UNIVERSITY OF ALBERTA

The DSMA/S Spiral-Ring MAC Protocol
for High-Speed Computer Networks

By ©

Paul A. McWeeny

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

Department of Computing Science

Edmonton, Alberta
Fall 1992

ISBN 0-315-77080-5

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Paul A. McWeeny

TITLE OF THESIS: The DSMA/S Spiral-Ring MAC Protocol
for High-Speed Computer Networks

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1992

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) Paul A. McWeeny

Permanent Address:
214-5045 Whitemud Road,
Edmonton, Alberta,
Canada

Date: October 7, 1992

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *The DSMA/S Spiral-Ring MAC Protocol for High-Speed Computer Networks* submitted by *Paul A. McWeeny* in partial fulfillment of the requirements for the degree of Master of Science.

co-supervisor: W. Dobosiewicz

co-supervisor: P. Gburzynski

examiner: J. Harms

external: B. Cockburn (Electrical Engineering)

Date: October 7, 1992

To Leslie,
for all of the encouragement, support and patience.

Abstract

A new topology and an efficient medium-access-control protocol for high-speed local and metropolitan area networks is studied. The *spiral-ring* topology is described and a slotted MAC-level protocol, called DSMA/S, is analysed.

DSMA/S is a *capacity-1* protocol: its maximum throughput is independent of network size and transmission rate. The protocol employs a token-passing mechanism to achieve fairness, but transmission is not restricted to token-possession. Thus, medium access delay is minimal under light loads, and bounded from above under heavy loads. Synchronous and asynchronous traffic are accommodated simultaneously without the need for synchronous channel allocation.

Some comparisons are made with two other well known high-speed MAC-level protocols, FDDI and DQDB. A simulated model is used to study the performance of DSMA/S under different traffic conditions.

Acknowledgements

I would like to express my thanks to my supervisors, Dr. Wlodek Dobosiewicz and Dr. Pawel Gburzynski, for their guidance in writing this thesis. The idea of a spiral-ring topology originated with them and I am grateful to them for suggesting its development as a thesis topic. Dr. Dobosiewicz was particularly helpful in guiding the analysis and simulation experiments. His many suggestions during the writing of the thesis were also greatly appreciated. In addition, I would like to thank the other members of my examination committee: Janelle Harms, Bruce Cockburn, and the chair, Joe Culberson. Their suggestions for improvement were helpful in producing the final version of the thesis. Finally, I would like to thank my wife, Leslie, for her constant support and encouragement. Her many contributions will not be forgotten.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A computer communications network provides a facility to share information and resources between computers and peripheral devices. This is achieved by an interconnection mechanism and a set of rules and conventions for exchanging data over the mechanism. The devices in the network, referred to here as *stations*, transmit and receive information by connecting to a *transmission medium*. Many network connection schemes, known as network *topologies*, exist — bus, star, ring and tree, to name a few. The rules and conventions which govern the exchange of data over the medium are known as *protocols*. Protocols can be classified by the aspects of data exchange which they govern. One class of protocols controls how the transmission medium is accessed: these are called *medium-access-control protocols* or MAC-protocols. Networks can be classified by the area which they cover: *local-area-networks* (LANs) connect a few tens of stations over a small area such as an office or small building; *metropolitan-area-networks* (MANs) can cover tens or hundreds of kilometres to connect stations in a large campus or small city; *wide-area-networks* (WANs) interconnect thousands of stations spanning whole continents. This thesis studies a new *spiral* topology and its access protocol called DSMA/S (*Distributed Spiral Multiple Access / Slotted*) suitable for large LANs or WANs.

The reader is referred to [41] and [39] for general information on computer network topologies and protocols.

1.1 Thesis Overview

The thesis is divided into seven chapters. We have already given a brief introduction to computer networks, and presented the concepts of topologies and protocols. The remainder of this chapter introduces some concepts relevant to high-speed networks and presents the notation and conventions used throughout the thesis.

Chapter 2 discusses *medium-access-control protocols*. The chapter begins with some general remarks about MAC-level protocols, then a generic *slotted* MAC-level protocol is discussed. The goals of an ideal MAC-level protocol are also included. FDDI and DQDB are briefly described as examples of MAC protocols for high-speed networks. The *spiral* topology and its characteristics are introduced in Chapter 3. Details of the DSMA/S protocol are presented in Chapter 4, where an analysis of the protocol's performance and behaviour is given.

Simulation experiments were used to study the network's behaviour under various conditions: the details are described in Chapter 5. A discussion of the simulation results is given in Chapter 6. Finally, Chapter 7 presents the conclusions of the study.

1.2 High Speed Networks

The rapid increase in numbers, capacity, and speed of computers, coupled with advances in fibre-optic transmission media has led to the emergence of *high-speed networks*. Current technology allows transmission of several gigabits[1] per second compared with tens of megabits per second a few years ago.

An important property of networks is the *normalised propagation delay*. The maximum time it takes for a unit of information to travel between two stations is the propagation delay L, measured in *bits*[2]. Let l_p be the average length (in bits) of packets transmitted on the network. The normalised propagation delay is $a = L/l_p$. Very high transmission speeds or very long networks result in large

[1] 1 gigabit is 10^9 bits of information.

[2] The *bit*, when used as a unit of time, is the amount of time required to transmit one bit of data. It is dependent on the transmission speed.

values of a, because the packet length is small relative to the propagation delay. Networks with large a values do not operate well using protocols developed for small a values. Consequently, protocols that perform well in *big-a* networks are in demand.

Protocols that are capable of using a fixed fraction of the network capacity independent of a are called *capacity-1 protocols.* The DSMA/S protocol studied in this thesis is a capacity-1 protocol designed for *big-a* networks. A number of such networks have been proposed in the literature [14], [26], [15], [9].

1.3 Conventions and Notation

Unless otherwise specified, the unit of time is the *transmission slot.* It is the amount of time required to transmit one slot of data. The value is determined by the transmission rate in bits/second and the slot length in bits. For example, a transmission rate of 1 Gb/second, and a slot length of 400 bits implies that 1 transmission slot = 4×10^{-7} seconds. Similarly, the unit of distance, unless otherwise stated, is the *propagation slot.* Its value is linked to the *metre* by the transmission slot and the propagation speed of the transmission medium. A propagation speed of 2×10^8 m/second and a transmission slot of 4×10^{-7} results in a *propagation slot* of 80 metres. Both the *transmission slot* and *propagation slot* will be referred to as a *slot.* Thus the *slot* unit refers to time and distance simultaneously.

A number of symbols are used throughout the thesis — their meanings are listed below:

$\mathcal{K}$ the number of loops in the spiral network;

L the length of each loop of the network (in slots);

n the number of stations in the network;

S_i station i in the network;

d_i the distance separating S_i and S_{i+1};

w_i the token holding time at S_i;

W the total holding time spent by the token, during one rotation of the token;

C_{max} the theoretical maximum throughput of the network under optimal parameters;

C_{obs} the observed maximum throughput of the network using experimental parameters;

R the number of full rotations completed by a slot before it is absorbed by the token.

Chapter 2

Medium Access Control Protocols

Local and metropolitan area networks share a common transmission medium. Without some means of controlling access to the medium, devices inevitably interfere with each other's transmissions. A *medium-access-control* protocol determines where and how transmissions take place so that devices can successfully exchange data on the shared medium. In the OSI reference model of network protocols, the MAC layer lies between the *Data Link Control* layer and the *Physical* layer. An introduction to MAC-level protocols for LANs is given in [38]; a more detailed study is in [31]. Reference [25] discusses access strategies for fibre optic networks in particular.

This chapter discusses the goals of a MAC-level protocol for high-speed networks, and describes general characteristics of *slotted* MAC protocols like DSMA/S. In addition, we briefly describe two high-speed MAC protocols which have been studied extensively in the literature.

2.1 Slotted MAC Protocols

In a slotted MAC protocol, the unit of transmission is a fixed-size slot consisting of a slot header and a payload. A predetermined station continuously creates empty slots and transmits them onto the medium. The protocol determines when a station can transmit data into the empty slots. The logical unit of transmission, the message, is typically longer than the slot payload and must be segmented into slot-sized chunks. Overall, some fraction of the slot capacity is wasted because the message lengths are not necessarily multiples of the slot length, and the final segment requires some padding. (Non-slotted protocols allow variable length packets which eliminates the need for padding, except when a minimum packet length must be maintained.) No bandwidth is lost due to collisions, however, because stations transmit only into unoccupied slots. When network traffic is light, the minimum access delay is small and independent of network size, because stations must wait only until the beginning of the next slot before beginning to transmit[1].

An example of a simple slotted protocol for high-speed networks is LCSMA-CD/S/P, described in [25]. A designated station emits empty slots continuously onto a unidirectional carrier, while at the same time it removes from the network all traffic that reaches it. The responsibility for slot emitting may remain at one station permanently, or may be passed from station to station. Every slot starts with a *busy-bit* which can be set to mark the slot as occupied. When a station sets the busy-bit, the station's receiver returns to the controller the old value of the bit for inspection. When a station that is willing to transmit senses the beginning of a slot at its transmitter, it sets the busy-bit (marking the slot as full) and checks the previous value of the bit. If the slot was empty, the station's transmitter fills the header and payload of the slot. If the slot was already occupied, the station simply waits for the next slot and tries again.

2.2 Ideal MAC Protocols

Characteristics of an ideal MAC-level protocol for high-speed networks as discussed in [8] are:

1. It must be simple enough to be implemented directly in hardware and to operate at a speed that can exploit the bandwidth offered by optical fibre

[1] The minimum access delay will be bigger if there are additional restrictions on transmission, such as the possession of a token.

carriers.

2. It must be fair: both the mean packet delay and the maximum throughput of each station must be independent of the station's location relative to other stations.

3. The total throughput of the network (measured as the ratio of bits received to the time elapsed, expressed in bits) must be independent of the network size and transmission rate.

4. The average medium access delay must approach zero as the traffic load approaches zero.

5. Its operation cannot depend on centralised processing of feedback from stations as such processing introduces delays proportional to L.

6. It must accommodate heterogeneous traffic demands, guaranteeing simultaneously a finite maximum packet delay for synchronous traffic and a sustainable throughput for asynchronous traffic.

7. It must be able to carry synchronous traffic of variable intensity, up to using the whole bandwidth of the network.

8. It must be self-synchronising, so that jitter can be made negligible.

9. It must be predictable, so that a critical failure can be recognised by at least one station in a time smaller than L.

These criteria will be used to evaluate DSMA/S and other MAC-level protocols.

2.3 The FDDI and DQDB Protocols

Two high-speed MAC-level protocols that have received much attention recently are FDDI (and its new version, FDDI-II) and DQDB. This section gives a brief description of each.

2.3.1 FDDI

FDDI [32] is designed for large LANs or campus-area networks of up to 1000 stations, interconnected by a fibre-optic medium to form a ring of up to 100 km. It is designed to operate at a transmission speed of 100 Mbps. The non-slotted protocol is based on the concept of a token ring[2], in which a unique token is passed from station to station. A station can transmit only when it possesses the token. When it is finished transmitting it immediately passes the token to the next station. Every station uses a *token holding timer* to determine when it must pass the token. The token can be released earlier if the station has nothing left to transmit. During network initialisation, the stations negotiate for a *target token rotation time* (TTRT) which is to be the average time the token takes to travel the complete ring. The actual rotation time can vary depending on the network traffic load, but the protocol guarantees that it cannot exceed twice the TTRT and that the average rotation time is TTRT [34].

The performance of FDDI depends on the length of the network. As the ring size increases, the maximum achievable throughput decreases because of the extra travel time used by the token (no station can transmit when the token travels between stations). Furthermore, the minimum time between token captures increases with ring size resulting in a higher minimum access delay. A performance study of FDDI and some other high-speed LANs appears in [33].

Synchronous data, which is sensitive to delays, is transmitted before asynchronous data. The TTRT must be chosen to provide adequate transmission access for the synchronous data application. The irregularity of token rotation times, however, results in irregular transmission times which are not favourable to synchronous applications. Another problem is that throughput decreases when the TTRT is small relative to the ring latency. Applications that require regular and small transmission delays would find these limitations unsuitable.

[2] FDDI actually uses two rings, but one is reserved for use in the event of a failure in the other ring.

2.3.2 FDDI-II

An improved version of FDDI adds the capability of circuit switched channels for connection-oriented isochronous data[3]. FDDI-II, or Hybrid Ring Control, keeps the same topology and data rate as FDDI but uses a different access method. An overview of the architecture and protocol can be found in [6]; the draft standard is in [1]. A single station in the network is designated as the *cycle master*. It generates special frames every 125-μs which circulate around the ring. Each frame contains a 12-byte Data Packet Group (DPG), used exclusively for asynchronous packet transmission, and 96 16-byte Cyclic Groups (CG). The CGs form 16 Wide Band Channels (WBCs) which are allocated dynamically by the cycle-master to either isochronous or asynchronous traffic. Each WBC of 6.144 Mbps can be further divided into sub-channels with a bandwidth of at least 64 kbps.

