transmissions from the same station. Because RATO is so simple, it is obviously limited in performance and, as we will see later, dependent on network parameters. However, RATO will be very useful when we perform comparative analysis in the next chapter. We will observe that at times a simple scheme can outperform more sophisticated protocols.

4.2 THE PROTOCOL

In contrast to previous schemes, RATO transmissions are controlled separately in each direction. If bidirectionality is required, a packet can be queued for independent transmission in opposite directions. However, when a session is established between processes residing in different stations, the processes may be able to determine their location relative to each other during the set-up phase, and consequently, stations may attempt to transmit only in a single direction.

Performance measures and assumptions are the same as in Section 1.2.2.2. We further assume that the receiver is able to detect a packet when the packet is immediately preceded by some truncated transmission.

When a station has a packet to transmit, it performs the following steps:

- (1) The station senses the bus. If the bus is busy it defers until the bus is idle.
- (2) The station starts transmitting the packet. If a collision with an upstream transmission occurs, the current transmission is aborted and the station repeats step 1. Otherwise, step 3 is performed next. Observe that the incoming transmission gets only corrupted in its first d seconds, where d is the station reaction delay. The packet preamble guarantees data integrity and allows reliable packet reception at downstream stations.
- (3) The station observes a time-out of T_0 seconds before it considers a new packet for transmission. If the transmission queue is empty after the elapsed T_0 seconds, the station goes idle until a new packet arrives. Then the station performs step 1 again.

From the above description we can see that all the needed steps can be easily implemented under complete hardware control.

4.2.1 MINIMUM VALUE T_0 FOR FAIRNESS

Time-out T_0 is critical to provide fair access to all stations in both directions. We determine T_0 such that all stations have a chance for a successful transmission in a finite time.

Consider the L-to-R bus. Due to transmission deferral, downstream stations are preempted by upstream transmissions. Therefore, the worst case condition for the insertion of a packet occurs for the station next to the most downstream station (the most downstream station only transmits on the R-to-L bus). Let us investigate the worst case for station $N-1$ trying to transmit to station N . Assume that station $N-1$ detects the bus idle and starts transmitting. After $T-\epsilon$, where ϵ is very small, the transmission is almost completed but a collision with a transmission from station 1 occurs. Station $N-1$ defers and attempts again when the bus is idle (within d seconds of reaction delay). When the transmission is almost completed a collision from station 2 now occurs. Collisions from other stations follow this pattern until station $N-1$ finally succeeds after transmission from station $N-2$. The sequence of events as seen by an observer on the bus is depicted in Fig. 4.1, where a worst case collision is represented by $\langle T-\epsilon \rangle$, $\langle i \rangle$ is a successful transmission of duration T by

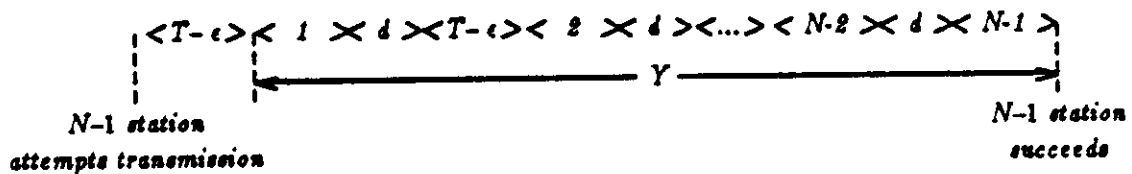


Fig. 4.1 - Worst Case Insertion Delay for RATO.

station i , and $\langle d \rangle$ is a station reaction delay.

From the figure we get:

$$Y = (N - 2)T + (N - 2)T_e, N > 2.$$

To provide a finite insertion delay to station $N-1$ we must guarantee that the next transmission by station 1 (also applicable to other stations) does not occur before T_0 seconds where T_0 is given by:

$$T_0 \geq \lim_{\epsilon \rightarrow 0} (Y - T).$$

Hence,

$$T_0 \geq (2N - 5)T + (N - 2)d, N > 2.$$

For $N \gg 1$ and $T \gg d$ we have $T_0 \geq 2NT$.

It is clear that all the other stations, not only station 1, must also be subjected to the same constraint.

4.3 PERFORMANCE ANALYSIS

Using the performance measures and assumptions defined in Section 4.4.3, we next proceed to the evaluation of RATO performance. We assume $T_0 = (2N - 5)T + (N - 2)d$.

4.3.1 UTILIZATION

At heavy load, time-out T_0 forces transmissions to be clustered together in rounds starting every $T_0 + T$ seconds. A round is depicted in Fig. 4.2.

From 4.2 the bus utilization when i stations are active is:

$$S(i) = \frac{iT_r}{T_0 + T} = \frac{i}{N-2} \frac{T_r}{2T + d}, \text{ for } i \leq N-1. \quad (4.1)$$

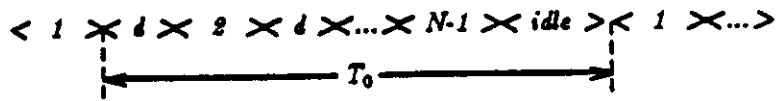


Fig. 4.2 - Heavy Load Round in Ratio.

Channel capacity or maximum utilization S is given by $S(N-1)$. The maximum utilization is independent of τ (the end-to-end delay) and approaches .50 as $T_r \gg T_p$ and $N \gg 1$. This relationship implies that packet lengths must be large to provide a good throughput, especially since the preamble must increase with transmission speed. Independence from τ makes it possible to cover large distances and still maintain acceptable throughput.

4.3.2 DELAY PERFORMANCE

Delay performance is measured in terms of the insertion delay (ID). At light load the bus is usually idle and very few collisions take place. With light traffic the insertion delay is negligible if packet interarrival time is greater than T_0 . In case of multipacket messages, the ID for the first packet is 0 and is T_0 for the other packets of the same message.

At heavy load, the insertion delay (IDH) always equals T_0 . However, the worst case for ID (MID) occurs at intermediate load when station $N-1$ (assuming left to right transmissions) tries to transmit after a T_0 second wait and finds the bus idle. When the transmission is almost completed, a collision occurs with a transmission from station 1. Then station $N-1$ attempts again and suffers a collision station 2. Collisions with the other stations follow this pattern until station

$N-1$ finally succeeds after the transmission from station $N-2$. The above pattern was also actually assumed as the worst case in calculating T_0 . Therefore, from Fig. 4.1 and remembering that station $N-1$ has already waited T_0 at the beginning of the events, $MID = 2T_0 + T$. For $N \gg 1$ and $T \gg d$ we find $MID = 4NT$. This worst case result is approximately twice the value of IDH . Of course, the events leading to the worst case are very unlikely, and MID should neither affect the average delay nor the delay distribution.

4.4 CONCLUSION

A very simple random access protocol with time-out control (RATO) was described. The protocol uses time-out T_0 as its only control and relies on deferral to upstream transmissions. Due to its simplicity, RATO implementation cost should be the lowest among all protocols.

A lower bound on the value of T_0 to guarantee fair access and bounded delays was given. A major drawback is the dependency of T_0 on the product NT . If T_0 is set to its minimum acceptable value, then a new station insertion should be followed by a correspondent increase in T_0 . In case of station deletion, T_0 should be decreased, to avoid wasted bandwidth and unnecessary delay.

Expressions for utilization and delay at light and heavy load are obtained. In the next chapter RATO is compared with the performance of the other protocols and it is shown that under certain conditions RATO is a good choice.

CHAPTER 5

TOKEN-LESS PROTOCOLS

5.1 INTRODUCTION

The main motivation for the development of the Token-Less family was to eliminate some of the implementation difficulties and performance limitations experienced with existing and proposed protocols.

U-Net and TDT-Net, described in Chapter 2, rely on the detection of special patterns to implement the signalling scheme which controls the channel. The need to recognize different transmission patterns may cause difficulties in implementation. Both protocols offer excellent performance when stations are symmetrically located and the network is equally loaded with single packet message traffic. However, in the simulation results in Chapter 6, we show that the performance of U-Net and TDT-Net degrades considerably with unbalanced and multipacket traffic.

Buzz-Net, described in Chapter 3, achieves optimal performance for a single sending station (single or multipacket messages) and negligible delay at light load (this behavior is common to all random access schemes). However, performance degrades when two or more stations collide due to the overhead from cycle reinitialization. Furthermore, Buzz-Net relies on the generation and detection of a special pattern to implement the buzzing scheme. All three schemes are adversely affected by an increase in network span.

Rato, described in Chapter 4, uses a single time-out T_0 to control the channel. However, T_0 is a function of the number of active stations and the maximum transmission time. Although Rato is insensitive to network span, its delay performance is only acceptable when the product NT is small. Utilization approaches .5 for large N , and the protocol unnecessarily delays multipacket messages even if bandwidth is available.

Fasnet, a protocol developed for the dual bus topology, is a synchronous slotted protocol, with the physical end stations being responsible for slot generation. Stations are required to maintain bit synchronization with the channel, and this requirement imposes strict tolerances in clock recovery and internal circuit delays. Synchronous implementation, seen also in rings (see comments in Chapter 1), requires a great deal of processing at channel speed and the active circuitry in series with the line compromises reliability.

Token-Less protocols achieve high performance standards using the detection of activity in the channel as the only low level hardware requirement. Token-Less provides round-robin access to active stations without using a token: hence the name Token-Less. Because detection of activity is essential to implement the deferral procedure in unidirectional channels, the complexity of the high speed circuitry must be kept to a minimum, improving reliability and cost. Starting from a simple scheduling concept, we develop four versions of differing complexity. Two versions, **TLP-2** and **TLP-4**, provide dynamic selection of end stations and are less sensitive to increases in network span or asymmetric placement of stations. **TLP-3** performs as U-Net or TDT-Net, while **TLP-1**, the simplest version, compromises between performance and simplicity of state diagram. A comprehensive comparative analysis of the Token-Less family

including its versions and other protocols is contained in Chapter 6.

The basic operating principles of Token-Less protocols are given in Section 5.2, and details of the different versions are described in Section 5.3. Section 5.4 addresses joining and recovery issues. Performance analysis is found in Section 5.5.

5.2 PRINCIPLES OF OPERATION

The Token-Less Protocol (TLP) runs on the dual bus architecture shown on Fig. 1.4. Stations are connected to each bus via two passive taps, a *receiver* tap and a *transmit* tap. Stations receive packets and monitor channel activity through the receiver tap. Specifically, the receiver can observe presence or absence of activity (i.e. data) and detect events such as *EOA* (End of Activity) and *BOA* (Beginning of Activity).

The transmit tap transmits (data) packets or an activity signal (*AS*). The activity signal keeps the downstream part of the channel busy. Its implementation (modulated or unmodulated carrier, random bits, continuous sequence of 1's, etc.) can be chosen according to the low level encoding utilized for transmission on the channel.

A maximum reaction delay of d seconds is assumed between the time a station senses *EOA* on one bus and the time it can start transmission on either bus. Likewise, there is a maximum d second delay between the sensing of activity from an upstream station and the interruption of an ongoing transmission. Moreover, an activity burst of d seconds is the minimum amount of energy reliably detected at any interface. The actual value of parameter d depends on

the speed and transmission delays of the detection logics in the hardware implementation. Experimentation shows that detection of activity in optical fibers can be done reliably in nanosecond intervals.

A transmitting station always defers to an upstream transmission by aborting its own. The upstream transmission proceeds with only the first d seconds corrupted regardless of the number of other downstream stations attempting to transmit. If the preamble is sufficiently long, this feature guarantees that a packet which has been completely transmitted by a station is correctly received by all (downstream) stations.

It is also assumed that an interface detects a packet even when the packet is immediately preceded by a truncated transmission. The underlying assumption is that the beginning-of-packet flag cannot be replicated within the packet data nor contained in the activity signal described above. Flags can be implemented as reserved bit patterns (in which case bit stuffing is required to preserve data transparency), or as code violations on the bit encoding level.

The goal of the protocol is to guarantee collision-free transmissions among all backlogged stations, and to achieve good throughput/delay performance for a variety of traffic conditions and station placement. Furthermore, the need to detect special packets (e.g., tokens) is avoided, and control is completely distributed. These goals are achieved by *EOA* events propagating in the two busses alternatively. *EOA* events can be viewed as virtual tokens which allow stations to transmit packets in a round-robin fashion. Some advantages of controlling the channel only through *EOA* events are simple, reliable, and low cost implementations even at very high speed. Another advantages are easy implementation of initialization and recovery procedures for the protocol.

5.3 THE PROTOCOL

The protocol basically consists of four procedures. Each of these procedures has a specifically defined purpose and is represented by a set of states in the protocol's state diagram. The first procedure, called probing, enables a station to recognize its turn to transmit in a round. The second procedure enables a station to determine whether it is an extreme (left most or right most active) station

and reverse rounds. The third procedure provides recovery when illegal events are detected. The fourth procedure enables a newly active station to synchronize with other active stations, if any, or to initialize the round-robin cycles in an empty net. An active station is a station that is neither idle nor powered-off.

Different parameters and options may be chosen when specifying the full protocol. Before exploring the details of the different implementations, the common foundation of the various versions of Token-Less protocol is presented below.

5.3.1 BASIC TOKEN-LESS PROTOCOL

In describing the basic protocol, A is a variable designating one channel and \bar{A} designates the opposite channel. Events on channel A are indicated by $EVENT(A)$.

Assume channel A has been sensed busy by station S_i with a backlog. S_i then waits for $EOA(A)$. If $EOA(A)$ occurs, S_i starts transmitting activity signal on channel A . If $BOA(A)$ occurs, S_i stops transmission and waits for the next $EOA(A)$. Otherwise, after time-out d , S_i starts packet transmission on both

channels.

The above probing procedure avoids starting transmission on channel \bar{A} when an interpacket gap is detected. If any burst of activity triggered by an interpacket gap is sent on channel \bar{A} , a collision with an upstream transmitting station may destroy the desirable collision free property of TLP. Actual packet transmission only starts on both channels when the end of a train of packets is detected. Prior to that, only the first d seconds of the incoming packet on channel A is corrupted.

After packet transmission is completed, the station tests its status as the extreme station. S_i sets time-out ES (Extreme Station) and continuously transmits the activity signal on channel \bar{A} until either a $BOA(\bar{A})$ is detected or time-out ES occurs. If $BOA(\bar{A})$ is detected, S_i cancels time-out ES and repeats the above procedure with A and \bar{A} reversed. If ES is reached, S_i realizes it is an extreme station and starts the *round restart procedure*. The *round restart procedure* enables a round to be initiated in the opposite direction. Different versions of TLP take slightly different actions at the end of a round. Therefore, this procedure is explained separately in each TLP version.

If both channels are initially idle, the *initialization procedure* is invoked. S_i sets time-out ND (Network Dead) and waits for the first of two events: BOA on either channel or time-out ND . If BOA occurs on channel A , S_i cancels time-out ND and performs as if channel A were initially sensed busy. Alternatively, if time-out ND occurs (no other station is active in the network), S_i begins the *recovery procedure* to initialize the idle network.

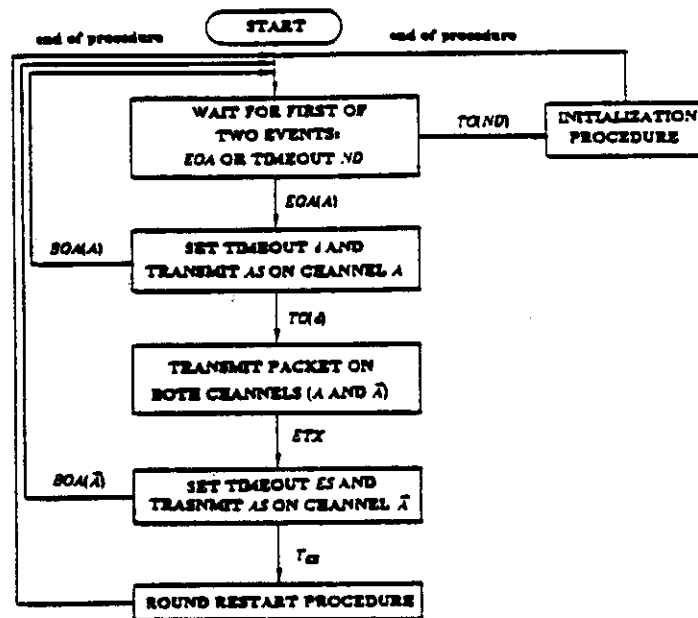


Fig. 5.1 - Flow Diagram of the Basic Token-Less Protocol.

The *recovery procedure* is also invoked during normal operation when illegal events are detected on the channels. Illegal events may be symptomatic of a temporary malfunction in one interface or a station out of synchronism. A thorough discussion of recovery and joining procedures is given in Section 5.4.

A block diagram of Basic Token-Less protocol is shown in Fig. 5.1.

5.3.2 VARIOUS IMPLEMENTATIONS

The several ways to specify round restart, initialization, and choose parameters *ND* and *ES* lead to different versions of TLP. Two versions, TLP-1 and TLP-3, require all powered-on stations to be active in the network. TLP-2 and TLP-4 require only backlogged stations to be active. Moreover, TLP-3 and TLP-4 use additional status variables to improve performance. All versions

are completely distributed, follow the basic protocol described in the previous section, and use the same recovery procedure.

These four versions, **TLP-1**, **TLP-2**, **TLP-3**, and **TLP-4**, constitute the family of Token-Less Protocols.

Definitions

Variable A denotes the channel where EOA is expected or where the station is currently transmitting the activity signal. Channel A is called the synchronizing channel. The identity of A changes during the execution of the protocol and is assigned value 0 or 1 depending on whether the synchronizing channel is, respectively, channel L-to-R or channel R-to-L.

Parameter $R = 2\tau + 2d$ is fundamental in the implementation of the protocols. τ is the end-to-end propagation delay. R may be interpreted as the interval of time needed for EOA to be propagated from one end station to the opposite end station (τ seconds), detected at the latter station and regenerated as a BOA on the other channel (reaction delay d), propagated back to the former station (τ seconds), and finally detected (reaction delay d).

A station physically located inside the present sweep of the virtual token is called an *inside* station. Similarly, a station physically located outside the present sweep of the virtual token is called an *outside* station.

A station is *idle* if it is in the **IDLE** state. A station that is neither idle nor powered-off is called an *active* station. In **TLP-1** and **TLP-3**, a powered-on station is always active. In **TLP-2** and **TLP-4**, an idle station only becomes active when a packet backlog is formed.

5.3.2.1 TLP-1

TLP-1 is described below in detail. This description includes background information which also pertains to **TLP-2**, **TLP-3** and **TLP-4**. The state diagram shown in Fig. 5.2 defines **TLP-1** operation.

States **ON** and **I** at the left of the figure represent the *initialization procedure*. Also at the left, states **R1** and **R2** represent the *recovery procedure*. **R** is a pseudo state which simplifies the drawing of state transitions into recovery. States **WFT**, **TTT**, **ST**, and **TXP** represent the probing and transmission procedures. State **ES** executes the *round restart procedure*.

A station enters **ON** only when it is powered-on. If one of the channels is busy, A is set to that channel and the state moves to **WFT** where the station waits for synchronizing $EOA(A)$ as described in Section 5.1. If both channels are idle, S_i sets time-out $ND = R + d$ and moves to state **I**. ND guarantees that if any station is active in the network it will be heard before any other action is taken. The reason for setting ND to the given value will be clear after the *round restart procedure* is explained. If activity is sensed in a channel (BOA detection) before time-out ND is reached, A is set to the corresponding channel, and the station leaves the *initialization procedure* moving to state **WFT**. Otherwise, when ND is reached, the *recovery procedure* is executed by states **R1** and **R2**. In the state diagram, the dotted lines which converge toward **R1** represent transitions due to illegal events. The *recovery procedure* is standard for all versions of **TLP** and will be explained in Section 5.4.

TLP-1

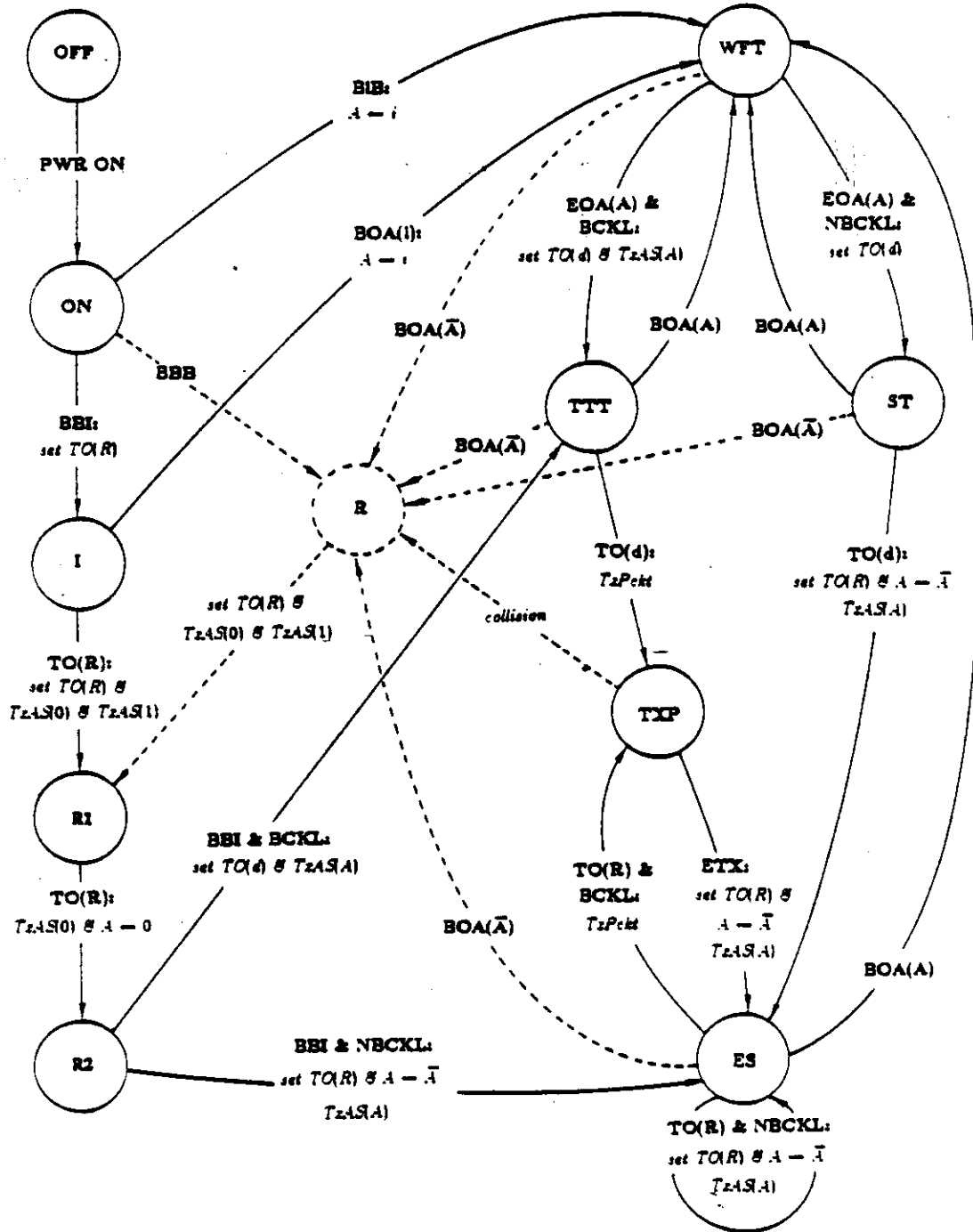


Fig. 5.2 - TLP-1 State Diagram.

States **WFT**, **TTT**, **ST**, and **TXP** allow a station to identify its turn and transmit a backlogged packet at the proper time. When in **WFT**, the station waits for $EOA(A)$ as described in 3.1. If there is a backlogged packet, detection of $EOA(A)$ moves the state to **TTT**, after setting time-out d . The purpose of **TTT** is to detect the end of a train of packets with an interpacket gap of at most d seconds. If no $BOA(A)$ is detected before time-out d expires, the state moves to **TXP** and the head of the backlogged packet queue is transmitted on both channels. At the end of packet transmission, the state moves to **ES** and the activity signal is transmitted on channel \bar{A} . However, if $BOA(A)$ is detected while in **TTT**, the state moves back to **WFT**. State **ST** performs as **TTT** except that no packet transmission occurs. Consequently, the state changes directly from **ST** to **ES**, without passing through **TXP**.

The *round restart procedure* is performed in state **ES**. In this state, the station transmits the activity signal in the channel opposite the channel where the virtual token is propagating. The former is the new synchronizing channel. If activity lasts $ES = R$ seconds the station realizes it is an extreme station. If the station has a backlogged packet it moves back to **TXP**. Otherwise, it remains in state **ES** and behaves accordingly after inverting the identity of the synchronizing channel.

Observe that if only one station is active in the network, periods of activity in either channel are separated by R seconds of idle time. Because any new active station waits for $ND = R + d$ seconds in state **I** before starting recovery, it is clear that this joining station will be synchronized with the network if at least another station is active. It is also clear that transmission by the joining station is detected by the active station before time-out $ES = R$

expires, because of the definition of R .

An example of the operation of TLP-1 for a network with 10 stations is

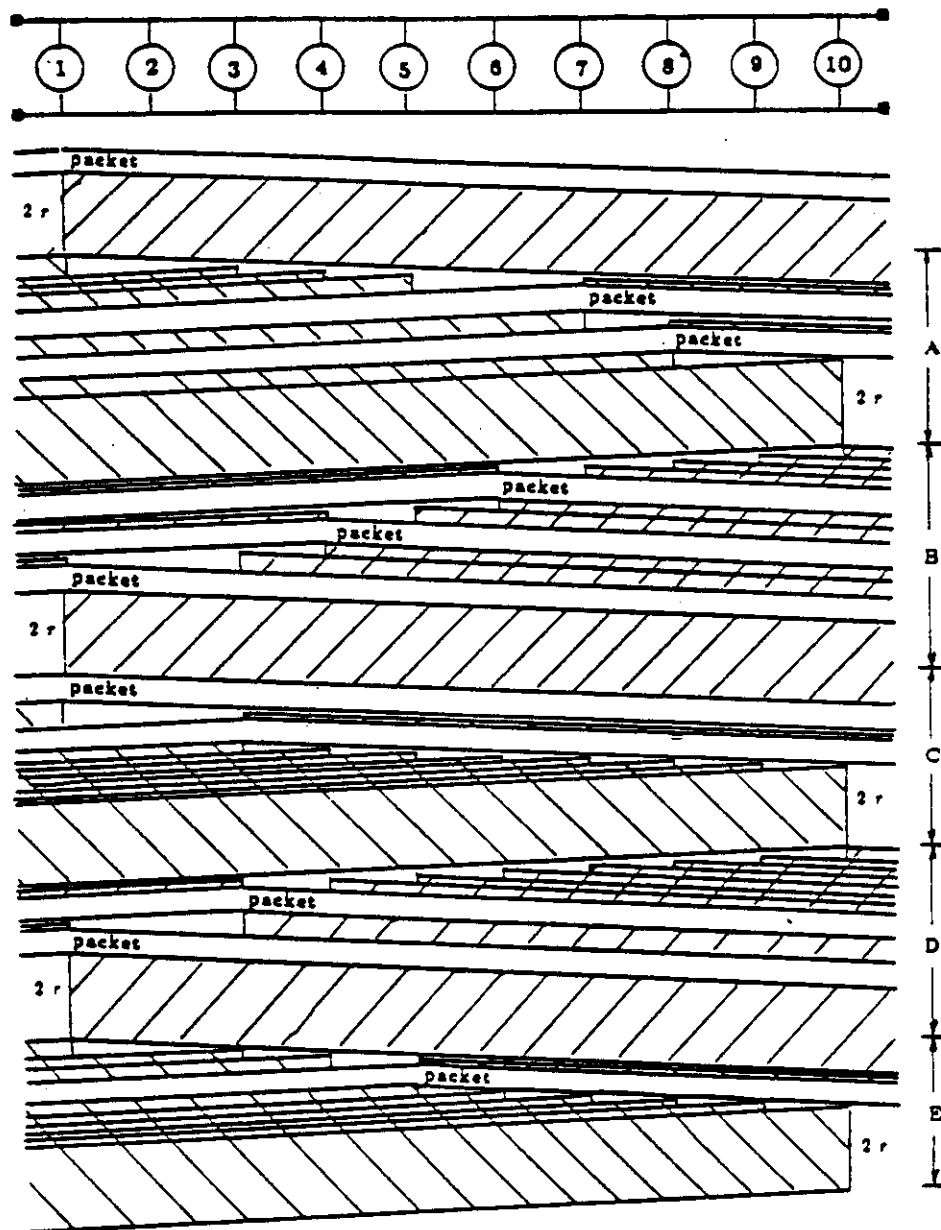


Fig. 5.3 - TLP-1 Space Time Diagram.

given in the space-time diagram shown in Fig. 5.3. The time intervals A, B, C, D, and E represent rounds. In round A the virtual token propagates from left to right, stations 1, 3, 4, 5, 7, 8, 9, and 10 are powered-on, and stations 1, 7, and 8 transmit packets. In round B, the virtual token propagates from right to left, station 6, is powered-on, and stations 6, 4, and 1 transmit packets. Rounds C,

D, and E are similar.

TLP-1's greatest advantage is simplicity. Only the first station which finds the network dead must execute the *initialization procedure*. All other stations detect activity when they come alive and gracefully join the set of active stations. Ease of network joining is a consequence of time-out $ES = R$ at the end of each round. However, performance is impaired due to this extra overhead.

Performance also degrades under other special circumstances. In **TLP-1** a powered-on station always performs activity on the channel even if the station has no packet to transmit. This implies that the virtual token in each round revolves between extreme powered-on stations. If traffic load is unbalanced and only a few stations are actually transmitting, this mode of operation introduces unnecessary delay because the virtual token must sweep the entire bus, rather than only the section of the bus containing the stations involved in transmission.

5.3.2.2 TLP-2

In **TLP-2**, the unnecessary delay observed in **TLP-1** is eliminated by allowing the virtual token to sweep only between extreme stations which have a packet to transmit. This efficiency is achieved by forcing a station with no backlog to idle. Fig. 5.4 shows the state diagram for **TLP-2**.

