# A Bandwidth Reducing Adapter for Token Rings

Ah Young P. Chung Tze Cheong

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montréal, Québec, Canada

September, 1990

ISBN 0-315-64713-2

# ABSTRACT


## A Bandwidth Reducing Adapter for Token Rings

Ah Young P. Chung Tze Cheong

Ring networks are a convenient and reliable way to connect various devices such as computers and other communicating interfaces. Different rings named according to their accessing schemes were developed. The most common rings are the token ring, the slotted ring and the register insertion ring. The token ring is the most popular configuration used. A comparative summary of the performance of the IEEE 802.5 ring, the Cambridge ring and the DLCN loop is presented to illustrate the differences in their accessing schemes.

Increasing the rate of token rings from 4 Mbps as specified by the IEEE 802.5 standard to 16 Mbps is welcomed as it will satisfy future communication needs and also avoid the congested 1 MHz bandwidth region. A bandwidth adapter is needed to save on the bandwidth required as well as to extend the reach of communicating links. Hence the design of a bandwidth reducing adapter for twisted pair wiring, is presented. Consisting of a transmitter and a receiver, it translates 16 Mbps Manchester codes into a modified duobinary signal thus achieving a bandwidth compression by a factor of two while being transparent to either communicating stations and also the rest of the network.

Lastly, the performance analysis of a token ring using the adapter is presented. The model assumes symmetric Poisson arrivals of voice and data traffic and allows for the retransmission of erroneous data packets. The conditional cyclic time approach is used in this approximate analysis. The performance of $M_{ary}$ PAM and multilevel partial response signaling is compared. The single and the multiple token release schemes are considered. The effects of various parameters such as the number of retransmissions allowed and the station latencies are investigated.

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. A.K. Elhakeem for his guidance and support throughout the course of this work. From his ideas and suggestions was born this work and through which he guided me till the end. With patience and dedication, he also allowed me to discover other aspects in the telecommunication field without loosing track of the present work. He was always attentive to new ideas and provided ample constructive suggestions. He allowed me to build up the necessary background and progress at my own pace. For all these, I would like to say thank you again.

I would also like to thank S. Bohm for cleaning up the circuit and provided the nice pictures in this work.

Lastly but not least, I would like to thank my parents for the constant encouragement support they provided, and J.L. for all the moral support I needed to bring this work to its end. Thank you all for your patience for the countless hours I could not spend with you.

*To my Family*

*and*

*J. L.*

# TABLE OF CONTENTS


LIST OF FIGURES ..... viii

LIST OF TABLES ..... xiii

FOREWORD ..... xiv

Scope of the thesis. ..... xiv

CHAPTER ONE LOCAL AREA NETWORKS ..... 1

1.1 Why Local Area Networks? ..... 1

1.1.1 The OSI model and the LAN. ..... 3

1.1.2 How are LAN's distinguished? ..... 4

1.2 Ring Networks. ..... 6

1.2.1 Reliability of rings. ..... 6

1.2.2 Ring latency. ..... 8

1.2.3 Ring surveys and taxonomy. ..... 9

1.3 Token rings. ..... 10

1.3.1 General ring operation. ..... 11

1.3.2 IEEE 802.5 token ring. ..... 12

1.3.2.1 IEEE 802.5 frame format. ..... 13

1.3.2.2 Token ring protocol. ..... 14

1.3.2.3 Performance analysis of token rings. ..... 15

1.3.3 Fiber Distributed Data Standard (FDDI). ..... 21

1.3.3.1 FDDI frame format. ..... 21

1.3.2.2 FDDI ring protocol. ..... 22

1.4 Slotted rings. ..... 25

1.4.1 The Cambridge ring. ..... 25

1.4.1.1 Cambridge ring protocol. ..... 27

1.4.1.2 Performance of Slotted rings. ..... 29

1.5 Register insertion rings. ..... 32

1.5.1 Distributed Loop Computer Network (DLCN). ..... 33

1.5.1.1 DLCN ring protocol. ..... 34

1.5.1.2 Performance of Register Insertion rings. ..... 35

1.6 Performance comparison of the different rings. ..... 40

CHAPTER TWO DESIGN OF A BANDWIDTH REDUCING ADAPTER ..... 43

2.1 Introduction. ........ 43
2.2 Modified duobinary (MD) signaling. ........ 44
2.3 Design guidelines and general operation. ........ 46
2.4 The Transmitter. ........ 47
2.4.1 Comparator. ........ 48
2.4.2 Zero crossing detector and clock recoverer. ........ 48
2.4.3 Data recoverer. ........ 49
2.4.4 Precoder. ........ 50
2.4.5 Modified duobinary generator. ........ 50
2.4.6 Low-pass filter. ........ 51
2.4.7 Line driver. ........ 52
2.5 The Receiver. ........ 52
2.5.1 TTL converter. ........ 53
2.5.2 Adaptive sampler. ........ 54
2.5.3 Data decoder. ........ 55
2.5.4 Manchester encoder. ........ 57
2.6 Overall design and results. ........ 58
2.7 Remarks. ........ 62

CHAPTER THREE PERFORMANCE ANALYSIS OF ADAPTER FOR A MIXED VOICE/DATA LOAD IN A TOKEN RING ........ 74

3.1 Introduction. ........ 74
3.2 Traffic model and accessing method. ........ 74
3.3 Analysis of model. ........ 75
3.4 Impact of demodulation on performance. ........ 80
3.4 Results. ........ 81
3.4 Conclusion. ........ 84

CHAPTER FOUR: SUMMARY AND CONCLUSIONS ........ 100

4.1 Summary and Conclusions ........ 100

REFERENCES ........ 104

APPENDIX ........ 112

## LIST OF FIGURES


| | | |
|---|---|---|
| Figure 1.1 | The OSI model and LAN. | 3 |
| Figure 1.2 | Four basic network topologies. | 5 |
| Figure 1.3 | (a) Ring braiding, (b) star wiring, (c) double ring and (d) double ring with loop-back. | 7 |
| Figure 1.4 | Schematic illustrating the different instants of token release in Single Packet (SP), Single Token (ST) and Multiple Token (MT) rings. | 12 |
| Figure 1.5 | Frame format of an IEEE 802.5 packet. | 13 |
| Figure 1.6 | Token ring model with cyclic server. | 16 |
| Figure 1.7 | Performance difference between the different ring accessing schemes for exponentially distributed packet length and a normalized ring latency of a = 0.1,1,10. | 20 |
| Figure 1.8 | Frame format of a FDDI packet. | 22 |
| Figure 1.9 | A typical Cambridge ring. | 26 |
| Figure 1.10 | Frame format of a Cambridge ring minipacket. | 27 |
| Figure 1.11 | Frame format of the Basic Block Protocol using minipackets. | 29 |
| Figure 1.12 | Model of slotted ring with the gap equally redistributed. | 30 |
| Figure 1.13 | Normalized average transfer delay of a Cambridge ring with respect to its throughput. | 32 |
| Figure 1.14 | Buffers of the register insertion ring interface. | 33 |
| Figure 1.15 | Frame format of a DLCN packet. | 34 |
| Figure 1.16 | Detailed schematic of a DLCN station interface. | 35 |
| Figure 1.17 | Submodel of the register insertion ring with (a) the submodel, (b) the simplified submodel, and (c) the equivalent submodel. | 37 |
| Figure 1.18 | Mean transfer characteristics of station and ring priorities and comparison with first order approximation. | 40 |
| Figure 1.19 | Comparative performance of the single token ring (ST), the slotted ring (ES), and the register insertion ring (RI) for 10 and 100 stations with a normalized latency in (a) of a = 1, and in (b) of a = 0.1. | 42 |
| Figure 2.1 | Modified duobinary signal in both frequency and time domain. | 45 |
| Figure 2.2 | Location of adapter on the ring. | 46 |

| | | |
|---|---|---|
| Figure 2.3 | Block diagram of the transmitter. | 47 |
| Figure 2.4 | Comparator. | 48 |
| Figure 2.5 | Zero crossing detector and clock recoverer. | 49 |
| Figure 2.6 | Clock and data recovery traces. | 49 |
| Figure 2.7 | Data recoverer. | 50 |
| Figure 2.8 | Precoder. | 50 |
| Figure 2.9 | Modified duobinary generator. | 51 |
| Figure 2.10 | Standard low-pass filter. | 52 |
| Figure 2.11 | Line driver. | 52 |
| Figure 2.12 | Block diagram of the receiver. | 53 |
| Figure 2.13 | TTL converter. | 54 |
| Figure 2.14 | Adaptive sampler. | 55 |
| Figure 2.15 | Bit sampling by adaptive sampler. | 55 |
| Figure 2.16 | Data decoder. | 56 |
| Figure 2.17 | Timing schematics of the data recoverer operations. | 58 |
| Figure 2.18 | Manchester encoder. | 58 |
| Figure 2.19 | Design of transmitter. | 59 |
| Figure 2.20 | Design of receiver. | 60 |
| Figure 2.21 | Design of pseudo-random generator. | 61 |
| Figure 2.22 | TTL Manchester signal and its zero-crossings. | 63 |
| Figure 2.23 | Recovered clock. | 64 |
| Figure 2.24 | Comparison of recovered and input data at transmitter. | 65 |
| Figure 2.25 | Unfiltered modified duobinary signal. | 66 |
| Figure 2.26 | Filtered modified duobinary signal. | 67 |
| Figure 2.27 | Eye-diagram of the filtered and unfiltered modified duobinary signal. | 68 |
| Figure 2.28 | Frequency spectrum of input binary data. | 69 |
| Figure 2.29 | Frequency spectrum of unfiltered modified duobinary signal. | 70 |
| Figure 2.30 | Frequency spectrum of filtered modified duobinary signal. | 71 |
| Figure 2.31 | Modified duobinary signal at receiver. | 72 |

| | | |
|---|---|---|
| Figure 2.32 | Recovered data as compared to the original input data at receiver. | 73 |
| Figure 3.1a | Performance of data and voice packets for Multiple Token release and PAM signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 86 |
| Figure 3.1b | Performance of data and voice packets for Single Token release and PAM signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 86 |
| Figure 3.1c | Performance of data and voice packets for Multiple Token release and duobinary signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 87 |
| Figure 3.1d | Performance of data and voice packets for Single Token release and duobinary signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 87 |
| Figure 3.2a | Performance comparison of data packets of PAM and duobinary signaling for Multiple Token release (SNR = 100, $\rho_v = 0.2$, m = 0). | 88 |
| Figure 3.2b | Performance comparison of voice packets of PAM and duobinary signaling for Multiple Token release (SNR = 100, $\rho_v = 0.2$, m = 0). | 88 |
| Figure 3.2c | Performance comparison of data packets of PAM and duobinary signaling for Single Token release (SNR = 100, $\rho_v = 0.2$, m = 0). | 89 |
| Figure 3.2d | Performance comparison of voice packets of PAM and duobinary signaling for Single Token release (SNR = 100, $\rho_v = 0.2$, m = 0). | 89 |
| Figure 3.3a | Performance comparison of Multiple Token and Single Token release schemes for data packets using PAM signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 90 |
| Figure 3.3b | Performance comparison of Multiple Token and Single Token release schemes for voice packets using PAM signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 90 |
| Figure 3.3c | Performance comparison of Multiple Token and Single Token release schemes for data packets using duobinary signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 91 |
| Figure 3.3d | Performance comparison of Multiple Token and Single Token release schemes for voice packets using duobinary signaling (SNR = 100, $\rho_v = 0.2$, m = 0). | 91 |
| Figure 3.4a | Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 2$, SNR = 5 and $\rho_v = 0.2$. | 92 |
| Figure 3.4b | Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 4$, SNR = 25 and $\rho_v = 0.2$. | 92 |
| Figure 3.4c | Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 8$, SNR = 100 and $\rho_v = 0.2$. | 93 |

| | | |
|---|---|---|
| Figure 3.4d | Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 2$, SNR = 9 and $\rho_v = 0.2$. | 93 |
| Figure 3.5a | Effect of packet retransmissions on packet delay for PAM using Multiple Token release (SNR = 45, $\rho_v = 0.2$). | 94 |
| Figure 3.5b | Effect of packet retransmissions on data load $\rho_d$ for PAM using Multiple Token release (SNR = 45, $\rho_v = 0.2$). | 94 |
| Figure 3.5c | Effect of packet retransmissions on packet delay for PAM using Single Token release (SNR = 45, $\rho_v = 0.2$). | 95 |
| Figure 3.5d | Effect of packet retransmissions on data load $\rho_d$ for PAM using Single Token release (SNR = 45, $\rho_v = 0.2$). | 95 |
| Figure 3.6a | Effect of packet retransmissions on packet delay for duobinary signaling using Multiple Token release (SNR = 45, $\rho_v = 0.2$). | 96 |
| Figure 3.6b | Effect of packet retransmissions on data load $\rho_d$ for duobinary signaling using Multiple Token release (SNR = 45, $\rho_v = 0.2$). | 96 |
| Figure 3.6c | Effect of packet retransmissions on packet delay for duobinary signaling using Single Token release (SNR = 45, $\rho_v = 0.2$). | 97 |
| Figure 3.6d | Effect of packet retransmissions on data load $\rho_d$ for duobinary signaling using Single Token release (SNR = 45, $\rho_v = 0.2$). | 97 |
| Figure 3.7a | Effect of different station latencies on delay for Multiple Token release scheme using PAM (SNR = 100, $M_{ary} = 2, 4$ and 8, m = 0). | 98 |
| Figure 3.7b | Effect of different station latencies on delay for Multiple Token release scheme using duobinary signaling (SNR = 100, $M_{ary} = 2, 4$ and 8, m = 0). | 98 |
| Figure 3.7c | Effect of different station latencies on delay for Single Token release scheme using PAM (SNR = 100, $M_{ary} = 2, 4$ and 8, m = 0). | 99 |
| Figure 3.7d | Effect of different station latencies on delay for Single Token release scheme using duobinary signaling (SNR = 100, $M_{ary} = 2, 4$ and 8, m = 0). | 99 |

## LIST OF TABLES


| | | |
|---|---|---|
| Table 1.1 | Main differences between a LAN and a WAN. | 2 |
| Table 1.2 | Classification of different ring networks. | 9 |
| Table 1.3 | Description of the fields in the IEEE 802.5 frame. | 14 |
| Table 1.4 | Differences between the IEEE 802.5 and FDDI. | 21 |
| Table 1.5 | Description of the fields in the FDDI frame. | 22 |
| Table 1.6 | Description of the fields in the minipacket. | 27 |
| Table 1.7 | Description of the fields in the DLCN frame. | 34 |
| Table 2.1 | Truth table of the modified duobinary generator. | 51 |
| Table 2.2 | Truth table of the combinational circuit in data recoverer. | 57 |

# *Foreword*

The evolution of mankind has been marked by distinct periods such as the Stone Age, the Bronze Age and the Iron Age. They reflect the different technologies that have changed its way of living and the structure of its society. The term Information Age has been used to describe this present era. Indeed the advent of computers has started an unprecedented need for information gathering, processing and distribution. This proliferation of computers and the need for information have led to the birth of local networks.

**Scope of the thesis.**

In this thesis, an overview of the main features of local area networks will be given in the first chapter. The thesis concentrates on one type of LAN which is becoming increasingly popular: Ring networks. Three types of ring networks are considered. They are (1) the token ring, (2) the slotted ring, and (3) the register insertion ring. A brief description of the rings and their performance analysis is included.

This introduction to rings overlooks another important aspect of rings and networks in general, that of connectivity cost. To decrease connectivity costs, the use of cheap twisted pair for data transmission is considered. Twisted pair is however prone to EMI and wastes bandwidth at high rate. Hence in chapter two, the design of an adapter using twisted pair for data transmission in a token ring is considered. The adapter uses a modified duobinary signaling scheme to increase the bandwidth efficiency of the transmission. A theoretical analysis and performance evaluation of a token ring using such an adapter will follow in chapter three. The last chapter is of course reserved for the conclusion of the thesis.

# Chapter One
## Local area networks: rings

### 1.1. Why Local Area Networks ?

Originally a computer consisted of a single mainframe stored in a room with a few peripheral devices close by. Only big organizations could afford them and access was reserved to only a selected few. With the progress in Very Large Scale Integration (VLSI) technology and the decrease in price due to mass production, computers have evolved into smaller and smaller machines with an increasing processing power and at a much lower affordable price. Their accessibility and their ease of use as compared to the timesharing mainframe have led to a proliferation of computers in all places where there is a need for some computing power to process, gather or forward any information. This proliferation at a given site, often an office or workplace soon led to the need to connect them to one another for obvious reasons: (i) The ability for one user to communicate with another allows the sharing and/or exchanging of any relevant data, (ii) The sharing of expensive peripherals such as printers and disk storage devices maximizes their utilization, and (iii) The sharing of softwares, libraries and other tools from central data banks fosters a uniformity in working procedures and facilitates their maintenance and upgrading.

To fulfill these needs the network had to satisfy the following requirements.

1. The network should be fast to allow large data transfer from one device to another within a tolerable delay for the user.

2. The data transferred should be practically error-free.

3. The addition or removal of any device from the network should be easy and should not affect the network or at most only momentarily.

Thus was born the concept of a local network, local because of the various devices connected were usually found within a campus, an office or a building. Stallings in [1] defined a local network as

> 'a communication network that provides interconnections to a variety of data communicating devices within a small area'.

The term 'data communicating devices' is not restricted only to computers, terminals or other peripherals. It also covers other devices such as process control monitors, data gathering devices and digital telephone systems. Nowadays, the term Local Area Network (LAN) is also often used as opposed to Wide Area Network (WAN) which is a long-haul network and covers a much larger geographical area (see Table 1.1). An example of a WAN is the ARPANET which was designed by the Advanced Research Projects Agency of the U.S. Department of Defense in the late 1960's [3] and which now links more than a hundred computers around the globe.

**Table 1.1. Main differences between a LAN and a WAN [2].**

| *Description* | **LAN** | **WAN** |
|---|---|---|
| *Range* | < 10 km. | > 100 km. |
| *Cost* | Low (incremental) | High |
| *Installation* | Cheap | Very expensive |
| *Maintenance* | Cheap | Expensive |
| *Routing* | None or very simple | Complex |
| *Bit rate* | Megabit range | Kilobit range |
| *Error rate* | Very, very low | Low |
| *Delay* | Milliseconds | Seconds |
| *Topology* | Simple | Usually irregular |

There has been lots of effort in the past decade to develop an adequate technology to bridge the gap existing between the LAN and the WAN. These networks called Metropolitan Area Networks (MAN) cover the intermediate range (10-100 Km) and use optical fibers as the transmission medium. MAN's and WAN's will not be discussed in this thesis.

### 1.1.1. The OSI model and the LAN.

The proliferation of computers brought a variety of manufacturers and vendors to the market with an even larger variety of products. Often, the computers of different manufacturers or even of different models were not compatible with the rest as they each have their own data formats and data exchange conventions. Linking them into a network was a major headache. Thus the International Organization for Standardization (ISO) developed an open architecture model that would allow any two systems conforming to their reference model communicate with each other [4]. The Open Systems Interconnection (OSI) model consists of seven layers. Each layer performs a distinct subset of functions required to communicate with another system. The layers are stacked vertically such that a lower layer services the next upper layer by performing a subset of more primitive functions thus concealing them from the upper layer. The different layers of the OSI are shown in Fig. 1.1.

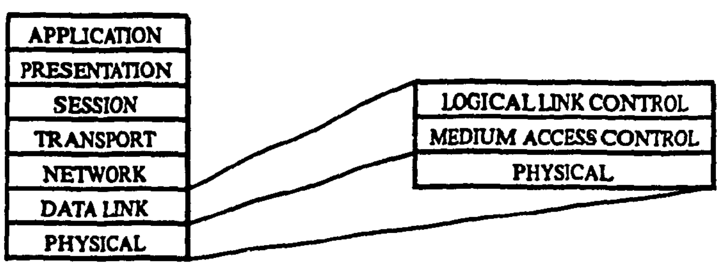



**Fig. 1.1. The OSI model and LAN.**

As LAN's are point-to-point connections, they are concerned by only the bottom two layers of the ISO model, namely the Physical layer and the Data Link layer [5].

The Physical layer is concerned with the transmission of data, bit by bit over a direct physical connection. It also describes the design specifications of all physical features of the signals (amplitude, period, frequency spectrum and modulating schemes) as well as any other mechanical, electrical or procedural characteristics required to establish, maintain and deactivate the physical link.

The Data Link layer for LAN's can de divided into two sublayers, the Logical Link Control (LLC) and the Medium Access Control (MAC), as shown in Fig. 1.1. The LLC sublayer is concerned about the type of service (with connection or connectionless) offered and the number of logical interfaces (users) being served at that particular station. The MAC sublayer is the most important feature that characterizes a LAN. The MAC assembles data bits into frame, appending all other necessary fields such as those for address and error checking. It also disassembles the received frames from the Physical layer, interprets them as in the case of control frames, and check for errors. Thus the MAC ensures the proper management of the communication link. There can be several different implementations of the MAC for the same LLC, each yielding a different LAN's accessing scheme.

In the same effort to streamline the LAN technology the ISO has adopted four LAN standards in 1987 [6-9]. They were drafted by the IEEE 802 committee and were already adopted by the American National Standards Institute (ANSI) in 1985 [10-13]. They are

1. IEEE 802.2 standard for the LLC [10],
2. IEEE 802.3 standard for the CSMA/CD for random accessing LAN's [11],
3. IEEE 802.4 standard for the Token Bus [12], and
4. IEEE 802.5 standard for the Token Ring [13].

These standards have been welcomed as manufacturers are more willing to make the necessary investments for a product which is more widely accepted by customers, nationally and internationally.

### 1.1.2. How are LAN's distinguished?

There has been numerous attempts to systematically classify LAN's in the literature [14-17]. For example, in [17], Thurber and Freeman came down to seven categories by considering

(1) the context in which the LAN was developed, (2) the reasons for its development, and (3) some aspects of the communication technology used. However all these taxonomies (as these systems for classification are called) found limited success as no single category could really represent the diversity of the LAN's. Hence, instead it is easier to identify physical characteristics common to some LAN's by which they can be discriminated from others. Three main features by which a LAN can be discriminated are (i) Topology, (ii) Transmission medium, and (iii) Medium Access Control.

LAN's have a simple structured topology. They can be divided into four basic network topologies, namely (1) bus, (2) tree, (3) star and (4) ring as shown in Fig. 1.2. This simplicity in structure reduces protocol overheads and facilitates their management. Of these topologies, ring networks have generated the most enthusiasm in the past decade. They will be dealt in greater details in the remaining of this chapter.

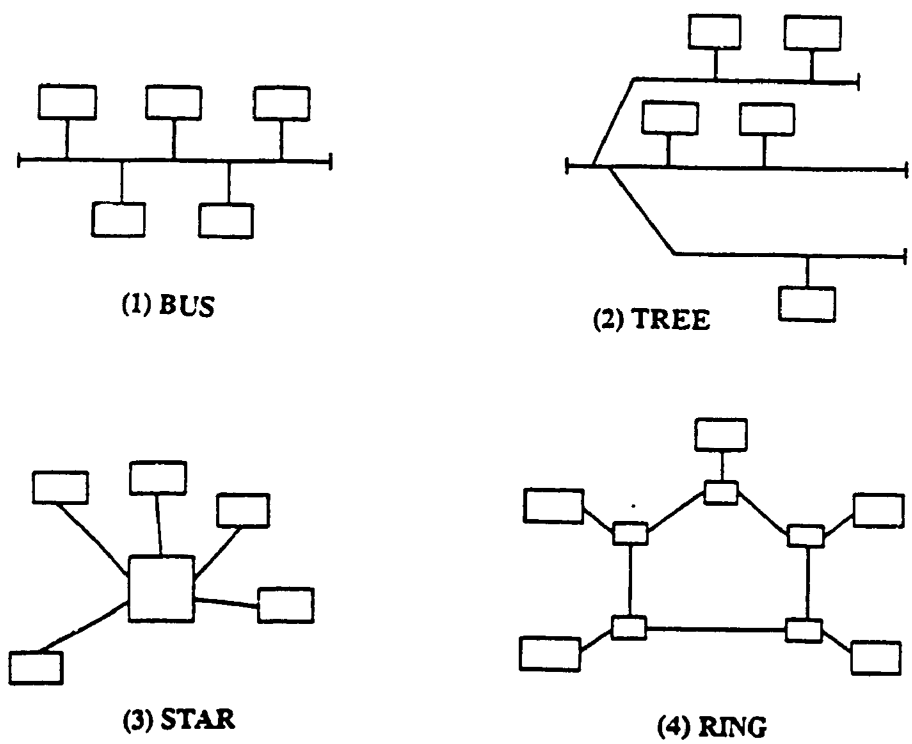



**Fig. 1.2. Four basic network topologies [1].**

The three types of transmission medium that are most frequently used in LAN connections are (i) Twisted pair wires, (ii) coaxial cables, and (iii) optical fibers. Of these, the twisted pair is the cheapest and is popular for short connection distances. The coaxial cable is the most popular as it can extend to a longer distance. The optical fibers can extend to even greater distances but still require opto/electrical interfaces which are still unproven and expensive.

Medium access control (MAC) can be broadly divided into two categories (i) Random Access Schemes and (ii) Controlled Access Schemes. The accessing schemes determine the protocol used by all stations to be able to converse via the transmission medium. Random access schemes are probabilistic and have less structure. Controlled access schemes can be either centralized or distributed. MAC's are usually quite dependent on the topology of the particular LAN as each topology will favor a particular type of MAC.

## 1.2. Ring networks

Ring networks have generated a lot of interest in the past decade [18,19]. The ease of integrating optical fiber technology into rings has turned them into the prime candidate for high speed connections for LAN's, inter LAN and MAN's communications [20]. Ring networks provide fair access and predictable perfomance, even under high loads. They can also accommodate both asynchronous and synchronous transmissions and can thus provide integrated services for more versatile types of communications such as packetized voice [21,22].

### 1.2.1 Reliability of rings.

There has been many attempts to solve the inherent drawback of single ring networks, their vulnerability to link or station failure. Three main techniques have been used.

The first technique due to Hafner [23] involves the addition of by-passing links all over the ring to join any two or more consecutive stations (see Fig. 1.3a) [24,25]. This braiding involves considerable wiring and a routing algorithm is now required as there is more than one path to and from any station. Raghavendra calculated the optimum distance between stations for a given number of stations in a ring [26]. However this optimality is hard to keep as the network grows because rewiring of the whole network will be required.

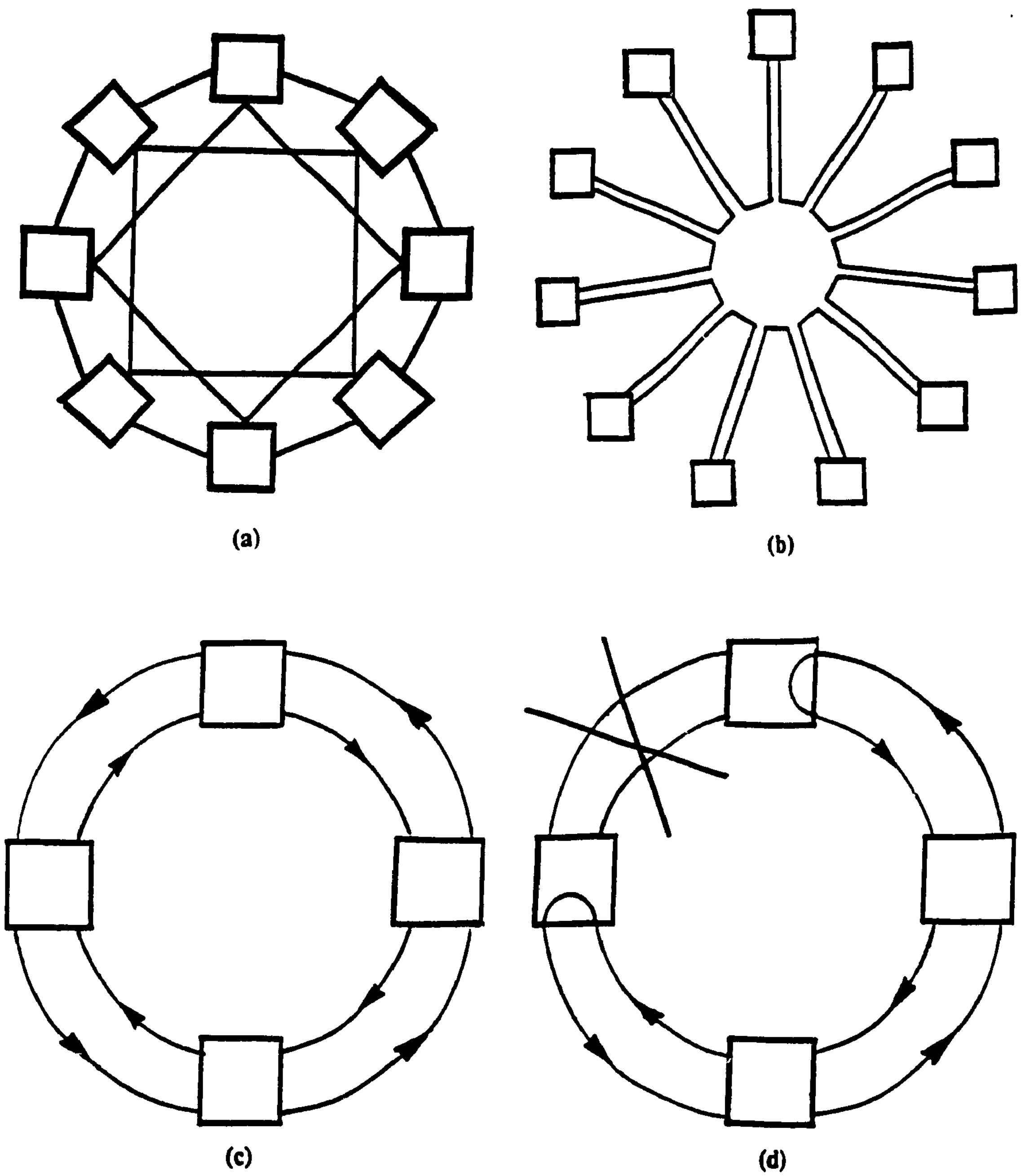


Fig. 1.3 (a) Ring braiding, (b) star wiring, (c) double ring, and (d) double ring with loop-back.

The second technique is the star-shaped ring, whereby all the connections to and from individual stations converge to a wire center or concentrator unit (see Fig.1.4b). Although this technique does not increase the fault tolerance of the ring as such, it greatly facilitates the isolation of any faulty link or station, allows its by-pass, and generally simplifies the installation and maintenance of the network [27]. An example of this topology is the IBM token ring where several wire concentrators are used [28].

The last technique is the double ring, where two different strategies have been developed. The first strategy uses a second ring as a stand-by subordinate ring where messages are routed when a link in the main ring fails (see Fig. 1.3c) [29]. This redundancy, though simple, is inefficient and hence avoided. In the second and most promising strategy, the double ring may be used as two counter-rotating rings each carrying information in the opposite direction. Automatic loop-back occurs whenever a faulty link or transmission is detected as shown in Fig. 1.3d [30-32]. Medium access protocol allowing almost a hundred percent ring utilization has been developed [33].

Of these three, it can be noted that one technique does not exclude the others. For example, the IBM token ring uses both a double counter-rotating loop with star-wiring to improve its reliability and serviceability [49].

### 1.2.2 Ring latency.

An important feature of any ring network is its latency. Ring latency is defined as the sum of latencies of every station connected to the ring and the round-trip propagation delay. The station latency is the time required by a station to repeat a bit from the time it receives it. Station latencies are measured in terms of bits and a minimal station latency of one bit is usually observed. The ring latency is also measured in bits and is sometimes also known as *ring length*.

### 1.2.3 Ring surveys and taxonomy.

Rings have been extensively studied in the literature. In 1979, Penney and Baghadi did a general survey of ring networks and their performance [18]. In 1983, Tang and Zaky surveyed twenty-four rings and proposed a taxonomy for their access protocols [19]. They identified two orthogonal dimensions namely time and space, which they each divided into three categories. In the time assignment, the instant at which a station has access to the transmission medium is distinguished. Thus, they identified (1) an ordered access, (2) an arbitrated access, and (3) a random access. Similarly, in the space assignment the message length is distinguished. They divided the messages into (1) a fixed size, (2) a bounded but variable size, and (3) an unbounded size. Table 1.2 shows their classification of the rings surveyed [19]. Each category has been identified by its access protocol. The ultimate objective of their taxonomy was to find a unified model which could be used for performance calculations of the local ring networks [35].

**Table 1.2. Classification of different ring networks [19].**

| *space assignment* / *time assignment* | F(fixed) | B(bounded variable) | U(unbounded) |
|---|---|---|---|
| O (ordered access) | (STDM)<br>IBM 2790 [35] | | (token)<br>DCS [47]<br>RINGNET[48]<br>IBM ZURICH RING [49]<br>C & C OPTONET [50]<br>TIMED-TOKEN RING [51] |
| A (arbitrated access) | (DASD)<br>WELLER RING [36] | | NEWHALL RING [52]<br>MLNET [18]<br>ORION [54] |
| R (random access) | (slot)<br>PIERCE RING [37]<br>SPIDER [38]<br>NSA RING [39]<br>CAMBRIDGE RING [40]<br>DATA HIGHWAY [41]<br>CARTHAGE [42]<br>IOWA RING [43] | (register insertion)<br>DLCN [44]<br>DDLCN [45]<br><br>(slot and register)<br>TORNET [46] | (CSMA-CD)<br>M.I.T. RING [27]<br>WESTN. ILLINOIS RING [55] |

In this thesis, this taxonomy will only be broadly followed. Only three of the most popular ring access protocols will be analyzed. They are (1) the token ring, (2) the slotted ring, and (3) the register insertion ring. Description and analysis of the other rings can be found in the annotated references.

## 1.3 Token rings.

Originally proposed by Farmer and Newhall in 1969 [52], token rings are by far the most popular ring network today. Although there are many reasons why token rings are so widely accepted (for example its performance, cost and simplicity), the primary reason for this popularity is standardization. Standardization channels the effort of the manufacturers to build a product that is compatible with its peers while increasing the level of confidence of its customers. There are three standards governing token rings. The first standard was adopted in 1982 in Europe by the European Computer Manufacturers Association (ECMA) and is known as ECMA-89 [56,57]. In 1985, the American National Standards Institute (ANSI) approved as national standard, the draft proposal submitted by the IEEE 802.5 committee on token rings [13]. Their work was greatly influenced by the european standard and in 1986 was ratified as an international standard by the ISO [9]. Works on a fourth standard based on token passing, is under way, to cater the need for a more efficient protocol with the advent of optical fibers. The Fiber Distributed Data Interface (FDDI) standards are being prepared by the Accredited Standards Committee (ASC) X3T9 for a general purpose high speed local or metropolitan area network [58-61]. Their work relies heavily on IEEE 802.5 standard to assure maximum compatibility with the slower networks.

In the following sections, the general characteristics of token rings will be covered. The IEEE 802.5 and the FDDI token rings will be used as examples.

### 1.3.1 General ring operation.

The token ring technique uses a unique bit pattern called token, that gives to its holder access to the transmission medium. In a simple system, the token may consists of one or more bits which will not be used for data transmission. Bit stuffing is performed if the token pattern occurs within the data to be transmitted.

There are two states in which any active station can be found, namely the idle state and the transmit state. In the idle state, the station acts as a repeater, regenerating incoming bits one by one on the outgoing link after a station latency time of usually one bit. In the transmit state, the station has a message to transmit and still acts as a repeater until it detects a free token. The token is then marked as busy and the station appends the necessary address and control fields and its data. A new token is then generated and released. Slight variations in the transmit state procedures will yield different types of token rings. They can be divided into three distinct processes. They are (1) the type of service, (2) the type of acknowledgement, and (3) the type of token release.

There are two types of service, namely the exhaustive and the non-exhaustive service. In the exhaustive service, the token holder is allowed to transmit all its accumulated packets before releasing the token while in the non-exhaustive service, packets up to a preset maximum number is allowed.

Token rings usually have an automatic acknowledgement protocol. Packets will circulate continuously in the ring if not removed. If the source station is responsible to remove the packets it sends, the destination station can set a bit after reading the packet, and the source station can check this bit before removing it from the line, thus achieving automatic acknowledgement. Otherwise, if it is the responsibility of the destination station, the latter can set a special bit on returning packets or it can send small acknowledgement packets, usually the header of the received packet, if there is no returning packets within a given time.

Lastly, three types of token release protocols can be observed (see Fig. 1.4). In the '*single* *packet'* (SP) token ring, the token is released after the source station has removed completely its own packet. In the '*single token*' (ST) token ring, the token is released as soon as it receives the header of its packet back. Lastly, in the '*multiple token*' (MT) token ring, the token is released as soon as packet transmission is over. Of these three protocols, it is obvious that the multiple token protocol allows the most efficient use of the ring's bandwidth, followed by the single token and the single packet protocols. The following sections will illustrate the implementation of the two token release schemes as the single packet scheme is rarely used.

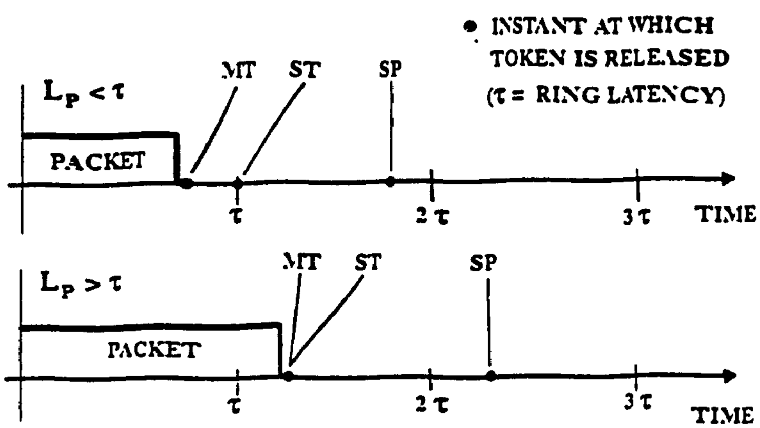



Fig. 1.4. Schematic illustrating the different instant of token release in single packet (SP), single token (ST) and multiple token (MT) rings.

### 1.3.2 IEEE 802.5 token ring.

The IEEE 802.5 token ring consists of a set of stations serially connected by a transmission medium. Information is transferred sequentially bit by bit, with each station regenerating and repeating each bit to the next active station. The station acts as an interface between devices and the medium for the purpose of communicating with other devices similarly attached to the ring. Access to the transmission medium is determined by a unique bit pattern called token which contain symbols that are violations to the differential Manchester encoding rule adopted by the standard. The token holder can transmit its message up to a maximum period of time as determined by the Token Holding Timer (THT), after which the token has to be released. Eight priority levels are available for different classes of services if required. A station assumes the role of network monitor with all the other stations as back-up. This ensures prompt error recovery through various error detection and recovery mechanisms. The IEEE 802.5 standard specifies a 1 or 4 Mbps transmission rate. The transmission medium is shielded twisted pairs using a differential Manchester encoding scheme for signaling.