Asynchronous packet data is transmitted only in the DPG and unallocated WBCs of each frame. The performance characteristics for asynchronous data are the same as in FDDI except that the bandwidth available for asynchronous data can vary depending on the number of WBCs used for isochronous data.

Because of the regular cycle time and channel allocation, the problems of token propagation and access delay do not arise for isochronous data. Each station is guaranteed transmission for isochronous data every 125-μs, provided it can allocate a channel. All of the bandwidth, except for the relatively small DPG channel, can be used for isochronous data, regardless of the network size.

2.3.3 DQDB

DQDB is a slotted, dual-bus architecture suited to Metropolitan Area Networks. It has no theoretical limit on the network length or number of stations, and is designed to operate at 155.520 Mbps. All of the network bandwidth can be used regardless of the length of the network. A good description of DQDB is in [17]; the proposed standard is in [14].

Each station has a connection to two unidirectional buses, BusA and BusB, which carry traffic in opposite directions. A headend station generates empty

[3] Isochronous data requires transmission at precise intervals and precise amounts. Digitised bit-stream video is an example.

slots on each bus. The 53-byte slots contain a 1-byte access control field (ACF), 4 bytes of segment header and 48-bytes of payload. To transmit, a station senses an empty slot and sets a busy-bit located at the front of the ACF (in the same way as described in section 2.1).

For asynchronous data, stations can request empty slots on one bus by sending requests on the other bus. If, for example, a station wishes to transmit on BusA, it sends a request on BusB. Every station on BusB that sees the request increments a request-counter, and decrements the counter each time an empty slot passes by on BusA. The station cannot transmit until the value of its counter drops below its value when the request was sent. Then it transmits in the first empty slot. Without this slot reservation scheme, stations nearest the headends could monopolise the network bandwidth. Even with its slot-reservation scheme, DQDB still exhibits throughput unfairness under heavy traffic conditions [43], although some attempts have been made to correct this [16].

When requested, the headend stations can generate dedicated isochronous slots which provide independent *virtual channels* for regular and guaranteed transmission of isochronous data.

2.4 Other MAC-level Protocols for High Speed Networks

Many other MAC-level protocols for high-speed networks have been proposed in the literature. These protocols will not be studied in this thesis, but a few are mentioned below for reference.

- CRMA, proposed by IBM Research Division, Zurich, is derived from DQDB. It uses a folded bus topology and a cyclic reservation scheme. A description is found in [29].

- A related protocol, called DQMA, is described in [26].

- In [4], a slotted protocol for a dual counter-rotating ring is described. The protocol, called DSDR, uses a token to achieve fairness in a way similar to DSMA/S.

- The *Machnet* access protocol for high-speed or long-haul networks is described in [15].

- The *Fasnet* LAN protocol is presented in [22].

- Limb [23] describes a simple slotted protocol for MANs.

Chapter 3

The Spiral Ring Topology

The spiral ring topology [7] is formed by looping a fibre-optic cable into a spiral. Each loop of the spiral is the same length. The top end of the spiral is located directly above the bottom end; the ends are connected. Every station in the network has a connection point or *port* on each loop of the spiral; these connections are directly one above the other. The ports are taps into the fibre-optic cable. The stations have a receiver and transmitter at each of their ports. Only one transmitter is active at a time to guarantee that messages are queued, transmitted and received in the same order. Note that since every station has a port on each loop, data must only travel one loop to pass a receiver at each station in the network. Figure 1 shows a 5-loop spiral; only one station is shown for clarity and the loops are not shown to be of equal length.

The familiar IEEE 802.5 [3] token-ring topology is an example of a spiral ring with a single loop; the *Pretzel Ring* [9] is a spiral ring with two loops.

3.1 Ring Latency

Each port consists of a signal repeater that either relays the incoming signal (receiving) or replaces it with the station's own outgoing signal (transmitting). These repeaters introduce a small, but constant delay. Since each loop of the spiral is exactly the same length, and has a port for each station, the time it takes for a signal to travel a complete loop is constant. More specifically, the

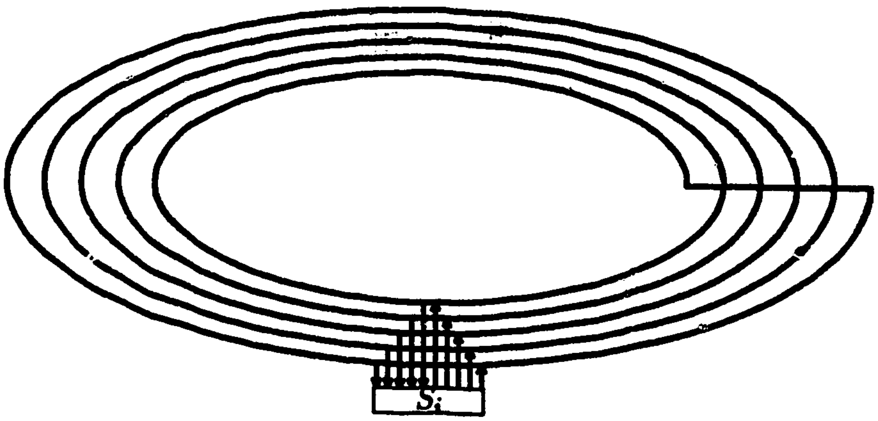

Figure 1: A 5–Loop Spiral Ring

time it takes for data to travel from port i to port $i+1$ of any given station is constant, and is the same for all stations.

3.2 Loop Length

The operation of the slotted DSMA/S protocol, described in the next chapter, requires that the length of the transmission medium meets certain criteria. The total length of the spiral must be an integer multiple of the length of a slot, and must not be divisible by $\mathcal{K}$, the number of loops. This results in a loop length that is not an integer multiple of the slot length. Without this condition, all of the ports at a given station would sense the beginning of a slot at the same instant. This raises the following implementation problem. A station attempts to transmit on one of its ports. If the slot at the port is not empty, an attempt is made at another port. The determination that a port is unable to transmit, and the process of re-trying the transmission on a different port requires a small but non-zero time. This time is provided by the delay between the arrival of slot headers at the different ports.

Chapter 4

The DSMA/S Protocol

The DSMA/S protocol is the slotted medium-access protocol for the spiral-ring topology[1]. It is a *weak-token* protocol: a token is used to mediate fair access to the transmission medium, but access is not solely dependent on the token. Like a true token-passing protocol, a token is circulated around the ring. When a station receives the token, it captures it and holds it for a period of time. When the holding time has expired, the station releases the token. A station that holds the token at one of its ports has unrestricted access to transmit at that port for as long as it holds the token. In DSMA, however, possession of the token is not required to transmit: if a station senses an empty slot at a port it may also transmit, providing certain conditions are met. This characteristic allows DSMA to offer access delays approaching zero when network traffic is light regardless of the network length.

In the remainder of this chapter, we present the details of the protocol and its properties

4.1 Token Timing

In this section we describe the timing characteristics of the token passing mechanism in DSMA/S.

[1] A version for unslotted rings is discussed in [9].

The token is passed around the spiral ring from station to station. Each station holds the token for a period called the *token holding time* (THT) that is associated with the station. The THT is constant for each station[2], but can vary between stations. Since each station holds the token for fixed time, and since each loop of the ring is the same length, the time it takes the token to travel around a loop is also fixed. Borrowing the terminology from FDDI, we call this the TTRT (*Total Token Rotation Time*)[3]. The value of TTRT is given by the formula:

$$\text{TTRT} = L + \sum_{i=1}^{n} \text{THT}_i + \varepsilon$$

L is the propagation delay for a single loop including the delays caused by the repeaters at each station, while ε represents the variability in repeater delays at the stations (i.e., the difference between maximum and minimum delay).

Each station is equipped with a *token rotation timer* (TRT) which indicates the time elapsed since the station captured the token. The timer is reset to zero when the station acquires the token, and counts until the token arrives again. The timer has two functions:

- It times when the THT has elapsed and the token is due for release.
- It allows the station to estimate how far away the token is. The rules for transmission, presented later, explain how this information is used.

In FDDI, the arrival time of the token conveys some information about the bandwidth available to the station: a token that arrives before the TTRT has elapsed confers additional token holding time, and therefore, more bandwidth. In DSMA/S, however, there is no significance to early token arrival, since the actual token rotation time is nominally fixed and varies only slightly due to small variations in the repeaters. Rather, the arrival of the token in DSMA/S serves to synchronise the timers in the stations by resetting the TRT.

[2] In FDDI, a station can release the token earlier, as soon as it has no more messages to transmit.

[3] Unlike FDDI's TTRT, this is a constant quantity rather than a target not to be exceeded.

Access Control		Segment Header	Segment Payload
busy (1 bit)	other bits (7 bits)	addressing, check sequence (4 bytes)	data (48 bytes)

Figure 2: Basic Slot Format for DSMA/S.

4.2 Slot Format

The format of the DSMA/S slot is loosely modeled on the DQDB slot [14], although some fields, such as the *request* field, are not needed[4]. The slot is a 53-octet transmission unit composed of a 1-octet *access control field* (ACF), a 4-octet *segment header*, and a 48-octet *segment payload*. Figure 2 illustrates the basic format.

As in DQDB, the first bit of the ACF is a *busy* bit which indicates if the slot contains information (1) or is empty (0). The other bits in the ACF are not yet defined: the *request* bits of DQDB are not needed. Other bits could be used to indicate a slot-type for different types of services. The segment header contains addressing information and a header check sequence, while the segment payload carries 48 bytes of data.

4.3 Port States

Recall that each station in a spiral network has one port on each loop of the spiral, and that transmissions can take place from any port. In each station, the ports have two possible states, which are dynamically determined by the location of the token in the spiral[5]. Note that, at any given time, exactly one of the station's ports is located upstream from the token at a distance less than or equal to L. This port is called the *secondary* port — all other ports are *primary* ports. Transmitting from a secondary port is conditional upon the token being far enough away that a transmitted slot will not hit the token before traveling at

[4] The *exact* format and function of fields in the slot are beyond the scope of this study. We restrict ourselves to parts of the slot that are relevant only to access control.

[5] The single-loop spiral is a special case, as discussed in 4.11.

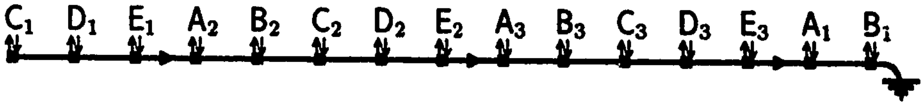

Figure 3: A 3-Loop Spiral (shown as a bus)

least one complete loop of the spiral. The timing condition is explained in Section 4.4. The same condition is not needed when transmitting from primary ports, because the token is always more than the loop-length L downstream.

The state of the ports is controlled by movement of the token. When the token holding station releases the token, two ports change state:

- The port on which the token is released becomes the secondary port.
- The station's former secondary port becomes a primary port.

No other ports change state at this time.

As an example, consider the 3-loop spiral in Figure 3. There are five stations labeled A, B, C, D, E. The spiral has been cut, flattened and straightened out so that the carrier can be seen as a bus, to which each station is connected three times. The subscripts in the station identifiers represent the spiral loops: C_1 represents the port connecting station C to the first loop of the spiral[6]. The three loops of the bus are shown as three groups of five ports. Station C holds the token at port C_1, as indicated in the figure by C_1 being at the head of the bus. Slots are generated at C_1 and travel down the full length of the carrier until they are absorbed by the token. Note that ports C_3, D_3, E_3, A_1, and B_1 are all at a distance less than or equal to the loop length from the end of the carrier (i.e.: the token): these ports are *secondary* ports. All other ports are primary. When station C releases the token, port C_1 will become a secondary port, and C_3 will become primary.

[6] First is just an abstract concept.

4.4 Rules for Transmission

Transmission on the fibre-optic medium is controlled by the set of rules described below.

1. Whenever a station senses an empty slot at a primary port it can use the slot for transmission. Slots are acquired by successfully setting the busy bit in the slot header. A station with something to transmit continuously attempts to set the busy bit in each slot as it passes. Whenever it succeeds, the station fills the payload portion of the slot with data.

2. Whenever a station senses an empty slot at a secondary port it can use the slot, provided that $\text{TRT} > \text{TTRT} - L$. As explained later, this timing condition guarantees that the slot will not hit the token before completing at least one rotation and therefore visiting each station at least once. If the timing condition is not met, the slot is left empty.

3. The station that holds the token empties all the slots that reach the port where the token was captured. This is done by resetting the busy bit in each slot. Of course, these empty slots can now be used again by the station at the token holding port.