Compared to **TLP-1**, **ST** is no longer necessary (only backlog stations are active) and **ON** is replaced by states **IDLE** and **B**. **IDLE** is initially entered when a station is powered-on. While in **IDLE**, transition to **B** only occurs when a packet is backlogged. When **ES** is left, the state moves back to

TLP-2

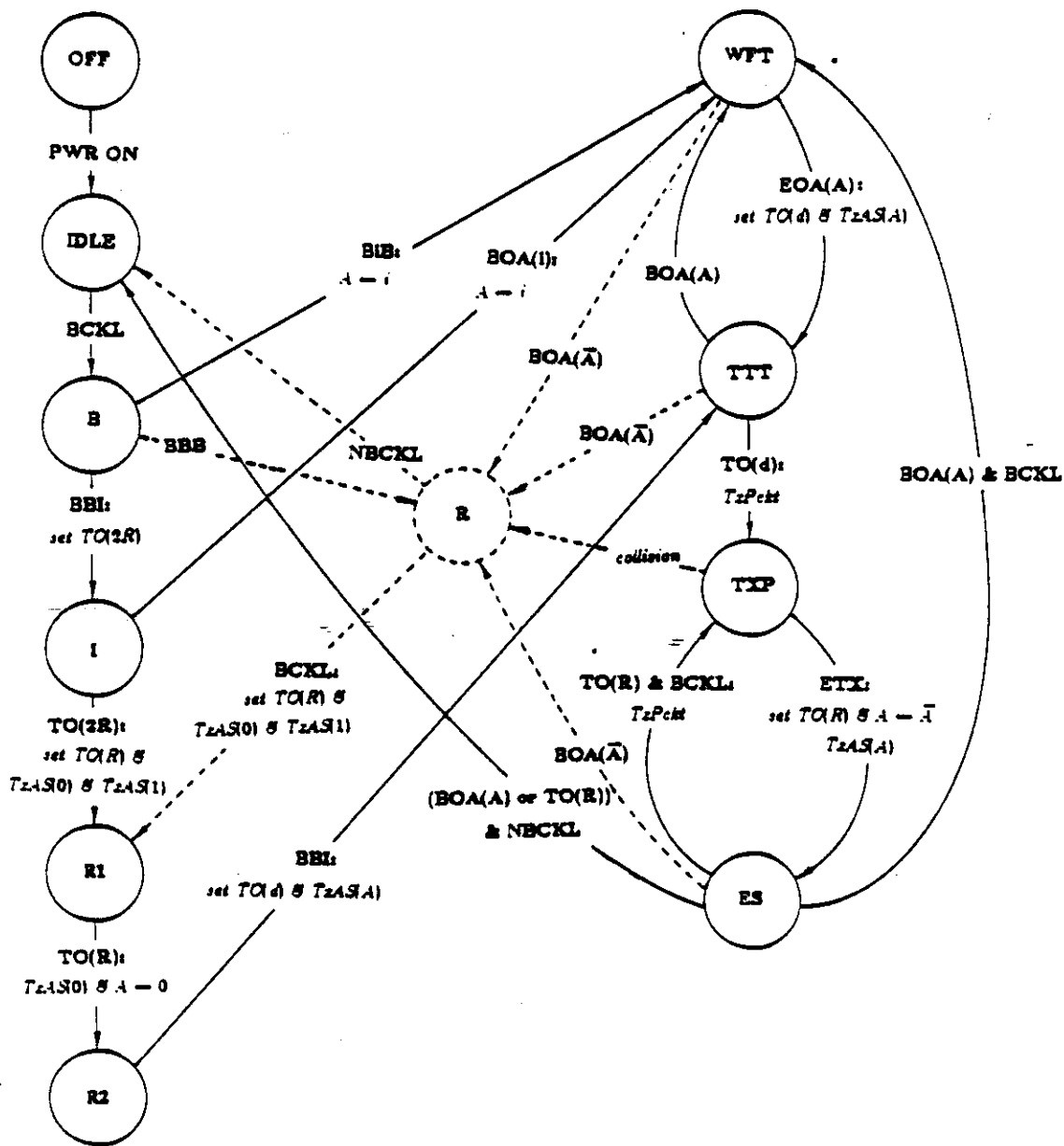


Fig. 5.4 - TLP-2 State Diagram.

IDLE if no backlogged packet is present.

Transitions from **B** are similar to those from **ON** in **TLP-1**, except that time-out ND is set to $2R$, allowing newly backlogged stations to smoothly join the other active stations. The longest delay for a joining station occurs as follows. S_2 , physically very close to S_1 , transmits a packet synchronized by channel R-to-L. Assume packet transmission ends on S_2 tap at time t_0 . After activity signal is transmitted on channel L-to-R for R seconds, the station moves to **IDLE**. Now, assume that the only other backlogged station participating in the round is S_N . S_N transmission starts on bus L-to-R at $t_0 + R + \tau_{2N} + d$. S_N packet is detected by S_1 $R + \tau_{2N} + \tau_{N1} + 2d - \tau_{21}$ ($= 2R - \tau_{12} - \tau_{21}$) seconds after transmission by station 2 has passed. If station 1 had backlogged a packet immediately after the transmission by station 2 had passed, the delay $ND = 2R$ would have provided the necessary waiting time to avoid erroneous initialization of the network by station 1, which had not sensed any activity for almost $2R$ seconds.

In terms of state diagram complexity, **TLP-1** and **TLP-2** are very similar. Performance, however, may differ substantially. Improved behavior for the identical traffic pattern as in the previous example for **TLP-1** is shown in the space time diagram of Fig. 5.5. When active stations are physically close (compared to whole network length) and activity continues for successive rounds, **TLP-2** is preferred to **TLP-1**. The virtual token sweep is confined only to the span of the network covering the active stations, and stations do not incur initialization overhead due to constant activity on the channels. An example of such a favorable situation occurs when physically near stations transmit multipacket messages. At heavy load, **TLP-1** and **TLP-2** perform identically.

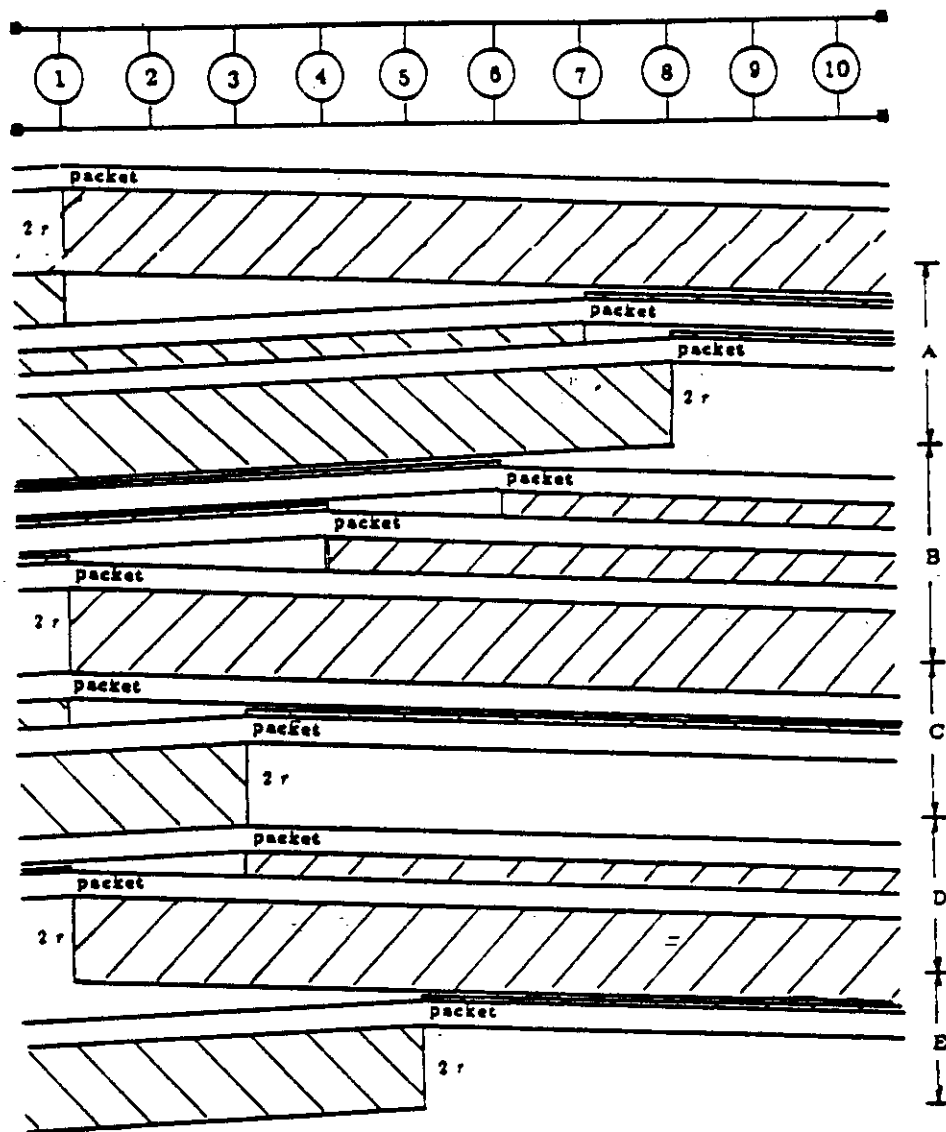


Fig. 5.5 - TLP-2 Space Time Diagram.

However, if the load is light, **TLP-2** shows inferior performance because a newly backlogged station must execute the *initialization procedure* whenever the network is idle.

5.3.2.3 TLP-3

A substantial contribution to overhead in both previous versions of the protocol is given by time-out R between rounds. This delay can be reduced by observing that in **TLP-1**, if a station is an extreme station in a round, then, in

the next round, the station is likely to be an extreme station again. **TLP-3** works similarly to **TLP-1**, with the exception that an extreme station starts a new round in the opposite direction as soon as a time-out of $2d$ seconds has elapsed since the last action of the station on the channel. Time-out $2d$ in **TLP-3** is negligible compared to time-out R used in **TLP-1**. The result is substantial performance improvement. Time-out $2d$ is sufficient to guarantee that a new powered-on outside station joins the set of active stations in a finite time. This joining procedure is thoroughly explained in Section 5.4.

The state diagram for **TLP-3** is shown in Fig. 5.6. As opposed to **TLP-1**, **TLP-3** substitutes states **ES0** and **ES1** for state **ES**. In addition, a flag $E(A)$ signals whether or not a station is the most upstream active station in channel A . **ES0** is entered after a packet transmission if flag $E(A)$, corresponding to the present synchronizing channel A , is 0. **ES0** performs similarly to **ES**. Nevertheless, if time-out R is reached while in **ES0**, $E(A)$ is set to 1, indicating that the station is currently an extreme station on that channel. Transition into recovery from **ES0** only occurs if activity on channel \bar{A} (i.e., $BOA(\bar{A})$) is detected. If $BOA(A)$ occurs, the state moves to **WFT**, as in normal procedure.

ES1, however, is entered after a packet transmission if flag $E(A)$, corresponding to the present synchronizing channel A , is 1. **ES1** performs similarly to **ES0** except that time-out $2d$ is used instead of time-out R and any activity on either channel (while in **ES1**, triggers recovery. Transition into recovery resets $E(0)$ and $E(1)$ to 0.

Transition from **ES0** to **ES1** occurs when time-out R expires, if $\overline{E(A)} = 1$ and there is no backlog. If $\overline{E(A)} = 0$ and there is no backlog, the state remains in **ES0**. In case of backlog, the state moves back to **TXP**. The

TLP-3

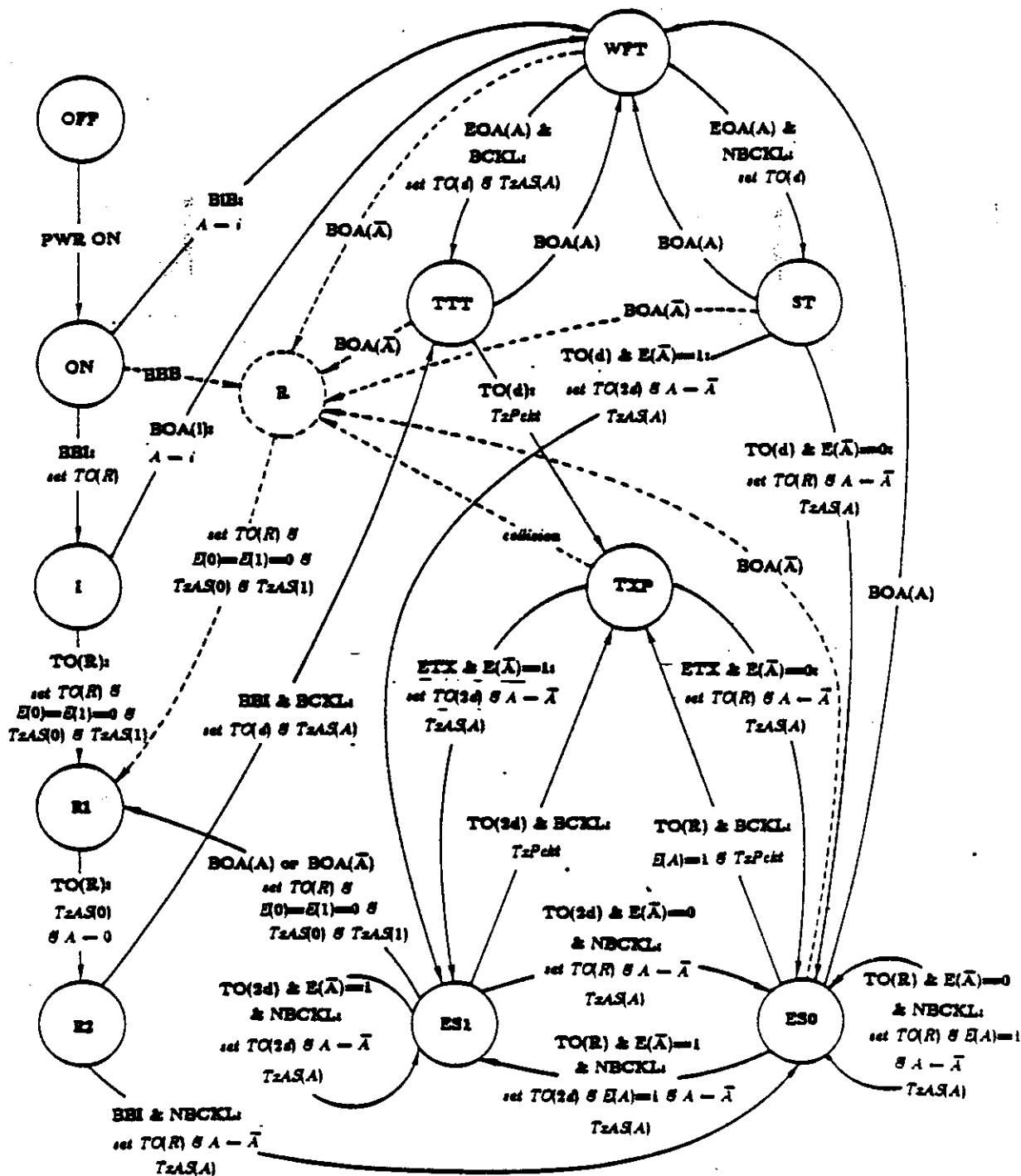


Fig. 5.8 - TLP-3 State Diagram.

reverse is true for transitions from **ES1** to **ES0** if time-out $2d$ is substituted for R .

TLP-3 is always superior to **TLP-1** except in unrealistic cases where stations turn on and off continuously. In such situation collisions could force additional recovery overhead of up $2R + d$ seconds per round (see Section 5.4). Under this circumstance **TLP-1** performs better because overhead (not including propagation delay) is kept at R per round.

The cost of this improved performance is a more complex state diagram and additional use of status flags. These flags are needed as internal hardware variables contributing to a more elaborate implementation.

The space-time diagram in Fig. 5.7 shows how this version works for the same example considered previously for the other versions. Observe that the backlogged packets are transmitted in a much shorter time with **TLP-3**.

5.3.2.4 **TLP-4**

TLP-4 combines features of both **TLP-2** and **TLP-3**. The token sweep is confined between the most widely separated backlogged active stations, as in **TLP-2**. Extreme stations preserve their status in flag variables set in the same manner as in **TLP-3**. The extreme station flag variable allows round reversal with a minimum overhead of $2d$ seconds.

Fig. 5.8 shows the state diagram for **TLP-4**. As in **TLP-2**, state **ST** found in **TLP-1** and **TLP-3** is unnecessary, because only backlogged stations are active. Also, transitions between **ES0** and **ES1** do not exist, because absence of backlog moves the state to **IDLE**. The need for state **WT** is

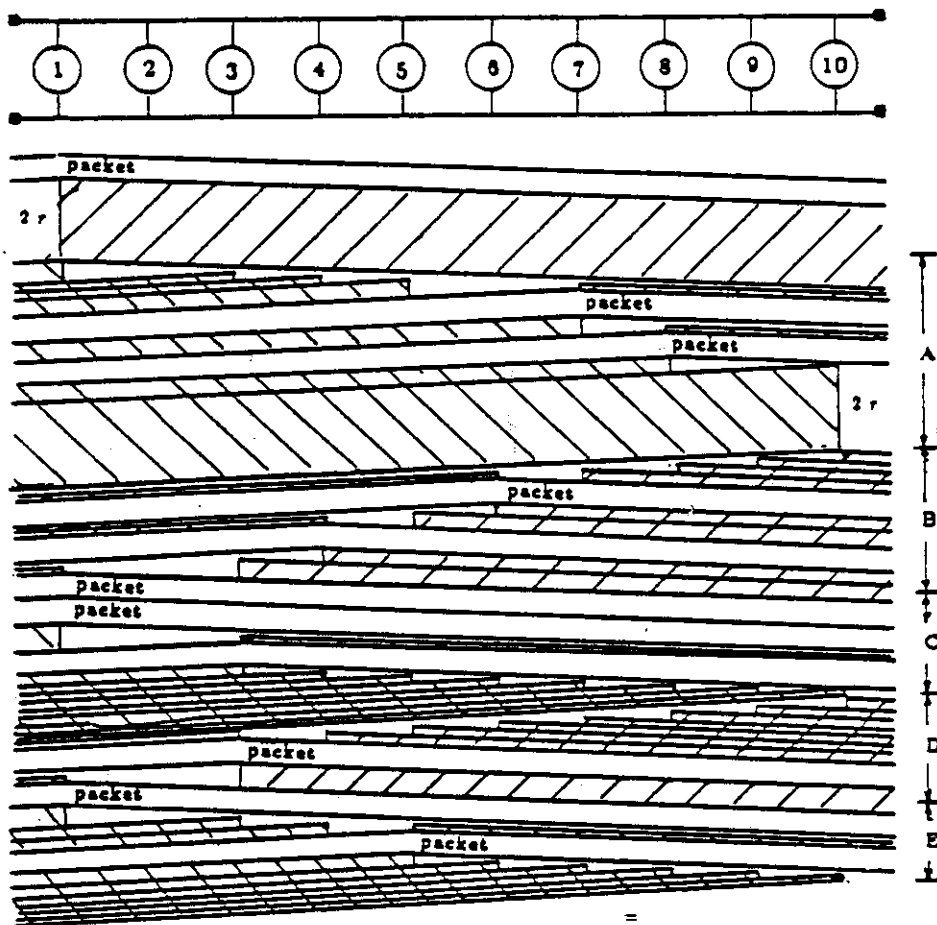


Fig. 5.7 - TLP-3 Space Time Diagram.

explained in the recovery section. Essentially, it prevents a lock-up condition which could provoke infinite delays in accessing the network.

Because flag variable status is preserved when the station returns to **IDLE**, initialization delay is diminished by allowing extreme status stations (any station with a channel flag variable set to 1) to transmit immediately synchronized on the corresponding channel, if both channels are sensed idle at packet arrival. If the station is an extreme station on both channels, the last value of *A* determines the synchronizing channel. This procedure is executed by the conditional transition from **B** to **TTT**. Also different from the initialization in **TLP-2**, a station does not start recovery if both channels are sensed busy while in **B**. Sensing both channels busy probably means that a recovery is occurring.

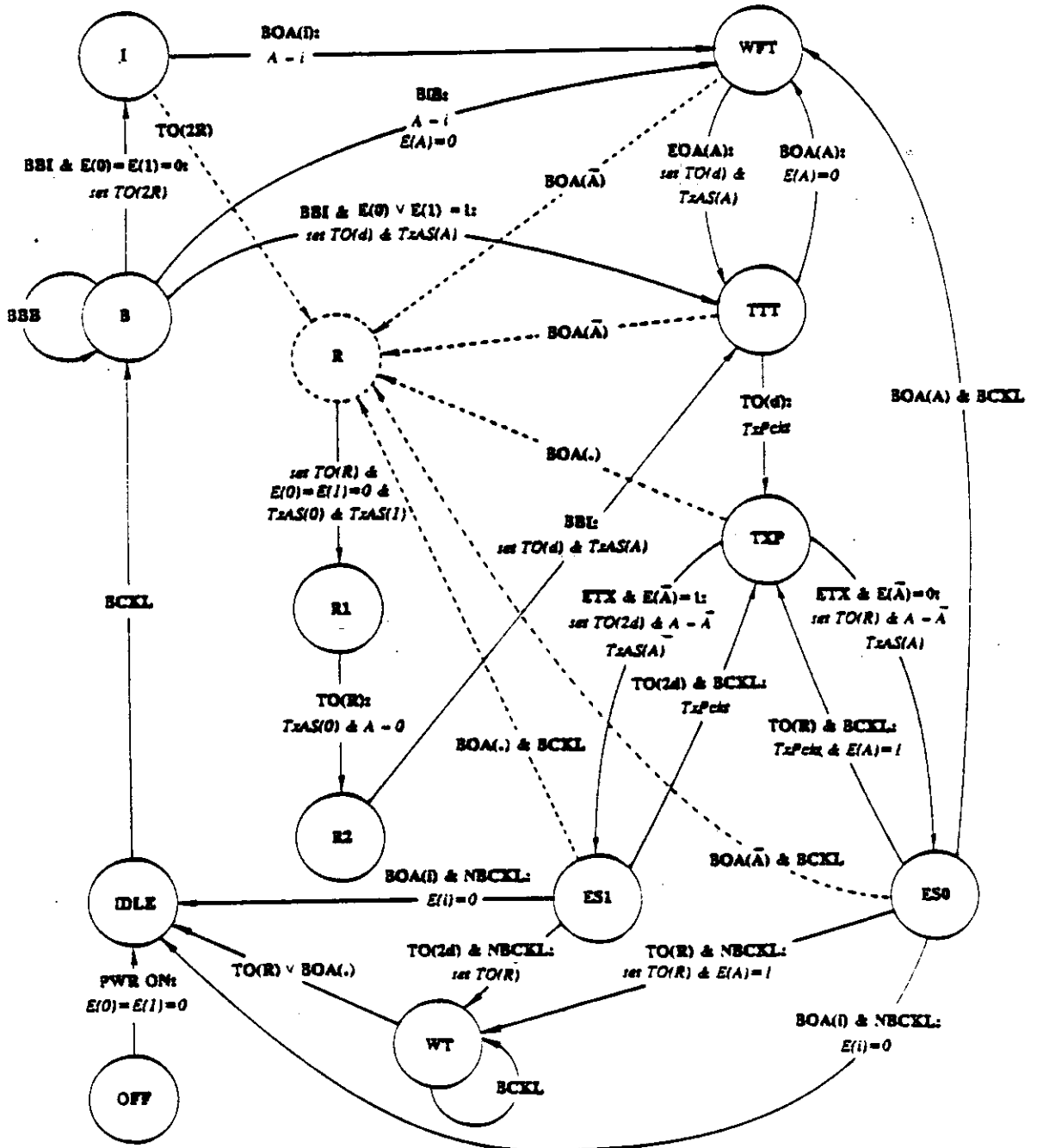


Fig. 5.8 - TLP-4 State Diagram.

There is no need to cause delay by starting recovery. The station simply remains in state **B** waiting for a channel to become idle. Then the station moves to state **WFT** synchronized on the busy channel. The busy channel flag is also reset to 0.

At heavy load, when all stations are active, **TLP-4** performs identically to **TLP-3**. There are no collisions or initializations. Overhead between rounds is kept at $2d$ seconds.

At light load, first stations go through the initialization procedure. However, the extreme station flag corresponding to the synchronizing channel is set to 1 after the first successful transmission on that channel, when the station is the momentarily extreme station. Subsequently, the station can access the network as in random mode, without any delay. If there are multipacket messages, packets will be transmitted successively with an interpacket gap of $2d$ seconds. While no collision occurs, stations access the network freely. Delay at light load is decreased to almost zero.

At intermediate load, **TLP-4** performance may degrade considerably. The token sweep may be confined to a smaller section of the network, so that a new backlogged outside station always causes collisions. Furthermore, a station may join the network synchronized by one channel, and if the station has a flag set for the other channel, it reverses the round at the end of its packet transmission, even if another station upstream is still active. This incorrect round reversal causes a collision with the upstream station transmission, triggering recovery and causing extra delay. Such inefficient processing is most common when the ratio π/T increases (T is the packet length) for equally distributed load and random traffic.

When leaving state B and going to state I, time-out ND set to $2R$ is optimal. As discussed in TLP-2, ND must be large enough to allow a station to find the network initially idle and acquire synchronization without starting an unnecessary recovery. Because rounds are reversed within $2d$ seconds, idle time in both channels is usually of $2d$ duration during a sequence of successive rounds where extreme station identities are constant. However, if a present extreme station is not extreme in the following round, the execution of the *round restart procedure* at the stations could lead to idle intervals as large as the worst idle intervals observed in TLP-2. Therefore, $ND = 2R$ is large enough to handle the worst situations explained in TLP-2.

Time-out ND should not be set less than $2R$. Monitoring the number of passes through recovery shows that the peak load of TLP-4 queueing delay coincides with the load that causes the maximum number of entries into recovery due to collisions. The number of passes through recovery due to network idle is negligible at light load (flags are set to 1 and stations transmit immediately) and heavy load (all stations always transmitting). At intermediate load, entries into recovery due to collision dominate completely. Simulation results show that decreasing the value of ND deteriorates TLP-4 performance for equally loaded network. Triggering recovery too soon wastes time because the station acquires sync in less time if another station is active. The number of stations satisfying $E(0) = E(1) = 0$ increases with load. At the limit, when ND is zero, stations having both flags at zero initialize the network without waiting for sync, if both channels are momentarily idle at packet arrival. Based on the above, the value of ND was set at $2R$.

TLP-4 always performs better than the other versions under conditions of light load, when delay becomes negligible. At heavy load, **TLP-4** and **TLP-3** both perform optimally. At intermediate load, unevenly distributed load and multipacket messages may cause **TLP-4** to outperform the other versions, as simulation results in Chapter 8 show. For very large networks, the improvement may be considerable.

5.4 RECOVERY AND JOINING

For all versions of **TLP**, the recovery procedure is executed by states **R1** and **R2**. **R** is a pseudo state which simplifies drawing state transitions to recovery. These transitions are drawn in dotted lines to distinguish them from transitions between regular states.

TLP protocols are structured so that stations sense only one channel busy at any one time. Furthermore, packet transmission is collision free and *BOA* events are only expected in the channel which is currently busy, or where the station is presently transmitting activity signal on.

Transition into **R1** is triggered by detection of simultaneous activity on both channels or upstream activity during packet transmission (collision). Either condition may be caused by station malfunctioning, or newly powered-on (**TLP-3** and **TLP-4**) or newly backlogged stations (**TLP-4**).

Newly active inside stations are always transparently absorbed by the network (the joining process occurs without extra overhead). State **ON** (**TLP-1**, **TLP-3**) or **B** (**TLP-2**, **TLP-4**) guarantees the correct behavior by moving the state to **WFT** when one of the channels is initially sensed busy. The variable *A*

is set to the busy channel.

A newly active outside station is still transparently absorbed in **TLP-1** and **TLP-2** because of delay R between rounds. This station detects end-of-train in the synchronizing channel, and its transmission reaches the current extreme station before the round is reversed. It then becomes the new extreme station.

However, in **TLP-3** and **TLP-4**, a newly active outside station only joins the active network after recovery is executed. The extreme station situated downstream to the joining station reverses the round before the joining station can transmit successfully. In both versions, the delay $2d$ in round reversal allows the new station to collide following its attempt to transmit at the end-of-train in the round. The joining station then starts the recovery process.

Stations perform recovery in a completely distributed fashion and a finite time. The following steps are executed during a recovery:

- (a) Detection of abnormal condition and transition into **R1**.
- (b) Transmission of activity on both channels for $R = 2\tau + 2d$ while in state **R1**. After R has expired, move to **R2**.
- (c) In **R2**, continue to transmit activity on channel L-to-R. After both channels are sensed idle, the station executes the standard procedure as if channel L-to-R had been initially detected busy. In **TLP-2** and **TLP-4** the state moves from **R2** to **TTT**, because a backlog always exists. In **TLP-1** and **TLP-3**, if a backlog exists, the state moves from **R2** to **TTT**, otherwise the state moves to **ES0**, where the station checks

whether or not it is an extreme station.

CLAIM: The above steps guarantee complete recovery within $2R + d$ seconds in the worst case.

PROOF:

Assume that station S_i is the first station to start recovery at time t_0 . Define $t_i[EVENT_j(A)]$ as the time that event $EVENT$ detected or originated at S_j tap on channel A reaches S_i tap on the same channel. Hence, $t_0 = t_i[BOA_i(.)]$.

S_i activity signal, transmitted on both busses, hits another station S_j at $t_j[BOA_i(.)] = t_0 + \tau_{ij}$. Here, if S_j is not yet in recovery, it moves to state **WFT** and waits for EOA in the normal procedure. Otherwise, S_j starts recovery. In the latter case, the activity signal transmitted on both channels by S_j starts at $t_j[BOA_j(.)] = t_0 + \tau_{ij} + d$. R seconds later, S_j activity on channel R-to-L stops, and S_j moves to **R2**. Any active station S_k , between S_i and S_j , not yet in recovery, moves into recovery at $t_k[BOA_j(.)] = t_0 + \tau_{ij} + \tau_{jk} + d$ when S_j activity signal is detected (observe that the other channel has been busy with S_i activity signal). In this event, S_k activity signal starts d seconds later and S_k activity on channel R-to-L stops at $t_k[EOA_k(RL)] = t_0 + \tau_{ij} + \tau_{jk} + 2d + R$, and S_k moves to **R2**.

Assume S_l , $l \leq i$, and S_r , $r \geq i$, are, respectively, the leftmost and the rightmost stations on recovery. S_l starts recovery at most by time $t_{R1} = t_l[BOA_l(RL)]$ and enters state **R2** by time

$t_{R2} = t_{R1} + d + R = t_0 + \tau_{ij} + d + R$. Nevertheless, S_l can only detect channel R-to-L idle by

$$\begin{aligned} t_{idle} &= \max \left\{ t_l \{EOA_k(RL)\} \mid S_k \text{ in recovery} \right\} \\ &= \max \left\{ t_0 + \tau_{ir} + \tau_{rk} + \tau_{kl} + 2d + R \mid l \leq k \leq r, S_k \text{ in recovery} \right\} \\ &= \left\{ t_0 + \tau_{ir} + \tau_{rl} + 2d + R \right\}, \end{aligned}$$

which depends only on the position of the extreme stations involved in recovery.

From the expressions above, $t_{idle} \geq t_{R2}$. Therefore, S_l starts packet transmission at most by $t_s = t_{idle} + d$. The worst case for t_s occurs for $i = l = 1$ and $r = N$. Therefore,

$$\max \left\{ t_s \right\} = t_0 + 2\tau + 3d + R = t_0 + 2R + d,$$

and complete recovery occurs within $2R + d$ from the detection of illegal events on the channels. ■

Now the need for state **WT** in **TLP-4** is explained. Assume for a moment that transitions to **WT** go directly to **IDLE**. Following recovery in **TLP-4**, if S_r has only one backlogged packet it can return to idle after leaving **ES0** and setting $E(RL) = 1$. However, if S_r receives another packet before it detects activity on channel L-to-R, S_r may transmit and cause another recovery before stations on left of S_l have the opportunity to transmit. This behavior may

repeat following each succeeding round, possibly preventing the low numbered stations from transmitting packets.

State **WT** prevents such a lock-up from occurring. After the transmission from the next active downstream station reaches S_r (at most R seconds after S_r leaves **ESO**), S_r is in synchronism again. Consequently, a station leaves **WT** and goes to **IDLE** after $BOA(.)$ has been detected or time-out R has expired. Time-out R is set when state **WT** is entered.

5.5 PERFORMANCE ANALYSIS

In all previously described protocols stations with backlogged packets transmit in sequential order from 1 to N and from N to 1 alternatively. This operation reduces the gap between two consecutive rounds but introduces differences in performance among the stations. In fact, the time needed to access the channel is dependent on the position of the station on the bus. If stations are uniformly spaced and traffic is balanced, only the central station receives access to the network at uniformly distributed time intervals. All other stations observe alternatively shorter and longer time intervals. This asymmetry in time access distribution introduces some unfairness in delay performance but does not affect station throughput which is the same for all stations.