### 1.3.2.1 IEEE 802.5 frame format.

The IEEE 802.5 general frame format is shown in Fig. 1.5.

| 8 bits | 8 bits | 8 bits | 16 or 48 bits | 16 or 48 bits | variable | 32 bits | 8 bits | 8 bits |
|---|---|---|---|---|---|---|---|---|
| **SD** | **AC** | **FC** | **DA** | **SA** | **INFO** | **FCS** | **ED** | **FS** |
| JKJKJKJK | PPPRRRTM | FFXXXYYY | | | | | JKJKJKIE | ACrrACrr |

**Fig. 1.5. Frame format of an IEEE 802.5 packet.**

The frame consists of a Start of Frame Sequence (SFS) and an End of Frame Sequence (EFS), with a Frame Control (FC) field, source and destination addresses, a variable-length information field, and a Frame Check Sequence (FCS) sandwiched in between. The token format consists only of a Starting Delimiter (SD), an Access Control (AC) field, and an Ending Delimiter (ED). Details of the various fields and their functions are given in the Table 1.3.

Apart from the information a station can extract from the various fields in a frame, each station on the ring has a number of timers which are used during the ring's initialization or recovery and to keep track of the general health of the network. It also has a latency buffer which ensures that there is at least a 24 bit latency in the ring and compensates for any phase jitter during transmission. Hence the IEEE 802.5 ensures that the single token transmission mode is observed.

**Table 1.3. Description of the fields in the IEEE 802.5 frame.**

| *Field* | *Contents and Description* |
|---|---|
| **Starting Delimiter (SD)** | Unique 8 symbol pattern indicating the start of a token or frame. The J non-data bit is a level signal having the same signal level as the preceding transition while the K non-data bit has the opposite level. To prevent the accumulation of DC components in the signals, the J and K bits are always paired. Both the J and K bits are violations to the differential Manchester encoding rule and hence can be easily detected. |
| **Access Control (AC)** | 3 priority bits (PPP) indicate the priority level of the frame or token.<br>3 reservation bits (RRR) allow stations with high priority messages to request the next token to be issued at their priority level.<br>1 token bit (T) indicates a token if it is set to 1 or a frame if it is set to 0.<br>1 monitor bit (M) is used by the active monitor to detect the continuous circulation of a high priority token or frame. |
| **Frame Control (FC)** | 2 frame control bits (FF) indicate whether it is a MAC or LLC frame. In case of a MAC frame, the remaining 6 bits are interpreted by all stations as control bits and the appropriate actions are taken. Otherwise if it is a LLC frame, the first three bits are reserved and set to 0 while the last 3 bits can be interpreted by the LLC entities as priority bits. |
| **Destination Address (DA)** | 16 or 48 bits address (depending on particular LAN implementation) identifies the particular station (with its first bit set to 0) or the particular group of stations (with its first bit set to 1) to which the frame is intended for. The broadcast address is when all 16 or 48 bits are set to 1. If a 48 bit address format is used, its second bit indicates whether the address is administered locally (when set to 0) or universally (when set to 1). No universal address administration exists for the 16 bit address format. |
| **Source Address (SA)** | Has the same length as the DA and identifies the station from which the frame originates. Its first bit is always set to 0. |
| **Information (INFO)** | 0,1 or more octets representing the data intended for the MAC, LLC or NMT depending on the FC field. Its maximum length depends on the maximum THT set for the station. |
| **Frame Check Sequence (FCS)** | 32 bit sequence Cyclic Redundancy Check (CRC) based on the bits in the FC, DA, SA, and INFO fields. |
| **Ending Delimiter (ED)** | Unique 8 symbol pattern indicating the end of a frame. The last two bits I and E, indicate if there is more frames (I = 1) or if it is the last or only frame (I = 0), or if there has been an error (E = 1). |
| **Frame Status (FS)** | 1 address recognized bit (A) is set to 1 if station recognizes its own address.<br>1 frame copied bit (C) is set to 1 if frame has been successfully copied.<br>2 reserved bits (rr) are set to 0 and are unused.<br>The remaining 4 bits duplicates the first 4 bits as these are not covered by the CRC and are used as a redundancy check to validate them. |

## 1.3.2.2 Token ring protocol.

When a station has a packet to transmit, it prefixes the appropriate FC, DA, and SA fields and queues it until a usable token is available. A usable token is a free token that has a priority level equal or less than that of the packet to be transmitted. If no usable token is available, the station can request a token at the required priority level, by setting the reservation bits in the passing token or frame. Meanwhile, the station will act as a repeater, looking out for any packets addressed to itself (including group, broadcast or control packets), checking for any errors, and waiting for its usable token. Relevant frames are copied and sent to the LLC sublayer or in the case of a MAC frame as indicated by the frame type bits, the appropriate actions are taken. When a usable token is received, the station converts it to a SFS by setting the token bit to 1. It stops repeating the incoming signal and starts the transmission of the frame. The FCS is accumulated and appended to the end of the information field. The station will remain in the transmit mode until it recognizes its own address on the SA field of the returning header when it releases the token with the same priority level it received. The transmitted frame or frames are removed from the line by the transmitting station.

If the transmitting station detects a reservation for a priority token, it will issue a token at the required priority level, making sure that it remembers the current priority level of the token being pre-empted. The requesting station can use the token and releases a new token at the same priority level after its transmission. Upon reception of this high priority level token, the transmitting station recognizes that the high priority token it has released previously, has made a complete cycle in the ring and hence removes it, and restores the original token. This mechanism assures fairness within each priority level for all stations and allows independent or dynamic bandwidth assignment depending upon the relative class of service required for a given message. For example, real-time voice (synchronous service), interactive sessions (asynchronous service) or network recovery (immediate service) can all be easily accommodated.

#### 1.3.2.3 Performance analysis of token rings.

Token rings have been extensively studied in the literature [63-73]. They have been readily modeled as queues representing stations, being served in a cyclic order with a generally non-zero switch-over time (see Fig. 1.6). As such, they can be regarded as a polling network and can be analyzed accordingly [62].

Two criteria can be used to discriminate the various approaches used in the analysis. They

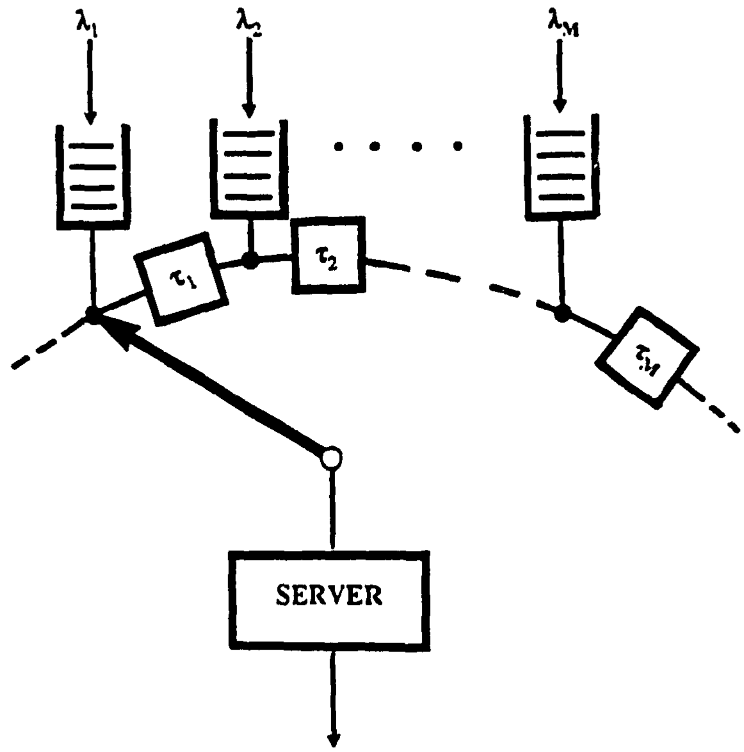



**Fig. 1.6. Token ring model with cyclic server.**

are the type of service (exhaustive or non-exhaustive) and the type of traffic (symmetric or asymmetric) to each station. Thus for the exhaustive service scheme, one of the earliest attempt was made by Konheim and Meister [63]. They used a discrete-time model and assumed a symmetric traffic to the stations. Other analysis used a discrete-time model with asymmetric traffic [64,65] or a continuous-time model and asymmetric traffic [66,67]. For the non-exhaustive service scheme, symmetric rings were analyzed in [68] while the asymmetric case were covered in [69,70]. Typically there is little difference in performance between exhaustive and non-exhaustive service schemes because the probability of having more than one packet accumulated at any station is very small, especially when the packets are equally distributed over the network [73].

The following discussion is based on the derivations given in [74]. The assumptions made are for mathematical tractability during the analysis and are

(1) The ring is symmetric, that is, the equally spaced stations have similar characteristics with equiprobable arrivals to each station.

(2) The arrival process is Poisson with a mean of $\lambda$.

(3) The packet length distribution is exponential for each station with mean $\bar{X}$ and a second moment $\overline{X^2}$, where $X$ denotes the random variable representing the packet length.

(4) The service scheme is exhaustive.

The transfer delay for token rings can be easily obtained from the average waiting time of a polling network by doing the proper modifications to reflect their differences. Thus borrowing from the polling network analysis [74]

$$W = \frac{Mw(1-S^*/M)}{2(1-S^*)} + \frac{S^*\overline{(X/R)^2}}{2(\bar{X}/R)(1-S^*)}, \tag{1.1}$$

where M - number of stations,

w - walk time,

$S^*$ - effective throughput of network,

and R - channel bit rate.

The general transfer time can be written as

$$T = \frac{\bar{X}}{R} + \frac{\tau}{2} + W, \tag{1.2}$$

where $\frac{\bar{X}}{R}$ represents the time required to transmit a packet, $\frac{\tau}{2}$ is the average ring latency as the packet has to travel over half of the ring on the average before reaching its destination, and $W$ is the waiting time for the server to become available.

Substituting for the walk time $w = \tau/M$, the average transfer time can be written as

$$T = \frac{\bar{X}}{R} + \frac{\tau}{2} + \frac{\tau(1-S^*/M)}{2(1-S^*)} + \frac{S^*\overline{E^2}}{2\bar{E}(1-S^*)}, \tag{1.3}$$

where $\bar{E}$ and $\overline{E^2}$ are the first and second moments of the effective service time, and $S^* = M\lambda\bar{E}$.

Hence by calculating the proper $\bar{E}, \overline{E^2}$ and $S^*$ and substituting in (1.3), the average transfer delay of the particular scheme can be obtained.

Thus for *single packet* (SP) operation, as all stations have to remove entirely the packet they have transmitted, the effective service time is always equal to the sum of the packet transmit time and the ring latency. Hence

$$\bar{E} = \frac{\bar{X}}{R} + \tau, \tag{1.4}$$

and

$$\overline{E^2} = \overline{\left[\frac{X}{R}\right]^2} + 2\tau\frac{\bar{X}}{R} + (\tau)^2, \tag{1.5}$$

where $\overline{\left[\frac{X}{R}\right]^2} = 2\left[\frac{\bar{X}}{R}\right]^2$ for exponentially distributed packet length. Now $S^*$ is given as

$$\begin{aligned} S^* &= M\lambda\bar{E} \\ &= M\lambda\frac{\bar{X}}{R} + M\lambda\tau \\ &= S + M\lambda\tau \end{aligned}$$

$$S^* = S(1 + a), \tag{1.6}$$

where $S = M\lambda\frac{\bar{X}}{R}$ and is the network throughput, and $a = \frac{\tau}{\bar{X}/R}$ is the normalized ring latency with respect to service time.

Substituting (1.4), (1.5) and (1.6) in (1.3), the transfer delay for single packet operation is given by

$$T_{SP} = \frac{\bar{X}}{R} + \frac{\tau}{2} + \frac{\tau[1-(1+a)S/M]}{2[1-(1+a)S]} + \frac{S\bar{X}/R\,[(1+a)^2+1]}{2[1-(1+a)S]}. \tag{1.7}$$

Normalizing with respect to the service time $\bar{X}/R$, equation (1.7) becomes

$$T^*_{SP} = 1 + \frac{a}{2} + \frac{a[1-(1+a)S/M]}{2[1-(1+a)S]} + \frac{S[(1+a)^2+1]}{2[1-(1+a)S]}. \tag{1.8}$$

It is obvious that the maximum throughput that can be obtained from (8) is given by $S = \frac{1}{1+a}$.

For *single token* (ST) operation, two distinct cases have to be considered as seen previously. Thus when ($\bar{X}/R \geq \tau$) corresponds to the multi-token operation, its transfer delay will be calculated in the next paragraph. For the case when ($\bar{X}/R < \tau$), the first and second moments of the effective service time can be written as [74]

$$\bar{E} = \frac{\bar{X}}{R}e^{-a} + \tau, \tag{1.9}$$

and

$$\overline{E^2} = (\tau)^2 + 2\left[\frac{\bar{X}}{R}\right]^2 e^{-a}(1+a). \tag{1.10}$$

Hence

$$\begin{aligned} S^* &= M\lambda(\frac{\bar{X}}{R}e^{-a} + \tau) \\ &= S(e^{-a} + a), \end{aligned} \tag{1.11}$$

and

$$T_{ST} = \frac{\bar{X}}{R} + \frac{\tau}{2} + \frac{\tau[1 - S(e^{-a} + a)/M]}{2[1 - S(e^{-a} + a)]} + \frac{\bar{X}}{R}\frac{S[(a)^2 + 2(1+a)e^{-a}]}{2[1 - S(e^{-a} + a)]}. \tag{1.12}$$

Normalizing with respect to the service time,

$$T^*_{ST} = 1 + \frac{a}{2} + \frac{a[1 - S(e^{-a} + a)/M]}{2[1 - S(e^{-a} + a)]} + \frac{S[(a)^2 + 2(1+a)e^{-a}]}{2[1 - S(e^{-a} + a)]}. \tag{1.13}$$

The maximum throughput achieved is given by $S = \dfrac{1}{(e^{-a} + a)}$.

For *multiple token* (MT) operation, the effective service time is given by

$$\bar{E} = \frac{\bar{X}}{R}, \tag{1.14}$$

and

$$\overline{E^2} = 2\frac{\overline{X^2}}{R^2}, \tag{1.15}$$

as the station having a packet to transmit releases the token as soon as its transmission is over. Thus $S^* = S$ and substituting in (1.3), the transfer delay is given by

$$T_{MT} = \frac{\bar{X}}{R} + \frac{\tau}{2} + \frac{\tau(1-S/M)}{2(1-S)} + \frac{S\bar{X}/R}{1-S}. \tag{1.16}$$

Normalizing (1.16) with respect to service time, the normalized delay is given as

$$T^*_{MT} = 1 + \frac{a}{2} + \frac{a(1-S/M)}{2(1-S)} + \frac{S}{1-S}. \tag{1.17}$$

From (1.17), the maximum throughput is achieved when $S = 1$.

Fig. 1.7 illustrates the performance differences between the three modes. It is obvious that the multiple token scheme is the most efficient and the single packet scheme as the least efficient of them. Moreover Fig. 1.7 illustrates the effect of ring latency on token rings. In all cases, a higher ring latency implies increasingly worst performance.

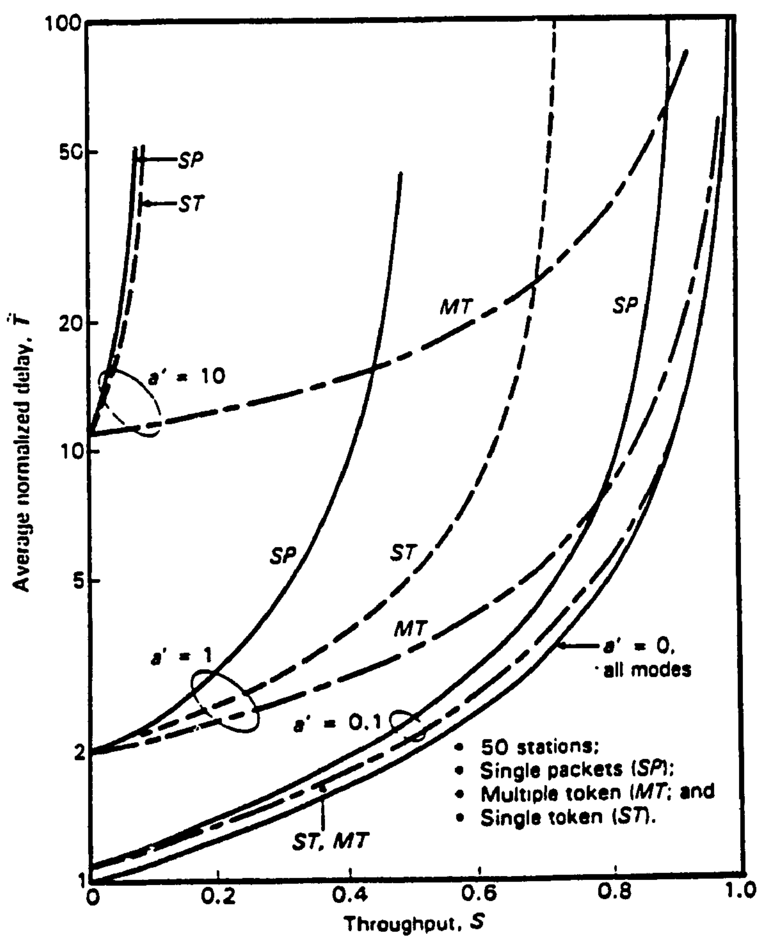



**Fig. 1.7. Performance difference between the different token ring accessing schemes for exponentially distributed packet length and a normalized ring latency of a = 0.1, 1, 10 [74].**

### 1.3.3 Fiber Distributed Data Standard (FDDI).

The Fiber Distributed Data Standard (FDDI) was developed to provide high performance connections among mainframes and its peripherals, to serve as a backbone network for other slower LAN's, and to satisfy heavy bandwidth users such as desktop image processing devices. It can handle both synchronous and asynchronous traffic and can accommodate diverse services such as real-time voice and video. The FDDI has been based on the IEEE 802.5 token ring protocol because (1) the token protocol has proved its effectiveness even at high loads, (2) the frame format compatibility facilitates internetworking between the high and lower speed IEEE 802.5 LAN, and (3) their similarities allow an easier understanding and implementation by those already familiar with the IEEE 802.5 standard. However, the FDDI has been optimized for high speed operation and differs in many aspects from the IEEE 802.5 standard as illustrated below in Table 1.4.

**Table 1.4. Differences between IEEE 802.5 and FDDI.**

| *Description* | IEEE 802.5 | FDDI |
|---|---|---|
| **Medium** | Twisted pair | Optical fiber |
| **Data rate** | 1 or 4Mbps | 100 Mbps |
| **Encoding** | Diff. Manchester | NRZI 4B/5B code |
| **Reliability** | No specifications | Explicit specifications |
| **Clock** | Centralized | Distributed |
| **Medium access** | Priority and reservation bits | Timed token rotation |
| **Token release** | After reception of header | Immediately after transmission |
| **Token capture** | Bit flipping | Complete removal by absorption |
| **Max. frame size** | Depends on THT | 4500 octets |
| **Address length** | 16 or 48 bits | 16 or 48 bits or both |
| **Error recovery** | Active monitor | Distributed |

#### 1.3.3.1 FDDI frame format.

The general frame format of the FDDI standard is shown in Fig. 1.8. It is similar to the IEEE 802.5 standard except that it uses a preamble (PA) before its Starting Delimiter (SD) for clock synchronization. Its token format consists of the PA, the SD, the Frame Control bits (FC), and the Ending Delimiter (ED). Table 1.5 provides more details of the various fields in the frame. Each symbol is equivalent to four bits.

| 16 symbols | 2 symb. | 8 bits | 16,48 bits | 16,48 bits | variable | 32 bits | 2 symb. | 1,2 symb |
|---|---|---|---|---|---|---|---|---|
| PA | SD | FC | DA | SA | INFO | FCS | ED | FS |

**Fig. 1.8. Frame format of a FDDI packet.**

**Table 1.5. Description of the fields in the FDDI frame.**

| *Field* | *Contents and Description* |
|---|---|
| **Preamble (PA)** | 16 non-data idle symbols transmitted by the frame originator and used for clock synchronization by the other stations. The length of PA will vary according to each local clock. |
| **Starting Delimiter (SD)** | 2 non-data symbols (JK) indicates the beginning of a frame or a token. |
| **Frame Control (FC)** | 1 bit (C) indicates whether frame is synchronous or asynchronous.<br>1 bit (L) indicates if address is 16 or 48 bits<br>2 bits (FF) indicates if this is a LLC frame or a MAC frame. In the latter case, the 4 remaining bits (ZZZZ) indicate the type of MAC frame. |
| **Destination Address (DA)** | 4 or 12 symbols indicating the station to which the frame is intended for. It has the same format as the DA in the IEEE 802.5. However the ring can accommodate both 4 and 12 symbol addresses. |
| **Source Address (SA)** | 4 or 12 symbols indicating the originator of the frame. |
| **Information (INFO)** | 0 or more symbol pairs containing LLC data or information related a control operation as specified by the FC field. |
| **Frame Check Sequence (FCS)** | 32 bit Cyclic Redundancy Check (CRC) sequence covering the DA, SA, and INFO fields. |
| **Ending Delimiter (ED)** | 2 non-data terminate symbols (TT) in a token or a single terminate symbol (T) in a frame indicating the end of the token or frame. |
| **Frame Status (FS)** | 1 error detected field (E)<br>1 address recognized field (A)<br>1 frame copied (C)<br>These fields use the Set (S) or Reset (R) symbols to represent their status. Additional fields can be present depending on the particular LAN implementation. A terminate symbol (T) may or may not be present to make the EFS an integral number of octets. |

### 1.3.3.2 FDDI ring protocol.

FDDI uses a timed token to support a mixture of stream and bursty traffic. Its basic operation is different from the IEEE 802.5 token ring in that when a station has a packet to transmit and detects a usable token, it will remove the token completely from the ring and start the transmission of the frame. After its transmission, the token is immediately released, allowing another station to use it while the frame is proceeding to its destination. Hence more than one frame may be circulating on the ring but only one token will be present. Thus FDDI uses the multiple token mode of operation. However, both standards are similar in the sense that the originator of the frame is responsible to remove its frame from the ring.

During the initialization process, stations on the ring negotiate a Target Token Rotation Time (TTRT) which will determine, on the average, how frequently they will see a token. The station requiring the fastest TTRT will win the claiming process and starts the ring operation by issuing a token. If two or more stations have the same highest TTRT, the one with the highest address magnitude will win. The FDDI protocol ensures that the time between consecutive token arrivals is equal to TTRT on the average, but never greater than 2TTRT [75,76]. Hence stations with time critical data will set their TTRT claims equal to half of their maximum tolerable delay.

Prior to any ring operations, each station carrying synchronous traffic is allocated an amount of bandwidth for synchronous transmission. The remaining bandwidth is then available for asynchronous transmission. This bandwidth allocation should satisfy the following constraint [1].

$$\sum_{i=1}^{i=all} SYN(i) + LEN + FRA + TOK \leq TTRT \tag{1.18}$$

where $SYN(i)$ - time allocated for synchronous transmission to station i,

$LEN$ - time required to go around the ring when idle,

$FRA$ - time required to transmit a maximum length frame (4500 octets),

$TOK$ - time required to transmit a token.

All stations supporting synchronous traffic have access to the transmission medium for its allocated $SYN(i)$ period when they capture a token. For asynchronous traffic, the time allocated for transmission will vary, depending on how early they receive the token as compared to the negotiated TTRT. The protocol is more complex and is explained in the next section.

During ring operation, each station monitors the time between consecutive token arrivals by using a timer called Token Rotation Timer (TRT). The TRT is initially set to zero and start counting until a token is received. The token is said to be early if the count on TRT is less than TTRT or is said to be late if the TRT exceeds TTRT. If the token is early, the count on the TRT is stored in another timer called Token Holding Timer (THT) and TRT is reset and resumes counting. When the station finishes transmitting its synchronous data up to its allocated SYN(i), THT is started and asynchronous data transmission can proceed until THT reaches TTRT. Thus THT determines the asynchronous bandwidth available to the station in this cycle. A token is issued thereafter. If the token is late, the TRT is reset to zero and a late counter LC is incremented by 1 to indicate that TTRT has expired. TRT resumes counting, measuring the lateness of the token. When the token arrives, LC is reset but TRT continues its count such that the the lateness of the token is accumulated throughout this token cycle. If LC exceeds 1, an error recovery procedure is initiated as it indicates that the token rotation time has exceeded 2*TTRT. No asynchronous transmission is allowed when LC is set. The TRT, on the long run, will average to TTRT by penalizing the stations that come after TTRT has been exceeded. Hence the notion of fairness as seen in the IEEE 802.5 is not present in FDDI, but line hogging is not possible.

There are also eight priority levels, TPR(i) where i = 1 to 8, for asynchronous transmission. Each TPR(i) specifies a THT time threshold above which the particular data will have access to the medium. Hence the larger the TPR(i), the higher the probability that a smaller THT will be found, and hence the higher probability of transmission thus implying a higher priority status. It is obvious that the maximum value of TPR(i) is TTRT. Another refinement of the asynchronous transmission is the use of a restricted token that will allow multiframe exchange between a group of stations on the ring. The total asynchronous bandwidth available can be used for large data transmission for an extended period of time. The restricted token operation is simple. First, the initiator of a restricted exchange captures an unrestricted, that is regular token and informs all its participants that a restricted dialogue will be initiated. Upon the capture of another regular token, it issues a restricted token which can only be used by the forewarned dialogue participants. The exchange can go on for an extended period of time where only these participants have exclusive use of all the asynchronous bandwidth available. Upon termination of the dialogue, all the participants are informed and an unrestricted token is released by the dialogue terminator. Hence only one restricted dialogue can be supported by the ring but synchronous transmission is always allowed by all stations, independent of the nature of the token.

## 1.4 Slotted Rings.

The slotted ring, also known as *empty slot ring*, was originally proposed by Pierce in 1972 [37]. Its concept is based on a number of slots of fixed length, circulating continuously around the ring such that any station wishing to transmit a message may seize any empty slot and use it for transmission. The number of slots that can be accommodated by a ring depends on the slot size and the ring latency. Slots are chosen to be small so that the network can accommodate at least one of them, but not too small to carry an excessively large overhead. By dividing the ring latency with the slot size, the maximum integral number of slots in the ring is obtained. A small gap which cannot fit a slot is frequently left. This gap is welcomed as it can be used for station synchronization. However the gap is kept to a minimum by the ring monitor by introducing artificial delays through a variable length register to fine-tune the ring latency.

There has been numerous implementations of the slotted ring [77-81]. The most popular of them is the Cambridge Ring and it will be described in the next sections.

### 1.4.1 The Cambridge ring.

The Cambridge ring was developed in 1977 by Hopper [40]. Improvements to the ring were implemented by Wilkes and Wheeler [77], and by Hopper and Wheeler [78] in 1979 . A typical Cambridge ring is shown in Fig. 1.9. It can be noted that the interface to the transmission medium consists of a repeater that is connected to a station unit and which in turn is connected to an access box with interfacing circuits to the host. A ring monitor is also present for ring

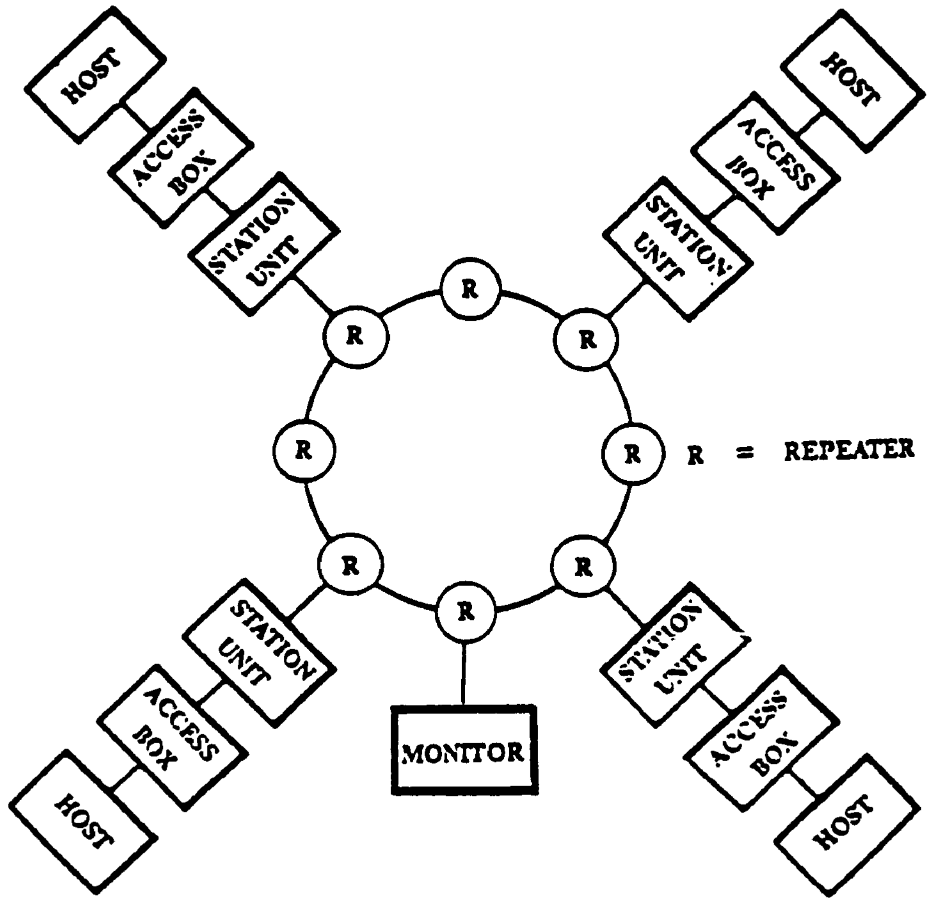



Fig. 1.9. A typical Cambridge ring.

initialization, general ring maintenance and error recovery. The Cambridge ring operates at a 10 Mbps rate. It uses a very small fixed-length packet, called *minipacket*, which is 38 bits long. Assuming a typical channel propagation delay of 5 $\mu s$ per kilometer and a station latency of 3 bits per station, a network of 30 stations will hold about 150 bits per kilometer.

The minipacket format is shown in Fig. 1.10. A minipacket always starts with a leading 1 to prevent the occurrence of a long stream of 0's used as gap bits. More details on the format of the minipacket is given in Table 1.6. It can be observed that of its 36 bits, the minipacket conveys only 16 bits of data. This large overhead is one of the drawbacks of the slotted ring but is partly compensated by the possible simultaneous transmission of more than one minipackets on the ring and a shorter access time.

| 1 | 1 | 1 | 8 bits | 8 bits | 16 bits | 2 | 1 |
|---|---|---|---|---|---|---|---|
| **1** | **F** | **M** | **DA** | **SA** | **INFO** | **RR** | **P** |

**Fig. 1.10. Frame format of a Cambridge ring minipacket.**

**Table 1.6. Description of the fields in the minipacket format.**

| *Field* | *Contents and Description* |
|---|---|
| **Start (S)** | 1 bit indicating the start of a minipacket and is always a '1'. It is also used by the ring interfaces for clock synchronization. |
| **Full/Empty bit (F)** | 1 bit indicating if slot is occupied or not. |
| **Monitor bit (M)** | 1 bit used by ring monitor to prevent the continuous circulation of a minipacket. |
| **Destination Address (DA)** | 8 bits indicating the station to which the minipacket is intended for. |
| **Source Address (SA)** | 8 bits indicating the station from which the minipacket originates. |
| **Information (INFO)** | 16 bits containing data or information. |
| **Response bits (RR)** | 2 bits indicating the state or response of the destination station.<br>(00) - destination is absent.<br>(01) - minipacket is accepted.<br>(10) - destination is deaf.<br>(11) - destination is busy. |
| **Parity bit (P)** | 1 bit parity check covering the minipacket for error detection. |

### 1.4.1.1 Cambridge ring protocol.

During the initialization period, bit level synchronization is first achieved by having the local oscillator of each station to lock on a bit stream emitted by the ring monitor. Slot synchronization is then performed by the framing circuit of each station unit as explained below. The ring monitor transmit slots with a leading 1 and all zeros. The framing circuit skips over gap bits until a 1 is encountered. It assumes it is the leading 1 of a slot and counts 38 bits. If the next bit is a 1 again, it counts another 38 bits as it assumes it is another slot and so on until eventually it encounters a 0. Now the framing circuit assumes that it is the gap and skips over any following zeros until a 1 is encountered again. By repeating this process, all the stations will eventually be locked onto the true gap of the ring and hence slot synchronization is achieved.

The station unit has separate transmission and reception circuits which are independently controlled by the access box. When a station wants to transmit a minipacket, the access box writes the destination address and the data in the transmit register. When an empty slot is detected, the minipacket is shifted onto the ring upon the proper signal from the access box. The source address and the control bits are automatically added by the station unit. After transmission, the station starts counting the number of slots passing by. As the number of slots in a given ring is known, it knows exactly when its slot has returned. The station marks it as empty and reads its response bits. Retransmission of the minipacket, if necessary will be decided by the access box. The station cannot reuse the same slot and hence line hogging is prevented.

On the receiving side, bits are shifted to a receive register parallelly as they flow through the repeater. A receiving circuit recognizes the minipackets addressed to the station and signals the access box. A source select register is also present and is used to decide accepting minipackets from (1) any station, (2) no station, or (3) a particular source station only. If a minipacket is accepted, it is kept in the receive register until cleared by the access box. The response bits are set appropriately to reflect the condition of the receiving station.

Because of the size of the minipacket, most messages have to be broken into many minipackets before being sent. The Cambridge ring uses a protocol known as the Basic Block Protocol (BBP) to perform this operation [82]. The basic block consists of a sequence of minipackets beginning with a header minipacket and a route minipacket followed by 1 to 1024 data minipackets. The basic block is terminated by a checksum minipacket for error checking purposes (see Fig. 1.11). By using the BBP together with the source select register, the destination station does not have to sort or reorder the minipackets received and can reconstruct the original message using the checksum. Higher level protocols especially developed for the Cambridge rings such as the single-shot protocol [83] and the byte stream protocol [84], are used in conjunction with the BBP to facilitate the data exchange between two stations [85].

Lastly, in an effort to improve the performance of the Cambridge ring, the use of short and long packets in a hybrid ring has been implemented in 1983 [86]. The short packets have an 8-

bit data field and is used mainly for control while the long packets have a data field of 64 bits. Another upgrade of thc ring called the Cambridge Fast Ring has been designed using VLSI technology [87]. It operates at 100 Mbps and uses optical fiber links.

| 1 | 1 | 1 | 8 bits | 8 bits | 16 bits | | | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | F | M | DA | SA | header 4 bits | type 2 bits | size 10 bits | RR | P |
| 1 | F | M | DA | SA | flags 4 bits | port number 12 bits | | RR | P |
| 1 | F | M | DA | SA | data data minipacket 1 | | | RR | P |
| | | | | | . . . | | | | |
| | | | | | . . . | | | | |
| 1 | F | M | DA | SA | data upto data minipacket 256 | | | RR | P |
| 1 | F | M | DA | SA | checksum | | | RR | P |

**Fig. 1.11. Frame format of the Basic Block Protocol using minipackets.**

### 1.4.1.2 Performance of slotted rings.

There has been various attempts to study the effect of slotted rings in the literature [88-95]. Hayes and Sherman developed two models in 1971 to study Pierce loops where a source station is allowed to reuse the same slots[88]. Their analysis proved to be quite accurate as shown by the simulation results obtained by Anderson *et al* [89]. The analysis by Spragins [90] confirmed the more general derivations perfomed by Konheim and Meister [63]. Bux [73] analyzed the slotted ring by using a 'processor-sharing' model developed by Kleinrock [91] and analyzing it as a simple mixed queueing network. These networks have been readily analyzed and are sometimes referred to as the BCMP model [92]. Other perfomance evaluations of the slotted ring, or more specifically the Cambridge ring, include those by King and Mitrani [94] and by Sorensen [95] where they considered the use of the BBP (Basic Block Protocol) for message transfer, in their analysis.

The heuristic approach in [74] makes the following assumptions: (1) there are independent Poisson arrivals at each station at a rate of $\lambda$ messages per second, (2) the message length are exponentially distributed with a mean of $\bar{X}$ bits, and (3) the message length is much larger than the minipacket length. The ring is assumed to have no gaps as the gap bits can be equally distributed between the slots and regarded as part of the header (see Fig. 1.12).

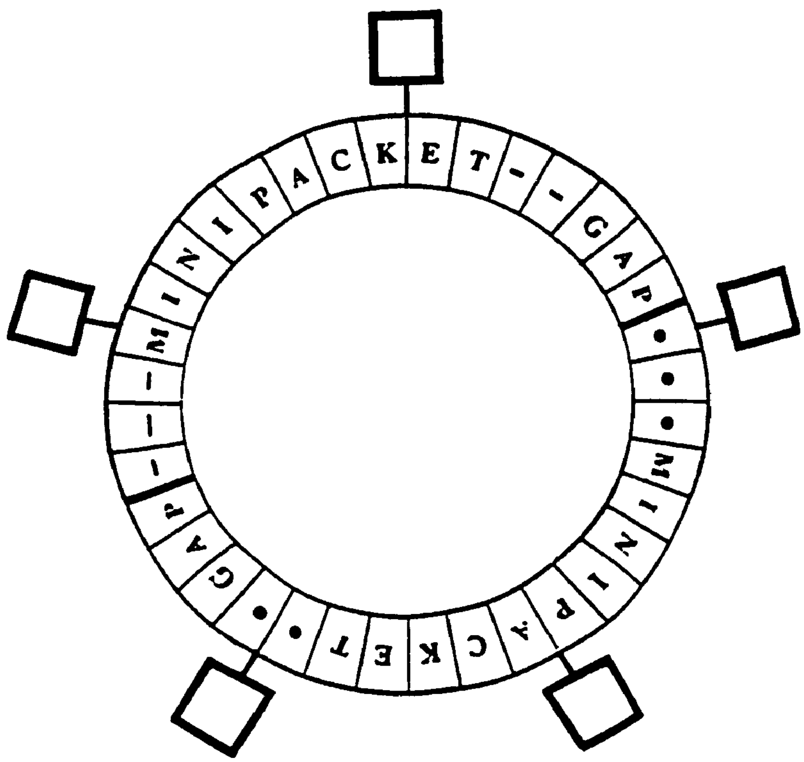


**Fig. 1.12. Model of slotted ring with the gap bits equally redistributed.**

The network is approximately modeled as distributed M/M/1 queues with an arrival rate of $M\lambda$ messages per second and a service time of $\bar{X}/R$ where M is the number of stations on the ring and R is the bit rate on the channel. As any station cannot reuse the same slot, the effective bit rate observed is actually R/2.

Using the average transfer delay for M/M/1 queues and modifying it for slotted rings, the transfer delay T can be written as

$$T = \frac{2}{1-S}\left[\frac{\bar{X}}{R}\right] + \frac{\tau}{2}, \tag{1.19}$$

where $S = \dfrac{M\lambda\bar{X}}{R}$ and denotes the throughput of the ring and $\dfrac{\tau}{2}$ has been added to denote the average ring latency experienced by any transmitted packet. Note that the average ring latency has been added in (1.17) to reflect the propagation delay of the minipacket. To take into account the overhead in a minipacket due to its header, an overhead factor 'h' has been defined, that is

$$h = \frac{L_h}{L_d}, \tag{1.20}$$

where $L_h$ is the length of the header and $L_d$ is the number of data bits per slot. Hence the average length of the minipacket can be considered as increased by a factor of $(1 + h)$ as well as the throughput. Thus equation (1.17) becomes

$$T_{ES} = \frac{2(1 + h)}{1-S(1 + h)}\left[\frac{\bar{X}}{R}\right] + \frac{\tau}{2}, \tag{1.21}$$

A similar expression has been more rigorously derived by Bux in [73] where an arbitrary packet length distribution has been assumed. Normalizing (1.19) with respect to the service time $\bar{X}/R$, the transfer delay can be written as

$$T^{*}_{ES} = \frac{2(1 + h)}{1-S(1 + h)} + \frac{a}{2}, \tag{1.22}$$

where 'a' denotes the normalized ring latency. Fig. 1.13 illustrates the relation between the throughput and the normalized average transfer delay. Note that the ring latency has been subtracted in the vertical scale as it is independent of the throughput.