   The arrival of a non-empty slot whose sender is *not* the token-holding station, is an indication of a station failure. The transmission rules assure that used slots complete at least one loop before hitting the token, otherwise the slots are left empty.

4. Every station either empties or reuses slots that it has transmitted. Since a station knows the distance between its ports, it is able to time when a slot transmitted at one port is about to arrive at the next port in the spiral. When the used slot arrives, the station either retransmits into it or empties it. Details of how this timing is accomplished are discussed in Section 4.4.1.

5. When the station's token holding time has expired, the token is transmitted on the same port at which it arrived.

The timing condition for secondary port transmission is necessary to avoid sending slots that will be absorbed by the token before reaching their destination. The reasons are:

- A slot is guaranteed to have passed its destination if it travels at least one complete loop (i.e.: it reaches the sending station's next port) before reaching the token.

- It takes a slot L time units to travel one loop.

- Recall that the station's TRT measures the time since the token was last captured. Let t be the reading of the TRT in the sending station. Then the station will receive the token again at $\text{TTRT} - t$.

- The token must arrive before the slot, so the condition for successful transmission is

  $$\text{TTRT} - t < L;$$

  otherwise, the slot would be absorbed by the token before returning to its sender.

4.4.1 Slot Removal and Re-use

In slotted rings, there are two possible methods for reusing slots. In the *source-reuse* method, the station that originally sent the message retransmits into the slot when it returns. Since the slot has traveled the entire ring it is guaranteed to have passed its destination already. Stations must recognise their own slots in order to reuse them. Since the number of slots in the ring is known, stations only need to count passing slots to determine when a transmitted slot has returned. In this method, slots travel an average of half a ring length after passing the destination station before being reused. This results in significant wasted capacity, since the data is not needed after passing the destination.

The *destination-reuse* method addresses this problem. Stations receiving a slot, immediately re-use it or mark it as empty. However, stations must buffer the slot up to the destination address field in the header, introducing significant delays at each station. Furthermore, if the intended destination station fails or does not exist, slots are not re-used and may circulate forever unless some other mechanism is used to absorb them.

DSMA/S uses the source-reuse method. Each station empties and, if possible, reuses the slots that it transmitted. Stations know the time it takes a slot to travel a complete loop, so they can anticipate the arrival of their previously transmitted slots. This is accomplished by each station maintaining a FIFO event queue of expected slot arrival times and a timer to trigger the events. When a station transmits a group of one or more slots sequentially, it adds a record to the event queue. The record indicates:

- when the group will arrive,
- and the number of slots in the group.

When the timer matches the event at the head of the event queue, the next slot to arrive will be the station's own. It is not necessary to record the port at which the slot will arrive: since L is not a multiple of the slot length, only one of the station's ports will be sensing the start of a slot at the event time. The arrival time is $t + L - \delta$, where t is the time that the slot was transmitted, and δ is a small time constant less than the transmission time of a slot. The subtraction of δ ensures that the arrival event is triggered just before the slots actually arrive.

4.5 Resulting Properties

The transmission rules give rise to a number of interesting properties of the network. This section discusses these results.

4.5.1 Slot Cycles

Consider the life of a slot in the spiral network. In short, it is generated by the token holding station and travels along the medium until it reaches a token holding port again where it is absorbed and dies. The slot undergoes a number of use-cycles during its lifetime. The cycle begins by a station filling the slot with data as the slot passes one of its ports. It ends when the slot returns to the station's port on the next loop in the spiral where it is either refilled, emptied or absorbed. The slot travels exactly one loop, or does one *rotation* of the spiral during a cycle.

During each rotation, the slot passes a port belonging to each station in the network so the receiving station can read the data in the slot. Since the token moves, slots can make many rotations before they are absorbed at the token holding port, and can therefore be reused many times. The number of times a slot can be reused is discussed later.

4.5.2 Transmission Windows

One consequence of the transmission rules for the protocol is that each station gets a number of guaranteed opportunities to transmit. We will refer to these as *transmission windows*. An obvious transmission window occurs when a station holds the token, but another window occurs when the slots transmitted reach the station's port on the next loop of the ring. In particular, when the token-holding station transmits m slots starting at time t from any primary port, it is guaranteed a transmission window of (at least) m slots at time $t + L$, where L is the length of a loop. It takes a time of L for the burst of m slots to arrive at the station's port on the next loop of the ring, where they can all be reused.

More generally, a transmission window occurs whenever a group of slots arrives at a primary port belonging to the station that transmitted them. If the port is in the secondary state, a transmission window will occur if the timing conditions described in the rules for transmission are met.

4.5.3 Slot Re-uses

Recall that, according to the transmission rules, a station can reuse slots that it transmitted earlier at another port. Slots can make a number of rotations in the spiral before being absorbed. At the end of each rotation, the slot can be reused if the station has something to transmit, provided that the slot is guaranteed to make at least another full rotation before hitting the token.

The number of times a slot can be used from the time it is generated at a token holding station to the time it is re-absorbed at the token holding station depends on the number of loops, $\mathcal{K}$, in the spiral, the sum of the THTs, and the loop length L.

Consider the following situation. At time 0, station S_0 gets the token and generates a group of slots. At time THT_0 it releases the token. A race begins: the slots travel around the spiral some number of times and eventually catch up with the token again at some station S_m.

We can make the following observations:

- Obviously, a slot can only hit the token while it is being held at a station, since when the token moves, it travels at the same speed as the slot.
- The slot must make at least $\mathcal{K}$ rotations in a $\mathcal{K}$-loop spiral because it must pass its starting point at least once to catch up with the token.
- As a consequence of the previous observation, the slot makes exactly $\mathcal{K}$ more rotations than does the token.

We now derive an expression for the number of times a slot can be used.
Let w_i be the THT at station S_i.
Let d_i be the distance from S_i to S_{i+1}.
Let n be the number of stations.
Let W be the total holding time of the token in one loop: $\sum_{i=0}^{n-1} w_i$.
Let L be the length of one loop of the spiral: $\sum_{i=0}^{n-1} d_i$.
Let R be the number of full loop rotations the slot completes.
Let S_m be the station where the slot hits the token[7].
The token arrives at station S_m at time t_a after traveling $R - \mathcal{K}$ full rotations plus the partial rotation to station S_m:

$$t_a = (R - \mathcal{K})(W + L) + \sum_{i=0}^{m-1} (d_i + w_i)$$

[7] Note that S_m cannot be chosen arbitrarily. Rather, it is determined by the values of L, $\mathcal{K}$, and W.

and departs at t_d

$$t_d = t_a + w_m.$$

When the token arrives, the slot in which it was sent can be immediately reused. When the token is released, the slot which is absorbed by the token-holding port when the THT expires becomes the token.

The first slot in the group generated by S_0 arrives at S_m at time

$$t_0 = RL + \sum_{i=0}^{m-1} d_i.$$

The j^{th} slot in the group arrives at S_m at time

$$t_j = t_0 + j \qquad (0 \leq j \leq w_0 - 1).$$

Slot j will hit the token if it arrives at S_m while the token is present:

$$t_a + 1 \leq t_j \leq t_d.$$

Substituting the expressions for t_a, t_j, and t_d we get

$$\begin{aligned}(R-\mathcal{K})(W+L) + \sum_{i=0}^{m-1}(d_i + w_i) + 1 \; &\leq \; RL + \sum_{i=0}^{m-1}(d_i) + j \\ &\leq \; (R-\mathcal{K})(W+L) + \sum_{i=0}^{m-1}(d_i + w_i) + w_m\end{aligned}$$

Expanding and simplifying the terms gives

$$R + \frac{\sum_{i=0}^{m-1} w_i + 1}{W} \leq \mathcal{K} + \frac{\mathcal{K}L + j}{W} \leq R + \frac{\sum_{i=0}^{m-1} w_i + w_m}{W} \quad . \tag{4.1}$$

Since

$$\frac{\sum_{i=0}^{m-1} w_i + 1}{W} < \frac{\sum_{i=0}^{m-1} w_i + w_m}{W} \leq 1$$

we can simplify the inequality further to get

$$R < \mathcal{K} + \frac{\mathcal{K}L + j}{W} \leq R + 1,$$

and by employing the *ceiling* function,

$$\left\lceil \mathcal{K} + \frac{\mathcal{K}L + j}{W} \right\rceil = R + 1.$$

Finally, the number of times slot j is used is

$$R = \mathcal{K} + \left\lceil \frac{\mathcal{K}L + j}{W} \right\rceil - 1 \; . \tag{4.2}$$

4.5.4 Regular Transmission Windows

Under certain conditions, stations can receive transmission windows at regular time intervals.

Consider the case of a saturated network: transmissions only take place during slot re-use or token possession. A station S_i receives the token repeatedly at intervals of $L+W$, and transmits a group of w_i slots. It takes time L for the slots to return to the station, and a further W for the token to return. If W is a divisor of $\mathcal{K}L$, that is, if $jW = \mathcal{K}L$ for some positive integer j, then the spiral carrier will contain j groups of slots from each station, separated by W slots. Therefore, each station will receive a transmission window at regular intervals of W.

4.6 Fairness

The DSMA protocol guarantees fair access to the medium for each station in the network. Under light traffic conditions, when a small portion of the bandwidth is in use, access is effectively allocated on a demand basis. Those stations that have some data to transmit can do so whenever some unused slots pass their ports[8]. Under heavy traffic loads and particularly at network saturation, each station is allowed access in a more regular way that is controlled by the movement of the token. If each station has the same value for THT, each station holds the token for the same amount of time and has the same opportunities for transmission. FDDI's token passing scheme allows it to be perfectly fair also, although not necessarily within one token rotation.

The same cannot be said about DQDB's access protocol which is based on slot reservation. Since each station is allowed to reserve only one slot a time, those stations that are closer to the ends of the bus get access more frequently [43]. This is because the closer a station is to the end, the quicker it can have its request satisfied and send another request.

Solutions to the fairness problems of DQDB have been attempted. A bandwidth balancing mechanism [11] provides a fairer bandwidth allocation among

[8] Subject, of course, to the transmission rules.

nodes in a saturated network, regardless of their position, but assumes symmetric, single-priority, steady traffic loads, which are unlikely in real MANs. Others have proposed alternative fairness strategies [36, 37, 27, 21, 12, 18], but they tend to increase the complexity of the protocol.

DSMA is inherently fair, and needs no complicated strategies to achieve it.

4.7 Secondary Port Availability

As discussed in section 4.4, transmissions from secondary ports are subject to a timing constraint $t > \text{TTRT} - L$ which guarantees a slot will complete at least one loop of the spiral before being absorbed. When the constraint is not met, transmission on the secondary ports is disabled. Since t ranges from 0 to TTRT, the condition is met for a period of L during each token rotation. Hence, the proportion of time that secondary ports are enabled for transmission is simply $t_{sc} = \frac{L}{\text{TTRT}}$.

Note that even though a station's secondary port may be enabled for transmission, it is still possible that it cannot transmit: the passing slots may not be empty or reusable by the station. This can be true during saturated traffic conditions where transmission opportunities are limited to the token holding period and reuse of a station's own earlier transmitted slots. These conditions are discussed in Section 4.10.

4.8 Access Delay

There are several measures of delay in accessing the transmission medium in networks. They all measure the time it takes a station to transmit some unit of data. Physically, from the point of view of a slotted MAC protocol, transmission is done in slots, while logically, the units of transmission are messages. This leads to the following access delay measures:

1. The mean *slot access time* is the mean time elapsing from the moment when a station becomes ready to transmit a slot and the moment it begins

transmitting it. The equivalent measurement in unslotted networks is the *packet access time*.

2. The mean *message waiting time* is the time elapsing between the moment a message is enqueued and the moment when it is completely transmitted.

Recall that DSMA/S does not require stations to possess the token to transmit: if a station senses an empty slot at a port, it may transmit subject to the secondary port timing restrictions described in the rules for transmission. Under light traffic conditions (i.e., when there are many empty slots) the average slot access time is half a slot, since stations only have to wait for the beginning of the next slot. The light-load delay is independent of the length of the network medium, the number of stations in the network and the token holding times at the stations[9]. The message waiting time in DSMA/S during light traffic is simply the sum of the message length (in slots) and the slot access delay (half a slot). DQDB exhibits the same light-load delays as DSMA/S.

In FDDI, possession of the token is required before transmission can occur. Even though there is the facility for early token release in FDDI when traffic is light, the minimum packet access delay is still of the order of the ring latency which increases with the length of the ring. On average, a station can expect a packet access time of not less than one-half the ring latency even under very light traffic conditions. For long networks, this can be a large delay. The light-load mean message waiting time is the same with the addition of the message transmission time.

4.9 Throughput

The term *effective throughput* is a measure of the capacity of the network to carry data. It is the ratio of the amount of data transmitted (excluding protocol overhead) over the amount of time it takes. Recall that in our abstraction, the unit of data and time is the slot.