Some symbols and assumptions used in the analysis are listed below:

N = number of stations connected to the network.

T = packet transmission time (includes preamble overhead) assumed constant.

$\tau_{ij} = \tau_{ji}$ = propagation delay between stations i and j . Stations are assumed to be uniformly spaced along the busses.

$\tau = \tau_{i1} + \tau_{iN}$ = end-to-end propagation delay on the bus.

S_r = righthmost active station

S_l = leftmost active station

S_i = i -th station

5.5.1 NETWORK UTILIZATION

Under equilibrium conditions, network utilization $S_i(M)$ is defined as the ratio between the time in a round spent for packet transmissions and the round duration, given that M stations are active and always transmitting in each round under TLP version i . The round duration, $R(M)$, defined as the time between the detection of the end of round at one end station and the detection of the next end of round at the other end station, is given by $R(M) = M(T + 2d) + T_{ES} + \tau_{lr}$, which is maximum when S_1 and S_N are the extreme stations ($\tau_{lr} = \tau_{1N} = \tau$).

Station reaction time is usually equal to a few bits of time and, therefore, $2d \ll T$. T_{ES} represents the time needed for a station to discover that it is an extreme station and is equal to the time-out set during the *round restart procedure*. In TLP-1 and TLP-2, T_{ES} is R seconds. In TLP-3 and TLP-4, T_{ES} may be assumed $2d$ at heavy load. At heavy load the identity of the extreme stations does not change, and no extra overhead is incurred due to the initialization procedure. The occurrence of errors and consequent recovery procedure activation is neglected in all protocol evaluations. Thus, in terms of $a = \pi T$:

$$S_{1,2}(M, a) = \frac{1}{1 + \frac{3a}{M}} \quad (8.1), \quad \text{and} \quad S_{3,4}(M, a) = \frac{1}{1 + \frac{a}{M}} \quad (8.2)$$

Maximum network utilization is achieved for $N = M$. Versions 1 and 2 perform identically because the worst case is assumed when stations 1 and N are the extreme stations. The comparison of the utilizations of different versions of TLP with other protocols is found in Chapter 6.

5.5.2 DELAY PERFORMANCE

Delay performance in this section is measured in terms of insertion delay (ID), as defined in Section 1.2.2.2. Analytical expression for ID at light (IDL) and heavy load (IDH) are derived, whereas results for general load are obtained in Chapter 6 by simulation and in Chapter 7 by analytical approximation.

5.5.2.1 LIGHT LOAD

In TLP-4 insertion delay is negligible. The first packet transmitted after power-on suffers a delay of $3R$ due to network initialization, but all subsequent packets are immediately transmitted after arrival. In case of multipacket messages, packets are transmitted with an interpacket gap of $2d$ seconds. The probability of collision during message transmission is assumed negligible.

In TLP-2 insertion delay is the time needed to initialize the idle network, which is $3R$. All single packets suffer this delay. In case of multipacket messages, the first packet suffers delay $3R$ and subsequent packets are transmitted with an interpacket gap of R seconds.

For TLP-1 and TLP-3 all stations are assumed powered-on. Therefore S_1 and S_N are the extreme stations. Consider station S_i . At light load, access instants for S_i are alternatively separated by x_i and y_i time intervals where

$x_i = n(x_i)(T + 2d) + T_{ES} + 2\tau_{1i}$; and $y_i = n(y_i)(T + 2d) + T_{ES} + 2\tau_{iN}$. $n(\cdot)$ represents the number of packets transmitted in the corresponding interval and can be assumed equal to be 0 at light load.

The average insertion delay for packets generated at station S_i at random points in time is:

$$\begin{aligned}
 IDL_i &= \frac{x_i}{2} \text{Prob}\{\text{arrival in } x_i\} + \frac{y_i}{2} \text{Prob}\{\text{arrival in } y_i\} \\
 &= \frac{x_i}{2} \frac{x_i}{x+y} + \frac{y_i}{2} \frac{y_i}{x+y} \\
 &= \frac{(T_{ES} + 2\tau_{1i})^2 + (T_{ES} + 2\tau_{iN})^2}{2(2T_{ES} + 2\tau)} \quad (5.3)
 \end{aligned}$$

Maximum IDL occurs at the end stations and minimum IDL occurs at the central station(s).

TLP-1 shows $\frac{3}{2}\tau \leq IDL_i \leq \frac{5}{3}\tau$, and TLP-3 shows $\frac{1}{2}\tau \leq IDL_i \leq \tau$, which demonstrates that the difference in IDL among stations is always less than $\tau/2$. Averaging over all stations yields:

$$IDL_{\text{TLP-1,3}} = \frac{1}{N} \sum_{i=1}^N IDL_i = \frac{1}{T_{ES} + \tau} \left[\frac{T_{ES}^2}{2} + T_{ES}\tau + \frac{\tau^2}{3} \left(2 + \frac{1}{N-1} \right) \right] \quad (5.4)$$

5.5.2.2 HEAVY LOAD

At heavy load stations always have a packet to transmit, and the time intervals between consecutive access rights at station S_i are alternatively

$$x_i = (2(N-i) + 1)(T + 2d) + T_{ES} + 2\tau_{iN} \quad \text{and} \quad y_i = (2(i-1) + 1)(T + 2d) + T_{ES} + 2\tau_{i1}.$$

The average insertion delay is:

$$IDH_i = \frac{x_i + y_i}{2} - T$$

$$= (N - 1)T + 2Nd + T_{ES} + \tau = IDH. \quad (5.5)$$

IDH is independent of station location and increases linearly with the number of stations. As expected, ID is bounded for any value of offered traffic.

5.6 CONCLUSIONS

This chapter describes four versions of Token-Less protocols designed for the dual unidirectional bus architecture. The control operation of the protocols is solely based on the detection of activity on the channel and is completely distributed. The circuitry needed at line speed is kept simple and small. Access is collision free and packet delay is bounded. Joining and recovery actions are analyzed and TLP behavior under adverse conditions is proved correct. Exact expressions for behavior at light and heavy load are derived.

CHAPTER 6

COMPARATIVE ANALYSIS AND SIMULATION RESULTS

6.1 INTRODUCTION

In this chapter we present a comparative analysis of the dual bus protocols previously introduced. For reference we also consider some of the protocols developed for the single unidirectional bus topology.

We start by deriving utilization and delay (IDL and IDH) expressions for Fasnets, Express-net, D-net and Ethernet, using the same assumptions with which expressions for the proposed protocols have been obtained. Next we compare utilization and insertion delay of all protocols for different values of network length, packet size and number of stations.

Because analytical results are constrained to light and heavy load, we utilize a discrete simulator to evaluate performance under different traffic conditions. The basic simulator is briefly explained and results for insertion delay versus utilization for Buzz-Net, U-Net, TLP and a variation of CSMA/CD are presented. Due to the adaptability nature of some of its versions, TLP covers all ranges of performance shown by other protocols, with some advantages specifically applicable to asymmetric traffic and load. In view of the above findings, our simulation efforts concentrate on the various versions of TLP under five different traffic and network conditions. We plot results for the insertion

delay (ID), queueing delay (QD) and utilization for the various versions. 95% confidence intervals are collected, and the value of the intervals as a percentage of the plotted average point are given for the most critical cases.

6.2 PERFORMANCE MEASURES FOR EXISTING PROTOCOLS

6.2.1 EXPRESS-NET, D-NET AND C-NET

For Express-net [Frat81] and D-net [Tsen82] the throughput and delay expressions can be derived following the same procedures as in U-Net (Session 2.4.2). The locomotive is assumed to be a burst of carrier of d seconds, where d is the station reaction time. Both protocols perform identically and their utilization and insertion delays are given by the following formulas:

$$S(i) = \frac{iT_r}{2\tau + 2d + i(T + d)}, \text{ for } i > 1. \quad (6.1)$$

$$IDL = \tau + \frac{3d}{2}, \text{ at light load.} \quad (6.2)$$

$$IDH(i) = MID(i) = 2\tau + 2d + i(T + d) - T, \text{ at heavy load.} \quad (6.3)$$

Obviously, the maximum utilization S is given by $S(N)$. For $NT \gg 2\tau$, the asymptotic utilization is T_r/T . For the usual case where $\tau \gg d$, $T \gg d$, and assuming $\alpha = \tau/T$ and $T \gg T_p$ we get:

$$S = S(N) = \frac{1}{1 + \frac{2\alpha}{N}}. \quad (6.4)$$

$$IDL = \tau, \text{ at light load.} \quad (6.5)$$

$$IDH(i) = 2\tau + (i - 1)T, \text{ at heavy load.} \quad (6.6)$$

For C-net, $IDL = 0$ and $IDH(i)$ is the same as above, except that MID , in the worst case, is greater than $IDH(N)$ [Mars81]. On the other hand, C-net throughput is the sum of the throughput given in (6.1) plus a fraction δ which depends on packet-length, transmission rate and physical location of stations. δ represents the contribution of successful transmissions between trains [Mars81]. Because of these nuances in performance, we proceed without comparing C-net with other protocols.

6.2.2 FASNET

Fasnet [Limb82] uses a synchronized approach with transmissions occurring in a slotted bus. Collisions do not occur nor is a preamble required for the data field.

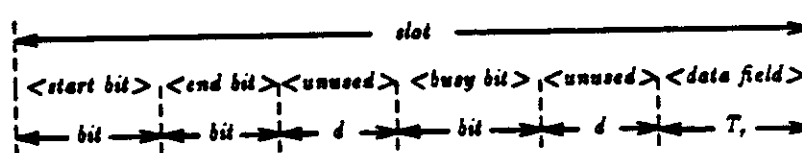


Fig. 6.1 - Fasnet slot.

The diagram of a slot in Fasnet is shown in Fig. 6.1 (not in scale), where we have assumed that a time equal to the reaction time of the station is representative of the length of the unused portions inside the slot. Ignoring single bit times, we assume that the slot transmission time T equals $T_r + 2d$. Following the derivations in [Limb82] the performance expressions for Fasnet are given by:

$$S(i) = \frac{iT_r}{2\tau + 2d + (i+1)T_r}, \text{ for } i \geq 1 \quad (6.7)$$

$$IDL = \tau + T_r/2, \text{ at light load.} \quad (6.8)$$

$$IDH(i) = MID(i) = 2\tau + 2d + iT_r, \text{ at heavy load} \quad (6.9)$$

Of course, $S = S(N)$. The term $T_r/2$ in IDL and the extra T_r in $S(i)$ account for the lack of synchronization between the two channels, which delays the out-of-band request for starting a new cycle. For $NT_r \gg 2\tau$, the asymptotic utilization is $N/(N-1)$. Assuming $N \gg 1$, and $T \cong T_r$, the utilization as a function of α is given by:

$$S = S(N) = \frac{1}{1 + \frac{2\alpha}{N}} \quad (6.10)$$

If we compare (6.10) and (6.4), we see that they are equal. Under most conditions, Fasnet and Express-net (or D-net) perform similarly. Fasnet was developed to have small fixed slots, and under that condition, IDL is affected very little by the lack of synchronization between the busses.

6.2.3 ETHERNET

Ethernet-like systems utilize CSMA-CD as the transmission protocol. At present CSMA/CD is one of the most frequently used protocols for LANs, although its performance degrades as the factor $\alpha = \tau/T$ increases and delay is unbounded. Nevertheless, we include Ethernet results as a motivation for the development of new protocols for high speed LANs. Configurations of Ethernet systems vary from bidirectional bus systems to star shape topologies as Fibernet

I and Fibernet II. However, all these different implementations perform the same, because they follow strictly the same CSMA-CD protocol.

At light load, IDL is negligible. Metcalfe and Boggs have calculated some performance parameters for Ethernet when stations pump data at heavy load [Metc76]. When N stations are transmitting, they assume an ideal retry mechanism where each station transmits with probability $1/N$. Activity in the bus is modelled as a succession of successful transmission and contention periods, although idle periods may happen during contention. Time is slotted, and slot time is 2τ . Transmissions only occur at the beginning of a slot. After a successful transmission, a delay τ is observed to clean the channel and allow equal access to all stations. They derive:

$$S(N) = \frac{T_r}{T + 2\tau f(N)}, \quad T \geq 2\tau, \quad (6.11)$$

where $f(N) = (1 - 1/N)^{N-1}$. For instance, $f(5)=1.44$, $f(10)=1.58$, $f(15)=1.63$, $f(50)=1.69$ and $f(100)=1.70$. $f(N)$ is interpreted as the number of slots devoted to contention prior to the acquisition of the ether by some station, when all N stations are transmitting at full load. As all stations are equally likely to acquire the ether, for an arbitrary station i , $E[IDH_i]$ (mean value of IDH_i) can be calculated as follows. If the selected station is successful (prob. $1/N$), then $E[IDH_i] = 2\tau f(N)$. However, if the selected station is unsuccessful (prob. $(N-1)/N$), it must wait for the mean acquisition time of the successful station plus $T + \tau$ plus another $E[IDH_i]$, given that contention periods are independent and equally distributed. Therefore:

$$E[IDH_i] = E[IDH] = 2\tau f(N) + (N-1)(T + \tau). \quad (6.12)$$

The above formula describes the mean value of IDH only. IDH is not bound, and in actual implementations packets are discarded after some number of unsuccessful retransmissions.

When a collision occurs in Ethernet, the transmission is aborted. Therefore, the preamble is only necessary to permit a station to adapt to the amplitude and phase of the new signal and extract timing information which enables signal recovery. Nevertheless, we assume the same T_p for all protocols.

To detect collision Ethernet requires that $T \geq 2\tau$. When $T < 2\tau$, bit padding forces the transmission time to 2τ . Under those conditions, Ethernet capacity can be expressed as:

$$S = S(N) = \frac{T_r}{2\tau(1 + f(N))} \quad (6.13)$$

6.3 UTILIZATION AND INSERTION DELAY COMPARISON

6.3.1 S vs α

Comparing the simplified expressions of S for TLP-3,4 (eq. 5.2), TLP-1,2 (eq. 5.1), U-Net (eq. 2.1), TDT-Net (eq. 2.2), Buzz-net (eq. 3.1) and Rato (eq. 4.1) with those derived above, we can categorize the protocols into six groups of equally maximum utilization. The groups are the following:

group 1 - TLP-3, TLP-4, U-net and TDT-Net.

group 2 - Express-net, D-net and Fasnet.

group 3 - TLP-1,2.

group 4 - Buzz-net.

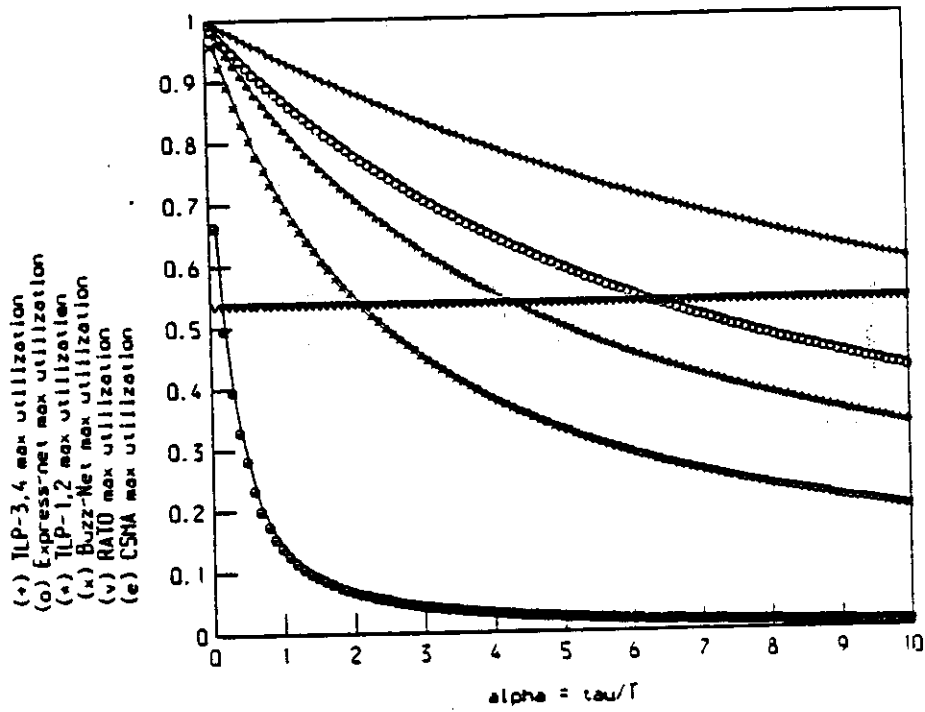


Fig. 6.2 - Utilization vs α for $N = 15$.

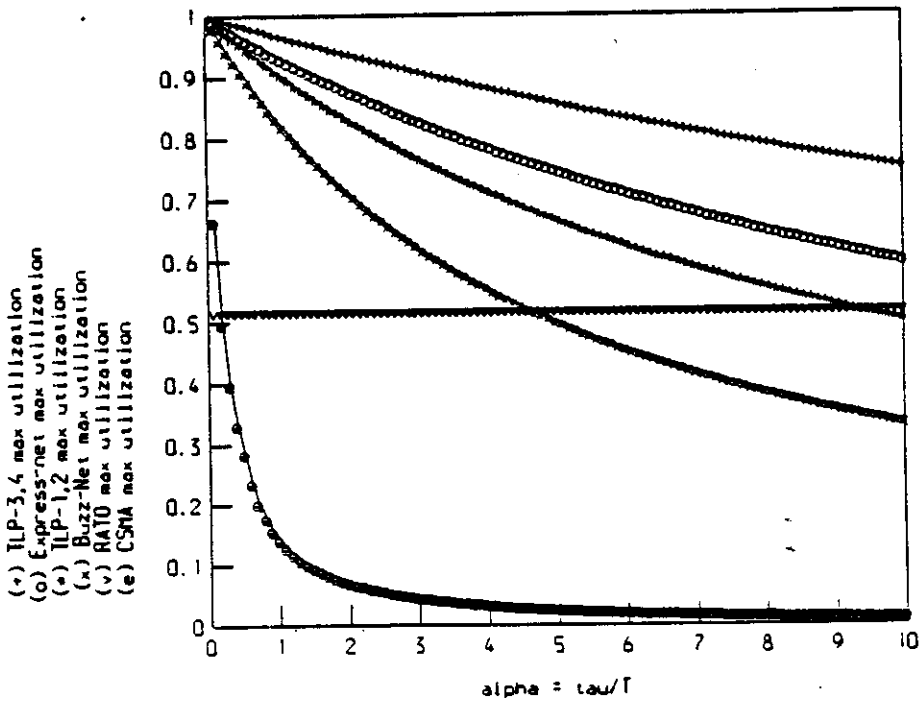


Fig. 6.3 - Utilization vs α for $N = 30$.

group 5 - Ethernet.

group 6 - Rato.

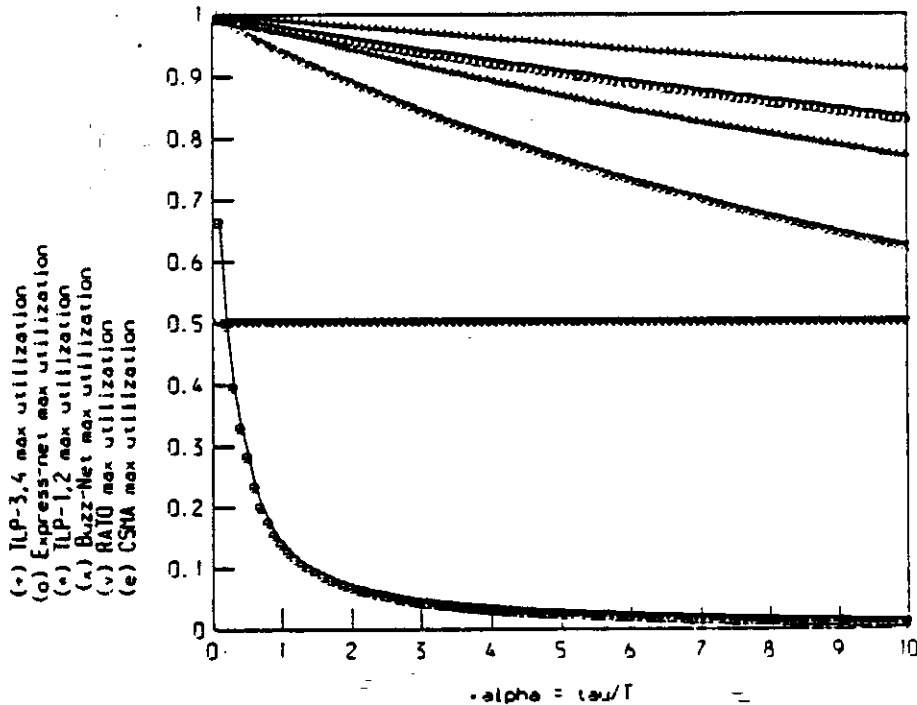


Fig. 6.4 - Utilization vs α for $N = 100$.

Figs. 6.2, 6.3 and 6.4 show the plot of utilization versus α for $N = 15, 30$ and 100 , respectively. Groups 1-5 are in decreasing order of maximum utilization. For groups 1-4 utilization improves as the number of stations increases, while Ethernet (group 5) is insensitive to changes in N . Ethernet only shows acceptable utilization when $\alpha \ll 1$. Rato is also insensitive to N and has a constant utilization of $\cong 0.5$ for all ranges.

6.3.2 S , IDL AND IDH

To develop a feeling for the absolute value of delay and throughput expected when using the protocols in actual implementations, we tabulated the

performance measures of the various protocols, assuming the following parameter selection:

speed of light in fiber (ν) = 2×10^8 m/s.
 transmission rate (G) = 1 Gbps.
 max station reaction time (d) = 20 ns.
 synchronizing slot (d_s) = 20 ns.
 token transmission time (T_k) = 100 ns.
 preamble transmission time (T_p) = 100 ns.
 packet length = 500, 1000 and 10,000 bits.
 network span (l) = 1 and 5 km.

TABLE 6.1							
N = 15, Span = 1 km, G = 1 Gbps, $\nu = 2 \times 10^8$ m/s							
packet length = 500 bits							
	TDT-net	U-net	Fasnet	D-net	Buzz-net	Rato	Ethernet
IDL(μs)	3.69	3.50	5.27	5.00	0	0	0
IDH(μs)	12.4	13.8	18.0	18.4	39.1	15.3	93.3*
S(5)	.32	.30	.19	.19	.07-.22	.16	.02
S(10)	.48	.44	.32	.31	.14-.22	.32	.02
S(15)	.58	.52	.42	.39	.19	.44	.02
packet length = 1000 bits							
IDL(μs)	3.69	3.50	5.52	5.00	0	0	0
IDH(μs)	19.4	20.8	26.0	25.4	46.1	27.8	100*
S(5)	.48	.47	.31	.38	.14-.29	.17	.04
S(10)	.65	.61	.48	.48	.24-.29	.35	.04
S(15)	.73	.68	.58	.57	.32	.49	.04
packet length = 10,000 bits							
IDL(μs)	3.69	3.50	10.0	5.00	0	0	0
IDH(μs)	145	147	170	151	172	253	226*
S(5)	.90	.90	.71	.83	.57-.62	.19	.41
S(10)	.95	.94	.88	.90	.74-.76	.38	.39
S(15)	.97	.96	.88	.93	.82	.53	.38

* mean value only.

Table 6.1 - Performance results for $N = 15$ and $l = 1$ km.

Results for $N = 15$ are shown in Tables 6.1 and 6.2, and results for $N = 100$ are shown in Tables 6.3 and 6.4. TLP-3 performs as U-Net and is a reference for the comprehensive comparison of TLP protocols in Section 6.4.3.

Our first observation notes that at this very high transmission rate, Ethernet performs very poorly at heavy load, even for packet lengths of 10,000 bits. This performance was expected from the results of the previous section. To improve Ethernet to the level of the other protocols, packet lengths at over 100,000 bits would have to be used, what is completely impractical. Ethernet has only negligible delay at light load. However, even at light load, we cannot

TABLE 6.2							
N = 15, Span = 5 km, G = 1 Gbps, $v = 2 \times 10^8$ m/s							
packet length = 500 bits							
	TDT-net	U-net	Fasnet	D-net	Buzz-net	Ratio	Ethernet
IDL(μs)	17.3	17.3	25.3	25.0	0	0	0
IDH(μs)	32.3	33.8	58.0	58.4	162	15.3	439*
S(5)	.09	.09	.05	.05	.02-.14	.16	.004
S(10)	.17	.16	.09	.09	.03-.14	.32	.004
S(15)	.23	.22	.13	.13	.05	.44	.004
packet length = 1000 bits							
IDL(μs)	17.3	17.3	25.5	25.0	0	0	0
IDH(μs)	39.3	40.8	66.0	65.4	169	27.8	445*
S(5)	.17	.16	.09	.09	.03-.24	.17	.008
S(10)	.28	.28	.16	.16	.06-.15	.35	.008
S(15)	.38	.36	.23	.23	.09	.49	.008
packet length = 10,000 bits							
IDL(μs)	17.3	17.3	30.0	25.0	0	0	0
IDH(μs)	165	167	210	191	295	253	572*
S(5)	.66	.66	.45	.50	.24-.34	.19	.08
S(10)	.80	.79	.62	.66	.39-.44	.38	.08
S(15)	.86	.85	.71	.74	.49	.53	.08

* mean value only.

Table 6.2 - Performance results for $N = 15$ and $l = 5$ km.

guarantee a bounded delay because of statistical fluctuations in input traffic.

U-Net and TDT-Net perform very similarly. Some differences are that $S(i)$ values for U-Net do not depend on N , while TDT-Net, for a few active stations in a large population, has a slightly lower throughput than U-Net due to

TABLE 6.3							
N = 100, Span = 1km, G = 1 Gbps, $v = 2 \times 10^8$ m/s							
packet length = 500 bits							
	TDT-net	U-net	Fasnet	D-net	Buzz-net	Rato	Ethernet
IDL(μ s)	4.72	3.40	5.27	5.00	0	0	0
IDH(μ s)	56.5	66.5	60.5	69.4	89.5	119	561*
S(5)	.26	.30	.19	.19	.07-.25	.02	.02
S(10)	.41	.44	.32	.31	.14-.23	.04	.02
S(15)	.51	.52	.42	.39	.19-.31	.06	.02
S(50)	.78	.69	.70	.63	.41-.44	.21	.02
S(100)	.88	.74	.83	.71	.55	.42	.02
packet length = 1000 bits							
IDL(μ s)	4.72	3.40	5.52	5.00	0	0	0
IDH(μ s)	106	116	111	119	129	216	610*
S(5)	.41	.47	.31	.38	.14-.29	.02	.04
S(10)	.58	.61	.48	.48	.24-.36	.05	.04
S(15)	.68	.68	.58	.57	.32-.40	.07	.04
S(50)	.88	.82	.82	.77	.58-.59	.23	.04
S(100)	.93	.85	.90	.83	.71	.46	.04
packet length = 10,000 bits							
IDL(μ s)	4.72	3.40	10.0	25.0	0	0	0
IDH(μ s)	97.0	1007	1020	1010	1030	1971	1501*
S(5)	.88	.90	.71	.83	.55-.62	.03	.41
S(10)	.93	.94	.83	.90	.71-.76	.05	.39
S(15)	.95	.96	.88	.93	.79-.83	.08	.38
S(50)	.99	.98	.96	.97	.93-.93	.25	.37
S(100)	.99	.98	.98	.98	.96	.50	.37

* mean value only.

Table 6.3 - Performance results for $N = 100$ and $l = 1$ km.

reservation slots overhead. However, TDT-Net utilization is almost always higher than U-net utilization, especially when packet length is small and all stations are active. This edge is a result of the lack of a large preamble in TDT-Net data packets. As packet size increases and transmission times become greater than τ , U-Net and TDT-Net perform similarly for all proportions of active stations.

Fasnet and D-net perform approximately the same. Their performance is always inferior to TDT-Net. U-Net always perform better than D-net and Fasnet, except when N is large and packet lengths are of small to medium duration. Those conditions are the ideal environment for Fasnet.

When the span of the network is large and the number of stations is small, Rato performs better than the other schemes if packet length is kept to a maximum. When packet length increases beyond a maximum, TDT-Net, Fasnet and D-net improve their performances and eventually surpass Rato.

For a single sending station, Buzz-net utilization approaches 1, because packets are sent consecutively without interference. However, in an equally loaded network, without this advantage, Buzz-Net performs poorly. This capacity for sending multipacket bursts over the net is also explored in TLP-4, which performs extremely well even when more than one station is sending (see Section 5.3.2.4).

To summarize the results in this section, we present, in Table 6.5, the best choice of protocols for the conditions depicted in Tables 1-4.

TABLE 6.4							
N = 100, L = 5 km, G = 1 Gbps, $v = 2 \times 10^8$ m/s							
packet length = 500 bits							
	TDT-net	U-net	Fasnet	D-net	Buzz-net	Rato	Ethernet
IDL(μs)	16.8	16.8	25.3	25.0	0	0	0
IDH(μs)	78.5	86.5	101	109	210	119	2610*
S(5)	.09	.09	.05	.05	.02-.17	.02	.004
S(10)	.16	.16	.09	.09	.03-.17	.04	.004
S(15)	.22	.22	.13	.13	.05-.23	.06	.004
S(50)	.49	.45	.33	.31	.14-.23	.21	.004
S(100)	.57	.57	.50	.45	.24	.42	.004
packet length = 1000 bits							
IDL(μs)	16.8	16.8	25.5	25.0	0	0	0
IDH(μs)	126	136	151	159	261	216	2659*
S(5)	.16	.16	.09	.09	.03-.20	.02	.008
S(10)	.27	.28	.16	.16	.06-.26	.05	.008
S(15)	.36	.36	.23	.23	.09-.26	.07	.008
S(50)	.66	.62	.49	.48	.24-.32	.23	.008
S(100)	.80	.73	.66	.63	.38	.46	.008
packet length = 10,000 bits -							
IDL(μs)	16.8	16.8	30.0	25.0	0	0	0
IDH(μs)	1017	1027	1060	1050	1161	1971	3547*
S(5)	.65	.66	.45	.50	.24-.36	.03	.08
S(10)	.79	.79	.62	.66	.38-.48	.05	.08
S(15)	.85	.85	.71	.74	.48-.55	.08	.08
S(50)	.95	.94	.89	.90	.75-.75	.25	.08
S(100)	.98	.96	.94	.94	.86	.50	.08

* mean value only.

Table 6.4 - Performance results for $N = 100$ and $l = 5$ km.

TABLE 6.5								
NETWORK PARAMETERS			PROTOCOL BEST CHOICE					
N	Span	Pckt (bits)	1	2	3	4	5	6
15	1 km	500	TDT-Net	U-Net	Rato	Fasnet	D-net	Buzz-Net
		1000	TDT-Net	U-Net	Fasnet, D-net		Rato	Buss-Net
		10,000	TDT-Net, U-net		D-net	Fasnet	Buzz-Net	Rato
	5 km	500	Rato	TDT-Net, U-net		Fasnet, D-net		Buzz-net
		1000	Rato	TDT-Net, U-net		Fasnet, D-net		Buzz-net
		10,000	TDT-Net, U-Net		D-net	Fasnet	Rato	Buzz-net
100	1 km	500	TDT-Net	Fasnet	U-net	D-net	Buzz-net	Rato
		1000	TDT-Net	Fasnet	U-net	D-net	Buzz-net	Rato
		10,000	TDT-Net, U-Net, Fasnet, D-net				Buzz-net	Rato
	5 km	500	TDT-Net	U-Net	Fasnet	D-net	Rato	Buzz-net
		1000	TDT-Net	U-Net	Fasnet	D-net	Rato	Buzz-net
		10,000	TDT-Net, U-Net, Fasnet, D-net				Buzz-net	Rato

Table 6.5 - Best choice of protocols.

