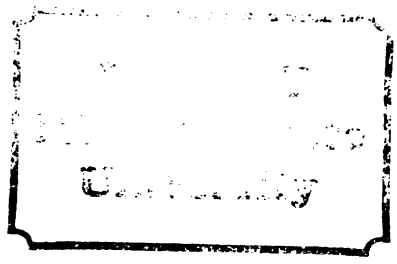
THES...



This is to certify that the

dissertation entitled

THE ANALYSIS OF TWO VIRTUAL-TOKEN PROTOCOLS
FOR LOCAL AREA COMPUTER NETWORKS

presented by

Liang Li

has been accepted towards fulfillment
of the requirements for

Ph.D degree in Computer Science

Major professor

Date Jan. 26, 1982

THE ANALYSIS OF TWO VIRTUAL-TOKEN PROTOCOLS FOR LOCAL AREA COMPUTER NETWORKS

By

Liang Li

A DISSERTATION

Submitted to
Michigan State University
in partial fulfilment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science

1982

ABSTRACT

THE ANALYSIS OF TWO VIRTUAL-TOKEN PROTOCOLS
FOR LOCAL AREA COMPUTER NETWORKS

By

Liang Li

Local area computer networking is fast becoming an area of intense research for its wide range of applications. To date, many different types of local networks have been proposed or developed. These network designs can roughly be categorized according to their design purposes, network topologies, transmission media, switching techniques, and communication control strategies. Among these designs, the decentralized multi-access schemes with a global databus topology provides one of the most popular approaches. It was recognized, however, that a serious trade-off exists between efficiency and control overhead for the many multi-access protocols. Additionally, most of these protocols are very sensitive to the network propagation delay and/or the size of the user population.

In this thesis, we look into two new network protocols which employ a virtual-token concept to reduce the efficiency/overhead trade-off, and to support a large number of users under a wide range of bus propagation delays. The shortest-delay access method (SDAM) minimizes the network control's changeover time between two consecutive transmissions. The group broadcast recognizing access method (GBRAM), on the other hand, utilizes the clusters of user machines on the network to construct a two-level scheduling function which reduces the control overhead.

Analysis shows that both these protocols are efficient, stable, reliable, and easy to implement. They drastically improve the delay performance of the conventional token passing schemes; and their non-exhaustive transmission option guarantees an upper bound to the node's response time, which is particularly desirable for voice-data transmissions and real-time applications. The performances (i.e., throughput-delay relationship, channel capacity, and offered load to throughput relationship) of SDAM and GBRAM are compared with the performances of some of the most popular protocols and the M/D/1 perfect scheduling in this thesis.

To My Wife, Susie

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to Herman Hughes, who introduced me to the field of local computer networks, and supported me throughout the years; to Lewis Greenberg, who provided many inspirations and insights to this work; to Anil Jain, who brought me into the Ph.D program at Michigan State University in the first place; to Lionel Ni, who provided many valuable discussions and suggestions; and to R. V. Erickson, who helped a great deal in the statistical aspects of this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Evolution of Local Networks

The invention of computer networking has been regarded by many as one of the major milestones in the history of computer evolution, much the same way as the invention of telephone to human civilization [Mar81]. Prior to the 70's, most of the computer systems that existed were stand-alone systems, each providing limited computing resources and services to its immediate surrounding. With the advancements of software and hardware technologies in the 70's, the idea of computer networking has become more and more widespread in industry, business, government, education, and the like. Today, the majority of the computer installations participate in some form of computer

netwo

machi

large

[Sym

reli

vari

buil

netw

and

With

and

proc

auto

prol

to c

1.2

comp

geog

Gene

[Th

networking, may it be a local connection to a front-end machine or data concentrator, or a link to a nation-wide large-scale network such as ARPAnet [McQ77] or TYMNET [Tym71], etc.

In recent years, due to the growing demands for a reliable, high-speed, inexpensive communication between various machines in a close vicinity (e.g., within a building or cluster of buildings), local area computer networking has emerged from the general computer networking and has become an area of intense interest in its own right. With the new technologies (e.g. LSI and VLSI technologies) and potential applications (e.g. distributed parallel processing, back-end file/storage sharing, and office automation, etc.), it is only reasonable to expect a proliferation of local area computer networks in the decades to come.

1.2 Distinctions Between Long-haul and Local Networks

While no clear boundary exists between a local area computer network (LACN or LCN) and an otherwise geographically distributed ("long-haul") network, a LACN can generally be characterized by the following properties [Thu79a,Cot80,Cla78,Chl79b]:

1. generally local; i.e., the network spans on the order of a few miles or less;

2. generally owned and operated by a single organization;
3. usually has a high bandwidth (e.g., 1 Mbits/second and up) over an inexpensive transmission medium;
4. the devices (nodes) which are connected to the network could be either active (intelligent) or passive (non-intelligent).

Many of the technologies developed in long-haul networks have been applied to the local networks, e.g., the store-and-forward packet-switching concept, and the IMP (interface message processor) [Hea70] of the ARPAnet, etc.. However, the characteristics listed above render the local network a unique operating environment in which short propagation delays and high data rates at low costs are typical, thus giving rise to the simplification of network control structure and many new communication protocols [Cla78].

1.3 Problem Statement

To date, many different types of local computer networks have been developed or proposed [Thu79a,Luc78,Tob80,Tro80]. These networks can roughly be categorized according to their (1) design purposes, (2) network topologies, (3) transmission media, (4) switching techniques, and (5) communication control strategies or

network access schemes (protocols). In the next chapter, we shall explore these classifications in some detail. In this thesis, we are primarily interested in local networks with the most popular global databus topology and a network access scheme that is multi-accessing/broadcasting in nature; i.e., when any node has a message (packet) to send, it broadcasts the message onto the global databus so that everybody can receive an identical copy of the message. It is the responsibility of the destination node to recognize the message that is intended for it.

While many multi-access schemes have been developed or proposed, it was recognized that each of these schemes has its advantages and limitations. A trade-off exists between efficiency and control overhead under different traffic loads [Tob80]:

> "If a scheme performs nearly as well as perfect scheduling at low input rates, then it is plagued by a limited achievable channel capacity. Conversely, if a scheme is efficient when the system utilization is high, the overhead accompanying the access control mechanism becomes prohibitively large at low utilization."

In view of the above difficulty, this report looks into two new network protocols: the shortest-delay access method (SDAM) protocol, and the group broadcast recognizing access method (GBRAM) protocol. Both of these protocols reduce the efficiency-overhead trade-off mentioned above; namely, they provide very efficient data transmissions when the system

utilization is high, and yet at low utilization the overheads of the access control mechanisms remain relatively low. This is true even in situations where the number of nodes in the network is large and the propagation delay is long (i.e., the most unfavorable condition). Furthermore, by its very definition, the SDAM protocol defines a lower bound for queueing delays among protocols of its class -- the token-passing protocols. Both analytic and simulation models are employed to investigate the performance (e.g., the throughput-delay relationship, the throughput versus traffic load trade-off, the maximum achievable network capacity, etc.) of the SDAM and the GBRAM protocols.

1.4 Organization of the Thesis

In the next chapter, we will give an overview of the local area computer networks, including the components of a local network and the classifications of the many LACNs. In Chapter 3, we define the algorithm for the SDAM, and describe the network configuration on which SDAM will be applied. Two variants of the SDAM, the closed SDAM (C-SDAM) and the open-ended SDAM (OE-SDAM), are possible depending on the manner in which the network control (the virtual token) is passed on the network. These two variants will be analyzed via queuing models in Chapter 4. Chapter 5 evaluates the relative merits of C-SDAM and OE-SDAM, as well

as other performance issues in the algorithm, such as the network overheads and the effect of mixed-size packets. Simulation methods will be used to validate the analytic results, and to obtain experimental data for situations where analytic approach is difficult, if not impossible.

The GBRAM protocol is defined in Chapter 6 and analyzed in Chapter 7. Simulation models will also be used to study the performance of various GBRAM variants in Chapter 8.

In Chapter 9, we compare the algorithm of SDAM with two other similar conflict-free protocols for their relative merits. We also compare the performance of SDAM and GBRAM to that of the perfect scheduling and some other popular protocols such as the carrier sense multiple access with collision detection (CSMA/CD) [Met76], and the broadcast recognizing access method (BRAM) [Chl79a] protocol.

Finally, in Chapter 10, we present a summary and conclusion of this work, with some comments on the possible applications of these protocols. We will also point out a few directions for further research beyond the scope of this study.

CHAPTER 2

AN OVERVIEW OF LOCAL AREA COMPUTER NETWORKS

2.1 Components of A Local Area Computer Network

According to Clark et. al., a local area computer network (LACN) shares with the long haul networks three basic hardware elements and one basic software element [Cla78]. They are:

(1) a transmission medium, e.g., coaxial cable, fiber optics, twisted pair, etc.,

(2) a mechanism for control of transmission over the medium,

(3) an interface to the network for each of the connected nodes.

and

(4) a set of software protocols which control the

transmission of information from one node to another via the hardware elements of the network.

This set of software protocols functions at various levels (or control layers) of the network architecture. Typically, the higher level protocols reside in the host machines, and are designed to be machine and network independent. The lower level protocols, on the other hand, are strongly influenced by the network topology, transmission medium, and the control mechanism, and may be partly implemented into the network's interface units. The International Standard Organization (ISO) has defined a seven-layered network control structure for the long haul networks [Mar81]. These layers from bottom to top are:

1. physical control (basically hardware)
2. link control
3. network control
4. transport end-to-end control
5. session control
6. presentation control
7. application (process) control.

An illustration of the layered structure of network control is given in Figure 2-1. It is very likely that this layered architecture will also be adopted into a local network standard [IEE81], so as to make the interconnection of networks (local or long haul) possible. However, the currently existing LACNs may not conform strictly to this

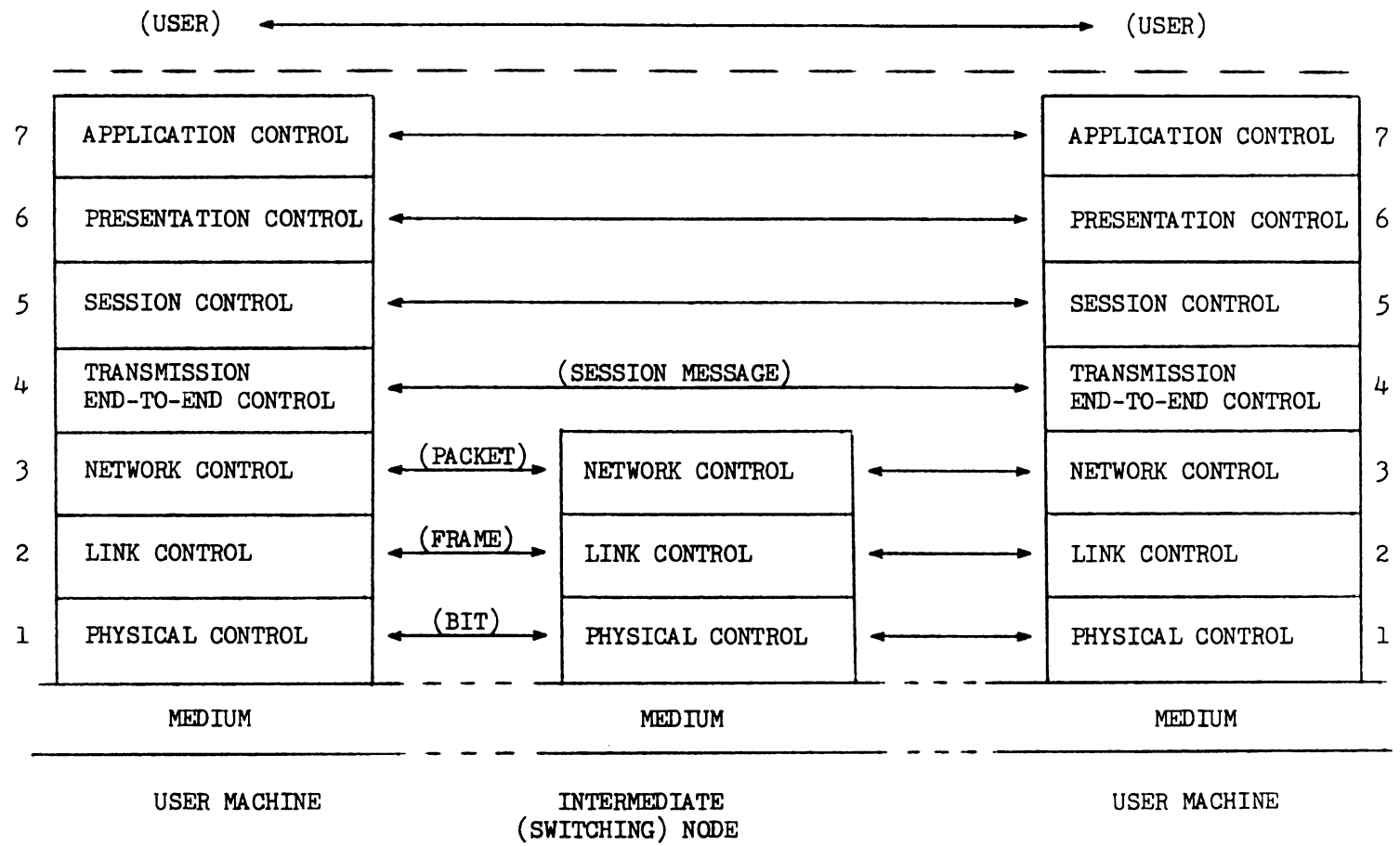


Figure 2-1. The layers of network control

architecture, although some form of layering does exist in each of these LACNs.

It is apparent that any good design of LACN must encompass different functions and services at all levels, such as fault detection and isolation, error recovery, flow control, and intra-process communication, etc.. But this report mainly focuses on the network hardware and communication technology dependent lower level protocols (i.e., the lowest two layers of the ISO model), upon which the higher level protocols will be built. Therefore, in this work the term "communication protocol" will be used to refer to the network access level (levels 1 and 2) protocols unless otherwise specified.

2.2 LACN Taxonomy

In order to better understand the LACN concept, to describe the current available technologies, and to compare these technologies and evaluate them for various applications, it is very important to categorize the various types of local networks currently available. This has generally been regarded as a difficult task due to the diversity of these local network designs. Several taxonomies of LACN have been cited in the literature, and are briefly presented in this section.

2.2.1 Clark's Taxonomy

Clark, Pogran, and Reed [Cla78] distinguish LACNs according to the following characteristics:

(1) topology, including the star, the ring, the bus, and the unfavorable arbitrary topologies.

(2) network control structure, including:

--Daisy chain, control token, and message slots

--register insertion

--contention control.

They observe that it is possible for any control structure to be used in conjunction with any topology. In addition, the LACN can also be classified according to:

(3) transmission medium, such as coaxial cable, twisted pair, CATV, fiber optics, radio broadcast, and light signals.

They point out that some of these transmission media have such distinctive characteristics that they can profoundly influence the control structure of the network.

2.2.2 Shoch's Taxonomy

In his thesis [Sho79], Shoch classifies five major designs of local networks as a result of a systematic effort to outline the dimensions of the design space.

(1) partially connected, store-and-forward networks

using either IMPs (interface message processors) or hosts for switching.

(2) simple star networks and strictly hierarchical systems.

(3) rings and loops, with one of the following control strategies:

--control passing or "token passing" techniques

--"empty slot" techniques

--"buffer insertion" techniques

--loops with centralized control or switching

--specialized loops for terminal systems or CPU/IO buses.

(4) radio-based approaches.

(5) multi-access bus structures.

Shoch noted that the qualitative evaluations of these designs suggest that one of the more attractive architectures for a local computer network is the shared bus with distributed control such as the Ethernet system.

2.2.3 Thurber and Freeman's Taxonomy

Thurber and Freeman have developed a tree which enumerates the various types of existing network architectures that are considered to be LACNs [Thu79a]. Three levels of classification are employed in enumerating the tree: the evolution context, the reason, and the

subnetwork communication technology.

--new system/subsystem concepts

--distributed processing

--packet switching(1)

--circuit switching(2)

--bus structure(3)

--I/O channel(4)

--existing system improvement

--communication bound

--packet switching(5)

--I/O or memory bound

--bus structure(6)

--I/O channel(7).

As a result of their classification, seven categories at the top of the tree are identified among the 51 LACN systems considered. However, they failed to mention many proposed LACN communication protocols, many of which offer interesting new LACN concepts.

2.2.4 Luczak's Taxonomy on Global Bus Computer Communication Techniques

Luczak has a very detailed classification for the large number of techniques that have been proposed for global bus computer communication on local computer networks [Luc78]. The following is an outline of his classification.

(1) By configuration -- bidirectional or dual-unidi-

rectional shared channels.

(2) By channel access

```
--selection

    --centralized control
        --daisy chaining
        --polling
        --independent request

    --decentralized control
        --decentralized daisy chaining
        --decentralized polling
        --decentralized independent request

--random access

    --access control
        --slotted
        --unslotted
            --pure ALOHA
            --CSMA
                --collision detection
                --deference/acquisition

    --collision resolution
        --non-adaptive retransmission delays
        --adaptive retransmission delays
        --node priority delays
        --reservation upon collision

    --message protocol
        --message establishment
            --separate phase
            --integral with  message transmission
              phase
        --acknowledgment
            --message-level ACK
            --dedicated ACK internal

--reservation

    --static
        --TDMA

    --dynamic
        --centralized control
            --connection based
            --message based
        --distributed control
```

--implicit reservation
--explicit reservation

Note that certain techniques may be classified into more than one classification in Luczak's taxonomy.

2.2.5 Cotton's Distinguishing Features for Local Network Technologies

The following distinguishing features are proposed by Cotton [Cot80] in comparing the various local area computer network technologies.

(1) topology -- e.g., fully connected, star, ring, distributed.

(2) medium -- e.g., digital baseband signalling or modulated; twisted pair, cable or radio.

(3) (bandwidth) sharing techniques -- e.g., dedicated (non-shared), or frequency division multiplexing, statistical multiplexing, contention.

(4) user service and protocol -- functions supported by higher level protocols, regardless of the internal transmission mechanisms.

Cotton also presented some comments of the pros and cons for each of the technologies when used with the most common network configurations.

2.2.6 Tobagi's Taxonomy for Multi-access Protocols in Packet Communication Systems

Tobagi [Tob80] groups various multi-access techniques into the following five categories:

(1) fixed assignment techniques, including frequency division multiple access (FDMA) [Rub79], time division multiple access (TDMA) [Lam77], and code division multiple access (CDMA), etc..

(2) random access techniques, including ALOHA, slotted ALOHA [Rob75], carrier sense multiple access (CSMA) [Kle75], busy-tone multiple access (BTMA) [Tob75], and spread spectrum multiple access (SSMA) [Kah78], etc..

(3) centrally controlled demand assignment, including circuit oriented systems, polling systems [Tob76], adaptive polling or probing [Hay78], split-channel reservation multiple access (SRMA) [Tob76], global scheduling multiple access (GSMA) [Mar78], etc..

(4) demand assignment with distributed control, including reservation ALOHA [Cro73], the first-in-first-out (FIFO) reservation scheme [Rob73], the round-robin (RR) reservation scheme [Bin75a]; mini-slotted alternating priorities (MSAP) [Kle77a], mini-slotted round-robin (MSRR), the assign-slot listen-before-transmit protocol [Han79]; distributed tree retransmission algorithm

[Cap79], and distributed control algorithms such as control token passing, etc..

(5) adaptive strategies and mixed modes such as: the UNR scheme [Kle77b], the CSMA/TDMA dynamic management of packet radio slots [Ric78], the reservation upon collision scheme (RUC) [Bor78], the mixed ALOHA carrier sense (MACS) [Sch79], and group random access (GRA) [Rub77], etc..

It should be noted that some of the schemes presented in this classification are used in satellite and radio-channel networks. However, with some modifications they can be equally applicable to the local network concepts.

2.2.7 Summary of LACN Taxonomy

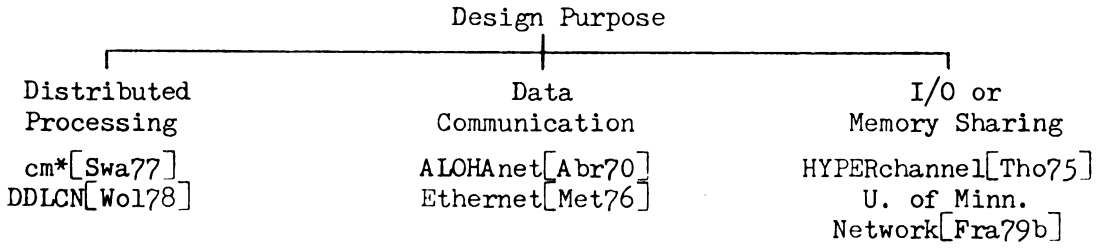
In summarizing the taxonomies presented in the preceeding sections, an outline of the LACN taxonomy is listed in Figure 2-2. No attempts will be made in this work to classify all of the existing or proposed local network designs into this taxonomy because of the numerous systems or protocols involved, and because this type of classification is subject to the network designer's point of view. Some examples of the classifications, however, are presented in this outline whenever possible.

In the following section, we shall focus our attention on the most popular bus topology, and examine some of the multi-access techniques on the bus networks.

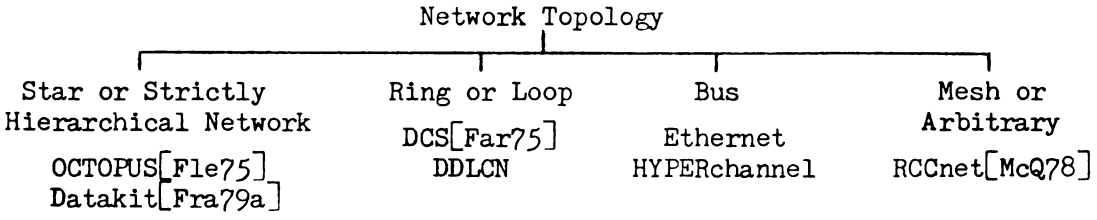
2.3 Multi-access Protocols on Bus Networks

2.3.1 Advantages of Bus Topology

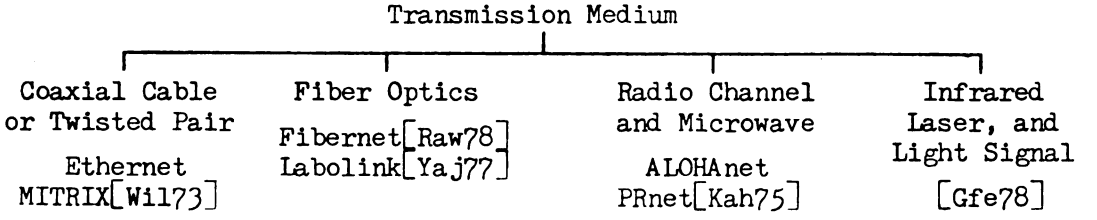
So far we have identified four types of local network topologies: the star (and the hierarchical systems), the ring or loop, the bus, and the mesh (or arbitrary) network (see Figure 2-3). In long haul networks, the mesh topology is frequently employed to arrange the communication links so as to optimize the use of costly transmission media, and to provide alternate data path between nodes to achieve network reliability. However, in local networks where transmission media are inexpensive, this topology becomes cumbersome due to the complexity of the routing mechanisms at each node. Additionally, since the propagation delay in local networks is short and the data transmission rate is typically very high, the control overheads induced by store-and-forward switching as compared to the actual data transmission time become intolerable.