[9] Networks with a single loop have higher access delays, because a station's only port is subject to the secondary port transmission timing constraints. See 4.11.

The theoretical maximum effective throughput for DSMA/S can be derived easily. For each loop in the spiral, the total available throughput is 1. The amount of throughput lost due to slot overhead is simply h_s/s for slots of length s with headers of length h_s. For a $\mathcal{K}$-loop network the maximum effective throughput C_{max} is

$$C_{max} = \mathcal{K} \times (1 - h_s/s).$$

Note that throughput is not affected by the length of the network[10]. A 2-loop network has the same maximum throughput as DQDB without bandwidth balancing

$$2 - 2h_s/s.$$

In FDDI, the maximum effective throughput can be derived formally as:

$$\left(1 - \frac{L}{\text{TTRT}}\right) \times \left(1 - \frac{h}{p_f}\right)$$

for packets of average length p_f with headers[11] of length h. Note that for a fixed TTRT, an increase in the length L of the network decreases the throughput. This is due the fact that in FDDI, no transmission takes place when the token is moving. A longer network increases the travel time of the token.

In Section 4.10 we show that a number of factors can affect practical levels of throughput in DSMA/S, but the network parameters can be chosen to maximize throughput.

4.10 Factors Affecting Maximum Throughput

The relationship between $\mathcal{K}$, the TTRT, and L affects the maximum achievable throughput. In equation 4.2 we showed the number of times a slot is reused before hitting the token. This is the number of *complete* loop-rotations the slot makes. Only under the right conditions will the slot hit the token at the end of a complete rotation (i.e., at its originating station). Otherwise, the slot travels the

[10] We will show later that the length should meet certain conditions, but the length is not limited.

[11] The headers in FDDI are not the same length as in DSMA/S, so the resulting overhead is different.

remainder of its journey empty, thus wasting network capacity. Consequently, maximum throughput will occur when the network parameters result in slots hitting the token only at their originating station.

To determine the optimal parameters, we first derive an expression for the station number, S_m, where the slot hits the token. Recall equation 4.1 from Section 4.5.3:

$$\begin{aligned} R + \frac{\sum_{i=0}^{m-1} w_i + 1}{W} &\leq \mathcal{K} + \frac{\mathcal{K}L + j}{W} \\ &\leq R + \frac{\sum_{i=0}^{m-1} w_i + w_m}{W} \qquad (0 \leq j \leq w_0 - 1) \end{aligned}$$

which describes the rotations made by the j^{th} slot in the group transmitted by station S_0 while it held the token. By rearranging the terms, we get the following expression:

$$\sum_{i=0}^{m-1} w_i + 1 \leq \mathcal{K}(L+W) - WR + j \leq \sum_{i=0}^{m-1} w_i + w_m \ .$$

To simplify further, consider the special case where the THT is the same in all stations (i.e., $w_i = w$ for all i). The above relation becomes

$$mw < \mathcal{K}(L+W) - RW + j \leq (m+1)w,$$

which can be simplified to give

$$m = \left\lceil \frac{\mathcal{K}L - (R-\mathcal{K})W + j}{w} \right\rceil - 1 \qquad (0 \leq j < w)\ . \tag{4.3}$$

To achieve maximum throughput m must be zero which occurs when

$$\mathcal{K}L = (R - \mathcal{K})W\ , \tag{4.4}$$

because $0 \leq j < w$. Since R is an integer and $R > \mathcal{K}$, $(R-\mathcal{K})$ can be any positive integer. Hence, to achieve maximum throughput, W must be a divisor of $\mathcal{K}L$, or equivalently, the ratio $\mathcal{K}L/W$ must be an integer. Solving equation 4.4 for R, the exact number of rotations that a slot completes is

$$R = \mathcal{K}\left(\frac{L}{W} + 1\right). \tag{4.5}$$

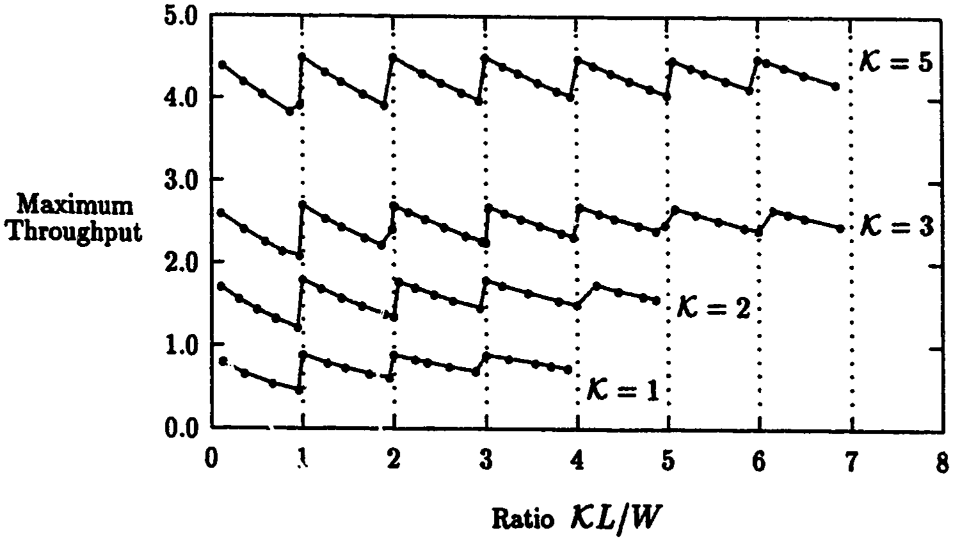

Figure 4: The Effect of Ratio $\mathcal{K}L/W$ On Maximum Throughput. (32 stations, 1 Mb loops).

Simulation experiments[12] were conducted to determine the maximum achievable throughput for networks with different ratios of $\mathcal{K}L/W$. The networks contained 32 stations equally spaced, with uniform THTs. The loop length was fixed at 2496 slots (approximately[13] 1.06 Mbit), while the the THT was varied. The networks were modeled under saturated traffic conditions.

Figure 4 shows maximum throughput as $\mathcal{K}L/W$ varies for spiral networks with 1, 2, 3, and 5 loops. For each network, the throughput repeatedly drops, then peaks sharply around integer values of $\mathcal{K}L/W$. Close examination of the results reveals that throughput peaks exactly at integer values of the ratio, except when $\mathcal{K}L/W$ is a multiple of $\mathcal{K}$ and $\mathcal{K} > 1$, when it reaches a minimum. This effect is due to a timing condition that results in the conditions for a throughput minimum.

Recall that when $\mathcal{K}L/W$ is an integer, slots travel an integer number R of

[12] Details of the simulation environment are discussed in Section 5.1.

[13] We start with 2^{20} bits, divide by 424 bits/slot and round up to get an integer number of slots between the 32 stations.

rotations before hitting the token. The token travels $R - \mathcal{K}$ rotations. If R is a multiple of $\mathcal{K}$, then whenever the slots have completed $i\mathcal{K}$ rotations, the token has completed $i(\mathcal{K} - 1)$ rotations, where $i = 1, 2, 3, \ldots, (L/W + 1)$. This means that when the slots return to their originating port, the token is at another port of the same station. Since the transmission rules give priority to the token holding port, no transmission takes place at the originating port and the slots are not reused. Instead, they travel empty to the next station in the network. This allows the next station to transmit a burst of slots earlier than if it waited to reuse its own slots[14]. If each station has a THT of w, then the slots are effectively *ahead of schedule* by w. They hit the token at a time w earlier than expected, thus they do not complete their final rotation and are not reused at their last reuse point.

4.11 The Single Loop Network

A $\mathcal{K}$-loop spiral network with $\mathcal{K}$=1 is a special case and deserves some explanation. The network has the following characteristics.

1. The topology of a single loop spiral network is simply a ring.

2. Each station has only one port.

3. The ports are always in the *secondary* state, because the token is never more than a loop length away. However, possession of the token gives the right to transmit without meeting the secondary port timing condition.

4. Since transmissions can happen only during token possession or when the secondary port is enabled, the mean slot access delay, even under light traffic loads, will depend on the secondary port availability[15]. When the port is enabled, the slot access time under light loads is half a slot. However, the ports are disabled for a period of TTRT $- L$ during each token rotation. Transmissions will be delayed until the port becomes enabled again, which, at any instant, depends on the value of the TRT at the station.

[14] In fact, it cannot reuse its own slots for the same reason its upstream neighbour could not.

[15] Discussed in section 4.7

The average slot access time for light loads can be derived discretely in the following way. Consider the simple case of a single-loop network with n stations, each station S_i separated from the next by a distance d and having THT of w. Let t_i be the value of the TRT at S_i. Our approach is to consider the network at discrete time intervals during a representative period, determine the delays at all stations for each interval, then average the delays over the period. The interval corresponds to one slot of time. We will consider a period of $d + w$ slots that begins when station S_{n-1} releases the token, and ends just before S_0 releases it. The delays at the stations will be the same during any similar period, as the relative states of the stations will be identical.

At the beginning of the period, as the token is released by station S_{n-1}, $t_{n-1} = w$ and more generally, at station S_i, $t_i = (n-i)(d+w) - d$. Furthermore, at a time δ past this release, where $0 \leq \delta < (d+w)$,

$$t_{i,\delta} = ((n-i)(d+w) - d + \delta) \bmod (n(d+w)).$$

The port is disabled until $t_i \geq \text{TTRT} - L$ or, in our simplified case $t_i \geq nw$. The delay until the port is re-enabled is simply $D_{i,\delta} = nw - t_{i,\delta}$. If the port is already enabled the delay is 0.5 slots. The delay at station S_i during time interval δ is

$$D_{i,\delta} = \max(nw - t_{i,\delta},\ 0.5)$$

The average delay over the entire period $\delta = [0, d+w)$ is then

$$D_{av} = \frac{1}{d+w} \sum_{\delta=0}^{(d+w)-1} \left[\frac{1}{n} \sum_{i=0}^{n-1} D_{i,\delta} \right] \tag{4.6}$$

Table 4.1 shows some sample average delays for various values of n, d, and w.

5. The maximum throughput is still limited by $1 - h_s/s$ in agreement with the formula for general $\mathcal{K}$-loop networks given in Section 4.9.

6. Slot re-use by the slot sender is still possible provided that TTRT $< 2L$. To see why this must be true, consider the following two cases.

n	10			100			500		
d	250			25			5		
w	1	10	100	1	10	100	1	10	100
D_{av}	0.5	2.1	116.2	2.4	140.5	3920.9	42.0	1660.5	23714.9

Table 4.1: Sample Average Slot Access Times, D_{av}, for a 2500-slot Single Loop Network

- A station transmits a slot while holding the token. The slot will return when $t = L$, but to be re-used, the secondary port timing constraint $t > \text{TTRT} - L$ must be met. This can only be true if $\text{TTRT} < 2L$.
- A station transmits a slot, without holding the token, at

  $$t_0 = \text{TTRT} - \delta, \qquad (\delta < L).$$

  The slot returns to the station after the token has visited the station once, resetting the TRT clock. The value of TRT when the slot returns is $t_1 = L - \delta$. Again, the port is enabled for transmission if $t_1 > \text{TTRT} - L$, so

  $$\text{TTRT} < (2L - \delta) < 2L.$$

Chapter 5

Simulation Experiments

A series of simulation experiments were conducted to investigate the behaviour of the DSMA/S protocol under various conditions. While analytical models for networks are useful and tractable when simplistic assumptions are made, their complexity can become overwhelming when some real life parameters are used. Simulation provides a useful tool for testing hypotheses under non-trivial conditions.

This chapter describes the simulation environment used, and the experiments that were performed. The results of the experiments are presented in the next chapter.

5.1 Simulation Environment

The simulator was written in C^{++}. The unit of time for the simulation was the *slot*. The transmission medium was implemented as an array of pointers to slots. Stations had ports associated with fixed locations in the array. As each slot of time passed, the pointers were just shuffled to point to the next slot. A minimum amount of information was kept in each slot: source and destination addresses, and access information; no data needed to be stored. Two queues were maintained for each station's messages awaiting transmission: one each for synchronous and asynchronous messages. Separate message generating functions were implemented for asynchronous and synchronous traffic patterns. Statistics that were kept for both synchronous and asynchronous traffic included: throughput, message waiting time, and total slots transmitted.

A pseudo-random number generator was used to produce distributions for message lengths and message interarrival times. The random number generator employed was the one in the LANSF[10] simulator which has been used extensively in network simulation studies. The reader is referred to [19] for a comprehensive discussion of random number generators.