6.4 COMPARATIVE ANALYSIS THROUGH SIMULATION

This section presents simulation results that supplement our understanding of protocol behavior and provide extra data for comparative analysis. The first subsection describes simulator implementation. The second subsection presents results for insertion delay for Buzz-net, U-Net, CSMA-CD, TLP-1, TLP-2, and TLP-3. The final subsection compares the four TLP versions in five examples under various traffic conditions.

6.4.1 DISCRETE EVENT SIMULATOR

To evaluate the protocols at intermediate load and varying traffic conditions, a discrete event simulator was written in C language. The basic primitives of the simulator assume an underlying dual bus topology with equally spaced stations. Later, we show how unevenly spaced stations are accommo-

dated. The simulator consists of two parts: a common core and a protocol specific, high-level language description. The common core handles the basic functions of the simulator: initialization, management of the event queue, traffic generation, collection of statistics, etc. The protocol description is a set of procedure calls which represent a modified diagram of states and transitions from the original protocol. Because our protocols are described by state diagrams, the transition to the modified diagram is simple, although some care is needed to ensure a one to one correspondence between the two. No automated reproduction or checking available is available, so the implementor must conduct the final debugging of the simulation program. The simulator includes a debug option that produces a detailed, selective list of actions at running time. Unix symbolic debuggers and screen editors provide support to the debugging phase. The simulation program is validated after a thorough checking of the event debug list for deterministic or quasi-deterministic situations. A deep understanding of the protocol operation is essential at this stage.

The simulator uses a global event queue for the entire system. Scheduled events are of two types: bus events and external events. Bus events processed for a station are always rescheduled in the event queue for the next successive stations. Scheduling for only the next successive stations avoids the potential need to delete numerous events scattered throughout the event queue. It also allows for simplified future expansion of the simulator to process multiple local networks interconnected by bridges. Fig. 6.2 shows the modified state diagram for Buzz-net, which corresponds to the state diagram in Fig. 3.1. In Fig. 6.2, $EVENT(i,j)$ means that station i scheduled the event for station j . The propagation direction is implied by the numbering order of the stations. For Buzz-net the bus event types are: EOP (*end of packet*), BOP (*beginning of packet*), EOB

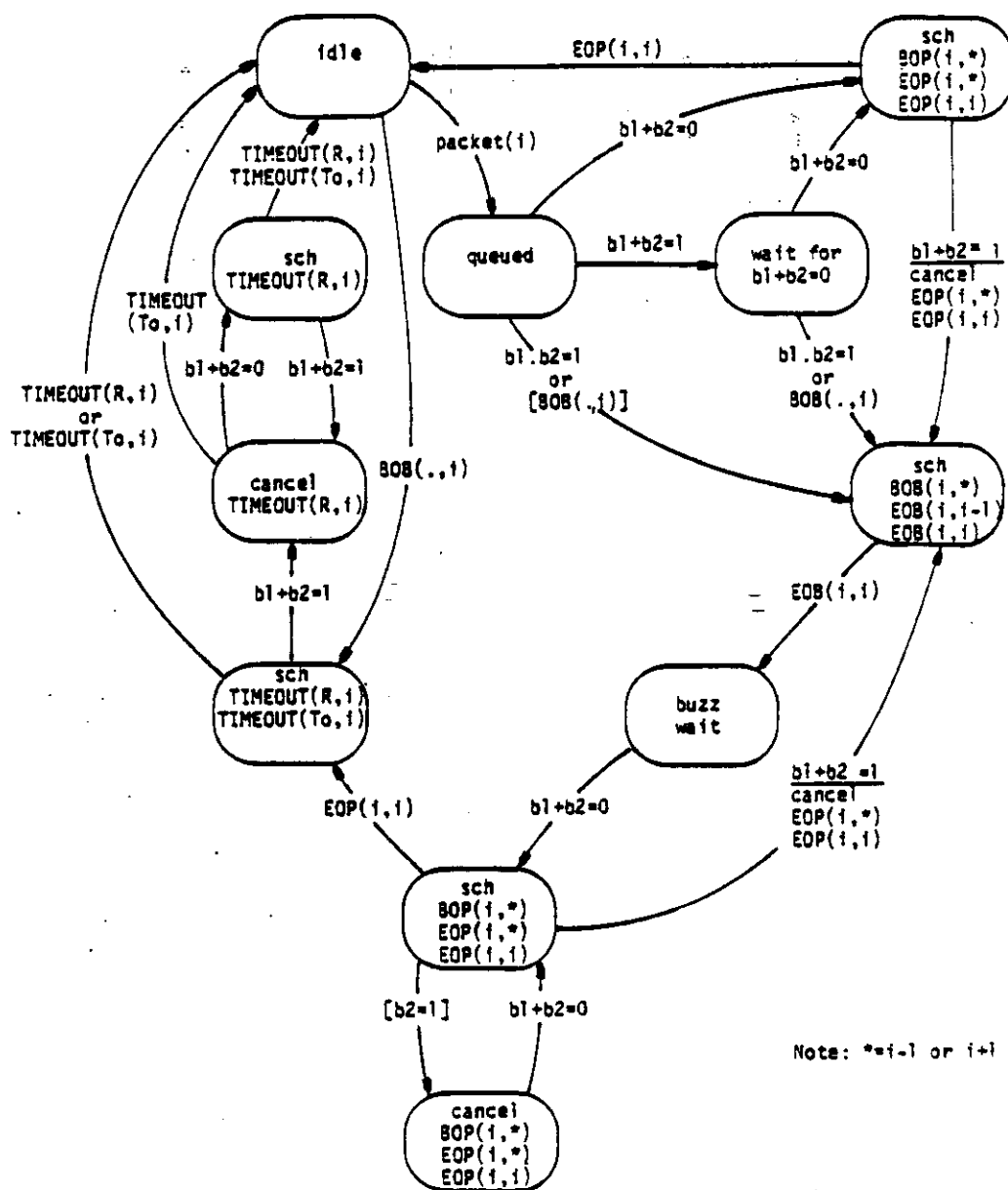


Fig. 6.5 - State Diagram for Buzz-Net Simulation.

(*end of buzz*) and BOB (*beginning of bus*). External events are associated with time-out conditions for a particular station. These events are represented as TIMEOUT(<*duration*>, <*station*>#).

A bus status variable is zero if the bus is idle, and one if the bus is busy. Status variables are updated at the beginning of event processing. Changes in status variables are coupled with corresponding events. Transitions are caused by events or can be arbitrary functions of status variables. Instantaneous state visits are possible if transitions out of the state are triggered by simultaneous events occurring when the state was entered. We use [event] to indicate that the transition only occurs if the state visit is instantaneous. Cancellations of events are necessary because of collisions (abortion of ongoing transmission).

The basic steps for event processing are:

EVENT (i):

 reschedule event for subsequent stations;

 update status variables;

 save time of event;

 IF a transition occurs

 THEN BEGIN

 update state;

 execute state;

 update time of state transition;

 END;

END OF EVENT (i);

Message length can be deterministic or exponentially distributed. A maximum allowable packet length forces long messages to be broken into packets, allowing multipacket traffic generation. Message interarrival time is deterministic or exponentially distributed. Therefore, combining the two possibilities for message length generation with the two possibilities for message interarrival time, four types of traffic can be generated. Stations can be assigned arbitrarily to one of four groups. Each group can be assigned a traffic type and a load level. Stations inside a group share the load equally. Load 0 can be assigned to a group to force a set of stations to be inactive. This way an uneven placement of stations can be simulated.

A set of supporting C programs and C-shell scripts allow the collection of 95% confidence intervals and automation of the simulation operation. Statistics collection start time and simulation end are defined in terms of the total number of packets transmitted. This primitive control is simple, but requires extra care to ensure that the statistics for individual stations are relevant. We used experimental runs to determine the end of the transient phase, and collected statistics for a minimum of about 1000 departures per station. Confidence intervals were collected by batch runs.

6.4.2 A GENERAL INSERTION DELAY COMPARISON

A network with 15 stations, end-to-end delay of 5 μ s (corresponding to a span of 1 km), fixed packet length of 1000 bits, exponentially distributed interarrival time and infinite buffer per station was simulated. The insertion delay is plotted against the utilization in Fig. 6.6. U-Net and TDT-Net were not simulated, but their behavior is similar to TLP-3.

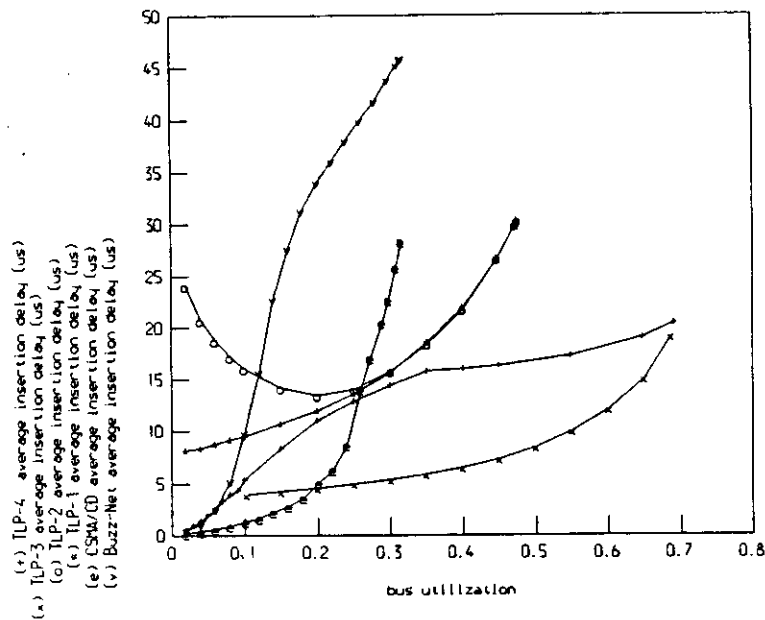


Fig. 6.6 - Insertion Delay vs Bus Utilization (span=1000m).

Although the original CSMA/CD protocol is not adequate to high speed LANs because it requires packet transmission time greater than the round trip delay for collision detection and fair access, a modified version of CSMA/CD adapted to the dual unidirectional bus topology was simulated. The CSMA/CD plotted corresponds to the following modification of the 1-persistent CSMA/CD. A packet is transmitted simultaneously in both busses, and collisions are recognized by detecting an incoming upstream packet during transmission. Note that, as a difference from bidirectional CSMA/CD in single bus, after a packet is successfully transmitted, it cannot be destroyed by any other station. In fact, stations always defer to an incoming packet. In case of collision, the Ethernet exponential binary backoff algorithm is used to randomize the retransmission delay, with no limit in the number of allowed retransmissions. If after the random delay, either bus is still sensed busy, the station persists sensing until both

busses are sensed idle (1-persistent CSMA/CD); the station then retransmits. When a transmission is successful, the station waits for a fixed delay ($= \tau$) before attempting to transmit again. Note that, in spite of the fact that $T \ll \tau$, the performance of this version of CSMA/CD is much better than the standard Ethernet CSMA/CD. However, since this random scheme does not show a bounded delay and TLP-4 and TLP-3 clearly offer higher throughput, it was not thoroughly investigated in this dissertation.

TLP-2 shows an increase in delay for very light traffic, because when all stations are back to idle the time consuming initialization procedure must be performed by all stations. TLP-2 does not show any improvement over TLP-1 because the load is evenly distributed among all stations. Buzz-net performance degrades as soon as collisions force the protocol into control mode, but insertion delay is kept bounded. TLP-3 offers better performance but has a constant delay at light load. TLP-4 performs like a random scheme at light load, but shows ID greater than TLP-3 as the utilization increases beyond $\cong .09$. The equally loaded network does not allow TLP-4 to take capitalise on its adaptability. In the next section, we identify traffic conditions that allow TLP-4 to perform better than TLP-3 over the whole input load range. Nevertheless, for the given example, TLP-4 performs better than the other schemes. We also observed that IDH , IDL and S for all schemes match the analytical predictions, giving us an indication that our simulation is valid and sound.

Because Buzz-Net performance seems to be lower bounded by TLP-4 performance over almost all utilization values, and U-Net performs as TLP-3, in the next section we concentrate our simulation efforts on the comparison of the TLP versions under various network conditions.

6.4.3 TLP SIMULATION RESULTS

To study differences in the performance of the various TLP versions, we selected five examples where network conditions favor the protocols differently. For all simulations we assumed a network with 15 stations ($N = 15$), infinite buffer per station, transmission rate of 1 Gbps, and fixed message length with message interarrival time exponentially distributed. The preamble in each packet was 100 bits. 95% confidence intervals were collected through batch runs, and experimental runs were used to identify the transient phase.

6.4.3.1 EXAMPLE 0: EQUALLY LOADED, SINGLE PACKET MESSAGE

In this example the influence of parameter $\alpha (= \tau / T)$ on the delay of TLP-3 and 4 is studied. We assumed an equally loaded network with a span of 10,000 m. Messages are single packets of 1000 bits (preamble not included). Figs. 6.7 and 6.8 show the insertion delay (ID) and queueing delay (QD), respectively, against bus utilization.

Comparing Fig. 6.7 with Fig. 6.6 from the previous section, we see that an increase in α is detrimental to TLP-4. In Fig. 6.7 we observe that the maximum insertion delay (MID) for TLP-4 occurs at some intermediate utilization, rather than the point of maximum utilization as before. However, from Fig 6.8 we note that this degradation in ID does not affect the queueing delay. Comparing the curves for TLP-3 and 4, we observe that TLP-3 performs better than TLP-4 for the equally loaded network except at light load, when TLP-4 offers negligible delay. The shape of the delay curves for TLP-3 are not affected by an increase

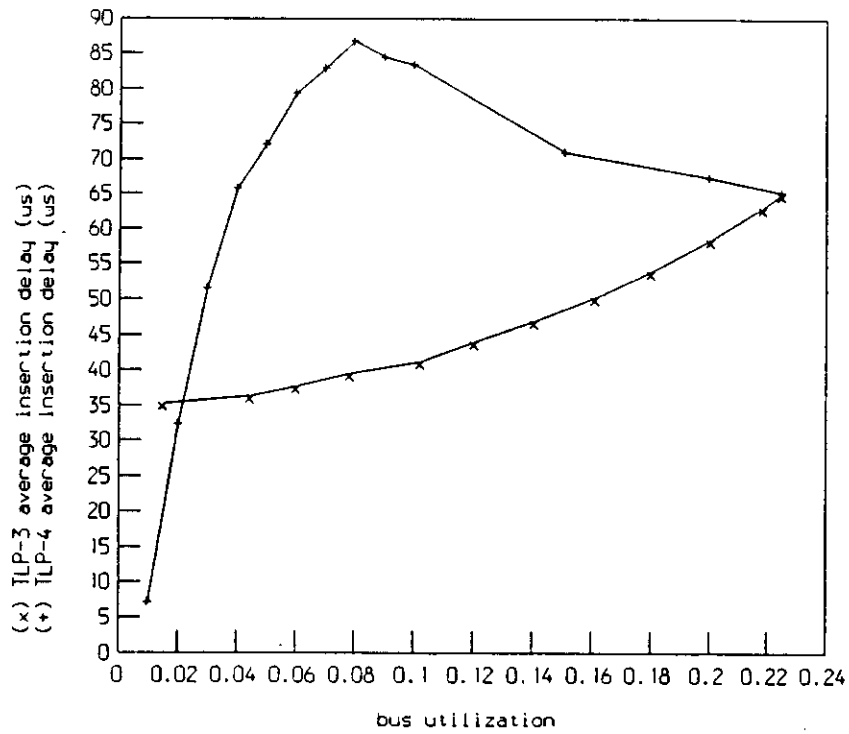


Fig. 6.7 - Ex.0: TLP-3,4 ID vs Bus Utilization (span=10,000m).

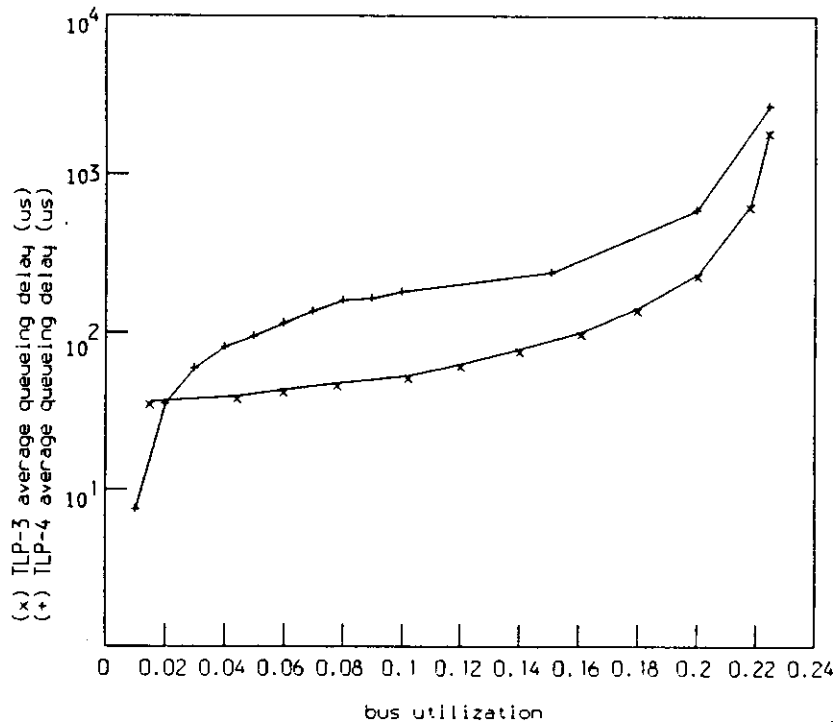


Fig. 6.8 - Ex.0: TLP-3,4 QD vs Bus Utilization (span=10,000m).

in α .

Because TLP-4 shows worse MID in a large network, for the next examples we assume a network span of 10,000 m.

6.4.3.2 EXAMPLE 1: SINGLE HEAVY LOADED STATION, SINGLE PACKET MESSAGE

In this example stations generate single packet messages of 1000 bits (preamble not included). Station 8 has increasing load, while the other stations offer a constant background load of 5 Mbps. Fig. 6.9 presents the insertion and queueing delays (ID and QD) for TLP-4. For TLP-1,2 and 3, station 8 delays are shown in Fig. 6.10 and the delays for background stations are shown in Fig. 6.11.

Among TLP versions, TLP-4 is clearly the best. Station 8 maximum utilization under TLP-4 is about 10-fold the maximum utilization achieved by TLP-3 (the next best). In TLP-4, station 8 ID is a decreasing function of the load for high load values, showing that the protocol gives all necessary bandwidth to the heavy load station without further overhead. Because insertion and queueing delays for background stations are practically equal and constant with offered load, station 8 traffic does not interfere with the performance of the other stations after their delay reaches the stable value.

Unlike the equally loaded case shown in Fig. 6.6, TLP-2 presents better queueing delay than TLP-1 when the load is more than ≈ 4.2 Mbps. That is because more throughput is given to the heavy load station without affecting the delay performance of the background stations. For all versions, insertion and

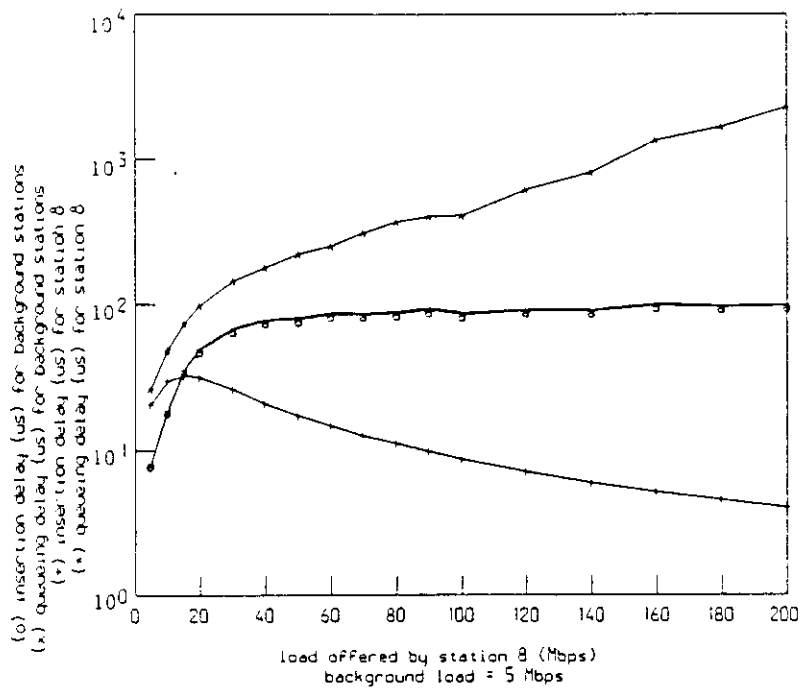


Fig. 6.9 - Ex.1: TLP-4 ID and QD vs Station 8 Load.

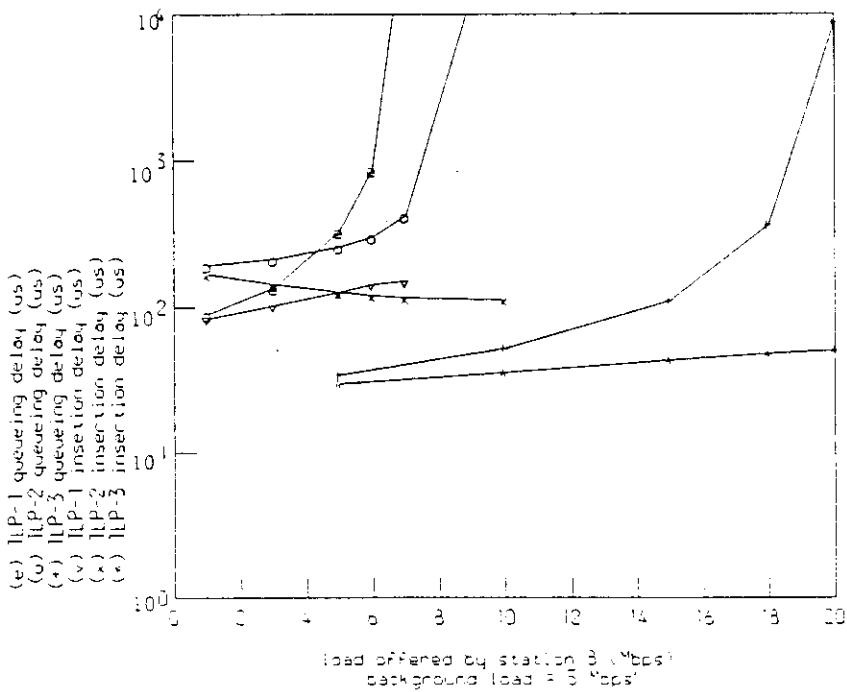


Fig. 6.10 - Ex.1: TLP-1,2,3 Station 8 Delays vs Station 8 Load.

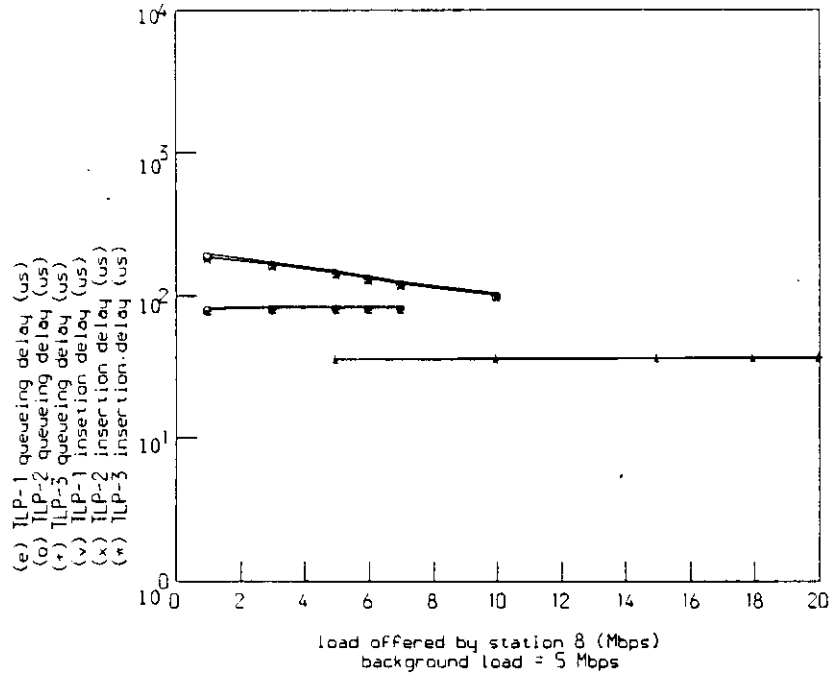


Fig. 6.11 - Ex.1: TLP-1,2,3 Background Delays vs Station 8 Load.

TABLE 6.6	
EXAMPLE 1	
PROTOCOL	MAX BUS UTILIZATION
TLP-1	0.012
TLP-2	0.014
TLP-3	0.024
TLP-4	>0.20

Table 6.6 - Ex.1: TLP-1,2,3,4 Maximum Bus Utilization.

queueing delays for the background stations remain approximately the same for all input loads. This behavior is a consequence of the bounded delay suffered by all packets and the light load condition where all the background stations are. At the background stations, when a packet arrives, the previous packet is guaranteed to have been transmitted. Therefore insertion and queueing delays are the same.

The maximum bus utilization measured via simulation for the various protocols is shown in Table 6.6. Confidence intervals for the queueing delay at station 8 are the most variable. In Table 6.7 we show the collected 95% confidence intervals. TLP-1, 2 and 3 show excellent results. Due to the adaptability of TLP-4, the confidence intervals tend to fluctuate greatly. However, the large values for confidence intervals observed at increasing load do not compromise our interpretations, because the TLP-4 performance is one order of

TABLE 6.7

EXAMPLE 1

95% CONFIDENCE INTERVALS FOR
QUEUEING DELAY AT STATION 8

TLP-1

Load (Mbps)	1-3	5	8	7	15
Conf. Int.	< 5%	6%	12%	10%	< 5%

TLP-2

Conf. Int.	< 5%
------------	------

TLP-3

Load (Mbps)	5-18	20	30
Conf. Int.	< 5%	8%	< 5%

TLP-4

Load (Mbps)	5-50	60	70	80-90	100	120	140	160	180	200
Conf. Int.	< 5%	7%	5.5%	8%	16%	12%	20%	32%	17%	49%

Table 6.7 - Ex.1: 95% Confidence Intervals for QD at Station 8.

magnitude above the other protocols.

6.4.3.3 EXAMPLE 2: SINGLE HEAVY LOADED STATION, MULTIPACKET MESSAGE

The effect of multipacket traffic on TLP's performance is investigated in this example. We assume the same conditions as in Example 1 except that the traffic offered by station 8 consists of messages 10,000 bits long. The messages are broken into 10 packets of 1000 bits (preamble not included) that are queued for immediate transmission. Message queuing delay equals the queuing delay of its last packet. Message insertion delay is the interval between the arrival of the first packet at the head of the output queue, and the start of the successful

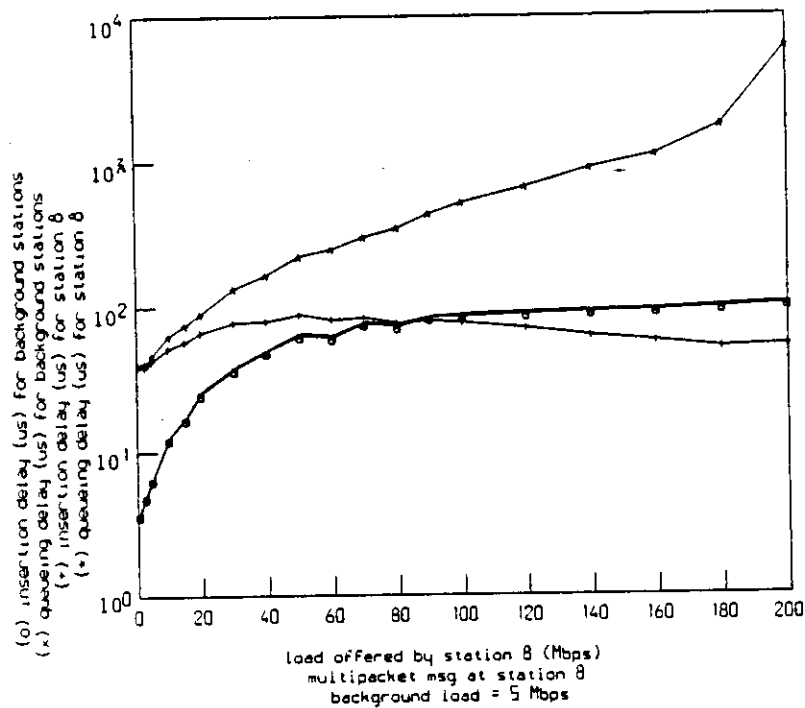


Fig. 6.12 - Ex.2: TLP-4 ID and QD vs Station 8 Load.

transmission of the last packet.

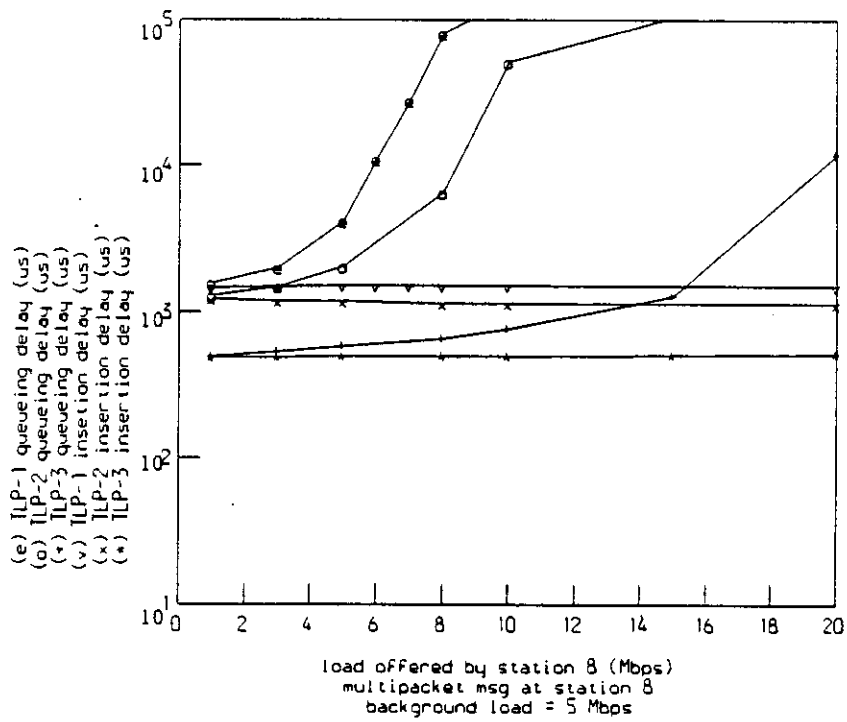


Fig. 6.13 - Ex.2: TLP-1,2,3 Station 8 Delays vs Station 8 Load.