(a) Classification by design purpose

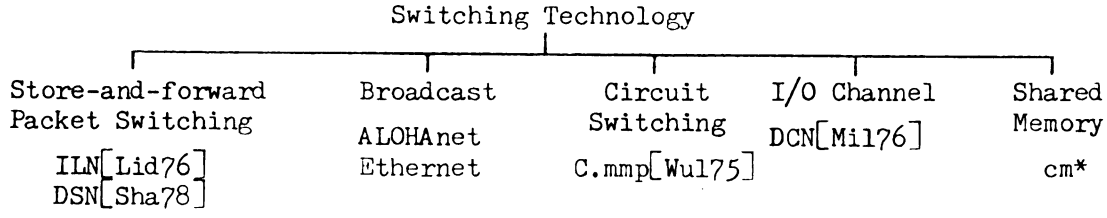


(b) Classification by network topology

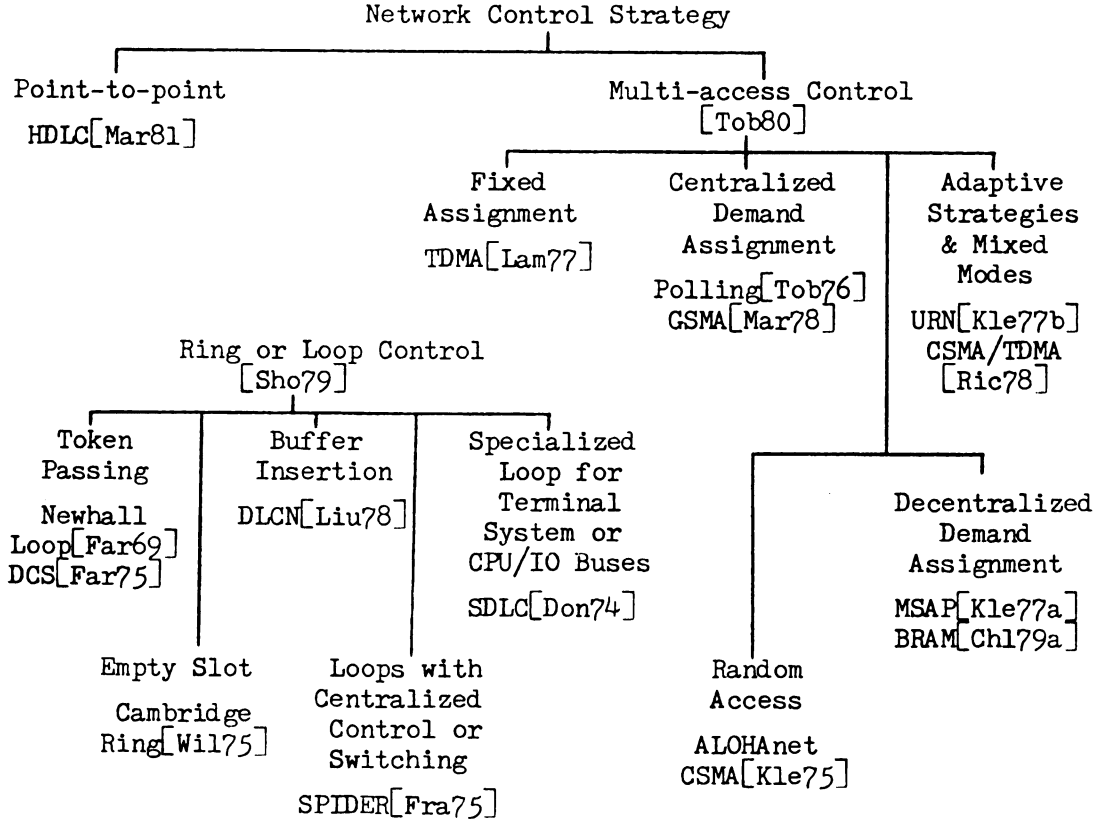


(c) Classification by transmission medium

Figure 2-2. Summary of LACN taxonomy

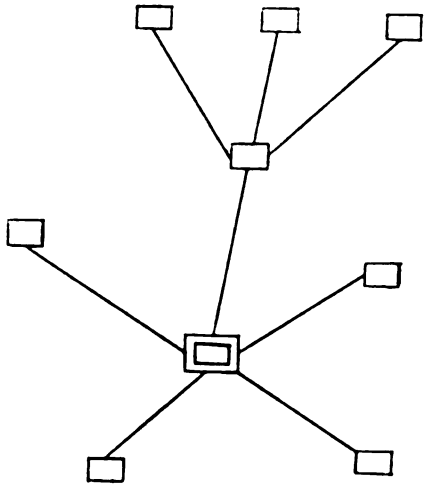


(d) Classification by switching technology

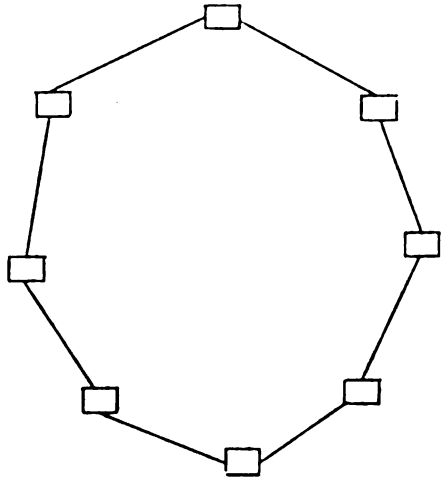


(e) Classification by network control strategy

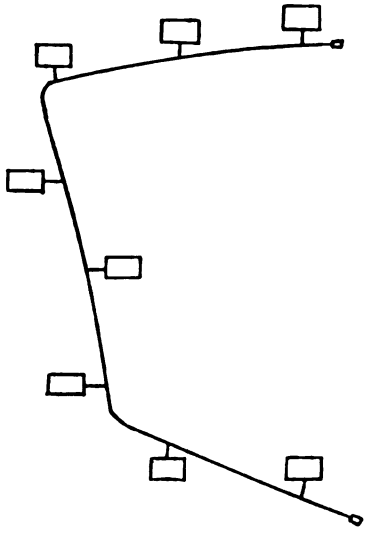
Figure 2-2(Cont'd).



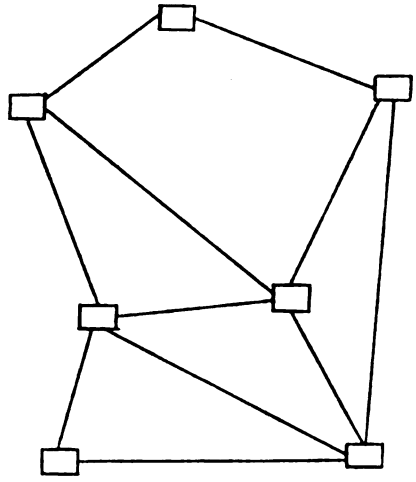
a. The star/tree



b. The ring or loop



c. The bus



d. The mesh

Figure 2-3. Local network topologies

The star network simplifies the routing by utilizing a central node to connect each of the nodes in the network, and by performing message switching only at this central node. However, this configuration could cause data congestion at the central node. Moreover, the reliability of the network depends on the proper functioning of this central node.

The ring or loop network attempts to eliminate the central node on the network, without sacrificing the simplicity of the other nodes. There are no routing decisions to make (except for a double loop; see [Wol78]); the message is relayed single-directionally from the originating node to each of the nodes on the ring (loop). However, any malfunctioning of any one of the nodes (in case of the double loop, two nodes) will bring down the entire network. It is also difficult to picture a ring or loop that contains a large number of nodes and still maintains high efficiency after all the message-relaying operations.

The bus network, using broadcast transmission techniques, is also free of routing decisions. Futhermore, there is no need for any node to relay or regenerate any passing-by messages; and no single point of failure exists on the network. Adding on new nodes to the bus is easy: one simply taps a transceiver to the passive cable, and connects this transceiver to the host interface unit. In summary, the bus topology has seen the most popular use due to the

following advantages [Chl79b]:

1) the ease of (physically) adding on or deleting nodes from the network;
2) the absence of a need for routing mechanisms;
3) the reduced number of failure points on the network; and
4) the absence of a need for nodes to regenerate messages.

2.3.2 Performance Criteria for Multi-access Protocols

Multi-access schemes can be evaluated according to various performance criteria [Tob80]. Some of these criteria are heavily application-oriented, such as the response-time constraints or the allowable error rates, etc.. Other criteria pertain to unquantifiable characteristics of the protocol, such as the ease of implementation and reconfiguration, or robustness of the network, etc.. But in general, the three most commonly used quantifiable performance criteria for the multi-access schemes are:

1) the message delay to throughput (channel utilization) relationship. That is, the average queuing delay the messages (packets) experience during various levels of channel throughput. The lower the delay, the better the network performance;

2) the network capacity, defined as the maximum achievable channel throughput under the access scheme;

3) the traffic load to the network throughput relationship. This is used mainly to evaluate the contention schemes, where some of the useful channel bandwidth is wasted due to data collision and retransmission.

These three criteria as a whole are sometimes referred to as the "network performance". The other quantifiable performance criteria include:

4) the distribution of the message delay, and the percentage of the delay that meets a certain response-time constraint;

5) the stability of the scheme, in terms of the throughput behavior under extremely heavy traffic load. In certain contention schemes such as CSMA, the channel throughput actually drops as the traffic load increases beyond network's capacity. This is due to the increased probability of collision under heavy traffic, resulting in decreased successful transmission rate. Such an access scheme is said to be "unstable";

6) sensitivity to the user population in terms of increased transmission delay. The polling scheme, for example, is extremely sensitive to the user

population;

7) sensitivity to the channel propagation delay.

Finally, there are those criteria that are non-quantifiable, but nevertheless are important to the network's performance. They include:

--robustness, defined as the insensitivity to errors resulting in mis-information or even network failure;

--simplicity of the access algorithm, and the ease of procuring necessary hardware elements to support the scheme;

--ability to handle messages of different types, lengths, priorities, and different traffic patterns;

--fairness to all user nodes (optional), and the guarantee of deadlock-free operations;

--other application-specified requirements.

In the next section, we will briefly evaluate some of the currently existing or proposed protocols based on the above performance criteria, following Tobagi's taxonomy [Tob80].

2.3.3 Review of the Multi-access Protocols

All the multi-access protocols evolve around one fundamental question, that is: how to efficiently share the common channel bandwidth among all users (nodes). The first class of such protocols -- the fixed assignment techniques,

takes a very straightforward approach. It allocates the channel to the users, independent of their needs, by partitioning the time/bandwidth into slots which are assigned to the users in a fixed predetermined fashion. Time division and frequency division multiple accesses (TDMA [Lam77] and FDMA [Rub80]) are two such examples. They provide the highest transmission efficiency for a small user population if the traffic pattern consists of uniform steady streams from all users. However, the efficiency suffers greatly if the user demands are dynamic and/or the data transmissions are bursty. Additionally, there is only a limited number of users that the FDAM is able to serve due to the limitations in available bandwidth.

The random access (contention) techniques form the other extreme class of the multi-access protocols. These protocols provide no fixed allocation of bandwidth at all. Instead, whenever a node has messages to send, it initiates the transmission attempt immediately. The first such technique is the ALOHA system developed in 1970 at the University of Hawaii, employing packet-switching on radio channels [Abr70] in a star configuration. Since there is no coordination between users wishing to transmit, a portion of one user's packet may overlap with another user's transmission, resulting in what we call a "collision", and both packets are assumed lost. When this collision occurs, the packets must be retransmitted again after a randomized

delay so as to avoid repeated collisions.

It has been observed that the ALOHA system provides very efficient channel utilization when there is a large population of bursty users [Kle75]. However, due to the waste of bandwidth in collisions and retransmissions as the traffic increases, the maximum channel throughput can only reach 18 percent. An improved scheme, called the slotted-ALOHA [Rob75], divides the channel time into slots of the size of one packet time. Any transmission attempt must start at the beginning of a slot, thus eliminating any partial collisions. The result is an improvement of the channel capacity to 37 percent. However, the synchronization of slot times among different users may be difficult to achieve.

Another improved random access scheme is the carrier sense multiple access (CSMA) protocols [Tob74], in which the nodes listen to the on-going traffic before attempting to start transmission, thereby eliminating the majority of collisions. A collision can now occur only if two nodes start their transmission at nearly the same time, before they can sense each other's packets. It is also possible to have a slotted version of CSMA, where a slot is defined as the maximum propagation delay on the channel--the time that a packet is vulnerable to collision after the start of its transmission.

Yet another improvement of CSMA protocol is possible when this scheme is applied to cable (bus) communications. That is, each node will listen to its own packet during the transmission, hence can detect any interferences or collisions much sooner (this was not possible in radio broadcast because the transmitting signals will overload the receiver). This variation of CSMA is referred to as the CSMA with collision detection (CSMA/CD) [Met76,Tob79].

Studies of these CSMA variants show that they are able to support a large number of users and still have very good performance in light to medium traffic loads. This is because in light traffic, collison is less likely to occur, hence the nodes can complete their transmission attempts with almost zero delay. However, as the traffic load increases, the probability of collision also increases, resulting in wasted bandwidth, and eventually leading to "unstability" as mentioned earlier. One exception is the Ethernet network access scheme, where a dynamic exponential backoff-retransmission scheme is used [Met76]. But this leads to a large variance in queueing delays in heavy traffic loads. Still another vulnerability of CSMA schemes is their sensitivity to the channel propagation delay. A study shows that if the propagation delay is longer than a certain threshold, the channel capacity of CSMA will drop to the level of the slotted-ALOHA, which is a mere 37 percent [Kle75,Net76].

The third class of the multi-access protocols pertains to the centralized demand assignment techniques. A most obvious example is the polling scheme [Tob76], where the central node polls each node consecutively for any transmissions. But generally, polling is efficient only if (1) the round-trip propagation delay is small, (2) the overhead in polling is small, and (3) the user population is not a large bursty one [Tob80].

A modified polling technique based on a tree searching algorithm has been proposed [Hay78]. This scheme, called probing, attempts to reduce the polling time by successively putting inquiries on the line, and dividing the user group into subsets according to the users' responses, eventually identifying the busy user(s).

An attractive alternative to polling is the use of some reservation schemes. For example, in split-channel reservation multiple access (SRMA) [Tob76], the channel is split into a request channel, which employs some form of random access scheme for users' reservation requests; and a data channel, which is granted to one of the busy user by the central node.

These access schemes are able to handle a large number of bursty users; but they all require a central node to control the network's operations. Therefore the network's reliability depends entirely on the proper functioning of this central node.

One reason why the fourth class, the decentralized demand assignment technique, is more desirable is because it leads to an improved network reliability and performance [Tob80]. Each node obtains the same information from the channel, executes an identical algorithm independently, and yet together they are able to achieve network coordination. Since there is no data collision, the network performance will remain stable at high traffic loads. Two of such techniques are the mini-slotted alternating priority (MSAP) [Kle77a] and the broadcast recognizing access method (BRAM) [Ch179a]. In both these schemes, the time is divided into mini-slots of the size of the maximum propagation delay. All the users are assigned time slots in a logical ring fashion. When one of the users wishes to send a packet, it does so in its mini-slot, provided the channel is not already busy. The succeeding nodes will then sense the presence of this packet, and stop the "rotation" of turns. As soon as the transmission is done, the rotation resumes. Since the transmission times from different nodes are separated by at least one mini-slot, collisions can be completely avoided by carrier sensing. This type of protocols is sometimes referred to as the "virtual-token passing" protocols.

The above schemes work very well with a small number of users. But the performance degrades, however, as the number of users increases.

Finally, in the fifth class, the adaptive strategies and mixed-mode access schemes attempt to choose an access mode which is itself adaptive to the varying needs of the users, so that optimality is achieved at all times. As an example, the URN shceme [Kle77b] estimates the number of busy nodes (m) among the total number of users (M) at a given time slot (defined as the size of a packet time), and grants the channel access to a subgroup of size (M/m). This leads to an optimal probability that exactly one node in this subgroup is busy, resulting in a successful transmission. The problem of this URN scheme is, of course, how to determine the number m in a distributed environment, and to select the subsequent subgroup in a negligible time.

A second example of the adaptive strategies is the parametric-BRAM [Chl79a]. Again the user population is partitioned into K groups, with the nodes in the same group sharing a group slot. To achieve optimality, this K will vary according to the traffic load so as to reduce both the scheduling period and the probability of intra-group collisions. Additionally, this partition must be such that the aggregated traffic load among the groups be approximately equal, which is very difficult to achieve in practice. In Chapter 6 we shall examine both the BRAM and the parametric BRAM protocols in more detail.

2.4 TWo Proposed Protocols for Local Computer Networks

2.4.1 The Background

As we have discussed earlier, the adaptive schemes such as the parametric-BRAM, although providing a theoretically optimal access control because of their adaptive nature, are very difficult to implement for the very same reason. The random access schemes are able to handle a large population of bursty users, yet they suffer greatly when traffic load is high. On the other hand, the distributed demand-assignment techniques handle traffic well in high loads, but introduce large scheduling overheads in light traffic loads. In addition, these schemes are sensitive to the number of users on the network. Both the random access and the demand assignment schemes are sensitive to the channel propagation delay.

2.4.2 The Approaches

In the remainder of this report, we shall look into a new type of access protocol, called the shortest-delay access method (SDAM), and a modification of the BRAM protocol, called the group-BRAM (GBRAM). Both SDAM and

GBRAM utilize the "virtual token" concept employed by MSAP and BRAM. The SDAM protocol minimizes the token passing time between two consecutive transmissions from any two nodes, hence significantly improves the performance of the virtual-token passing schemes. The GBRAM protocol, on the other hand, groups the nodes into node clusters according to their locations on the channel; then it utilizes a two-level (a group level and a node level) scheduling structure to reduce the token-passing overheads. More specifically, it is the objective of both protocols to:

1) have decentralized controls;
2) maintain conflict-free transmissions;
3) use simple algorithms and little control overheads;
4) perform closely to M/D/1 perfect scheduling in ideal cases (taken into account the bus propagation delay). In particular, its performance exceeds that of the CSMA/CD protocol in medium to high load;
5) tolerate large bus propagation delays;
6) provide adequate services to a large number of users (e.g., 1000 nodes); and
7) be fair to all user nodes.

In the following chapters, we shall define the algorithms of such protocols, and then evaluate the attainment of these goals in the subsequent chapters.

CHAPTER 3

DEFINITION OF THE SDAM PROTOCOL

3.1 Basic Concept of SDAM

The underlying concept of SDAM is to reduce the time-delay between two consecutive transmissions from two different nodes (i.e., the network control's changeover time). In most conflict-free protocols, this time-delay is typically equal to one end-to-end bus propagation delay, so that the collisions between these two consecutive transmissions can be avoided [Kle77a,Chl79a]. However, this changeover time introduces an excessive overhead when the user population is large (see Figure 3-1).

If we study the analogy between token-passing of a bus network and the disk-access scheme of a disk drive, we find that the waste of the changeover time can be minimized by using similar techniques. The network token may be seen as

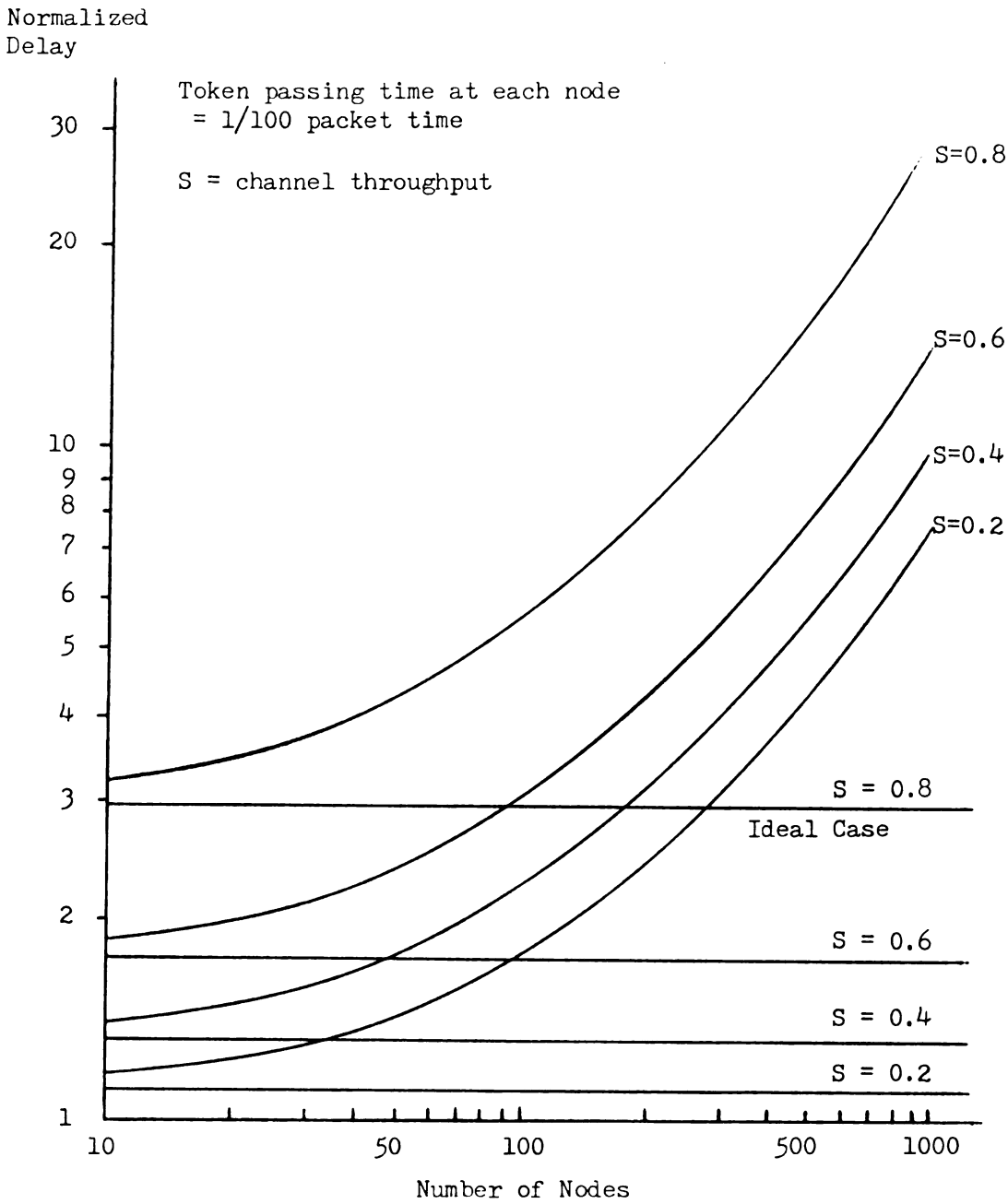


Figure 3-1. Delay vs. user pop. for token-passing schemes

the disk head, moving about the tracks (i.e., the nodes on the bus), and processing requests referencing those tracks (i.e., permitting nodes to transmit their packets) along the way. In a first-come-first-serve (FCFS) discipline, if the requests are arriving randomly at the tracks, then one would observe the disk head to move back and forth, not in an optimized fashion, while processing these requests. Denning et. al. observe that with the shortest-seek-time-first (SSTF) algorithm, where the disk head always chooses to process the request that has a track address nearest to the head's current position, the head's movement can be minimized in most cases (see Figure 3-2) [Den67].