Simulations take some time to reach a state where the operation of the model is steady. This is because the start up conditions can be quite different than the final steady state that is being modeled. For example, in modeling heavy traffic conditions, it takes some time for the initially empty transmission medium and empty message queues to fill up. The behaviour of a network under these startup conditions is different than when the medium and queues are nearly full. Taking performance measurements during this initial period can distort the measurements taken later when operation is more stable. To reduce the impact of the start-up conditions in the simulation, we allowed a *warm-up* time before measurements were started. After some experiments to determine how long the model took to reach a steady state under various conditions, we chose a period equal to the time it takes the token to travel around the whole spiral once. After the warm-up period, the simulation experiments each ran for 235840 time slots, which is approximately equal to 0.1 seconds of real time given the parameters used for the synchronous traffic model.

5.2 Performance Measurement

In this study we are interested in the performance of the DSMA/S protocol under various network traffic conditions. Two factors to consider when evaluating network performance are access delay and effective throughput.

Two measures of delay in accessing the transmission medium were introduced in Section 4.8: packet or slot access time; and message waiting time. Packet or slot access times can be misleading in oversaturated network conditions. For example, in an over-saturated FDDI network, once a station receives the token, it can transmit a large number of packets with all but the first packet encountering a delay consisting only of the packet transmission time. The first packet waits a relatively large tim‹ for the token to arrive, but this is amortised over the many packets transmitted during that token possession. The mean packet access time can decrease as traffic load increases. We chose the mean message waiting time to measure delay, because it accounts for the time a message waits in the queue, which is significant in over-saturated traffic conditions.

Effective throughput is a measure of the data carrying capacity of the network. It measures the number of units of data[1] received over a unit of time.

As the traffic load on a network increases, more data is transmitted and received and the throughput increases up to a certain limit. However, due to the increasing load, the access delay can also increase. Studying the access delay behaviour over varying throughput levels is one common way of evaluating the performance of network protocols.

5.3 Asynchronous Traffic Conditions

Traffic in computer networks can be broadly classified into ***asynchronous*** and ***synchronous*** types[2]. This section is devoted to explaining our asynchronous traffic experiments, while synchronous traffic is described in the next section.

Asynchronous traffic has the following characteristics:

- irregular arrival times to the network
- non-critical delivery time
- possibly non-uniform message sizes
- vulnerability to packet loss

Some examples of asynchronous traffic are file transfers, terminal key strokes, electronic mail, and facsimile. Some studies have been done on the typical characteristics of asynchronous traffic in computer networks [24, 30, 35]. The nature of

[1] Protocol overhead, such as slot headers, are not counted as data.

[2] Also called *non-real-time* and *real-time*, respectively.

the traffic is undoubtedly dependent on the particular environment (i.e., types of computers, applications, users, etc.). For example, a network connecting remote terminals to hosts transmits very short packets of 1 or 2 bytes of data corresponding to keystrokes, while a network of workstations with large amounts of memory is likely to perform file transfers of hundreds or thousands of bytes each.

Little is known about the traffic characteristics in Metropolitan Area Networks for which DSMA/S is intended. We chose to model asynchronous traffic with the following characteristics:

- Message inter-arrival times (MITs) were exponentially distributed.
- Message lengths were exponentially distributed. The mean message length was chosen to be 4K bits, but message lengths were constrained to be multiples of slot data lengths. This resulted in a mean message length of 11 slots.
- Traffic load was uniformly and randomly distributed over the stations; (i.e., each station had equal probability of sending a message). Destination addresses were ignored because slots must pass all stations in the network before they are absorbed or emptied.
- The network consisted of 32 stations, equally spaced on the network.
- The loop length was 1 Mb.
- Experiments were conducted for spiral networks with 1, 2, 3 and 5 loops.

The traffic load was varied by specifying different message inter-arrival times. By choosing a sufficiently short MIT it is possible to cause a saturated condition, where all stations always have a message to transmit.

The results are presented in the next chapter.

5.4 Synchronous Traffic Conditions

Network applications that send data at regular time intervals and require a (usually small) guaranteed maximum delay generate *synchronous* data traffic. The data is generated in real-time at regular periods, and must be delivered within some time limit. Examples of synchronous traffic are: digitised voice, digitised video, and sensor information. In this section we describe some of the requirements of synchronous traffic, transmission strategies for synchronous traffic in DSMA/S, and the model of synchronous traffic used in our simulation. We then describe the synchronous traffic experiments.

5.4.1 Synchronous Data Requirements

The requirements for packetised voice traffic are well established. Human speech can be band limited to 4 kHz and digitally sampled at 8 KHz to produce bit rates of 64 Kbits/s. Packetisation rates of 10 ms to 50 ms give good speech quality [5, 40, 42]. A maximum packet delay must be imposed to ensure good quality reproduction. The public telephone network has a maximum delay of 600 ms [20]. In a voice conversation, participants typically take turns talking in spurts, interspersed with short periods of silence. These silent periods can be suppressed and not transmitted to save network bandwidth. However, when speech is compressed this way at the sending station it must be reconstructed at the receiving end. Commonly used methods are discussed in [2, 40].

Digital video images are created by sampling a scene at regular intervals to produce a series of frames. Simple sampling produces packets at regular periods, but video images are highly redundant and are usually compressed before transmission. Most common compression techniques result in variable bit rate outputs, because the amount of compression depends on the complexity of the images and the amount of change between frames. Broadcast quality video can be transmitted at 15-30 Mbits/s [28, 44], while teleconferencing requirements are as low as 56 Kbits/s [13].

5.4.2 Transmission Strategy

Synchronous data applications are sensitive to variations in data delivery delays. For example, video frames of a moving scene must arrive at regular intervals to faithfully reproduce the image[3]. The variation in arrival delay is called *jitter*. The DSMA/S protocol is particularly well suited to synchronous data applications. Unlike FDDI, where the token rotation time can vary depending on the traffic load, DSMA/S possesses extremely regular token movement. This regularity can be used to provide a low jitter synchronous transmission service. There are three choices for a transmission strategy for synchronous traffic.

Transmit Only During Token Possession

A simple transmission strategy for synchronous traffic is for stations to transmit synchronous data only when they have possession of the token. Other transmission opportunities would only allow asynchronous data to be sent. Asynchronous messages could also be sent during token possession if no synchronous messages were awaiting transmission. Because no packet re-use would take place for synchronous data, the maximum throughput for synchronous data would be limited to

$$\left(1 - \frac{L}{\mathrm{TTRT}}\right) \times \left(1 - \frac{h}{l_s}\right)$$

for slots of length l_s with headers of length h, that is, the same throughput as FDDI. Furthermore, adding more loops to the spiral would not increase synchronous throughput: the additional ports on the extra loops would not transmit synchronous data until they received the token[4]. There are two advantages to this strategy:

1. very low jitter due to the extremely regular rotation time of the token,

2. no packet loss if the TTRT is less than the maximum packet delay of the synchronous traffic (provided the *volume* of traffic does not exceed the network's capacity).

[3] The methods used to decode the compressed digitised images help to relax this requirement somewhat.

[4] However, additional bandwidth would be available for asynchronous traffic.

Transmit At All Opportunities

Another strategy is simply to treat synchronous traffic in the same way as asynchronous traffic, but give it priority access. Synchronous messages could be transmitted whenever there is an opportunity according to the transmission rules. Asynchronous traffic would have to wait until there were no synchronous messages queued. This approach would result in maximum bandwidth availability for synchronous traffic, but because of the possibly irregular access times, the highest jitter. Also, it would be possible for synchronous traffic to completely preempt asynchronous messages.

Transmit During Token Possession and Slot Re-use

A third approach is a compromise. Synchronous transmissions would be allowed during token possession and at all the predictable transmission windows as described in section 4.5.2, by stations re-using their own slots. Recall that if a station transmits m slots at time t from a primary port, it is guaranteed to get a transmission window of m slots at time $t + L$. Also, as discussed in section 4.5.4, under the right conditions it is possible to have not only predictable, but *regular* transmission windows. This is a useful property for synchronous transmission.

We now discuss the conditions necessary to take advantage of this property. The necessary conditions for a station to receive regular-interval transmission windows of size m between token captures are:

1. The sum of the token-holding times is a divisor of $\mathcal{K} \times L$. We showed in section 4.5.4 that this condition results in an integer number of slot rotations before hitting the token, and that it can lead to transmission windows at regular time intervals.

2. It must transmit m full slots during token possession.

3. It must re-use the same m slots whenever they pass by one of the station's ports[5].

[5] Subject to the transmission rules, of course.

4. In order to restrict transmission to these regular intervals, the station must distinguish between slots transmitted during these regular intervals and those transmitted at other times. The timer and event-queue scheme described in section 4.4.1 could be adapted to do this: events that happen at multiples of W after the token capture are regular transmission windows and can be used for synchronous messages.

Unfortunately, the above conditions are not likely to be met. For example, if a station is not under very heavy load, it may not have any messages to transmit during its token holding time or during one of its re-use opportunities. If condition 2 is not met, then according to the transmission rules, another station can transmit into the resulting empty slots. This means that slots left empty by a station are not guaranteed to be available when they pass the station's next port[6]. The timing of transmission windows could now be irregular until sometime after the token is captured again.

This strategy would result in more available synchronous bandwidth than the *token-only* strategy and less jitter than the *transmit-anytime* strategy described above.

5.4.3 Synchronous Traffic Model

In our simulation model, we define a *message stream* as a synchronous data source with a constant sender and destination address pair, a fixed packet size, fixed packet inter-arrival time, and a maximum packet access time. Packets that are not transmitted before the maximum packet access time are considered lost.

We chose to use a simple model of packetised voice traffic to study the performance of the DSMA/S protocol under synchronous traffic conditions[7]. We modeled packetised voice calls as separate message streams. Though simplified, our model uses more demanding traffic parameters than are necessary in real life:

- Silent periods are not suppressed. Rather we transmit packets at regular intervals, whether they contain voice-data or silence.

[6] Of course, a station could effectively reserve the slots by marking them *busy* anyway, but some bandwidth would be wasted.

[7] Fixed-bit-rate video could also have been considered.

Parameter	Model	Simulator
Transmission rate	1 Gb/s	NA
Propagation speed	2×10^8 m/s	NA
Network length	100 km	1200 slots
Ring Latency	5×10^{-4} sec	1200 slots
Packet arrival rate	10 ms	23500 slots
Voice packet length	640 bits	2 slots[9]
Number of stations	100	100
Station separation	1 km	12 slots
Maximum TTRT	10 ms	23500 slots
Maximum THT	9.5×10^{-5} sec	223 slots

Table 5.2: Parameters for Synchronous Traffic Experiments

- The bit rate was set at 64 Kb/s.
- Packets are produced every 10 ms.
- The maximum packet access time is 10 ms.

We used a network transmission rate of 1 gigabit/s, in a 100 km[8] network of 100 stations. The stations were equally spaced on the network. Each station was assigned an equal number of message streams when possible. Otherwise, the number of streams per station differed by no more than one. Since the units of length and time in the simulation environment are slots, the simulation parameters were sometimes rounded to the nearest reasonable value. For example, a distance of 100 km corresponds to 1179.245 slots - we chose 1200 slots, which corresponds to 101.76 km. The parameters of the model and the resulting simulation parameters are summarised in Table 5.2.

5.4.4 Synchronous Traffic Experiments

To investigate the performance of DSMA/S under synchronous traffic conditions, we measured throughput and packet loss under various intensities of synchronous

[8] This is the length of each loop in the spiral network.

[9] Only 1.66 slots are needed for a 640-bit voice packet.

traffic. Packet loss occurs when some part of a synchronous message is not transmitted before the maximum packet access time allowed for the synchronous traffic stream. Groups of experiments were conducted with spiral networks consisting of 1, 2, 3, and 5 loops. Within each group, the synchronous traffic load was varied by changing the number of synchronous message streams. We used the transmission strategy described in section 5.4.2 which allows synchronous transmissions during token possession or during slot re-use. The other simulation parameters were presented earlier in Table 5.2.

The effective throughput should increase linearly with increases in the synchronous message streams up to network saturation. However, the maximum theoretical throughput will not be achieved because the parameters chosen are not optimal for throughput[10] as explained in section 4.10.

Packet loss should only occur near network saturation. All message streams are distributed uniformly between the stations. The maximum number of message streams S_{max} that should be serviceable without packet loss is related to the network's maximum bandwidth, synchronous message length l_s and synchronous message interarrival time T_s. The network's maximum bandwidth[11] is $\mathcal{K}$. Thus the maximum number of slots that can be transmitted in T_s is $\mathcal{K} \times T_s$, and the maximum number of message streams is:

$$S_{max} = \frac{\mathcal{K}T_s}{l_s}. \tag{5.1}$$

Synchronous messages arrive at the packetisation rate of 10ms which corresponds to $T_s = 23584$ slots. The synchronous message length $l_s = 2$ slots. For these parameters, the theoretical message stream limits for networks with 1, 2, 3, and 5 loops are given in Table 5.3.

5.5 Mixed Traffic Conditions

Experiments were performed to investigate the interaction between synchronous and asynchronous traffic in the DSMA/S network. We simulated 100-kilometre

[10] They were chosen to satisfy the timing requirements of our synchronous model.