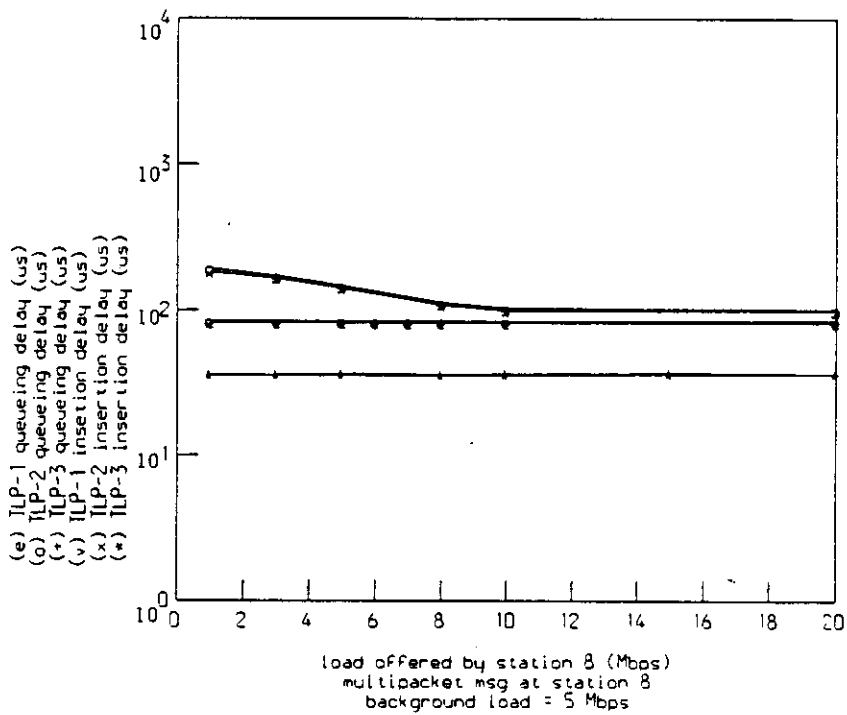


Fig. 6.14 - Ex.2: TLP-1,2,3 Background Delays vs Station 8 Load.

Fig. 6.12 shows insertion and queueing delay for TLP-4. Comparing Fig. 6.12 with Fig 6.9 from the previous example, we note that station 8 delay in the present case is little affected by the multipacket traffic, and, surprisingly, the delay for background stations even improve with the multipacket traffic. Although insertion delays in Figs. 6.12 and 6.9 have slightly different interpretations and are different, queueing delays are very close.

For TLP-1, 2 and 3, station 8 delays are shown in Fig. 6.13 and background stations delays are shown in Fig. 6.14. Compared with the single packet message case in Example 1, queueing delay at station 8 has increased by one order of magnitude, although background stations delays show no change in their order of magnitude. In Figs. 6.13 and 6.14 TLP-2 now performs better

TABLE 6.8	
EXAMPLE 2	
PROTOCOL	MAX BUS UTILIZATION
TLP-1	0.012
TLP-2	0.014
TLP-3	0.024
TLP-4	0.19

Table 6.8 - Ex.2: TLP-1,2,3,4 Maximum Bus Utilization.

than TLP-1 for the entire input load range.

Table 6.9 shows the collected 95% confidence intervals for queueing delay at station 8 and Table 6.8 shows the maximum bus utilization for the protocols. Comparing the maximum bus utilization in Tables 6.8 and 6.6 we observe that TLP-1,2 and 3 present the same values with single or multipacket heavy load traffic, while TLP-4 shows a slight decrease in bus utilization for the multipacket condition. The results for TLP-1,2 and 3 are expected. For those protocols only one packet is sent per round and thus the bus utilization is independent of the

TABLE 6.9

EXAMPLE 2

95% CONFIDENCE INTERVALS FOR
QUEUEING DELAY AT STATION 8

TLP-1

Load (Mbps)	1	3	5	8	7	8	10	20
Conf. Int.	< 5%	6%	19%	35%	41%	28%	8.5%	< 5%

TLP-2

Load (Mbps)	1-3	5	8	10	20-50
Conf. Int.	< 5%	6%	52%	23%	< 5%

TLP-3

Load (Mbps)	1-15	20	30-60
Conf. Int.	< 5%	43%	< 5%

TLP-4

Load (Mbps)	1	3	5	10	15	20	30	40	50
Conf. Int.	19%	10%	6%	21%	10%	8%	14%	5%	8%
Load (Mbps)	60	70	80	90	100	120	140	160	180
Conf. Int.	12%	10%	13%	8%	12%	6%	39%	7%	68%

Table 6.9 - Ex.2: 95% Confidence Intervals for QD at Station 8.

number of packets per message and depends on the total offered load only. In TLP-4 a multipacket message is sent as successive packet transmissions if background stations have no packet to transmit. However, the longer activity of multipacket message transmission increases the probability of collision with background traffic. The overhead of resynchronizing the cycles for transmission of collided packets accounts for the small loss in bus utilization observed in TLP-4.

The performance achieved by TLP-4 in the latter two examples is unmatched by any other LAN protocol, placing TLP-4 in a unique class for LAN protocols.

6.4.3.4 EXAMPLE 3: EQUALLY LOADED NETWORK, SMALLER ACTIVE SET

This example investigates TLP performance when the set of active stations is smaller than the total number of stations, or equivalently, when stations are not symmetrically located in the network. We assume that stations 8 to 15 are inactive. The load is equally distributed among the active stations, and mes-

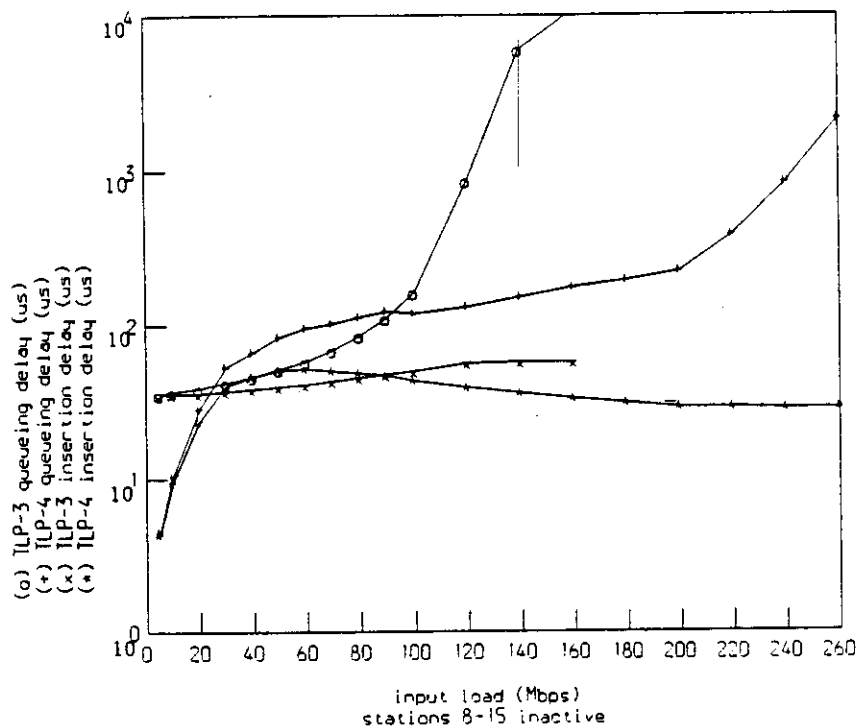


Fig. 6.15 - Ex.3: TLP-3,4 ID and QD vs Input Load.

sages are single packets of size 1000 bits (w/o preamble).

Fig. 6.15 shows results for TLP-3 and 4, and results for TLP-1 and 2 are shown in Fig. 6.16. Table 6.10 shows the maximum bus utilization achieved by the protocols.

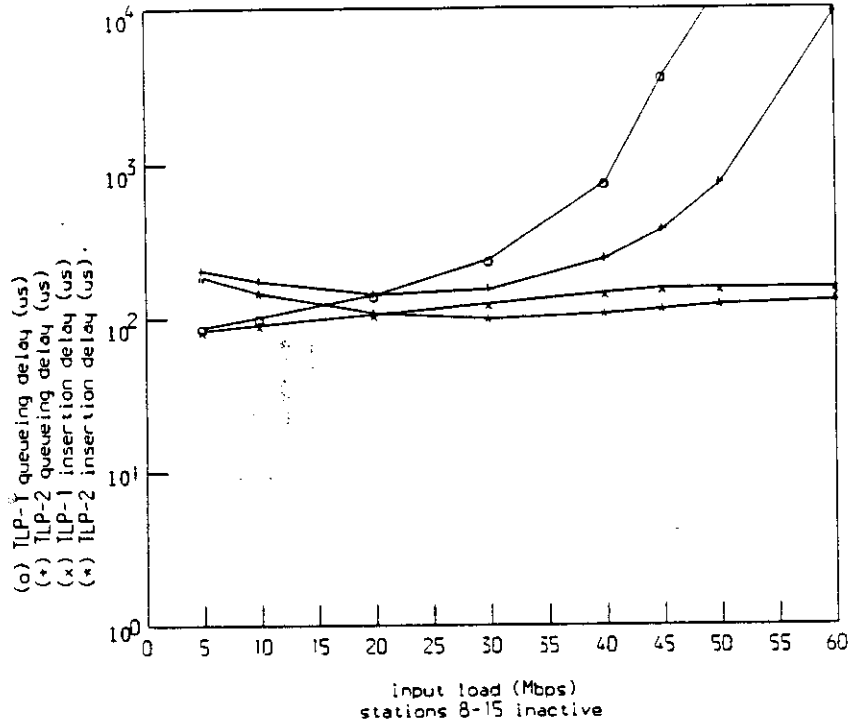


Fig. 6.16 - Ex.3: TLP-1,2 ID and QD vs Input Load.

Comparing Fig. 6.15 with Fig. 6.8, the maximum utilization for TLP-4 increases as the active set is reduced, but the maximum utilization decreases for TLP-3. TLP-3 is not adaptive, so a fewer packets per cycle must share the same round trip propagation delay overhead which remains constant independently of the size of the active set. Once again TLP-4 adapts well to new conditions. TLP-3 slightly outperforms TLP-4 in the range $\approx 25 - 95$ Mbps. At light load TLP-4 ID approaches 0, while TLP-3 ID stops improving around 35 Mbps ($\approx 2/3$ of the end-to-end trip delay, as expected).

Fig. 6.16 shows that the maximum utilizations for both TLP-1 and 2 are very limited. TLP-2 performs better than TLP-1 for loads higher than 20 Mbps. TLP-2, therefore, adapts better to a smaller active set. Table 6.11 shows 95% confidence intervals for the queuing delay averaged over all active stations.

TABLE 6.10	
EXAMPLE 3	
PROTOCOL	MAX BUS UTILIZATION
TLP-1	0.044
TLP-2	0.054
TLP-3	0.12
TLP-4	0.24

Table 6.10 - Ex.3: TLP-1,2,3,4 Maximum Bus Utilization.

TABLE 6.11

EXAMPLE 3

95% CONFIDENCE INTERVALS FOR
QUEUEING DELAY AVERAGED OVER ALL ACTIVE STATIONS

TLP-1

Load (Mbps)	5-30	40	45	50	60
Conf. Int.	< 5%	16%	36%	15%	6%

TLP-2

Load (Mbps)	5-30	40	45	50	60	70
Conf. Int.	< 5%	6%	10%	20%	17%	7%

TLP-3

Load (Mbps)	5-90	100	120	140	160
Conf. Int.	< 5%	10%	37%	15%	7%

TLP-4

Load (Mbps)	5	10	20	30-40	50-70	80-100	120	140	160	180	200	220	240	260
Conf. Int.	13%	11%	8%	6%	5.5%	7%	9%	13%	20%	27%	21%	32%	38%	24%

Table 6.11 - Ex.3: 95% Confidence Intervals for QD Averaged Over All Stations. Again, we observe that the larger confidence intervals at increasing load (> 160 Mbps) for TLP-4 do not compromise our analysis, because the other protocols fail to perform closer.

6.4.3.5 EXAMPLE 4: SINGLE HEAVY LOADED STATION, SMALLER ACTIVE SET

We now examine the case where, for the reduced set of active stations 1 to 7, station 4 has increasing load while the other stations stay in the background offering a collective load of 5 Mbps. Messages are still single packets of

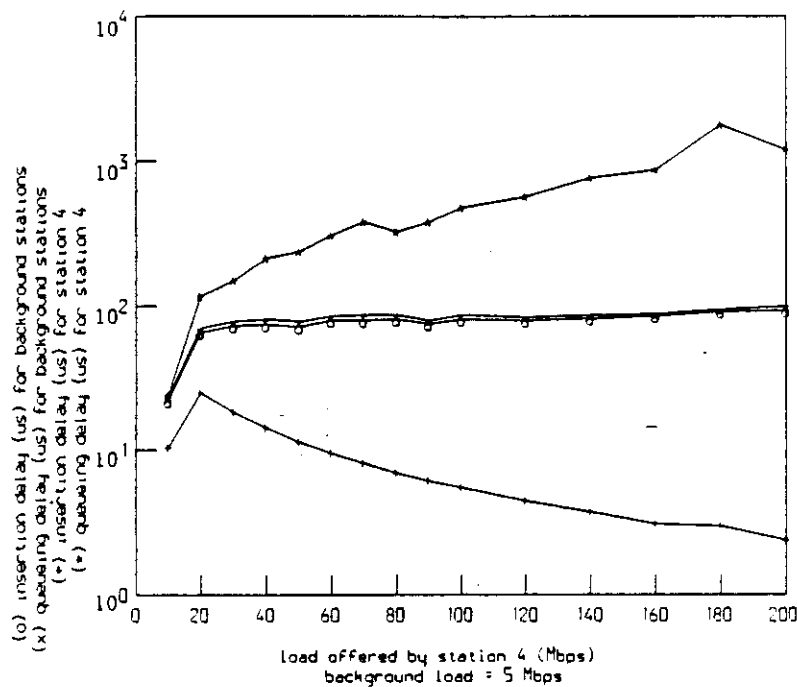


Fig. 6.17 - Ex.4: TLP-4 ID and QD vs Station 4 Load.

1000 bits.

Fig. 6.17 shows the results of insertion and queuing delay for TLP-4. Fig. 6.18 presents the results of station 4 delay for TLP-1, 2 and 3. The delay for the background stations under TLP-1, 2 and 3 is shown in Fig. 6.19. The maximum bus utilization for the various protocols is given in Table 6.12. 95% confidence intervals for the queuing delay at station 4 is shown in Table 6.13.

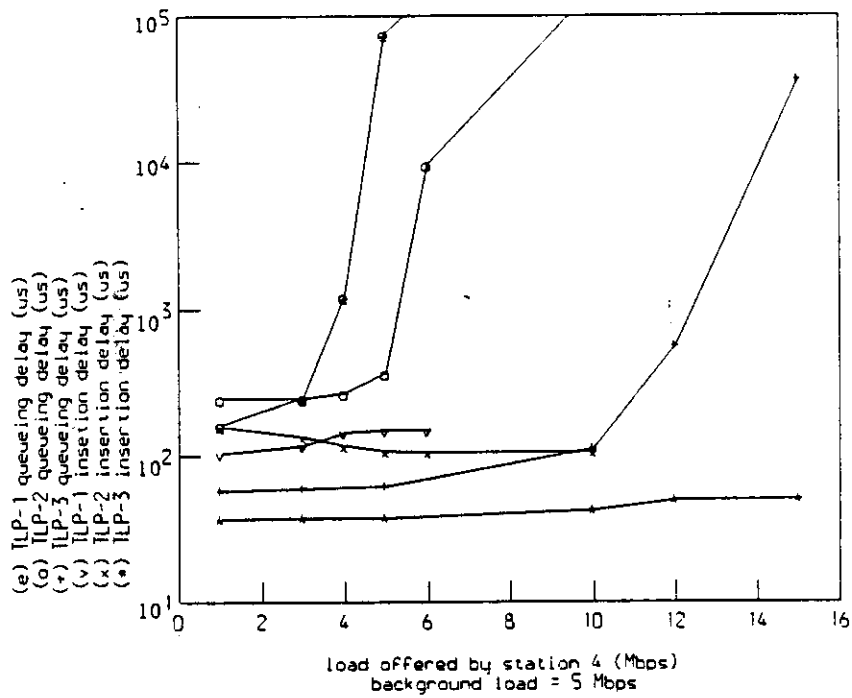


Fig. 6.18 - Ex.4: TLP-1,2,3 Station 4 Delays vs Station 4 Load.

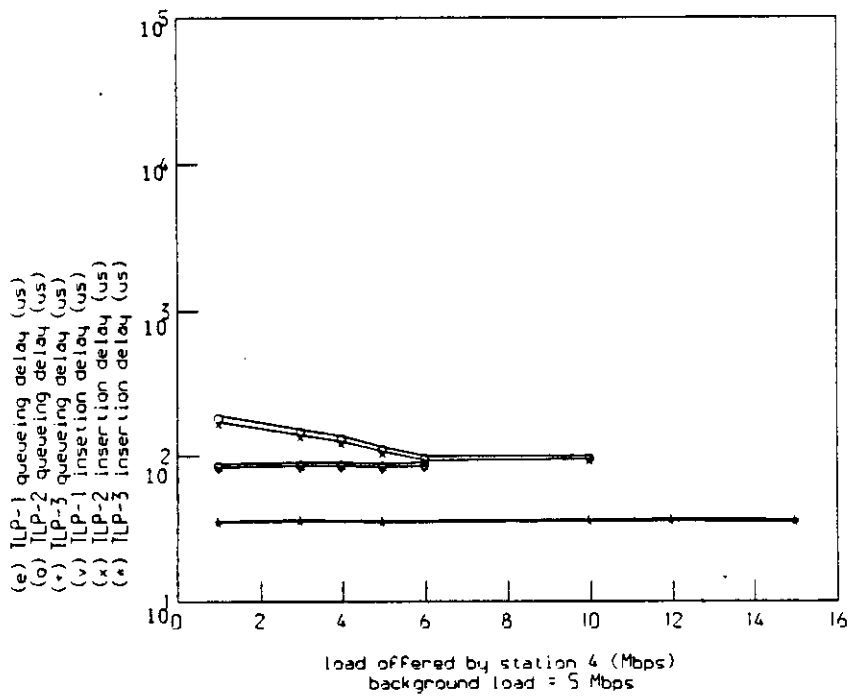


Fig. 6.19 - Ex.4: TLP-1,2,3 Background Delays vs Station 4 Load.

TLP-4 is by far the best. TLP-4 maximum utilization is more than 10 times greater than the maximum utilization achieved by TLP-3 (the next best), and at light load TLP-4 ID tends to 0 while the ID for the other protocols levels off or increases as for TLP-2. TLP-2 shows lower delay than TLP-1 for increasing load, what is a direct consequence of its adaptivity to a smaller active set and single heavy load station. As in Example 2 with one single heavy loaded station, background stations are not noticeably affected by the heavy traffic on

TABLE 6.12	
EXAMPLE 4	
PROTOCOL	MAX BUS UTILIZATION
TLP-1	0.009
TLP-2	0.011
TLP-3	0.17
TLP-4	>0.20

Table 6.12 - Ex.4: TLP-1,2,3,4 Maximum Bus Utilization.

the network.

6.4.3.6 COMPARING TLP VERSIONS

TLP-3 and 4 decisively outperform TLP-1 and 2 in all examples. For equally loaded and symmetrically spaced stations, TLP-3 outperforms TLP-4 and TLP-1 outperforms TLP-2. However, single heavy load station and asymmetrical placement favor the adaptive versions TLP-2 and 4.

TLP-4 is the only version to show no deterioration of performance for single heavy loaded multipacket traffic. The achieved maximum utilization of TLP-4 in this cases is more than 10 times better than the next best maximum utilization (TLP-3), and queueing delay at the heavy load station is not affected by traffic type (single or multipacket).

TABLE 6.13

EXAMPLE 4

95% CONFIDENCE INTERVALS FOR
QUEUEING DELAY AT STATION 4

TLP-1

Load (Mbps)	1-3	4	5-10
Conf. Int.	< 5%	26%	< 5%

TLP-2

Load (Mbps)	1-4	5	6	10-15
Conf. Int.	< 5%	9%	17%	< 5%

TLP-3

Load (Mbps)	1-10	12	15-40
Conf. Int.	< 5%	37%	< 5%

TLP-4

Load (Mbps)	10	20	30	40	50	60	70-80	90	100	120	140	160	180	200
Conf. Int.	15%	8.5%	10%	< 5%	15%	23%	18%	10%	17%	26%	33%	43%	37%	44%

Table 6.13 - Ex.4: 95% Confidence Intervals for QD at Station 4.

For all protocols, background stations are unaffected by the heavy load traffic. This tolerance is valuable because it guarantees a fair share of resources. Furthermore, bounded delays are inherent to all protocols.

The fact that TLP-4 provides queueing delays of some order of magnitude less than other simple polling protocols like TLP-3 makes TLP-4 an excellent choice for applications where bursty, high bandwidth traffic occurs (file transfers, graphics, etc.). The insensitivity of the background traffic to the heavy load use of the network guarantees proper service to interactive and priority traffic.

CHAPTER 7

APPROXIMATE ANALYSIS FOR OSCILLATING POLLING

7.1 INTRODUCTION

In this chapter we derive an approximate solution to the queueing delay for the oscillating polling scheme which characterizes both TLP-3 and U-Net. This solution assumes that load is equally distributed among all stations and only one packet is transmitted per polling instant (transmission scheme also called "chaining").

The approximation is heavily based on the assumption that, for any two stations S_i and S_j , S_i transmissions are independent of S_j transmissions. This independence assumption permits an easy formulation of the Laplace-Stieljes (LT) transform of the time between polling instants at a station based on the probability that a packet is present when a station is polled. This same approach was used by Heyman [Heym83] and Lehoczky [Leho81] to obtain approximate solutions for the regular polling scheme where a cycle consists of polling stations in the order $\{1,2,\dots,M\}$. In TLP-3, however, a cycle consists of polling stations in the order $\{1,\dots,N,N,\dots,1\}$. Because of above polling order, we call this scheme oscillating polling.

Different from what occurs with regular polling in a symmetric network, with oscillating polling the performance is not uniform over all stations. Intervals between transmission opportunities for a station vary depending on its location in the network. Only the central station, in the case of equally loaded network and symmetrically spaced stations, has interpacket transmission instants equally distributed. So, we do expect stations to present different queueing delays depending on their position. The asymmetric behavior of stations in oscillating polling represents a major obstacle when trying to derive exact solutions. In fact chaining, i.e. single packet transmissions, complicates matters even further. No exact solution has been obtained for the chaining scheme, even for regular polling. Therefore, an exact solution for oscillating polling under chaining must be ruled out.

The exhaustive (as opposed to "chained") model for oscillating polling has been studied by Eisenberg [Eise72], and a gated model has been investigated by Swartz [Swar81]. Solutions in both cases are not in closed form and calculations become intractable for large numbers of stations.

7.1.1 THE MODEL

Packets are assumed to arrive at each station according to a Poisson process with rate λ . Packet transmission time (including overhead) is a constant T . The propagation delay between stations i and j is τ_{ij} . τ is the end-to-end propagation delay. N is the number of stations. Cycle c_i at station i , is defined as the time between returns of the virtual token to station i , on the same bus. Because the system is cyclic and each station has exactly two opportunities to transmit in each cycle, in the long run the cycle length distribution will be

independent of the reference station.

Transmission instants for station i are t_{i1} when the token is travelling from i to 1, and t_{iN} when the token is travelling from i to N where 1 and N are the two end stations. Subcycle c_{i1} is defined as the period of time starting at t_{i1} and lasting until next t_{iN} , and subcycle c_{iN} as the period of time starting at t_{iN} and lasting until next t_{i1} . In equilibrium, the distributions for c_{i1} and c_{iN} are expected to exist. The LT for c_{i1} and c_{iN} are, respectively, $C_{i1}^*(s)$ and $C_{iN}^*(s)$. To evaluate these LT's some simplifying assumptions are needed.

The major simplifying assumption for the model is that station i transmissions occur independently of station j transmissions, whenever $i \neq j$. Moreover, S_i transmissions at t_{i1} and t_{iN} are assumed to occur independently.

Under the above assumptions, packet transmission at instant t_{i1} occurs with probability b_{i1} and packet transmission at instant t_{iN} occurs with probability b_{iN} . Probabilities b_{i1} and b_{iN} are the key parameters to be determined.

If probabilities b_{i1} and b_{iN} are known for all stations, then $X_{i1}^*(s) =$ LT of service time at t_{i1} and $X_{iN}^*(s) =$ LT of service time at t_{iN} are obtained as follows:

$$X_{i1}^*(s) = (1-b_{i1}) + b_{i1} e^{-sT}, \quad (7.1)$$

$$X_{iN}^*(s) = (1-b_{iN}) + b_{iN} e^{-sT}. \quad (7.2)$$

Given that service times at the various stations are assumed to be independent random variables, $C_{i1}^*(s)$ is the product of the LT of the service time at t_{i1} times the LT of service times at t_{j1} and t_{jN} , $j < i$, times the LT of the round trip propagation delay between S_1 and S_i . Thus,

$$C_{i1}^*(s) = e^{-2\tau_{i1}s} X_{i1}^*(s) \prod_{j=1}^{i-1} X_{j1}^*(s) X_{jN}^*(s), \quad (7.3)$$

Similarly, $C_{iN}^*(s)$ is expressed as:

$$C_{iN}^*(s) = e^{-2\tau_{iN}s} X_{iN}^*(s) \prod_{j=i+1}^N X_{j1}^*(s) X_{jN}^*(s). \quad (7.4)$$

Also,

$$C^*(s) = C_{i1}^*(s) \cdot C_{iN}^*(s) = e^{-2\tau s} \prod_{j=1}^N X_{jN}^*(s) X_{j1}^*(s), \quad (7.5)$$

where $C^*(s)$ is the LT of a complete cycle and is independent of the reference station.

The expressions in (7.3) and (7.4) depend only on the round trip delay between station i and the end stations. This indicates that the stations could be unevenly separated in each bus and the expressions would still be valid.

The LT's calculated in (7.3), (7.4), and (7.5) are fundamental in obtaining the queueing delay at each station, because they allow the computation of the z-transform of the number of packets found in the system by a random arrival. As these LT's are completely defined by b_{i1} and b_{iN} , the determination of these probabilities is the key step in this analysis.

Expressions similar to those above were derived by Heyman [Heym83] for regular polling. In that case, however, the probabilities b_{i1} and b_{iN} merge into a single b_i because of symmetry. Then, b_i can then be approximated by the long run probability that the system is busy, which is easily obtained. In this case the solution is not as simple, and the procedure to determine b_{i1} and b_{iN} is

explained in the next subsection.

7.1.1.1 DETERMINATION OF b_{i1} AND b_{iN}

To obtain b_{i1} and b_{iN} , the behavior of the queue size at instants t_{i1} and t_{iN} for the system in equilibrium is studied.

Define:

$Q_{i1}(z) = z$ -transform of the number of packets in station i at t_{i1} ,

$Q_{iN}(z) = z$ -transform of the number of packets in station i at t_{iN} .

Define the following probabilities:

$p_{i1}(k) = \text{Pr}\{k \text{ packets present in station } i \text{ at } t_{i1}\}$,

$p_{iN}(k) = \text{Pr}\{k \text{ packets present in station } i \text{ at } t_{iN}\}$.

Observe that $p_{i1}(0) = 1 - b_{i1}$ and $p_{iN}(0) = 1 - b_{iN}$. For ease of notation, further define:

$$b_i = b_{i1} + b_{iN},$$

$$\bar{c}_{i1} = E[c_{i1}], \quad \bar{c}_{iN} = E[c_{iN}],$$

$$A(z) = C_{i1}^*(\lambda - \lambda z), \quad B(z) = C_{iN}^*(\lambda - \lambda z),$$

$$C(z) = A(z) B(z)$$

$$= e^{-2\lambda\tau(1-z)} \prod_{j=1}^N \left[1 - b_{j1}(1 - e^{-\lambda T(1-z)}) \right] \left[1 - b_{jN}(1 - e^{-\lambda T(1-z)}) \right]. \quad (7.6)$$

$A(z)$, $B(z)$, and $C(z)$ may be interpreted as the z -transform of the number of arrivals during, respectively, subcycle c_{i1} , subcycle c_{iN} , and cycle c_i . Thus, if

$v_{i1}(k)$ is the probability that k packets arrive during c_{i1} , $\sum_{k=0}^{\infty} v_{i1}(k)z^k = A(z)$.

Similarly, if $v_{iN}(k)$ is the probability that k packets arrive during c_{iN} ,

$$\sum_{k=0}^{\infty} v_{iN}(k)z^k = B(z).$$

$Q_{iN}(z)$ can be related to $Q_{i1}(z)$ by conditioning on the number of packets found at t_{i1} . More precisely,

- (1) with probability $p_{i1}(0) + p_{i1}(1)$ no more than one packet is found at t_{i1} . Thus, the queue size at next t_{iN} is equal to the number of packets arriving during c_{i1} and

$$Q_{iN}(z) = A(z).$$

- (2) with probability $1 - p_{i1}(0) - p_{i1}(1)$ more than one packet is found at t_{i1} .

Thus,

$$\begin{aligned} Q_{iN}(z) &= Q_{i1}(z | \text{number of packets} \geq 2)A(z) \\ &= \left[\frac{p_{i1}(2)z + p_{i1}(3)z^2 + \dots}{1 - p_{i1}(0) - p_{i1}(1)} \right] A(z), \end{aligned}$$

where the denominator in the above expression accounts for the fact that we must use queue size conditional probabilities.

Unconditioning we get:

$$Q_{iN}(z) = [p_{i1}(0) + p_{i1}(1)] A(z) + [p_{i1}(2)z + p_{i1}(3)z^2 + \dots] A(z)$$

$$\begin{aligned}
&= \left[p_{i1}(0) + p_{i1}(1) \right] A(z) + \left[\frac{Q_{i1}(z) - p_{i1}(0) - p_{i1}(1)z}{z} \right] A(z) \\
&= \frac{p_{i1}(0)(z-1) + Q_{i1}(z)}{z} A(z).
\end{aligned}$$

The last fraction on the RHS corresponds to the z-transform of the queue size at t_{i1} without counting the packet served. Following the same reasoning, we can obtain:

$$Q_{iN}(z) = \frac{p_{iN}(0)(z-1) + Q_{iN}(z)}{z} B(z).$$

Solving the last two equations for $Q_{i1}(z)$ and $Q_{iN}(z)$ we get:

$$Q_{i1}(z) = \frac{(z-1) B(z) \left[p_{i1}(0)A(z) + p_{iN}(0)z \right]}{z^2 - C(z)}, \quad (7.7)$$

$$Q_{iN}(z) = \frac{(z-1) A(z) \left[p_{iN}(0)B(z) + p_{i1}(0)z \right]}{z^2 - C(z)}. \quad (7.8)$$

By imposing the conditions $Q_{i1}(1) = Q_{iN}(1) = 1$ we obtain, after applying L'Hôpital:

$$p_{i1}(0) + p_{iN}(0) = 2 - \lambda \left(\bar{c}_{i1} + \bar{c}_{iN} \right). \quad (7.9)$$

However, from the definitions of c_{i1} and c_{iN} it follows that:

$$\bar{c}_{i1} + \bar{c}_{iN} = 2\tau + T \left(\sum_{\text{all } j} (b_{j1} + b_{jN}) \right). \quad (7.10)$$

From (7.9) and (7.10), and solving for b_i :

$$b = b_i = \frac{2\lambda\tau}{1 - N\lambda T} \quad (7.11)$$

Equation (7.11) states that the sum of b_{i1} and b_{iN} is a constant, therefore independent of the station index for the equally loaded network. This result holds true even for uneven placement of stations on the net. For evenly spaced stations, the symmetry of the network causes all functions calculated at t_{i1} to be related to their counterparts at t_{iN} by the relation:

$$f_{i,1}(\cdot) = f_{N-i+1,N}(\cdot), \quad 1 \leq i \leq N$$

Therefore, only expressions at t_{i1} are derived.