Assuming:

number of tracks on the disk = 32;

original head position = track 18;

requests in order of arrival : tracks 3, 20, 4, 15, 10.

The first-come-first-serve (FCFS) algorithm
head movement: 18-→3-→20-→4-→15-→10. number of tracks scanned: 15 + 17 + 16+ 11 + 5 = 64 tracks.
The shortest-seek-time-first (SSTF) algorithm
head movement: 18-→20-→15-→10-→4-→3. number of tracks scanned: 2 + 5 + 5 + 6 + 1 = 19 tracks.

Figure 3-2. Example of the SSTF algorithm

However, the tracks on the inner and outer boundaries of the disk are less likely to be near the disk head (as opposed to the center tracks), therefore will sometimes suffer excessive delays. A straightforward way to remedy this situation is to have an one-directional SSTF, called SCAN [Cof73], where the disk head scans for the nearest track in one direction only, until the head either reaches the boundary of the disk or finds no more requests to process ahead in its scanning direction; then it turns around and scans the other direction.

Here, then, we can apply this SCAN algorithm to the network's token-passing; i.e., the token is passed from one end of the bus to the other end, and en route triggers the busy nodes to transmit their packets. When the token reaches the end of the cable, the token-passing direction is reversed, and the token is passed back to the other end. To achieve distributed control and still avoid conflict, SDAM uses a "token direction" code on each packet to indicate the direction of the current scan. The "virtual token", as perceived by a user node, is actually the absence of any more packets within a time interval following a passing packet on the bus, thus allowing the node to start its packet transmission.

3.2 Network Configuration

Before we describe the rules of such a shortest-delay access method (SDAM), let us define a general local network configuration for which the algorithm will apply.

1. The network topology is a bus topology with a finite number of branches. For simplicity, we will first assume a network with a single bus, then later generalize this topology into a branching bus topology.
2. The transmission medium is a coaxial cable with baseband signalling at, say, 10 Mbits per second.
3. There is a set of N nodes (computers, terminals, fast printers, etc.) connected to the bus (the common channel). These nodes are numbered sequentially from left to right or vice versa.
4. Each node is connected to the bus via a communication inteface unit, CIU (sometimes referred to as the bus interface unit, BIU, or network interface unit, NIU), which functions as a decoupler and buffer (see Figure 3-3). Henceforth we may consider the network as being composed of homogeneous nodes.
5. Each node (or CIU) has the carrier-sensing capability (i.e., the ability to sense the bus busy/idle status). The carrier sensing action

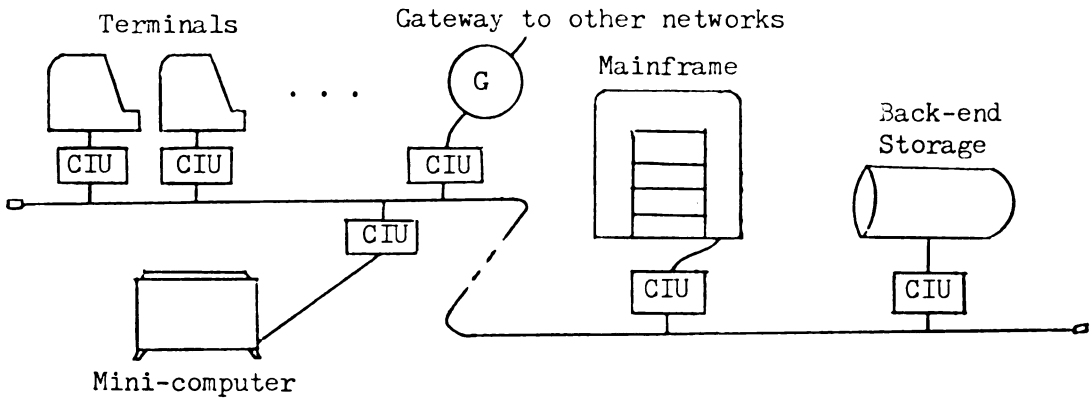


Figure 3-3. A single-bus local computer network

consumes a very small (but nonzero) amount of time, d.

6. Each node (or CIU) can identify the source and the destination addresses of the passing packets on the bus, as well as a "token direction" code on each of the packets.
7. Each node is able to transmit and receive, but not simultaneously. A turnaround time t is needed for the node to change from the receiving state to the transmitting state or vice versa, or to "digest" a long data packet the node just received. This is sometimes referred to as the inter-packet or inter-frame time.
8. There may be either one or two end-nodes attached to the cable. These nodes gain access to the common bus much the same way as the normal nodes do -- only they generate a control packet (or token

initialization packet, TIP) that contains a reversed token direction to the current token passing direction. It is also possible to add this feature to the user nodes located at the ends of the bus, so that they serve as both user- and end-nodes.

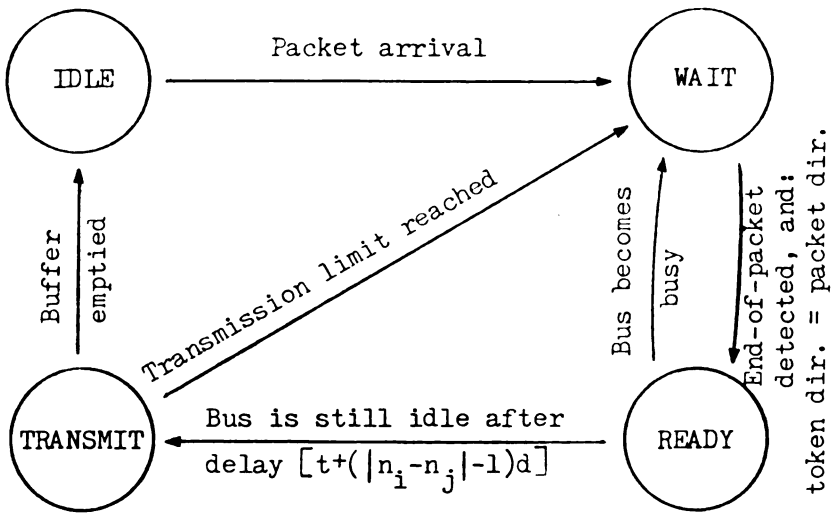
With these settings, we can then define the algorithm of the SDAM protocol in the following section.

3.3 SDAM on a Single Bus Topology

There are two variants of the SDAM protocol. The first variant uses both end-nodes to pass the token initialization packet (TIP) back and forth on the bus, and is called the closed SDAM, or C-SDAM. The second variant uses only one end-node to generate the TIPs regularly, and is referred to as the open-ended SDAM, or OE-SDAM.

Under both SDAM variants, each user node can be represented as having four states: IDLE, WAIT, READY, and TRANSMIT, as depicted in Figure 3-4.

1) IDLE. Originally, all nodes are in the IDLE state. When a packet is generated at a node, say node n_i, the node becomes "busy", and enters the WAIT state.
2) WAIT. Node n_i waits for any packet to go by on the bus. When an end-of-packet signal from node n_j is sensed on the channel, and if the token direction on that packet is the same as the packet's traveling



$\underline{t}$ = turnaround time for each node
$\underline{d}$ = carrier-sensing time of each node
n_i = the node wishing to transmit
n_j = the node that transmitted the last packet.

Figure 3-4. State diagram of the SDAM protocol

direction, then node n_i enters the READY state.

3) READY. Node n_i waits in the READY state for a turnaround time $\underline{t}$ plus the cumulated carrier-sensing delays $[(|n_i - n_j|)\underline{d}]$ along the token-passing route, and is now ready to send a message. But before the transmission actually takes place, node n_i keeps monitoring the channel status. If the channel becomes busy during n_i's READY state, the node goes back to the WAIT state. Otherwise it enters the TRANSMIT state.

4) TRANSMIT. In this state, node n_i keeps on transmitting until either its buffer is emptied (for

same t

packet

node c

diagr

end-n

initi

and t

so a

the (

(N+1

of t

acti

When

inte

turn

cour

anot

OE-9

3.7

exhaustive transmission) or some transmission limit is reached (for non-exhaustive transmission), and then it returns to either the IDLE or the WAIT state.

Note that all packets transmitted by node n_i carry the same token direction as that of the most recently passing-by packet. A schematic diagram of the SDAM protocol for a user node can also be found in Figure 3-5.

For the end-nodes of the C-SDAM protocol, the state diagram is the same as that of a user node, except that the end-node always have at least one packet (the token initialization packet, TIP) to send when the token arrives; and this packet always carries an opposite token direction so as to send the token backwards. A schematic diagram of the C-SDAM for the end-node is presented in Figure 3-6.

For the end-node of OE-SDAM, a counter of [2a + t + (N+1)d] is used, where $\underline{a}$ is the end-to-end propagation delay of the bus. After generating the first TIP, the end-node activates the counter and monitors the channel constantly. Whenever a packet from a node is detected, the countdown is interrupted until the packet has passed, and a packet turnaround time $\underline{t}$ is added back to the counter. When this counter expires, the end-node generates a new TIP and starts another round of token passing. A schematic diagram of the OE-SDAM protocol for this end-node can be found in Figure 3.7.

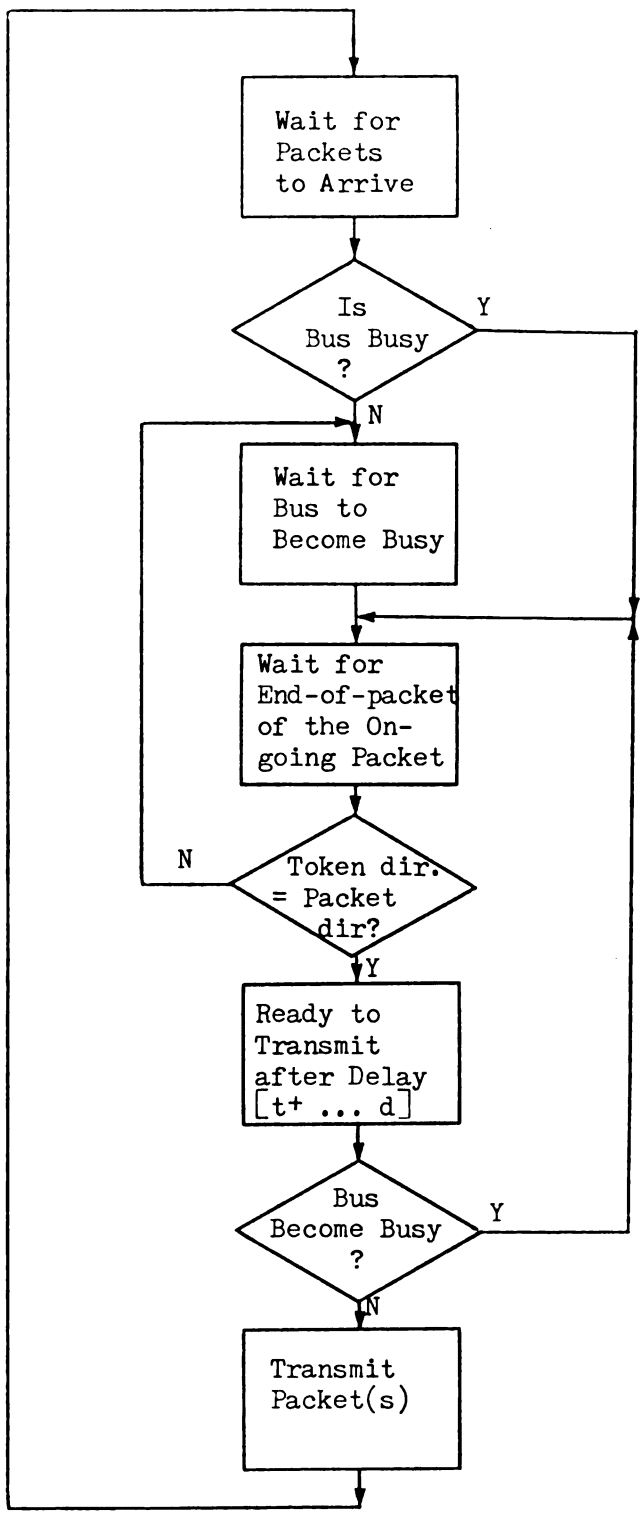


Figure 3-5. The flowchart of SDAM for a user node

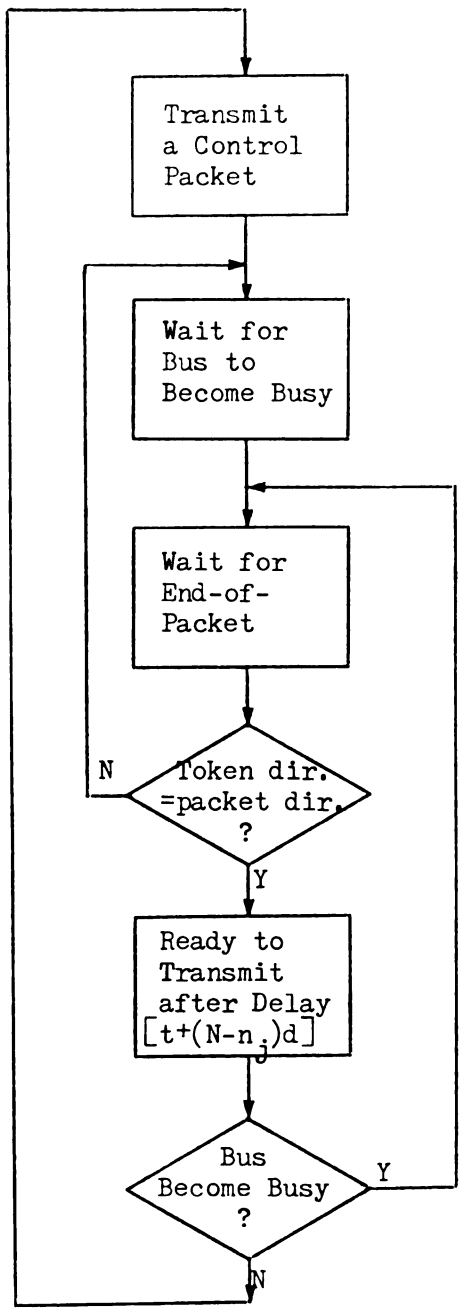


Figure 3-6. The flowchart for the end-nodes of C-SDAM

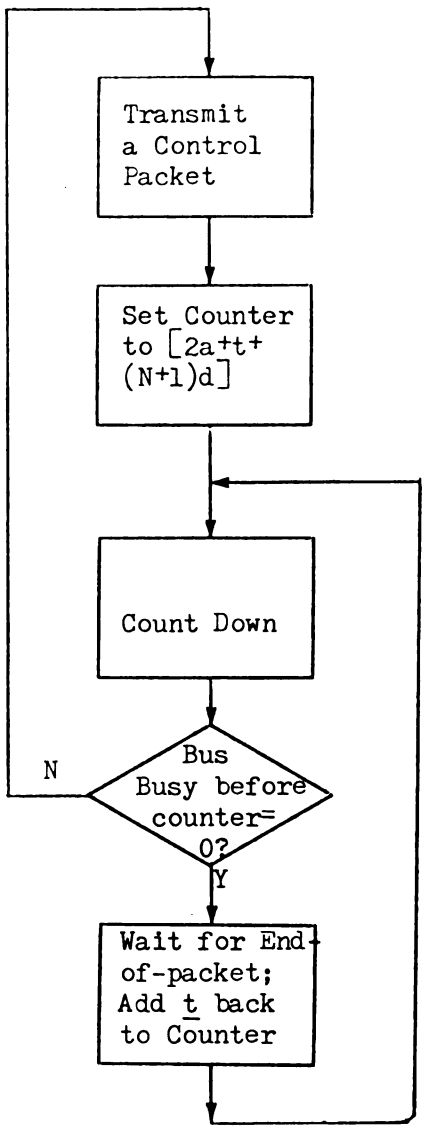
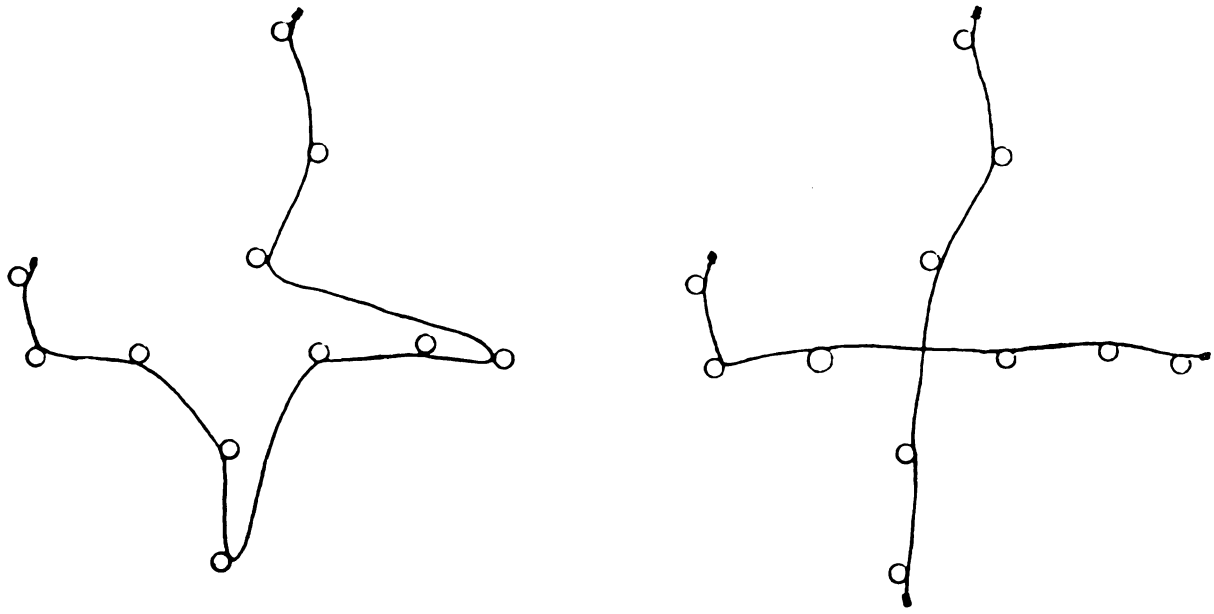


Figure 3-7. The flowchart for the end-node of OE-SDAM

3.4 SDAM on a Branching Bus Topology

Although it is alway possible to use a single bus to connect any set of nodes scattered in a local area, a branching bus network is more desirable for its shorter propagation delay as well as its flexibility in regards to future expansion and reconfiguration (see Figure 3-8). SDAM can easily be expanded to support this topology. Since any complex branching topology can be decomposed into simple three-branch structures as shown in Figure 3-9, it is sufficient to show that the algorithm of SDAM works on such a three-branch network. The reader can easily see that the same principle applies to networks with any finite number of branches.