[11] The *effective throughput* is lower because it does not count the slot headers.

$\mathcal{K}$	1	2	3	5
S_{max}	11792	23584	35376	58960

Table 5.3: Maximum Synchronous Message Streams

networks with 100 stations spaced equally along the loops. Since synchronous traffic gets priority transmission, we investigated the impact of synchronous traffic load on the throughput and delay for asynchronous messages.

Asynchronous throughput was measured with increasing synchronous traffic for networks with 1, 2, 3, and 5 loops. The characteristics of the asynchronous and synchronous traffic were the same as those in the earlier experiments, where each type of traffic was investigated independently. Asynchronous message interarrival times were chosen to present a saturated asynchronous load, while the synchronous load was varied.

In studying asynchronous message delay, a network with two loops was simulated. The synchronous traffic load was fixed at different levels. Then, for each fixed synchronous load level, simulation runs were performed for a range of asynchronous traffic loads and the corresponding message access times were measured. Five load levels for the synchronous traffic were chosen: 0%, 25%, 50%, 75%, and 90% of the maximum load. The maximum load was defined to be the number of synchronous message streams which resulted in a packet loss of 0.5% in the earlier experiments. The results are discussed in Section 6.3.

Chapter 6

Simulation Results

This chapter presents the results of the simulation experiments described in the previous chapter.

6.1 Asynchronous Traffic Conditions

The performance characteristics of four spiral networks carrying asynchronous traffic are shown in Figures 5 and 6. Each figure shows curves for spirals with 1, 2, 3 and 5 loops with lengths of 1Mb per loop. The simulations modeled networks with loops containing 2496 slots[1] and 32 stations. To allow a fair comparison between networks with different numbers of loops, the stations in each network were assigned a THT that was optimal for the network length and loop-count.

Both figures show the *mean message waiting time* as the *effective throughput* varies, but in Figure 6, the throughput figures were normalised to a proportion of the maximum theoretical throughput for each spiral network.

Note the message waiting time when the traffic load is light (i.e.: at lowest throughput). For the multi-loop spirals, the waiting time is in the neighbourhood of 11 slots. This measurement includes the time it takes to transmit the entire message. Since the minimum slot access time is a half-slot and the mean message length in these experiments was 11 slots, the observed delays are clearly the

[1] This is equivalent to 1.058 Mbits per loop.

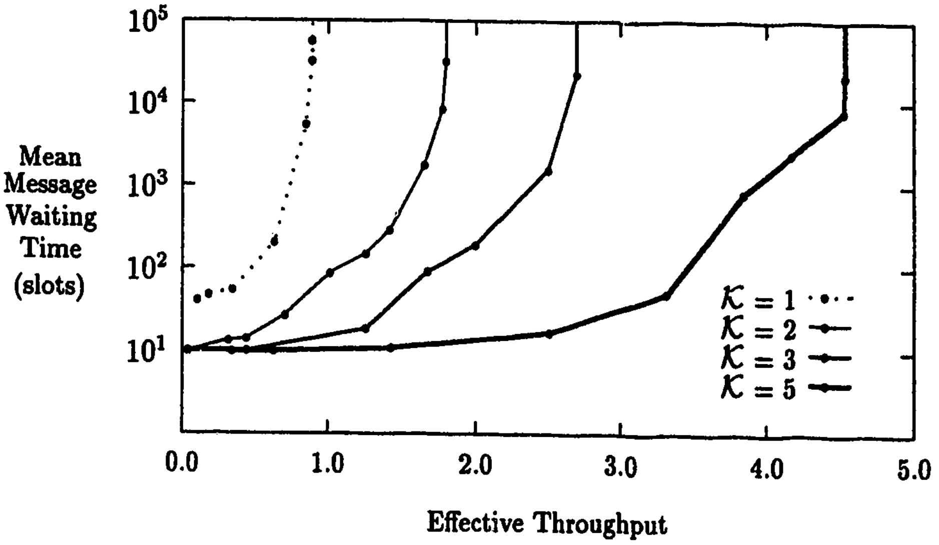

Figure 5: Asynchronous Traffic Performance (1 Mb Loops, 32 Stations)

smallest possible. This demonstrates that in multi-loop spirals under light loads, DSMA/S allows transmissions to begin with a small half-slot delay.

The single loop network does not possess this property. The minimum message access delay was 40.6 slots. Equation 4.6 in Section 4.11 gives the theoretical *slot access delay* under light loads. Using the simulation parameters above, the equation gives an average slot delay of 29.3 which, when augmented by the 11 slots per message, yields a mean message waiting time of 40.3, which agrees closely with the simulation results.

As Figure 6 shows, all of the spiral networks achieve 98% of their theoretical maximum throughput.

6.2 Synchronous Traffic Conditions

Our simulation experiments measured synchronous throughput and packet loss of spiral networks with 1, 2, 3, and 5 loops under varying synchronous traffic loads.

Figure 7 shows effective throughput for synchronous traffic as a function of the synchronous load. To compare between spirals with different numbers of loops, the

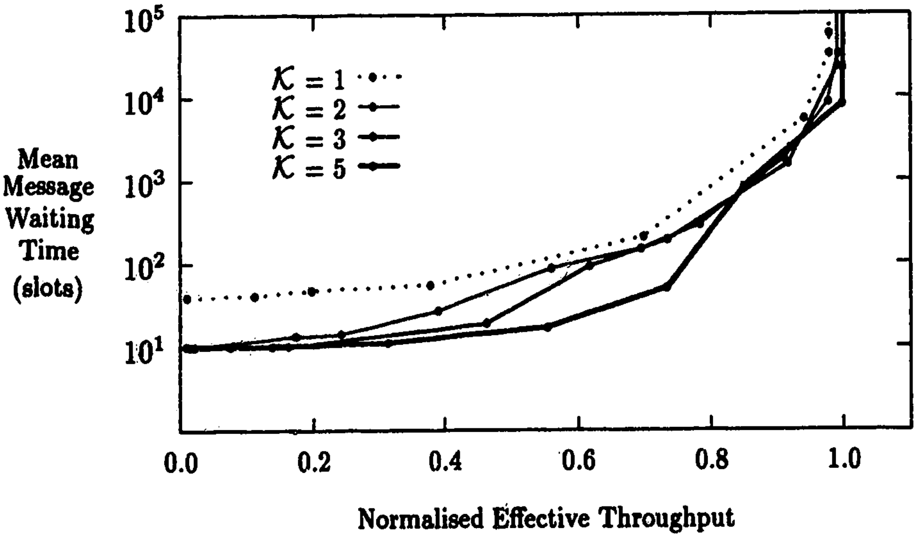

Figure 6: Asynchronous Traffic Performance Showing Normalised Throughput (1 Mb Loops, 32 Stations)

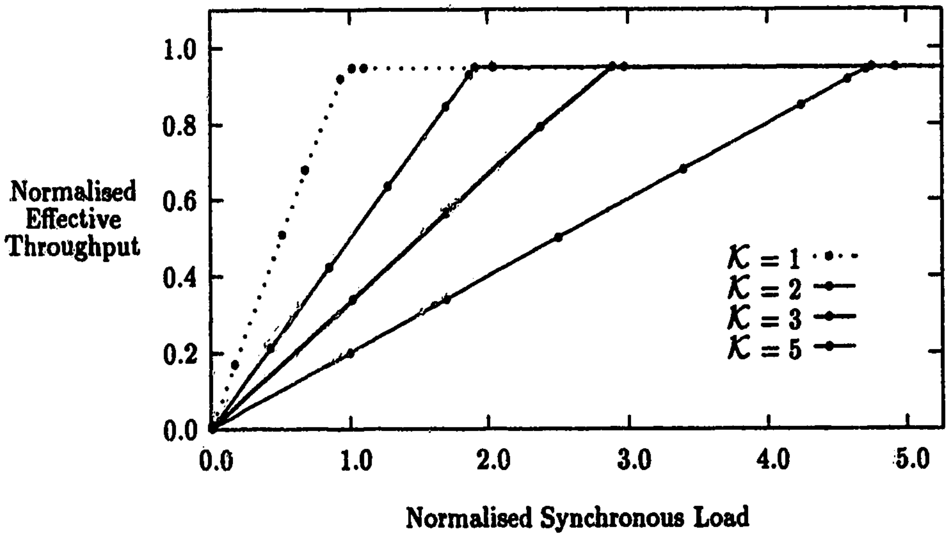

Figure 7: Synchronous Throughput at Synchronous Traffic Loads

$\mathcal{K}$	C_{obs}	C_{max}	C_{obs}/C_{max}
1	0.86	0.91	0.95
2	1.72	1.81	0.95
3	2.58	2.72	0.95
5	4.30	4.53	0.95

Table 6.4: Observed and Theoretical Throughput for Synchronous Traffic

throughput was normalised to the theoretical maximum[2] C_{max} for each spiral net. Similarly, the synchronous load was normalised to the maximum number of synchronous message streams for a single-loop network. The normalised synchronous load $\bar{S}$ is related to the number of message streams S by $\bar{S} = S/S_{max,1}$. From equation 5.1 in section 5.4.4, the maximum serviceable number of synchronous message streams for a $\mathcal{K}$-loop network is:

$$S_{max,k} = \frac{\mathcal{K}T_s}{l_s}.$$

Thus, the maximum normalised synchronous load for a $\mathcal{K}$-loop network is:

$$\bar{S}_{max,k} = \frac{S_{max,k}}{S_{max,1}} = \mathcal{K}.$$

As expected, the throughput increased linearly (until saturation) with increasing synchronous load.

The observed maximum throughput C_{obs} is compared with C_{max} for each network in Table 6.4. Each network achieved 95% of the maximum theoretical throughput. The loss of 5% is due to the chosen THT value not being optimal for throughput, as explained earlier in Section 5.4.4.

Figure 8 shows packet loss for networks with 1, 2, 3, and 5 loops as a function of the normalised synchronous traffic load. Packet loss does not occur until the traffic load approaches S_{max}, the theoretical maximum number of synchronous message streams that each network can service. In Section 5.4.4 we calculated the S_{max} values for each network and presented them in Table 5.3. The normalised

[2] From section 4.9.

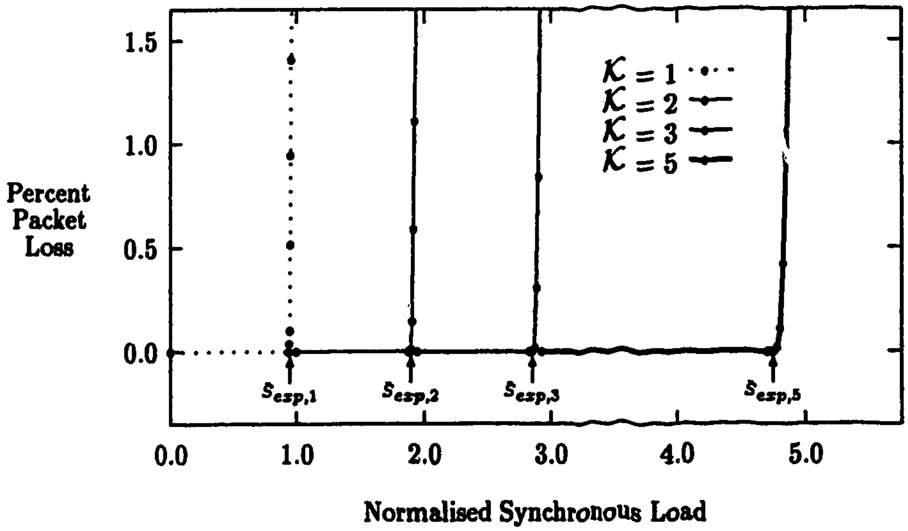

Figure 8: Packet Loss at Synchronous Traffic Loads

load $\bar{S}$ is based on $S_{max,1}$. However, these S_{max} values assume that network parameters are optimal, and that maximum theoretical throughput C_{max} can be achieved, which is not the case in these experiments. Consequently, we should see packet loss occurring at loads somewhat lower than $\bar{S}_{max}$. Since throughput was not optimal, we estimate the *expected* maximum normalised load $\bar{S}_{exp}$ for each $\mathcal{K}$. To obtain $\bar{S}_{exp,\mathcal{K}}$ we multiply $\bar{S}_{max,\mathcal{K}}$ by the normalised maximum observed throughput:

$$\bar{S}_{exp,\mathcal{K}} = \frac{C_{obs,\mathcal{K}}}{C_{max,\mathcal{K}}} \times \bar{S}_{max,\mathcal{K}} \approx \frac{C_{obs,\mathcal{K}}}{C_{max,\mathcal{K}}} \times \mathcal{K} .$$

Since $C_{obs,\mathcal{K}}/C_{max,\mathcal{K}} = 0.95$ for each $\mathcal{K}$, then $\bar{S}_{exp,\mathcal{K}} = 0.95\mathcal{K}$. In Figure 8, the $\bar{S}_{exp,\mathcal{K}}$ values are indicated by arrows.