To determine b_{i1} and b_{iN} explicitly, observe that since $Q(z)$ is analytic within the unit circle any root of the denominator in (7.7) and (7.8) located inside the unit circle should also be a root of the numerator. Observing that $|1 - b_{i1}(1 - e^{-sT})| \leq 1$, we obtain from (7.6) the upper bound $|C(z)| < |e^{-2\tau(\lambda - \lambda z)}|$. Consequently, $C(-1) < e^{-4\tau\lambda} < 1$. As $C(0) > 0$, it is clear that z_0 exists such that $z_0^2 = C(z_0)$ and $-1 < z_0 < 0$.

Equating the numerator of (7.7) to zero and using (7.9) to eliminate $p_{iN}(0)$ produces the following result:

$$p_{i1}(0) = \frac{(2 - b)}{z_0 - A(z_0)} \quad (7.12)$$

Hence,

$$b_{i1} = 1 - \frac{(2-b)z_0}{z_0 - A(z_0)}, \quad b_{iN} = b - b_{i1}. \quad (7.13)$$

Probabilities b_{i1} and b_{iN} can be calculated by computing the b_{i1} and b_{iN} with a fixed z_0 and calculating a new z_0 based on previously iterated b_{i1} and b_{iN} . Fig. 7.1 shows the iteration steps. The dashed boxes correspond to steps where z_0 is numerically evaluated. b in the first box is the basic parameter. The next two boxes calculate initial values for z_0 and b_{i1} . Although we could have started the iterations with $z_0 = -1$ and $b_{i1} = 1$, a faster convergence is obtained by taking upper bounds on $A(z)$ and $C(z)$ as follows:

$$\begin{aligned} A(z_0) &= C_{i1}^*(\lambda - \lambda z_0) \\ &= e^{-2\lambda r_{i1}(1-z_0)} \prod_{j=1}^{i-1} \left[1 - b_{j1}(1 - e^{-\lambda T(1-z_0)}) \right] \left[1 - b_{jN}(1 - e^{-\lambda T(1-z_0)}) \right] \\ &\leq e^{-2\lambda r_{i1}(1-z_0)} \prod_{j=1}^{i-1} \left[1 - \lambda T(1-z_0) b_{j1} \right] \left[1 - \lambda T(1-z_0) b_{jN} \right] \\ &\leq e^{-2\lambda r_{i1}(1-z_0)} \prod_{j=1}^{i-1} \left[1 - \lambda T(1-z_0)(b_{j1} + b_{jN}) \right] \\ &\leq e^{-2\lambda r_{i1}(1-z_0)} \left[1 - \lambda T(1-z_0) \sum_{j=1}^{i-1} (b_{j1} + b_{jN}) \right] \\ &\leq e^{-2\lambda r_{i1}(1-z_0)} \left[1 - \lambda T(1-z_0)(i-1)b \right]. \end{aligned} \quad (7.14)$$

Similarly:

$$C(z_0) \leq e^{-2\lambda r_{i1}(1-z_0)} \left[1 - N\lambda T(1-z_0)b \right]. \quad (7.15)$$

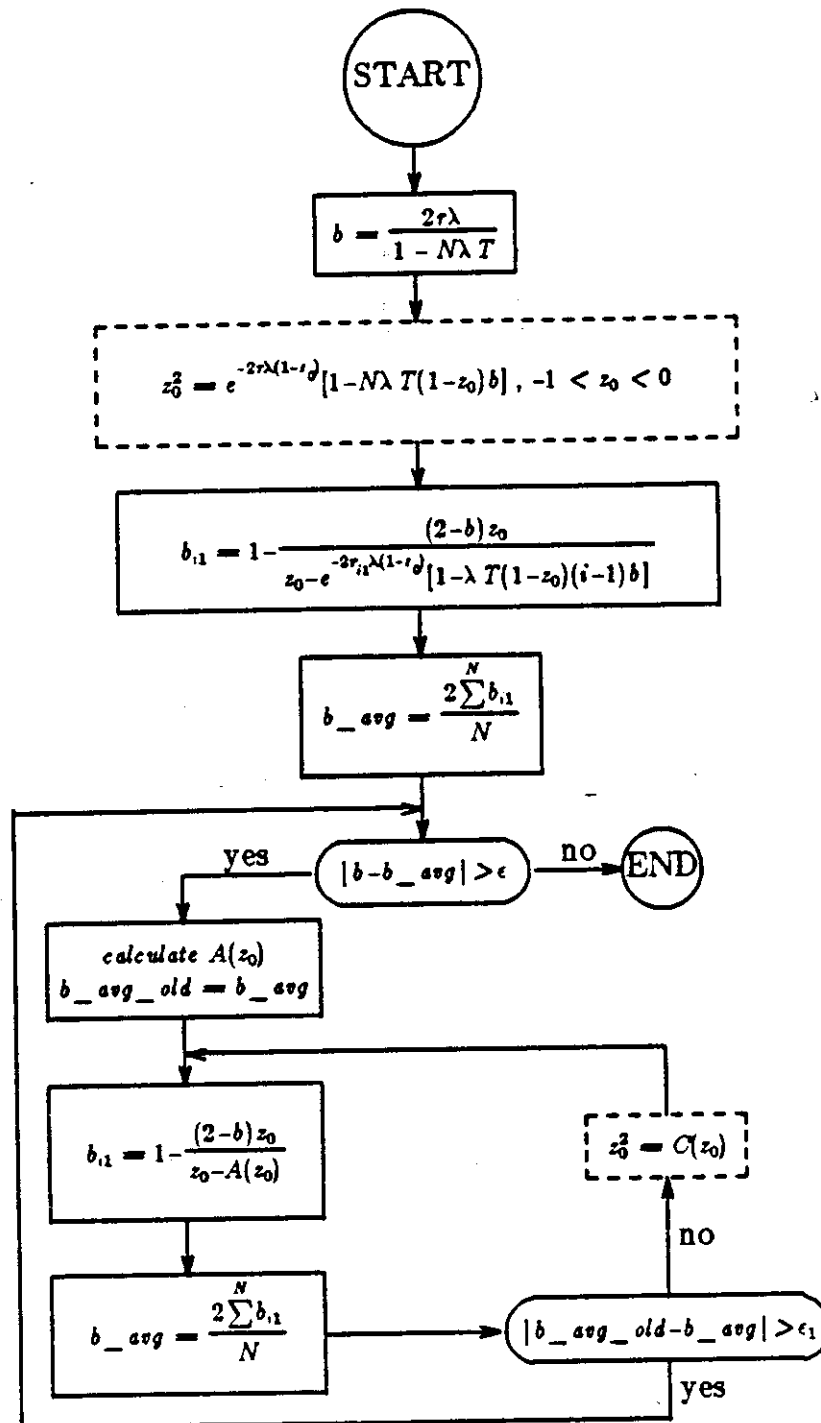


Fig. 7.1 - Iterative Procedure To Calculate b_{i1} .

The initial value for z_0 is obtained from $z_0 = -\sqrt{C(z_0)}$, with $C(z_0)$ approximated by the value of the RHS of (7.15). Substituting in (7.13) $A(z_0)$ for its upper bound in (7.14) gives the initial value for b_{i1} .

Our objective is to calculate b_{i1} and b_{iN} such that their sum approaches b for all stations. The iteration is divided in two parts. First, for a given z_0 we iterate the b_{i1} 's until twice their average converges within ϵ_1 of its limiting value. When no further improvement is obtained, a new value for z_0 is calculated from the last set of b_{i1} 's values. Then, we start iterating the b_{i1} 's again. We proceed until twice the average converges to b within ϵ . We observed that the convergence is very robust. In our calculations we used $\epsilon = 0.000001$ and $\epsilon_1 = 0.00001$. Convergence was achieved within 2 to 3 iterations on z_0 . In many cases only one iteration was needed. At the conclusion, $b_{i1} + b_{iN} \cong b$ for all stations. This result was achieved with different pairs (ϵ, ϵ_1) . As an example, we show in Table 7.1 the values of b_{i1} and b_{iN} for a network with 15 stations, span of 10,000m, packet length of 1000 bits, and load of 100 Mbps.

7.1.2 AVERAGE QUEUEING DELAY W

The average queueing delay W for a random arrival is easily calculated because the distribution of a subcycle is independent of the queue sizes at the prior transmission instant due to the independence assumption. W is calculated similarly whether the tagged arrival occurs during c_{i1} or c_{iN} . Therefore, we assume that the tagged arrival occurs during c_{i1} with delay $W_{c_{i1}}$.

The tagged packet must wait for all packets already in queue when it arrives. Those packets in queue consist of all those present at the beginning of

$N = 15$, packet length = 1000 bits, span = 10,000 m			
load = 100 Mbps, $b = 0.749064$			
$b_{11} =$	0.55019	$b_{1N} =$	0.19888
$b_{21} =$	0.52617	$b_{2N} =$	0.22290
$b_{31} =$	0.50166	$b_{3N} =$	0.24740
$b_{41} =$	0.47675	$b_{4N} =$	0.27232
$b_{51} =$	0.45149	$b_{5N} =$	0.29757
$b_{61} =$	0.42598	$b_{6N} =$	0.32232
$b_{71} =$	0.40030	$b_{7N} =$	0.34876
$b_{81} =$	0.37453	$b_{8N} =$	0.37453
$b_{91} =$	0.34876	$b_{9N} =$	0.40030
$b_{10,1} =$	0.32308	$b_{10,N} =$	0.42598
$b_{11,1} =$	0.29757	$b_{11,N} =$	0.45149
$b_{12,1} =$	0.27232	$b_{12,N} =$	0.47675
$b_{13,1} =$	0.24740	$b_{13,N} =$	0.50166
$b_{14,1} =$	0.22290	$b_{14,N} =$	0.52617
$b_{15,1} =$	0.19888	$b_{15,N} =$	0.55019

Table 7.1 - Example of values for b_{i1} and b_{iN} .

the interval c_{i1} minus the packet eventually served, plus all those which arrived before the tagged packet during c_{i1} . The first group of packets can be calculated from the z-transform of the queue size at t_{i1} . The second group is the number of arrivals during the "age" of the selected interval c_{i1} , where the "age" is the complement of the residual life of the interval. Age and residual life have identical distributions [Ross83]. Calling $P_{i1}(z)$ the z-transform of the number of packets found in queue by the tagged arrival during c_{i1} , follows:

$$P_{i1}(z) = \left[\frac{(1 - b_{i1})(z - 1) + Q_{i1}(z)}{z} \right] \left[\frac{1 - C_{i1}^*(\lambda - \lambda z)}{(\lambda - \lambda z)\bar{c}_{i1}} \right]. \quad (7.16)$$

The first factor is the z-transform of the queue size at t_{i1} after eliminating the packet eventually served. The second factor is the z-transform of the

number of packets arriving during the "age" of interval c_{i1} .

To calculate $W_{c_{i1}}$, observe the following. The first queued packet delays the tagged packet on the average by \bar{c}_{iN} . The second packet in queue, by its turn, delays the tagged packet on the average by \bar{c}_{i1} . Average delays caused by other queued packets alternate between \bar{c}_{iN} and \bar{c}_{i1} . If $p_i = Pr \{ i \text{ packets in queue at the end of selected } c_{i1} \}$, then

$$W_{c_{i1}} = \bar{c}_{iN} (p_1 + p_2 + 2p_3 + 2p_4 + 3p_5 + 3p_6 + \dots) + \bar{c}_{i1} (p_2 + p_3 + 2p_4 + 2p_5 + 3p_6 + 3p_7 + \dots).$$

Evaluating the series in terms of $P_{i1}(z)$, leads:

$$W_{c_{i1}} = \frac{\bar{c}_{iN}}{4} (P'_{i1}(1) - P_{i1}(-1) + 1) + \frac{\bar{c}_{i1}}{4} (2P'_{i1}(1) + P_{i1}(-1) - 1), \quad (7.17)$$

where $P'_{i1}(z)$ is the first derivative of $P_{i1}(z)$. $W_{c_{iN}}$ is computed similarly. Finally, the desired delay W is obtained as:

$$W = \frac{W_{c_{i1}} \bar{c}_{i1} + W_{c_{iN}} \bar{c}_{iN}}{\bar{c}_{i1} + \bar{c}_{iN}}, \quad (7.18)$$

where $\bar{c}_{i1}/(\bar{c}_{i1} + \bar{c}_{iN})$ and $\bar{c}_{iN}/(\bar{c}_{i1} + \bar{c}_{iN})$ are the probabilities that the tagged arrival occurs during c_{i1} and c_{iN} , respectively.

7.1.3 RESULTS

The analytic approximation was compared with simulation for networks with 15 stations, packet sizes of 500, 1000, and 5000 bits, and network lengths of 1000 and 10000 meters. The percentage error between calculated and simulated

queueing delay is plotted against bus utilization normalized to the maximum

AVG ERROR (%) VS NORMALIZED UTILIZATION (N=15)

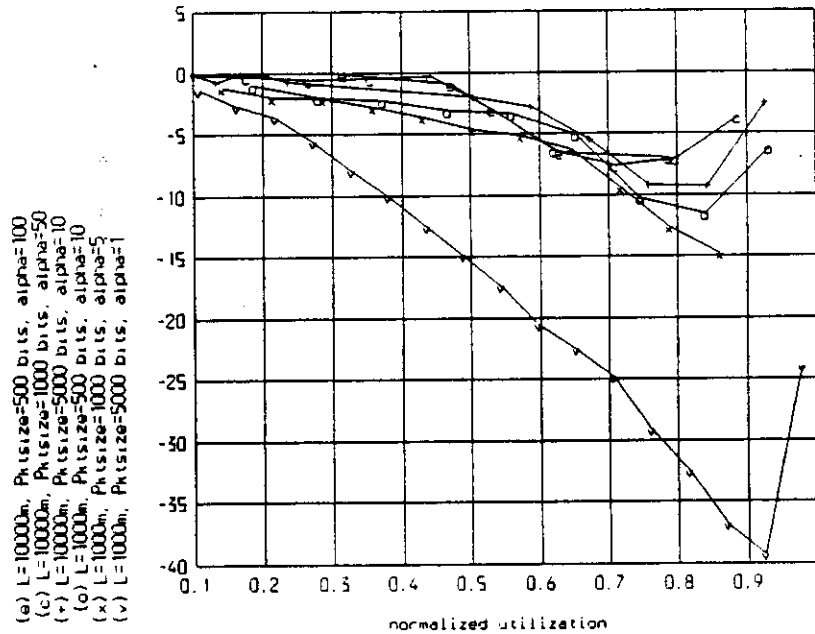


Fig. 7.2 - Average Error (%) vs Normalized Utilization (N=15).

achievable utilization for the given network parameters.

Fig. 7.2 shows the error averaged over all stations. Observe that the different cases correspond to $\alpha = \pi/T$ of 1,5,10,50 and 100. Fig. 7.3 shows the results for station 8 (the central station), while Figs. 7.4 and 7.5 show the results for the end stations (as expected these two figures are very similar). From those figures we observe that the error in the approximation is maximum for the central station. This fact has also been observed for networks with different number of stations. However, all figures emphasize the fact that the approximation improves for increasing α . For $\alpha \geq 5$, the error is less than 10% for normalized utilizations less than 0.7. At heavy load, the errors increase, but this should not cause severe concern, since we are mostly interested in the performance at intermediate load. Increasing the number of stations also favors the

STATION 8 ERROR (%) VS NORMALIZED UTILIZATION (N=15)

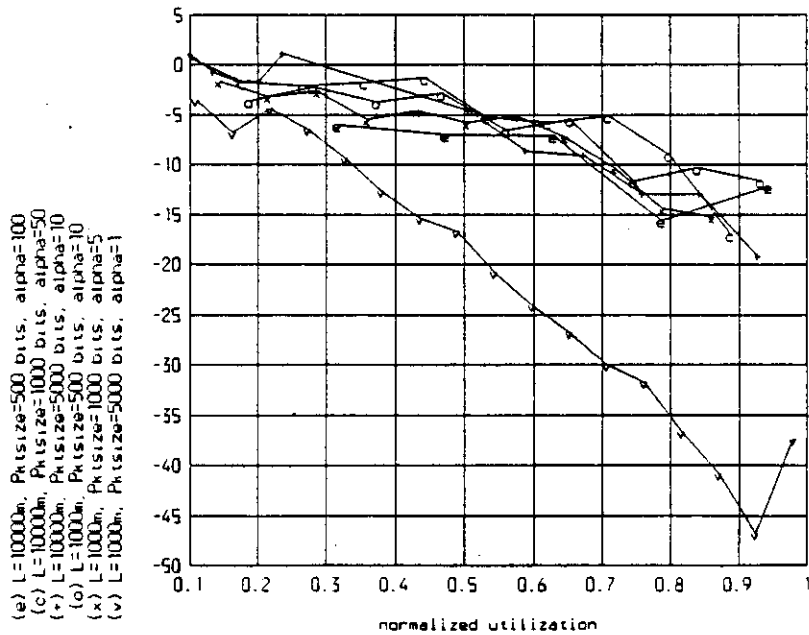


Fig. 7.3 - Station 8 Error (%) vs Normalized Utilization (N=15).

approximation, as shown by additional results for $N = 30$, which are not reported here.

7.2 CONCLUSIONS

A queueing delay approximation for oscillating polling under chaining has been presented. The approximation is based on the assumption that the probability that a station transmits a packet on a given transmission instant can be approximated by a deterministic value. From these probabilities, we obtain the Laplace Transform of the subcycles at each station, and the z-transform of the number of arrivals during each subcycle. The transforms above allow us to derive the queueing delay at each station. A robust iterative procedure is used to calculate the unknown probabilities. Comparing the analytical approximation

STATION 1 ERROR (%) VS NORMALIZED UTILIZATION (N=15)

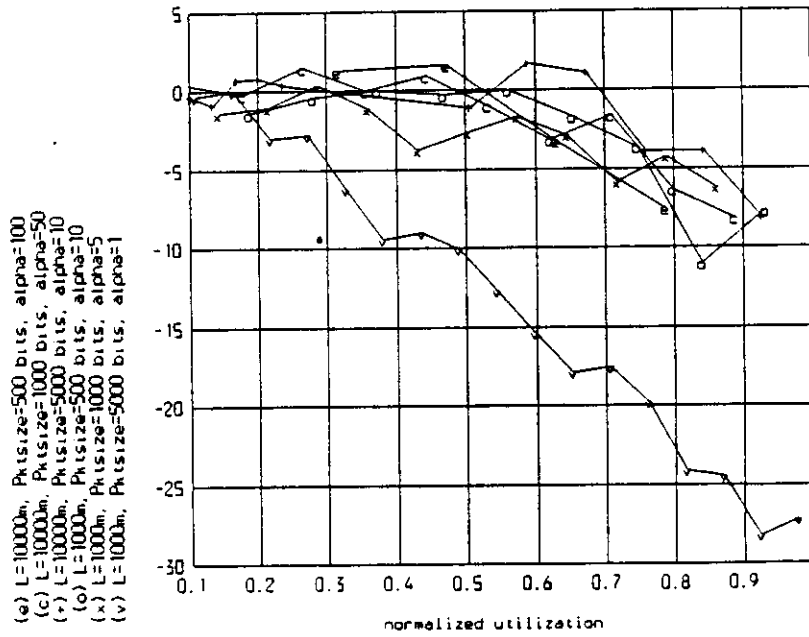


Fig. 7.4 - Station 1 Error (%) vs Normalized Utilization (N=15).

STATION 15 ERROR (%) VS NORMALIZED UTILIZATION (N=15)

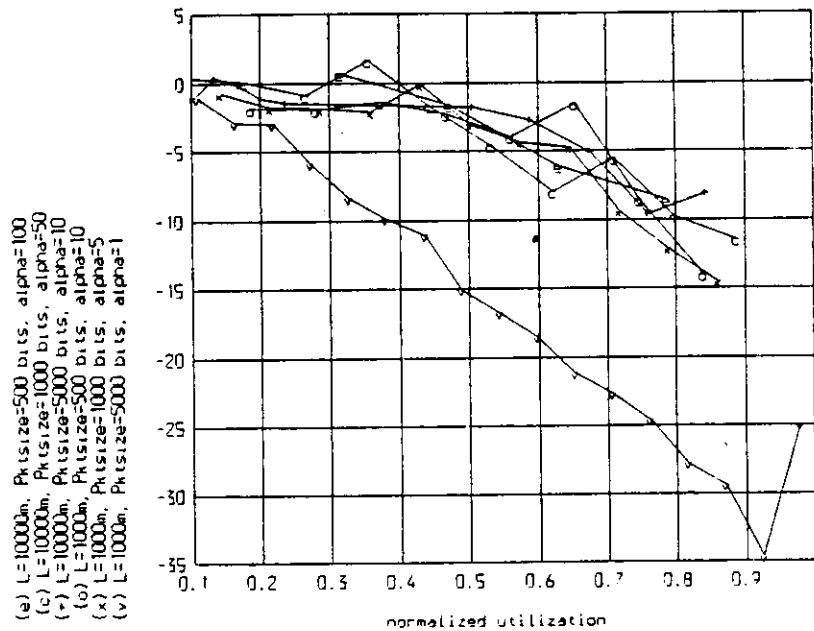


Fig. 7.5 - Station 1 Error (%) vs Normalized Utilization (N=15).

with simulation, the errors show that the approximation reflects well the asymmetric behavior of the stations and is acceptable for medium load situations and for high values of α . More specifically, the error is less than 10% for normalized utilization $\leq 70\%$ and for $\alpha \geq 5$.

A more complex treatment of the problem tried to differentiate between subcycles c_i where a transmission by a station i occurred and subcycles c_j where station i did not transmit. Although the refinement of the approximation follows the same steps, much more complex expressions had to be derived and some extra difficulties had to be overcome during the numerical calculations. The final results, however, did not show any clear improvement over the described simplified approach. For this reason, the treatment is omitted.

CHAPTER 8

BUILDING SYSTEMS WITH A LARGE NUMBER OF STATIONS

8.1 INTRODUCTION

Fiber optics LANs can be implemented using multimode or single-mode fibers. In a multimode fiber the center core is large enough to propagate the light in many different modes, while in a single-mode fiber the core is so small that only one mode propagates. The large core in multimode fibers has clear advantages. LED (light emitting diode), which is an inexpensive and reliable technology, can be used as a light source because enough light can be coupled into the large core (LED's irradiate over a large area). Multimode couplers and connectors have long been fabricated and are commercially available at reasonable prices. However, modal dispersion in multimode transmission limits the available bandwidth/km. In contrast, single-mode fibers do not present modal dispersion and high bandwidth/km is achieved. Consequently, if the network span is large and data rates are high, single-mode fiber must be used.

In a single-mode fiber the small core diameter couples insufficient light from a LED. Therefore, lasers are required. Lasers used to be unstable, expensive and unreliable devices. Recently laser fabrication has been improved tremendously, leading to reliability and life expectancy comparable to LED. Since single-mode technology is essential for very high data rates/large spans and more restrictive in terms of component availability, we only consider single-

mode solutions to systems with large number of stations. Of course, the single-mode analysis is directly applicable to multimode systems, if components of same functionality are used.

In earlier chapters we introduced protocols developed for the dual unidirectional bus architecture in which stations are interfaced directly to busses via couplers, each coupler housing a transmitter and a receiver tap. In practical implementations, it is necessary to interconnect these couplers to the bus via optical connectors or splices (see Figs. 8.1 and 8.2). Single-mode fiber splicing techniques are well-developed and provide interconnection with minimum loss ($< -0.05\text{dB}$) under field conditions. However, splicing requires the intervention of skilled personnel with refined tools and may be a costly solution to station insertion and removal in the field. Furthermore, splicing requires access to the fiber core, and may not be a feasible solution if a sturdy and reliable cable implementation is required. Low-loss lens connectors for single-mode fiber have been developed to provide an average connection loss of -0.54 dB and an average minimum loss of -0.35 dB [Masu82a]. We expect in the near future further improvement in single-mode connector loss as a result of intensive research and high demand.

Single-mode couplers can be constructed using different techniques such as biconical tapering [Kawa77], sapphire ball lenses [Masu82b], or evanescent fields [Beas83]. Evanescent wave couplers fabricated by cementing fibers into plates or embedding fibers in lower melting temperature glass before grinding and polishing have shown excess loss as low as -0.1 dB and allow the coupling ratio to be adjusted at will [Beas83]. To date, these excess loss figures are the best for single-mode couplers. Further progress in this area is expected.

The losses mentioned above contribute to limit the number of couplers that can be connected in a single bus. The maximum number of couplers depends on the available power margin (difference in dB between the maximum power inserted in the bus and the minimum power reliably detected at the receiver) and the individual coupling ratios, as shown in the next sections.

A first step in building systems with a large number of stations is to investigate the optimum selection of coupler parameters to maximize the total number of stations directly connected to the dual unidirectional bus architecture. We are interested in the maximum number of stations on the bus, whether or not the stations are used as multiplexers.

Expansion of the dual topology may occur in a single-level of peer connections through the use of active repeaters (working as signal regenerators), bridges (implementing only routing between two networks, no flow control or buffering), or hybrid topology using a single-mode passive star. Hierarchical interconnections can be achieved by gateways, performing flow control and routing over high level interconnections.

In the following sections we discuss each solution to the problem of building systems with a large number of stations in detail, explaining the limitations of the previous protocols in the new environment.

8.2 DUAL BUS TOPOLOGY OPTIMIZATION

To describe our optimization problem mathematically, we adopt the representation and nomenclature in [Schm83]. Fig. 8.1 shows optical taps and connections for one station, while Fig. 8.2 shows a configuration with N stations.

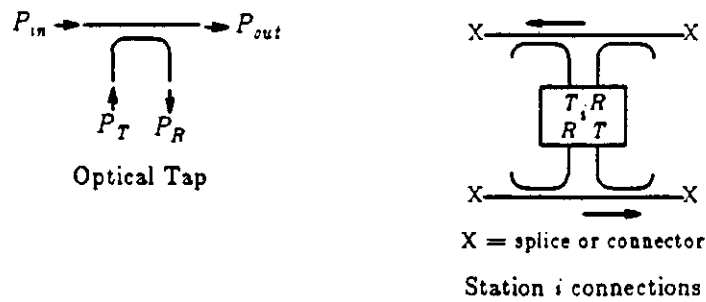


Fig. 8.1 - Optical Tap and Station Connections.

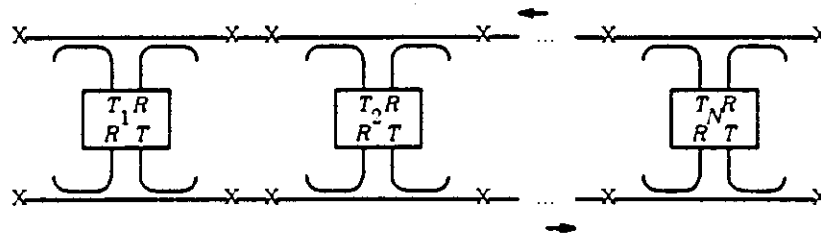


Fig. 8.2 - Configuration with N stations.

Biconical or evanescent field couplers can be represented by the following transmission matrix: [Schm83]

$$\begin{bmatrix} P_R \\ P_{out} \end{bmatrix} = \begin{bmatrix} (1-C)\beta & \beta C \\ \beta C & (1-C)\beta \end{bmatrix} \begin{bmatrix} P_T \\ P_{in} \end{bmatrix}$$

where C = fraction of power coupled between fibers (parameter to be calculated) and β = excess loss through the coupler, with $L_c(\text{dB}) = 10 \log \beta$.

The fraction of power transmitted through a connector or splice is designated as α , with $L_c(\text{dB}) = 10 \log \alpha$. Transmission loss factor due to fiber attenuation is represented by η , with $L_f(\text{dB}) = 10 \log \eta$.

If transmitters have maximum output power P_T and the minimum power reliably detected by receivers is P_S , the ratio P_T/P_S is defined as power margin

M (fiber attenuation is already incorporated) and can be expressed in dB as:

$$M(\text{dB}) = P_T(\text{dBm}) - P_S(\text{dBm}) .$$

If the minimum power received by a transmitter is P_{\min} , the maximum loss in the system is $L_{\max}(\text{dB}) = 10\log(P_{\min}/P_T)$, and the power budget simply requires:

$$M + L_{\max} > 0 .$$

Our optimization task consists in determining coupling ratios which maximize the number of stations attached to the network while still satisfying the above inequality. In the following subsections we analyze four cases where:

- (1) all taps in all couplers have the same coupling ratio.
- (2) taps in the same coupler have equal coupling ratios but couplers may differ.
- (3) a hybrid approach mixing the two previous cases is adopted.
- (4) each tap is independently optimized.

An important issue in practical implementations is the dynamic range (the difference between the maximum and minimum power to be detected in dB) required at each receiver. Received power must be inside the receiver dynamic range to avoid saturation effects that may delay response and cause erroneous operation. Complexity and sophistication of receiver design increases with increasing required dynamic range. An economically feasible local area network would tune receivers for a limited power range. The receiver dynamic range

issue is discussed in each optimization technique.

8.2.1 OPTIMIZATION WITH EQUAL COUPLERS

For this case, all taps in all couplers have the same coupling ratio. In [Schm83] it is shown that the minimum received power occurs between end stations and that P_{\min} is expressed as:

$$P_{\min} = \eta \alpha^{2N-2} C^2 (1-C)^{2N-4} \beta^{2N-2} P_T.$$

It is easily shown that P_{\min} is maximized for $C = 1/(N-1)$ [Schm83]. The correspondent maximum loss (in dB) is given by:

$$L_{\max} = L_f + (2N-2)(L_c + L_e) + 20 \log(N-2) + 20(N-1) \log \left(1 - \frac{1}{N-1} \right).$$

The power margin required for different numbers of stations and different values of $L_e + L_c$, assuming negligible L_f , is shown on Table 8.1.

Typical values for P_T and P_S are -0 dBm and -45 dBm, respectively, giving us a power margin of 45 dB. For this margin, we plot in Table 8.2 the maximum number of stations obtained directly from Table 8.1.

If we look at bus R-to-L, station N receives minimum power from the opposite end station. The power received from the other stations increases as we move closer to N , assuming constant output power from all stations. To diminish the required dynamic range for station N , adjustment of the output power from stations 2 to $N-1$ into the R-to-L bus must be performed, or an optical attenuator must be inserted in series with the transmitter of those stations. Assuming all stations deliver equal power to station N , receivers at stations 2 to

TABLE 8.1					
Power Margin Required for N Equal Couplers					
$L = L_c + L_r, L_f = 0$					
N	L = -0.2 dB	L = -0.4 dB	L = -0.6 dB	L = -0.8 dB	L = -1.0 dB
3	13.04	14.04	15.04	16.04	17.04
4	17.99	19.39	20.79	22.19	23.59
5	21.34	23.14	24.94	26.74	28.54
6	23.93	26.13	28.33	30.53	32.73
7	26.08	28.68	31.28	33.88	36.48
8	27.94	30.94	33.94	36.94	39.94
9	29.58	32.98	36.38	39.78	43.18
10	31.07	34.87	38.67	42.47	46.27
11	32.44	36.64	40.84	45.04	49.24
12	33.71	38.31	42.91	47.51	52.11
13	34.90	39.90	44.90	49.90	54.90
14	36.02	41.42	46.82	52.22	57.62
15	37.09	42.89	48.69	54.49	60.29
16	38.11	44.31	50.51	56.71	62.91
17	39.09	45.69	52.29	58.89	65.49
18	40.03	47.03	54.03	61.03	68.03
19	40.95	48.35	55.75	63.15	70.55
20	41.83	49.63	57.43	65.23	73.03
21	42.69	50.89	59.09	67.29	75.49
22	43.52	52.12	60.72	69.32	77.92
23	44.33	53.33	62.33	71.33	80.33
24	45.13	54.53	63.93	73.33	82.73
25	45.91	55.71	65.51	75.31	85.11

Table 8.1 - Power Margin Required for N Equal Couplers.