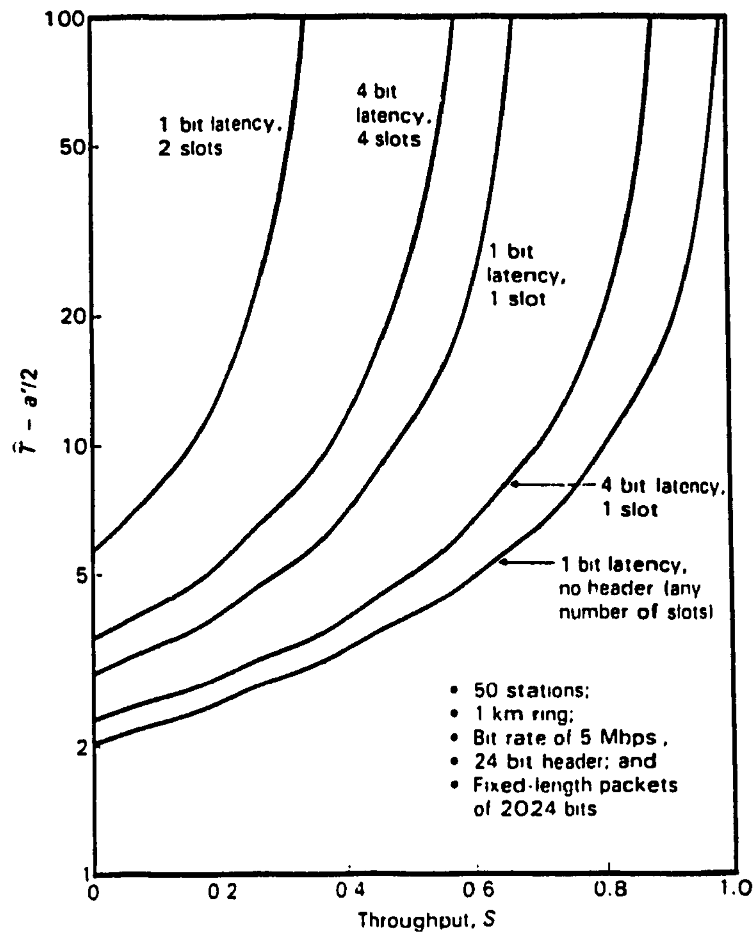



**Fig. 1.13.** **Normalized average transfer delay of a Cambridge ring with respect to its throughput.**

## 1.5 Register insertion rings.

The register insertion ring was first devised by Hafner, Nenadal and Tshanz in 1973 [96]. Its medium access principle is based on the buffering of the ring's traffic at a station to allow the transmission of its message with the minimum access delay. This is achieved by using a variable length shift register called the *insertion buffer* , in series in the ring, as shown in Fig. 1.14. The insertion buffer effectively increases or decreases the ring latency and hence there is no guarantee on the message delivery time. Whenever a station has a packet to transmit, it is assembled in a *transmit buffer* which can be connected to the ring at the proper instant. When transmission is proceeding, incoming bits are buffered in the insertion register and are inserted

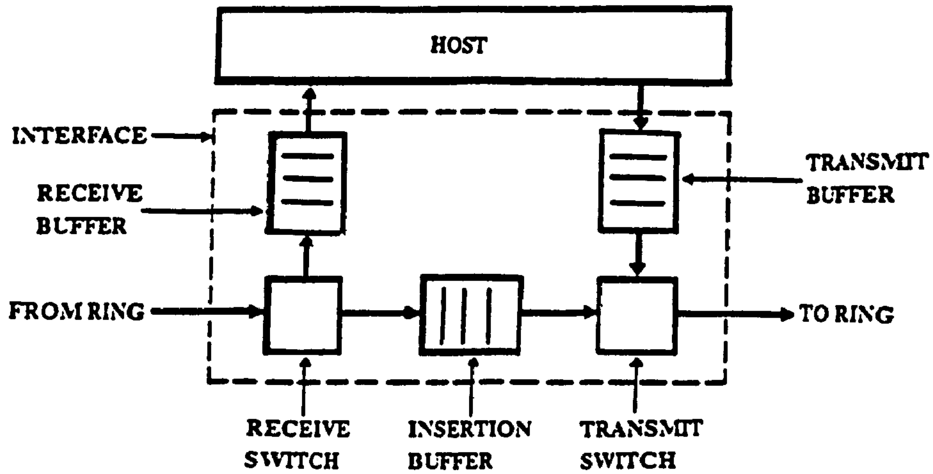



**Fig. 1.14. Buffers of the register insertion ring interface.**

back in the ring when transmission is complete. The instant at which the transmit buffer is connected to the ring depends on the priority scheme of the medium access protocol. In the *ring priority* scheme, packets on the ring will always have priority over any packets in the transmit buffer. On the other hand, in the *station priority* scheme, the packets in the transmit buffer have priority over those already on the ring. Either way, access to the ring is usually faster and more than one message can be circulating simultaneously around the ring. Message removal is performed by the receiving station.

In the following sections, the Distributed Loop Computer Network (DLCN) [44] will be described to illustrate the functioning of a register insertion ring. Other implementations of the register insertion rings are the DDLCN [45] and SILK [97].

## 1.5.1 Distributed Loop Computer Network (DLCN).

The DLCN was developed at the Ohio State University by Reames and Lui in 1975 [44]. It uses a variable length frame format as shown in Fig. 1.15. Details of the various field of the frame is shown in Table 1.7 [98].

| 8 bits | 8 bits | 8 bits | 16 bits | variable | 16 bits |
|---|---|---|---|---|---|
| **FLAG** | **DA** | **SA** | **CNTRL** | **DATA** | **FCS** |

**Fig. 1.15. Frame format of a DLCN packet.**

**Table 1.7. Description of the fields in the DLCN frame.**

| *Field* | *Contents and Description* |
|---|---|
| **FLAG** | 8 bit reserved sequence indicating the start or end of a frame. Bit stuffing is used if sequence occurs elsewhere in data. |
| **Destination Address (DA)** | 8 bit address indicating the loop interface to which the frame is intended for. 8 bit process number identifying the destination process. |
| **Source Address (SA)** | 8 bit address indicating the loop interface from which the frame originates. 8 bit process number identifying the source process. |
| **Control (CNTRL)** | 2 bits indicating the message type : (00) - information; (01) - acknowledgement; (10) - control; (11) - diagnostics.<br>1 bit indicating broadcast mode.<br>3 bits indicating function or response depending on the message type.<br>2 bits for message lost detection.<br>1 bit for lockout prevention.<br>7 bits for various control depending on the message type. |
| **Data** | 0 or more octets containing data or information. |
| **Frame Check Sequence (FCS)** | 16 bit Cyclic Redundancy Check (CRC) sequence covering the DA, SA, CNTRL and DATA fields. |

### 1.5.1.1 DLCN ring protocol.

A more detailed schematic of the station interface is shown in Fig. 1.16. Assuming that all buffers are empty, the DLCN ring operates as follows. When the station does not have any message to transmit, switches $S_1$ and $S_2$ are closed to position A and C respectively. As a packet passes by, $S_1$ checks if the packet is addressed to the station. If yes, $S_1$ is closed to position B and the packet is stored in the receive buffer. Hence the DLCN uses destination packet removal as opposed to source removal as seen in the previous rings. After the packet has been received, $S_1$ is reset to its original position. The host can read the receive buffer and signals $S_1$ thereafter to indicate that it is ready to receive more packets. When the station has a packet to transmit, it is assembled and stored in the transmit buffer. As the interface detects a inter- packet gap (an empty or idle bit), $S_2$ is toggled to position D such that the transmit buffer is connected to the ring and the packet can be sent. $S_2$ is toggled back to C when transmission is over and hence the ring latency has increased by the inserted packet. The insertion buffer empties itself slowly as it receives idle bits by shifting its buffer pointer to the right. If during ring operation there is less empty spaces in the insertion buffer than bits in the transmit buffer, no transmission is allowed as incoming bits on the ring may be lost as the insertion buffer will be full before the transmission is over. Hence the DLCN uses a mixture of station and ring priorities depending on the load on the network.

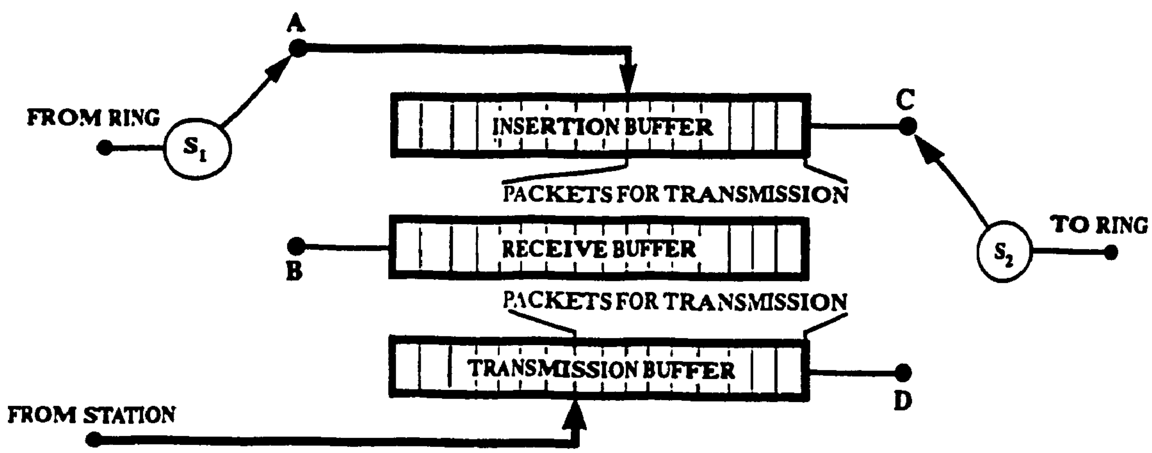



Fig. 1.16. Detailed schematic of station interface.

An enhanced version of the DLCN has been developed. The Double Distributed Loop Computer Network (DDLCN) uses a double counter rotating ring carrying information in opposite directions for enhanced reliability and performance [45].

### 1.5.1.2 Performance analysis of Register Insertion rings.

There has been numerous attempts to analyze the performance of register insertion rings [99-103]. However none could yield an exact solution because of the difficulties involved in the modeling of the interaction of the queues of the transmit, receive and insertion buffers of a station interface, and also the interaction of one station with another in the ring. In an effort to achieve mathematical tractability, they all attempted to model the ring as a collection of submodels each representing a station interface, and use the Kleinrock's independence assumption [104, p.321] to analyze each submodel separately. This independence assumption is clearly untrue as the interarrival time of messages to a given station depends on the service time of its upstream neighbor and its own message arrivals. However, refinements to the submodel yields close approximations of the simulated performance under the same conditions. Other assumptions that were made are (1) ring symmetry, that is all stations have the same characteristics such as buffer size, ring latency, arrivals and service rates, (2) arrivals to all transmit buffers are Poisson distributed, (3) the transmission pattern is that of a balanced ring, that is the probability of transmitting to a particular station depends on the distance separating them, and (4) the buffer size is large such that no packets are blocked or lost at any intermediate station.

Lui [99] and Thomasian and Kanakia [100] used the independence assumption and broke the ring into independent M/M/1 queues which use a non-preemptive priority assignment for the transmit and insertion queues. The independence assumption requires that an independent choice of the message length is performed each time the message enters a new node [104, p.321-323]. Their results over estimated the delay performance because (1) the independence assumption is more appropriate for queues in series when there is a lot of queues to and from each node. The superposition of their characteristics will approach a Poisson distribution as proved by the simulation studies in networks by Kleinrock [104] and (2) an average non-zero waiting time is always counted at each node, which is not necessarily true especially at low loads. Hammond and Reilly [74] tried to account for this effect by using the virtual cut-through method as developed by Kermani and Kleinrock [107]. However their results still diverged from the simulated ones especially at high loads. Thus inter-station dependencies of the different queues cannot be ignored and have to be reflected at least partially in the submodels used.

To this effect, Schlatter and Bux [101] considered both the previous and the actual station to model the traffic to that particular station. Their analysis indeed yielded closer results to those obtained through simulation. Hilal, Chiu and Liu [103] used a superposition technique developed by Rubin [105,106] to model the traffic. Their results confirmed those obtained in [101] for low and medium loads, and give tighter bounds for high loads. In the next section, a summary of the analysis and results of the works in [101] will be presented.

The submodel used by Schlatter and Bux [101] to model the register insertion ring is shown in Fig. 1.17a.

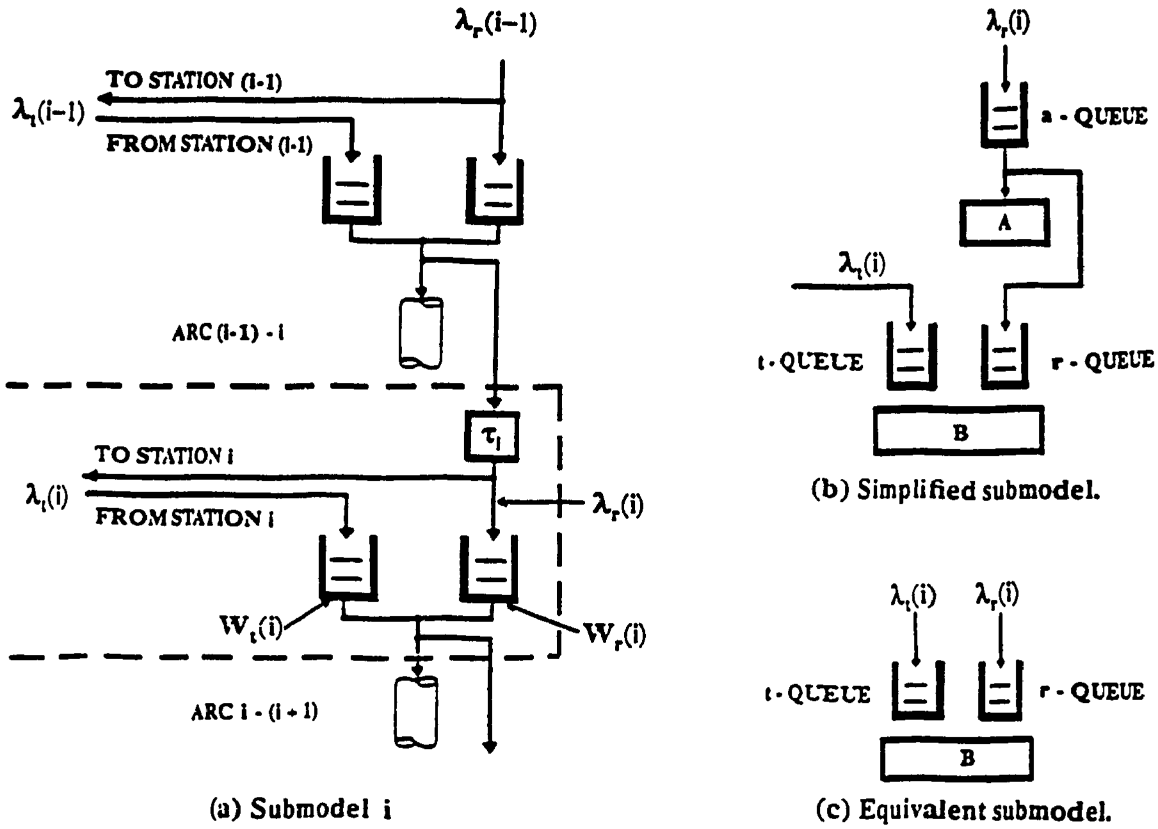



**Fig. 1.17. Submodel of register insertion ring with (a) the submodel, (b) the simplified submodel, and (c) the equivalent submodel [101].**

The submodel illustrates the fact that as soon as there is a transmission, that is service, at a station (i-1), there is an arrival at the latency box of station 'i'. The latency box represents the delay required for header recognition and propagation delay. The submodel is further refined by noting that (a) the characteristics of the random process do not change through a time shift and hence the latency boxes can be removed, and (b) the server at arc ((i-1)-i) sees the traffic from both the 'r' and 't' buffers and hence can be regarded as one buffer called the 'a'-buffer (see Fig. 1.17b). Noting that the $r^{th}$ buffer at the $i^{th}$ station sees the same traffic as the a-buffer, the submodel can finally be redrawn as Fig. 1.17c. The submodel can be recognized as a single M/G/1 model with non-preemptive priorities. Depending on the priority scheme used, the delay can be readily written. Noting that the delay $d_r(i)$ of a frame at the a-queue until its service begins at B is equal to the queuing delay of a frame in the r-queue, $d_r(i)$ at the receive buffer can be written as follows.

For ring priority,

$$d_r^{(r)}(i) = \frac{[\rho_t(i) + \rho_r(i)]\overline{T_p^2}}{2[1 - \rho_r(i)]\overline{T_p}}, \tag{1.23}$$

and for station priority,

$$d_r^{(s)}(i) = \frac{[\rho_t(i) + \rho_r(i)]\overline{T_p^2}}{2[1 - \rho_t(i) - \rho_r(i)][1 - \rho_t(i)]\overline{T_p}}, \tag{1.24}$$

where $\rho_t(i) = \lambda_t(i)\overline{T_p}$, $\rho_r(i) = \lambda_r(i)\overline{T_p}$ and $\overline{T_p}$ and $\overline{T_p^2}$ are the first and second moments of the frame transmission time respectively.

The mean delay of the a-queue is given as

$$w_a(i) = \frac{\rho_r(i)\overline{T_p^2}}{2[1 - \rho_r(i)]\overline{T_p}}. \tag{1.25}$$

Hence for the insertion buffer the delay is given as

$$\begin{aligned} w_r^{(r)}(i) &= d_r^{(r)}(i) - w_a(i) \\ &= \frac{\rho_t(i)\overline{T_p^2}}{2[1 - \rho_r(i)]\overline{T_p}}. \end{aligned} \tag{1.26}$$

Similarly for station priority,

$$\begin{aligned} w_r^{(s)}(i) &= d_r^{(s)}(i) - w_a(i) \\ &= \frac{\rho_t(i)[1+\rho_r(i)(1 - \rho_t(i) - \rho_r(i))]\overline{T_p^2}}{2[1 - \rho_t(i) - \rho_r(i)][1 - \rho_t(i)][1-\rho_r(i)]}. \end{aligned} \tag{1.27}$$

The mean delay for the transmit buffer can be similarly derived for both ring and station priorities.

$$w_t^{(r)}(i) = \frac{[\rho_r(i) + \rho_t(i)]\overline{T_p^2}}{[1+\rho_r(i)(1-\rho_t(i)-\rho_r(i))]\overline{T_p^2}}. \tag{1.28}$$

$$w_t^{(s)}(i) = \frac{[\rho_r(i) + \rho_t(i)]\overline{T_p^2}}{2[1-\rho_t(i)]\overline{T_p}}. \tag{1.29}$$

By calculating the mean delay of each individual component, the mean transfer delay can be obtained by summing the average waiting times of the transmit and insertion buffers with the average frame transmission time. Hence, the mean transfer time can be written as

$$t_f = \frac{\sum_{j=1}^{s} \Omega(\rho_r(i) + \rho_t(i)) - \Omega(\rho_r(i))}{\sum_{i=1}^{s} (\lambda_r(i)+\lambda_t(i))}. \tag{1.30}$$

where

$$\Omega(\rho) = \frac{\rho^2\overline{T_p^2}}{2(1-\rho)(\overline{T_p})^2}. \tag{1.31}$$

Equation (1.30) is valid regardless of which priority schemes is applied (see Fig. 1.18). This is expected as the system is work conserving and the selection of a customer by the queuing discipline is independent of its service time [104, p.113]. A comparison of the mean transfer delay obtained by Schlatter and Bux [101] with first order approximations such as in [99] and [100] are shown in Fig. 1.18.

The shortcoming in the analysis of [101] is that it ignores the departures of the messages which reach their destination. Thus their results deviate from the ones obtained through simulation as soon as they exceed 10% of the original traffic. This was pointed out by Hilal *et al* in [103] where they tried to correct this shortcoming by performing separate analysis for traffic exceeding $\rho = 0.5$.

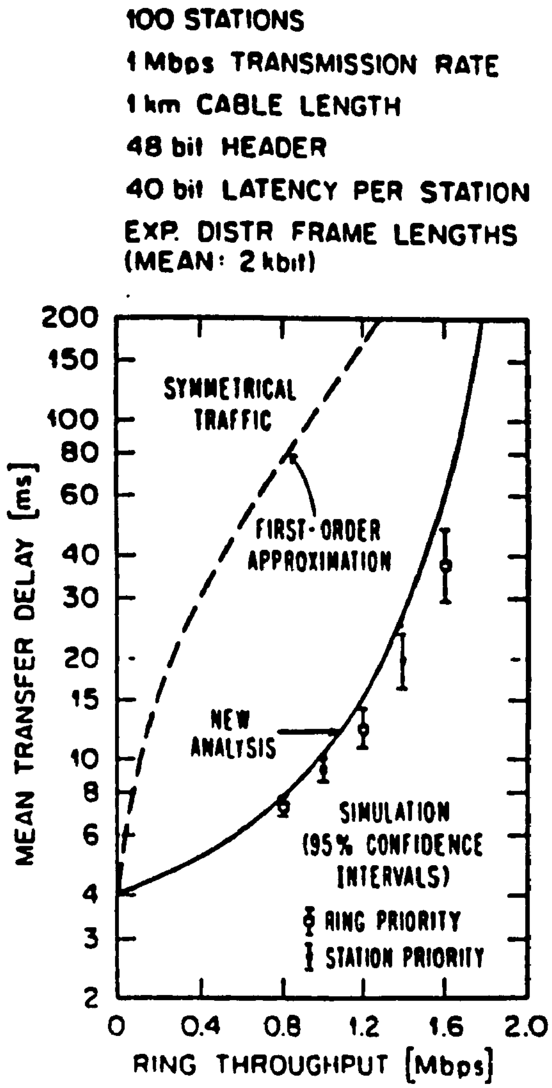



**Fig. 1.18. Mean transfer characteristics of station and ring priorities and comparison with first order aproximation [101].**

## 1.6 Performance comparison of the different rings.

There has been numerous attempts to compare the performance of different rings [18,37,74,108]. The reason is obvious. It provides the ability to choose the most appropriate type of ring given some traffic characteristics and performance requirements. The inherent difficulty in such comparisons is to find common parameters such that the different rings are fairly represented for the comparison to be valid. Nevertheless, the various comparisons lead to similar conclusions. Thus in this section, the comparison in [74] will be discussed.

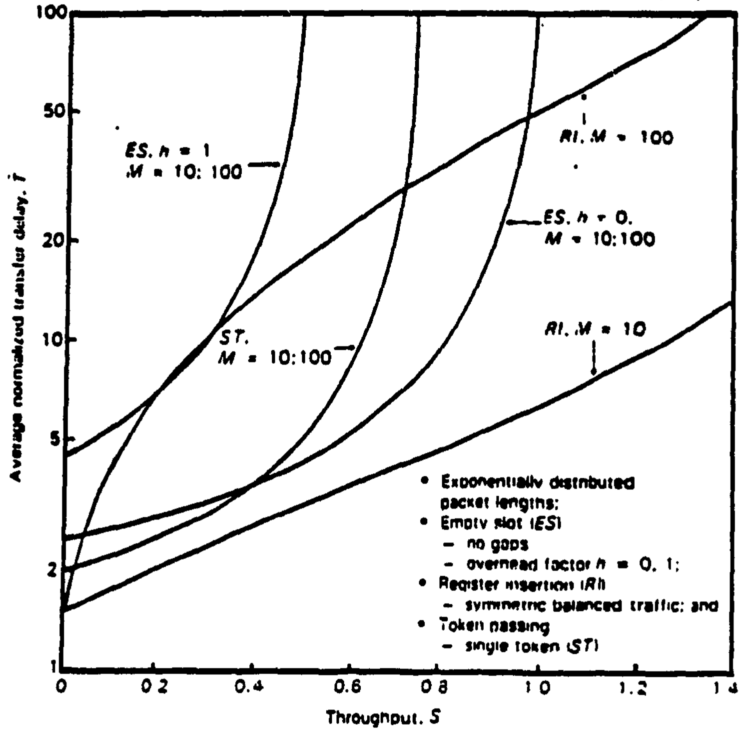
The single-token ring (ST), the slotted ring with no gaps (ES) and the register insertion ring (RI) with symmetric, balanced traffic characteristics will be compared. The slotted ring can be considered as having no gaps as the gaps can be included in the slots and considered part of the header. The following assumptions are made for all three rings.

1. There are identical Poisson arrivals to each station,
2. Consecutive stations are equally spaced, and
3. The packet lengths are exponentially distributed.

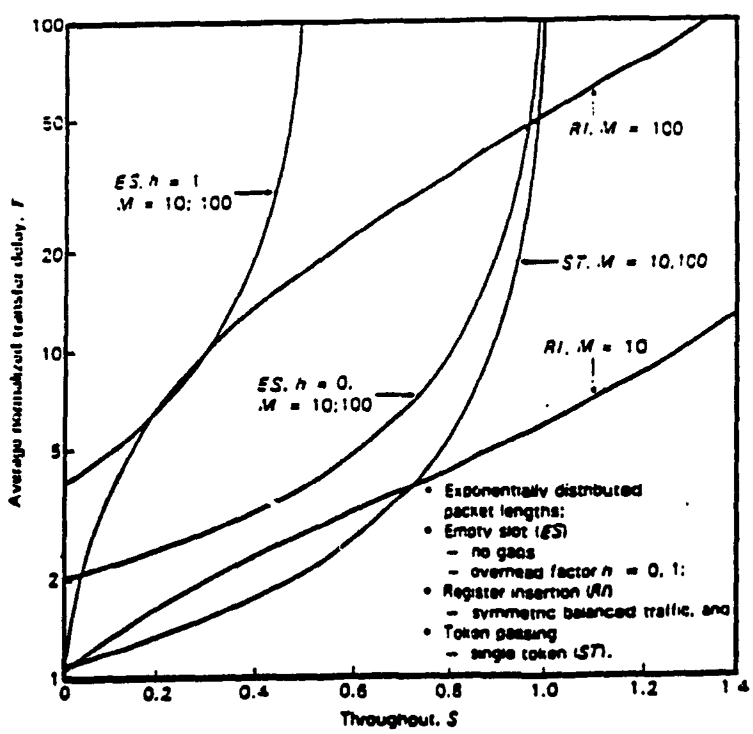
The performance of the three different rings with a normalized ring latency of $a = 1$ is shown in Fig. 1.19a and that for $a = 0.1$, in Fig. 1.19b. From these figures, it can be observed that for a ring latency of 1 and a 10 station ring, the register insertion ring gives the best performance while for a 100 station ring, it yields the worst performance. At low loads, the token ring is better while at high loads the slotted ring prevails, provided its overhead is low ($h << 1$). For low ring latency ($a = 0.1$) and a 10 station ring, the token ring is best at low loads while the register insertion ring is still better at high loads. However for a 100 station ring, the token ring is still the best.

Generally, it can be said that the register insertion ring is the best when the number of stations is small or when the load is very low for a given number of users. This is due to the almost immediate access to the ring due to the register insertion medium access protocol and at low loads, the ring is not excessively extended to introduce prohibitive packet delays. On the other hand, the token ring is good under almost all conditions. It only requires that the station latencies are short so that the maximum throughput can be achieved with minimum delay. The slotted ring is good only when there is a large number of stations and where the propagation delay is large. However the choice of the size of the minipacket should minimize the overhead as only a low overhead factor will yield adequate performance.

Lastly, one possible shortcoming of the comparison in [74] is that the same normalized ring latency has been used for all the rings during the analysis. This is not necessarily true as for example the register insertion ring needs a 7 bit station latency to allow address recognition, whereas the token ring requires only a single bit station latency. Nevertheless, the conclusions obtained seem to be reasonable and agree with the other comparisons in [73,103].




(a)




(b)

**Fig. 1.19. Comparative performance of the single token ring (ST), the slotted ring (ES) and the register insertion ring (RI) for 10 and 100 stations with a normalized latency in (a) of $a = 1$, and in (b) of $a = 0.1$ [74].**

# Chapter Two

# Design of a bandwidth reducing adapter.

## 2.1 Introduction.

Apart from performance considerations, another important factor has to be considered during the design of a LAN. It is the cost to link various stations in a network. For example, the cost of transceivers in a planned optical network can be too prohibitive to command an entirely fiber optics connectivity. Thus for office networks, the use of cheap unshielded twisted pair (telephone wires) is a common practice, even for integrated services while keeping a fiber optic backbone. In fact, twisted pair is considered as the transmission medium by the IEEE 802.9 work group which is drafting the Integrated Voice and Data Networks Standard to connect desktop devices to 802 LAN's and Integrated Service Digital Networks (ISDN's) [1, pp. 47-48].

Two factors hinder the use of unshielded twisted pair for inter-station connections. First, unshielded twisted pair is very prone to Electro-Magnetic Interference (EMI) and impulse noise, and secondly, the proliferation of digital systems operating at 1 to 4 Mbps has led to an increasing competition for bandwidth around the 1 MHz region. Thus to minimize these factors and reduce inter-system crosstalks, a 16 Mbps bit rate has been proposed [109]. It has the advantage of avoiding the congested bandwidth region and also of accommodating the increasing needs and new services evolving in office technology. On the other hand, the use of a 16 Mbps rate on twisted pair greatly reduces the reach of the links as more bandwidth is required.

A 16 Mbps token ring using a Manchester or a differential Manchester line code requires a 16 MHz bandwidth. Thus since only the 1 MHz region has to be avoided, the use of any signaling scheme that would require less bandwidth is acceptable or even welcomed as it will extend the reach of the links. Hence this chapter presents the design of an adapter that will reduce by two the required bandwidth for a 16 Mbps token ring. This reduction is possible through the use of a modified duobinary signaling scheme that would need only half of the required bandwidth. Its general theory is given in the next section, followed by a detailed description of its implementation in the adapter.

## 2.2 Modified duobinary (MD) signaling.

Modified duobinary (MD) signaling is a partial response signal of class 4 (PR4) and uses controlled intersymbol interference (ISI) to be able to transmit at the Nyquist rate of $2W$ symbols per second, given a bandwidth of W hertz. The Nyquist rate is the maximum achievable rate for a fixed given bandwidth without any ISI [110]. For ordinary pulse amplitude modulation (PAM) bandlimited signaling, the Nyquist rate of $2W$ symbols/s can only be achieved by using an ideal rectangular filter having the following frequency response

$$G(f) = \begin{cases} T & |f| < 2W, \\ 0 & |f| > 2W. \end{cases} \tag{2.1}$$

Hence normally, physically realizable filters need an excess bandwidth ranging from 15 to 100 percent to achieve Nyquist rate, depending on the shape of the signal used [111]. In 1963, A. Lender pointed out that Nyquist rate can be achieved without requiring any excess bandwidth if some correlation between consecutive signals was allowed instead of considering totally independent signals as before [112]. This correlation introduced some controlled ISI which can be accounted for at the receiving end. Thus was born a new set of bandwidth efficient signaling schemes called partial response or correlative signaling schemes and embraced both the duobinary and modified duobinary signaling. These schemes were generalized by Kretzmer through the use of a transversal linear filter and proper coefficients for its taps [113].

MD signaling can be easily be achieved by combining two binary pulses, one of which having been delayed by 2T seconds, where T is the period of the transmission rate R [114].

$$y_n = x_n - x_{n-2}, \tag{2.2}$$

where $x_n$ and $x_{n-2}$ are the binary pulses at t = nT and at t = (n-2)T respectively, and 'n' is an integer. As $x_n$ can assume two values $\pm 1$, three signaling levels are obtained, namely 0 and $\pm 2$, corresponding to the bits '0' and '1' respectively. The decoding of the MD signal can thus be achieved by using the following decoding rule:

$$x_n = \begin{cases} 0 & |y_n| < 1, \\ 1 & |y_n| \geq 1. \end{cases} \tag{2.3}$$

The shape of the signal is shown in Fig. 2.1. The correlation spans over three bits and yields a signal having zeros at any nT interval except at $t = \pm T$. Its frequency response can be written as

$$\begin{aligned} G(f) &= T\,(1 - e^{-j4\pi fT}) = T\,(e^{+j2\pi fT} - e^{-j2\pi fT})\,e^{-j2\pi fT} \\ &= j2T\,e^{-j2\pi fT}\sin 2\pi fT. \end{aligned} \tag{2.4}$$

Hence,

$$\begin{aligned} |G(f)| &= 2T\sin 2\pi fT \\ &= \frac{1}{W}\sin\frac{\pi f}{W}, \quad where\ W = \frac{1}{2T}. \end{aligned} \tag{2.5}$$

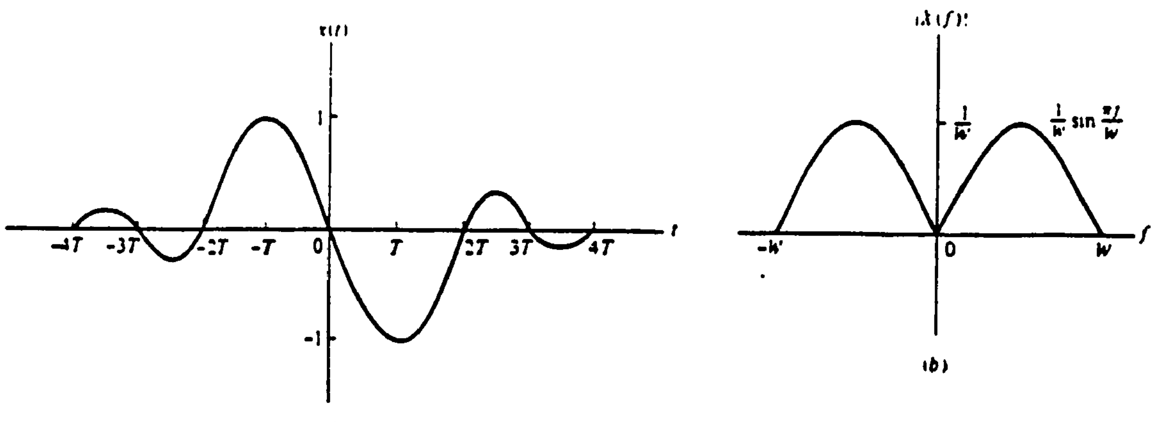



**Fig. 2.1. Modified duobinary signal in frequency and time domain [114].**

Hence the Nyquist rate can be achieved and no d.c. components exist. This contrasts with the duobinary signaling scheme where Nyquist rate is achieved by delaying one of the binary pulse by only T seconds, but a large d.c. component exists. D.C. components are not very much desirable especially if the circuit is transformer coupled.

## 2.3 Design guidelines and general operation.

The design of the adapter assumes that the token ring operates at 16 Mbps using a Manchester or differential Manchester code. This assumption is justified as the IEEE 802.5 standard for token rings stipulates a 1 or 4 Mbps bit rate using differential Manchester encoding [13].

Because of the nature of the application of the adapter, its unit cost is very important. Hence the design has been implemented in TTL technology, using the IC's of the Advanced Schottky (AS) family. Not only do they provide the proper performance (an acceptable propagation delay of approximately 5-10 ns.), they also have a reasonable power supply demand (+5 V), and are cheap and readily available. Moreover delays in the design are implemented using IC's from another family, for example, the LS series which have a longer propagation delay. Basic precautions used in high frequency design such as the use of a ground plane, short wire connections, and high tolerance components have been observed during the wire-wrapping of the prototype.

The adapter consists of two parts namely the transmitter and the receiver. The transmitter fits at the output of the transmitting station while the receiver fits at the input of the receiving station as shown in Fig. 2.2. The transmitter converts the outgoing differential Manchester code into a modified duobinary signal, thus achieving a bandwidth reduction by a factor of two. At the receiving station, the receiver converts the modified duobinary signal back to the original differential Manchester signal.

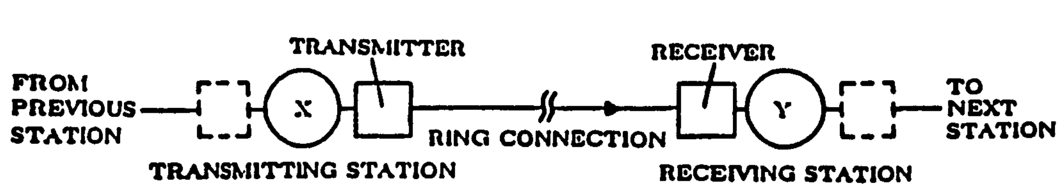



**Fig. 2.2. Location of adapter on the ring.**

Hence the whole process is transparent to both the transmitting and receiving station, and does not affect the operation of other stations which may be present on the ring. The following sections will discuss the design of the transmitter and receiver in greater details.

## 2.4 The Transmitter.

The block diagram of the transmitter is shown in Fig. 2.3. The transmitter performs the following functions as identified by each block.

1. It converts the differential Manchester signal into a TTL Manchester signal (comparator). The TTL Manchester signal is a version of the original differential Manchester signal where the negative part has been truncated.

2. It recovers the imbedded clock from the TTL Manchester signal (zero crossing detector and clock recoverer).

3. It interprets the data sent in the differential Manchester signal from the TTL Manchester signal (data recoverer).

4. It converts the recovered data into a modified duobinary signal (precoder, duobinary generator, and low-pass filter).

5. It sends the modified duobinary signal to the line (line driver).

The design of each block will be covered in the following sections.

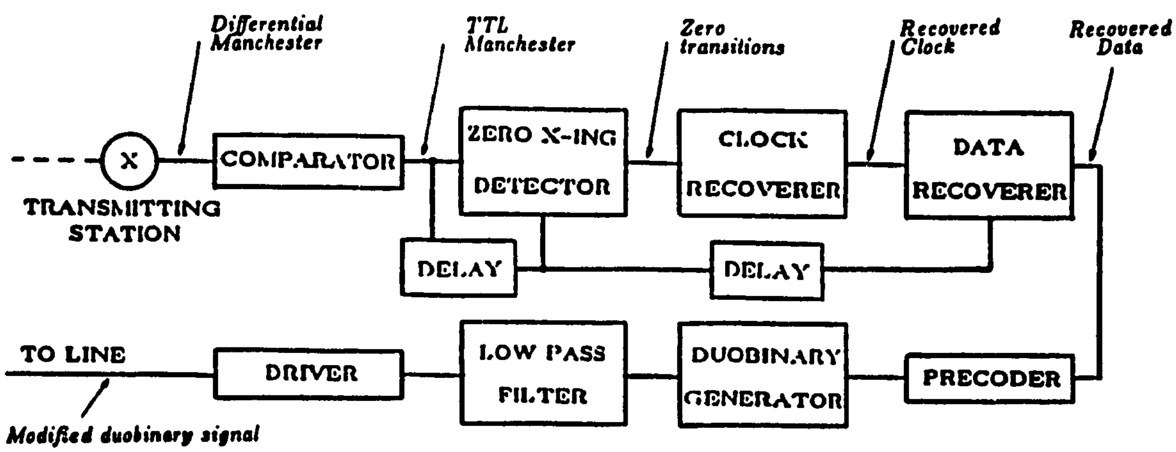



**Fig. 2.3. Block diagram of the transmitter.**

## 2.4.1 Comparator.

The comparator rectifies the incoming differential Manchester signal into a TTL signal by truncating all the negative portion of the incoming signal. A high speed comparator, the AD9686 from Analog Devices Ltd., is used to perform this operation as shown in Fig. 2.4. A zero volt reference is used for the comparison threshold.