For the C-SDAM protocol, because there are three end-nodes E1, E2, and E3 on the three branches, we need to change the token direction code from 'left' or 'right' to: 'E1-->E2', 'E2-->E3', or 'E3-->E1'. Each user node will be on precisely two of such paths (refer to Figure 3-10), and the node's access scheme remains unchanged except for the new token directions. After receiving the network token, an end-node will now generate a TIP carrying a token direction that points toward the next end-node in sequence (for example, end-node E2 will generate a new token direction 'E2-->E3', and E3 will generate 'E3-->E1', and so on).



a. A single-branch bus b. A multi-branch bus

Figure 3-8. Bus topologies for a set of local nodes

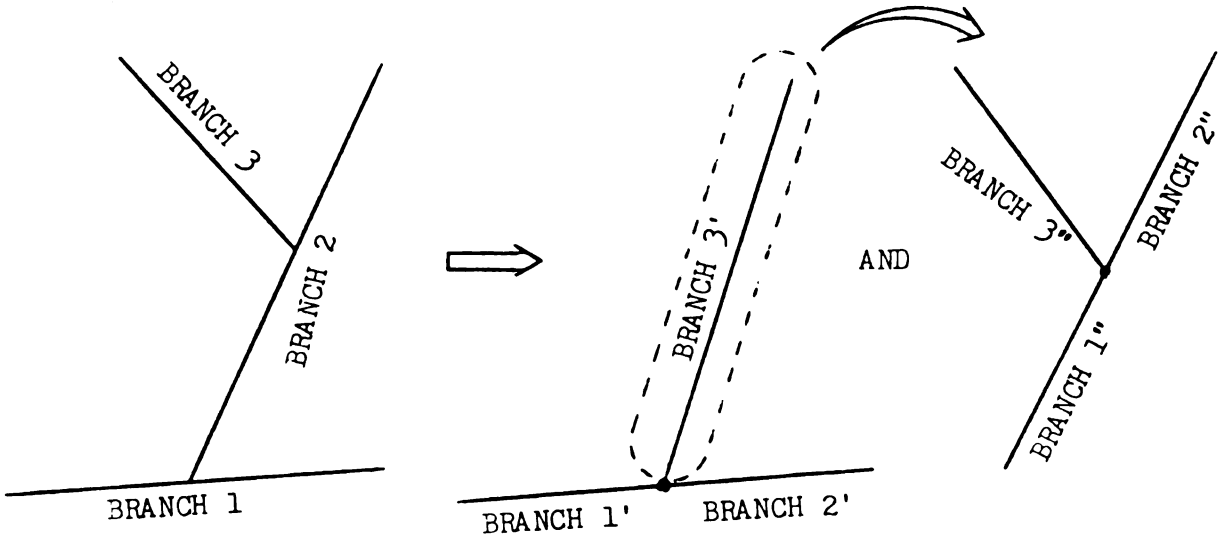
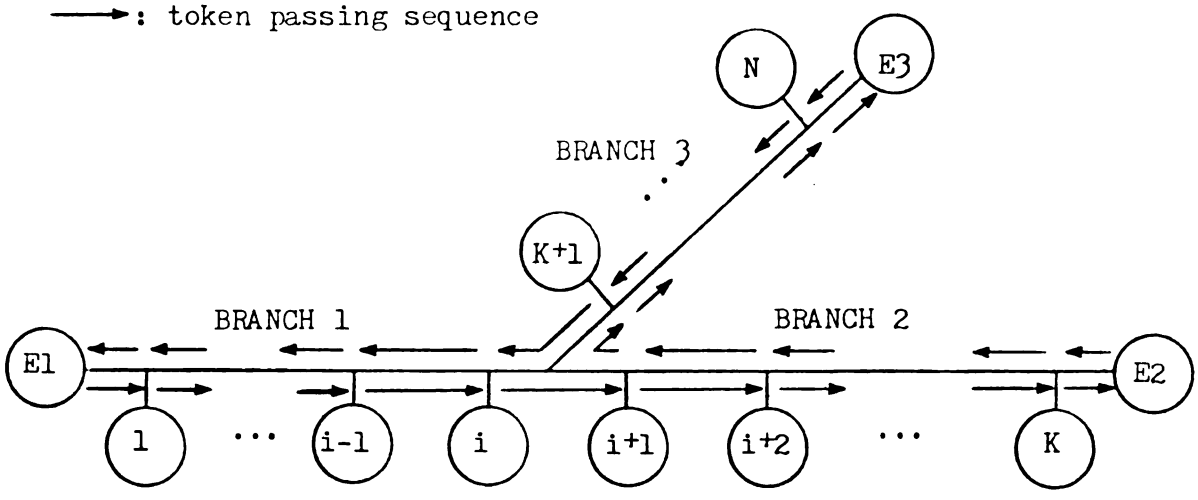


Figure 3-9. Decomposition of a multi-branch bus

For the OE-SDAM protocol, two token directions are possible: one is going 'left to right' on branches 1 and 2, and the other is going 'up right' on branch 3 (refer to Figure 3-11). Let us assume that the K nodes on branches 1 and 2 are sequentially numbered from 1 to K, and the remaining (N-K) nodes on branch 3 are sequentially numbered from K+1 to N. Then the K nodes on branches 1 and 2 are treated as if they are on a single bus network. For each node n_i on the third branch, however, a counter of $[(n_i-1)d + t + 2a_2]$, where a_2 is the propagation delay of branch 2, is used to determine when the token will arrive at node n_i. As soon as the node n_i detects the end of a TIP, it activates this counter, and monitors the channel status. Whenever a packet from branch 1 or 2 is heard, the countdown process is temporarily interrupted, and a turnaround time $\underline{t}$ is added back to the counter. If instead, a packet from a node on branch 3 is sensed, then n_i switches back to the single-bus access scheme as described earlier.

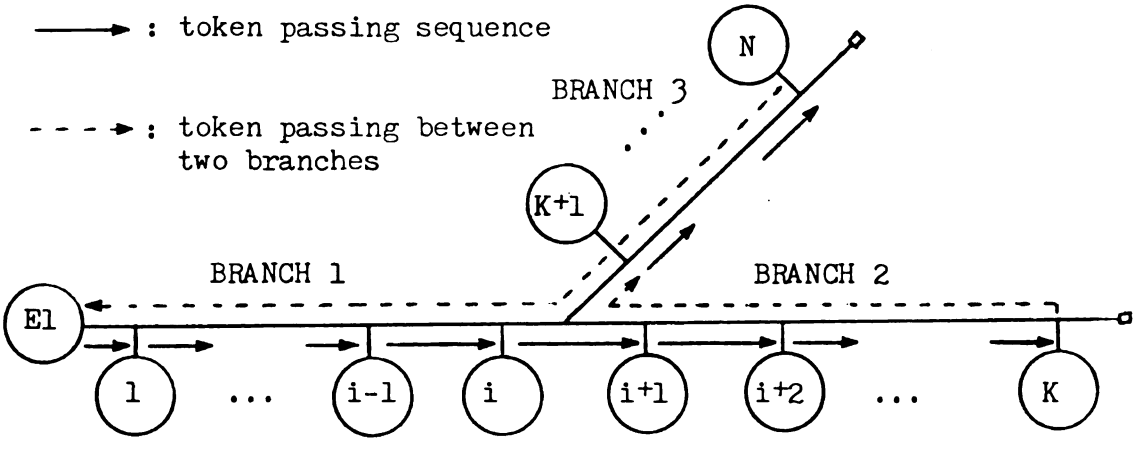
3.5 Illustration of the SDAM Protocol

To illustrate how the packet transmissions from different nodes are coordinated, we let Figure 3-12 depict a section of the bus, and Figure 3-13 shows its corresponding timing diagram. We assume that the network has been up and running for a while. Now, suppose node i received the token



The token is passed from E1 of Branch 1 to E2 of Branch 2. Then E2 changes the token direction toward E3 instead of to E1. Finally, E3 passes the token back to E1 and completes a token-passing cycle.

Figure 3-10. Token-passing of C-SDAM on a branching bus



The end-node E1 passes the token toward the end of Branch 2. The first busy node on Branch 3 waits for a time-out period, then starts the token passing on Branch 3. After another time-out period, the end-node E1 recovers the token, and completes a token-passing cycle.

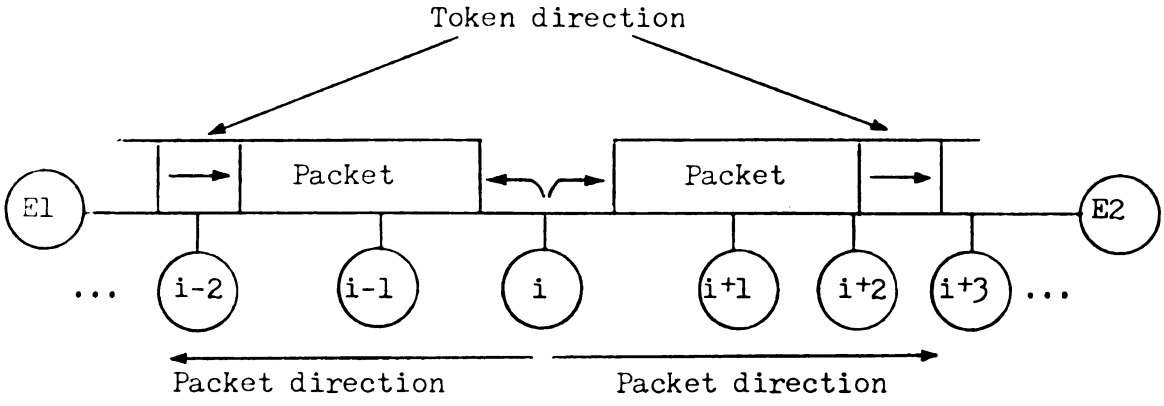
Figure 3-11. Token-passing of OE-SDAM on a branching bus

at time x_0, and had just finished its packet transmission at time x_1, with the token direction pointing toward the right (refer to Figure 3-12). After some delays later, the end-of-packet signals reach nodes i-1 and i separately. Node i-1, noticing that the packet's traveling direction (to the left) is different from the token direction (to the right), will then refrain from any transmission attempts. On the other hand, node i+1 will be able to go through the READY state to the TRANSMIT state and sends its packets onto the channel. Nodes i+2, i+3, ..., although later sensing the same information as node i+1 did, will stop at the READY state when they detect the carrier generated by node i+1, and will be routed back to the WAIT state.

When node i+1 finally completes its transmissions, the above procedure is repeated, except this time we assume that node i+2 is idle (refer to Figure 3-13). Then node i+3 is able to go through the READY state and starts its transmission after detecting the absence of a carrier from node i+2.

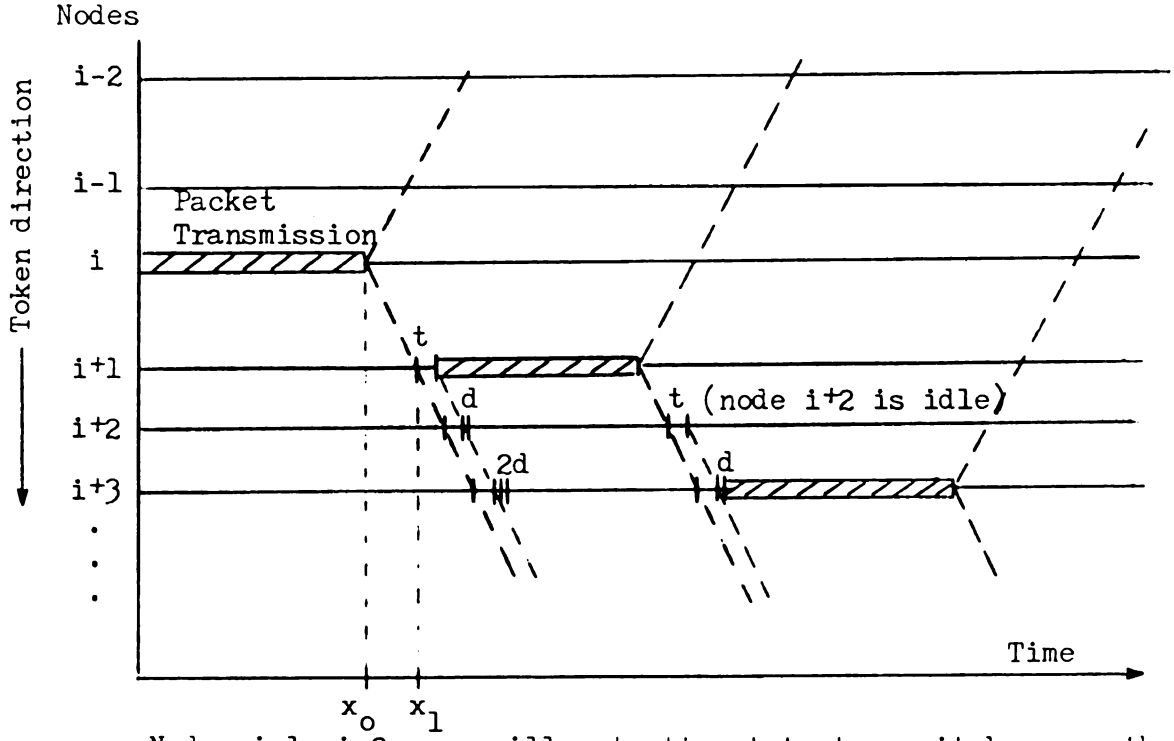
Continuing this process, the token will eventually reach the right end of the bus. With C-SDAM, the right end-node simply generates a TIP with a reversed token direction and sends the token backward (i.e., to the left). With OE-SDAM, the left end-node (the only end-node on the bus) will wait for the countdown to expire, then generate another TIP to start another round of token passing.

Token direction



Node i generates a packet with the token direction pointing toward the right. The packet direction is pointing toward either end of the bus from node i.

Figure 3-12. Packet traveling on a single bus



Nodes i-1, i-2,..., will not attempt to transmit because the packet direction (as seen by them) and the token direction are different. Nodes i+1, i+2,..., on the other hand, use carrier-sensing to avoid collisions.

Figure 3-13. Timing diagram of Figure 3-12

For the branching bus topology, again we will make use of the decomposed three-branch network as shown in Figure 3-14. Let's suppose that node i has just finished a packet transmission at time x_0, with the token direction pointing toward branch 2 from branch 1 (i.e., 'E1-->E2', refer to Figure 3-14). After some propagation delays later, the end-of-packet signals reach each of the following nodes: i-1, i+1, and K+1. Node i-1 and K+1, noticing that the packet's traveling directions (toward E1 and E3, respectively) is different from the token direction (toward E2), will not attempt to transmit. On the other hand, node i+1 will be able to go through the READY state to the TRANSMIT state and sends its packets onto the channel. Nodes i+2, i+3, ..., K will also defer their transmission attempts for the same reason as stated earlier in the single-bus example.

Continuing the token passing, the virtual token will eventually reach the end of branch 2. For the C-SDAM protocol, the end node E2 simply generates a TIP with a new token direction 'E2-->E3', and sends the token toward E3 of branch 3. The OE-SDAM protocol requires that the first busy node n_i ($K<n_i \leq N$) on the third branch wait for its counter to expire; then it claims the token, transmits the packet, and passes the token to the remaining nodes on branch 3.

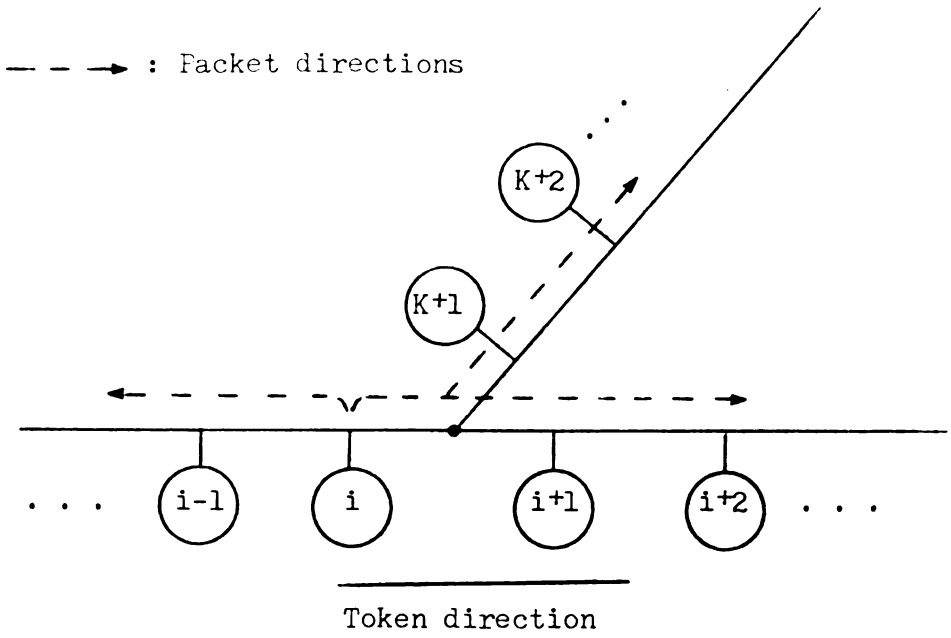


Figure 3-14. Packet traveling on a branching bus

3.6 Message Acknowledgment in SDAM

The algorithms of the SDAM protocol does not support low-level message acknowledgments (i.e., an ACK reply generated in the physical layer for every correctly received packet). Instead, a acknowledgment packet may be generated from a higher-lever control. This ACK packet will then be treated as a normal packet, and goes through the normal channel-access procedures. This decision is justified by the low error rate of a cable system (approximately one

error per 20,000 packets) observed by Shoch [Sho79], and the fact that such low-level acknowledgments could bring a severe degradation to the network's performance [Tob78].

3.7 Addition and Deletion of Nodes on the Network

In a SDAM configuration, all nodes on the network must be numbered sequentially from one end of the cable to the other end (see section 3.2). One question immediately comes to mind is: whether it is possible to add or delete nodes conveniently under such a constraint. In this section, two viable method will be described. But first we need to assume that the entire length of the cable be marked (and numbered) at a regular interval, and that the nodes (transceivers) be attached to the network only at one of such marks (such markings are also found in Ethernet for similar reasons [DEC80]).

METHOD 1. Because the SDAM protocol is relatively insensitive to the user population (as will be shown later), we may assume that at each of these marks is connected a (dummy or active) node, whose address is just the associated mark number. When a node wishes to sign off from the network, it simply stops transmitting, and becomes a "dummy" node. If a node wishes to join the network, it finds an unused mark, and takes the mark's number as the node address. Then it starts to act as an "active" node and

waits for the network token to arrive.

This method is obviously very simple and straightforward. But it may cause the average packet delay to increase due to the extra carrier-sensing times wasted on the dummy nodes, particularly when the cable is lengthy and the network nodes are sparse.

METHOD 2. In this method, only those nodes who are currently active are assigned sequence numbers (node addresses). When a node is signing off from the network, it broadcasts its intention, and any subsequent (higher-numbered) nodes will then subtract their addresses by 1, hence deleting this node from the network's token passing sequence.

Adding new nodes in this approach requires a bit more work. First, we need to assume that there is a set of "administrative" nodes, who periodically (or dynamically) generate a special sign-on inquiry. The active nodes, upon receiving such an inquiry, will remain inactive for this round of token passing. A node wishing to sign on, on the other hand, will have the chance to broadcast its presence, along with the cable mark number it is attached. The user nodes with higher numbered marks will then add 1 to their network sequence numbers, hence "making room" for the newcomer. A similar method has been proposed by the IEEE local network standard group [IEE81]. But unlike that proposed scheme, there will be no collisions in SDAM during

this sign-on period, because each mark can only be attached by at most one node, and because these new nodes also obey the token-accessing rules of the SDAM protocol.

CHAPTER 4

ANALYSIS OF THE SDAM PROTOCOL

Since in Chapter 3 we have made an analogy bewteen the SDAM protocol and the disk-access algorithm SCAN, it is only natural to apply the same analysis of SCAN [Cof73] to the SDAM protocol. A little study shows, however, that the analysis of SCAN is not entirely applicable to SDAM. In what follows, we shall briefly discuss the SCAN model analyzed by Coffman et. al., and point out the places where it is applicable and where it is not. Two alternative models will also be presented in our analysis of the SDAM protocols.

4.1 Coffman's Model

Assuming that the number of tracks on the disk is large, Coffman replaces the set of discrete track addresses by the unit interval [0,1] (i.e., the set of real numbers between 0 and 1). The speed at which the head can move across this interval is a. The input of requests is assumed to be Poisson with a mean rate of λ, and the "tracks" referenced are assumed uniformly distributed across [0,1]. Formally, the probability that a given arrival falls in the interval $(x,x+\Delta x)$ is given by Δx. But informally, we may regard the interval Δx as corresponding to a track. Hence the probability of more than one arrival in a small interval Δx is of order $(\Delta x)^2$, which we should be able to ignore if $\Delta x \rightarrow 0$.

Next, it is assumed that the time required to serve any request is a constant T, and that the direction of the scan is reversed only when the head reaches the boundary at 0 or 1 (hence conforms to the C-SDAM requirement).

If we let $\bar{y}(x)$ denote the mean time taken by the head to move a distance x from position 0 (and process requests along the way), as depicted in Figure 4-1, then we have

$$\bar{y}(x+\Delta x) = \bar{y}(x)+a\Delta x+\lambda T\Delta x[\bar{y}(x)+\bar{y}(1)-\bar{y}(1-x)] \qquad (4.1)$$

which we can then solve as [Cof73]:

$$\bar{y}(x) = \frac{1 - \lambda T(1-x)}{1 - \lambda T} ax \qquad 0 \leq x \leq 1 \qquad (4.2)$$

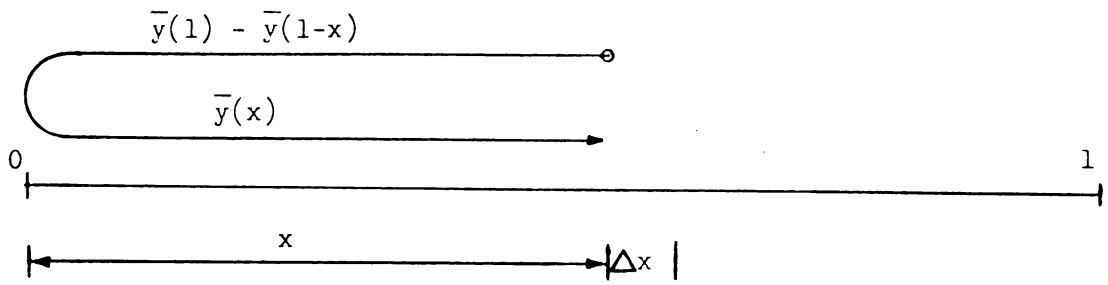


Figure 4-1. Head movement of SCAN

Making use of the fact that the requests arrive uniformly within [0,1], we can calculate the mean waiting time for a request arriving at x as:

$$w(x) = \frac{a}{2} \frac{1+(1-2x)^2}{1-\lambda T} \qquad (4.3)$$

Equation (4.3) indicates an uneven waiting time distribution across the tracks (see Figure 4-2), with the worst case occurring at the boundaries 0 and 1 of the interval:

$$w(1) = w(0) = \frac{a}{1-\lambda T} \qquad (4.4)$$

If we modify the rule of SCAN such that the requests are served only in one fixed direction only, then we have a model equivalent to OE-SDAM. Following a complete "service" scan, the head is returned to the starting position 0 at rate a before it starts another service scan. This scheme is referred to as the CSCAN algorithm, and the mean waiting

time is derived as [Cof73]:

$$w(x) = \frac{a}{1-\lambda T}, \qquad 0 \leq x \leq 1, \tag{4.5}$$

which is independent of the track address x (see Figure 4-2). This mean waiting time corresponds to the worst-case waiting time of SCAN in (4.4).

Note that the above results are based on a continuous approximation (i.e., the interval [0,1]) of the discrete track addresses. Clearly, this approximation worsens if the number of tracks (in the case of SDAM, the number of nodes) is not too large.