N receive a constant power at different levels. If all receivers are to be tuned at the same power level, further attenuators are required in front of each receiver.

8.2.2 OPTIMIZATION WITH SYMMETRIC COUPLERS

In a symmetric coupler the two taps (i.e. transmitter and receiver) have the same parameters C and β . Having two taps with the same parameter may

TABLE 8.2	
Equal Coupler Optimization	
$M = 45 \text{ dB}, L_f = 0$	
$L = L_s + L_c$	N_{\max}
-0.2	23
-0.4	16
-0.6	13
-0.8	10
-1.0	9

Table 8.2 - N_{\max} for Equal Coupler Optimization.

facilitate a one step coupler construction. Most fabrication procedures require monitoring fusion temperature, etching, physical polishing of surfaces, or tension. These processes may eventually be performed in two close taps in parallel leading to accurate dual taps.

The problem of minimizing throughput loss when coupling ration C is allowed to vary along the link length has been solved by Altman and Taylor [Altm77] and by Auracher and Witte [Aura77]. Their analysis was applied to a planar Tee coupler for multi-mode fiber which presented a transmitting matrix slightly simpler than our coupler matrix, which has second degree dependencies.

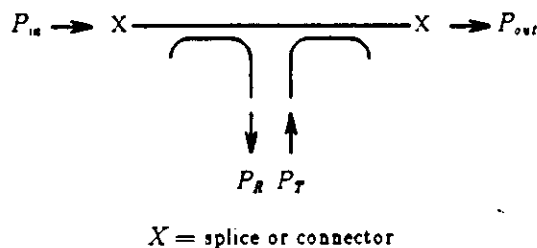


Fig. 8.3 - Two Tap Coupler.

That analysis can be directly extended to our case, and we proceed to do so.

At station i , the transmission matrix for the two tap coupler shown in Fig. 8.3 is given by:

$$\begin{bmatrix} P_R \\ P_{out} \end{bmatrix} = \begin{bmatrix} 0 & \alpha\beta C_i \\ \alpha\beta C_i & \alpha^2\beta^2(1-C_i)^2 \end{bmatrix} \begin{bmatrix} P_T \\ P_{in} \end{bmatrix}$$

Without loss of generality, we focus our attention to the R-to-L bus. The first step in the optimization process consists of a recursive procedure that starts at station N and moves backward toward low numbered stations. Assuming $P_R = P_S$ and $C_N = 1$ at station N , the minimum required power level before station $N-1$ is found by calculating C_{N-1} so that $P_R = P_S$ at station $N-1$. In so doing, only the necessary power is absorbed before reaching station N . This procedure is repeated recursively until station m is reached, when due to other constraints, the iteration can not proceed.

Calling P^i the power level observed on the bus half-way between the connectors or splices of stations i and $i-1$, we write the following:

$$P^N = \frac{P_S}{\alpha\beta}, \quad (8.1)$$

$$P^{i-1} = \frac{P^i}{\alpha^2\beta^2(1-C_{i-1})^2}, \quad (8.2)$$

$$P^{i-1} = \frac{P_S}{C_{i-1}\alpha\beta}. \quad (8.3)$$

Eliminating P_S yields:

$$\frac{(1-C_{i-1})^2}{C_{i-1}} = \frac{1}{\alpha^2\beta^2 C_i}, \text{ for } C_i < 1, C_N = 1. \quad (8.4)$$

Defining $b = \frac{1}{\alpha^2 \beta^2 C_i}$, the solution for C_{i-1} is:

$$C_{i-1} = \frac{2 + b - \sqrt{(2 + b)^2 - 4}}{2} \quad (8.5)$$

We observe that $C_{i-1} < C_i$. Consequently, less and less power is coupled from P_T into P_{out} . We also note that C_i values are independent of P_S or P_T . Limiting station m is reached when $P_T \alpha \beta C_m < P^{m+1}$, and (8.4) cannot be used to determine C_m anymore. P^{m+1} is the minimum power level that guarantees $P_R = P_S$ at station N .

In the second step, we start with $C_1 = 1$ for station 1 and calculate coupling ratios so that any preceding station generates the same output level as the last station in the series. We proceed recursively until we reach station l such that the output level does not satisfy the minimum requirement P^{m+1} from the first step. Developing the expressions for this recursion we find that the new C_i s still obey (8.4). Therefore, the couplers are completely symmetric and $C_i = C_{N-i+1}$. The middle point in the link is located between stations $l = N/2$ and $m = N/2 + 1$.

It is interesting to note that all stations to the right of the middle point receive equal power P_S from all stations situated to the left of that point. To equalize the dynamic range of all stations, it is only necessary to attenuate the signal level received at stations 2 to $N/2$ and adjust output power for stations $N/2 + 1$ to $N-1$. Compared to the former optimization case, only half the ports must be compensated.

Given the ratio P_S/P_T and the coupling ratios C_i calculated according to (8.4), the maximum number of stations in the network is equal to $N = 2n$, where n is the lowest integer to satisfy the following inequality:

$$C_n C_{n+1} < \frac{P_S/P_T}{\alpha^2 \beta^2} \quad (8.6)$$

For a power margin $M = 45$ dB, $L_f = 0$, and different values of $L = L_e + L_c$, the maximum number of stations is calculated and shown in Table 8.3. Comparing these values with those for the equal coupler case, a gain of approximately 2 is achieved.

TABLE 8.3	
Symmetric Coupler Optimization	
$M = 45$ dB, $L_f = 0$	
$L = L_e + L_c$	N_{\max}
-0.2	50
-0.4	32
-0.6	24
-0.8	20
-1.0	18

Table 8.3 - N_{\max} for Symmetric Coupler Optimization.

However, for this case, $N/2$ different coupler ratios are needed. Although these ratios are independent of M , the network may not be easily upgraded because the ratios must correspond directly to the physical position of the station in the network. A solution to this problem is in the next section.

8.2.3 HYBRID OPTIMIZATION

To avoid the problem of a large number of couplers with different ratios, we extend the analysis to investigate a hybrid approach where only an equal number of stations from each end of the network have coupling ratios according to (8.4), and all the other stations have equal coupling ratios.

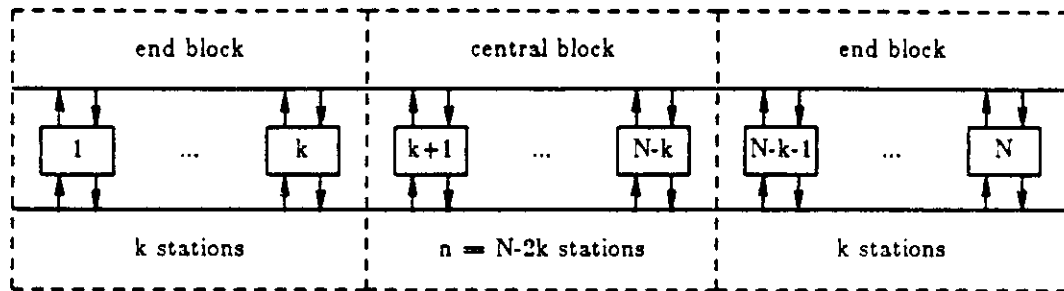


Fig. 8.4 - 3-block network layout.

As with symmetric couplers, we focus on the to R-to-L bus. We assume 3-block network layout shown in Fig. 8.4. The first and the last blocks contain k stations each with coupling ratio C_i optimized according to (8.4). The central block contains n stations with equal coupling ratio C . Of course, $N = 2k + n$. Transmitter power and receiver sensitivity are the same as before, and fiber attenuation is also neglected. For a given k , we want to find a lower bound to C , namely C_{\min} , that allows the maximum number of stations in the central block, and consequently in the entire network.

In the analysis below P^i is defined as in the previous section. P_{in}^b denotes the power ahead of the connector of the first station in the central block and is equal to P^{k+1} . P_{out}^b denotes the power following the connector of the last station in the central block and is equal to P^{n+k+1} . Because of the optimization procedure, the first k stations produce equal P_{in}^b such that:

$$P_{in}^b = C_k \alpha \beta P_T. \quad (8.7)$$

To produce minimum detection levels in the last block, P_{out}^b must satisfy the following inequality:

$$P_{out}^b \geq \frac{P_S}{C_k \alpha \beta}. \quad (8.8)$$

To satisfy (8.8) for a given input power P_{in}^b , we must have:

$$P_{in}^b \left[\alpha^2 \beta^2 (1-C)^2 \right]^n \geq \frac{P_S}{C_k \alpha \beta}. \quad (8.9)$$

Using (8.7) in the above inequality and taking the logarithm of both sides yields:

$$n \leq \frac{\ln \left(\frac{P_S / P_T}{\alpha^2 \beta^2 C_k^2} \right)}{\ln \left(\alpha^2 \beta^2 (1-C)^2 \right)}. \quad (8.10)$$

Another constraint requires that the last station in the block receives minimum power P_S from input P_{in}^b . Thus:

$$P_{in}^b \left[\alpha^2 \beta^2 (1-C)^2 \right]^{n-1} \geq \frac{P_S}{C \alpha \beta}. \quad (8.11)$$

Using (8.7) and (8.11) we get:

$$\frac{(1-C)^2}{C} \leq \frac{1}{\alpha^2 \beta^2 C_k}. \quad (8.12)$$

Because the left-hand side is a decreasing function of C , for $0 < C < 1$, and comparing equation (8.12) with equation (8.4), we conclude that C_{min} is:

$$C_{\min} = C_{k+1} \cdot \quad (8.13)$$

Equation (8.13) is equivalent to requiring that a transmission from the first equation in the central block produces P_{out}^b satisfying equation (8.8). We observe that if equation (8.13) is true, any path from a transmitter to a receiver inside the central block meets the requirements for maximum path loss.

To equalize the dynamic range in this hybrid implementation, attenuators must be added to all transmitters and receivers, except to station 1 and the receivers in the last block. If $k \ll N$, the cost is approximately the same as for equal couplers.

TABLE 8.4										
HYBRID OPTIMIZATION										
$M = 45 \text{ dB}, L = L_c + L_r, L_f = 0$										
k	$L = -0.2 \text{ dB}$		$L = -0.4 \text{ dB}$		$L = -0.6 \text{ dB}$		$L = -0.8 \text{ dB}$		$L = -1.0 \text{ dB}$	
	n	N_{\max}	n	N_{\max}	n	N_{\max}	n	N_{\max}	n	N_{\max}
1	10	12	9	11	9	11	8	10	8	10
2	14	18	13	17	12	16	10	14	9	13
3	18	24	15	21	13	19	11	17	9	15
4	20	28	16	24	13	21	10	18	8	16
5	22	32	16	26	12	22	9	19	7	17
6	23	37	16	28	11	23	8	20	5	17
7	24	40	15	29	10	24	6	20	3	17

Table 8.4 - N_{\max} for Hybrid Optimization.

Table 8.4 shows the maximum number of stations achieved for different values of parameter k and loss L . Comparing the values of N_{\max} with those found in Table 8.2, we note that the hybrid approach is always better than the equal coupler optimization for $k \geq 3$ in the range of chosen parameters. Therefore, optimizing only a few couplers may lead to substantial improvement in the maximum number of stations allowed. For example, for $k = 5$, there is an

increase of about 9 in N_{\max} for all L . The hybrid approach represents an improvement over symmetric optimization in that only a small set of coupling ratios is necessary, and insertion in the central block does not affect the connection of the other stations.

8.2.4 SINGLE TAP OPTIMIZATION

The final step in the optimization of the dual bus architecture is calculating each individual tap to maximize the number of couplers inserted in a series. At station i (see Fig. 8.3), the receiver tap is assumed to have a coupling ratio of C_i^R , while the transmitter tap has a coupling ratio of C_i^T . P^i is defined as before.

Optimization becomes a simple task. Looking at the R-to-L bus, we start from station 1 and maximize the number of succeeding stations which get minimum power P_S at their receiver. Assuming $C_1^T = 1$, the following relations are immediately obtained from the architecture in Fig. 8.2:

$$P_T \alpha \beta = P^2, \quad (8.14)$$

$$\alpha^2 \beta^2 (1 - C_i^R)(1 - C_i^T) P^i = P^{i+1}, \quad (8.15)$$

$$\alpha \beta C_i^R P^i = P_S, \quad (8.16)$$

$$P^i = \alpha \beta C_{i-1}^T. \quad (8.17)$$

Using (8.17) in (8.16), and solving for C_i^R brings:

$$C_i^R = \frac{P_S / P_T}{\alpha^2 \beta^2 C_{i-1}^T}. \quad (8.18)$$

Equations (8.15) and (8.17) allow us to derive the following relation:

$$\frac{(1-C_i^T)(1-CRi)}{C_i^T} = \frac{1}{\alpha^2\beta^2 C_{i-1}^T} \quad (8.19)$$

From (8.19) and (8.18) we derive the following recursive formula for C_i^T :

$$C_i^T = \frac{1}{1 + \frac{1}{\alpha^2\beta^2 C_{i-1}^T - P_S/P_T}}, i > 1. \quad (8.20)$$

Starting with $C_1^T = 1$ we calculate the other C_i^T 's using (8.20). C_i^R is obtained from (8.18). The iteration stops when we find index N such that $C_{N+1}^R > 1$. N is the maximum possible number of stations in the network.

TABLE 8.5	
Single Tap Optimization	
$M = 45$ dB, $L_r = 0$	
$L = L_r + L_c$	N_{\max}
-0.2	61
-0.4	37
-0.6	28
-0.8	22
-1.0	19

Table 8.5 - N_{\max} for Single Tap Optimization.

Table 8.5 shows N_{\max} using the same parameters as in previous optimizations. Compared to previous techniques only a small percentage increase in N_{\max} is achieved, especially when loss L is high.

We observe that the coefficients C_i^R and C_i^T are functions of the power margin. This dependency was not present in previous optimizations. Moreover, the coupling ratio is associated with the position on the network, making

insertions a difficult task, as observed for symmetric couplers. The complexity of such a variety of coupling ratios is only worthwhile because the receivers input power is equalized. In field implementations this optimization approach would not be practical.

8.3 PASSIVE STAR/BUS CONFIGURATION

Single-mode fiber star directional couplers have been successfully fabricated to present excellent uniformity and throughput (excess) loss of less than

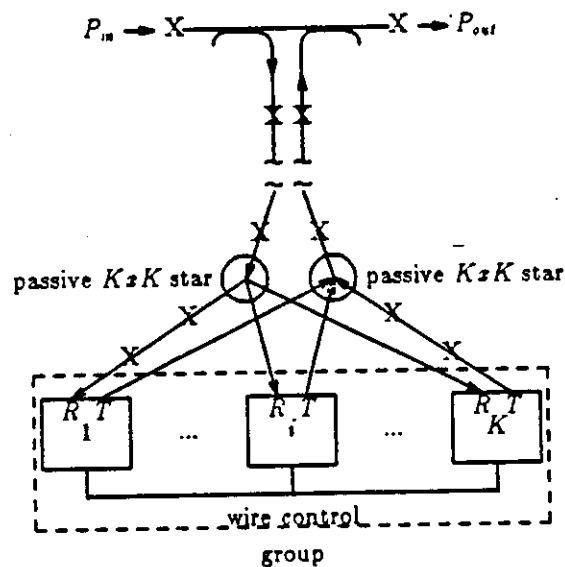


Fig. 8.5 - Passive Star/Bus Configuration.

0.5 dB for a 10x10 mixer [Shee79]. As shown in Fig. 8.5, two passive directional stars can be connected in place of a station in a bus to provide access to a group of physically near stations. A full connection to two busses requires four stars. The star functions as a passive multiplexer. We study the power budget of this star/bus connection and identify the limitations imposed in our previous protocols.

8.3.1 WIRE CONTROL INSIDE A GROUP

Assuming the same propagation delay from the bus coupler to any station in the group, all stations in the group react synchronously to events that have been propagated from the main bus. In our protocols we exploit the physical sequential location of the stations in the network to implement deferral or implicit control. Without an additional control, stations in a group react identically and the possibility of a conflict arises. To arbitrate among the stations, we propose a simple wire control as seen previously in the literature [Mark80, Eswa81, Frat83]. The wire control is logically explained below.

We assume that stations are numbered by decreasing priority inside a group (i.e. station 1 has highest priority). Station j is represented by S_j . S_j receives a control signal $C(j)$ from S_{j-1} and propagates $C(j+1) = C(j) + P(j)$, where $P(j)$ is its own priority signal. We call EOA_i^* the end-of-activity detected at the i th station in the group (generalization of EOC). Similarly, the beginning-of-activity at i th station in the group is BOA_i^* . Both EOA_i^* and BOA_i^* are derived from the signal that is the logical sum of the signal at the receiver tap and the incoming control signal $C(i)$. Therefore, activity is an OR of the activity on main bus and the activity in the group.

Whenever S_i is transmitting a packet, it sets $P(i) = 1$. This action causes all $C(j)$, $j > i$, to be set to 1. Consequently, BOA_j^* occurs for all $j > i$. Low priority stations in the group then abort their eventual transmissions and wait for EOA_i^* , as in a regular deferral procedure. At the end of its transmission S_i sets $P(i)$ back to 0, generating EOA_i^* for stations S_j , $j > i$. The control wire performs the scheduling inside the group and works as a complement to the net-

work access protocol.

To allow transparent operation of the protocols when star groups are allowed, the priority scheduling to access one channel must be exactly the reverse of the access policy for the opposite channel. In a group, if S_i precedes S_j in accessing bus R-to-L, then S_j precedes S_i in accessing channel L-to-R. Moreover, because of the extra overhead in propagating activity between the group and the main channel, the value of parameter d in previous protocols must be adjusted. We call the new value d^* . Originally, d represented the maximum reaction delay of a station. Consequently, d was also the maximum time between consecutive packets in a train, and the maximum amount of preamble garbled when deferral occurred.

If ξ is the maximum propagation delay of the control signal between the first and last stations in a group, and γ is the maximum propagation delay between any station in a group and one of the main busses, the delay between *EOA* at the bus and *BOA* due to the first transmission of the group is in the worst case equal to $2\gamma + d$. The maximum delay between consecutive transmissions from the same group is $\xi + d$, assuming d seconds to start a packet transmission after detection of a transition from 1 to 0 in signal $C(\cdot)$. Therefore,

$$d^* = \max\{2\gamma, \xi\} + d.$$

Fig. 8.6 depicts a corruption caused by the first transmission from a group when the corrupted packet was transmitted by a station directly connected to the main bus (the packet follows previous packet within a station reaction delay). The two time axes represent events on the main bus and at the

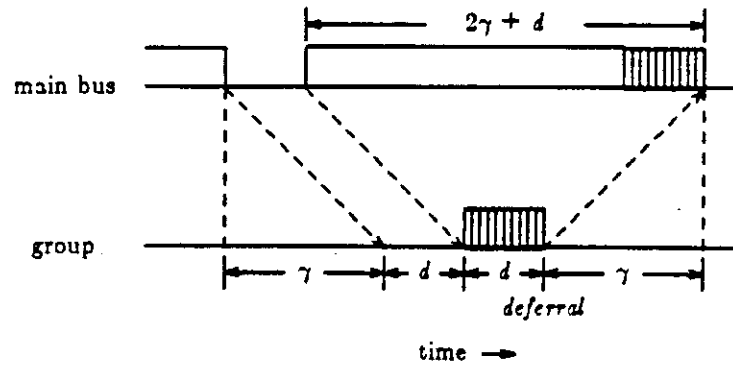


Fig. 8.6 - Corruption by first transmission from a group.

station in a group. The corruption starts 2γ seconds from the beginning of packet transmission and ends d seconds later. Note that the packet first 2γ transmission seconds are not affected.

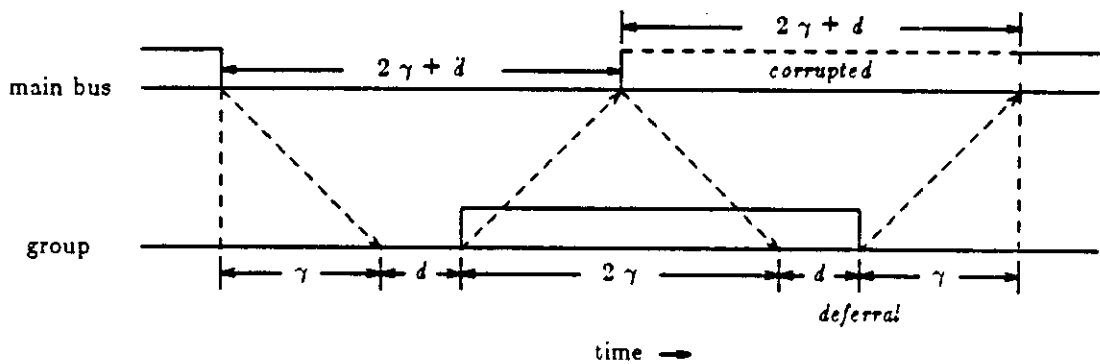


Fig. 8.7 - Group transmission corrupted by another group.

Fig. 8.7 depicts a group transmission corrupted by the first transmission from another group. In this case, the packet first $2\gamma + d$ transmission seconds are completely corrupted. Observe the idle time of $2\gamma + d$ between packets. In the case of consecutive transmissions from the same group the maximum idle

time would be ξ .

All previous protocols work correctly if the the preamble at the beginning of each packet accounts for d^* seconds of garbled data and end-of-train detection only occurs after d^* seconds of idle time. From Figs. 8.6 and 8.7 it is clear that idle periods of size d^* or less occur. The large preamble guarantees that a group corruption triggered by a preceding end-of-transmission occurs during its transmission, thus preventing destruction of any succeeding packets. The increase in preamble may degrade performance considerably and force the use of large packets. However, without the large preamble, packets may be destroyed on the way, and an end-to-end acknowledgement protocol is required for reliability. Even with end-to-end acknowledgment, transmission times would not longer be assuredly bounded anymore.

An improvement in utilization can be achieved if a lower bound for d^* , $d_{minimum}^*$, can be guaranteed and packet transmission times for stations directly attached to the main bus are such that $T < d_{minimum}^*$. These stations can transmit their packets with a regular preamble and, after the packet transmission is over, continue to transmit activity for a time long enough to guarantee a total transmission of d^* seconds. The relationship $T < d_{minimum}^*$ assures the packet integrity against a group transmission and the total transmission time $\geq d^*$ prevents eventual destruction of succeeding packets, as mentioned before.

8.3.2 POWER BUDGET AND UTILIZATION IN A STAR/BUS NETWORK

To achieve the maximum number of stations in a Star/Bus topology, namely N_{\max}^* , we assume that a star group is connected to each pair of correspondent couplers in the dual bus architecture. For this analysis we assume that all stars are equal and that all stations are equidistant from the couplers connected to the main channels. If the stations are not equidistant, then the maximum distance is assumed for all stations to determine the power budget's lower boundary.

The optimization procedures developed in previous sections are still valid if we substitute P_S^* and P_T^* for P_S and P_T , where P_S^* is such as to guarantee that the minimum power delivered to each station in a group is P_S , and P_T^* is the input power to the coupler in the main bus from a transmission originated in a group. Once the maximum number of couplers N_{\max} is calculated, the total number of stations in the network N_{\max}^* is given by $N_{\max} * K$, where K is the number of stations in a group.

Single-mode star unidirectional couplers have been fabricated using the encapsulated etching technique to overcome the geometric problems associated with single-mode fibers. These star couplers have relatively low losses which are characterized by the K port coupling loss, $-10\log K$, and excess loss L_e^* . The successful fabrication of 4-, 6-, 8-, and 10-fiber star couplers with L_e^* as low as 0.5 dB and excellent output uniformity was recently reported [Shee79]. This value for L_e^* is assumed in our numerical examples. A proposal for building a star coupler by the interconnection of smaller optical components appears in [Marh84]. How-

ever, it is highly unlikely that the proposed procedure could lead to star implementations with excess losses as low as reported above. Considering the connections shown in Fig.8.4, the requirements for P_T^* and P_S^* are:

$$P_S^* \geq P_S - L_c^* - 4L_c^* + 10\log K \text{ (dB)},$$

$$P_T^* = P_T + L_c^* + 4L_c^* - 10\log K \text{ (dB)},$$

and the new power margin is $M^* = P_T^* - P_S^*$ (dB). As before, L_f is assumed negligible. L_c^* is the loss of the connector for coupling with the star. We assume $L_c = L_c^*$.

For $M = 45$ (dB), $L_c = -0.1$ dB and $L_c^* = -0.5$ dB, we show N_{\max}^* in Table 8.6 for different values of connector loss and $4 \leq K \leq 32$. Entries where an increase in K leads to a decrease in N_{\max} are suppressed. For instance, for $L = -0.2$ dB and $K = 10$ we obtain $N_{\max} = 120$. This figure is not entered because it is less than 126, which is obtained for $K = 9$. If the connector maximum loss is known, then Table 8.6 gives us optimal values for K to maximize N .

To illustrate the degradation in performance due to an extensive preamble, we assume $N_{\max}^* = 120$. Assuming that the underlying protocol is **TLP-3**, the maximum utilization is given by:

$$S(N_{\max}^*) = \frac{N_{\max}^* T_r}{\tau + N_{\max}^* (T + d^*) + d^*}.$$

The above formula follows from (6.2) where the reaction time d is not neglected. We assume that $2\gamma > \xi$, and, therefore, $d^* = d + 2\gamma$.

TABLE 8.6					
STAR/BUS EXAMPLE					
$P_T = 0$ dBm, $P_S = -45$ dBm, $L_s = .1$ dB, $L_c^* = .05$ dB, $L_c = L_c^*$, $L = L_s + L_c$, $L_f = 0$					
k	N_{max}^*				
	$L = -0.2$ dB	$L = -0.4$ dB	$L = -0.6$ dB	$L = -0.8$ dB	$L = -1.0$ dB
4	96	64	48	40	32
5	100	70	60	-	40
6	108	72	-	48	-
7	112	84	70	56	42
8	-	-	-	64	48
9	128	90	72	-	54
10	-	100	80	-	60
11	132	110	88	66	66
12	144	-	-	72	-
13	-	-	-	78	-
14	-	112	-	84	-
15	150	120	90	90	-
16	160	128	96	-	-
17	-	-	102	-	68
18	-	-	108	-	72
19	-	-	114	-	76
20	-	-	120	-	80
21	168	-	126	-	84
22	176	132	-	-	88
23	184	138	-	92	92
24	-	144	-	96	96
25	-	150	-	100	100
26	-	156	-	104	104
27	-	162	-	108	108
28	-	168	-	112	112
29	-	-	-	116	116
30	-	-	-	120	120
31	186	-	-	124	-
32	192	-	128	128	-

Table 8.6 - N_{max} for a Star/Bus example.

If ν is the speed of light in the fiber, then l , the span of the main busses, and l^* , the maximum distance from any station in a group to the main bus, are related to the previous parameters τ and γ by $l = \tau\nu$ and $l^* = \gamma\nu$. Numerically, we assume $\nu = 2 \times 10^8 \text{ m/s}$. If the minimum preamble required for clock lock-in is T_p , the total required preamble in the star/bus topology is $d^* + T_p$. Consequently, $T = T_r + T_p + d^*$. Numerically, we assume $T_p = 100 \text{ ns}$ (100 bits \times 1Gbps).

TABLE 8.7				
MAX UTILIZATION FOR A STAR/BUS WITH 120 STATIONS				
$2\gamma > \xi$				
l^* (m)	$S(120)$			
	$l = 10 \text{ km}$	$l = 5 \text{ km}$	$l = 1 \text{ km}$	$l = 0.5 \text{ km}$
0	0.642	0.742	0.846	0.861
50	0.390	0.425	0.457	0.462
100	0.280	0.298	0.313	0.316
200	0.179	0.186	0.192	0.193
300	0.132	0.136	0.139	0.139
400	0.104	0.107	0.109	0.109
500	0.086	0.088	0.089	0.089
600	0.073	0.075	0.076	0.076
700	0.064	0.065	0.066	0.066
800	0.057	0.057	0.058	0.058
900	0.051	0.051	0.052	0.052
1000	0.046	0.047	0.047	0.047

Table 8.7 - Max Utilization for a Star/Bus with 120 stations.

Because $\tau \ll N_{\max}^*(T + d^*)$, we expect $S(120)$ to be somewhat independent of l . Table 8.7 depicts $S(120)$ for $l = 10000, 5000, 1000$ and 500 meters, and confirms our conjecture. Figures for $l^* = 0$ correspond to the ideal case, where all stations are connected directly to the main busses. We observe that substantial throughput is lost as l^* increases. However, if l^* is small, a large

number of stations can be supported while maintaining reasonably good utilization.

8.4 LINEAR EXPANSION THROUGH ACTIVE REPEATERS

The span of a network can be increased by simply adding repeaters (or regenerators) to a channel. The repeaters reconstitute the signal to the initial power level and shape, allowing another bus segment to follow the previous segment. This approach is completely transparent to the underlying access protocol. All segments are considered peers and no hierarchy is introduced. The whole network performs as a single entity, but performance may degrade due to increases in end-to-end propagation delay. High speed integrated regenerators for long haul optical systems have been fabricated for speeds up to 320 Mhz [Coch83]. These regenerators perform reshaping, retiming and regenerative functions. In a LAN shorter span length alleviates the regenerator performance requirements, and we expect further developments reaching the gigabit range in a near future.

The introduction of an active device may compromise reliability because a repeater failure may disrupt the whole network operation. To improve reliability repeaters can be constructed with internal redundancy, or by-pass optical switches that are activated in case of repeater failure [Alba82, Alfe81]. A hybrid approach using redundancy and by-pass may also prove sound.

8.5 LINEAR EXPANSION THROUGH BRIDGES

The total number of connected nodes in a dual bus topology can be increased by interconnecting separate segments of the network through bridges. Bridges are active devices which provide simple real time routing based on destination and/or source addresses. No flow control is implemented and no data buffers are provided. A small delay buffer may be required to allow processing of the routing decision. A segment is said to be local to the bridge to which it is connected and vice versa. If two segments of a LAN are interconnected by a bridge, local traffic (i.e. with destination within originating segment) is recognized at the bridge and ignored. External traffic (i.e. destination is external to the originating segment) is inserted in the neighboring segment at the highest preemptive priority to avoid blocking and buffering. In brief, the bridge works as a filter, with non-local traffic passing through. For dual unidirectional bus topology, high preemptive priority is inherent to end stations, which naturally perform as bridges. Another consequence of the absence of buffer for external traffic is that, if traffic loss is not tolerated, bridges can only interface with two segments. If three or more segments are interconnected, a conflict may result when two or more segments originate packets for a third segment simultaneously.

Bridge connection mandates that segments be at the same level, so stations are considered peers, the condition usual for LANs. Naming can be done by the concatenation of a unique segment identifier to a port identifier. A port identifier is an address recognized by the lowest level hardware interface connected to a coupler. A port could be a physical entity representing a station or a node, or a logical entity representing a set of stations or processes. A special

segment identifier could be also used for overall broadcast capability. The bridge stores the identifiers of its local segments enabling the recognition of local and external packets.