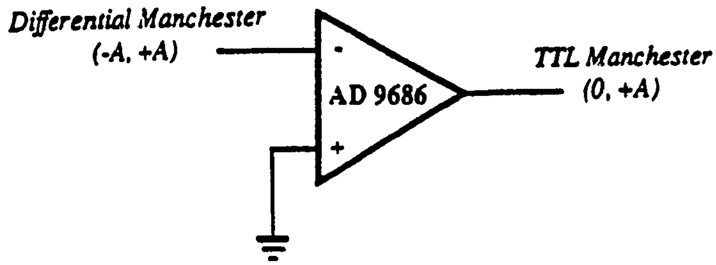



Fig. 2.4. Comparator.

## 2.4.2 Zero crossing detector and clock recoverer.

The design of the zero-crossing detector and clock recoverer is shown in Fig. 2.5. As both the Manchester and the differential Manchester signals have compulsory zero-crossings at each mid-bit, the adapter does not discriminate between them and treats both as a Manchester signal. Hence from this point onwards no distinction will be made between either signal. The clock recoverer retrieves the clock imbedded in the TTL Manchester signal by observing the compulsory zero-crossings at each mid-bit of the Manchester signal. This can be easily done by the exclusive-OR of the TTL stream with its slightly delayed version (zero crossing detector), and feeding the output to a monostable. Fig. 2.6 shows the various traces of a sample binary stream to illustrate the clock recovery process as well as the data recovery. The delay used in the zero-crossing generator is less than T/2 while the delay in the monostable is slightly greater than T/2 but less than T, where T is the duration of a clock period. Because of the regularity of the zero-crossings at each mid-bit of the Manchester bit, the monostable will quickly latch on the good clock even if it started with the wrong sequence as during the initial phase of the clock recovery trace.

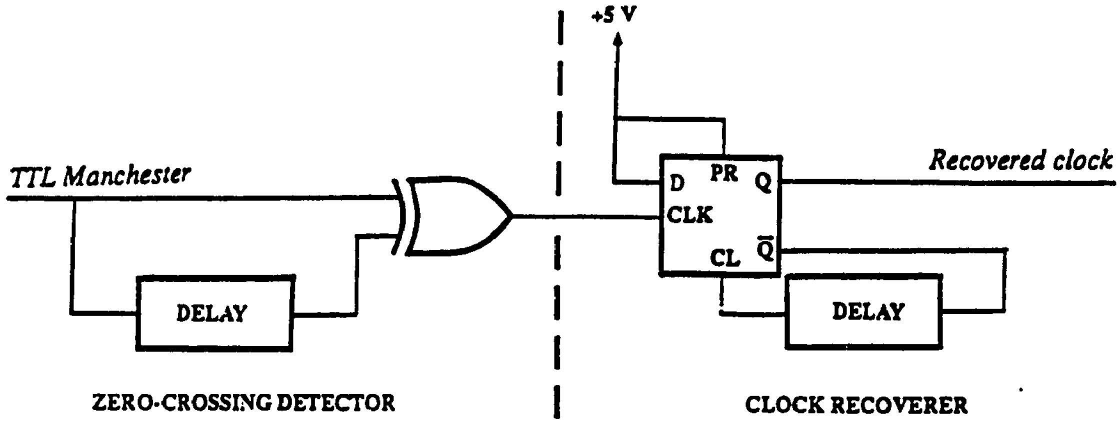



**Fig. 2.5. Zero-crossing detector and clock recoverer.**

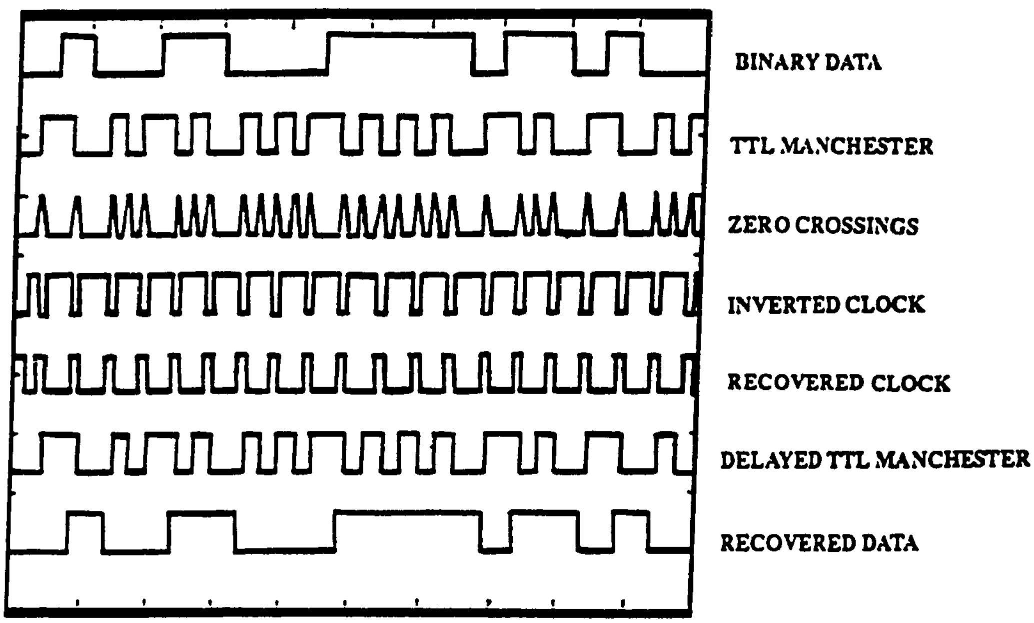



**Fig. 2.6. Clock and data recovery traces.**

### 2.4.3 Data recoverer.

The data recoverer uses the recovered clock to retrieve the data from the TTL Manchester signal, as shown in Fig. 2.6. Using a D flip-flop, the data can be easily recovered by latching on the corresponding half bit of the appropriately delayed incoming TTL Manchester signal. If the incoming signal has not been delayed properly, the inverted version of the data is recovered as the flip-flop would latch on the wrong half bit. The design of the data recoverer is shown in Fig. 2.7.

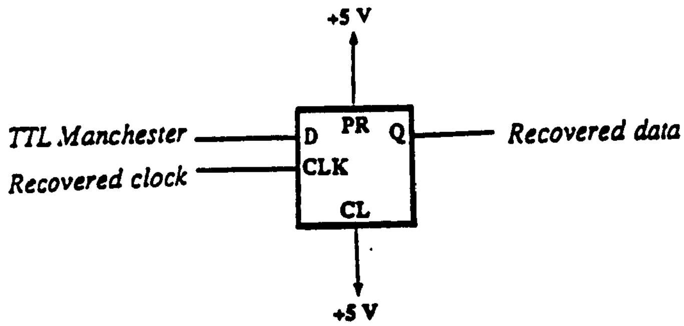



**Fig. 2.7. Data recoverer.**

## 2.4.4 Precoder.

The precoder combines the current stream of data with its delayed version such that error propagation is avoided when decoding the MD signal at the receiver. The precoding performed can be described by the following equation:

$$P_n = x_n + P_{n-2}, \tag{2.6}$$

where $P_n$ and $P_{n-2}$ are the $n^{th}$ and the $(n-2)^{th}$ precoded bit respectively, and $x_n$ is the current $n^{th}$ recovered data bit from the previous stage. The design of the precoder is shown in Fig. 2.8.

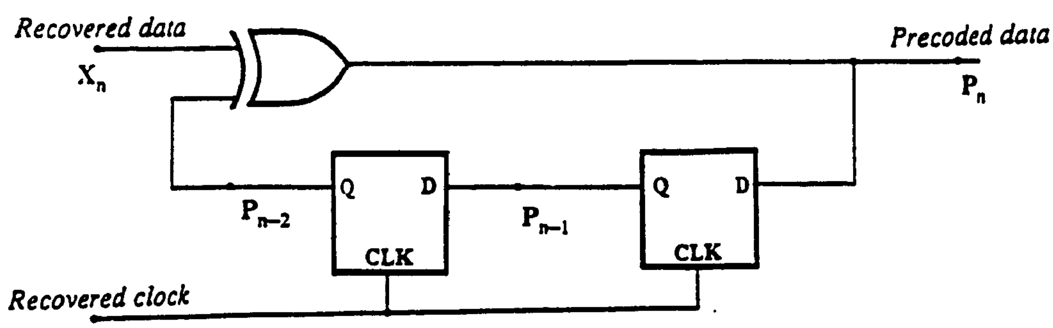



**Fig. 2.8. Precoder.**

## 2.4.5 Modified duobinary generator.

The modified duobinary generator implements the following equation as seen in section 2.3.

$$y_n = P_n - P_{n-2}. \tag{2.7}$$

Three signal levels are obtained, that is 0, ± 2. The equation has been implemented in a slightly different way by using a multiplexer to select the appropriate output level for a given set of inputs. The implementation can be justified by Table 2.1 below while the design of the generator is shown in Fig. 2.9.

**Table 2.1. Truth table of the modified duobinary generator.**

| *Bipolar levels* | | | *Logic levels* | | |
|---|---|---|---|---|---|
| $P_n$ | $P_{n-2}$ | $y_n$ | $P_n$ | $P_{n-2}$ | $y_n$ |
| -1 | -1 | 0 | 0 | 0 | 0 |
| -1 | 1 | -2 | 0 | +5 V | -5 V |
| 1 | -1 | +2 | +5 V | 0 | +5 V |
| 1 | 1 | 0 | +5 V | +5 V | 0 |

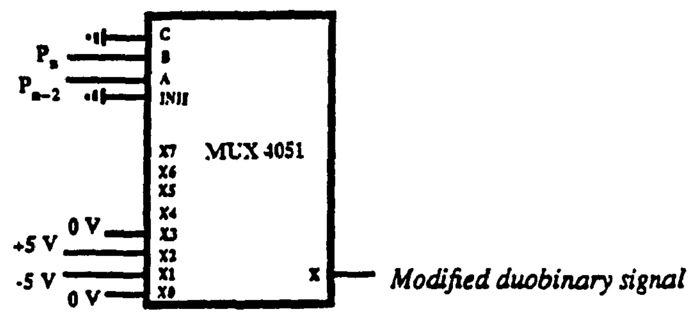



**Fig. 2.9. Modified duobinary generator.**

## 2.4.6 Low-pass filter.

A $4^{th}$ order passive Butterworth filter has been designed to remove the higher frequency components in the three level signal before sending it to the line. A $LC$ ladder filter has been realized using the standard design for a $4^{th}$ order Butterworth low-pass filter as given in [116], and scaling it properly to fit a cut-off frequency of 6 MHz (see Fig. 2.10). Hence using the respective frequency and magnitude scaling factors of $K_f = 6\times 10^6 \times 2\pi$ and $K_m = 1305$, the following values for the various components were obtained.

$$R_a = 1.305\,K\Omega \qquad L_1 = 1.36\,\mu H \qquad C_2 = 225\,pF$$
$$R_s = 0.130\,K\Omega \qquad L_3 = 5.6\,\mu H \qquad C_4 = 320\,pF$$

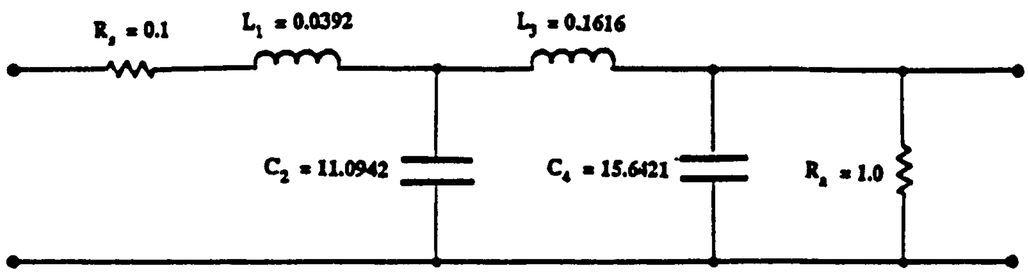



**Fig. 2.10. Standard low-pass filter.**

## 2.4.7 Line driver.

The signal is sent to the line by using coupling transformers as shown in Fig. 2.11. A turn ratio of 1:1.5 is used to boost the signal slightly at the transmitter. The signal can be easily recovered at the the receiver by using decoupling transformers. A center-tapped transformer is also used to provide both the non-inverted and the inverted version of the signal. The turn ratio at the receiver has been chosen such that the signal levels received are compatible to those of the TTL logic used thereafter.

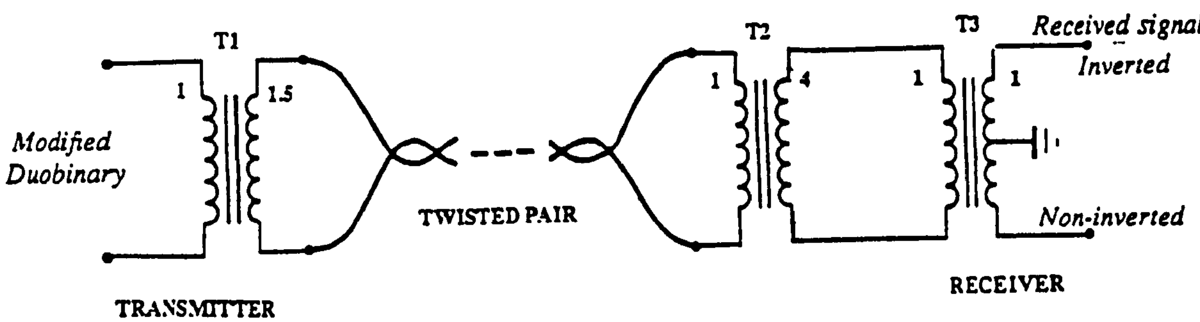



**Fig. 2.11. Line driver.**

## 2.5 The Receiver.

The block diagram of the receiver is shown in Fig. 2.12. The receiver performs the following functions as identified by each block.

1. It converts the three level MD signal into a TTL signal (TTL converter).

2. It recovers the data from the TTL signal by using an adaptive sampler (sampler and data decoder).

3. It converts the recovered data back into its differential Manchester signal (Manchester encoder).

4. It sends the differential Manchester signal to the receiving station (line driver).

The different blocks will be considered in greater details in the following sections.

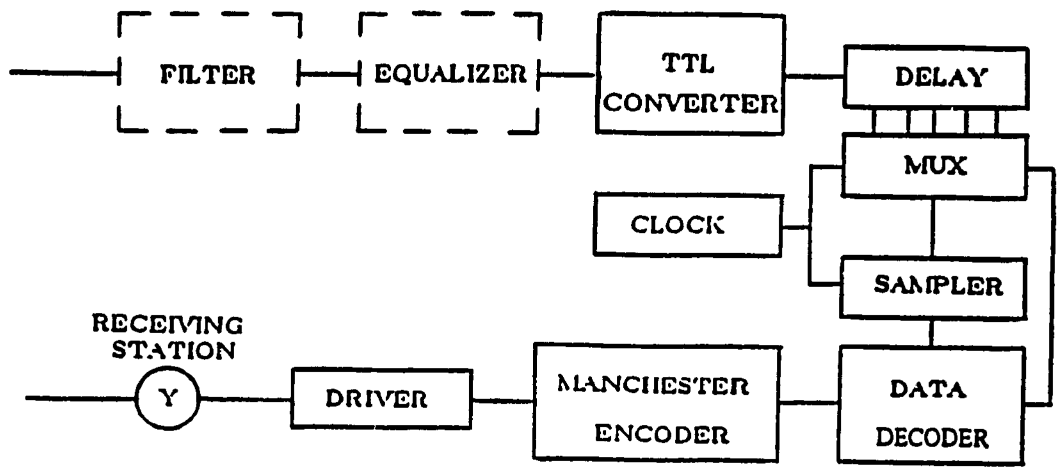



**Fig. 2.12. Block diagram of the receiver.**

## 2.5.1 TTL converter.

The TTL converter fully rectifies the MD signal such that a corresponding TTL signal is obtained. The non-inverted and inverted signal received are buffered from the following stage by using transistor followers (JFET) biased at 0 volts, as shown in Fig. 2.13. The signals are fed to exclusive-OR gates that act as threshold detectors and can be combined thereafter by using another exclusive-OR gate. The output signal is a binary signal that closely resembles the original data, except for some uneven bit widths as the 1's tend to shrink slightly during the conversion process. The data decoder will try to correct this discrepancy as it will be seen in the next sections.

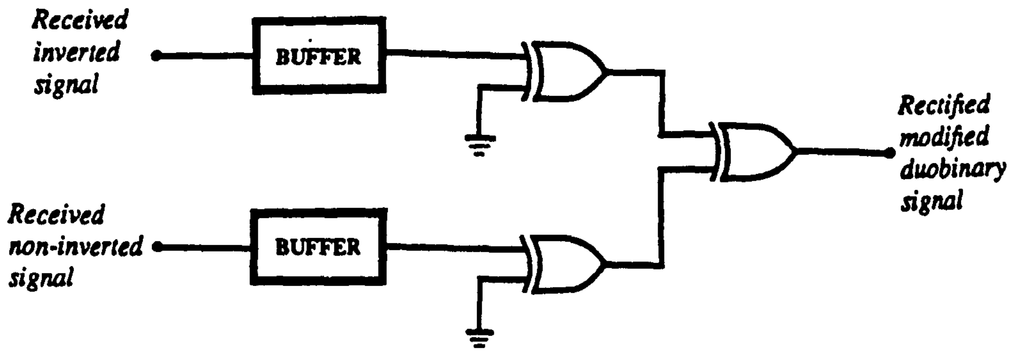



**Fig. 2.13. TTL converter.**

## 2.5.2 Adaptive sampler.

The adaptive sampler consists of a series of AND gates that will each delay the incoming TTL signal by a fraction of the bit period T as shown in Fig. 2.14. A multiplexer chooses by how much should the incoming TTL signal be delayed such that all the samples taken by the adaptive sampler all come from within the same bit period. A local clock is available but it does not have any synchronization with the transmitting station. The multiplexer is controlled by the signals from the data decoder such that no data is lost during any delay adjustment. Hence two signals from the data decoder block is available. One is the Enable signal (ENB) to enable the multiplexer and the other is the $U/\overline{D}$ signal which will tell the multiplexer increment or decrement the delay by a fraction of a bit period. The delays used are 0, 15, 30, and 45 ns, which roughly corresponds to a delay of 0, T/4, T/2 and 3T/4 bits for a rate of 16 Mbps.

The delayed signal from the multiplexer is sampled five times within a bit period. These samples are used to identify the bit transmitted and also to decide if any adjustment in the delay chosen is required. Delay adjustment is required if the samples taken indicate that they were from two adjacent bit periods rather than within the same bit period. A series of D flip-flops with their latching clocks each delayed by roughly nT/5 bits, is used to perform the sampling as shown in Fig. 2.15. The actual delays used are 0, 15, 30, 42, and 54 ns. Another set of D flip-flops is used to stabilize the samples taken ($S_0, S_1, \cdots, S_4$) such that they can all be latched at the same time to the data decoder to avoid any unstability.

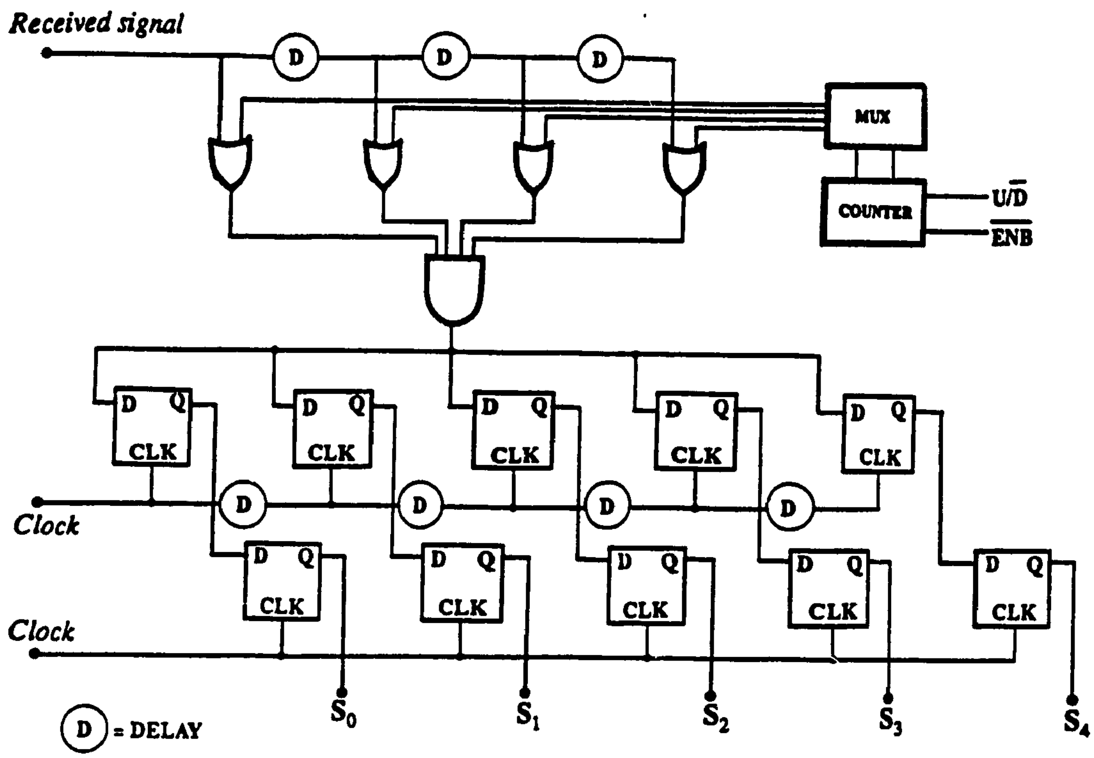



Fig. 2.14. Adaptive sampler.

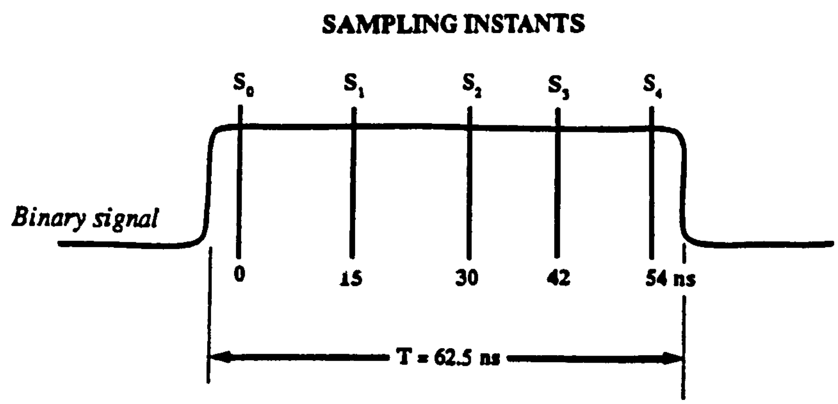



Fig. 2.15. Bit sampling by adaptive sampler.

## 2.5.3 Data decoder.

The data decoder is basically a combinational logic circuit with a feedback to the adaptive sampler. The combinational circuit implements a slightly biased truth table to decide which data bit has been received and whether any delay adjustments has to be made. The bias is in the decoding of the bit, for a '1' is slightly favored rather than a '0'. This bias can be justified as the conversion to a MD signal tends to shrink the bits slightly and hence during the recovery process, the bit period for a '1' is slightly smaller than the '0'. The truth table for the combinational circuit is given by Table 2.2. It is minimized using Karnaugh maps and the following minimized expression of the data, the enable signal for the multiplexer ($\overline{ENB}$) and that for up/down counter ($U/\overline{D}$) have been obtained. Their implementation is shown in Fig. 2.16.

$$DATA = s_2 + \bar{s}_0 s_1 + s_3 \bar{s}_4 + s_1 s_3 \tag{2.8}$$

$$\overline{ENB} = \bar{s}_0 \bar{s}_1 s_3 s_4 + s_1 s_2 \bar{s}_3 \bar{s}_4 \tag{2.9}$$

$$U/\overline{D} = s_0 s_2 + \bar{s}_0 \bar{s}_2 \tag{2.10}$$

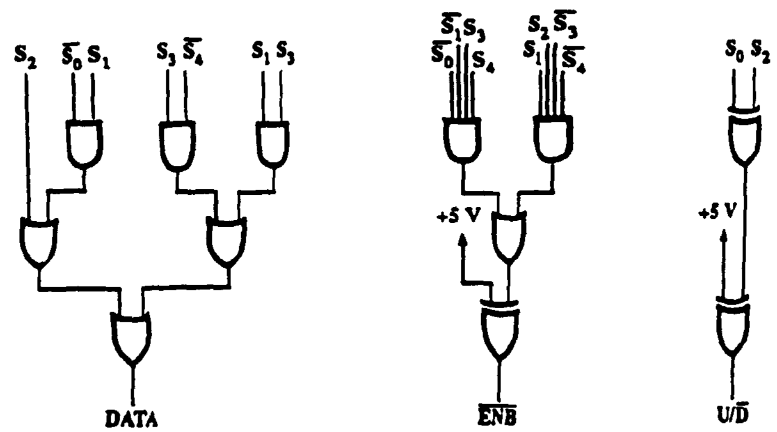



**Fig. 2.16. Data decoder.**

It can be observed that adjustment in the delay is only perfomed when it is certain that the incoming signal is not properly synchronized with the sampler. However no bits are lost as a suitable decision is still made even if any adjustment is needed. The delay of the incoming signal is verified for every two bits decoded. It allows more stability in the data decoder and avoids any races due to the feedback. Fig. 2.17 clarifies the timing sequences in the data decoder. First, the clock enables the D flip-flops to fetch a new set of samples $S_0, S_1, \cdots, S_4$. After about 18 ns to allow for propagation through the combination circuit, the $DATA$, $ENB$ and $U/\overline{D}$ signals are ready. The data is read every 30 ns after the sample fetch. The $ENB$ and $U/\overline{D}$ signals on the other hand, are only read every second cycles at the delay check. They are valid after about 30 ns to allow for propagation through the multiplexer and Up/Down counter, until the next delay check signal.

**Table 2.2. Truth table of combinational circuit in data decoder.**

| $S_0S_1S_2S_3S_4$ | $U/\overline{D}$ | $\overline{ENB}$ | $DATA$ |
|---|---|---|---|
| 00000 | x | 1 | 0 |
| 00001 | x | 1 | 0 |
| 00010 | x | 1 | 1 |
| 00011 | 1 | 0 | 0 |
| 00100 | x | 1 | 1 |
| 00101 | x | 1 | 1 |
| 00110 | x | 1 | 1 |
| 00111 | 0 | 0 | 1 |
| 01000 | x | 1 | 1 |
| 01001 | x | 1 | 1 |
| 01010 | x | 1 | 1 |
| 01011 | x | 1 | 1 |
| 01100 | x | 1 | 1 |
| 01101 | x | 1 | 1 |
| 01110 | x | 1 | 1 |
| 01111 | x | 1 | 1 |
| 10000 | x | 1 | 0 |
| 10001 | x | 1 | 0 |
| 10010 | x | 1 | 1 |
| 10011 | x | 1 | 0 |
| 10100 | x | 1 | 1 |
| 10101 | x | 1 | 1 |
| 10110 | x | 1 | 1 |
| 10111 | x | 1 | 1 |
| 11000 | 0 | 0 | 0 |
| 11001 | x | 1 | 0 |
| 11010 | x | 1 | 1 |
| 11011 | x | 1 | 1 |
| 11100 | 1 | 0 | 1 |
| 11101 | x | 1 | 1 |
| 11110 | x | 1 | 1 |
| 11111 | x | 1 | 1 |

$x$ - don't care bit
$U/\overline{D}$ - increase or decrease counter
$\overline{ENB}$ - enable counter

## 2.5.4 Manchester encoder.

The Manchester encoder converts the recovered data into the Manchester code before sending it to the receiving station. A D flip-flop is used to provide a stable output so that the exclusive-OR of the recovered data with a 16 Mhz clock and the recovered data yields the Man-

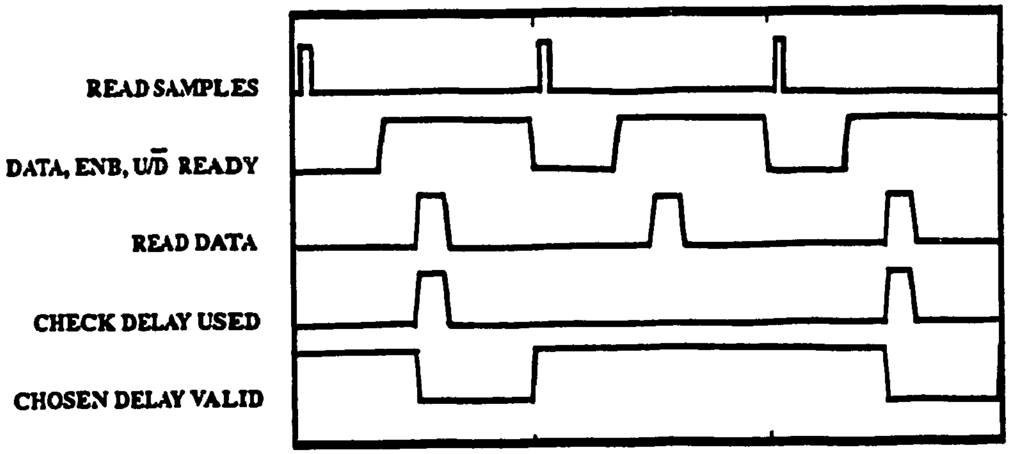



**Fig. 2.17. Timing schematic of the data decoder operations.**

chester signal as shown in Fig. 2.18. The signal is then level shifted so that the normal $\pm A$ levels are obtained.

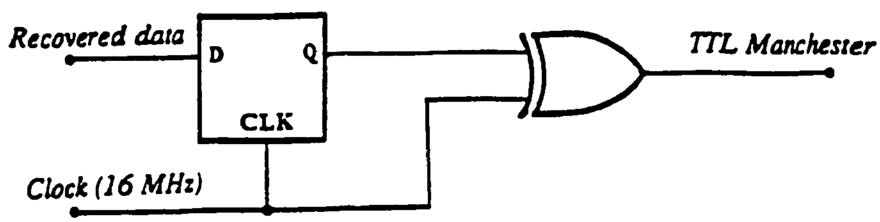



**Fig. 2.18. Manchester encoder.**

## 2.6 Overall design and results.

The overall design of the transmitter and receiver is shown in Fig. 2.19 and Fig. 2.20 respectively.

Fig. 2.22 to 2.24 show the various waveforms obtained with the prototype transmitter. Fig. 2.22 shows the trace of the Manchester signal received by the transmitter from the transmitting station and its zero-crossing pulses obtained after the zero-crossing detector but before being fed to the monostable of the clock recoverer. The output of the monostable ($Q$ and $\bar{Q}$) are shown in Fig. 2.23. Either one can be used as the recovered clock to recover the data from the incoming

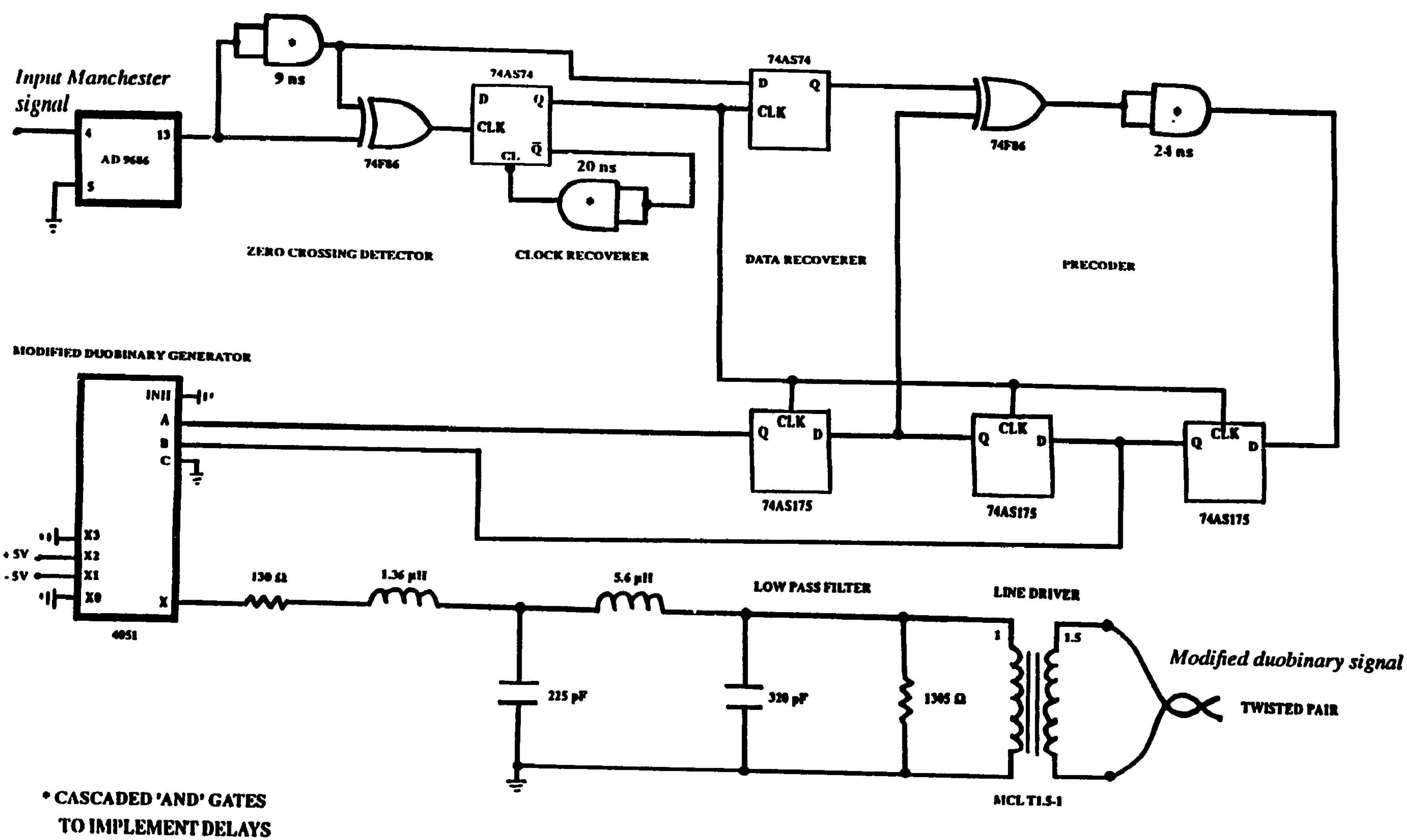



Fig. 2.19. Design of transmitter.

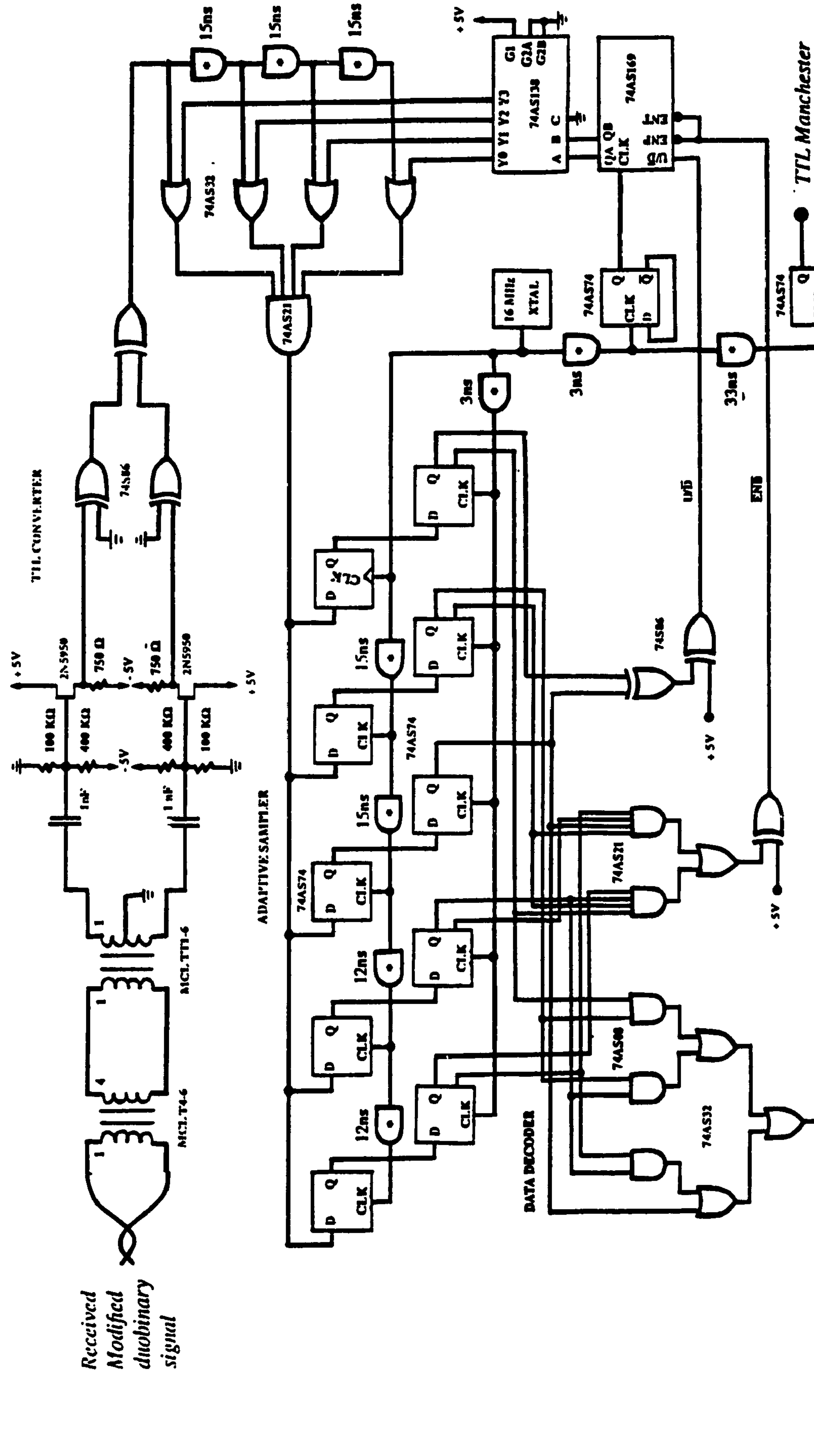



Fig. 2.20. Design of receiver.

• CASCADED 'AND' GATES TO IMPLEMENT DELAYS

TTL Manchester signal. Fig. 2.24 compares a given input binary sequence with its recovered version. A delay of about half a bit can be seen and justified by the various delays used as well as the propagation delay in the recovery circuit. After precoding, the three level MD signal obtained can be seen in Fig. 2.25 as compared to the original input data. It can be noted that by now a delay of about 2 bits has been experienced. This MD signal is filtered as shown in Fig. 2.26 to attenuate most of its higher frequency components.

To be able to see the eye diagrams of the MD signal and the frequency response of the transmitter, a pseudo-random bit generator was built to provide a long pseudo-random sequence of data. The design of the generator is shown in Fig. 2.21. and generates a maximal sequence of $2^{16}-1$ bits. In Fig. 2.27, the eye diagrams of the MD signal and its filtered version can be seen. The eye opening can be seen clearly and corresponds to their ideal sampling instants.