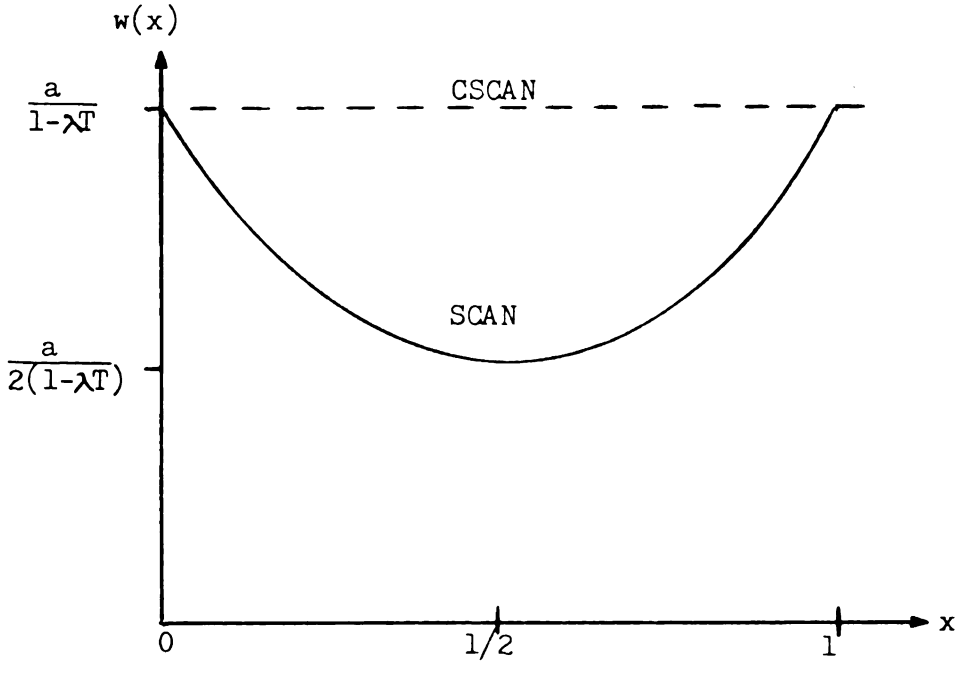


Figure 4-2. Queuing delay of SCAN and CSCAN

It is also understood that the head's movement time is much larger than the request processing time (i.e., $\underline{a}>T$) in a disk access scheme. For if this is not the case, then comparing Equation (4.5) with the well-known M/D/1 queuing system with perfect scheduling and without server walking time [Kle76]:

$$\text{(optimal) mean waiting time} = \frac{\lambda T^2}{2(1-\lambda T)}, \qquad (4.6)$$

we find that for $\underline{a}<(\lambda T^2)/2$, SCAN out-performs the optimal scheduling, which is a contradiction!

In summary, Coffman's analysis is based on the assumptions that the number of tracks is large and that $\underline{a}>T$, both of which may not be satisfied in a local network environment. A later study shows, however, that the uneven distribution of the waiting times across the tracks (refer to Figure 4-2) remains to be a valid observation.

4.2 Eisenberg's Model

As an alternative approach, since a user node under the SDAM algorithm is allowed to transmit its packets only when the network token arrives, we can visualize this token as being a single server, attending the multiple packet-queues forming at the network nodes. Moreover, since propagation delay exists on the bus, we must take into consideration the changeover time during which the server (the network token)

walks from one queue to another. Such a queuing system has been analyzed by Eisenberg in one of his papers [Eis72], and is summarized below.

In Eisenberg's work, the queuing system consists of N queues attended by a single server. The queues have independent Poisson arrivals and general service-time distributions. The server attends the queues in a repeating sequence (cycle) of I stages; each stage of the service cycle is spent working on a single queue until that queue is empty. This type of service discipline is sometimes referred to as the "alternating priority". The service sequence is defined by an ordered set of I integers $(n_1, n_2, \ldots, n_I)$, where n_i is the queue that is served during stage i. As an example, a possible sequence of service may be:

(stage) i:	1	2	3	4	5	6	7	8
(queue) n_i:	1	2	3	4	4	3	2	1

where there are N=4 queues that are served in a cycle of I=8 stages (note that this is exactly the service sequence of C-SDAM in a 4-node network).

When a queue is finally emptied, or when the server arrives at a queue that is already empty, the server immediately leaves and switches to the next queue in sequence. A switch from one queue to another always requires a changeover time whose distribution has finite

mean but is otherwise arbitrary. The mean waiting time w_i for jobs in queue n_i and serviced at stage i can then be expressed as [Eis72]:

$$w_i = [E(V_i^2)/2v_i] + [\lambda_{n_i} E(S_{n_i})/2(1-\rho_{n_i})] \qquad (4.7)$$

where:

v_i is the mean intervisit time, defined as the time at the begining of stage i since queue n_i was last visited;

$E(V_i^2)$ is the second moment of the i intervisit time;

λn_i is the arrival rate at queue n_i;

ρn_i is the traffic loading at queue n_i (=λ_{n_i} times average service time at n_i); and

$E(S_{n_i})$ is the second moment of the n_i service time.

If we let the number of states I be 2*N, and let the queue n_i serviced at stage i be (see Figure 4-3):

$$n_i = \begin{cases} i & \text{for } 1<i\leq N \\ I-i+1 & \text{for } N<i\leq I, \end{cases}$$

then Eisenberg's model can immediately be applied to the C-SDAM protocol. The average waiting time q_{n_i} at each node n_i can be dervied as:

$$q_{n_i} = [w_i(v_i+s_{n_i}) + w_{I-i+1}(v_{I-i+1}+s_{n_i})] / [v_i+v_{I-i+1}+2s_{n_i}] \qquad (4.8)$$

where s_{n_i} is the mean service time at node n_i, and v_i and w_i

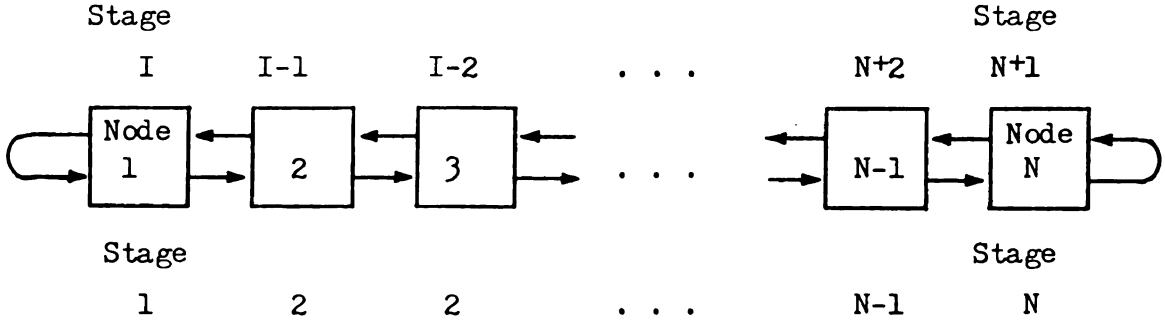


Figure 4-3. The walking server of C-SDAM

are as defined previously.

On the other hand, if we let I=N and $n_i = i$ at each stage i, then we have the OE-SDAM case. The average waiting time at each node i is just w_i given in Equation (4.7).

Although Eisenberg's model is directly applicable to both the C-SDAM and the OE-SDAM protocols, it was noted that when the number of queues in the model is large (e.g., N=50 queues), the calculation of $E(V_i^2)$ becomes virtually impossible [Eis72]. So, unless some good approximation schemes can be deviced to estimate the values of $E(V_i^2)$, Equation (4.7) can not be used to calculate the queuing delay of a general local network.

4.3 Konheim and Meister's Model

In the open-ended SDAM protocol, since token-passing is done single-directionally, it is possible to use a less complicated model for the purpose of our analysis; namely, we can visualize the network under OE-SDAM as being a system with polling, where the polling time in this case is simply the changeover time between two consecutive nodes. The same method has been adopted by Kleinrock and Sholl [Kle77a] in their analysis of MSAP (mini-slotted alternating priorities) and by Chlamtac et. al. [Chl79a] in their analysis of BRAM (broadcast recognizing access method). Both of these analyses are based on the analytic results given by Konheim and Meister [Kon74].

In Konheim and Meister's work, the system consists of a set of N buffered terminals. The common channel is seen as the server, and the data units to be transmitted at each node play the role of customers. The terminals are polled sequentially in order of T_1 , T_2 , ..., T_N (see Figure 4-4). The actual way by which polling is implemented (centralized or distributed) is irrelevant. Upon being polled, the terminal retains the use of the channel, removing data from its buffer for transmission at the rate of one data unit per unit of time. The arrival of data at the terminal continues during this transmission phase. When the buffer is emptied, the terminal transmits an

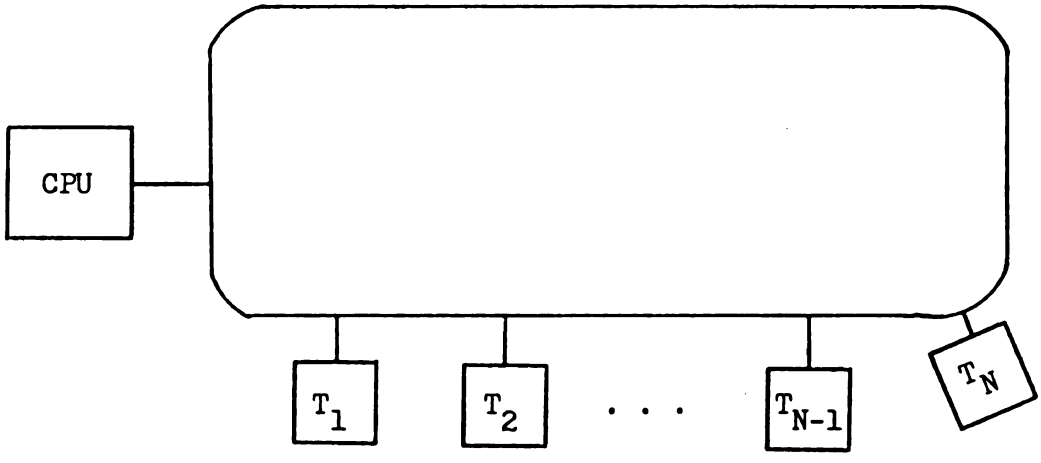


Figure 4-4. A polling system

end-of-message (EOM) mark and loses control of the channel. Thereafter the channel is not available to any terminal for transmission for an interval of variable length. This period of time is referred to as a reply interval. After the reply interval, the system continues with a poll of the next terminal in sequence. The stationary expected queuing delay E(D) is then derived as [Kon74]:

$$E(D) = \frac{\delta^2}{2r} + \frac{1}{2}\frac{N\sigma_1^2}{1-N\mu_1} + \frac{1}{2}(1-\mu_1) + \frac{Nr(1-\mu_1)}{2(1-N\mu_1)} \qquad (4.9)$$

where:

r = mean time for the reply interval,

δ^2 = variance of the reply interval,

μ_1 = mean arrival rate at one of the terminals,

σ_1^2 = variance of the arrivals at one of the terminals ($=\mu_1$ for Poisson arrivals).

4.4 Analysis of OE-SDAM

In order to apply Konheim and Meister's results to the OE-SDAM protocol, it is necessary to make the following assumptions about our local network's environment:

1) There are N nodes (excluding the end-node) connected to a single bus. Each node is associated to a queue with a FCFS discipline.
2) The queues have independent but identical Poisson arrivals with a mean arrival rate μ_1.
3) All packets are of fixed size, and each packet requires 1 time unit to transmit.
4) The bus propagation delay from the end-node to the last user node is $\underline{a}$ time units; and the N nodes are uniformly located on the bus.
5) The token initialization packet (TIP) takes $\underline{c}$ time units ($\underline{c}<1$) to transmit.

Under assumption (4), the changeover time (i.e., the reply interval) between two consecutive nodes is just $(a/N)+d$. Letting this be the new time unit $\underline{m}$ (mini-slot), we have:

$\mu = \mu_1/m$ = the packet arrival rate in mini-slots;

$\sigma = \mu_1/m^2 = \mu/m$ is the variance of arrivals.

If we define

$S = N\mu$ to be the channel throughput in equilibrium (i.e., $N\mu<1$),

Then (4.9) becomes:

$$E(D) = \frac{\delta^2}{2r} + \frac{N\mu/m}{2(1-N\mu)} + \frac{1}{2}(1-\mu)(1+\frac{Nr}{1-N\mu})$$

$$= \frac{\delta^2}{2r} + \frac{S/m}{2(1-S)} + \frac{1}{2}(1-\frac{S}{N})(1+\frac{Nr}{1-S}) \qquad (4.10)$$

in mini-slots.

Now for a single bus topology and a sufficiently large N (e.g., N=50), r and δ^2 can be approximated as follows. Excluding packet transmission times and their associated turnaround times, it takes the token [a+(N-1)d] time to reach the last node of the bus from the starting end-node; then it takes the token (a+d) time to travel back to the end-node. Therefore, the average token passing time is

$$r = \frac{1}{N}\{(c+t)+[a+(N-1)d]+(a+d)\}/(\frac{a}{N}+d)$$

$$= \frac{1}{N}(2a+c+t+Nd)/m \qquad (4.11)$$

in terms of mini-slots $\underline{m}$, where (c+t) is the network overhead associated with the initial TIP. The variance, δ^2, can then be determined as

$$\delta^2 = \frac{1}{N}\{(N-1)[1-r]^2+1\ [(a+d+c+t+\frac{a}{N})/m - r]^2\} \qquad (4.12)$$

where the second squared term pertains to the token passing delay between user-node N and user-node 1.

Finally, to normalize (4.10) into units of the packet transmission time, we multiply (4.10) by $\underline{m}$, so that:

$$E(D) = \frac{\delta^2 m}{2r} + \frac{S}{2(1-S)} + \frac{m}{2}(1-\frac{S}{N})(1+\frac{Nr}{1-S}) \qquad (4.13)$$

in units of packet transmission time, where m=(a/N)+d, and r and δ^2 are as defined in (4.11) and (4.12).

For the generalized branching bus as shown in Figure 4-5, we let

$$a = a_1 + a_2 + a_3$$

where a_i is the bus propagation delay of branch i, i=1,2,3. Then equations (4.11) and (4.13) still hold true, but the variance of the token passing time in (4.12) must be modified to:

$$\delta^2 = \frac{1}{N}\{(N-1)[1-r]^2 + 1\cdot[(a_2+d+\frac{a}{N})/m - r]^2$$

$$+1\cdot[(a_3+a_1+d+c+t+\frac{a}{N})/m - r]^2\}, \qquad (4.14)$$

where the second squared term is the token passing delay between the last node of branch 2 and the first node of branch 3, and the last squared term is again the token passing delay between user-nodes N and 1.

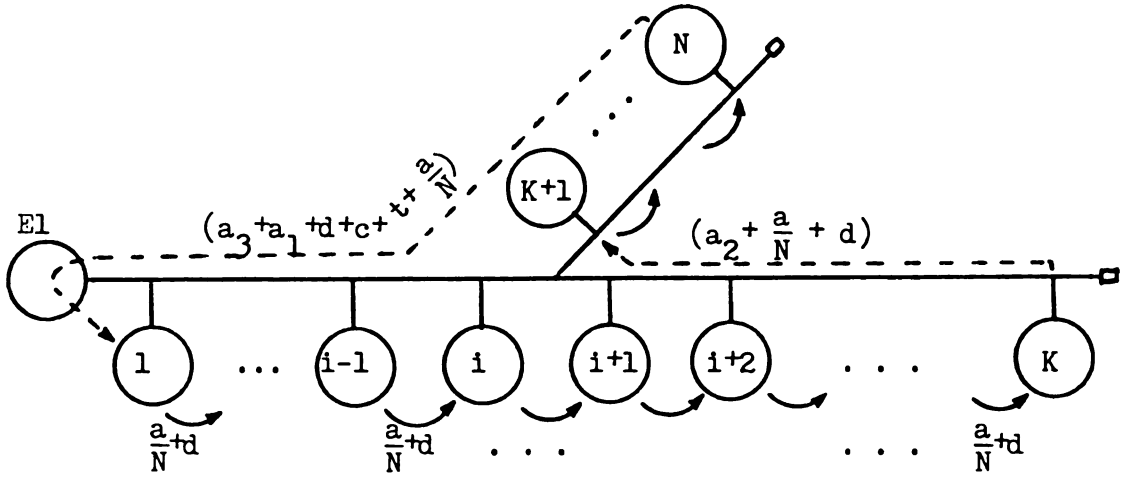


Figure 4-5. Token passing of OE-SDAM on a branching bus

4.5 Analysis of C-SDAM

The analytic model of Konheim and Meister's polling systems can not be directly applied to the C-SDAM protocol, becasue the polling sequence of C-SDAM is a "back-and-forth" polling rather than a "wrap-around" polling as in OE-SDAM. However, it was observed that, for a small bus propagation delay ($\underline{a}<0.1$), the average queuing delay of the C-SDAM protocol is very close to that of the OE-SDAM protocol. Therefore, we can use Konheim and Meister's formula (Equation (4.13)) to approximate the system's queuing delay under C-SDAM. The average polling time in this case is:

$$r = \frac{1}{N}\{(c+t)+[a+(N-1)d]+d\}/(\frac{a}{N+1}+d)$$

$$= \frac{1}{N}(a+c+t+Nd)/m \qquad (4.15)$$

in terms of mini-slots m, where m=a/(N+1)+d for C-SDAM. The variance δ^2 can also be determined as

$$\delta^2 = \frac{1}{N}\{(N-1)[1-r]^2+1\cdot[(d+c+t+\frac{a}{N+1})/m-r]^2\}. \qquad (4.16)$$

Having approximated the queuing delay of the entire system, the individual queuing delay for each node of the network can then be approximated by fitting Coffman's observation (Equation (4.3)) into Konheim and Meister's result. That is, we let

$$\int_0^1 \frac{aK}{2}\frac{1+(1-2x)^2}{1-\lambda T}\,dx = E(D) \qquad (4.17)$$

for some constant K. Then the estimated queuing delay of node i can be expressed as:

$$D_i = \frac{aK}{2} \frac{1+(1-2x)^2}{1-\lambda T} \qquad (4.18)$$

where $x=(i+1)/(N+2)$. An example of this approximation can be found in Figure 5-2. It should be noted, however, that this is just a very crude approximation. For more accurate values of such delays, simulation methods must be used.

CHAPTER 5

PERFORMANCE OF THE SDAM PROTOCOL

Having defined and analyzed both variants of the SDAM protocol, in this chapter we shall further investigate the performance of the protocol under various operating conditions. The emphasis is on the attainment of the objectives listed in section 2.5.2 of Chapter 2. There will also be comments regarding to some of the performance criteria as described in section 2.4.2, where appropriate.

For simplicity and without loss of generality, we will still assume a single bus network for our performance analysis in this chapter. Two GPSS simulation models have be developed to simulate the C-SDAM and the OE-SDAM protocols. Besides varifying the analytic results produced from the previous chapters, these simulation models will enable us to study cases where analytic approach is difficult (e.g., the non-exhaustive transmission discipline, or an unbalanced traffic load, etc.).

5.1 The Throughput-Delay Performance of SDAM

The following set of parameters represents a typical local network configuration, and is used for comparing the delay performances of C-SDAM and OE-SDAM.

N = 50 nodes;

Packet size = 1000 bits (fixed); packet time = 1;

$\underline{a}$ = 0.01, 0.1, 0.5, 1.0 for propagation delay;

$\underline{c}$ = 0.03 (30 bits) for TIP;

$\underline{t}$ = 0.02 (20 bit-time) for turnaround time;

$\underline{d}$ = 0.002 (2 bit-time) for carrier-sensing time.

When $\underline{a}$ is small ($\underline{a}$=0.01), the difference in the token passing time between these two protocols is small. Therefore, their performances are very close to each other, and to the M/D/1 perfect scheduling (see Figure 5-1). As $\underline{a}$ increases to 0.1 (i.e., 10 μs for a bus with 10 Mbits/sec bandwidth), the performance of C-SDAM begins to (slightly) exceed that of OE-SDAM , while both schemes still perform well. When $\underline{a}$=0.5, the average delay of OE-SDAM is 12% larger than that of C-SDAM. This difference expands to 22% as $\underline{a}$ increases to 1.0 (i.e., 100μs). However, even under such a long propagation delay, both SDAM variants still turn out an acceptable performance. This is due to the fact that the changeover time of the network control remains to be a very small fraction of the bus propagation delay, therefore the impact of this increased propagation delay is much less

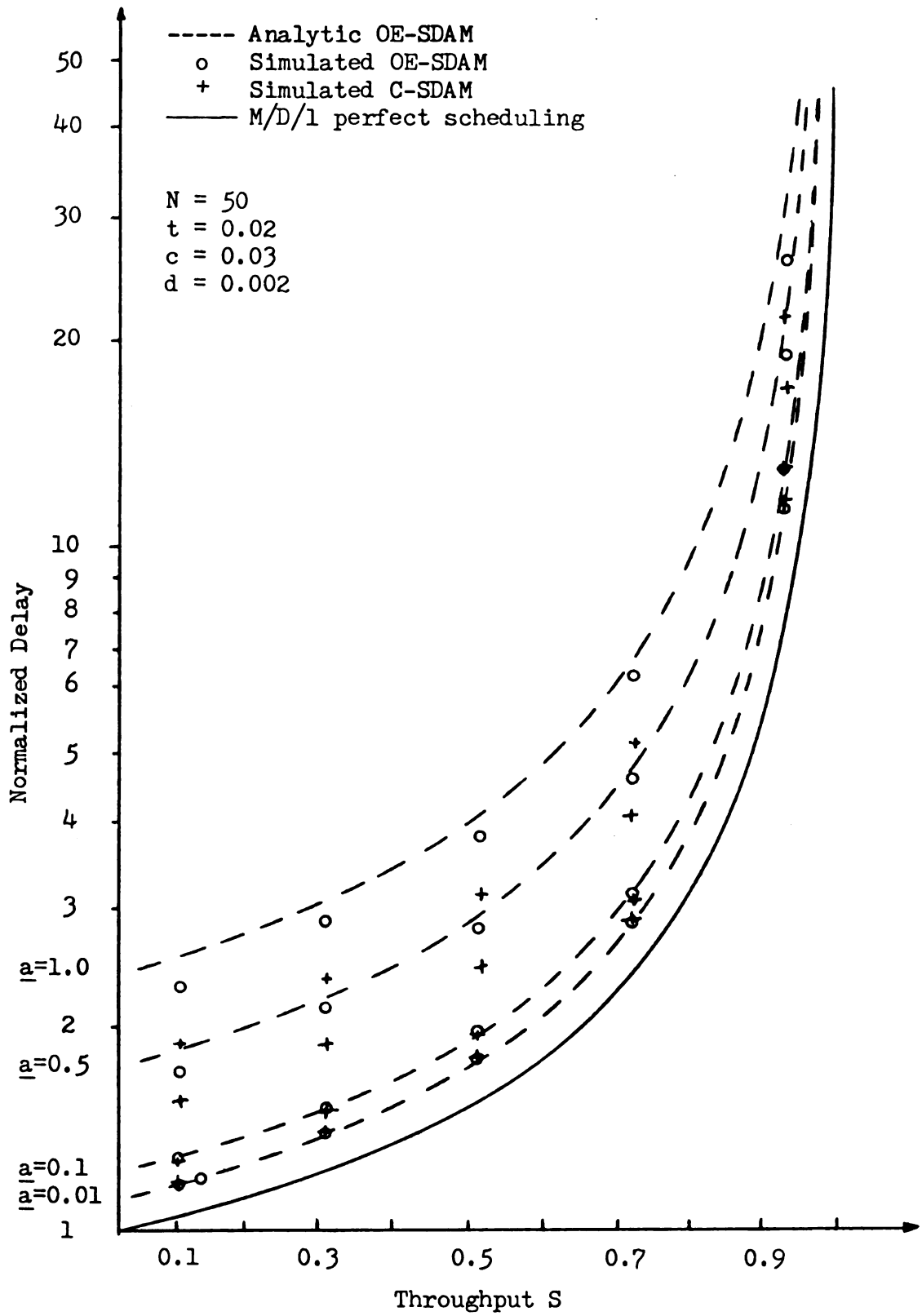


Figure 5-1. Delay performance of C-SDAM and OE-SDAM

severe as compared to other token-passing schemes.