Bridges can further filter external packets originated locally by retaining a set of the destination segments to which they are allowed to retransmit. Each set is called the reachable segment set for the incoming bus. In a local segment where transmissions are bidirectional, if a bridge maintains a reachable segment set A for one bus, and the opposite bridge maintains a reachable set B for the other bus such that A and B are a partition of the network (the two sets are mutually exclusive and their union covers the whole network), a local broadcast packet is retransmitted by only one bridge and, therefore, propagates externally in only one neighboring segment. If a segment has local bridges which satisfy the above conditions, the segment is denominated a *uni-segment*. If the local bridges retransmit all external packets, the segments are called *regular* segments. Regular and uni-segments imply that local transmissions are bidirectional. If transmissions are unidirectional, bridges always retransmit all external packets and segments have no special denomination.

Fig. 8.8 (a) is a schematic representation of a linear bridge expansion. This solution is adequate when a high percentage of external traffic goes only to neighboring segments. Operation is very simple for regular segments. An external packet is propagated from one segment to the next until it reaches the destination segment. In this implementation, bridges have only to compare the segment identifier of the destination address with the identifier of the originating segment.

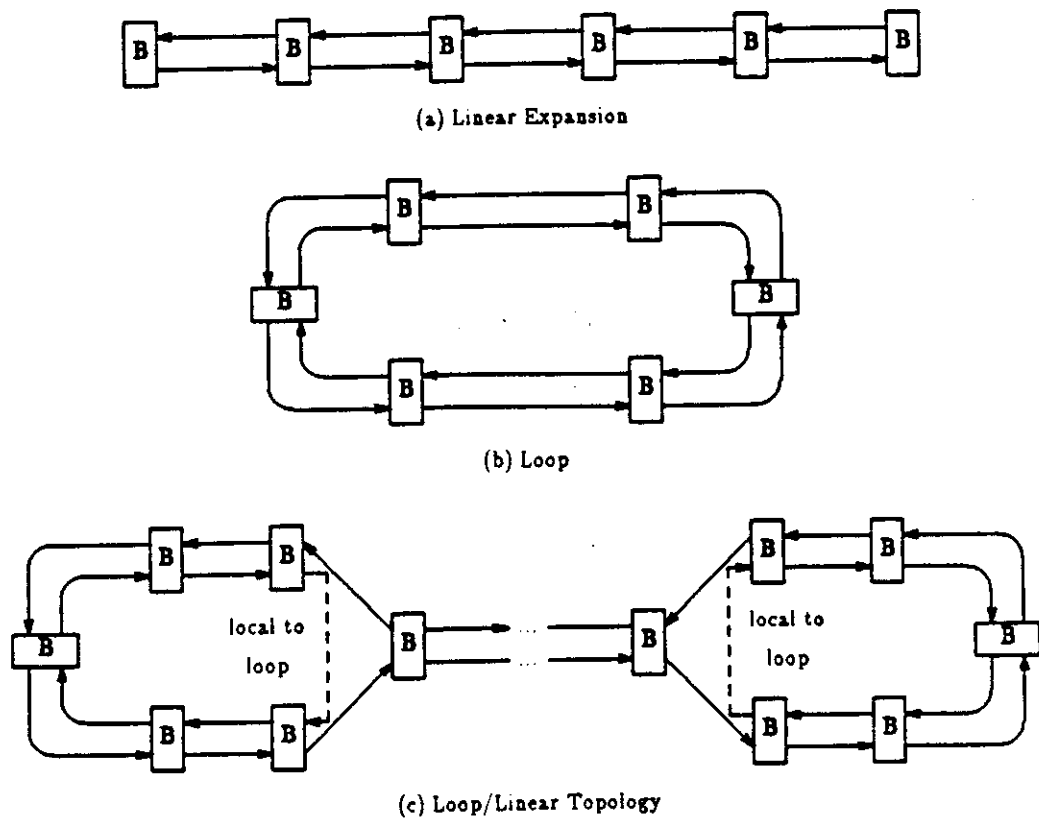


Fig. 8.8 - Bridge Connections.

Fig. 8.8 (b) shows a loop implementation of interconnected segments. This solution is convenient for equally balanced traffic between segments. If bi-segments are used, in absence of errors two copies of the same packet reach the destination segment (assuming the local transmission is bidirectional). The next level in the protocol hierarchy must be capable of filtering out multiple copies, or the destination port must store the sequence number of the received packets.

The problem of multicopies in long haul networks is very complex because packet delay is large and unbounded. In high speed LANs packet delay is small and bounded, and retransmission can occur from the source node (no intermediate buffering is required). Therefore, sequence number of received packets need only be stored for a fixed time (a function of the total number of nodes in the

network). Although it is possible to implement multiple copy control in hardware, for very fast processing it is advisable to perform this function at a higher protocol level. Solution choice is an implementation issue. Despite the cost of eliminating multiples, the presence of a copy travelling the loop in opposite direction enhances reliability and may improve delay. The extra bandwidth wasted by the copy is usually not a significant problem for high bandwidth LANs. If extra bandwidth is needed, uni-segments solve the problem.

The final example in Fig. 8.8 (c) shows a segment interconnecting two loops. In that case the clockwise direction in the loops carry only local loop traffic. Therefore, in bidirectional transmissions the reachable segment set for the counter clockwise direction in the bridges at each loop must to be such that it covers all segments external to the local loop. If regular segments are used, the endless circulation of external packets in the inner loop can be eliminated by removing the dashed connection *local loop* in both local loops. This elimination may cause extra delay for local loop packets because a longer path may be followed. If uni-segments are used, only local loop packets circulate in the inner loop and the above connection can be maintained. Depending on definition of partitions, local loop traffic performance may be improved.

8.5.1 COMPATIBILITY BETWEEN BRIDGES AND ACCESS PROTOCOLS

The requirement that bridge traffic be forced into the local segment in real time implies that, for asynchronous protocols, the local access protocol can assimilate the eventual collisions caused by the high priority insertion. Protocols such as Token-Less and Buzz-Net, which perform normally when collisions are

not frequent, are not suitable for bridge implementations. Recovery overhead due to constant collisions deadlocks the local traffic. TDT-Net does not tolerate out-of-sync traffic, making bridges unacceptable. The above protocols cannot be modified to recover from the problems caused by bridge traffic and are discarded from further consideration.

Fasnet and U-Net, on the other hand, appear to be suitable for bridge implementations. Fasnet provides easy bridge implementation because the end stations are responsible for generating time slots in the system. Since the system is synchronous, incoming external traffic may be forced to wait (in the worst case) for a slot time before insertion. Therefore, a buffer the size of a slot must be provided. No further modifications are required. Fasnet bridge connections are discussed in [Limb82].

As for U-Net, the external traffic inserted in a bus of a local segment can:

- (a) be external packets that were part of a train of consecutive packets following a token. External traffic interpacket gaps correspond to local packet transmissions in the original train that were filtered out by the bridge. These gaps are large enough to allow local traffic insertion.
- (b) be external packets transmitted when stations were synchronized by the token in the opposite bus. As explained in Section 2.2.1, the packets in the reverse bus are separated by gaps equal to twice the propagation delay between two sending stations, plus $2d$. After local traffic is filtered out, external traffic interpacket gaps are of the order of $2\tau_{ij}$ or $2\tau_{ij} + nT$. Also in this case local traffic can be inserted in these gaps.

A first problem with U-net is that, independent of the type of external traffic (a or b), collisions with external traffic occur, and collided stations abort their transmissions. Second, using bridges causes problems with single token and bidirectional transmissions, because the token is travelling in one direction, but external traffic is being inserted in both directions (assuming two local bridges). If external traffic gaps do not occur at the taps of a station in both busses simultaneously, a lock-up condition develops where stations always transmit in both directions simultaneously.

A solution to these problems is to modify the standard U-Net protocol to make transmissions either unidirectional or scheduled in independent output queues, one for each channel. The queues are managed separately, with packet transmissions occurring only when synchronized by tokens detected in the correspondent channels. Following this approach, the number of tokens can be selected as follows:

- (1) One token in the entire segment. End stations regenerate the token as usual. The token starts a train to which stations append their packets. If a collision with external traffic occurs, the single packet transmission is aborted and tried again when the bus is sensed idle. Local traffic fills external traffic gaps. Bandwidth is wasted because only the token bus is used for local transmissions. While the token is travelling along one bus, the reverse bus is used exclusively by external traffic. If the external traffic is light, the reverse is idle most of the time.
- (2) One token for each channel. Bandwidth is not wasted, but token regeneration is a problem.

Token regeneration can be implemented by out-of-band signalling in the reverse channel. Signalling can be imbedded in packets, use a special packet, or be an identifiable sequence of carrier bursts. If stations other than the end stations can detect the out-of-band signalling, some bandwidth can be further utilized by allowing stations to transmit following the detection of *EOA* events, after the regeneration signal has been detected in the reverse channel. This scheme is fair; although downstream stations detect the signalling first, upstream stations have preemptive priority over downstream. Of course, improvement is only achieved if $T \ll \tau$. As T approaches τ , collisions corrupt the extra transmissions.

If the maximum packet transmission time (T_{\max}) is less than τ , tokens can be automatically regenerated every T_{\max} . To maintain fairness (prevent upstream stations from monopolizing the network), out-of-band signalling in the reverse channel is still necessary. After transmission a station only transmits again after an *EOA* if out-of-band signalling has previously been detected in the other channel. The out-of-band signal is sent on the reverse channel each time an end-of-train is detected. The train always starts with a token (whatever implementation) and ends when a silent gap of more than $2d$ seconds is detected. Although automatic token regeneration may corrupt some packets, it may improve performance when $T \ll \tau$ and the external traffic inserted in the segment is light, failing to provide sufficient *EOA* events to trigger transmissions. This scheme is especially useful when the end-to-end propagation delay is high, external traffic is low, and token regeneration is slow. When external traffic is heavy, token generation is not necessary to trigger transmissions, but it provides the means to frame the channel and bring fairness through out-of-band signalling at the end-of-train..

External traffic gaps in this modified version are $nT + (n-1)d$ in length. If packets are fixed in size, collisions are minimized and performance is improved. This design resembles a synchronous system with slots of size $T + d$. The advantage of using bridges instead of repeaters for the synchronous system was explored in [Limb82]. If, in our asynchronous case, we ignore the packets transmitted after out-of-band signalling is detected and assume that (in the worst case) each external packet always collides with a local transmission and wastes on the average $T/2$ seconds of transmission, then we can easily show that bridge expansion always performs better than repeater expansion when there are more than two segments.

8.6 HIERARCHICAL CONNECTIONS USING GATEWAYS

A more general way to provide access to numerous stations in a LAN is to use gateways to interconnect independent segments. Because the segments are part of the same network (in our view), the protocol layers in each segment are the same, with the possible exception of the access protocols which are not equal. Access protocols, optimized for the traffic characteristics of the particular segment, are required to provide the same basic service to the immediate higher protocol layer, therefore eliminating the need for protocol conversion at the gateways. This property is very valuable because it avoids excessive processing at the gateways which could degrade delay performance.

Gateways interact directly with the access protocol providing routing, buffering and flow control to intersegment traffic. Flow control must be provided because the gateway has limited bandwidth and may run out of buffers if through traffic is heavy. If the gateway cannot accept an incoming packet for

retransmission, the packet must be dropped and a NACK sent back to the sender, perhaps with information about buffer availability. If the packet is accepted, an ACK is returned. In the latter case, if the sender is another gateway, a buffer in that gateway is freed. If the sender is a station, a transmission buffer is freed at the communication interface. To prevent deadlocks and simplify link control over the high speed broadcast bus, whenever a packet is accepted in the gateway a buffer reservation for accepting future ACK or NACK is made at the corresponding inbound link. Due to the broadcast nature of the segments, ACKs and NACKs must be implemented as control packets.

To guarantee bounded delay and required throughput, traffic through the gateways must be sent through virtual circuits (VCs). Each gateway can connect with a fixed number of VCs depending on local buffer availability and local segment utilization. VC characteristics are negotiated during the set-up phase. If VC requirements cannot be satisfied for one of the gateways in the desired path, the connection is not made. Some traffic may require bounded delays (real time, voice, video, etc.), others may require high throughput (graphic terminal refreshing, file transfer, etc.) A VC may span many segments. A gateway must handle local (segment destination is local) and through (segment destination is external) traffic with different priorities if necessary.

Because of short delay, the number of outstanding packets in a VC path may be very small and still satisfy throughput requirements. If segments are not heavily loaded, one outstanding packet will suffice, simplifying protocol handling and speeding up gateway processing. Protocols between gateways (G-G) or between gateway and stations (G-S) may be implemented as a stop-and-wait protocol ([Tane81], pp. 151-153), or as a sliding window with NACK ([Tane81]

, pp. 153-164) if high throughput is required. The stop-and-wait protocol can be viewed as a sliding window protocol with window size 1. This simple protocol is advisable for easy implementation directly in hardware or firmware. At very high speeds, software interaction must be minimized to avoid causing a bottleneck in the system [Magl82]. To speed up processing, associate memories may be used in the gateway interface implementation [Blau79].

VCs can be established as usual. Datagrams can be imbedded in VC call requests as provided in CCITT X.25 ([Tane81], pp. 244) or supported directly. A minimum throughput for datagram service is guaranteed if datagrams are sent through a permanent VC between source and destination segments. Because segments are broadcast, VC connections are primarily used for intersegment user sessions. Direct broadcasting provides an easy solution to VC establishment. Different stations with sessions to the same destination segment can be multiplexed on the VC. The choice of datagram approach is an implementation issue.

Paths are not necessarily unique between segments, especially if the segment is connected to more than one gateway. However, it may be possible for a network to provide a unique path between any pair of segments. If so, the gateway must maintain a routing table that provides for each segment destination the next segment to broadcast and/or the next gateway to which to send. This table must be maintained to avoid multiple paths or loops in case of gateway or network failure. One advantage of unique routing is that datagrams may arrive in order (this property can only be guaranteed if datagrams are sent in a VC). The routing table is consulted to establish VC or to route datagrams. For the VC service, a routing vector can be maintained. The routing vector gives the

destination gateway (or segment) depending on VC identification. For fast processing, the routing vector can be kept in an associative memory, for quick routing without software intervention.

If bandwidth is plentiful, one option is to have the first gateway in the path provide the full route for the packet. The routing processing takes place only at this first gateway, which is responsible for maintaining the VC connection and stamping the address in the packet header. Each intermediate gateway simply extracts its own address from the packet address field and uses the subsequent address specification to determine to which segment to broadcast the packet. Although the packet address field of the packet must be variable, the extraction procedure at each gateway can be fixed and executed by hardware.

The construction of the routing table at the gateways can be implemented using the algorithm described in [Mor183], with minor additions. This algorithm is an extension of the version presented in [Mer179]. The protocol maintains packet sequence and recovers from single segment or node failures without loss of packets, and from multiple failures occurring simultaneously with the possible loss of some packets. The protocol does not require a priori topological information and handles network initialization and reconfiguration automatically. In the brief description below we detail the necessary additions.

In the terminology of [Mor183], nodes are gateways and links are segments. An end station that is not a gateway must participate as a leaf node (as is a gateway that is connected to only one segment). For each node the protocol constructs a multi-branched tree (a spanning tree) rooted at the given node (the SINK). Each node selects a preferred neighbor to which it points.

Two types of messages are used in the protocol. The *distance update* (UPD) message carries an estimate of the distance (delay) to the SINK from the sending node. The estimate may be for the direction from the node to the SINK. The *flush control* (FLS) message is used to ensure that the old pathway is clear before making a route change.

The update cycle consists of four phases, with only the last one requiring some additions. In phase I, UPD messages move up-tree (away from SINK) to enable nodes to know their distance to the SINK. In phase II and III FLS messages verify connectivity and prepare nodes to clear the old path. In phase IV, a node propagates UPD along the down-tree (towards the SINK) after UPDs have been received from all up-tree links, and after the node has determined a preferred neighbor to the SINK (the node may maintain its previous preferred neighbor). We make the following addition. The node adds, to the UPD sent down the tree, its identification, its delay to the sink, and its set of local segments. The UPD is only sent after UPDs from all neighbors except the preferred neighbor have been received. Upon receiving the UPD, the SINK knows the minimum delay path to all segments, including multiple routes. If single path routing is implemented, in phase IV a node erases previous entries corresponding to its local segments if one or more local segments are already present in the received UPDs. This occurs because the present node receives the packets addressed to those segments first than the other nodes up-tree. We observe that intermediate gateways have acquired the routing to any local segment of the SINK. If that segment is also connected to another gateway, than the minimum delay route can be selected or multiple routing implemented. In summary, the above protocol allows implementation of individual routing at each gateway, or full routing at the first gateway of the path.

CHAPTER 9

CONCLUSIONS

9.1 SUMMARY OF RESULTS

The major contribution of this dissertation is a comprehensive study of asynchronous protocols for the high speed dual optical bus topology. In our opinion, optical fiber is the most promising medium for the implementation of high speed LANs. If bus is used, the dual bus topology offers the best solution to the problem of high insertion loss presented by optical couplers.

All proposed protocols are distributed and able to handle variable size packets. Initialization and recovery procedures are incorporated in the protocol definition (no external intervention, e.g. NCC). The above features are important to assure reliable and efficient operation of the high speed medium.

U-Net (Chapter 2) is a token protocol which circulates the token (a special pattern or packet) between end stations, and incorporates a distributed end station election procedure to improve reliability. TDT-Net (Chapter 2) utilizes the infrastructure of U-Net but provides corruption free transmissions by using synchronizing mini-slots to perform station scheduling. Both protocols perform optimally for equally loaded and symmetric network, and their operation can be modelled as an oscillating polling scheme.

Buzz-Net (Chapter 3) uses a hybrid random/token scheme to provide high bandwidth to a single sending station and optimal performance at light load. A special buzz pattern is used to force stations from random mode to control mode. However, cycle reinitialization overhead has significant impact on performance when multiple stations collide during the random phase.

Rato (Chapter 4) is a very simple pure random scheme which uses a time-out delay to bring fairness to buss access. An interesting feature of Rato is its complete insensitivity to end-to-end propagation delay. Performance, however, is dependent on the number of active stations and maximum packet transmission time. Rato reflects a compromise between simplicity of implementation and performance.

The most original contribution to high speed LANs is the Token-Less family (Chapter 5). The simple control of the channels by sensing activity only, provides the means for a reliable and easy hardware implementation. One version, TLP-3, perform as U-Net without relying on the detection of special patterns. The adaptive version TLP-4 outperforms any other protocol under unevenly loaded and multipacket traffic, and perform optimally at light load. Simulation experiments with single heavy loaded station have shown that background stations are not affected by the heavy load traffic in the network for all Token-Less versions. This fact makes TLP-4 an excellent choice for applications where bursty high bandwidth has to coexist with interactive and priority traffic (real time, etc.).

The approximate solution to the queueing delay for oscillating polling under chaining (Chapter 7) is useful contribution. The approximation is based on the assumption that station transmissions in a round are independent events

with a fixed probability dependent on total network load. Simulation results show that the approximation reflects well the asymmetric behavior of the stations and is acceptable for medium load situations and for high values of α .

In Chapter 8, the complete treatment of the biconical coupler optimization problem for the dual bus topology is novel. Our results show that a substantial increase in the number of couplers can be obtained by optimizing a few couplers closer to the ends and using a constant coupling ratio for couplers in the mid-section of the network. The star/bus solution to the problem of building systems with a large number of stations had not been analyzed before. Furthermore, our results show that a large number of stations can be passively interconnected using off-the-shelf optical elements. The gateway considerations emphasizes simplifications to the interconnection problem due to the high speed environment.

9.2 EXTENSIONS OF THIS RESEARCH

Our research concentrated in protocols with bounded delays. Simulation results for a version of CSMA/CD in Chapter 8 show that adapting the random scheme to the dual unidirectional topology can produce acceptable utilization even at very high speed. Further research is necessary to identify the critical parameters and guarantee no capture effects among the stations.

Although insertion and deletion is automatically handled by the proposed protocols, the side-effects of cable rupture has not been investigated. Whether new mechanisms can be incorporated to the protocols or new protocols have to be devised is an open question. New protocols should be investigate when multiple unidirectional busses are used in each direction, with a major emphasizes in

reliability.

Development of more accurate analytical models for oscillating polling is an open area for research. Extensions to handle unbalanced and chained multipacket message traffic would be very useful.

The existence of a very high speed interconnecting medium may enable applications running in a distributed environment to relinquish constraints applicable to low speed environments and simplify protocols and algorithms dealing with data transfer and consistency check. Currently, LANs use relatively small packet sizes. The availability of higher bandwidth allow the development of applications using large message transfers. High speed interfaces have to be devised to avoid the bottleneck caused by protocol processing and data transfer to the host. We believe that new ideas in the high-level-protocol/OS/architecture fields will match the above suggestions.

References

- [Abra73] Abramson, N., "Packet Switching with Satellites," in *Proceedings AFIPS Conference*, Montvale, NJ: 1973.
- [Alba82] Albanese, A., "Fail-Safe Nodes for Lightguide Digital Networks," *Bell System Technical Journal*, Vol. 61, No. 2, February 1982, pp. 247-256.
- [Alfe81] Alferness, R.C., N.P. Economou, and L.L. Buhl, "Fast compact optical directional coupler switch/modulator," *App. Phys. Lett.*, Vol. 38, 1981, pp. 214-216.
- [Altm77] Altman, D.E. and H.F. Taylor, *An Eight-Terminal Fiber Optics Data Bus Using Tee Couplers*: Crane, Russack & Company, Inc., 1977.
- [Aura77] Auracher, F. and H.H. Witte, "Optimized layout for a data bus system based on a new planar access coupler," *Applied Optics*, Vol. 16, No. 12, December 1977, pp. 3140-3142.
- [Beas83] Beasley, J.D., D.R. Moore, and D.W. Stowe, "Evanescent wave fiber optic couplers: three methods," in *Proceedings 1983 SPIE Symposium on Fiber Optics Multiplexing and Modulation - Vol 417*, Arlington, VA: April 1983, pp. 36-43.
- [Blau79] Blauman, S., *Labeled slot multiplexing: a technique for a high speed fiber optic based loop network*, Torrance, CA: TRW Communications Group, 1979.
- [Coch83] Cochrane, P., D.W. Faulkner, L. Bickers, I. Hawker, and R.J. Hawkins, "Minimum Chip Regenerator for High Speed Optical Transmission Systems," in *Proceedings ICC '83*, Boston, MA: June 19-22, 1983, pp. 686-689.
- [DEC80] DEC, INTEL, and XEROX, "The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specification," available from DEC Inc., Intel Inc. and Xerox Inc., Tech. Rep. version 1.0, September 30, 1980.
- [Eise72] Eisenberg, M., "Queues with Periodic Service and Changeover Time," *Operations Research*, Vol. 20, 1972, pp. 440-451.

- [Epwo77] Epworth, R.E., "ITT 140 Mbits Optical Fiber System," in *Proceedings National Telecommunications Conference*, Los Angeles, CA: December 1977.
- [Eswa81] Eswaran, K.P., V.C. Hamacher, and G.S. Shedler, "Collision-free access control for computer communication bus networks," *IEEE Trans. Soft. Eng.*, Vol. SE-7, 1981, pp. 574-582.
- [Farm69] Farmer, W.D. and E.E. Newhall, "An experimental distributed switching system to handle bursty computer traffic," in *Proceedings ACM Symposium on Problems in the Optimization of Data Communications*, October 1969, pp. 1-33.
- [Frat81] Fratta, L., F. Borgonovo, and F.A. Tobagi, "The EXPRESS-NET: A Local Area Communication Network Integrating Voice and Data," in *Proceedings International Conference on Performance of Data Communication Systems and Their Applications*, Paris, France: September 1981.
- [Frat83] Fratta, L., "An Improved Access Protocol for Data Communication Bus Network with Control Wire," in *Proceedings ACM Communication Protocols Symposium*, Austin, Texas: March 1983, pp. 219-225.
- [Heym83] Heyman, D.P., "Data-Transport Performance Analysis of Fasnets," *Bell System Technical Journal*, Vol. 62, No. 8, October 1983, pp. 2547-2560.
- [Hopk80] Hopkins, G.T., "Multimode Communications on the MITRENET," *Computer Networks*, No. 4, 1980, pp. 229-233.
- [IEEE82] IEEE, *IEEE Project 802, Local Network Standards Draft C*: IEEE, June 1982.
- [Jone76] Jones, J.R. and D.F. Hemmings, "Optical Fiber T-Carrier Transmission System," in *Proceedings National Telecommunications Conference*, Birmingham, Alabama: December 1976.
- [Kawa77] Kawasaki, B.S. and K.O. Hill, "Low-loss access coupler for multi-mode optical fiber distribution networks," *Applied Optics*, Vol. 14, July 1977, pp. 1794-1795.
- [Klei77] Kleinrock, L. and M. Scholl, "Packet switching in radio networks: New conflict-free multiple access schemes for a small number of data users," in *Proceedings ICC*, Chicago, Illinois: June 1977, pp. 22.1-105 - 22.1-111.
- [Konh74] Konheim, A.G. and B. Meister, "Waiting Lines and Times in a System with Polling," *Journal of the Association for Computing Machinery*, Vol. 21, No. 3, July 1974, pp. 470-490.

- [Lam75] Lam, S.S. and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," *IEEE Transactions on Communications*, Vol. COM-23, September 1975.
- [Leho81] Lehoczky, J.P., L. Sha, and E.D. Jensen, "A Comparative Study of Variable TDM Schemes and CSMA Schemes," Department of Statistics, Carnegie-Mellon University, Pittsburgh, PA, Tech. Rep. 27, August 1981.
- [Limb82] Limb, J.O. and C. Flores, "Description of Fasnet - A Unidirectional Local-Area Communications Network," *The Bell System Technical Journal*, Vol. 61, No. 7, September 1982, pp. 1413-1440.
- [Liss83] Lissack, T. and B. Maglaris, "Digital Switching in Local Area Networks," *IEEE Communications Magazine*, Vol. 21, No. 3, May 1983, pp. 26-37.
- [Liu75] Liu, M.T. and C.C. Reames, "The design of the Distributed Loop Computer Network," in *Proceedings International Computer Symposium*, Taipei, Taiwan: August 1975, pp. 273-282.
- [Loom73] Loomis, D.C., "Ring communication protocols," University of California, Irvine, CA, Tech. Rep. 26, January 1973.
- [Lute82] Lutes, G., "Optical Fiber Applications in the Nasa Deep Space Network," *Fiberoptic Technology*, September 1982, pp. 115-121.
- [Magl82] Maglaris, B. and T. Lissack, "Performance Evaluation of Interface Units for Broadcast Local Area Networks," in *Proceedings Comcon '82*, Washington, DC: September 1982, pp. 393-401.
- [Marh84] Marhic, M.E., "Combinatorial Star Couplers for Single-Mode Fibers," in *Proceedings FOC/LAN '84*, Las Vegas, NV: September 20, 1984.
- [Mark80] Mark, J.W., "Distributed Scheduling Conflict-Free Multiple Access for Local Area Communication Network," *IEEE Transactions on Communications*, Vol. COM-28, No. 12, December 1980, pp. 1968-1976.
- [Mars81] Marsan, M.A. and G. Albertengo, "C-Net: A Local Broadcast Communication Network Architecture," C.N.R. Progetto Finalizzato Informatica, Sottoprogetto P 1, April 1981.
- [Masu82a] Masuda, S. and T. Iwama, "Low-loss lens connector for single-mode fibers," *Applied Optics*, Vol. 21, No. 19, October 1982, pp. 3475-3483.

- [Masu82b] Masuda, S. and T. Iwama, "Single-mode fiber-optic directional coupler," *Applied Optics*, Vol. 21, No. 19, October 1982, pp. 3484-3488.
- [Merl79] Merlin, P.M. and A. Segall, "A Failsafe Distributed Routing Protocol," *IEEE Trans. on Communications*, Vol. COM-27, No. 9, September 1979, pp. 1280-1287.
- [Metc76] Metcalfe, R. and D. Boggs, "ETHERNET: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, Vol. 19, No. 7, July 1976, pp. 395-404.
- [Morl83] Morling, R.C.S. and G.D. Cain, "A Routing Protocol That Maintains Packet Sequency," in *Proceedings Mediterranean Electrotec. Conf. (MELECON 83)*, Greece: May 1983, p. A3.03.
- [Mull77] Mullins, J.H., "A Bell System Optical Fiber System - Chicago Installation," in *Proceedings National Telecommunications Conference*, Los Angeles, CA: December 1977.
- [Pfis82] Pfister, G.M. and B.V. O'Brien, "Comparison of a CBX to the local network," *Data Communications*, July 1982, pp. 103-113.
- [Pier72] Pierce, J.R., "Network for block switches of data," *Bell System Technical Journal*, Vol. 51, No. 6, July/August 1972, pp. 1133-1145.
- [Ping82] Pingry, J., "Local Area Networks in Fiber," *IFOC*, Summer 1982.
- [Raws78] Rawson, E.G. and R.M. Metcalfe, "Fibernet I: Multimode Optical Fiber for Local Computer Networks," *IEEE Transactions on Communications*, July 1978.
- [Raws82] Rawson, E.G. and R.V. Schmidt, "FIBERNET II: An ETHERNET-Compatible Fiber Optic Local Area Network," in *Proceedings 82 Local Net*, Los Angeles, CA: 1982, pp. 42-46.
- [Ross83] Ross, S.M., *Stochastic Processes*: John Wiley & Sons, Inc., 1983.
- [Rubi81] Rubin, I. and L.F. DeMoraes, "Polling Schemes for Local Communication Networks," in *Proceedings International Conference on Communications 1981*, Denver, Colorado: 14-18 June 1981, pp. 33.5.1-33.5.7.
- [Schm83] Schmidt, R.V., E.G. Rawson, R.E. Norton Jr., S.B. Jackson, and M.D. Bailey, "Fibernet II: A Fiber Optic Ethernet," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, No. 5, November 1983, pp. 702-711.

- [Shee79] Sheem, S.K. and T.G. Giallorenzi, "Single-mode fiber multiterminal star directional coupler," *Appl. Phys. Lett.*, Vol. 35, No. 2, July 1979, pp. 131-133.
- [Shoc80] Shoch, J. and J. Hupp, *Measured Performance of an Ethernet Local Network*. Xerox PARC Report, February 1980.
- [Swar81] Swartz, G.B, "Analysis of a Scan Service Policy in a Gated Loop System," in *Proceedings Meeting on Applied Probability - Computer Science - the Interface*, Atlant. Univ. Boca Raton, Florida: Jan 5-7, 1981, pp. 241-252.
- [Taka83] Takagi, A., S. Yamada, and S. Sugawara, "CSMA-CD with Deterministic Contention Resolution," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, No. 5, November 1983, pp. 877-884.
- [Tane81] Tanenbaum, A.S., *Computer Networks*, Englewood Cliffs, NJ 07632: Prentice-Hall, Inc., 1981.
- [Toba83] Tobagi, F.A. and M. Fine, "Performance of Unidirectional Broadcast Local Area Networks: EXPRESS-NET and FASNET," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, No. 5, November 1983, pp. 913-926.
- [Tsen82] Tseng, C. and B. Chen, "D-Net: A New Scheme for High Data Rate Optical Local Area Networks," in *Proceedings Globecom '82*, Miami, FL: November/December 1982, pp. 949-955.
- [Ulug81] Ulug, M.E., G.M. White, and W.J. Adams, "Bidirectional Token Flow System," in *Proceedings 7th Data Communications*, Mexico City, Mexico: October 27-29, 1981, pp. 149-155.
- [Yeh79] Yeh, J., "Simulation of Local Computer Networks," in *Proceedings 4th Conference in Local Computer Network*, Minneapolis, Minnesota: October 1979, pp. 56-66.