6.3 Mixed Traffic Conditions

We now present the results of experiments showing the interaction of synchronous and asynchronous traffic and the effects on performance. The experiments were described in section 5.5.

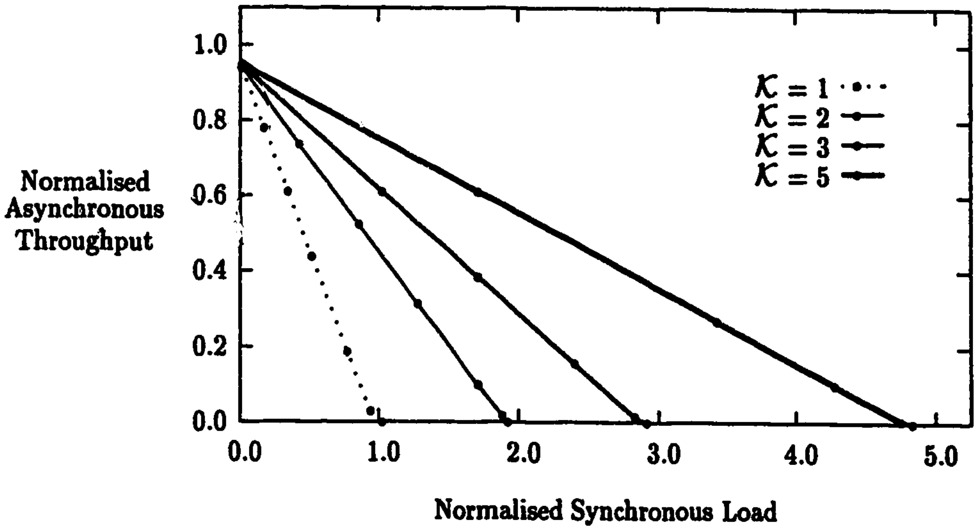

Figure 9: The Effect of Synchronous Traffic on Asynchronous Throughput. ($\mathcal{K}$=2, 100 km, 100 Stations)

6.3.1 Asynchronous Performance

Recall that synchronous messages get priority access over asynchronous messages. Figure 9 shows asynchronous throughput curves for networks with 1, 2, 3, and 5 loops. As expected, the asynchronous throughput decreases linearly with increasing synchronous load. For all networks, asynchronous throughput is maximum for low synchronous loads and decreases towards zero as synchronous load increases toward saturation.

The effects of different load conditions on asynchronous performance in a two-loop network is shown in Figure 10. Performance curves are shown for five levels of synchronous load l_s: 0, 0.25, 0.50, 0.75, and 0.90 of $\bar{S}_{max}$, the maximum synchronous load. Asynchronous mean message delay is plotted against normalised asynchronous throughput for each synchronous load level. As expected, maximum asynchronous throughput is lower during heavier synchronous loads. Similarly, the asynchronous message delay at light asynchronous loads increases when the synchronous traffic is heavy. For the l_s = 0 curve, the message delay at low

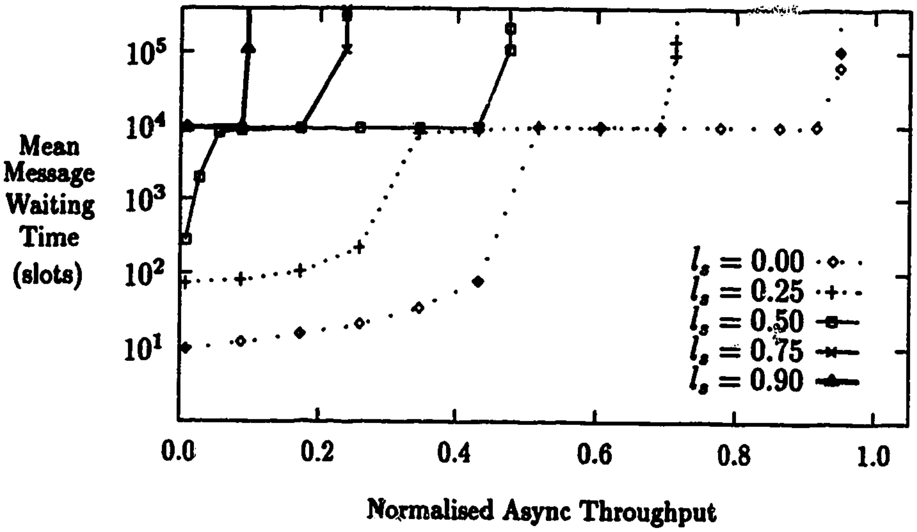

Figure 10: Asynchronous Performance at Different Synchronous Loads. ($\mathcal{K}$=2, 100 km, 100 Stations)

asynchronous loads is on the order of the mean message length of 11 slots. When the synchronous load is 0.90, the minimum asynchronous message delay is on the order of 10^4 slots.

The most interesting aspects of Figure 10 are the plateaux in all of the curves at the message delays of 10^4 slots. The flat portions of the curves are due to the use of previously stripped slots by the secondary ports. As load increases throughout the plateau region of the curves, the station uses more of the empty slots passing its secondary port.

Consider the following example. A station receives the token and transmits as many slots as it can: at medium loads, this could empty its message queue. At a time L later, the slots pass by the station's other port, (which is now in the primary state because the station has released the token). The slots are stripped, and most of them continue down the carrier empty, because the original station has not accumulated many messages during this short period of L. (Recall, the parameters in our model are such that $L << \text{TTRT}$.) Most of the slots remain empty until they reach the first secondary port that is enabled. This is because

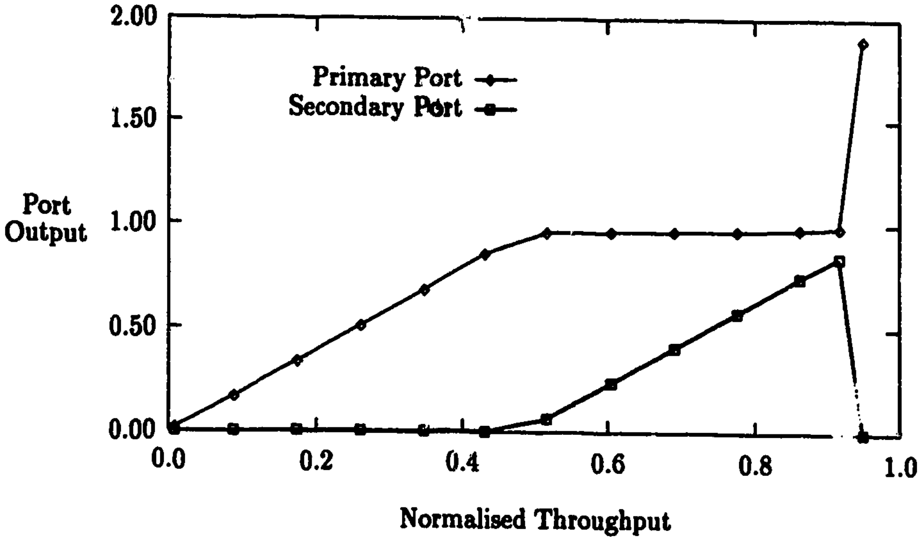

Figure 11: Output of Primary and Secondary Ports During Asynchronous Traffic. ($\mathcal{K}$=2, 100 km, 100 Stations)

the intervening ports belong to stations that have also just possessed the token, and have transmitted most of their messages. However, the station possessing the first secondary port that is enabled is expecting the token shortly, and has accumulated many messages for transmission. It makes use of whatever empty slots pass its secondary port. As traffic load increases, more of the secondary port slots are used.

Figure 11 shows that the increased load is transmitted on the secondary ports. The *output*[3] from primary and secondary ports is plotted against throughput. The same experiments were plotted on the $l_s = 0$ curve of Figure 10 which shows the asynchronous message delays when no synchronous traffic was present; the curve is reproduced in Figure 12 for convenience. Note that in Figure 11 the secondary port output increases while the primary port output remains static during the plateau region of the curve in Figure 12.

The asynchronous message delay plateaux are related to the TTRT parameter. Figure 13 compares asynchronous traffic performance for TTRTs of 10 ms and 1

[3] We define *output* as the number of transmitted slots per unit time.

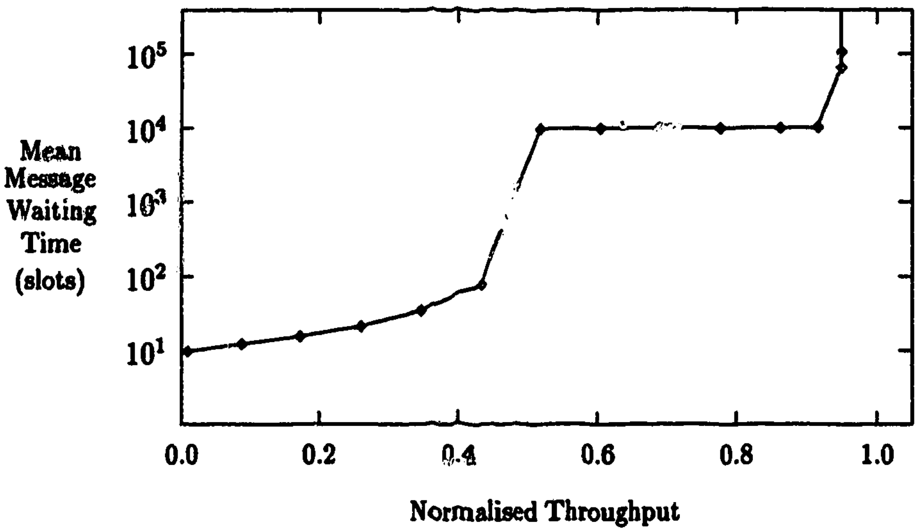

Figure 12: Asynchronous Performance With No Synchronous Load. ($\mathcal{K}$=2, 100 km, 100 Stations)

ms. The synchronous traffic load was fixed at 50%.

6.3.2 Synchronous Performance

Synchronous traffic performance was studied for a two-loop network under five levels of asynchronous load l_a: 0, 0.25, 0.5, 0.75, and 0.9. Message delay and throughput curves for the synchronous traffic are shown in Figure 14. The five curves are so similar that they overlap, demonstrating that synchronous traffic performance is virtually unaffected by asynchronous traffic levels. Note also that the synchronous mean message delay is almost constant throughout all traffic load combinations; a valuable property for synchronous applications. As Figure 15 shows, these delay characteristics are affected only slightly when a lower TTRT is used. The curves show synchronous message delay and throughput performance using a TTRT of 10 ms and 1 ms during a 50% asynchronous load condition.

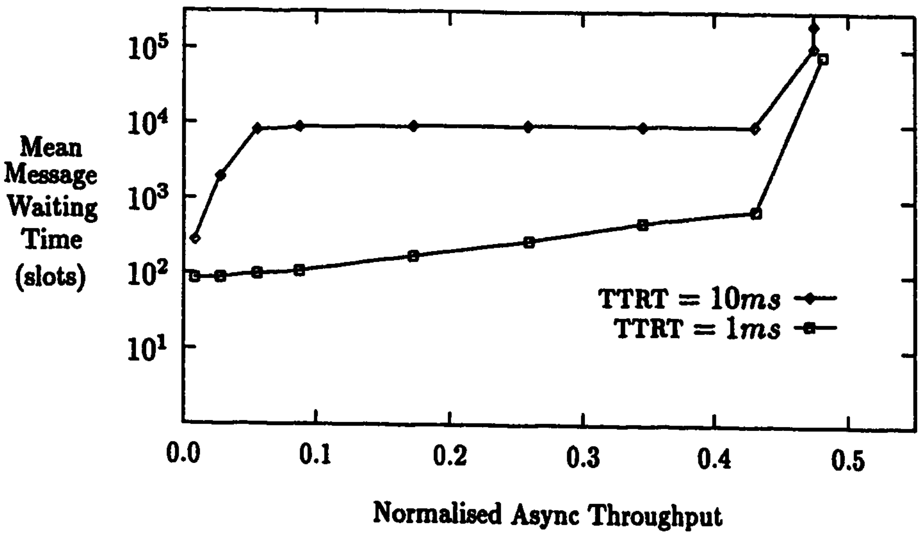

Figure 13: Asynchronous Traffic Performance at Different TTRTs. ($\mathcal{K}$=2, 100 km, 100 Stations, at 50% Synchronous Load)