The delay performance of the C-SDAM can be viewed as the "delay lower-bound" of the token-passing schemes in the following sense. The network control's changeover time is minimized to just the propagation delay between two adjacent busy nodes, plus one node's turnaround time (a necessity) and the "token handling" time of the idle nodes in between these two busy nodes. This token handling time is again being minimized to barely the time needed to carrier-sense the bus in order to ensure collison-free transmissions. All these overheads (propagation delay, turnaround time, and the carrier-sensing time) represent the minimum requirements for a token passing scheme. Therefore, by definition, no other token passing schemes can out-perform C-SDAM.

The C-SDAM protocol, however, has one performance drawback. That is, it tends to discriminate against the nodes located near either end of the bus (see Figure 5-2). This is because these nodes experience two uneven token intervist times. Consequently, the majority of the packets will arrive during the longer intervisit time (hence suffer longer delays) while only a small portion of the packets arrive during the shorter intervisit time, thus resulting in a longer averaged delay. The same phenomena has been observed by Coffman et. al. in their analysis of the disk access schemes (see section 4.1, Chapter 4). Therefore, C-SDAM may not be suited for implementation unless such a

discrimination can be justified for some practical reasons. For the remainder of this study, we shall concentrate on the performance of the OE-SDAM protocol for its virtue of fairness.

5.2 The Effects of the Protocol Overheads

In this study, three operating overheads of the SDAM protocols are considered: the turnaround time, the TIP time, and the carrier-sensing time. The effects of these overheads on network's performance are analyzed in this section.

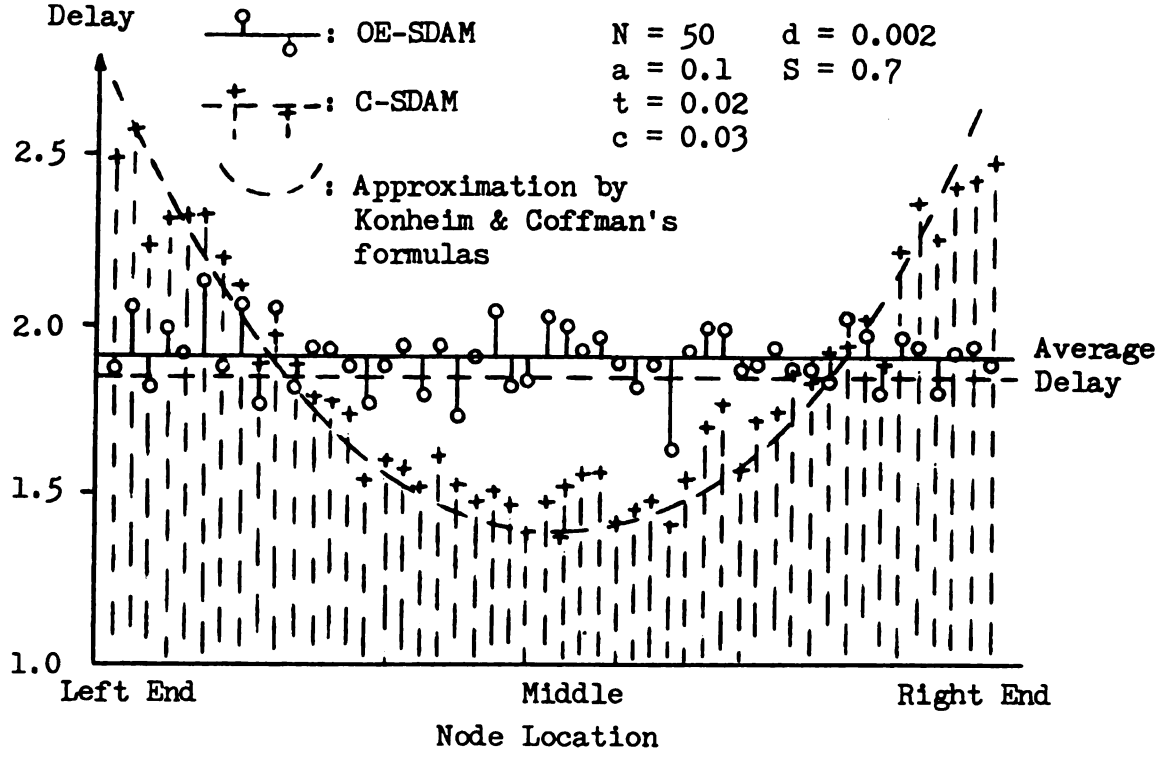


Figure 5-2. Packet delay of C-SDAM vs. node locations

5.2.

tran
bein
The
Equat
resul
bit-t
value
perfo
devas
the
netwo
maxim
traff
nonze
by (
netwo

5.2.2

cycle
bit-pa
Partic

5.2.1 The Turnaround Time and the Network Capacity

Since a turnaround time $\underline{t}$ is associated with each transmission of the packet, we can envision the packet as being enlarged by a ratio of t (i.e., new packet time=1+t). The throughput-delay curve can then be approximated by Equation (5.5), with S substituted by S'=S(1+t). The results are plotted in Figure 5-3 for t=0.0 to 0.1 (0 to 100 bit-times). This figure shows that, for light loads, the values of $\underline{t}$ do not significantly affect the network's performance. However, for high loads, larger $\underline{t}$ values have devastating effects on the average packet delay as well as the network's maximum achievable throughput (i.e., the network capacity). For an ideal case where t-->0, the maximum throughput of SDAM approaches 100% as the network's traffic load increases beyond 100%. But generally, for a nonzero $\underline{t}$, the maximum network throughput is bounded above by (1-t). So, if the turnaround time is t=0.1, then the network can only achieve an maximum of 90% throughput.

5.2.2 The Token Initialization Packet (TIP) Time

In SDAM protocol, a TIP is required to initialize each cycle of token-passing. This packet can be a special bit-pattern that all nodes recognize as being generated by a particular end-node; or it can be a shortened data packet

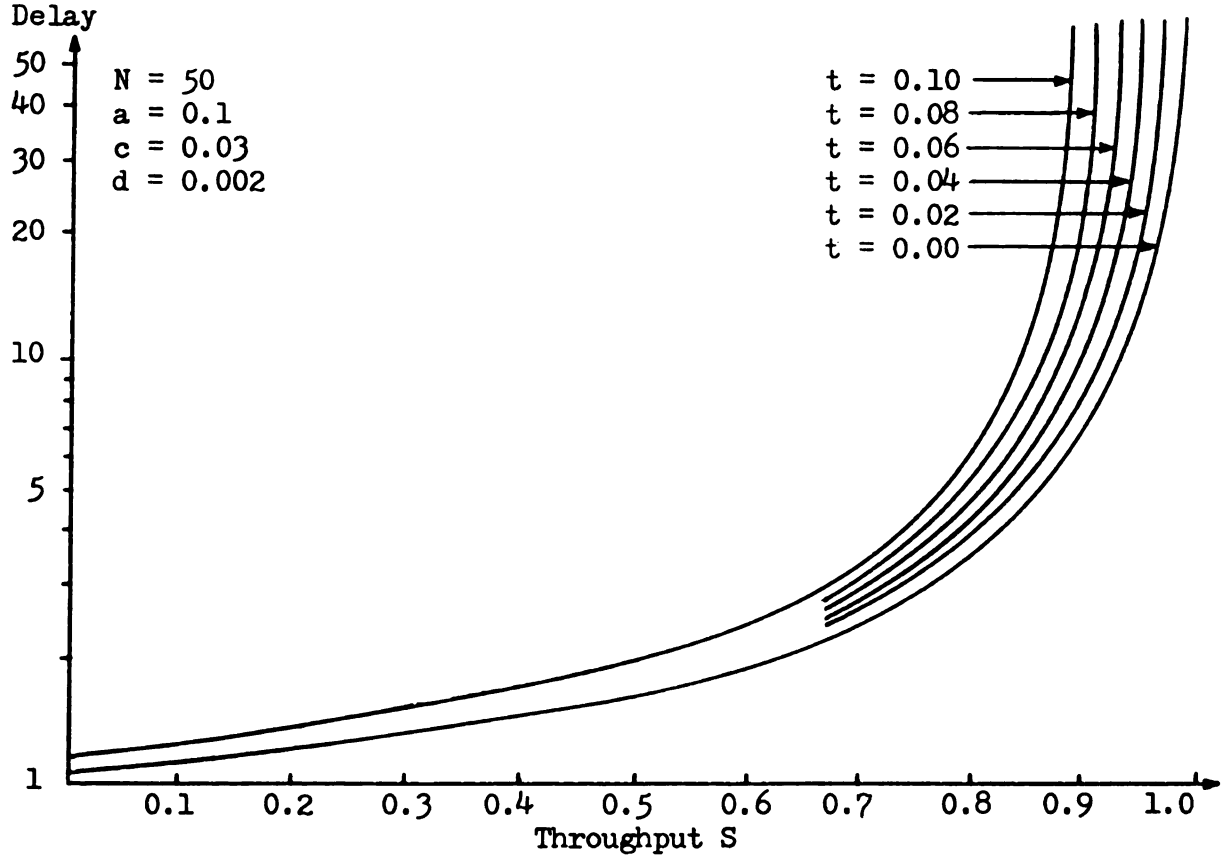


Figure 5-3. Effect of the turnaround time

that contains nothing but a source address and a token direction code. In any case, this packet will occupy a fraction of the channel time, and therefore must be considered as a network overhead. Analysis shows, however, that when the number of nodes on the network is large (e.g., n>20), the size of the TIP has little effect on the network's performance. The effect of varying TIP sizes under an extremely light load (S-->0) is plotted in Figure 5-4. In higher loads, this effect becomes negligible.

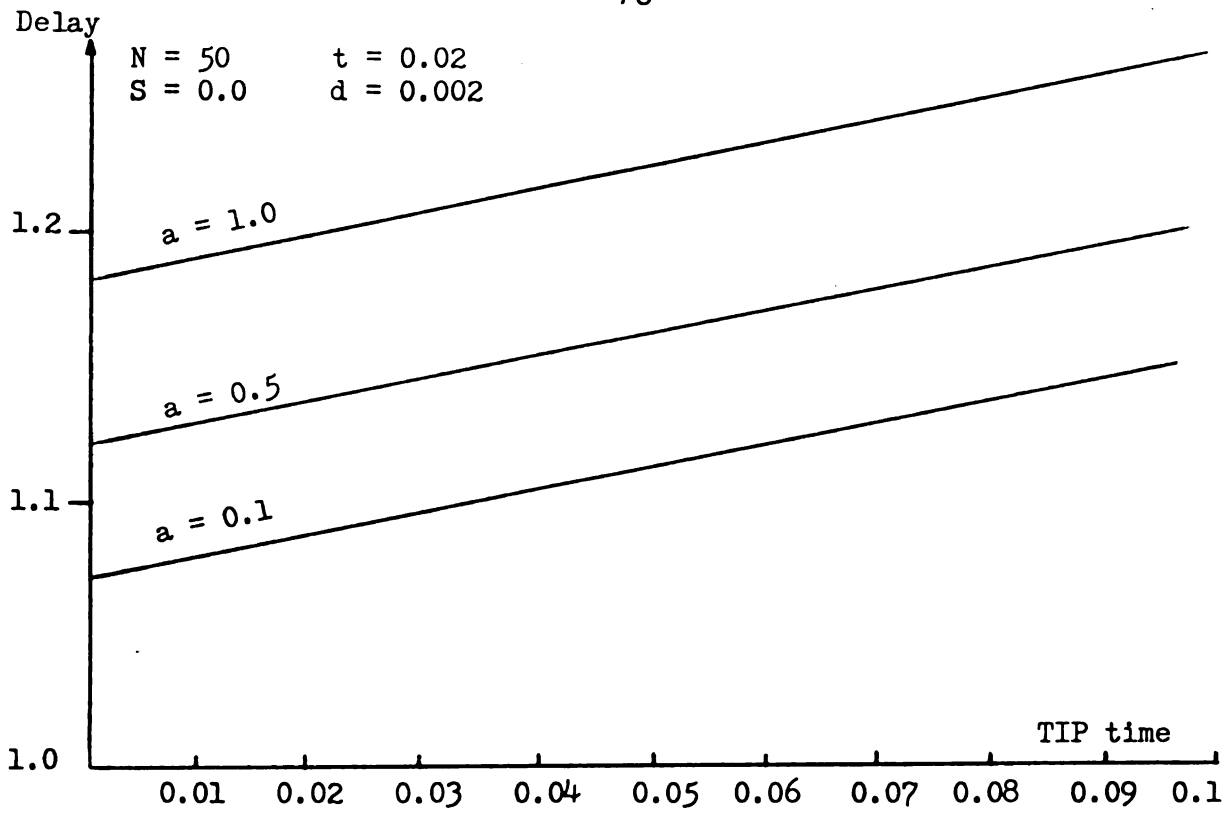


Figure 5-4. The TIP time versus network delay

5.2.3 The Carrier-Sensing Time and the User Population

In an ideal case (as most people assume for their protocols), the time d is negligible for each CIU to detect the absence (or presence) of a carrier and starts its own transmission. Under such an assumption, SDAM's performance is independent of the number of nodes on the network [Li 81a]. However, this is an unrealistic assumption. Since all nodes on the network take turn to sense and access the channel, each node must allow its predecessors enough time to complete their actions before it can safely start

its own. So, the time spent in carrier-sensing by each node will cumulate as the token is passed from node to node. Figure 5-5 shows the degradation of network performance under various values of $\underline{d}$ and N. It is clear that in a heavily populated network (e.g., N>100), even a subtle change in the carrier-sensing time will have a profound effect on the network's performance. However, if we increase the packet size by some factor, then $\underline{d}$ will be relatively decreased by the same factor. Therefore the network's performance can be made much less sensitive to the user population in this manner.

5.3 Exhaustive and Non-exhaustive Transmissions

In analyzing the exhaustive/non-exhaustive transmission disciplines, it is important to specify the type of workload imposed on the network. Here, we are mainly concerned about workloads with uniform Poisson arrivals; no attempt is made to study the situation where unbalanced loads are presented.

The exhaustive transmission discipline allows a node, upon receiving the network token, to transmit all the packets in its buffer, including the ones that arrived during the transmission process. The non-exhaustive discipline, on the other hand, puts a limit to the number of packets that a node may transmit at one time. In practice this limit may vary from node to node; but here, our focus

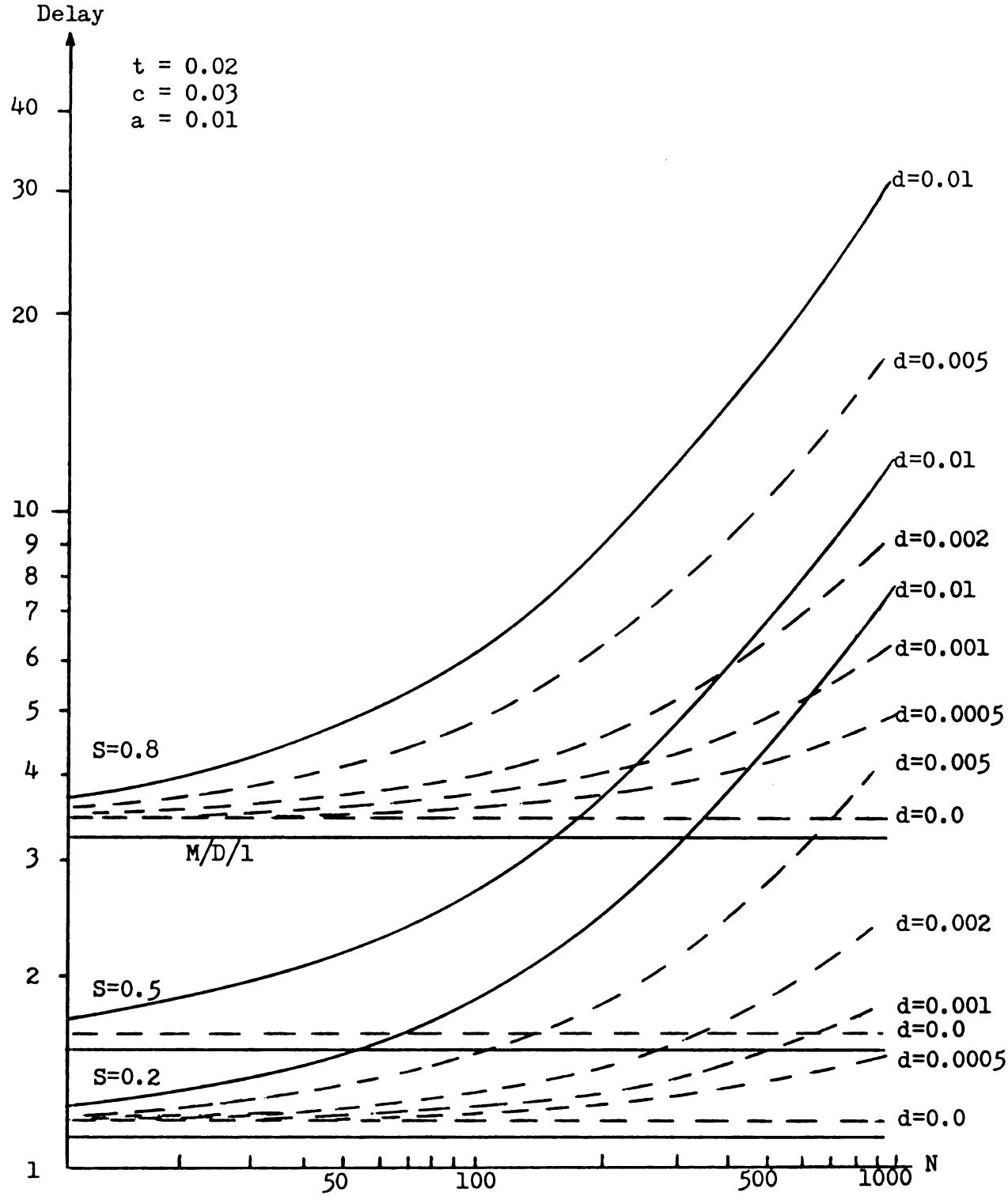


Figure 5-5. Effect of carrier-sensing time

is directed to the case where only one packet is allowed to be transmitted per channel access.

Generally speaking, the exhaustive discipline provides a better average delay and a higher throughput for the network. In extreme cases, a busy node with a large file to transfer may monopolize the entire channel for a long period of time, causing network's throughput to temporarily reach 1, while making other nodes suffer long waiting times. The non-exhaustive scheme, on the other hand, guarantees fairness among the users, and eliminates the above monopoly at the cost of increased token-passing time (hence the increased average delay). However, in light loads ($S<0.5$) where the average number of waiting packets at each node is much less than 1, these two schemes are practically the same. Also, if the bus delay is small ($\underline{a}<0.1$), then the performance of the non-exhaustive discipline remains close to that of the exhaustive one (refer to Figure 5-6). When the bus delay is large, the difference in performance becomes significant (at $\underline{a}=1.0$, $S=0.8$, the non-exhaustive scheme is 20% worse; at $S=0.9$, it is 66% worse).

In terms of the queuing delay distribution, the non-exhaustive discipline has a larger variance than its exhaustive counterpart. This is again due to the fact that the efficiency in consecutive transmissions is compromised by the requirement of fairness. Therefore, the distribution curve is "flatter" and the delay values are spread wider.

Figure 5-7 shows the queuing delay distribution of both exhaustive and non-exhaustive trasnmission disciplines at $\underline{a}$=0.1 and $\underline{a}$=1.0. Figure 5-8 summarizes the mean, standard deviation, median, and the 95 percentile of each of these distributions.