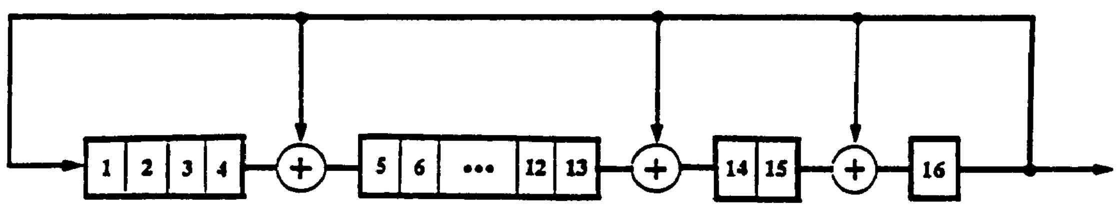



**Fig. 2.21. Design of pseudo-random generator used.**

The frequency spectrum of the input binary data can be seen in Fig. 2.28. It can seen that it occupies a bandwidth of 16 MHz with a lot of higher harmonics. Fig. 2.29 shows the frequency spectrum of the unfiltered MD signal and a bandwidth of 8 MHz can be easily recognized in the main lobe with other higher harmonics. Moreover, no D.C. component can be observed. Hence the binary data experiences a bandwidth reduction by a factor of two. The spectrum of the filtered MD signal is shown in Fig. 2.30. Most of the higher harmonics of the signal has been suppressed while most of the information has been retained by the main lobe.

At the receiver, the received MD signal and its inverted version can be seen in Fig. 2.31 The recovered data is compared to the original input test sequence in Fig. 2.34. It can be seen that the bits are recovered correctly and hence the process of bandwidth reduction is transparent to either transmitting or receiving station except for an additional delay of about two bits.

## 2.7 Remarks.

The adapter built is by no means the best circuit that can be designed to achieve bandwidth reduction. However it demonstrated that the process is easily viable using readily available components. Many improvements can be made to increase the adapter's performance but at the price of increased complexity and unit cost. For example, the whole design could be easily implemented using ECL (Emitter-coupled Logic) technology. It would provide a faster response and hence sharper pulses. However ECL chips are more expensive and require higher non-symmetric driving voltage supplies.

At the transmitter, improvement to the MD signal can be made by using a sharper low-pass filter. However the order of the improved filter is quite high before any improvement in the signal is noticed and hence is considered unfeasible as a higher filter order implies a much more complex circuit. Ordinary duobinary signaling can also be used to achieve bandwidth reduction. The presence of a large D.C. component is sometimes not welcomed in some implementations and has to be taken into account.

At the receiver, increasing the number of delays in the adaptive sampler increases the resolution of the data decoder and hence can provide greater accuracy for bit synchronization by the sampler. However the ideal number of delays cannot be computed easily except through exhaustive search and testing. Furthermore, increasing the number of samples taken for decoding decision can also improve the accuracy of the data decoder. However, it implies an increase in the number of delay levels and complexity in the decoding combinational logic. Moreover, the decoding truth table can be optimized to provide the best response. However, even with the present five samples used, its optimization has proved to be quite difficult as each solution represents a new combinational circuit to be tested. Any improvement to the data decoder should be done with the stability of the circuit in mind. The nature of the feedback from the data decoder to the adaptive sampler is quite susceptible to clock change and hence any instability may lead to data lost or perpetual clock search.

Lastly, it has to be mentioned that the adapter has not been tested for its maximum reach. Hence although a 200m twisted pair wire was used during the tests and provided excellent results, its maximum reach using MD signaling has not been measured.

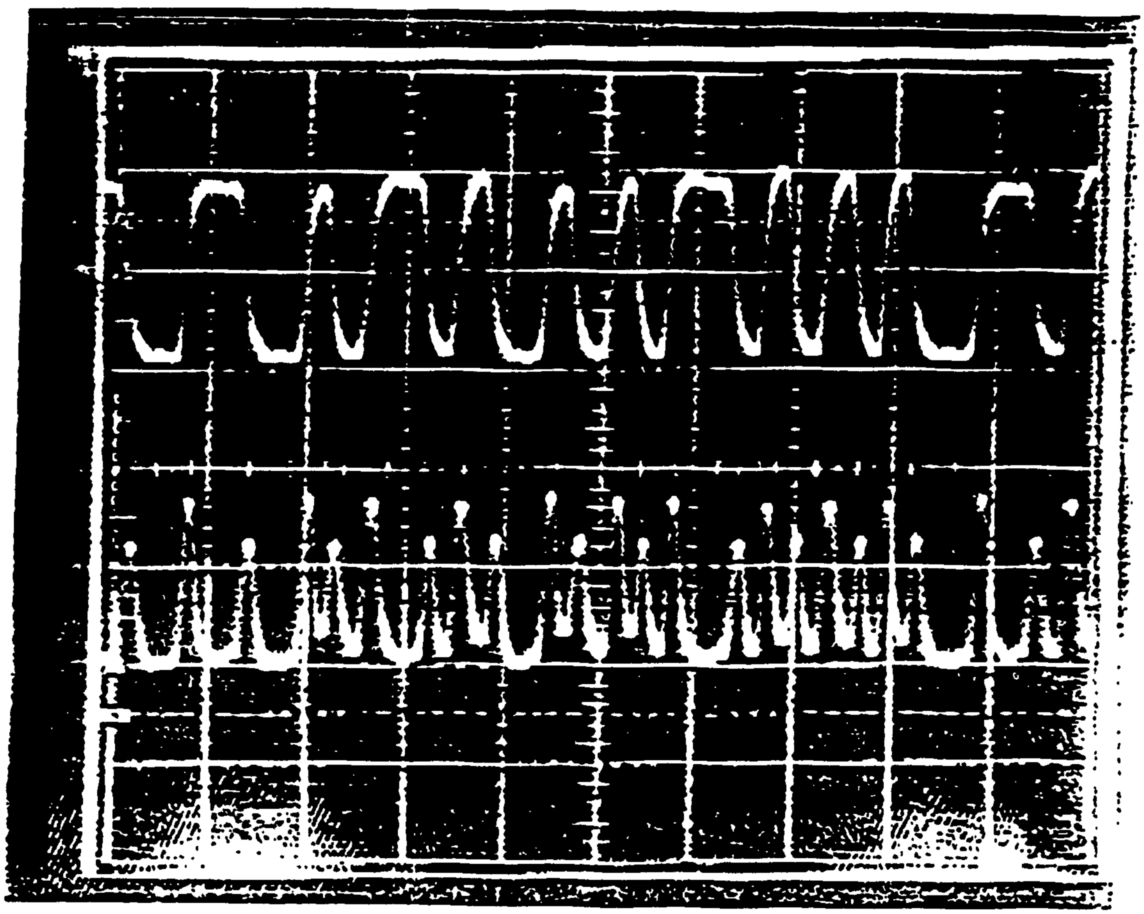


**Fig. 2.22. TTL Manchester signal (top: 2V/div) and its zero-crossings (bottom: 1V/div). (time base: 0.1 μs/div)**

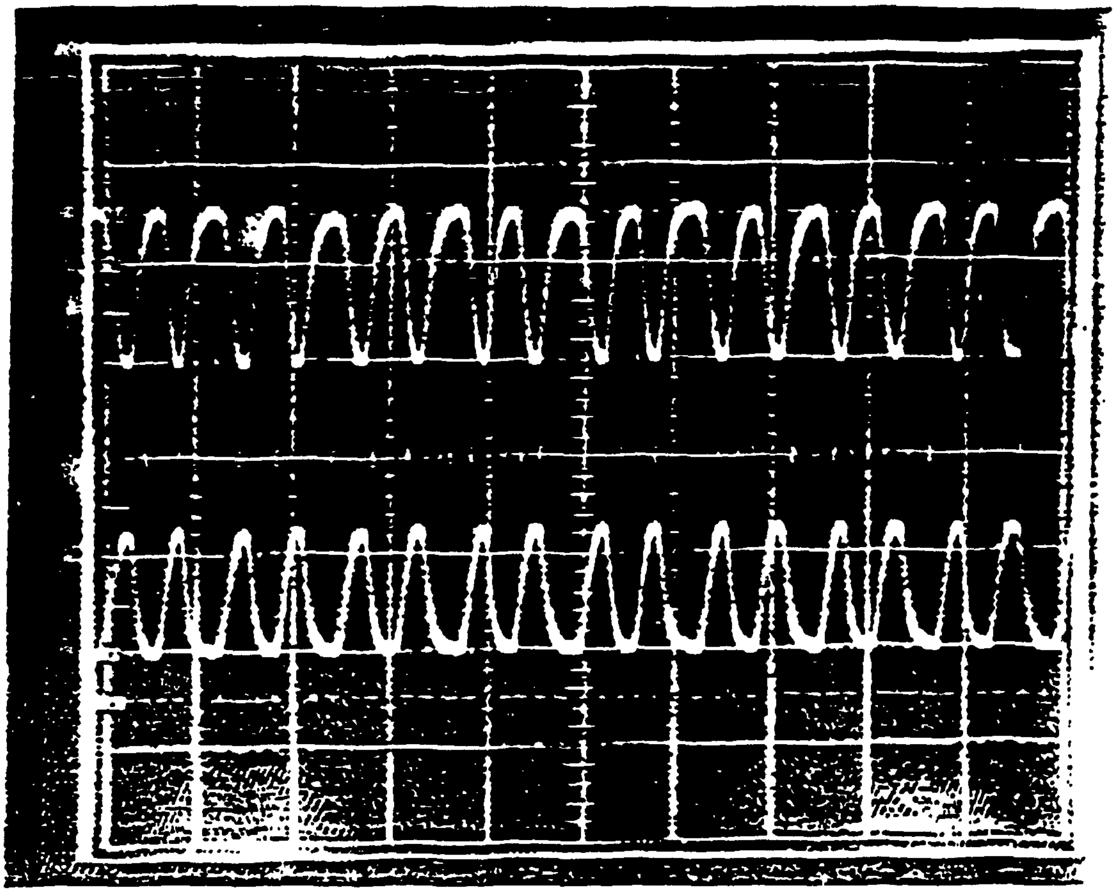


Fig. 2.23. Recovered clock (top) and its inverted version (bottom).
(2V/div, 0.1 $\mu s$/div)

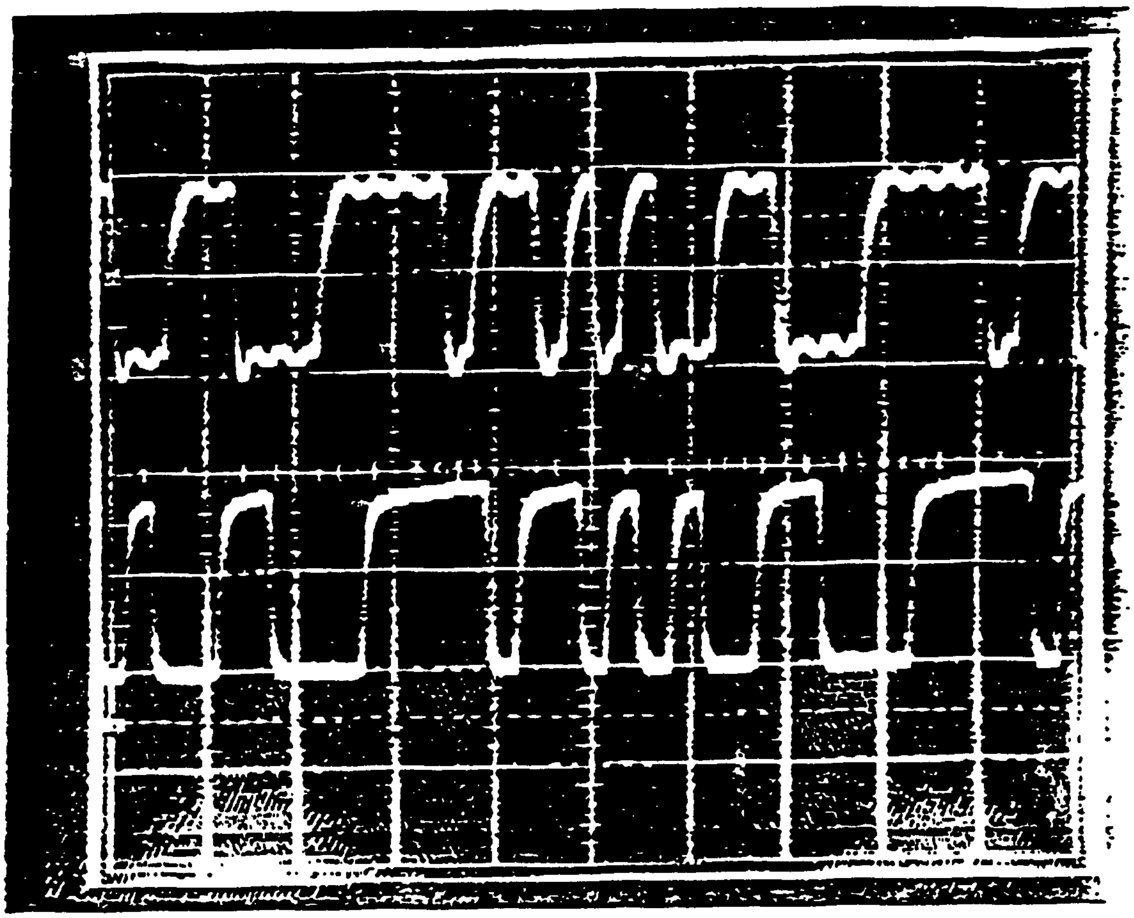


Fig. 2.24. Comparison of input (top) and recovered data (bottom) at the transmitter. (2V/div, 0.2 μs/div)

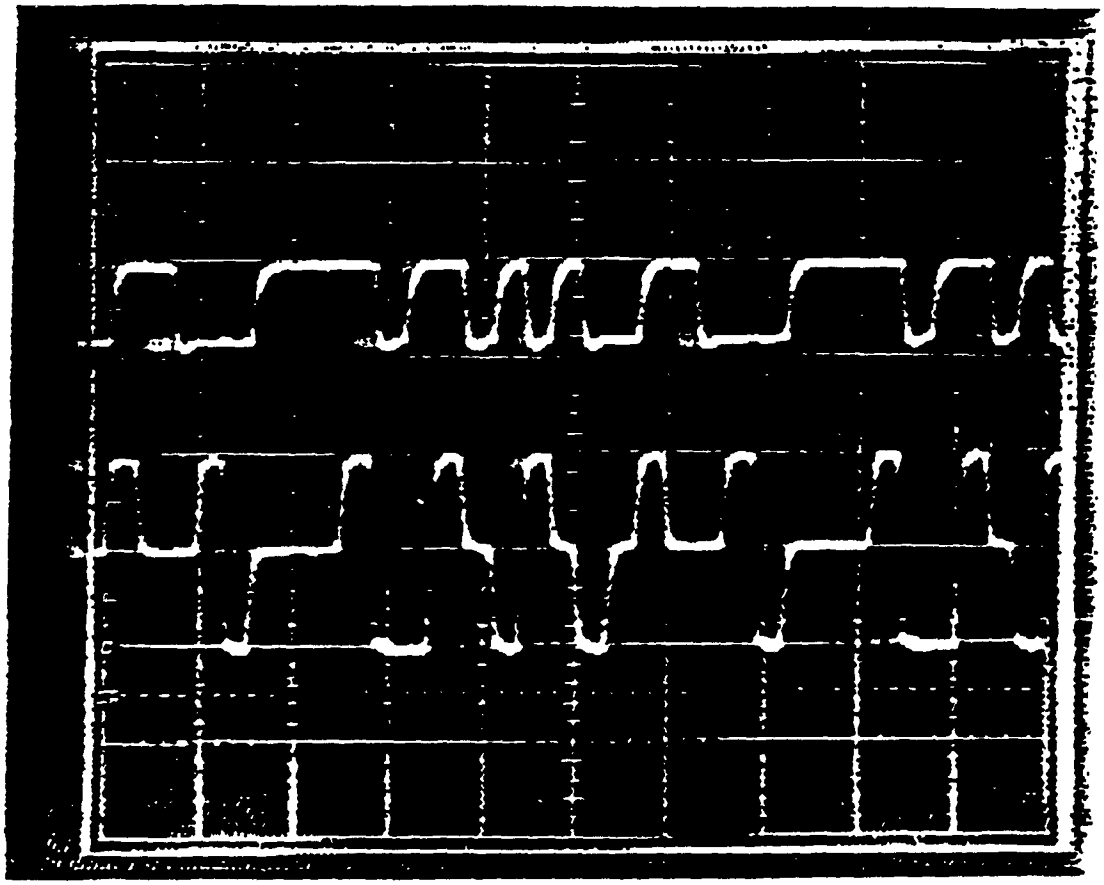


Fig. 2.25. Input data (top) and its corresponding unfiltered modified duobinary signal (bottom). (5V/div, 0.2 $\mu s$/div)

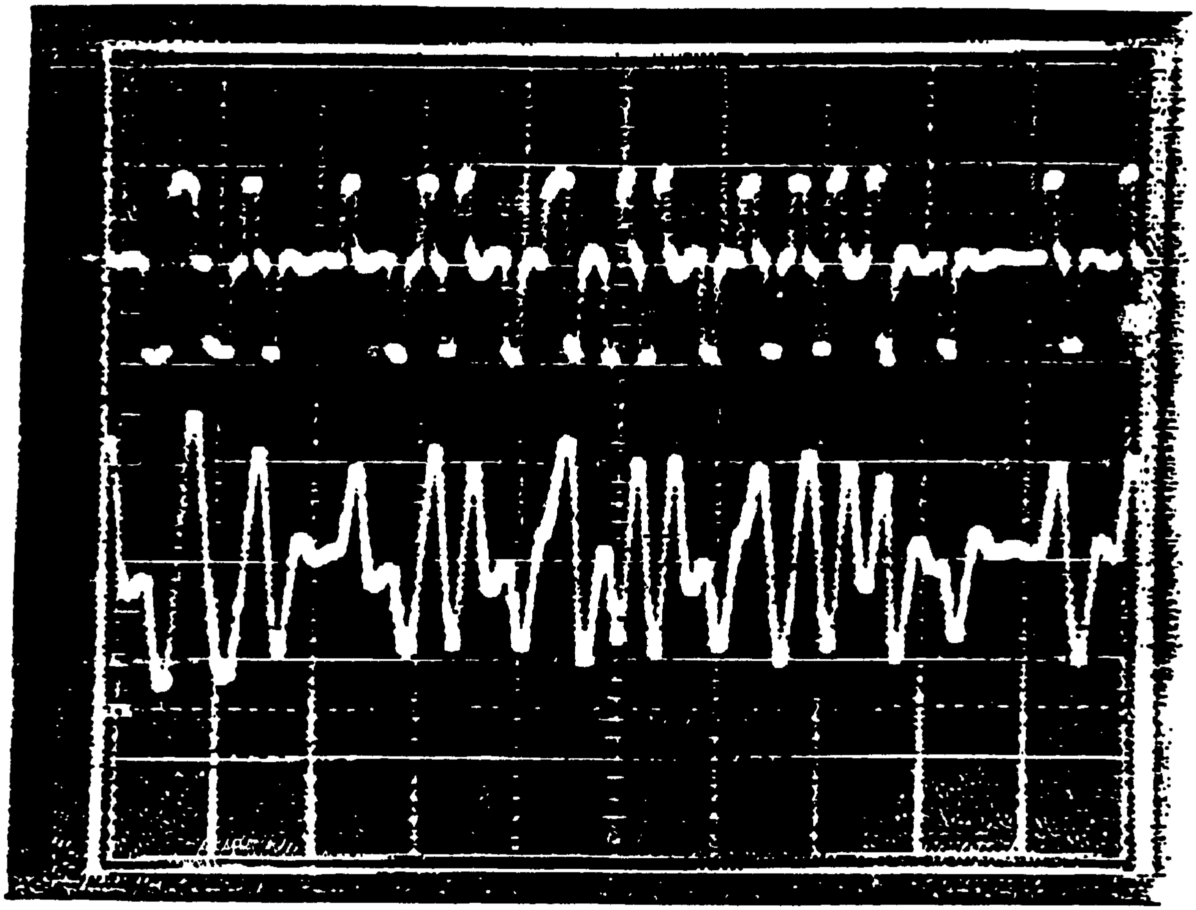


**Fig. 2.26. Unfiltered modified duobinary signal (top: 5V/div) and its filtered version (bottom: 2V/div). (0.5 $\mu s$/div)**

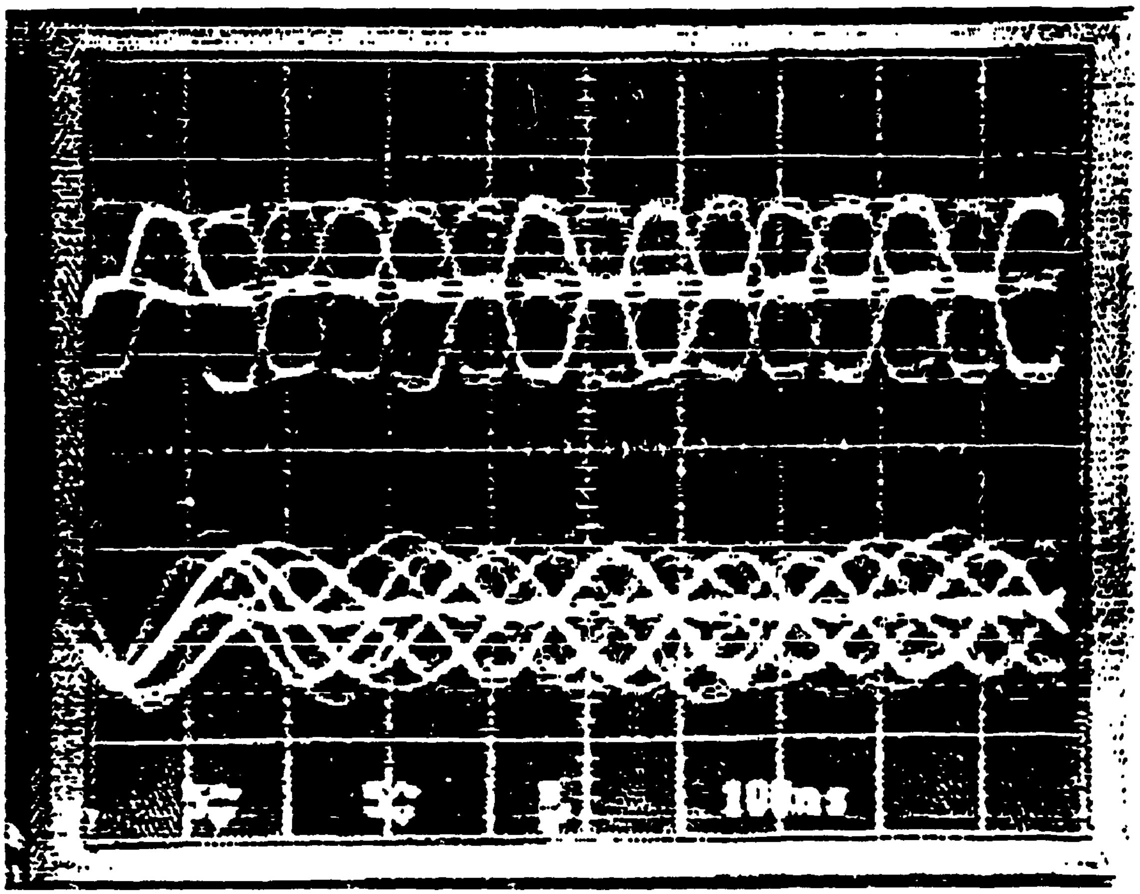


Fig. 2.27. Eye-diagram of the unfiltered (top) and filtered (bottom) modified duobinary signal.

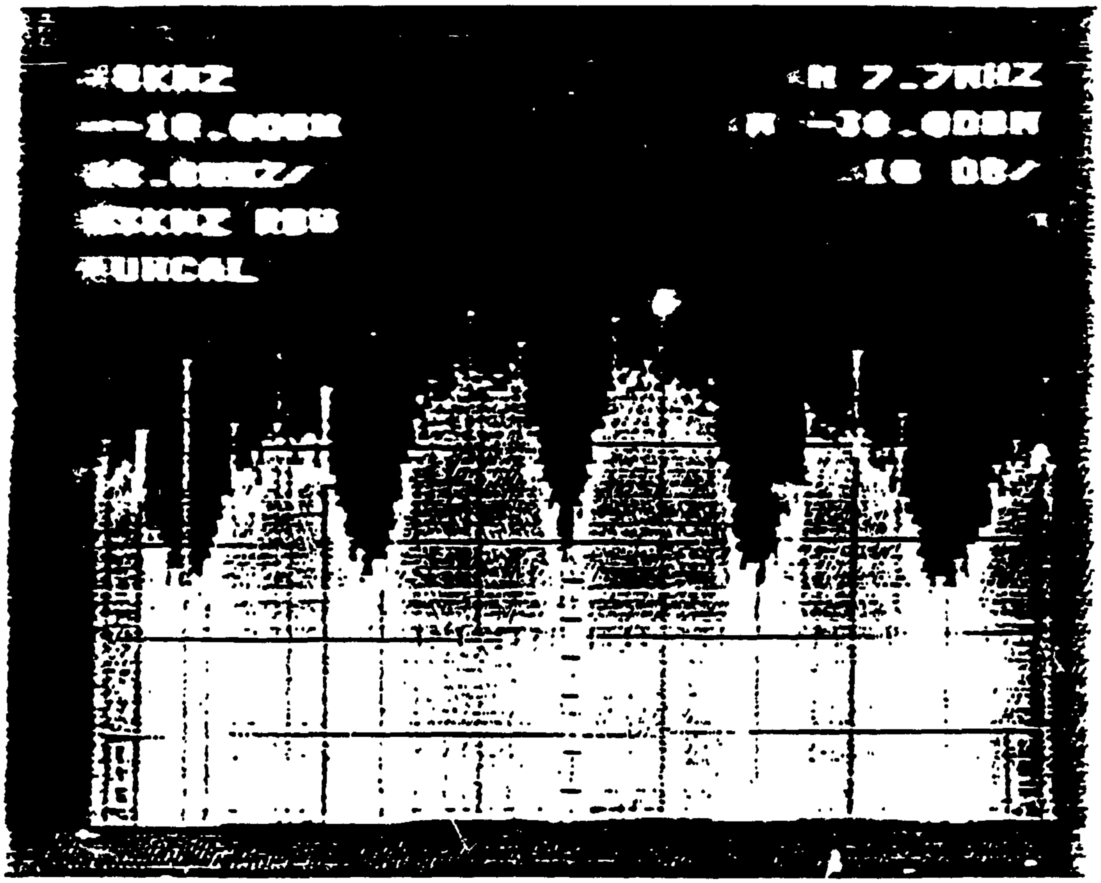


Fig. 2.28. Frequency spectrum of input binary data
(10 dBm/div, 8 MHz/div).

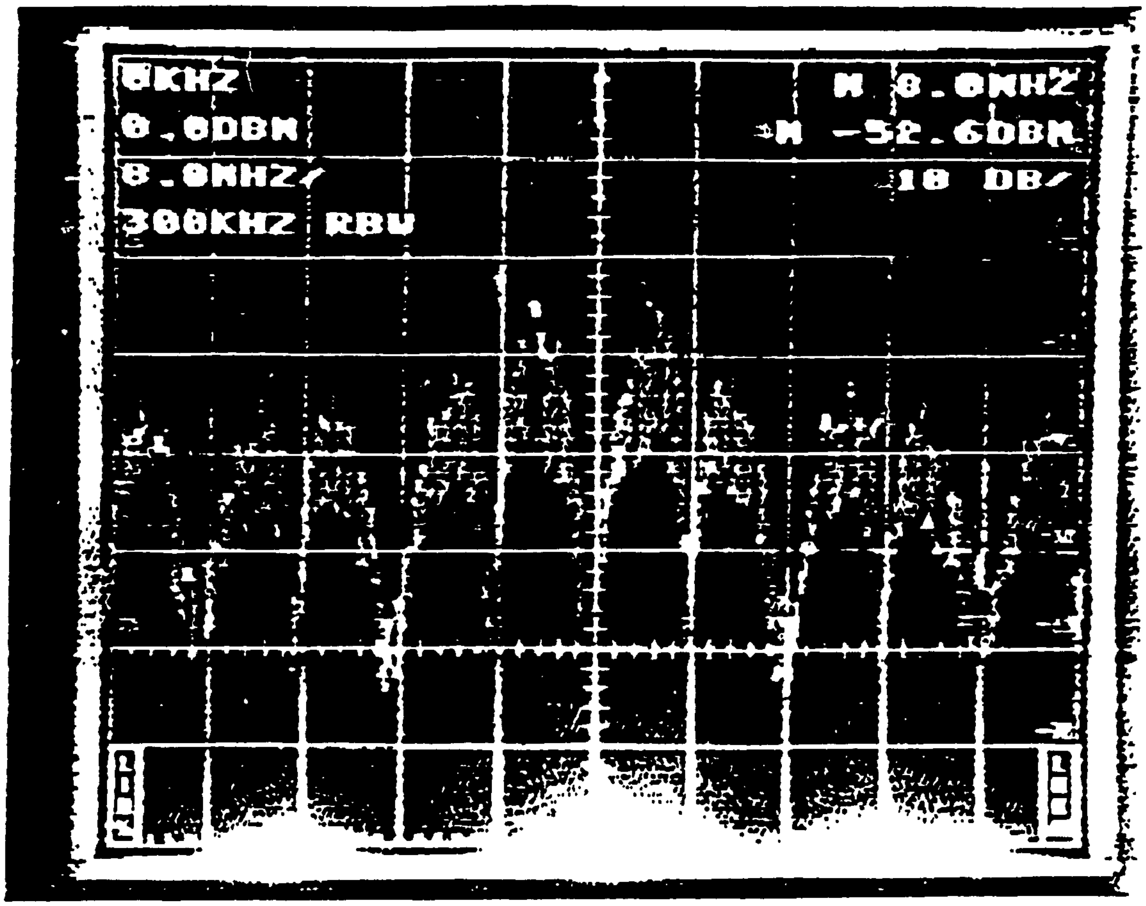



Fig. 2.29. Frequency spectrum of unfiltered modified duobinary signal (10 dBm/div, 8 MHz/div).

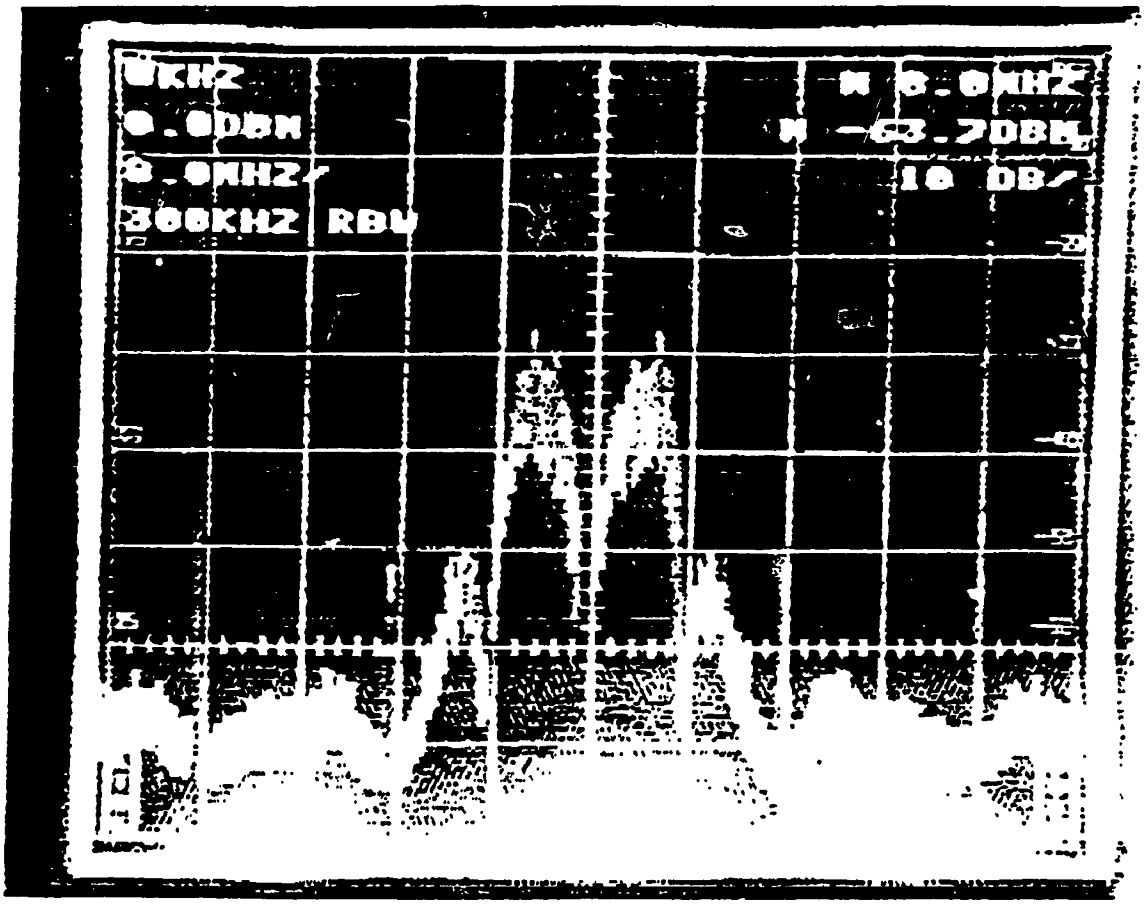



Fig. 2.30. Frequency spectrum of filtered modified duobinary signal (10 dBm/div, 8 MHz/div).

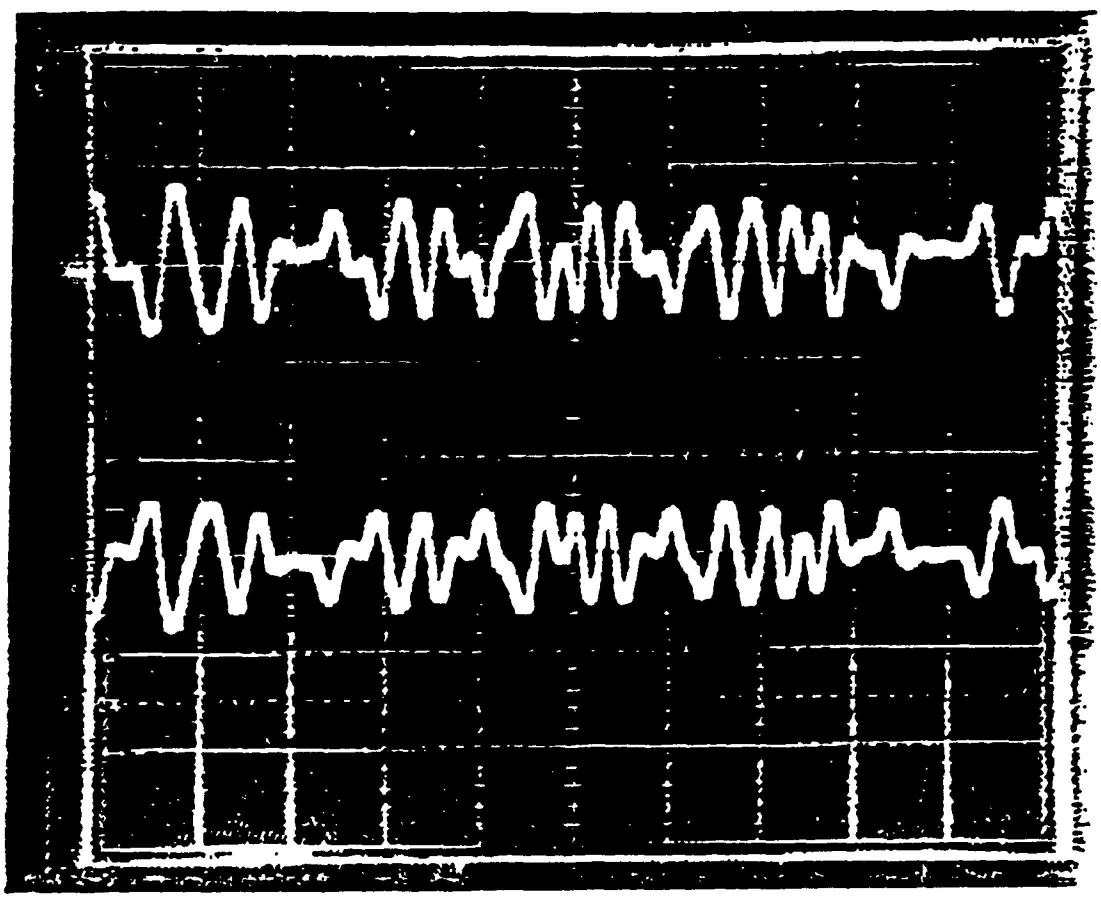


Fig. 2.31. Modified duobinary signal (top) and its inverted version (bottom) at the receiver. (5V/div, 0.5 $\mu s$/div)

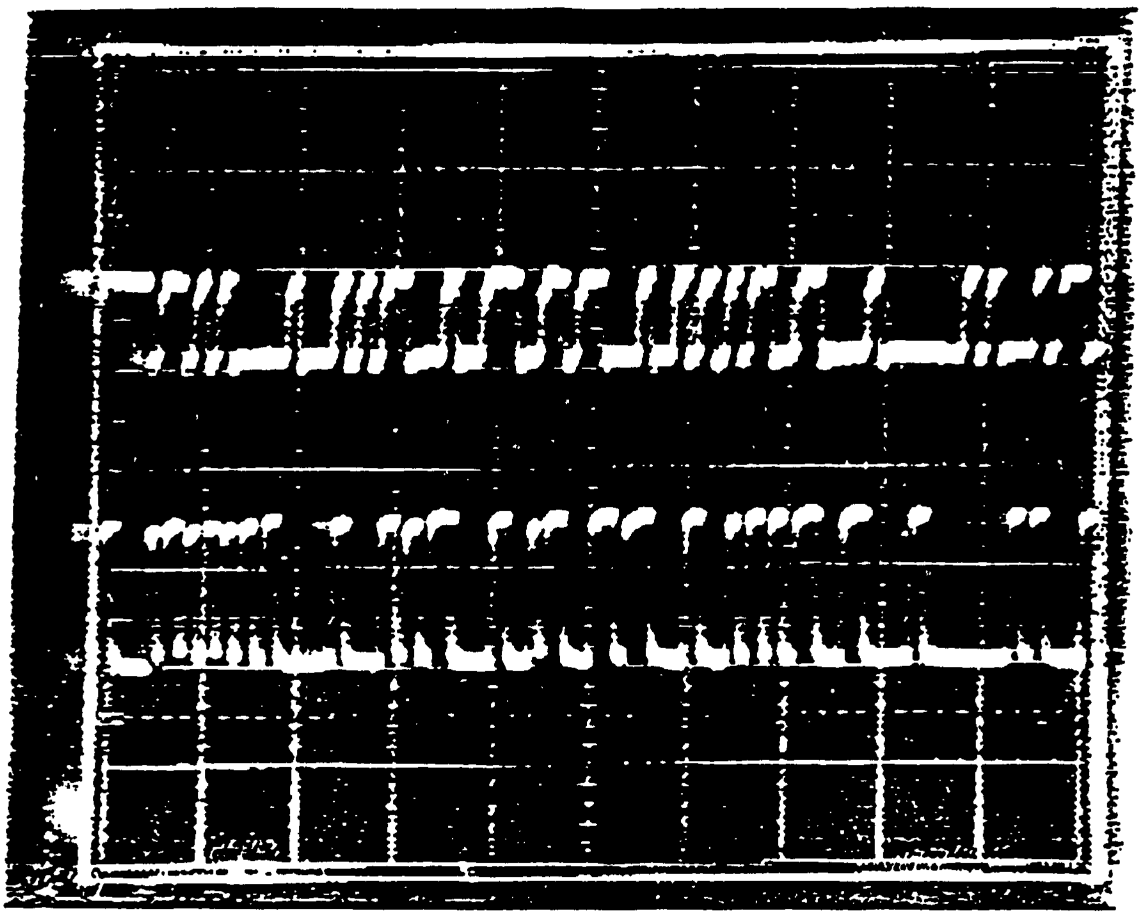


Fig. 2.32. Recovered data (bottom: 2V/div) as compared to original input data (top: 5V/div) at receiver.