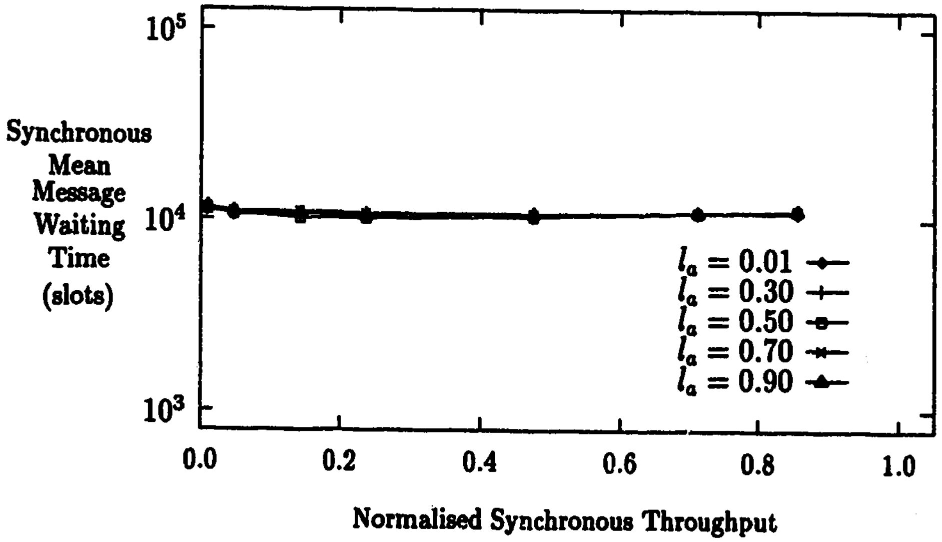

Figure 14: Synchronous Traffic Performance at Different Asynchronous Loads. ($\mathcal{K}$=2, 100 km, 100 Stations)

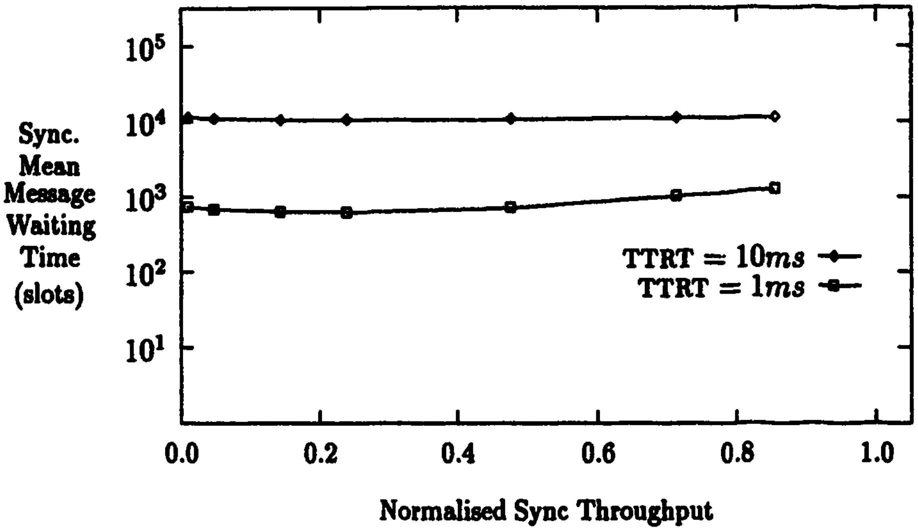

Figure 15: Synchronous Traffic Performance at Different TTRTs. ($\mathcal{K}$=2, 100 km, 100 Stations, at 50% Asynchronous Load)

Chapter 7

Conclusions

The spiral-ring topology and the DSMA/S medium access protocol were described and analysed. A simulation model was developed and experiments were conducted to investigate network behaviour under various traffic load conditions. DSMA/S possesses many of characteristics of an ideal MAC-level protocol for high-speed networks:

- The transmission rules are simple and could be implemented in hardware to exploit the bandwidth that is available in optical fibre carriers.

- The token-passing mechanism guarantees fair access for the stations. No single station is designated to perform any special function, so access is independent of relative location.

- DSMA/S is a *capacity-1* protocol: the total throughput is independent of the network size. Our analysis showed that the maximum effective throughput is achievable when the ratio of $\mathcal{K}L/W$ is an integer.

- Under light-load conditions, the average medium access time is as low as possible for a slotted protocol — half a slot.

- The protocol does not rely on feedback from other stations, which would introduce delays proportional to the network size.

- Both asynchronous and synchronous traffic are accommodated. The fixed-time token-passing scheme guarantees a finite maximum packet delay for synchronous data. Asynchronous data is transmitted in slots not used for synchronous data.

- Synchronous traffic of variable intensity can be accommodated. Bandwidth allocation is inherently determined by the synchronous traffic demand up to the full capacity of the network.

- The slotted nature of the protocol and the token-passing mechanism make the protocol self-synchronising, which is a requirement for low *jitter* in synchronous data.

- The regularity of the token-passing mechanism and the slot reuse scheme make the protocol predictable. A failure that disrupts this predictable behaviour would be noticed by at least one of the stations in a time smaller than L.

This study used a simple model of synchronous traffic. The reader is encouraged to investigate the protocol's performance using a more comprehensive model of synchronous data which includes variable-bit-rate sources.

The *jitter* in synchronous traffic service should also be measured. In Section 4.5.4, we showed that regular transmission windows are possible at intervals of W if the network parameters are such that KL/W is an integer. Employing the transmission strategy used in this study (i.e., restricting synchronous transmissions to token-possession and own-slot reuse) should provide a very regular, low jitter service for synchronous data.

Bibliography

[1] American National Standards Institute. *FDDI - Hybrid Ring Control (HRC), Draft ANSI X3186-199x Rev. 6.2,*, May 14, 1991.

[2] C. Barberis and D. Pazzaglia. Analysis and optimal design of a packet-voice receiver. *IEEE Trans. Commun.*, 28:217–227, February 1980.

[3] IEEE Standards Board. *IEEE Standards for Local Area Networks: Token Ring Access Method and Physical Layer Specifications.* The Institute of Electrical and Electronic Engineers, Inc., New York, 1985.

[4] A. Bondavalli and L. Strigini. DSDR: a fair and efficient access protocol for ring-topology MANs. In *IEEE INFOCOM '91*, pages 1022–1030, 1991.

[5] T.M. Chen, J. Walrand, and D.G. Messerschmitt. Dynamic priority protocols for packet voice. *IEEE Select. Areas in Commun.*, 7(5):632–643, June 1989.

[6] Greg Dejager. FDDI-II overview. In *Silicon Valley Networking Conference '92*, pages 7–16, 1248 Olive Branch Lane, San Jose, CA 95120, April 1992. SysTech Research.

[7] W. Dobosiewicz and P. Gburzynski. The spiral ring network. (unpublished paper), June 1992.

[8] W. Dobosiewicz and P. Gburzynski. A survey of network topologies and protocols. (unpublished paper), June 1992.

[9] Wlodek Dobosiewicz, Pawel Gbrzynski, and Paul McWeeny. On a new variant of the double ring topology for very fast lans and mans. In *Silicon Valley*

Networking Conference, pages 421–427, 1248 Olive Branch Lane, San Jose, CA 95120, 1992. SysTech Research.

[10] Pawel Gburzynski and Piotr Rudnicki. LANSF: a protocol modelling environment and its implementation. *Software Practice and Experience*, 21(1):51–76, January 1991.

[11] E.L. Hane, A. Choudhury, and N.F. Maxemchuk. Improving the fairness of DQDB networks. In *Proc., IEEE INFOCOM '90, San Francisco, CA*, pages 175–184, June 1990.

[12] E.Y. Huang and L.F. Merakos. On the access fairness of the DQDB MAN protocol. In *Proc., IEEE 1990 Phoenix Conference, Phoenix, AZ*, pages 556–559, June 1990.

[13] Shan-Shan Huang. Source modeling for packet video. In *IEEE ICC'88*, pages 1267–1276, 1988.

[14] IEEE. *Proposed Standard: Distributed Queue Dual Bus Subnetwork of a Metropolitan Area Network, P802.6/D15.* Institute of Electrical and Electronic Engineers, October 1990.

[15] P. Jaquet and P. Muhlender. Machnet: A simple access protocol for high speed or long haul communications. In *SIGCOMM '90 Symposium*, pages 90–100, September 1989.

[16] A.E. Kamal. Efficient multi-segment message transmission with slot reuse on DQDB. In *Proc., IEEE INFOCOM '91*, pages 869–878, April 1991.

[17] Gary C. Kessler and David A. Train. *Metropolitan Area Networks: Concepts, Standards and Services.* McGraw-Hill Inc., New York, 1992.

[18] K.M. Khalil and M.E. Koblentz. A fair distributed queue dual bus method. In *Proc., IEEE 14th Conf. on Local Computer Networks, Minneapolis, MN*, pages 180–188, October 1989.

[19] Donald E. Knuth. *The Art of Computer Programming*, volume 2. Addison-Wesley Publishing Co., Reading, Massachusetts, 1981.

[20] Bell Telephone Labs. *Engineering and Operations in the Bell System.* AT&T Bell Labs, 1984.

[21] Y.-S. Leu and D.H.C. Du. Cycle compensation protocol: a completely fair protocol for the unidirectional twin-bus architecture. In *Proc., IEEE 15th Conf. on Local Computer Networks, Minneapolis, MN*, October 1990.

[22] J. Limb and C. Flores. Description of Fasnet, a unidirectional local area communications network. *Bell Systems Technical Journal*, September 1982.

[23] J.O. Limb. A simple multiple access protocol for metropolitan area networks. In *ACM '90 SigComm Conf.*, September 1990.

[24] W.T. Marshal and S.P. Morgan. Statistics of mixed data traffic on a local area network. *Computer Networks and ISDN Systems*, pages 185–194, October 1985.

[25] N.F. Maxemchuk. Twelve random access strategies for fiber optic networks. *IEEE Trans. Comm.*, 36(8):942–950, 1988.

[26] H. Mueller, M. Nassehi, and J. Wong. DQMA and CRMA: new access schemes for gbit/s LANs and MANs. In *IEEE INFOCOM'90*, pages 185–191, 1990.

[27] B. Mukherjee and S. Banerjee. Alternative strategies for fairness in and an analytical model of DQDB networks. In *Proc., IEEE INFOCOM '91*, pages 879–888, April 1991.

[28] H. Murakami, S. Matsumoto, Y. Hatori, and H. Yamamoto. 15/30 mbits/s universal digital tv codec using a median adaptive predictive coding method. *IEEE Tran. Commun*, 35:637–645, June 1987.

[29] M. Nassehi. CRMA: an access scheme for high-speed LANs and MANs. In *IEEE Int. Conf. Communications*, pages 1697–1702, April 1990.

[30] P.F. Pawlita. Traffic measurement in data networks, recent measurements and results and some implications. *IEEE Tran. Commun.*, 29(4):525–535, April 1981.

[31] Raphael Rom and Moshe Sidi. *Multiple Access Protocols - Performance and Analysis.* Springer-Verlag, New York, 1990.

[32] F.E. Ross. FDDI - a tutorial. *IEEE Comm. Mag.*, 24(5):10–17, 1986.

[33] Ghassan Semaan. Isochronism and asynchronism: methods and performance of HSLANs. *Computer Networks and ISDN Systems*, 21:299–306, 1991.

[34] K.C. Sevcik and M.J. Johnson. Cycle time properties of the FDDI token ring protocol. *IEEE Trans. Software Eng.*, 13(3):376–385, 1987.

[35] J.F. Shock and J.A. Hupp. Measured performance of an ethernet local network. *Communications of the ACM*, 23(12):711–721, December 1980.

[36] M. Spratt. Allocation of bandwidth in IEEE 802.6 with non-unity ratio bandwidth balancing. In *Proc., IEEE Intl. Conf. Commun (ICC) 91*, pages 729–735, June 1991.

[37] M. Spratt. Bandwidth allocation over several networks with DQDB. In *Proc., EFOC/LAN '91 Conf., London, UK*, June 1991.

[38] William Stallings. *Local Networks - An Introduction.* Macmillan Publishing Company, New York, 1984.

[39] William Stallings. *Data and Computer Communications,.* Macmillan Publishing Company, New York, 3rd edition, 1991.

[40] T. Suda, H. Miyahara, and T. Hagesawa. Performance evaluation of a packetized voice system-simulation study. *IEEE Tran. Commun*, 32(1):97–102, January 1984.

[41] Andrew S. Tenenbaum. *Computer Networks.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[42] C.J. Weinstein and J.W. Forgie. Experience with speech communication in packet networks. *IEEE J. Select. Areas in Commun.*, 1(6):963–980, December 1983.

[43] J.W. Wong. Throughput of DQDB networks under heavy load. *EFOC/LAN '89, Amsterdam*, pages 146–151, June 1989.

[44] H. Yamamoto, Y. Hatori, and H. Murakami. 30 mbit/s codec for ntsc color tv signal using interfield-intrafield adaptive prediction. *IEEE Tran. Commun.*, 29:1859–1867, December 1981.