5.4 Packet Size and the Packet Transmission Delay

5.4.1 Different Packet Sizes and Their Normalized Delays

Because of the operating overheads involved in transmitting a packet (e.g., turnaround time, bus propagation delay, etc.), it is intuitively true that the channel bandwidth will be better utilized with larger packets than with smaller packets. One indicator of such a transmission efficiency, known as the "normalized delay", is the ratio of the time a node takes to complete a packet transmission (i.e., the packet's wait time + actual transmission time) to the packet's actual transmission time. This is one main reason why we have set the packet transmission time to be the unit of time in this study. If we reduce the size of the packet, then it has the same effect as increasing the bus propagation delay (and the other overheads, too) by the same factor. In this sense,

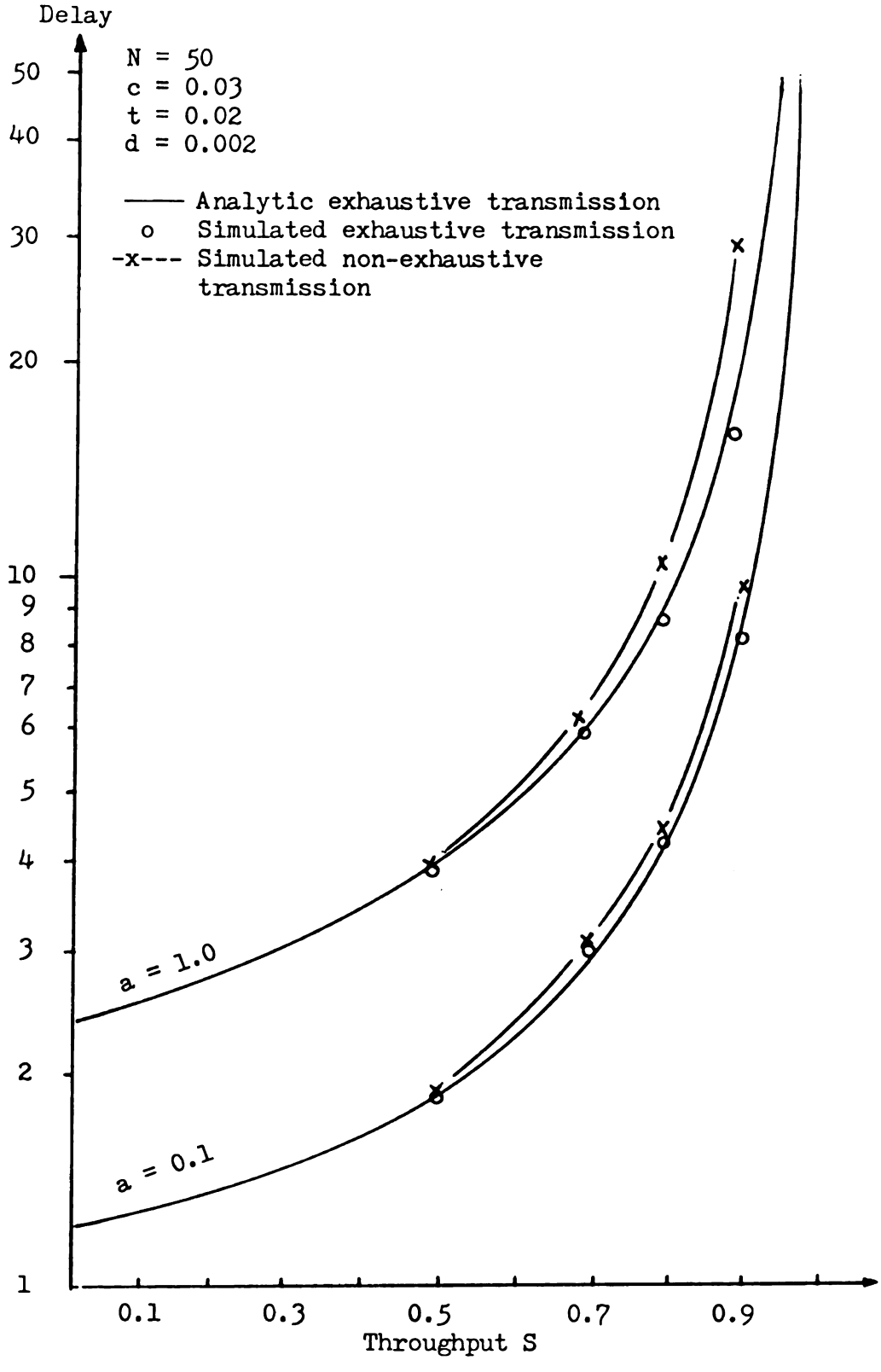


Figure 5-6. Delays of exhaustive/non-exhaustive trans.

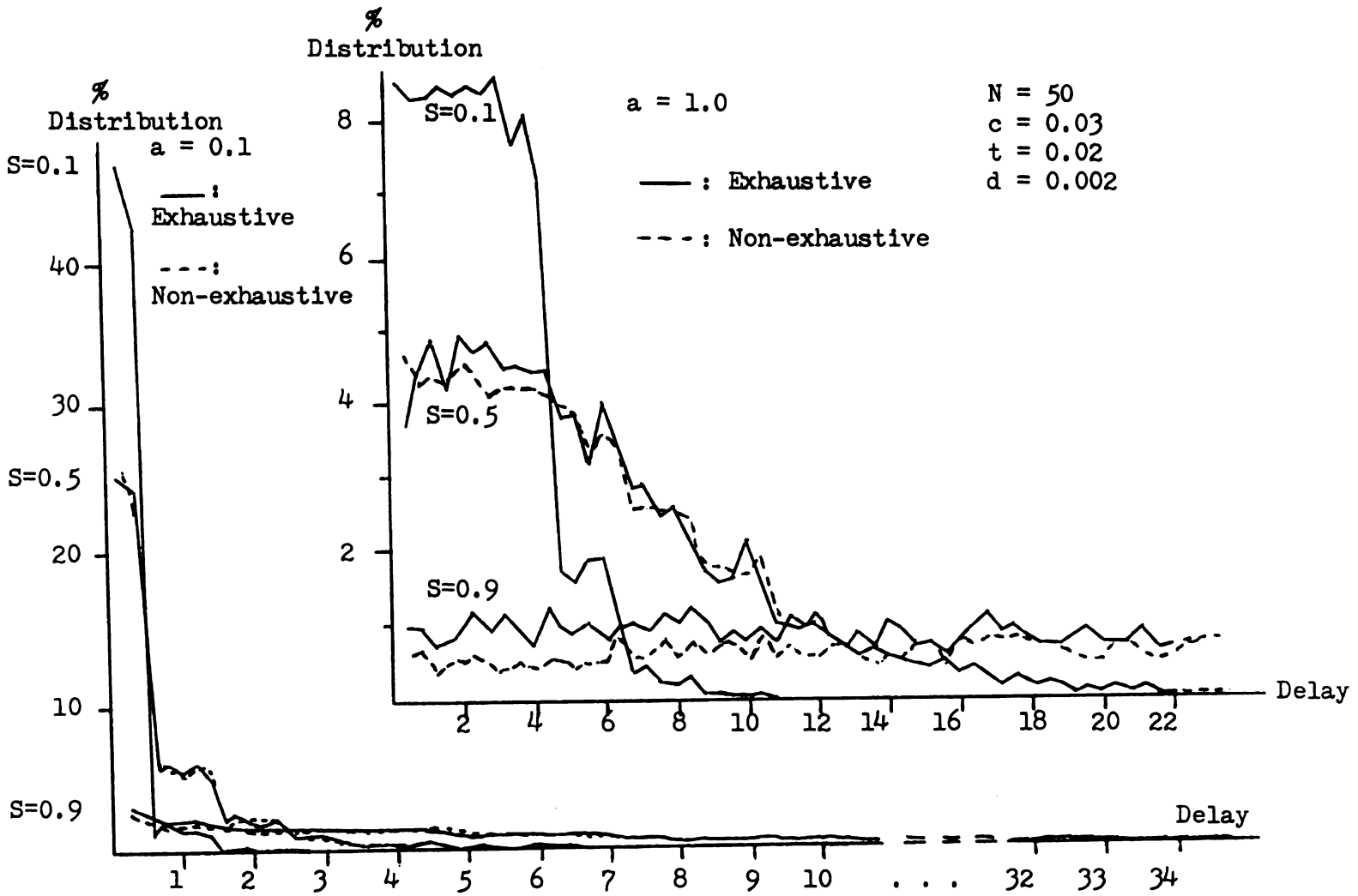


Figure 5-7. Distribution of queuing delays of OE-SDAM

		Propagation Delay a = 0.1					Propagation Delay a = 1.0				
Exhaustive	S	0.295	0.492	0.688	0.787	0.923	0.295	0.492	0.688	0.786	0.893
	Mean	1.496	1.916	2.932	4.208	10.766	2.769	3.738	5.873	8.397	17.681
	S.D.	0.627	1.123	2.344	3.996	10.987	1.300	2.183	4.287	6.534	13.122
	Median	0.3	0.4	1.2	1.8	6.0	1.5	2.3	3.4	5.6	12.2
	95%	1.8	3.2	6.8	11.2	33.6	4.2	7.0	13.4	20.4	***
Non-exhaustive	S	0.295	0.492	0.688	0.787	0.923	0.295	0.492	0.688	0.786	0.893
	Mean	---	1.927	2.977	4.354	12.850	---	3.878	6.503	10.070	29.407
	S.D.	---	1.160	2.522	4.587	15.618	---	2.391	5.292	9.229	31.736
	Median	---	0.4	1.2	1.9	6.8	---	2.3	4.2	6.6	18.4
	95%	---	3.4	6.8	11.2	41.6	---	7.6	15.0	25.8	***

---: omit ***: greater than 100

Figure 5-8. Tabulated distribution of Figure 5-7

the effect of different packet sizes on network's delay performance can be estimated by Figure 5-1 in section 5.1. Although the TIP time, the turnaround time and the carrier sensing time will not be the same in this case, their effects can also be estimated from Figures 5-3, 5-4, and 5-5. These figures clearly show the advantage of the larger-sized packets. However, it must be pointed out that, if one is concerned with the absolute transmission delay (i.e., the "un-normalized" delay), such as in the case of a real-time application, then the smaller packets will still provide a better average respond time.

5.4.2 Fixed-Size Packets versus Mixed-Size Packets

So far our analysis has been concentrated on a network with fixed-size packets (indeed, it is entirely possible to design such a local network). However, in practice, two common types of packets can be readily identified: a short type which contains control messages or terminal traffics, etc., and a long type packets that contain large chunks of data or file transfer. Shoch [Sho79] and Kleinrock & Naylor [Kle74] both observed the so-called "80/20 distribution", where 80% of the total traffic is carried in the 20% of the packet which are long [Sho79].

To investigate the effect of such mixed-sized packets, we let 80% of our packet be 250 bits long, and the rest 20% of the packets be 4000 bits long, so that the average packet size remains to be 1000 bits. Figure 5-9b shows that the delay performance for the mixed-sized packets is significantly worse than that of the fixed-size case. If we look at the delay distribution of these two cases in Figure 5-9a, we can clearly see the reason for this performance degradation. In the fixed-size case, there is a high concentration of delays within one packet's time, meaning a large portion of the packets need only to wait at most one packet's time before it gains channel access. The distribution then tappers off slowly beyond this point. For the mixed-size case, there is also a high density of delays within the 1/4 packet time (due to the deference to the 80% of the short packets); but the majority of the remaining packet delays are evenly spread within 4 packet-time range (caused by deferring to the 20% of the large packets). As a result, the mixed-size case has a significantly larger mean and 95 percentile point as indicated in Figure 5.9a.

5.5 Ease of Implementation of the SDAM protocol

It is not the intention of this study to discuss the implementation details of the SDAM protocol. Rather, we are interested in examining the general requirements of the SDAM

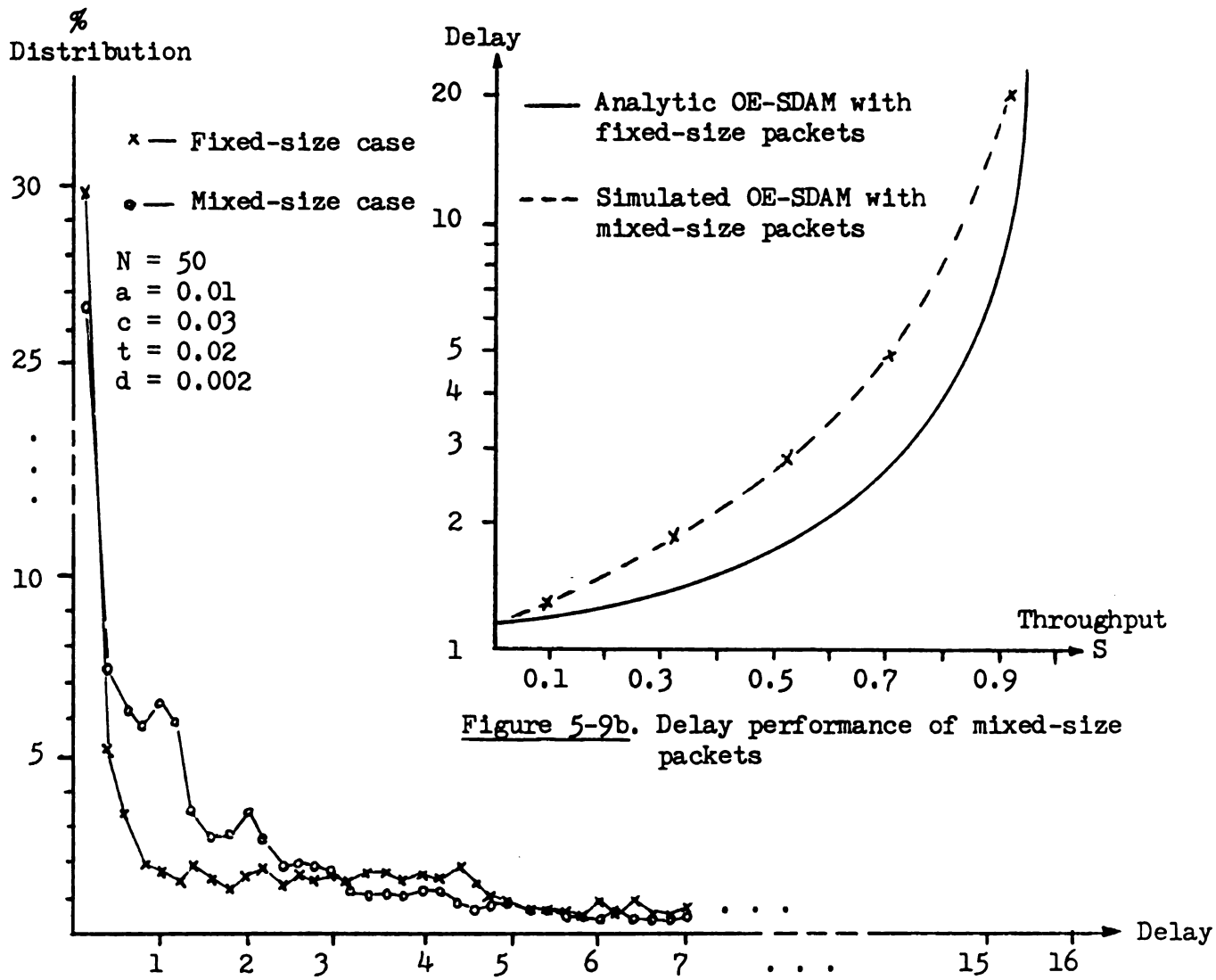


Figure 5-9b. Delay performance of mixed-size packets

Figure 5-9a. Delay distribution of mixed-size packets

schemes and evaulate their advantages in terms of the ease of implementation.

Since SDAM works on a bus network, it implies that all the advantages of a bus network are inherited by the SDAM protocol, such as the absence of the need for routing, the ease of (physically) adding or deleting nodes on the network, and the absence of a need to relay or regenerate messages, etc..

The SDAM protocol employes a very simple algorithm, as has been shown in Chapter 3. Each interface unit may require a few counters (for various countdown procedures), and the carrier-sensing capability, which is a well-developed technology in local area networking. There is no need for a centralized clock, as is the case for TDMA (time-division multiple access) or MSAP. The nodes are synchronized regularly by the presence of the token initializatin packet (TIP) as well as the most recent end-fo-packet signal passing through the bus. Given that each interface functions properly, data collisions will never occur, therefore no complicated backoff-retry schemes need to be implemented to ensure network's stability.

The end-nodes on the bus are the key elements in order for SDAM to work. But these end-nodes are nothing but "intelligent interfaces", which is entirely different from the role played by a central node in a hierarchical network. Should an end-node fail, a simple recovery procedure as will

be described later can be employed to maintain the integrity of the network.

In short, the SDAM protocol only requires technologies that are currently available; and the algorithm of the protocol is simple enough to be implemented into a simple interface unit. These characteristics render SDAM great advantages in network implementation and maintenance.

5.6 Network Reliability and Error Recovery

As we have described earlier (in Chapter 3), each node executes an identical algorithm independently according to the information (e.g., channel status, token direction, etc.) provided through the common channel. Therefore, any single node failure will not affect network's operation. However, network failure could still occur if

(1) the end node fails to generate a TIP;

(2) there is an error in transmission (e.g., a noise on the channel, causing a later node to start transmitting prematurely); and

(3) cable failure occurs, such as a section of the cable or the cable terminators being disconnected.

Pertaining to the first two possibilities, some procedure must be employed to restore the proper network operation from network failure. For the case of OE-SDAM, the error recovery procedure stipulates that:

rule 1: Whenever a node detects an unrecognizable address or token direction, it abandons any transmission attempts until the next token arrives;

rule 2: Each node is pre-assigned a time-out value, whose size varies with the distance between the user-node and the end-node (i.e., the shorter the distance, the smaller the time-out value);

rule 3: if the bus has been sensed idle by a node for a period of time longer than its time-out value, then this user-node may infer that all the nodes with smaller time-out values have failed; therefore, it will generate a (data or control) packet to start the token-passing again.

With this procedure, any error in transmission will be handled by using rule 1, followed by the end-node generating a new token. If the end-node should fail, then the user next to it will detect this fact after its time-out period (rule 2), and can resume the end-node's duty (rule 3). Any subsequent node fuilures can be handled by the same procedure.

For the C-SDAM protocol, if one of the end-nodes should fail, then the other end-node can detect this after a pre-determined time-out period, after which it can automatically switch to the single-end-node OE-SDAM scheme as described previously.

Pertaining to the third possibility, the cable failure, Shoch has an interesting observation on the cable system of the Ethernet at PARL [Sho79], in which he points out that these kinds of failure are extremely rare, and usually caused by human errors. With proper grounding and marking (coloring) of the cable and the cable terminators, these failures can be avoided. However, it is worth noting that, even when the cable is physically cut into several pieces, each piece of the cable (together with the attached nodes) can still form a network under SDAM, provided that the proper cable terminators are added.

CHAPTER 6

DEFINITION OF THE GBRAM PROTOCOL

6.1 Background and Environment

Before we define the GBRAM (group BRAM) protocol, we first briefly review the BRAM (broadcast recognizing access method) and the parametric BRAM protocols [Chl79a]. There are two variants to the simple BRAM, namely, fair BRAM and prioritized BRAM. The former limits each node to transmit at most one packet at a time (i.e., non-exhaustive); the latter allows each node to transmit all the packets in its buffer before relinquishing channel control (i.e., exhaustive). The channel state under BRAM can be viewed as consisting of a sequence of cycles composed of idle, scheduling, and transmission periods, as shown in Figure 6-1. We assume that x_n is a common time epoch known to all

nodes. This epoch, which could be the end of a previous transmission, marks the beginning of the n-th scheduling period. During the scheduling period the time axis is divided into time slots of size a, the end-to-end bus propagation delay, and channel control is granted to one of the ready nodes through the use of a scheduling function H, which specifies at which time-slot nodes will sense the channel and attempt to transmit. For fair BRAM we have:

$$H(n_i, n_j) = \begin{cases} [(n_i - n_j + N) \bmod N] \cdot a & n_i \neq n_j \\ N \cdot a & n_i = n_j \end{cases} \qquad (6.1)$$

where:

n_i = the index of the node wishing to transmit; and

n_j = the index of the node which transmitted last;

N = the total number of nodes on the network.

We note that the above function does not allow any one node

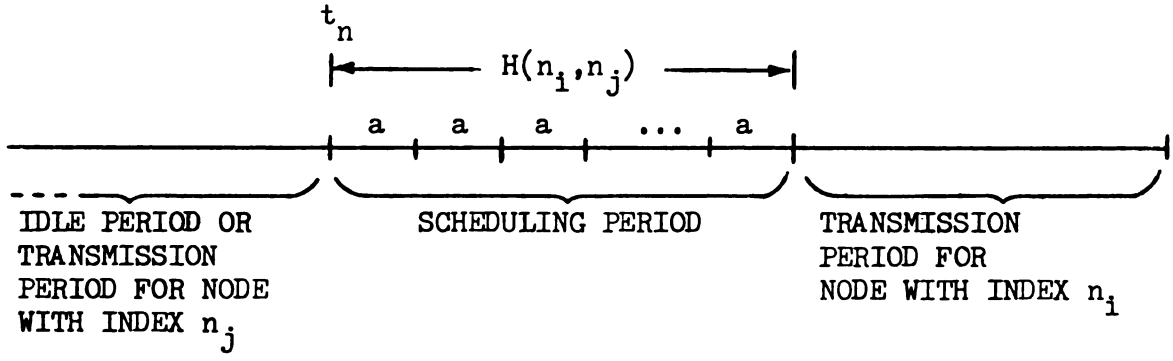


Figure 6-1. Channel scheduling for the BRAM protocol

to hold the channel for more than one consecutive transmission, hence it is used to define the fair BRAM protocol. For the prioritized BRAM, the following H function is used:

$$H'(n_i,n_j) = [(n_i-n_j+N) \bmod N]\cdot a \qquad (6.2)$$

which gives access priority to the node that just transmitted.

Analysis of BRAM shows that for small values of $N\cdot\underline{a}$ product, BRAM provides fair allocation of the channel and overall better throughput-delay and throughput-traffic load performances as compared to other published protocols [Chl79a]. However, as the N a product increases, the length of the scheduling period increases as well, resulting in degradation of performance.

In view of this shortcoming, the parametric BRAM partitions the N nodes into M groups, letting the nodes within each group share a common "group" slot. The scheduling functions in (6.1) and (6.2) become

$$G(g_i,g_j) = \begin{cases} [(g_i-g_j+M) \bmod M]\cdot a & g_i \neq g_j \\ M\cdot a & g_i = g_j \end{cases} \qquad (6.3)$$

and

$$G'(g_i,g_j) = [(g_i-g_j+M) \bmod M]\cdot a \qquad (6.4)$$

respectively, where g_i and g_j are now group indices instead of node indices. With (6.3) or (6.4) collisions can occur between two or more nodes sharing the same group slot, and some type of retransmission scheme must be employed. As one may already have observed, smaller M values tend to reduce the probability of collision, while larger values of M tend to increase the length of scheduling periods but reduce the probability of intra-group collision. An optimal value of M, say M*, is therefore needed to balance these tendencies with respect to a given traffic load G, so as to yield the best throughput, S, or the lowest delay, D. It has also been shown that for this M*, the optimal partition of the N nodes, in the sense of reducing the probability of intra-group collisions to a minimum, is the one which divides the N nodes into groups of equal aggregated arrival rate, which is difficult to achieve in practice.