# Chapter Three

# Performance analysis of adapter for a mixed voice/data load in a token ring.

## 3.1 Introduction.

The performance analysis of a token ring using the adapter to reduce the bandwidth required for transmission is presented in this chapter. It is assumed that the token ring can accommodate two types of packets namely voice and data, taking into considerations the effects of data retransmissions and modulation. The assumption is justified by the trends towards the integration of services especially with the higher bit rate of 16 Mbps. The analysis compares the performance of modified duobinary signaling with ordinary PAM and higher M-ary signaling. The possibility of using a higher M-ary partial response signaling scheme is also considered.

## 3.2 Traffic model and accessing method.

The token ring considered can host two types of stations, that is $N_d$ data-only stations and $N_v$ data/voice stations. All stations have equal access to the ring as long as they hold an empty token. The priority scheme is only limited to the data/voice stations, where the voice packets are transmitted before any buffered data packets. However there is no preemption if a data packet is being transmitted when there is a voice arrival. Two types of token management schemes will be considered, namely the single token (ST) and the multi-token (MT) token release schemes. In the ST token release scheme, the token holder keeps the token after its packet is transmitted until it receives the header. The packet is then removed from the ring and the token is released.

In the MT token release scheme, the token holder releases the token immediately after the transmission of its packet. Acknowledgement of the packet is either piggybacked on returning packets whenever possible or a separate acknowledgement packet is sent. As speech consists of talkspurts and long silence periods, speech activity detecting devices are present at each data/voice station to reduce the wasted bandwidth. The speech are digitized by ADPCMs (Analog/Digital Pulse Code Modulators) at a rate $r_v$ into packets of fixed lengths $l_v$. Hence each packet represents $T_v$ seconds of speech where $T_v = l_v / r_v$. To guarantee the quality of the voice, the access time to the ring for the data/voice stations should be less than $T_v$ and thus the maximum delay for a voice packet is bounded by $2T_v$. Note that voice packets can tolerate a delay of about 100 ms and still remain intelligible. Assuming that $R$ is the transmission rate of the ring and $l_d$ is the length of the data packet, the following bound on $T_v$ should be observed.

$$\frac{N_d \left[ l_d + o_d \right]}{R} + \frac{N_v \left[ l_v + o_v \right]}{R} < T_v, \tag{3.1}$$

where $o_d$ and $o_v$ denotes the data and voice packet overheads respectively. Obviously by choosing a number of data/voice stations $N_v$, the number of data-only stations are limited for fixed voice and data packet sizes. Conversely, by choosing a voice packet length $l_v$, the data packet length is also limited for a known number of stations on the ring.

The voice is modeled as a Poisson arrival of batches consisting of a geometrically distributed number of packets. The interarrival time of the batches is thus exponentially distributed and is available for data transmission at any station. Data packets are considered as a batch of packets and also have a Poisson arrival distribution.

## 3.3 Analysis of model.

The analysis of the token ring is divided into two parts. The first part follows the analysis performed by Karvelas and Garcia [117] of token rings for voice and data using the conditional cycle times introduced by Kuehn [118]. The second part deals with the modulation of the signal and the token management policy.

Assuming that $\bar{c}$ is the average cycle time and $\bar{u}_o$ is the average cycle time when there is no packets transmitted, the average cycle time is given as

$$\bar{c} = \bar{u}_o + \sum_{i=1}^{N_v} \left[\lambda_v \bar{c}\right] \bar{h}_v + \sum_{i=1}^{N_d} \left[\lambda_d \bar{c}\right] \bar{h}_d, \tag{3.2}$$

where $\lambda_v(i)$ and $\lambda_d(i)$ are the voice and data arrival rate at the $i^{th}$ station, $\bar{h}_v$ and $\bar{h}_d$ are their respective average transmission time. Hence the products $\lambda_v \bar{c}$ and $\lambda_d \bar{c}$ are the average number of voice and data packets arrived at a station and are less than one for the ring to be stable. They can also be regarded as the probability of voice and data arrival in a particular cycle.

Solving for $\bar{c}$ and denoting $\sum_{i=1}^{N_v} \lambda_v(i) \bar{h}_v$ as $\rho_v$ and $\sum_{i=1}^{N_d} \lambda_d(i) \bar{h}_d$ as $\rho_d$,

$$\bar{c} = \frac{\bar{u}_o}{1-\rho_v-\rho_d}. \tag{3.3}$$

For stability, it is obvious that $\rho_v + \rho_d \leq 1$. Focusing on a specific station $j$, we recognize 3 types of cycles in which it can be, namely (i) a voice cycle where it has transmitted a voice packet, (ii) a data cycle where it has transmitted a data packet and lastly (iii) an idle cycle where it has not transmitted at all.

Rewriting equation (3.2) to represent each of these conditional cycles, for voice cycles

$$\bar{c}_v = \bar{u}_o + \sum_{i \neq 1}^{N_v} \left[p_{vv}(i) \bar{h}_v\right] + \sum_{i \neq 1}^{N_d} \left[p_{dv}(i) \bar{h}_d\right] + \bar{h}_v. \tag{3.4a}$$

For data cycles,

$$\bar{c}_d = \bar{u}_o + \sum_{i \neq 1}^{N_v} \left[p_{vd}(i) \bar{h}_v\right] + \sum_{i \neq 1}^{N_d} \left[p_{dd}(i) \bar{h}_d\right] + \bar{h}_d. \tag{3.4b}$$

And for idle cycles,

$$\bar{c}_o = \bar{u}_o + \sum_{i \neq 1}^{N_v} \left[p_{vo}(i) \bar{h}_v\right] + \sum_{i \neq 1}^{N_d} \left[p_{do}(i) \bar{h}_d\right] \tag{3.4c}$$

where $p_{vk} = \lambda_v \bar{c}_k$, the probability of a voice packet arrival in a $k$ cycle, and $p_{dk} = \lambda_d \bar{c}_k$, the probability of a data packet arrival in a $k$ cycle and $k$ can be voice or data.

Solving for $\bar{c}_v$, $\bar{c}_d$ and $\bar{c}_o$,

$$\bar{c}_v = \frac{\bar{u}_o + \bar{h}_v}{1-\rho_v-\rho_d + \rho_v(j) + \rho_d(j)}, \tag{3.5a}$$

$$\bar{c}_d = \frac{\bar{u}_o + \bar{h}_d}{1-\rho_v-\rho_d + \rho_v(j) + \rho_d(j)}, \tag{3.5b}$$

$$\bar{c}_o = \frac{\bar{u}_o}{1-\rho_v-\rho_d + \rho_v(j) + \rho_d(j)}. \tag{3.5c}$$

Equations 3.4(a)-3.4(c) can also be written in terms of their Laplace-Stieltjes transforms as follows:

$$C_v(s) = U(s)\prod_{i \neq j} \left[p_{vv}(i)H_v(s) + p_{dv}(i)H_d(s) + p_{ov}(i)\right]H_v(s), \tag{3.6a}$$

$$C_d(s) = U(s)\prod_{i \neq j} \left[p_{vd}(i)H_v(s) + p_{dd}(i)H_d(s) + p_{od}(i)\right]H_d(s), \tag{3.6b}$$

$$C_o(s) = U(s)\prod_{i \neq j} \left[p_{vo}(i)H_v(s) + p_{do}(i)H_d(s) + p_{oo}(i)\right] \tag{3.6c}$$

By differentiating 3.6(a)-3.6(c), the various moments of the conditional cycle times can be found (see Appendix). Equations 3.5(a)-3.5(c) can be duplicated by differentiating 3.6(a)-3.6(c) once and substituting for $s = 0$. Similarly, differentiating twice and evaluating at $s = 0$, their second moments can be obtained. Hence denoting $\rho_v + \rho_d$ as $\rho_a$,

$$\begin{aligned}\overline{c_v^2} = \overline{u_o^2} + \overline{h_v^2} + 2\bar{u}_o\bar{h}_v + 2\bar{u}_o\bar{c}_v\left[\rho_a-\rho_v(i)-\rho_d(i)\right] \\ + 2\bar{h}_v\bar{c}_v\left[\rho_a-\rho_v(i)-\rho_d(i)\right] + \bar{c}_v\left[\sum_{k\neq j}A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2}\right] \\ + \left[\bar{c}_v\right]^2\left[\rho_a-\rho_v(i)-\rho_d(i)\right]\left[\rho_a-\rho_v(j)-\rho_d(j)-\rho_v(k)-\rho_d(k)\right]\end{aligned} \tag{3.7a}$$

$$\begin{aligned}\overline{c_d^2} = \overline{u_o^2} + \overline{h_d^2} + 2\bar{u}_o\bar{h}_d + 2\bar{u}_o\bar{c}_d\left[\rho_a-\rho_v(i)-\rho_d(i)\right] \\ + 2\bar{h}_d\bar{c}_d\left[\rho_a-\rho_v(i)-\rho_d(i)\right] + \bar{c}_d\left[\sum_{k\neq j}A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2}\right] \\ + \left[\bar{c}_d\right]^2\left[\rho_a-\rho_v(i)-\rho_d(i)\right]\left[\rho_a-\rho_v(j)-\rho_d(j)-\rho_v(k)-\rho_d(k)\right]\end{aligned} \tag{3.7b}$$

$$\overline{c_v^2} = \overline{u_o^2} + 2\overline{u}_o\overline{c}_o\left[\rho_a - \rho_v(i) - \rho_d(i)\right] + \overline{c}_o\left[\sum_{k \neq j} A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2}\right]$$

$$+ \left[\overline{c}_o\right]^2\left[\rho_a - \rho_v(i) - \rho_d(i)\right]\left[\rho_a - \rho_v(j) - \rho_d(j) - \rho_v(k) - \rho_d(k)\right] \quad (3.7c)$$

The second moment of the average cycle time can be found from

$$\overline{c^2} = p_v\overline{c_v^2} + p_d\overline{c_d^2} + \left[1 - p_v - p_d\right]\overline{c_o^2}, \quad (3.8)$$

where $p_k = A_k(j)\overline{c}$ and is the probabilities of arrivals of the different packets.

To evaluate the average delay of either type of packets the average delay of the first packet of a given batch is found and then averaged over the size of the batch. Let the average delay of the first voice packet in a batch be denoted by $w_v(1)$. This delay consists of a residual time of a previous packet being served upon its arrival, and the total waiting time of the packets that have arrived before itself. Hence $w_v(1)$ can be written as follows :

$$w_v(1) = c_r + \left[A_v(j)\overline{c}_v\right]\overline{w}_v, \quad (3.9)$$

where $c_r = \overline{c^2}/2c$ is the residual cycle of the previous packet upon its arrival [119], and $A_v(j)\overline{c}_v$ is the number of voice batches that have arrived before this present batch. For an arbitrary packet $m$ in the batch, its delay is given as

$$w_v(m) = w_v(1) + (m-1)\overline{c}_v, \quad (3.10)$$

where the second term on the right represents the service time of the packets ahead of itself. Averaging over a geometric batch distribution, the average waiting time of voice packets is given by

$$\begin{aligned}\overline{w}_v &= E\left[w_v(m)\right] \\ &= E\left[w_v(1) + (m-1)\overline{c}_v\right] \\ &= w_v(1) + \left[\frac{\overline{B_v^2} - \overline{B}_v}{2\overline{B}_v}\right]\overline{c}_v. \end{aligned} \quad (3.11)$$

where $\overline{B}_v$ is the mean batch size of the voice message.

Substituting (3.9) in (3.11),

$$\bar{w}_v = c_r + \left[A_v(j)\bar{c}_v\right]\bar{w}_v + \left[\frac{\overline{B_v^2}-\bar{B}_v}{2\bar{B}_v}\right]\bar{c}_v. \tag{3.12a}$$

Hence $\bar{w}_v$ is given as

$$\bar{w}_v = \frac{c_r + \left[\dfrac{\overline{B_v^2}-\bar{B}_v}{2\bar{B}_v}\right]\bar{c}_v}{1-A_v(j)\bar{c}_v}. \tag{3.12b}$$

The delay for the data packets can be similarly evaluated. However, the effect of the local priority of the voice packets has to be considered. Similarly as in equation (3.9), the delay for the first data packet in a given batch can be written as

$$w_d(1) = c_r + \left[A_v(j)\bar{c}_v\right]\bar{w}_v + \left[A_d(j)\bar{c}_d\right]\bar{w}_d + \left[A_v(j)\bar{c}_v\right]w_d(1). \tag{3.13}$$

where the second and third term on the right represents the respective voice and data batches to be served before its turn, and the last term represents the voice batches that have arrived during its waiting time. Thus,

$$w_d(1) = \frac{c_r + \left[A_v(j)\bar{c}_v\right]\bar{w}_v + \left[A_d(j)\bar{c}_d\right]\bar{w}_d}{1-A_v(j)\bar{c}_v}. \tag{3.14}$$

For the $m^{th}$ packet, the waiting time is given as

$$w_d(m) = w_d(1) + (m-1)c_d^*, \tag{3.15}$$

where $c_d^*$ is the average cycle time for a data cycle taking into account the possibility of a voice arrival and is given by

$$\begin{aligned} c_d^* &= \bar{c}_d + \left[A_v(j)\bar{c}_v\right]c_d^* \\ &= \frac{\bar{c}_d}{1-A_v(j)}. \end{aligned} \tag{3.16}$$

Averaging over the batch distribution,

$$
\begin{aligned}
\overline{w}_d &= E\left[w_d(m)\right] \\
&= E\left[w_d(1) + (m-1)c_d^*\right] \\
&= w_d(1) + \left[\frac{\overline{B_d^2} - \overline{B}_d}{2\overline{B}_d}\right] c_d^*,
\end{aligned}
\tag{3.17}
$$

where $\overline{B}_d$ is the data batch size. Hence substituting (3.17) in (3.13),

$$
w_d(1) = \frac{c_r + \left[A_v(j)\overline{c}_v\right]\overline{w}_v + \left[A_d(j)\overline{c}_d\right]\left[\frac{\overline{B_d^2} - \overline{B}_d}{2\overline{B}_d}\right] c_d^*}{1 - \left[A_v(j)\overline{c}_v\right] - \left[A_d(j)\overline{c}_d\right]}.
\tag{3.18}
$$

Otherwise, substituting (3.14) in (3.17), the average data delay is given by

$$
\overline{w}_d = \frac{c_r + \left[A_v(j)\overline{c}_v\right]\overline{w}_v + \left[1 - A_v(j)\overline{c}_v\right]\left[\frac{\overline{B_d^2} - \overline{B}_d}{2\overline{B}_d}\right] c_d^*}{1 - \left[A_v(j)\overline{c}_v\right] - \left[A_d(j)\overline{c}_d\right]}.
\tag{3.19}
$$

## 3.4 Impact of demodulation on performance.

The choice of any particular modulation technique determines the probability of correct transmission on the channel. To provide reliable service, retransmission of the wrong messages may be required, and hence an increase in traffic is observed. In the case considered, only data packets are retransmitted. Incorrect voice packets are considered as lost as the maximum packetization time can be exceeded. Hence for a PAM M-ary modulation scheme, the probability of symbol error is given by [114]

$$
P_{PAM}(e) = \left[\frac{1}{1 - M_{ary}}\right] erfc\left[\sqrt{\frac{3}{M_{ary}^2 - 1} SNR}\right]
\tag{3.20}
$$

For an M-ary duobinary signal, the corresponding probability of error is given by [114]

$$P_{DUO}(e) = \left[\frac{1}{1-M_{ary}^2}\right] erfc\left[\frac{\pi}{4}\sqrt{\frac{3}{M_{ary}^2-1}SNR}\right] \tag{3.21}$$

where $\log_2 M_{ary}$ is the alphabet size. Hence for a packet of size *psize* the probability of a correct packet is given as

$$p = \left[1-p(e)\right]^{\left[\frac{psize}{\log_2 M_{ary}}\right]} . \tag{3.22}$$

This probability of correctness is reflected on the traffic as follows

$$\lambda = 2\lambda_f\left[\frac{1-p^{M+1}}{p}\right], \tag{3.23}$$

where $\lambda_f$ is the first offered traffic, $p$ is the probability of of correct packet and $m$ is the maximum number of retransmissions allowed. Hence increase in the data traffic will increase the delay of the data packets and also that of voice packets from the other stations.

The token management scheme is reflected on the cycle time of both data and voice. Hence for MT token release, the service time for a station is given as

$$h_k = \left[\frac{l_k + o_k}{R}\right], \tag{3.24}$$

where $l_k$ is the packet length, $o_k$ is the packet overhead and $k = v, d$ ( voice or data). Similarly, for ST token release, the service time is given as

$$h_k = \left[\frac{l_k + o_k + N^* l_{lat}}{R}\right], \tag{3.25}$$

where $N$ is the total number of stations on the ring and $l_{lat}$ is the latency of each station.

## 3.5 Results.

The token ring considered has a transmission rate $R$ of 16 Mbps. As the maximum delay tolerable for voice reconstruction between packets is about 100 ms, the maximum packetization time $T_v$ allowed is bounded by 50 ms as $T_v$ is required for packetization and another $T_v$ at most, can be allowed for propagation delay and station latency. Hence for a 32 Kbps ADPCM encoder, the maximum voice packet length $l_v$ is 1600 bits. In this case, the packets are assumed to be of fixed length, namely $l_v = l_d = 976$ bits for both voice and data packets, that is $T_v = 30.5\ ms$. An overhead of $o_v = 48$ bits is assumed for each packet. Recalling the relation given by equation (3.1), the following bound can be derived for the ring.

$$N_v + N_d < 476. \tag{3.26}$$

Hence the total number of stations on the ring cannot exceed 476 to guarantee a reasonable voice quality and the maximum voice load it can carry is $\rho_v = 0.33$ . It is well known that speech consists of talkspurts and silence periods. An average talkspurt period of 350 ms and an average silence period of 650 ms was assumed [120]. Thus, for a 32 Kbps ADPCM, the average number of bits processed in a talkspurt is 11,200. For an average batch size of 11.4754, a maximum of 285 data/voice stations can be accommodated, for a total voice load of $\rho_v = 0.2$. Hence the number of data stations assumed to be connected has been chosen to be 190.

Fig. 3.1a-3.1d show the performance of the ring for voice and data packets. To reflect the bandwidth compression when a higher alphabet size is used, the horizontal scale has been expanded by the bandwidth efficiency $\eta_{eff}$, of the signaling scheme, that is $\log_2 M$ for PAM signals and $2\log_2 M$ for duobinary signals, where M is the alphabet size. Hence for a fixed allocated bandwidth, the ring can carry a higher load for larger alphabet size as illustrated by Fig. 3.2a-3.2d. It can be observed that the delay for voice is still acceptable at low loads. Higher alphabet size seems to be more efficient as a higher throughput per Hertz is achieved. However, as it will be seen later in Fig. 3.4a-3.4d, the higher is the alphabet size, the larger is the probability of incorrect reception. Hence the quality of the voice is severely degraded. Increasing the number of retransmissions can solve the problem for data packets, but for voice packets, retransmission will extend the packet delay over the limit tolerable for intelligible voice reproduction at the receiver.

Fig. 3.3a-3.3d illustrates the advantage of MT token release scheme over the ST token release scheme for both PAM and duobinary signaling. The net improvement on the delay is due to the more efficient use of the ring's cycle time as no time is wasted by the transmitting station to release the token after its packet transmission. Although the difference between MT token release and ST token release shrinks at high load, the effective first arrival rate tolerated by the MT token release scheme is much higher than by the ST token scheme (see Fig. 3.7a-3.7d and Fig. 3.8a-3.8d). This is because of the inherently better efficiency of MT token release as compared to the ST token release.

Fig. 3.4a-3.4d illustrates the effect of retransmissions on the probability of error for both voice and data packets. The graphs are valid for both ring's accessing modes. It can be observed that the packet error in duobinary signaling is much larger than that of the PAM signaling for the same SNR. This is because the duobinary signals are harder to decode as more signaling levels have to be discriminated. The graphs also show the effect of increasing alphabet size with the probability of packet error. For an 8-ary PAM signal, a probability of error of 0.46 can be achieved with a SNR of 100 (20 dB). For a 4-ary PAM signal, the SNR required is 25, while for a binary signaling scheme, a SNR of 5.3 only is sufficient to achieve the same probability of error. Hence it can be seen that the higher the alphabet size, a more powerful signal is required, or for the same power budget, a smaller reliability figure can be achieved. However, it can also be seen on the same figure that the probability of error can be reduced to almost zero if retransmission of the wrong message is allowed. In our case, the number of retransmission $m=8$ would be a reasonable choice. However. the maximum delay can not be guaranteed and thus voice packets are not retransmitted.

Fig. 3.5a-3.5d and Fig. 3.6a-3.6d show the effect of retransmission on packet delay and as well as on the data traffic. Note that $\rho_d$ has been normalized per bandwidth unit to account for the bandwidth efficiency of the different signaling schemes., and also all the external parameters such as the first arrival rate $\lambda_f$, SNR and the voice load $\rho_v$, are constant for easier comparison. Comparing Fig. 3.5a and 3.5c, it can be seen that indeed the MT token release scheme is much more efficient than the ST token release scheme. The delay for the ST token release scheme increases dramatically for higher alphabet size as the probability of error is large and hence extra traffic due to retransmissions overload the network. This can be readily observed in the corresponding graphs, Fig. 3.5b and 3.5d. The same phenomenon or even worst can be observed for duobinary signaling as its probability of error is even larger than the one for PAM (see Fig. (3.6a-3.6d). It can also be observed that voice packets are equally affected by the retransmission of the data packets even though they are no retransmitted. This is due to the fact that voice packets have only a local priority and hence have to wait for the data-only stations to be served before their turn. However their traffic load remains constant.

Lastly, the effect of different latency times of the stations on the ring are shown in Fig. 3.7a-3.7d for ST token release and in Fig. 3.8a-3.8d for MT token release. As expected, larger latency at each station increases the delay of both voice and data packets. Moreover, larger latency times increase the load on the ring as the probability of an arrival in an extended period is larger and hence more stations will require service. The latency time has been assumed to be small as compared to the length of a packet. Otherwise the bound on the number of stations on the ring, equation 3.1, has to be rewritten to take into account the time wasted at each station. The PAM and duobinary signaling schemes show similar delay characteristics except for the better bandwidth efficiency of the latter scheme. However, comparing their accessing modes, the advantage of the MT token release is quite obvious. Looking at Fig. 3.7a, it can be observed that increasing the latency of the stations increases tremendously the load on the network as the probability of any station requiring service is larger for they have a longer time to receive an arrival. However looking at Fig. 3.8a no such characteristics can be observed as the ring can handle the extra load easily, thus proving its superiority.

## 3.5 Conclusion.

The analysis showed that voice can be accommodated easily on the ring as packets. The end-to-end delay for a reliable voice quality can be respected. However the use of higher signaling schemes can be profitable only if a high SNR is available or else retransmission of the incorrect packets should be considered. Retransmission of the voice packets is not possible as the end-to-end delay will not be respected. The MT token release scheme was shown to be much more efficient than the ST token release scheme. Not only did it yield a smaller delay, it showed that it can handle the extra load due to retransmissions and/or a higher station latency readily.

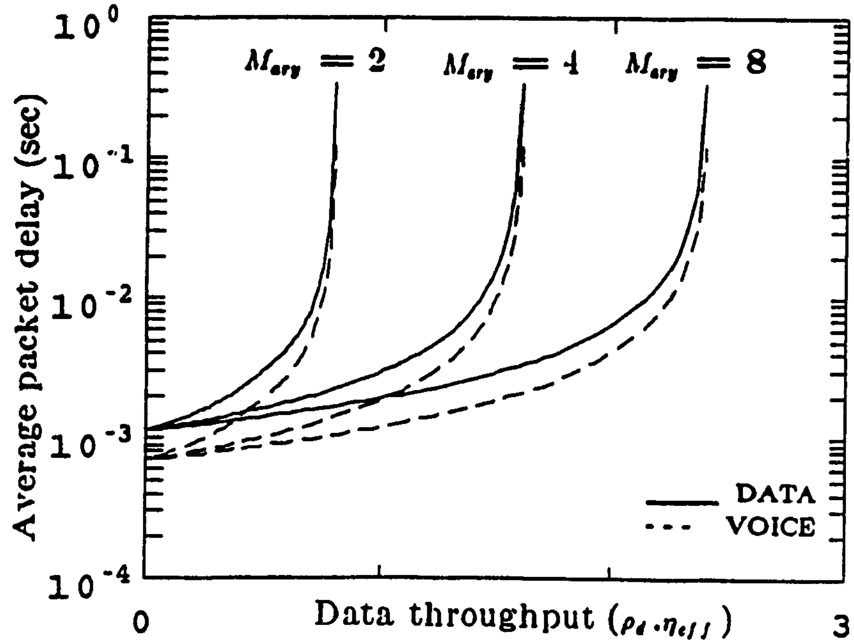



**Fig. 3.1a. Performance of data and voice packets for Multiple Token release and PAM signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

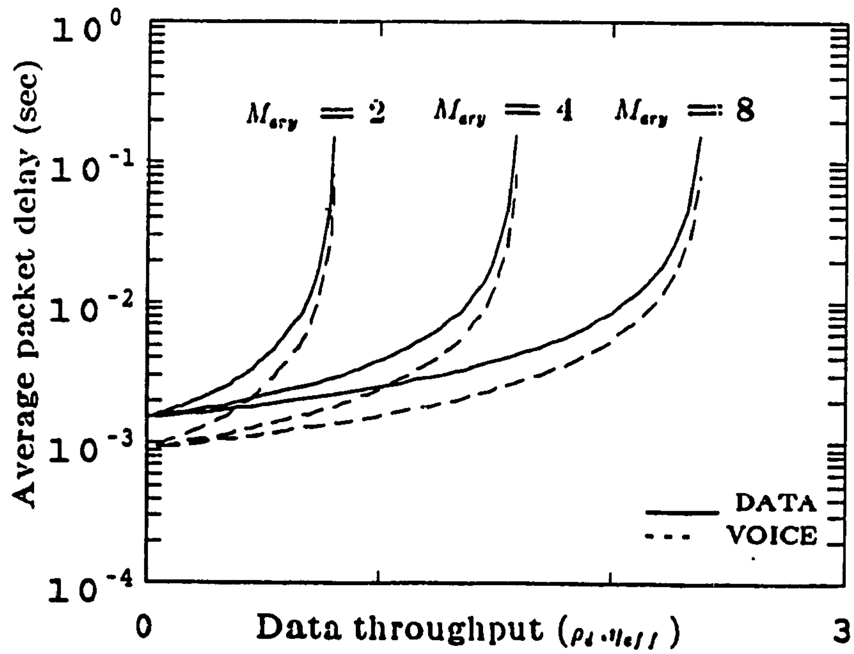



**Fig. 3.1b. Performance of data and voice packets for Single Token release and PAM signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

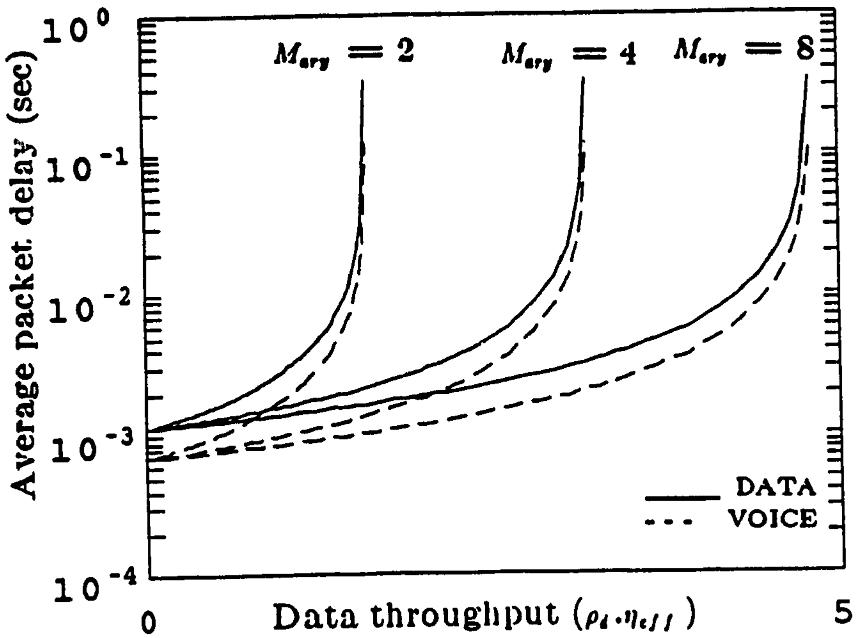



**Fig. 3.1c. Performance of data and voice packets for Multiple Token release and duobinary signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

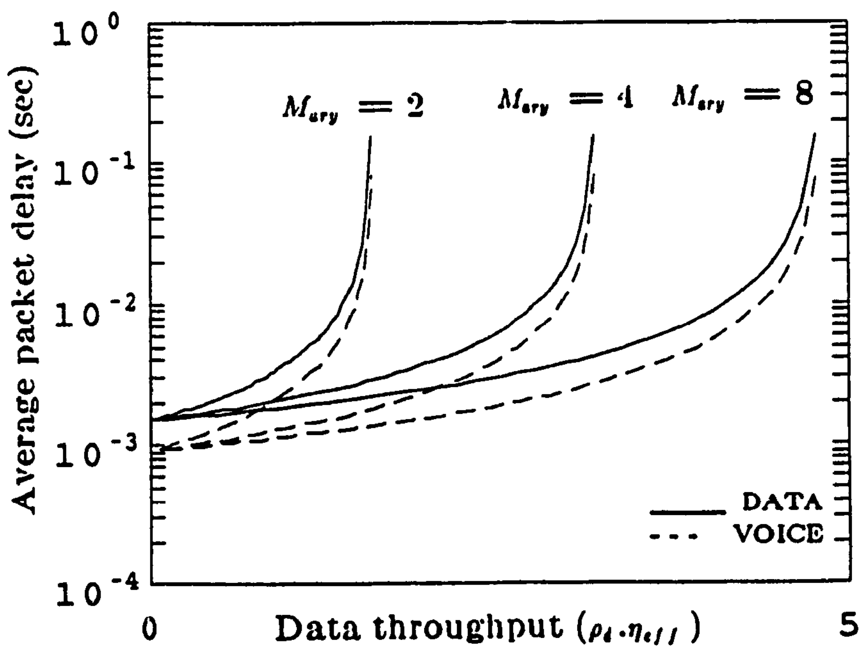



**Fig. 3.1d. Performance of data and voice packets for Single Token release and duobinary signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

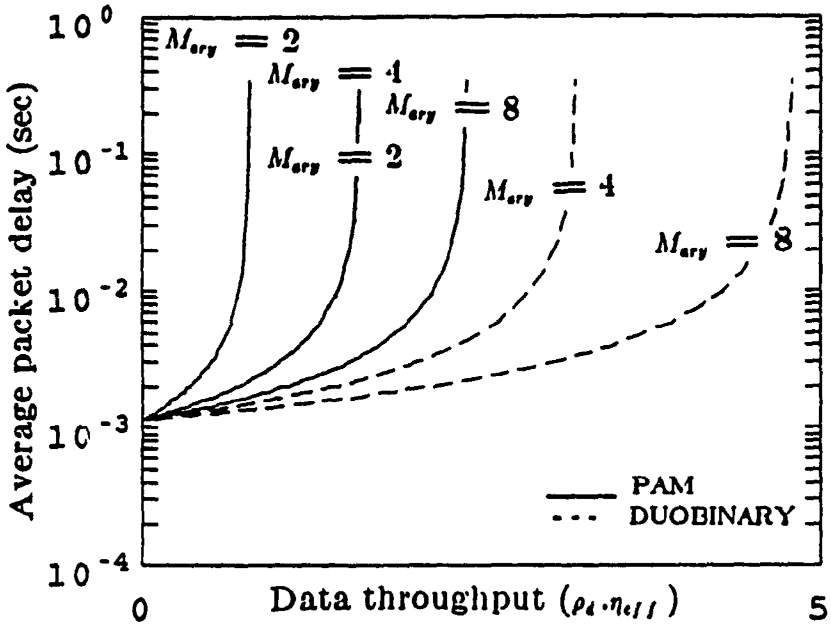



**Fig. 3.2a. Performance comparison using PAM and duobinary signaling of data packets and Multiple Token release (SNR = 100, $\rho_v$ = 0.2, m = 0).**

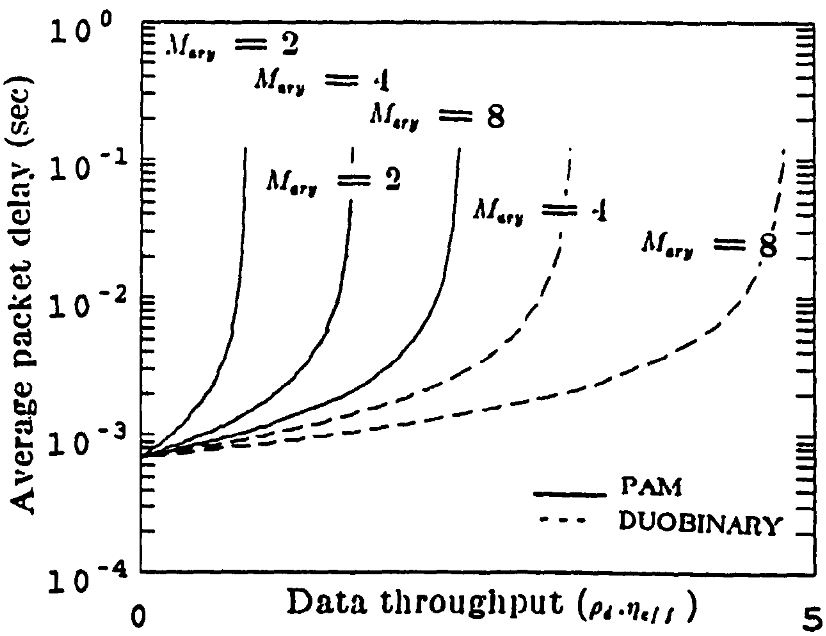



**Fig. 3.2b. Performance comparison using PAM and duobinary signaling of voice packets and Multiple Token release (SNR = 100, $\rho_v$ = 0.2, m = 0).**

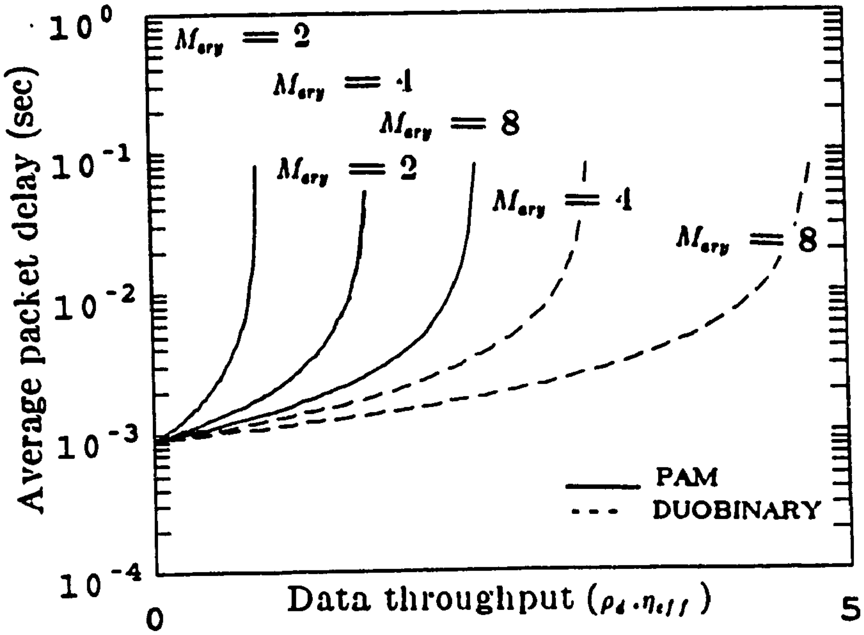



**Fig. 3.2c. Performance comparison using PAM and duobinary signaling of data packets and Single Token release (SNR = 100, $\rho_v$ = 0.2, m = 0).**

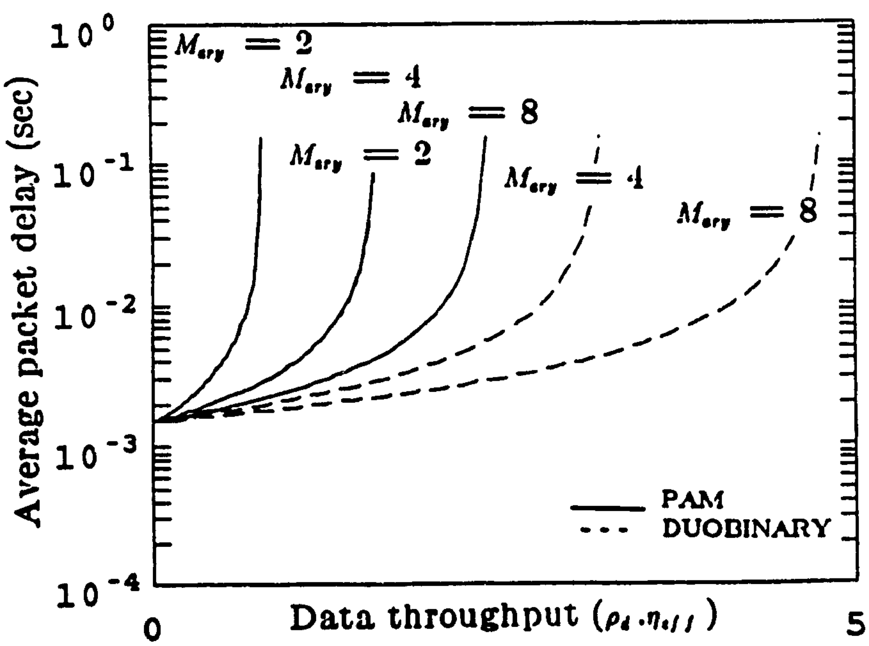



**Fig. 3.2d. Performance comparison using PAM and duobinary signaling of voice packets and Single Token release (SNR = 100, $\rho_v$ = 0.2, m = 0).**

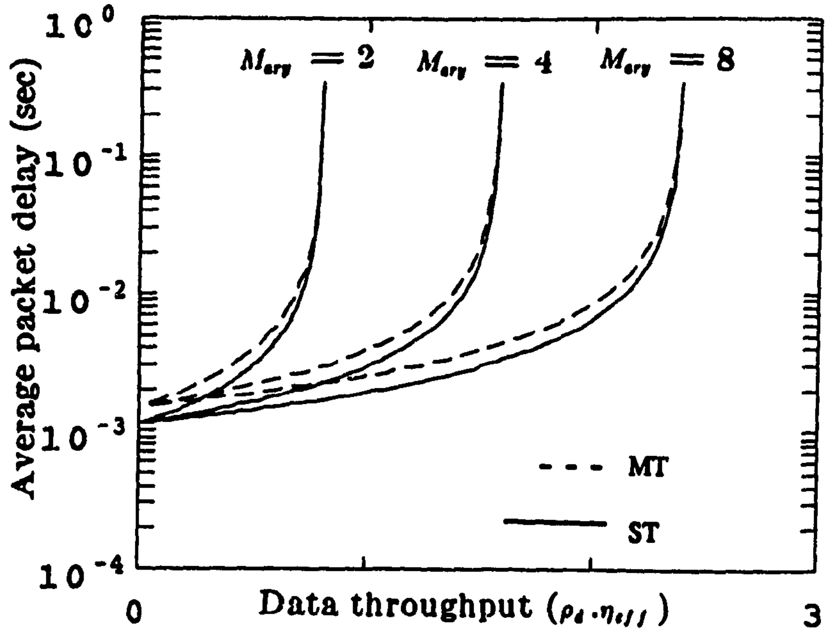



**Fig. 3.3a. Performance comparison of Multiple Token and Single Token release schemes for data packets using PAM signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

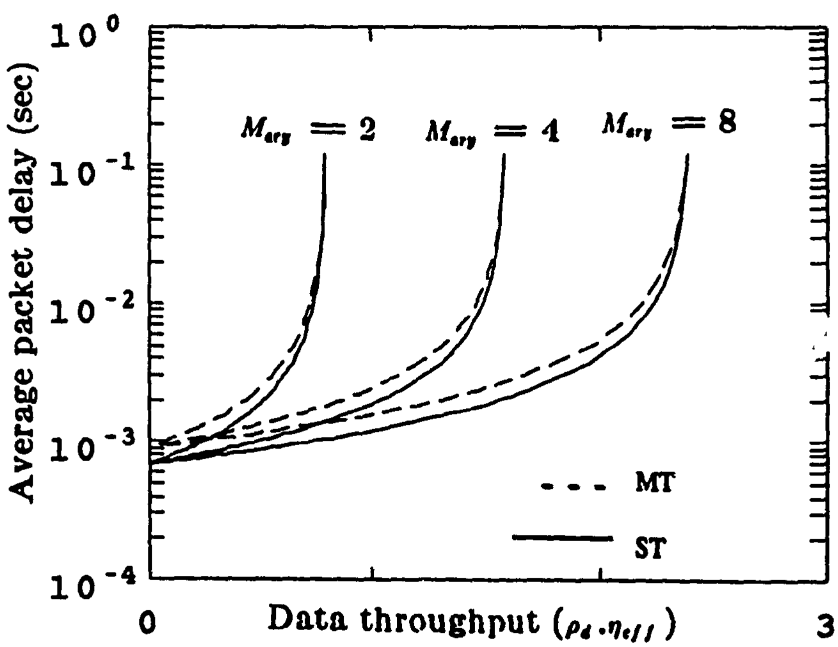



**Fig. 3.3b. Performance comparison of Multiple Token and Single Token release schemes for voice packets using PAM signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

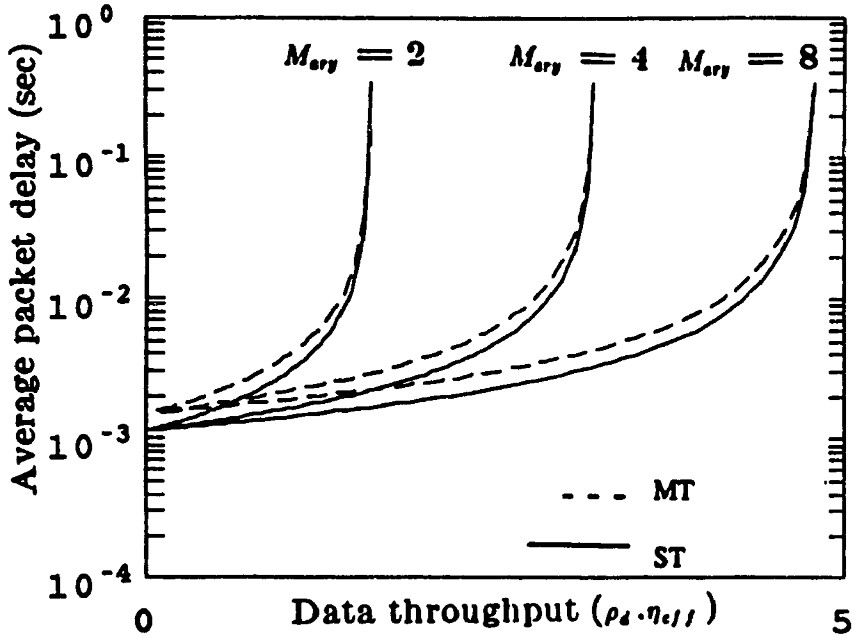



**Fig. 3.3c. Performance comparison of Multiple Token and Single Token release schemes for data packets using duobinary signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

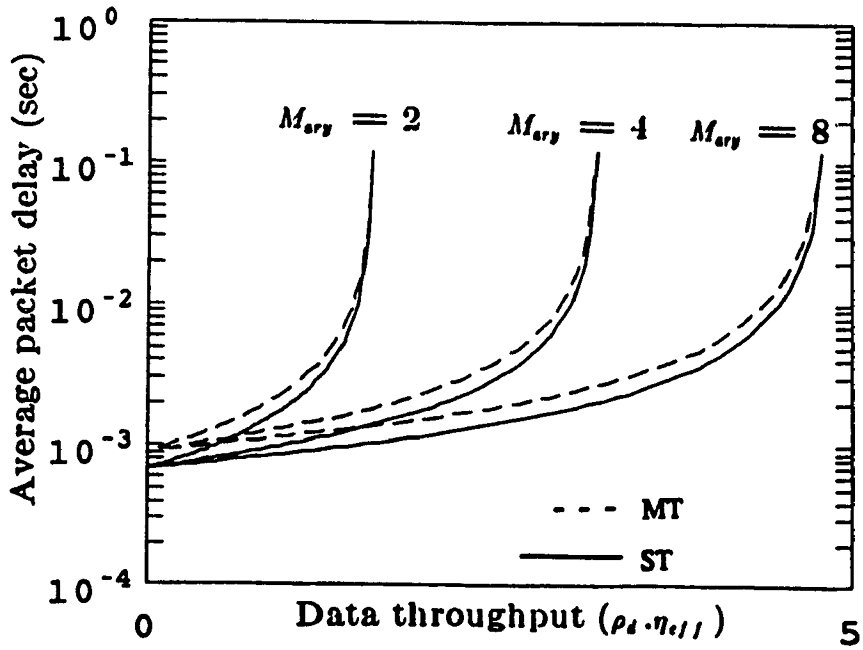



**Fig. 3.3d. Performance comparison of Multiple Token and Single Token release schemes for voice packets using duobinary signaling (SNR = 100, $\rho_v$ = 0.2, m = 0).**

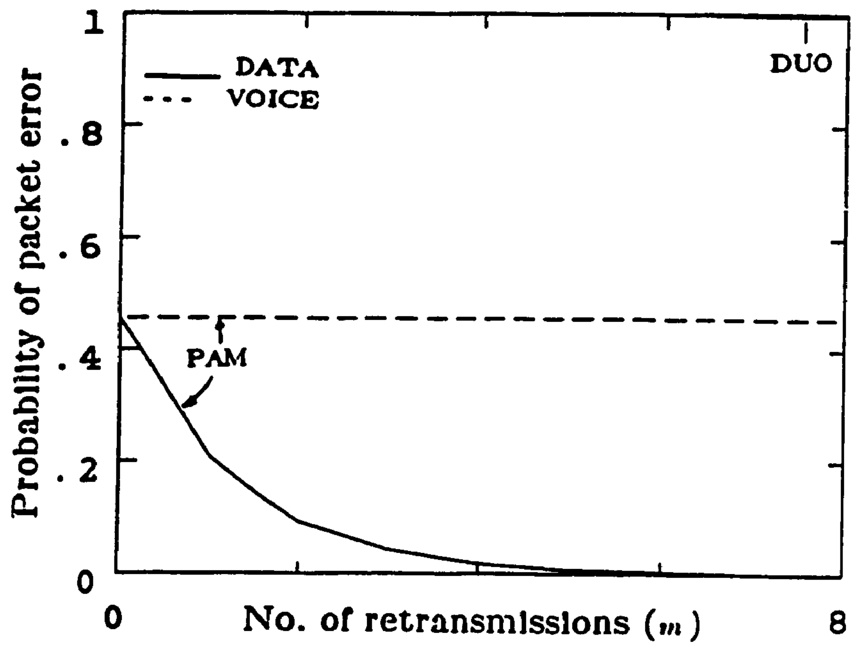



**Fig. 3.4a. Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 2$, SNR = 5 and $\rho_v = 0.2$.**

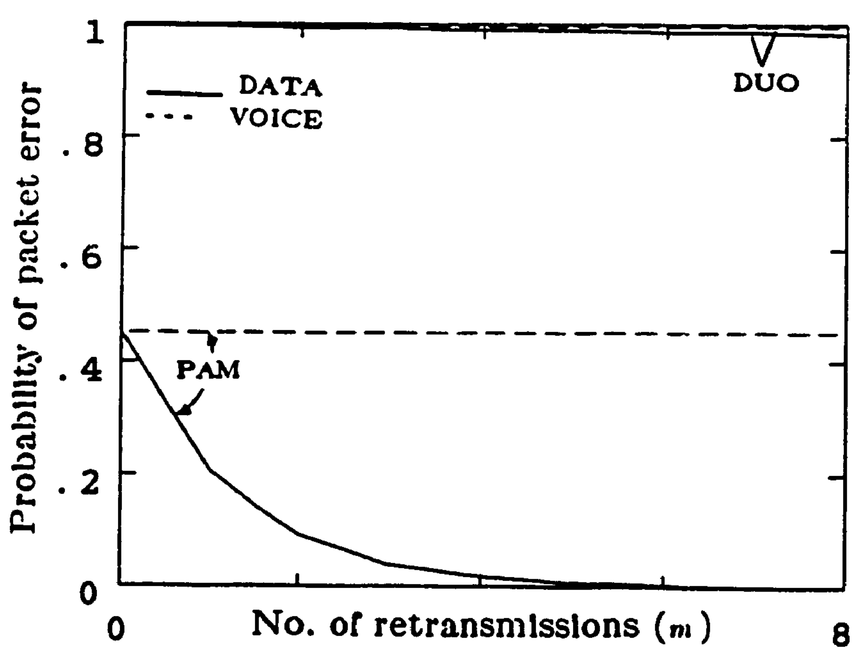



**Fig. 3.4b. Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 4$, SNR = 25 and $\rho_v = 0.2$.**

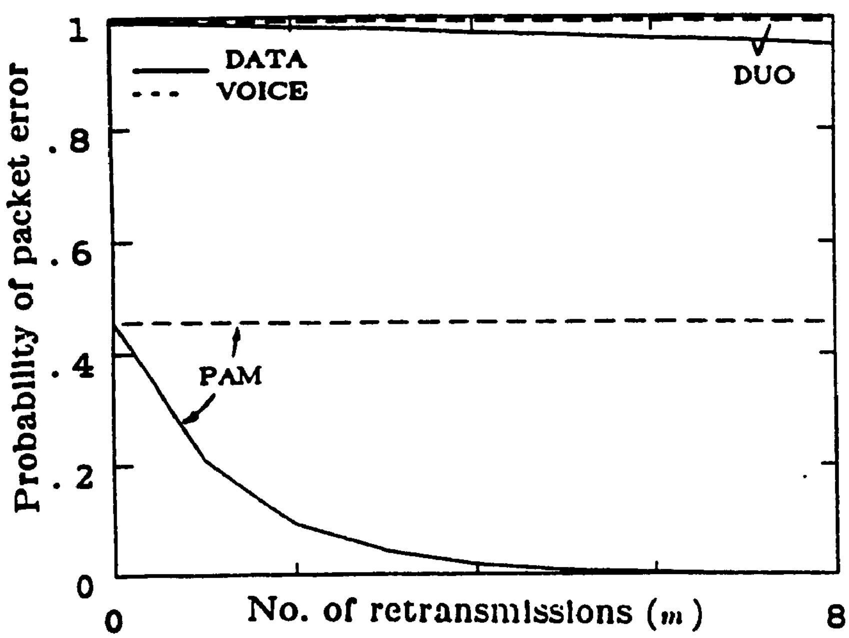



**Fig. 3.4c. Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 8$, SNR = 100 and $\rho_v = 0.2$.**

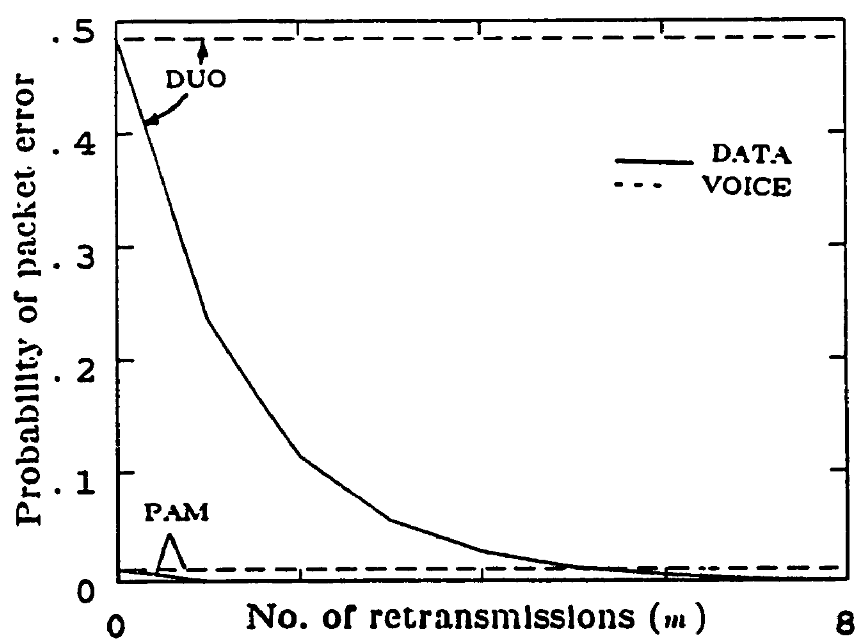



**Fig. 3.4d. Probability of packet error for PAM and duobinary signaling with retransmissions for $M_{ary} = 2$, SNR = 9 and $\rho_v = 0.2$.**

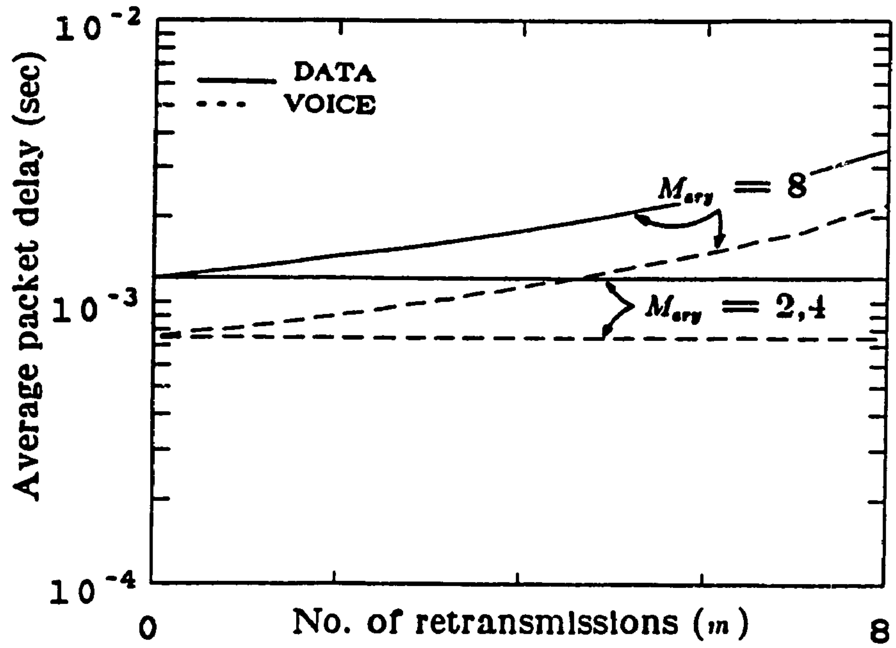



**Fig. 3.5a. Effect of packet retransmissions on packet delay for PAM using Multiple Token release (SNR = 45, $\rho_v = 0.2$).**

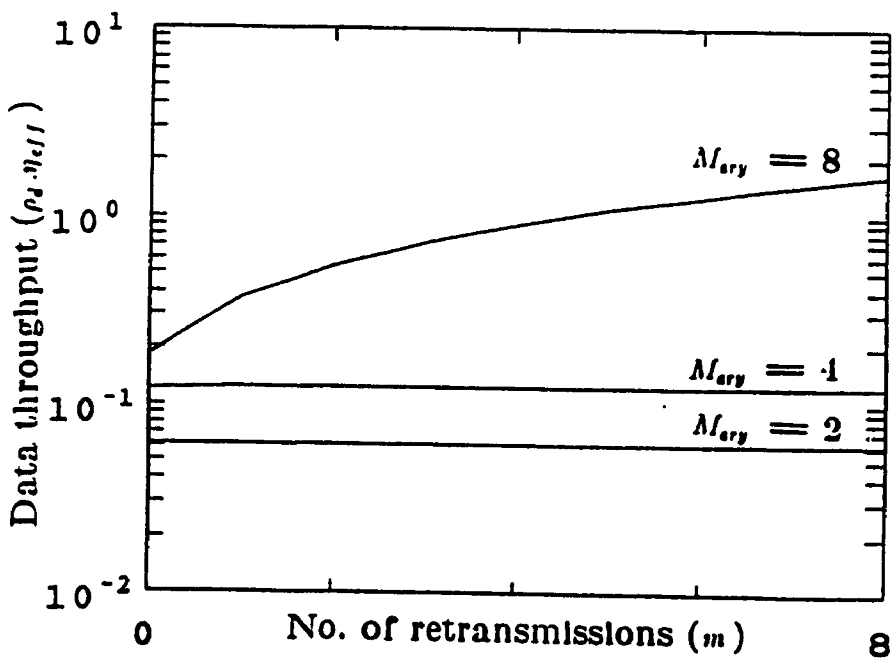



**Fig. 3.5b. Effect of packet retransmissions on data load $\rho_d$ for PAM using Multiple Token release (SNR = 45, $\rho_v = 0.2$).**

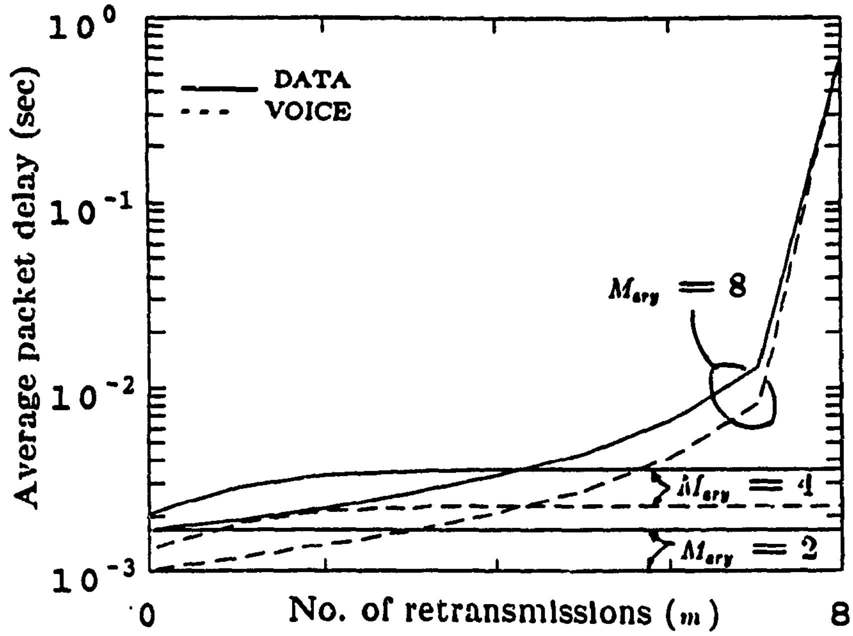



**Fig. 3.5c. Effect of packet retransmissions on packet delay for PAM using Single Token release (SNR = 45, $\rho_v$ = 0.2).**

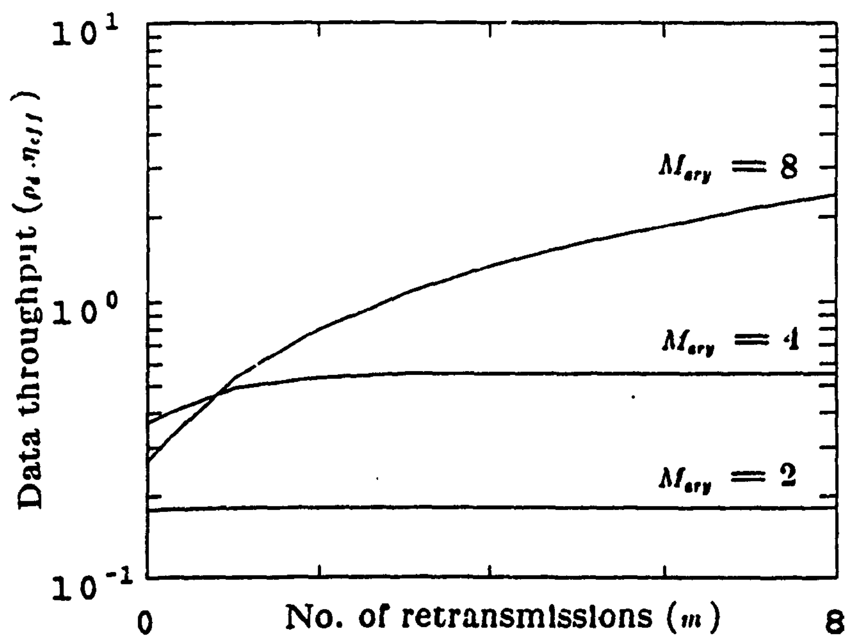



**Fig. 3.5d. Effect of packet retransmissions on data load $\rho_d$ for PAM using Single Token release (SNR = 45, $\rho_v$ = 0.2).**

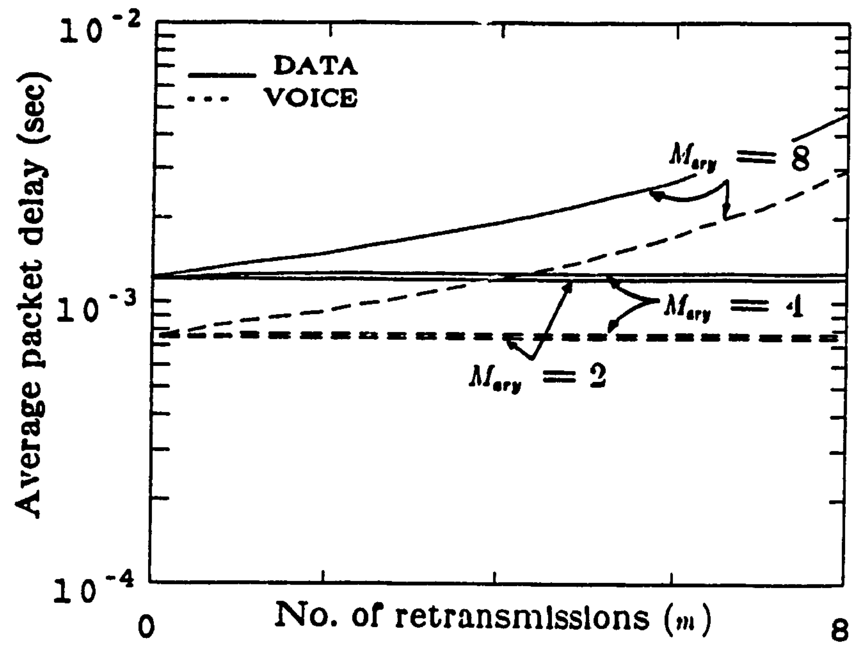



**Fig. 3.6a. Effect of packet retransmissions on packet delay for modified duobinary using Multiple Token release (SNR = 45, $\rho_v = 0.2$).**

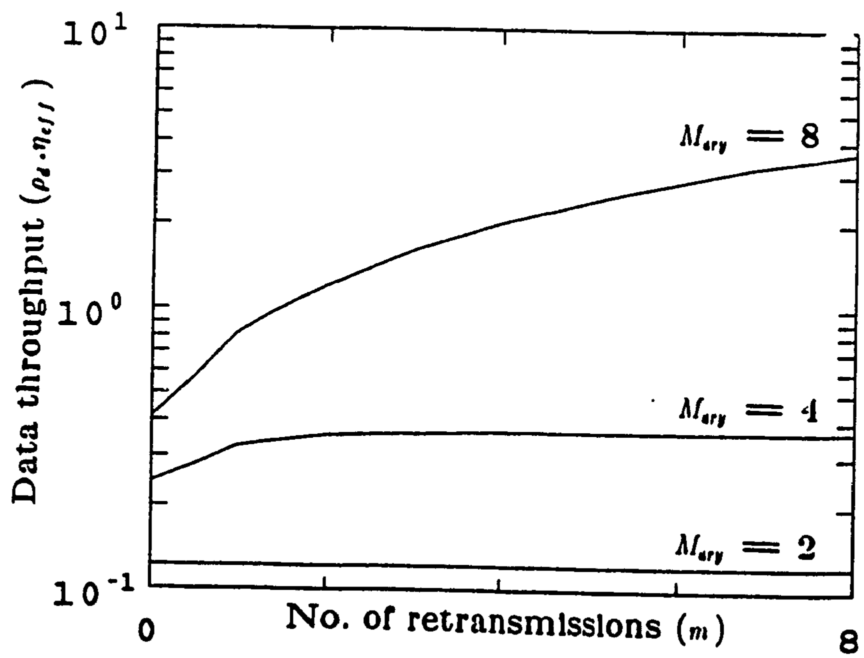



**Fig. 3.6b. Effect of packet retransmissions on data load $\rho_d$ for PAM using Multiple Token release (SNR = 45, $\rho_v = 0.2$).**

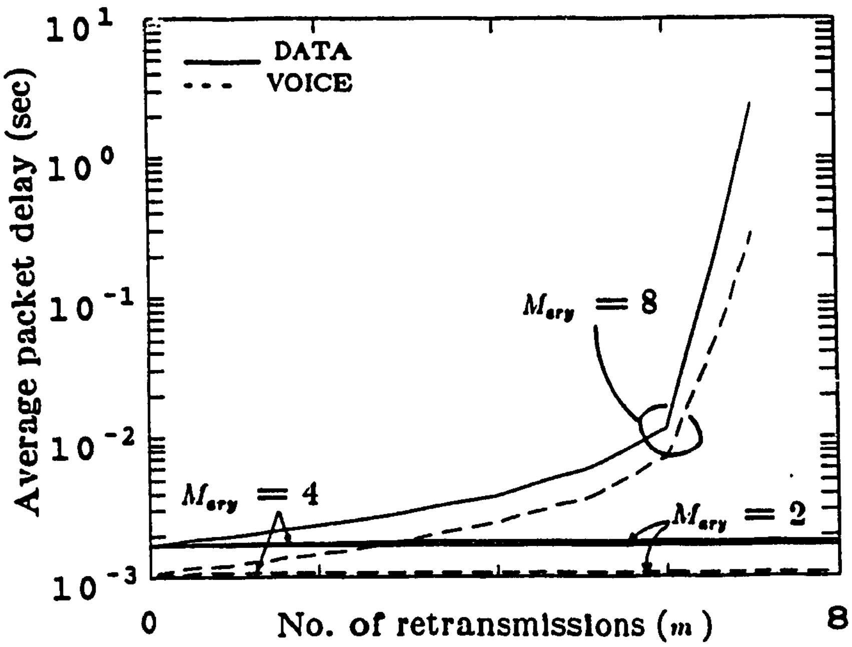



Fig. 3.6c. Effect of packet retransmissions on packet delay for PAM using Single Token release (SNR = 45, $\rho_v = 0.2$).

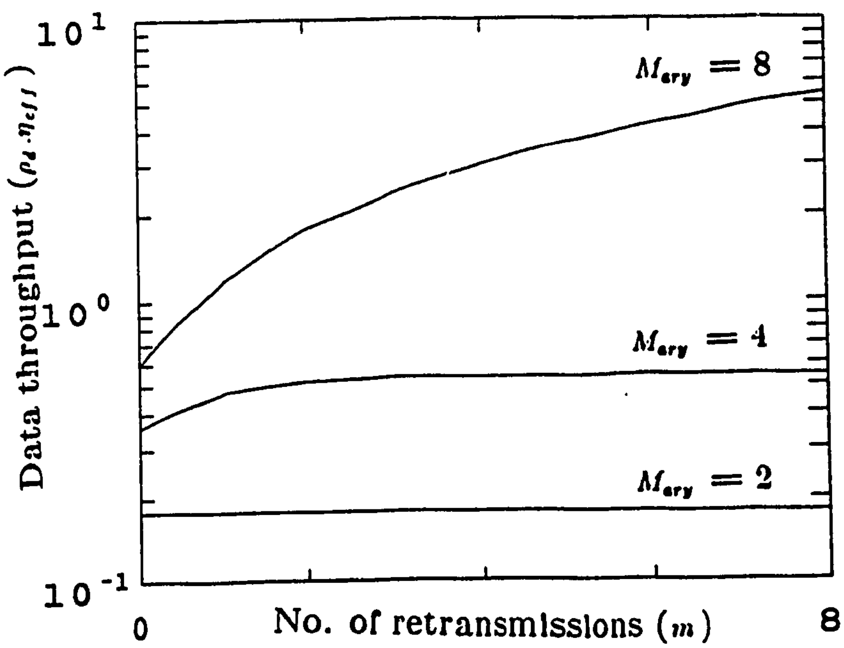



Fig. 3.6d. Effect of packet retransmissions on data load $\rho_d$ for modified duobinary using Single Token release (SNR = 45, $\rho_v = 0.2$).

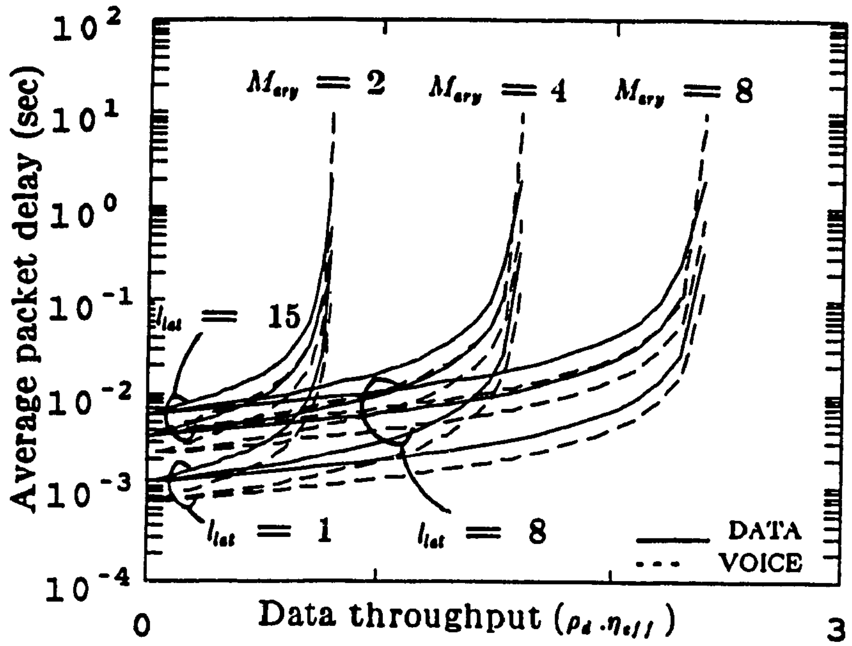



**Fig. 3.7a. Effect of different station latencies on delay for Multiple Token release scheme using PAM (SNR = 100, $M_{ary}$ = 2, 4 and 8, m = 0).**

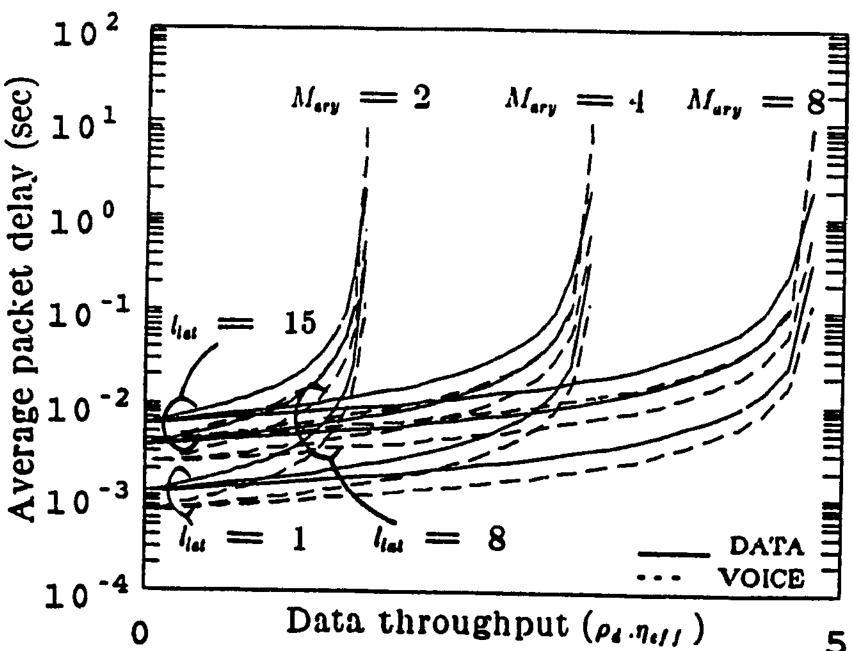



**Fig. 3.7b. Effect of different station latencies on delay for Multiple Token release scheme using duobinary signaling (SNR = 100, $M_{ary}$ = 2, 4 and 8, m = 0).**

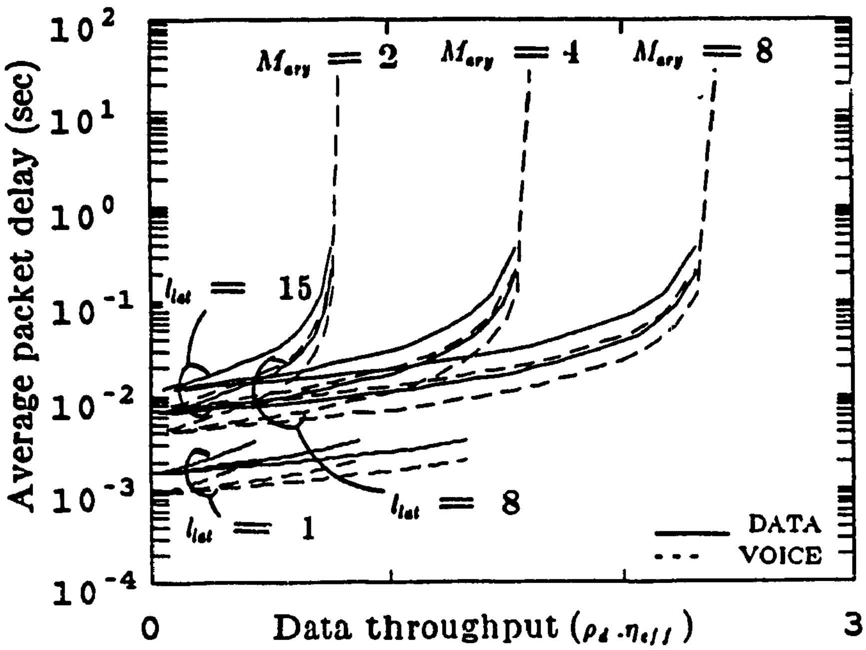



**Fig. 3.7c. Effect of different station latencies on delay for Single Token release scheme using PAM (SNR = 100, $M_{ary}$ = 2, 4 and 8, m = 0).**

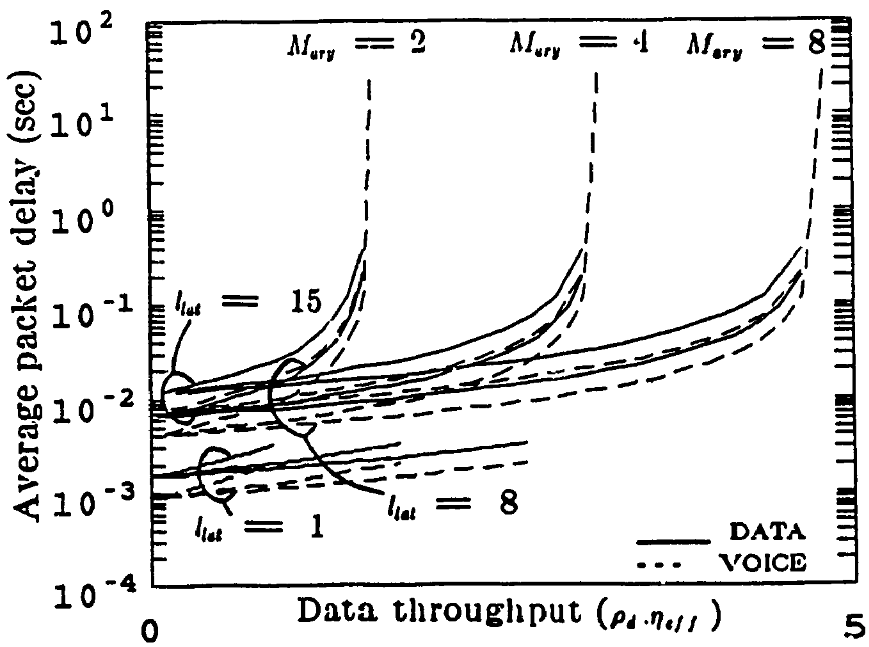



**Fig. 3.7d. Effect of different station latencies on delay for Single Token release scheme using duobinary signaling (SNR = 100, $M_{ary}$ = 2, 4 and 8, m = 0).**

# Chapter Four
## Summary and Conclusions.

### 4.1 Summary and Conclusions.

Exchange of data between different computers or between computers and other interfacing devices and peripherals, has become an integral part of any computer network. This need for communication has led to the creation of local area networks and they now play a strategic role in their daily functioning. Of the different LAN configurations, the ring network is emerging as the choice configuration for network designers. It has the ability to provide high bandwidth point-to-point connections, especially when using optical fiber technology, while yielding a predictable and bounded performance figures. Various techniques to enhance its reliability, as all traffic has to transit through every station, have been developed and include the double ring configuration with its self-healing recovery procedures.

Three of the most representative medium accessing techniques developed for ring networks have been covered. They are namely the token ring, the slotted ring and the register insertion ring. The token ring is by far the most popular and mainly uses the IEEE 802.5 standards. It provides the best overall performance with respect to delay and throughput under various loads, but works best at low loads and with small station latencies. An upgraded version of the IEEE 802.5 ring capable of handling a much higher rate (100 Mbps as compared to 4 Mbps) is being developed and is known as FDDI standards. The FDDI optimizes the token-passing technique to provide increased performance to maximize the bit rate by using a set of timers.

The slotted ring, typified by the Cambridge ring, is very popular in rings where the propagation delay is long and where the number of stations is large. An intelligent choice of the size of the minipacket to minimize its overhead, will provide the best performance.

Lastly the register insertion ring, as exemplified by the DLCN loop, provides almost immediate access to the ring. It works best when the number of stations is small or when the load is very low for a fixed number of stations. An excessive number of users or a high load will tend to extend the logical ring prohibitively, thus yielding untolerable delays in packet delivery.

The proliferation of ring networks has led to its exposure to different environment. In an office environment, token rings have to compete for bandwidth about the 1 MHz region as their rate is limited to 4 Mbps, as specified by the IEEE 802.5 standards. Thus a higher bit rate of 16 Mbps is welcomed as it avoids the congested region and also satisfy future projected needs and applications. However bandwidth reduction is necessary as it would improve the efficient use of bandwidth as well as increase the reach of the transmission medium. Hence the need for an adapter to compress the bandwidth is required.

The adapter converts the Manchester signal into a modified duobinary signal, thus reducing the required bandwidth by two. As the adapter consists of a transmitter and a receiver, its use is not confined to token rings only but can be extended to any point-to-point links which is carrying a 16 Mbps Manchester signal. The adapter has been implemented with the cost of the final product in mind. Clearly such a design does not represent the sole way to perform the proposed bandwidth reduction, nor is the best way to do it. It merely confirms the feasibility of such an operation with minimum cost ( < 100 dollars).

Several improvement can be made to both the transmitter and the receiver. For example, in the transmitter, the initial comparator can be replaced by a TTL gate or a clipping transistor. An analog summer could have been used instead of the multiplexer to generate the modified duobinary (MD) signal. A sharper filter could have been used to remove more residual harmonics. However all these possibilities have their own advantages and drawbacks too. The final choice was more often dictated by the availability of the components at the time of the design.

Similarly in the receiver, smaller delays, meaning finer tuning of the sampling instant, could have been used without any major change as the multiplexer used can choose one out eight delays instead of the presently used four. However more delays imply shorter response time, and may result to instability due to the sensitive nature of the feedback in the adaptive sampler. Hence good synchronization of the operations cannot be guaranteed. Moreover, more samples can be taken from the signal to provide a more accurate decision on the nature of the bit. Unfortunately, increasing the number of samples will increase the number of combinations dramatically ($2^n$), as well as the complexity of the decoding combinational circuit. Moreover, even with the present circuit, the truth table used to interpret the samples, has been chosen by educated guess and experience. Thus the decoder may not be the optimum decoding circuit. The number of permutations with the $2^2$ possible output from the DATA and the $\overline{\overline{\mathrm{ENB}}}$ signals with the 5 samples taken, is prohibitively too large ($4^{32}$) to test as each will generate a combinational circuit. Only a computer simulation could resolve this problem, with a lot of CPU time and some luck. Thus because of the logic behind the choice of the output in the present table, it is reasonable to say that it is not too far from the optimum one. A search for a better combination in the number of delays used and the number of samples taken may be more fruitful.

The addition of an equalizer and a filter at the input of the receiver can improve the signal and hence reduce the probability of bad decoding. However both will add complexity and cost to the circuit. Moreover, because of the nature of the channel used, the gain by fixed equalizer is minimal. Adaptive equalizers are too expensive for this low cost adapter. Thus, for these reasons and due to the distance used (~ 200 m), the equalizer and filter were deemed unnecessary.

The performance analysis of the token ring using such an adapter investigates the possibility of using higher $M_{ary}$ signaling to further increase the bandwidth efficiency and also compares the performance of the $M_{ary}$ PAM signaling with the modified duobinary signaling. Two schemes of token release, namely the Single Token (ST) and the Multiple Token (MT) release schemes, are considered. The effect of retransmissions on performance is also investigated as well as the effect of station latency.

The analysis shows that the higher $M_{ary}$ signaling is feasible only if a very high SNR is available to outweigh the probability of incorrect reception and decoding. Hence, it can be safely concluded that higher order partial response signaling is not a very viable option as the probability of error is high, and thus leads to a lot of retransmissions, causing an increase in the traffic offered. On the other hand, the performance of modified duobinary signaling can be rightly compared to 4-ary PAM signaling as they both exhibit similar characteristics because of their bandwidth gain. Throughout the analysis, the superiority of the MT release scheme is obvious. It is due to the more efficient handling of the token at each station as the token is released immediately after the transmission of the packet instead of waiting for the header as in the case of the ST scheme. Retransmission due to error can be performed using either signaling schemes. Although the modified duobinary signaling has a higher probability of error, it does not show a marked performance difference as its bandwidth efficiency partly compensates for the increase in traffic due retransmissions. However, voice packets cannot be retransmitted in either case due to the time limitation imposed on the delay experienced. Lastly, it can be seen that increasing the station latency deteriorates the performance of the ring. This is expected as the token ring is very susceptible to station latencies. However, modified duobinary signaling is slightly less affected as it can cope with the increased traffic due to longer waiting time for the server because of its more efficient bandwidth use.

# *References*


[1] W. Stallings, Handbook of Computer : Communications Standards and Local network Standards, vol. 2, MacMillan Publ. Co., New York, 1987.

[2] W.R. Franta and I. Chlamta, Local Networks - Motivation, Technology, and Performance, D.C. Heath and Co., Lexington, 1981.

[3] J.M. Mc Quillan and D.C. Walden, 'The ARPA network design decisions', Computer Networks, vol. 1, Aug. 1977, pp. 243-289.

[4] International Organization for Standardization, Open Systems Interconnection - Basic Reference Model, ISO 7498, 1984.

[5] The Institute of Electrical and Electronics Engineers, Overview, Interworking and Systems Management, IEEE Std. 802.1, 1986.

[6] International Organization for Standardization, Logical Link Control, DIS 8802/2, 1987.

[7] International Organization for Standardization, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, DIS 8802/3, 1987.

[8] International Organization for Standardization, Token-Passing Bus Access Method and Physical Layer Specifications, DIS 8802/4, 1986.

[9] International Organization for Standardization, Token Ring Access Method and Physical Layer Specifications, DIS 8802/5, 1987.

[10] The Institute of Electrical and Electronics Engineers, Logical Link Control, American National Standard ANSI/IEEE Std. 802.2, 1985.

[11] The Institute of Electrical and Electronics Engineers, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, American National Standard ANSI/IEEE Std. 802.3, 1985.

[12] The Institute of Electrical and Electronics Engineers, Token-Passing Bus Access Method and Physical layer Specifications, American National Standard ANSI/IEEE Std. 802.4, 1985.

[13] The Institute of Electrical and Electronics Engineers, Token Ring Access Method and Physical Layer Specifications, American National Standard ANSI/IEEE Std. 802.5, 1985.

[14] J.F. Shoch, An annoted bibliography on Local Computer Networks, Xerox, 1980.

[15] I.W. Cotton, 'Technologies for local area computer networks', Proc. Office Automation Conf., Mar. 1980, pp. 183-291.

[16] D. Way, 'Build a local network on proven software', Data Communications, Dec. 1981, pp. 70-73.

[17] K.J. Thurber and H.A. Freeman, 'The many faces of local networking', Data Communications, Dec. 1981, pp. 62-70.

[18] B.K. Penney and A.A. Baghdadhi, 'Survey of computer communications loop network (part 1 and 2)', Comp. Commun., vol. 2, no.4, Aug. 1979, pp. 165-180, and vol. 2, no. 5, Oct. 1979, pp. 224-241.

[19] Y.T. Tang and S.G. Zaky, 'A survey of ring networks: topology and protocols', IEEE INFOCOM, 1985, pp. 318-325.

[20] F.E. Ross, 'Rings are 'round for good', IEEE Network Mag., vol. 1, no. 1, Jan. 1987, pp. 31-38.

[21] J.R. Brandsma, A.A.M.L. Bruekers and J.L.W. Kessels, 'Philan: a fiber-optic ring for voice and data', IEEE Comm. Mag., vol. 24, no. 12, Dec. 1986, pp. 16-22.

[22] L.M. Casey, R.C. Dittburner and N.D. Gamage, 'FXNET: a backbone ring for voice and data', IEEE Comm. Mag., vol. 24, no. 12, Dec. 1986, pp. 23-28.

[23] E.R. Hafner and Z. Nenadal, 'Enhancing the availability of a loop system by meshing', Proc. Int. Zurich Seminar on Digital Communications', Zurich, Switzerland, Mar. 1976, pp. D4(1-5).

[24] C.S. Raghavendra, M. Gerla and A. Avizienis, 'Reliable loop topologies for large computer networks', IEEE Trans. on Comput., vol. C-34, Jan. 1985, pp. 46-55.

[25] B. Jackson et al, 'SILK: an integrated voice and data system', Local Networks and Distributed Office Systems, Online Conferences, pp. 47-63.

[26] C.S. Raghavendra and M. Gerla, 'Optimal loop technologies for distributed systems', Proc. 7th Data Commun. Symp. ACM SIGCOMM, vol. 11, no. 4, Oct. 1981, pp. 218-223.

[27] J.H. Saltzer and K.T. Pogran, 'A star-shaped ring network with high maintanability', Proc. Local Area Network Symp., May 1979, pp. 179-189.

[28] N.C. Strole, 'The IBM token ring network: A functional overview', IEEE Network Mag., vol. 1, no. 1, Jan. 1987, pp. 23-30.

[29] P. Zafiropoulo, 'Performance evaluation of reliability improvement techniques for single-loop communications systems', IEEE Trans. on Commun., vol. COM-22, June 1974, pp. 742-751.

[30] M.T. Lui M.T. et al, 'Design of the Distributed Double Loop Computer Network (DDLCN)', J. Digital Syst., vol. 5, Spring 1981, pp. 3-37.

[31] R. Rom and N. Shacham, 'A reconfiguration algorithm for a double loop token ring local area network', IEEE Trans. on Comput., vol. C-37, no. 2, Feb. 1988, pp. 182-189.

[32] D.J. Paulish, 'The Exploratory System control Model (ESM) multiloop network', NCC '79, pp. 821-826, 1979.

[33] R.M. Newman, Z.L. Budrikis and J.L Hulett, 'The QPSX Man', IEEE Comm. Mag., vol. 26, no. 4, April 1988, pp. 20-28.

[34] Y.J. Tang and S.G. Zaky, 'A unified queueing model for performance evaluation of local ring networks', IEEE INFOCOM, 1986, pp. 556-563.

[35] E.H. Steward, 'A loop transmisson system', Proc. IEEE $6^{th}$ Int. Conf. on Commun., vol. 2, June 1970, pp 36.1-36.9.

[36] D.R. Weller, 'A loop communication system for I/O to a small multi-user computer', Proc. $5^{th}$ Annl. IEEE Int. Conf. on Hardware, Software, Firmware and Tradeoffs, Sept. 1971, pp. 49-50.

[37] J.R. Pierce, 'Network for block switching of data', Bell Syst. Tech. Jour., vol. 51, no. 6, July/Aug. 1972, pp. 1133-1145.

[38] A.G. Fraser, 'A virtual channnel network', Datamation, vol. 21, no. 2, Feb. 1975, pp. 51-56.

[39] T.E Hassing et al, 'A loop network for general purpose data communications in a heterogeneous world. Data networks: Analysis and Design', Third Data Communications Symp., Nov. 1973.

[40] A. Hopper, 'Data ring at computer laboratory, University of Cambridge', U. S. National Bureau of Standards Special Publication, no. 500-31, Aug. 1977.

[41] K. Yada et al, 'The large scale integrated service local network using optical fiber data highway', Proc. IEEE 1981 Computer Networking Symp., Dec 1981, pp. 18-23.

[42] J.L. Favre, 'CARTHAGE: A multiservice local network on a fiber optics loop', Proc. of the IFIP TC 6 Intl. In-Depth Symp. on Local Computer Networks, April 1982, pp. 23-37.

[43] J.A. Davis et al, 'A local network for experiment support', Proc. of the National Electronics Conf., vol. 36, Oct. 1982, pp. 356-362.

[44] C.C Reames and M.T. Lui, 'A loop network for simultaneous transmission of of variable length messages', Proc. ACM SIGARCH, $2^{nd}$ Annl. Symp. on Computer Architecture, Jan. 1975, pp. 7-12.

[45] J.J. Wolf and M.T. Liu, 'A distributed double loop computer network (DDLCN)', Proc. $7^{th}$ Texas Conf. on Comp. Syst., Nov. 1978, pp. 6.19-6.34.

[46] Z.G. Vrasenic et al, 'TORNET: A local area network', Proc. $7^{th}$ Data Commun. Symp., Oct. 1981, pp. 180-187.

[47] P.V. Mockapetris, M.R. Lyle and D.J. Farber, 'On the design of local network interfaces', Proc. IFTP Congress 77, Aug. 1977, pp. 427-430.

[48] R.L. Gordon, W.W. Farr and P. Levine, 'Ringnet: A packet switched local network with decentralized control', $4^{th}$ Conf. on Local Computer Networks, Oct. 1979, pp. 13-19.

[49] W. Bux et al, 'A local area communication network based on a reliable token-ring system', Proc. of the IFIP TC 6 Intl. In-Depth Symp. on Local Computer Networks, April 1982, pp. 69-82.

[50] H. Ikuta et al, 'Role of optical fiber loop in C & C OPTONET', IEEE COMPCON 1982, Sept. 1982, pp. 471-477.

[51] J.M. Ulm, 'A timed token ring local area network and its performance characteristics', $7^{th}$ Conf. on Local Computer Networks, Oct. 1982, pp. 50-56.

[52] W.D. Farmer and E.E. Newhall, ' An experimental distributed switching system to handle bursty computer traffic', Proc. ACM Symp. on Problems in the Optimization of Data Communications Systems, Oct. 1969, pp. 1-33.

[53] M. Hatada, K. Hiyama and H. Ihara, 'A microprocessor-based multi-loop network system', IEEE COMPCON 1979, Sept. 1979, pp. 454-461.

[54] E. Aramis et al, 'Orion: A new distributed loop computer network', IEEE Proc. 1979 Computer Networking Symp., Dec. 1979, pp. 164-168.

[55] C.R. Dhas and D.C. Rine, 'Design of a ring-based local area network for microcomputers', $5^{th}$ Conf. on Local Computer Networks, Oct. 1980, pp. 65-68.

[56] E. Enrico, 'The European approach to LAN standardization', IEEE INFOCOM, Miami, Fl., 1986, pp. 343-354.

[57] ECMA, Standard ECMA-89: Local Area Networks - Token Ring Technique, Sept. 1982.

[58] ANSI, 'FDDI token ring Media Access Control (MAC)', American National Standard, X3.139-1987.

[59] ANSI, 'FDDI token ring Physical Layer Protocol (PHY)', Draft proposed American National Standard, X3.148-198X.

[60] ANSI, 'FDDI token ring Physical layer Medium Dependent (PMD), Draft proposed American National Standard, X3.166-198X.

[61] ANSI, 'FDDI token ring Station Management (SMT)', Draft proposed American National Standard, X3T9/85, X3T9.5/84-49, REV. 3.0, AUG. 1987.

[62] H. Takagi, Analysis of polling systems, MIT Press, 1986.

[63] A.G. Konheim and B. Meister, 'Waiting lines and times in a system with a system with polling', Jour. Ass. Comp. Mach., vol. 21, 1974, pp.470-490.

[64] G.B. Swartz, 'Polling in a loop system', Jour. Ass. Comp. Mach., vol. 27, 1980, pp. 42-59.

[65] I. Rubin and L.F.M. DeMoreas, 'Messages delay analysis for polling and token multiple-access schemes for local communication networks', J. Sel. Areas in Commun., vol. SAC-1, 1983, pp. 935-947.

[66] M. Eisenberg, 'Queues with periodic service and changeover times', Oper. Res., vol. 20., 1972, pp. 935-947.

[67] M.J. Ferguson and Y.J. Aminetzah, 'Exact results for non-symmetric token ring systems', IEEE Trans. on Commun., vol. COM-33, 1985, pp. 223-231.

[68] H. Takagi, 'Mean average waiting time in symmetric multiqueue systems with cyclic service', Proc. Int. Conf. on Commun., Chicago, 1985. pp. 1154-1157.

[69] K.S. Watson, 'Performance evaluation of cyclic service strategies - A survey', Performance '84, (Ed. E. Gelenbe), Elsevier Science Publishers B.V. (North Holland), 1984, pp. 521-533.

[70] W. Bux and H.L. Truong, 'Mean-delay approximation for cyclic service queueing systems', Perf. Eval., vol. 3, 1983, pp. 187-196.

[71] A.S. Sethi and T. Saydam, 'Performance analysis of token ring local area networks', Computer Networks and ISDN Systems, vol. 9, 1985, pp. 191-200.

[72] M.E. Ulug, 'Calculations of waiting times for a real time token passing bus', Proc. IEEE Comp. Networking Symp., 1983, pp.

[73] W. Bux, 'Local area subnetworks: A performance comparison', IEEE Trans. on Commun., vol. COM-29, no. 10, 1981, pp. 1465-1473.

[74] J.L. Hammond and J.P. O'Reilly, Performance analysis of Local Computer Networks, Addison-Wesley, 1986.

[75] K.C. Sevcik and M.J. Johnson, 'Cycle time properties of the FDDI token ring protocol', IEEE Trans. on Soft. Eng., vol. SE-13, no. 3, March 1987, pp. 376-385.

[76] M.J. Johnson, 'Proof that timing requirements of the FDDI token ring protocol are satisfied', IEEE Trans. on Commun., vol. COM-35, no. 5, June 1987, pp. 620-625.

[77] M.V. Wilkes and D.J. Wheeler, 'The Cambridge Digital Communication Ring', Proc. Local Area Communications Network Symposium (Eds. N.B. Meinner and R. Rosenthal), Boston, May 1979, pp. 47-61.

[78] A. Hopper and D.J. Wheeler, 'Maintenance of ring communication systems', IEEE Trans. on Commun., vol. COM-27, no. 4, 1979, pp. 760-761.

[79] R.M. Needham and A.J. Herbert, 'The Cambridge Distributed Computing System', Addison-Wesley, London, 1982.

[80] E.B. Spratt, 'Operational experiences with a Cambridge ring local area network in a university environment', Local Networks for Computer Communications (Eds. A. West and P. Janson), North-Holland, 1981, pp. 81-106.

[81] P.T. Kirstein et al, 'The Universe Project', Pathways to the Information Society (Ed. M.B. Williams), North-Holland, 1982, pp. 442-447.

[82] R.D.H. Walker, 'Basic ring transport protocol', Project Note, Computer Laboratory, University of Cambridge, Nov. 1978.

[83] N.J. Ody, 'A single-shot protocol', Project Note, Computer Laboratory, University of Cambridge, April 1979.

[84] M.A. Johnson, 'Ring byte stream protocol specifications', Project Note, Computer Laboratory, University of Cambridge, April 1980.

[85] A. Hopper, S. Temple and R. Williamson, Local Area Network Design, Addison-Wesley, 1986.

[86] A. Hopper and R.C. Williamson, 'Design and use of an integrated Cambridge ring', J. Sel. Areas in Commun., vol. SAC-1, no. 5, Nov. 1983, pp. 775-784.

[87] A. Hopper and R.M. Needham, 'The Cambridge Fast Ring Networking System', IEEE Trans. on Comput., vol. 37, no. 10, Oct. 1988, pp. 1214-1223.

[88] J.F. Hayes and D.N. Sherman, 'Traffic analysis of a ring switched data transmission system', Bell Syst. Tech. Jour., vol. 50, no. 9, 1971, pp. 2947-2977.

[89] Anderson et al, 'Simulated performance of a ring-switched data network', IEEE Trans. on Commun., vol. COM-20, no. 3, 1972, pp. 576-591.

[90] J.D. Spragins, 'Loop transmission systems - Mean value analysis', IEEE Trans. on Commun., vol. COM-20, no. 3, 1972, pp. 592-602.

[91] L. Kleinrock, 'Time-shared systems: A theoretical treatment', Jour. Ass. Comp. Mach., vol. 14, 1967, pp. 242-261.

[92] F. Baskett et al, 'Open, closed, and mixed networks of queues for different classes of customer', Jour. Ass. Comp. Mach., vol. 22, 1975, pp.248-260.

[93] E. Gelenbe and I. Mitrani, Analysis and Synthesis of computer systems, Academic Press, London, England, 1980.

[94] P.J.B. King and I. Mitrani, 'Modeling a slotted ring local-area network', IEEE Trans. on Comput., vol. C-36, no. 5, 1987, pp. 554-561.

[95] S.A. Sorensen, 'Cambridge Ring performance', Computer Networks and ISDN systems, vol. 9, 1985, pp. 345-352.

[96] E.R. Hafner, Z. Nenadal and M. Tshanz, 'A digital loop communications system', Proc. Int. Conf. on Commun., Seattle, June 1973, pp. 50.24-50.29.

[97] D.E Huber, W. Steinlin and P. Wild, 'SILK: An implementation of a buffer insertion ring', J. Sel. Areas in Commun., vol. SAC-1, no. 5, 1983, pp. 766-773.

[98] Oh Young and M.T. Liu, 'Interface design for Distributed Control Loop Networks', Proc. Nat. Telecom. Conf., 1977, pp. 31:4-1,31:4-6.

[99] M.T. Lui, G. Babic and R. Pardo, 'Traffic analysis of the Distributed Loop Control Computer Networks', Proc. Nat. Telecom. Conf., 1977, pp. 31:5-1, 31:5-7.

[100] A. Thomasian and H. Kanakia, 'Performance study of loop networks using buffer insertion', Computer Networks, vol. 3, 1979, pp. 419-425.

[101] W. Bu · and M. Schlatter, 'An approximate method for the performance analysis of buffer insertion rings', IEEE Trans. on Commun., vol. COM-31, no. 1, 1983, pp. 50-55.

[102] W. Hilal and M.T. Liu, 'Analysis and simulation of the register insertion protocol', Proc. Comp. Net. Symp., 1982, pp. 91-100.

[103] W. Hilal, J.F. Chiu and M.T. Lui, 'A study of register insertion rings', IEEE INFOCOM, 1987, pp. 479-491.

[104] L. Kleinrock, Queueing Systems, vol. II, Wiley Interscience, 1976.

[105] I. Rubin, 'Message path delays in packet-switching communication networks', IEEE Trans. on Commun., vol. COM-23, no. 2, 1975, pp. 186-192.

[106] I. Rubin, 'An approximate time delay analysis for packet-switching communication networks', IEEE Trans. on Commun., vol. COM-24, no. 2, 1976, pp. 210-221.

[107] P. Kermani and L. Kleinrock, 'Virtual cut-through: A new computer communication switching technique', Computer Networks, vol. 3, 1979, pp. 267-286.

[108] L.N. Bhuyan, D. Ghosal and Q. Yang, 'Approximate analysis of single and multiple ring networks', IEEE Trans. on Comput., vol. 38, no. 7, July 1989, pp. 1027-1040.

[109] F. Daaboul, 'Effect of impulse noise on the performance of 16Mbits/s systems', Contribution to IEEE 802.9-88, Nov. 7,1988.

[110] H. Nyquist, 'Certain topics in telegraph transmisssion theory', Trans. AIEE, vol. 47, April 1928, pp.617-644.

[111] S. Pasapurthy, 'Correlative coding : A bandwidth efficient signaling scheme', IEEE Commun. Soc. Mag., vol. 15, no. 4, July 1977, pp. 4-11.

[112] A. Lender, 'The duobinary technique for high-speed data transmission', IEEE Trans. on Commun. and Electro., vol. 82, May 1963, pp. 214-218.

[113] E.R. Kretzmer, 'Generalization of a technique for binary data communication', IEEE Trans. on Commun. Technol., vol. COM-14, Feb. 1966, pp.67-68.

[114] J.G. Proakis, Digital Communications, Mc Graw Hill, New York, 1983.

[115] M. Schwartz, Information Transmission, Modulation and Noise, Mc Graw Hill, New York, 1980.

[116] A.B. Williams, Electronic Filter Design, Mc Graw Hill, New York, 1981.

[117] D. Karvelas and A. Leon-Garcia, 'Performance of intergrated packet voice/data token passing rings', J. Sel. Areas in Commun., vol. SAC-4, no. 6, Sept. 1986, pp. 823-832.

[118] P.J. Kuehn, 'Multiqueue systems with non-exhaustive cyclic service', Bell Syst. Tech. Jour., vol. 58, Mar. 1979, pp. 671-698.

[119] L. Kleinrock, Queueing Systems : Theory, vol. 1, John Wiley, New York, 1975.

[120] Y.C. Jenq, 'Performance of a packet voice multiplexer', IEEE INFOCOM, Apr. 1984, pp. 256-259.

# Appendix

The first and second moments of the conditional cycle times of the ring can be found by the differentiation of the Laplace-Stieltjes transforms and evaluating them at $s = 0$. Hence for the conditional cycle time of the idle period, the following equation can be written :

$$C_o(s) = U_o(s)\prod_{i \neq j}\left[p_{vo}(i)H_v(s) + p_{do}(i)H_d(s) + p_{oo}(i)\right] \tag{A1}$$

or

$$C_o(s) = U_o(s)\prod_{i \neq j}\alpha_o(s,i) \tag{A2}$$

where $\alpha_o(s,i) = p_{vo}(i)H_v(s) + p_{do}(i)H_d(s) + p_{oo}(i)$. The probability of arrivals for voice or data packets in the idle cycle is given as $p_{vo}(i) = A_v(i)\bar{c}_o$ and $p_{vd}(i) = A_d(i)\bar{c}_o$, respectively. Differentiating (A2),

$$\frac{\partial C_o(s)}{\partial s} = \frac{\partial U_o(s)}{\partial s}\prod_{i \neq j}\alpha_o(s,i) + U_o(s)\sum_{k \neq j}\frac{\partial \alpha_o(s,k)}{\partial s}\prod_{i \neq j \neq k}\alpha_o(s,i) \tag{A3}$$

at $s = 0$,

$$C_o(0) = C_d(0) = C_v(0) = 1\,; \quad U_o(0) = H_d(0) = H_v(0) = 1 \tag{A4a}$$

$$\frac{\partial C_o(0)}{\partial s} = -\bar{c}_o; \quad \frac{\partial C_d(0)}{\partial s} = -\bar{c}_d; \quad \frac{\partial C_o(0)}{\partial s} = -\bar{c}_v \tag{A4b}$$

$$\frac{\partial H_d(0)}{\partial s} = -\bar{h}_d; \quad \frac{\partial H_v(0)}{\partial s} = -\bar{h}_v; \quad \frac{\partial U_o(0)}{\partial s} = -\bar{u}_o \tag{A4c}$$

$$\prod_{i \neq j}\alpha_o(0,i) = \prod_{i \neq j \neq k}\alpha_o(0,i) = 1 \tag{A4d}$$

Differentiating $\alpha_o(s,k)$ with respect to $s$,

$$\sum_{k\neq j}\frac{\partial\alpha_o(s,k)}{\partial s} = \sum_{k\neq j}\left[p_{vo}(k)\frac{\partial H_v(s)}{\partial s} + p_{do}(k)\frac{\partial H_d(s)}{\partial s}\right] \tag{A5}$$

$$\begin{aligned}\sum_{k\neq j}\frac{\partial\alpha_o(s,k)}{\partial s} &= -\sum_{k\neq j}\left[p_{vo}(k)\bar{h}_v + p_{do}(k)\bar{h}_d\right] \\ &= -\sum_{k\neq j}\left[A_v(k)\bar{c}_o\bar{h}_v + A_d(k)\bar{c}_o\bar{h}_d\right] \\ &= -\sum_{k\neq j}\left[\rho_v(j)\bar{c}_o + \rho_d(k)\bar{c}_o\right] \\ &= -\left[\rho_a - \rho_v(j) - \rho_d(j)\right]\bar{c}_o \end{aligned}\tag{A6a}$$

Hence

$$\sum_{i\neq j}\frac{\partial\alpha_o(s,i)}{\partial s} = -\bar{c}_o\left[\rho_a - \rho_v(j) - \rho_d(j)\right] \tag{A7}$$

where $\rho_a = \rho_v + \rho_d$. Evaluating equation A3 at $s = 0$, the mean conditional cycle time when no transmission is made at station $j$ is obtained.

$$-\bar{c}_o = -\bar{u}_o - \bar{c}_o\left[\rho_a - \rho_v(j) - \rho_d(j)\right]$$

$$\bar{c}_o = \frac{\bar{u}_o}{1-\rho_a + \rho_v(j) + \rho_d(j)} \tag{A8}$$

Differentiating A3 again to find its second moment,

$$\begin{aligned}\frac{\partial^2 C_o(s)}{\partial s^2} &= \frac{\partial^2 U_o(s)}{\partial s^2}\prod_{i\neq j}\alpha_o(s,i) + \frac{\partial U_o(s)}{\partial s}\sum_{k\neq j}\frac{\partial\alpha_o(s,k)}{\partial s}\prod_{i\neq j\neq k}\alpha_o(s,i) \\ &+ \frac{\partial U_o(s)}{\partial s}\sum_{k\neq j}\frac{\partial\alpha_o(s,k)}{\partial s}\prod_{i\neq k\neq j}\alpha_o(s,i) \\ &+ U_o(s)\sum_{k\neq j}\frac{\partial^2\alpha_o(s,k)}{\partial s^2}\prod_{i\neq j\neq k}\alpha_o(s,i)\end{aligned}$$

$$+ U_o(s) \sum_{k \neq j} \frac{\partial \alpha_o(s,k)}{\partial s} \sum_{l \neq k \neq j} \frac{\partial \alpha_o(s,l)}{\partial s} \prod_{i \neq j \neq k \neq l} \alpha_o(s,i) \tag{A9}$$

Now

$$\sum_{k \neq j} \frac{\partial^2 \alpha_o(s,k)}{\partial s^2} = \sum_{k \neq j} \left[ p_{vo}(k) \frac{\partial^2 H_v(s)}{\partial s^2} + p_{do}(k) \frac{\partial^2 H_d(s)}{\partial s^2} \right]$$

at $s = 0$, let

$$\frac{\partial^2 H_d(0)}{\partial s^2} = \overline{h_d^2}; \quad \frac{\partial^2 H_v(0)}{\partial s^2} = \overline{h_v^2} \tag{A10a}$$

$$\frac{\partial^2 C_d(0)}{\partial s^2} = \overline{c_d^2}; \quad \frac{\partial^2 U_o(0)}{\partial s^2} = \overline{u_o^2} \tag{A10b}$$

$$\prod_{i \neq j} \alpha_o(0,i) = \prod_{i \neq j \neq k} \alpha_o(0,i) = \prod_{i \neq j \neq k \neq l} \alpha_o(0,i) = 1 \tag{A10c}$$

Therefore

$$\begin{aligned} \sum_{k \neq j} \frac{\partial^2 \alpha_o(0,k)}{\partial s^2} &= \sum_{k \neq j} \left[ p_{vo}(k) \overline{h_v^2} + p_{do}(k) \overline{h_d^2} \right] \\ &= \sum_{k \neq j} \left[ A_v(k) \overline{h_v^2} \right] \overline{c}_o + \left[ A_d(k) \overline{h_d^2} \right] \overline{c}_o \\ &= \overline{c}_o \sum_{k \neq j} \left[ A_v(k) \overline{h_v^2} + A_d(k) \overline{h_d^2} \right] \end{aligned} \tag{A11}$$

Hence at $s = 0$,

$$\begin{aligned} \overline{c_o^2} = \overline{u_o^2} &+ 2\overline{u}_o \sum_{k \neq j} \left[ p_{vo}(k) \overline{h}_v + p_{do}(k) \overline{h}_d \right] + \sum_{k \neq j} \left[ p_{vo}(k) \overline{h_v^2} + p_{do}(k) \overline{h_d^2} \right] \\ &+ \sum_{k \neq j} \left[ p_{vo}(k) \overline{h}_v + p_{do}(k) \overline{h}_d \right] \sum_{k \neq j} \left[ p_{vo}(k) \overline{h}_v + p_{do}(k) \overline{h}_d \right] \end{aligned}$$

The second moment of the idle cycle is given by

$$\begin{aligned} \overline{c_o^2} = \overline{u_o^2} &+ 2\overline{u}_o \overline{c}_o \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] + \overline{c}_o \sum_{k \neq j} \left[ A_v(k) \overline{h_v^2} + A_d(k) \overline{h_d^2} \right] \\ &+ \left[ \overline{c}_o \right]^2 \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] \left[ \rho_a - \rho_v(j) - \rho_d(j) - \rho_v(k) - \rho_d(k) \right] \end{aligned} \tag{A12}$$

Assuming symmetric load to all stations on the ring, A12 can be simplified to

$$\overline{c_o^2} = \overline{u_o^2} + 2\bar{u}_o\bar{c}_o\left[\rho_a - \rho_v(j) - \rho_d(j)\right] + \bar{c}_o\sum_{k\neq j}\left[A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2}\right]$$
$$+ \left[\bar{c}_o\right]^2\left[\rho_a - \rho_v(j) - \rho_d(j)\right]\left[\rho_a - 2\rho_v(j) - 2\rho_d(j)\right] \tag{A13}$$

Similarly, the conditional time for the data cycle can be found. The equilibrium equation for the data cycle is given as

$$C_d(s) = U_o(s)\prod_{i\neq j}\left[p_{vd}(i)H_v(s) + p_{dd}(i)H_d(s) + p_{od}(i)\right]H_d(s) \tag{A14}$$

or

$$C_d(s) = U_o(s)H_d(s)\prod_{i\neq j}\alpha_d(s,i) \tag{A15}$$

Differentiating A14 with respect to $s$,

$$\frac{\partial C_d(s)}{\partial s} = \frac{\partial U_o(s)}{\partial s}H_d(s)\prod_{i\neq j}\alpha_d(s,i) + U_o(s)\frac{\partial H_d(s)}{\partial s}\prod_{i\neq j}\alpha_d(s,i)$$
$$+ U_o(s)H_d(s)\sum_{k\neq j}\frac{\partial \alpha_d(s,k)}{\partial s}\prod_{i\neq j\neq k}\alpha_d(s,i) \tag{A16}$$

At $s = 0$, and with A4,

$$H_d(0) = 1\,; \quad \prod_{i\neq j}\alpha_d(0,i) = 1 \tag{A17a}$$

$$\frac{\partial C_d(0)}{\partial s} = -\bar{c}_d; \quad \frac{\partial U_o(0)}{\partial s} = -\bar{u}_o; \tag{A17b}$$

$$\sum_{k\neq j}\frac{\partial \alpha_d(0,k)}{\partial s} = -\bar{c}_d\left[\rho_a - \rho_v(j) - \rho_d(j)\right] \tag{A17c}$$

Hence evaluating A16 at $s = 0$, the mean of the conditional data cycle is given by

$$-\bar{c}_d = -\bar{u}_o - \bar{h}_d - \bar{c}_d\left[\rho_a - \rho_v(j) - \rho_d(j)\right]$$

$$\bar{c}_d = \frac{\bar{u}_o + \bar{h}_d}{1 - \rho_a - \rho_v(j) - \rho_d(j)} \tag{A18}$$

Differentiating A16 again to find its second moment,

$$
\begin{aligned}
\frac{\partial^2 C_d(s)}{\partial s^2} = {} & \frac{\partial^2 U_o(s)}{\partial s^2} H_d(s) \prod_{i \neq j} \alpha_d(s,i) + \frac{\partial U_o(s)}{\partial s} \frac{\partial H_d(s)}{\partial s} \prod_{i \neq j} \alpha_d(s,i) \\
& + \frac{\partial U_o(s)}{\partial s} H_d(s) \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \prod_{i \neq j \neq k} \alpha_d(s,i) \\
& + \frac{\partial U_o(s)}{\partial s} \frac{\partial H_d(s)}{\partial s} \prod_{i \neq j} \alpha_d(s,i) + U_o(s) \frac{\partial^2 H_d(s)}{\partial s^2} \prod_{i \neq j} \alpha_d(s,i) \\
& + U_o(s) \frac{\partial H_d(s)}{\partial s} \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \prod_{i \neq j \neq k} \alpha_d(s,i) \\
& + \frac{\partial U_o(s)}{\partial s} H_d(s) \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \prod_{i \neq j \neq k} \alpha_d(s,i) \\
& + U_o(s) \frac{\partial H_d(s)}{\partial s} \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \prod_{i \neq j \neq k} \alpha_d(s,i) \\
& + U_o(s) H_d(s) \sum_{k \neq j} \frac{\partial^2 \alpha_d(s,k)}{\partial s^2} \prod_{i \neq j \neq k} \alpha_d(s,i) \\
& + U_o(s) H_d(s) \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \sum_{k \neq j} \frac{\partial \alpha_d(s,k)}{\partial s} \sum_{l \neq k \neq j} \frac{\partial \alpha_d(s,l)}{\partial s} \prod_{m \neq l \neq k \neq j} \alpha_d(s,m)
\end{aligned}
\tag{A19}
$$

At $s = 0$, let

$$\frac{\partial^2 C_d(0)}{\partial s^2} = \overline{c_d^2}; \quad \frac{\partial^2 U_o(0)}{\partial s^2} = \overline{u_o^2} \tag{A20a}$$

$$\frac{\partial^2 H_d(0)}{\partial s^2} = \overline{h_d^2}; \quad \frac{\partial^2 H_v(0)}{\partial s^2} = \overline{h_v^2} \tag{A20b}$$

$$\sum_{k \neq j} \frac{\partial^2 \alpha_d(0,i)}{\partial s^2} = \sum_{k \neq j} \left[ p_{vd}(k) \frac{\partial^2 H_v(0)}{\partial s^2} + p_{dd}(k) \frac{\partial^2 H_d(0)}{\partial s^2} \right]$$

$$= \bar{c}_d \sum_{k \neq j} \left[ A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2} \right] \tag{A20c}$$

Substituting in A19,

$$\overline{c_d^2} = \overline{u_o^2} + \bar{u}_o \bar{h}_d + \bar{u}_o \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right]$$

$$+ \bar{u}_o \bar{h}_d + \overline{h_d^2} + \bar{h}_d \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right]$$

$$+ \bar{u}_o \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] + \bar{h}_d \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right]$$

$$+ \bar{c}_d \sum_{k \neq j} \left[ A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2} \right]$$

$$+ \left[ \bar{c}_d \right]^2 \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] \left[ \rho_a - \rho_v(j) - \rho_d(j) - \rho_v(j) - \rho_d(j) \right] \tag{A21}$$

Assuming symmetric loads to all stations on the ring, A21 becomes

$$\overline{c_d^2} = \overline{u_o^2} + \overline{h_d^2} + 2\bar{u}_o \bar{h}_d + 2\bar{u}_o \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right]$$

$$+ 2\bar{h}_d \bar{c}_d \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] + \bar{c}_d \sum_{k \neq j} \left[ A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2} \right]$$

$$+ \left[ \bar{c}_d \right]^2 \left[ \rho_a - \rho_v(j) - \rho_d(j) \right] \left[ \rho_a - 2\rho_v(j) - 2\rho_d(j) \right] \tag{A22}$$

The conditional cycle time for voice can be similarly found. The same derivation as for the data cycle time applies as their Laplace-Stieltjes transform equations are similar. Recall for voice the equation is given as

$$C_v(s) = U_o(s) \prod_{i \neq j} \left[ p_{vv}(k)H_v(s) + p_{dv}(k) + p_{ov}(k) \right] H_v(s) \tag{A23}$$

Comparing A23 to A14, it can easily be verified that the first and second moments of the voice conditional cycle time are given by

$$\bar{c}_v = \frac{\bar{u}_o + \bar{h}_v}{1 - \rho_a - \rho_v(j) - \rho_d(j)} \tag{A24}$$

and

$$\overline{c_v^2} = \overline{u_o^2} + \overline{h_v^2} + 2\bar{u}_o\bar{h}_v + 2\bar{u}_o\bar{c}_v + \left[\rho_a - \rho_v(j) - \rho_d(j)\right]$$

$$+ 2\bar{h}_v\bar{c}_v\left[\rho_a - \rho_v(j) - \rho_d(j)\right] + \bar{c}_v\sum_{k\neq j}\left[A_v(k)\overline{h_v^2} + A_d(k)\overline{h_d^2}\right]$$

$$+ \left(\bar{c}_v\right)^2\left[\rho_a - \rho_v(j) - \rho_d(j)\right]\left[\rho_a - 2\rho_v(j) - 2\rho_d(j)\right] \qquad \text{(A25)}$$