The complications of the parametric BRAM lead one to wonder if there is an easier way to partition the nodes into groups and yet maintain conflict-free transmissions. A surprisingly straight-forward answer is that such groupings may have already existed in many local networks in the form of node physical locations. For example, we may have a global data bus that runs through several buildings, or from one floor to another. Then the terminal-nodes or device-nodes in a single room may be viewed as a "natural group" (cluster) of nodes on the bus. Typically, the

intr

tota

span

furt

each

to

will

defi

vari

6.2

on

loca

leng

Howe

grou

the

With

subs

atte

intra-group transmission delay is much smaller than the total bus delay. This is especially true if the data bus spans several buildings. It would seem possible, then, to further sub-divide the group slot into mini-slots so that each node in the group can be allocated a unique mini-slot to attempt its transmission. In the following section we will show how this can actually be implemented, giving the definitions of the group-BRAM (GBRAM) protocol and its variants.

6.2 Basic Concept of GBRAM

The GBRAM is basically a two-level BRAM. The N nodes on the network are divided into M groups by their physical locations. Each group is assigned a unique time slot of length $\underline{a}$, just as each node is assigned a slot in BRAM. However, to resolve the contention within each group, the group slot is further divided into sub-slots of length a_1, the intra-group delay, which is just a fraction of $\underline{a}$. Within each group we can then assign each node a unique subslot (or mini-slot) in which transmission can be attempted.

6.3 Network Configuration

Beofe we describe the algorithms of the GBRAM protocol, we reiterate the basic assumptions or requirements about the network configurations where GBRAM will be applied.

1) There are N nodes connected to a common communication medium (e.g., coaxial cable or radio channel) with baseband signalling. These nodes are partitioned into M groups such that each group occupies only a fraction of the total bus length.
2) Each node is assigned a unique index pair (g_i, n_i) as its identification, where g denotes the group index and n_i the node index within the group.
3) Each node is connected to the bus through a communication interface unit (CIU) which functions as the buffer and decoupler.
4) Each node can sense the bus status (i.e., carrier-sense) in a negligible time (or alternatively, we may include this time as part of the time slot or mini-slot).
5) Each node is able to transmit and receive, but not simultaneously; and the "turnaround time" from receiving state to transmitting state (or vice versa) is $\underline{t}$.

6.4 A

W

token"

all no

token

sequer

simil

is sl

2a t

actua

secon

the

grou

ther

in

alwa

6.4 Algorithms of GBRAM

We can envision the GBRAM protocol as having a "virtual token" circulating from one node to another, such that if all nodes in one group have been visited by the token, the token is passed from that group to the next group in sequence (see Figure 6-2). The channel periods of GBRAM are similar to those of BRAM, except that the scheduling period is slightly more complicated (see Figure 6-3). Note that a $2\underline{a}$ time is needed for each group, where the first $\underline{a}$ is the actual group slot used to accommodate the subslots, and the second $\underline{a}$ is the worst-case time for the token to travel to the next group. Furthermore, we assume that the nodes in group g_i do not have any information on how many more nodes there are behind node n_j (the node which transmitted last) in group g_j. Therefore they assume the safest action: they always allow group g_j another full $2\underline{a}$ time units to complete

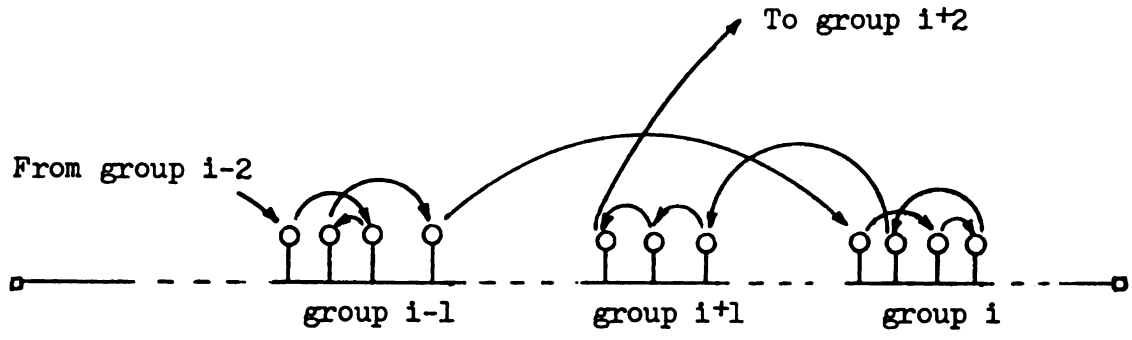


Figure 6-2. Virtual token passing in GBRAM

its

takes

for

for

pro

rul

its group scheduling. The scheduling function therefore takes the form

$$F[(g_i,n_i),(g_j,n_j)] =$$

$$\begin{cases} 2a[(g_i-g_j+M) \bmod M]+a_i(n_i-1) & g_i \neq g_j \\ a_i(n_i-n_j) & g_i=g_j \text{ and } n_i>n_j \\ 2aM+a_i(n_i-1) & g_i=g_j \text{ and } n_i \leq n_j \end{cases} \tag{6.5}$$

for the fair GBRAM, and

$$F'[(g_i,n_i),(g_j,n_j)] =$$

$$\begin{cases} 2a[(g_i-g_j+M) \bmod M]+a_i(n_i-1) & g_i \neq g_j \\ a_i(n_i-n_j) & g_i=g_j \text{ and } n_i \geq n_j \\ 2aM+a_i(n_i-1) & g_i=g_j \text{ and } n_i<n_j \end{cases} \tag{6.6}$$

for the prioritized GBRAM, where a_i is the intra-group propagation delay for group g_i.

Under GBRAM, each node observes the following simple rules:

STEP 1. If the channel when approached by node (g_i,n_i) is sensed idle, then the node schedules its transmission at time $x=x_n+F[((g_i,n_i),(g_j,n_j)]$, where x_n is the last time epoch which marks the end of packet transmission(s) by node (g_j,n_j). If the channel is sensed busy on or before time

IDEL PE
TRANSMI
PERIOD
GROUP W
INDEX e

Fi

6.5

GBRAM
only
Figur
group
that
priori

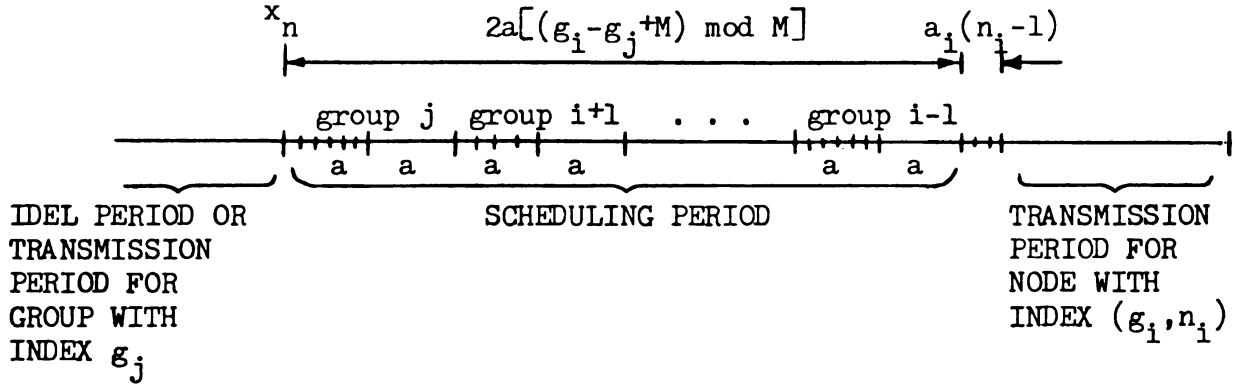


Figure 6-3. Channel scheduling for the GBRAM protocol

x, then step 2 results; otherwise node (g_i,n_i) can start its transmission at time x.

STEP 2. If the channel is sensed busy, then node (g_i,n_i) waits for channel to become idle so that x_{n+1} can be determined, after which it returns to execute step 1.

6.5 Extensions of GBRAM

It is also possible to extend both fair and prioritized GBRAM to the extreme cases. The extreme-fair GBRAM allows only one node in each group to transmit per group slot (see Figure 6-4). To ensure fairness among the nodes in the same group, the intra-group priority can be rotated constantly so that the previously transmitting node will have the lowest priority in the next group slot. The scheduling function

for t

wher

m_i i

the

once

all

The

Fro

t

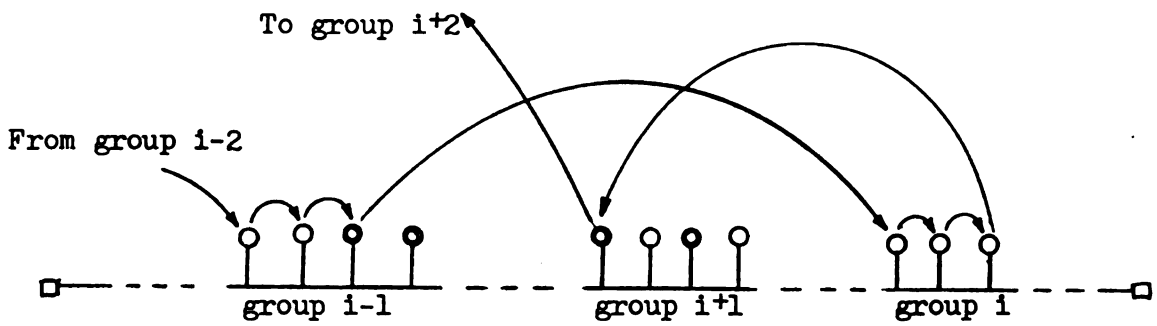
for this extreme-fair case is:

$$L[(g_i,n_i),(g_j,n_j)] =$$

$$\begin{cases} 2a[(g_i-g_j+M) \bmod M] + a_i[(n_i-l_i+1+m_i) \bmod m_i] & g_i \neq g_j \\ 2aM+a_i[(n_i-n_j-1+m_i) \bmod m_i] & g_i=g_j \text{ and } n_i \neq n_j \\ 0 & g_i=g_j \text{ and } n_i=n_j \end{cases} \quad (6.7)$$

where l_i is the node that transmitted last in group g_i, and m_i is the number of nodes in group g_i.

On the other hand, the extreme-prioritized GBRAM allows the nodes in a group to be visited by the token more than once; and the token will be passed on to next group only if all nodes in the current group are idle (see Figure 6-5). The scheduling function for this case becomes:



Each group is allowed to have at most one node transmission.
● indicates the node with packet to send.

Figure 6-4. Token passing of the extreme-fair GBRAM

L

Both

transr

6.6

form

diffe

(logi

netwo

funct

From
grou

A
o

Fig

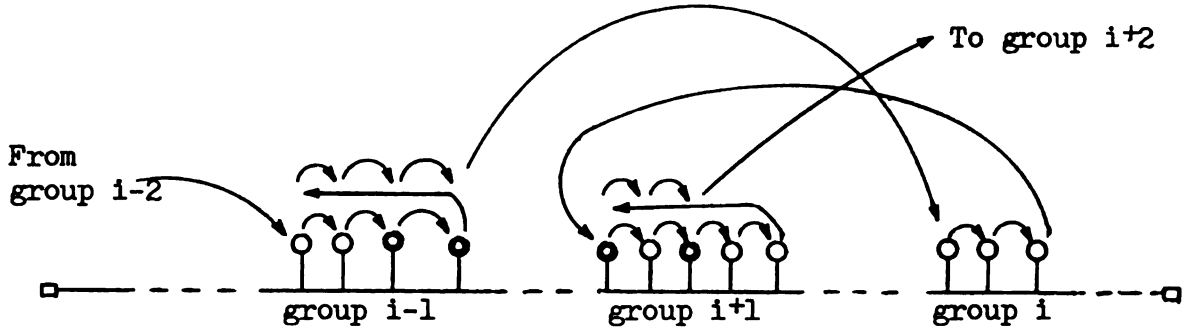
$$L'[(g_i,n_i),(g_j,n_j)] =$$

$$\begin{cases} 2a[(g_i-g_j+M) \bmod M]+a_i(n_i-1) & g_i \neq g_j \\ a_i[(n_i-n_j+m_i) \bmod m_i] & g_i=g_j \end{cases} \qquad (6.8)$$

Both GBRAM extensions above still assume exhaustive transmissions at each node.

6.6 Addition and Deletion of Nodes on the Network

Since the "network token" in GBRAM takes on a different form than in the SDAM protocol, we need to look into a different approach for adding nodes onto the token-passing (logical) ring. For this we assume that there is a set of network monitor nodes on the netowrk [IEE81], whose function, in addition to being user nodes, is to perform a



A group holds the token until all nodes in the group are idle.
● indicates a node with a packet to send.

Figure 6-5. Token passing of the extreme-prioritized GBRAM

limited network management, such as the recovery of a lost token. These monitor nodes differ from a central node in that they do not govern the entire operation of the network; rather, they serve only when token initialization or network recovery is required. The network token may be lost due to channel noises or a collision between two sign-on requests. Under such a circumstance the monitor nodes hold the token-recovery responsibility.

The procedure for an active node to sign off from the network is straightforward: it simply broadcasts its intention, then waits for an acknowledgment from one of the monitor nodes to ensure that the sign-off message has been properly received. Only the nodes that belong to the same group and were originally sequenced behind the signing-off node need to update their sequence numbers accordingly.

As for the nodes wishing to sign on to the network, there are several procedures proposed for maintaining a dynamic logical ring [Liu81b,IEE81]. But here we will present only one of the simplest procedures as follows:

1. The set of monitor nodes periodically (or dynamically) broadcasts a sign-on inquiry onto the common channel.
2. All active nodes, upon receiving the sign-on inquiry, will temporarily interrupt their token passing scheduling functions, and remain silent.
3. The node wishing to sign on, on the other hand, will

transmit a reply message (a sign-on request) which contains the node's location information (e.g., the cable mark number as described in section 3.7; or the zone in which the node is situated; or the names of the nearest two neighbors, etc.).

4. The monitor node, upon receiving the reply message, will assign this node into one of the groups according to its location, and broadcasts this addition onto the network.
6. The token-passing sequence will then be resumed from the monitor node that originated the sign-on inquiry.
7. In case there is more than one node attempting to sign on at the same time, a collision will occur. In this case, a randomized delay will elapse before each of the nodes will attempt to transmit again.

As a result of continuing sign-on and/or sign-off operations, a group may eventually become empty or overflowed with nodes. It is therefore the network manager's responsibility to frequently monitor the network's population and decide when to merge or subdivide the groups of nodes. This decision is then broadcasted through a monitor node to effect a network-wide update of addresses.

CHAPTER 7

ANALYSIS OF THE GBRAM PROTOCOL

7.1 General Case Analysis

We begin our analysis of the prioritized GBRAM protocol by assuming a bus-network with a general topology wherein the N nodes are divided into M groups. For a group (say group k), m_k represents the number of nodes in the group, and a_k is the maximum intra-group delay of the group. Since the nodes take turns to access the common bus, the analysis of polling systems (see section 4.3 of Chapter 4) can again be applied, where in our case the polling time is the time it takes to pass the virtual token from one node to the next. The expected queuing delay E(D) (excluding transmission time) is given by Equation (4.13):

in

a_o=

pas

var

bet

2a-

muc

alg

the

(id

gro

com

thi

her

($m_{.K}$

whe

gro

inc

gro

R b

be

$$E(D) = \frac{a_o\delta^2}{2r} + \frac{1}{2}\frac{S}{1-S} + \frac{a_o}{2}(1-\frac{S}{N})(1+\frac{Nr}{1-S}) \qquad (7.1)$$

in terms of packet transmission time, where in GBRAM $a_o=\min\{a_k\}_1^M$ is the mini-slot size.

It remains to solve for r, the average time needed to pass the token to the next node in sequence, and δ^2, its variance. For this we note that the token passing time between group g_k and the next group in sequence is $2a-[(m_k-1_k)]\cdot a_k]$, where the first term ($2\underline{a}$) represents how much time the next group is willing to wait (see the algorithm of GBRAM in section 6.2); and the second term is the time it takes the token to go through the remaining (idle) nodes in group g_k, i.e., the time token remains in group g_k after a packet transmission by node 1_k. While complicated occupancy theory may be applied to calculate this quantity, a much simpler approximation can be made here. At throughput S=0, since all nodes are idle, $(m_k-1_k)\cdot a_k$ is simply one group slot $\underline{a}$. On the other hand, when S approaches 1, 1_k is likely to be the last node in the group, therefore, $(m_k-1_k)\cdot a_k \dashrightarrow 0$. In general, as S increases from 0 to 1, the token passing time between two groups increases from $\underline{a}$ to $2\underline{a}$. Thus, the token passing time R between two consecutive nodes (regardless of grouping) can be approximated as

$$R = \begin{cases} (1+S)a/a_o & \text{mini-slots with probility } M/N \\ a_k/a_o & \text{mini-slots with probability } (m_k-1)/N, \\ & k=1,2,\ldots,M. \end{cases} \tag{7.2}$$

Later simulation studies indicate that this is a good approximation.

From (7.2) we have

$$r = E(R) = (1+S)(a/a_o)M/N + \sum_{k=1}^{M}(a_k/a_o)(m_k-1)/N$$

$$= \frac{1}{N}[(1+S)Ma/a_o + \sum_{k=1}^{M}(m_k-1)a_k/a_o] \tag{7.3}$$

and

$$\delta^2 = Var(R) = \frac{1}{N}\{M[(1+S)a/a_o - r]^2 + \sum_{k=1}^{M}(m_k-1)[a_k/a_o - r]^2\} \tag{7.4}$$

The analysis of fair GBRAM and the two extreme GBRAM protocols are considerably more difficult. For the case where both N and $\underline{a}$ are small (say $N \leq 10$ and $\underline{a} < 0.1$), the M/D/1 queuing equation or the prioritized GBRAM may be used to approximate these GBRAM variants (see [Chl79a] for example). For large N and $\underline{a}$ values, simulation results must be used.

7.2

sim

gro

nod

div

eac

bei

gro

wh

an

fr

nu

$\frac{a}{a_c}$

T

i

s

7.2 Worst Case Analysis

In order to compare the performances of GBRAM and the simple BRAM, let us consider the case where no physical groups exist among the nodes on the network; namely, all nodes are evenly spaced on the bus. So assuming we can divide the total length of the bus into M sectors, so that each sector captures a number of nodes, with the number being approximately N/M (i.e., $\lfloor \frac{N}{M} \rfloor$ or $\lfloor \frac{N}{M} \rfloor + 1$ nodes per group). Hence we have $a_o = a/M$, and (7.1) becomes:

$$E(D) = \frac{a\delta^2}{2Mr} + \frac{1}{2}\frac{S}{1-S} + \frac{a}{2M}(1-\frac{S}{N})(1+\frac{Nr}{1-S}) \qquad (7.5)$$

where

$$r = \frac{1}{N}[(1+S)M^2+(N-M)] \quad \text{mini-slots,} \qquad (7.6)$$

and

$$\delta^2 = \frac{1}{N}\{M[(1+S)M-r]^2+(N-M)[1-r]^2\} \qquad (7.7)$$

from (7.3) and (7.4), respectively.

Since the intra-group delay in this case is $a_o = a/M$, the number of nodes that can be placed in a group cannot exceed $\frac{a}{a_o}+1 = M+1$ in order to ensure collision-free transmissions. Therefore the total maximum number of nodes on the network is M(M+1). The optimal M, say M*, can then be chosen as the smallest M such that $M(M+1) \geq N$.

7.

sl
ma
po
ma
to
the
of
the
nec
Reg
per
min

bus
The
a_0(N
node
intr
[a+a
becor

7.3 Selection of the Optimal Number of Groups

In the previous analysis, it was assumed that the time slot assigned to each group is of a fixed size $\underline{a}$, the maximum bus propagation delay. Therefore, the selection policy for the optimal number of groups, M*, was to put as many nodes in a group as possible (i.e., M+1 nodes) in order to fully utilize the group slot, and at the same time reduce the number of groups. In practice, however, the selection of M depends heavily on the network configuration, i.e., how the nodes are spread on the bus. Furthermore, it is not necessary that the size of a group slot be restricted to $\underline{a}$. Regarding to the latter observation, the following analysis pertains to the choice of the optimal M* value which minimizes the transmission delay on the channel.

Again, assume that the N nodes are evenly spaced on the bus, and are therefore evenly partitioned into M groups. The intra-group delay $a_o = a/M$ is then the mini-slot, and $a_o(N/M - 1)$ is the group slot required to hold the (N/M) nodes of a group. as we have discussed in section 6.2, the intra-group token passing time varies between $\underline{a}$ and $[a+a_o(N/M - 1)]$ according to the traffic load. Therefore, r becomes

$$r = \frac{1}{N}\{M[\frac{a}{a_o} + S(\frac{N}{M} - 1)] + (N-M)1\}$$

$$= \frac{1}{N}[M^2 + S(N-M) + (N-M)] \quad \text{mini-slots,} \qquad (7.8)$$

and our goal is to find the M that minimizes

$$\bar{D} = E(D) = \frac{a\delta^2}{2Mr} + \frac{1}{2}\frac{S}{1-S} + \frac{a}{2M}(1-\frac{S}{N})(1+\frac{Nr}{1-S}) \qquad (7.9)$$

We note from the previous analysis that the first term in (7.9) contributes little to the delay function and can be ignored. The term S/2(1-S) is a constant relative to M. Therefore it is sufficient to minimize:

$$\Delta = \frac{a}{2M}(1-\frac{S}{N})(1+\frac{Nr}{1-S})$$

$$= \frac{a}{2M}(1-\frac{S}{N})(1+\frac{M^2+(S+1)(N-M)}{1-S}) \qquad (7.10)$$

with respect to M. After some routine manipulations, the optimal M, M*, is calculated as:

$$M^* = \sqrt{(N+1)(S+1)} \qquad (7.11)$$

Equation (7.11) indicates that M* varies as the system load changes, that it differs from the optimal M selected from our earlier analysis (see Figure 7-1), and that the delay surface around the M* value is rather flat, as is shown in Figure 7-2. So, a minor deviation from M* will not greatly affect network's performance. In fact, over-estimation of M* may be a safe strategy.

When the nodes are not evenly spaced on the network, the above analysis does not apply. Nevertheless it can be served as a guideline or a starting point for the selection of the optimal M.

4

3

2

1

Figur

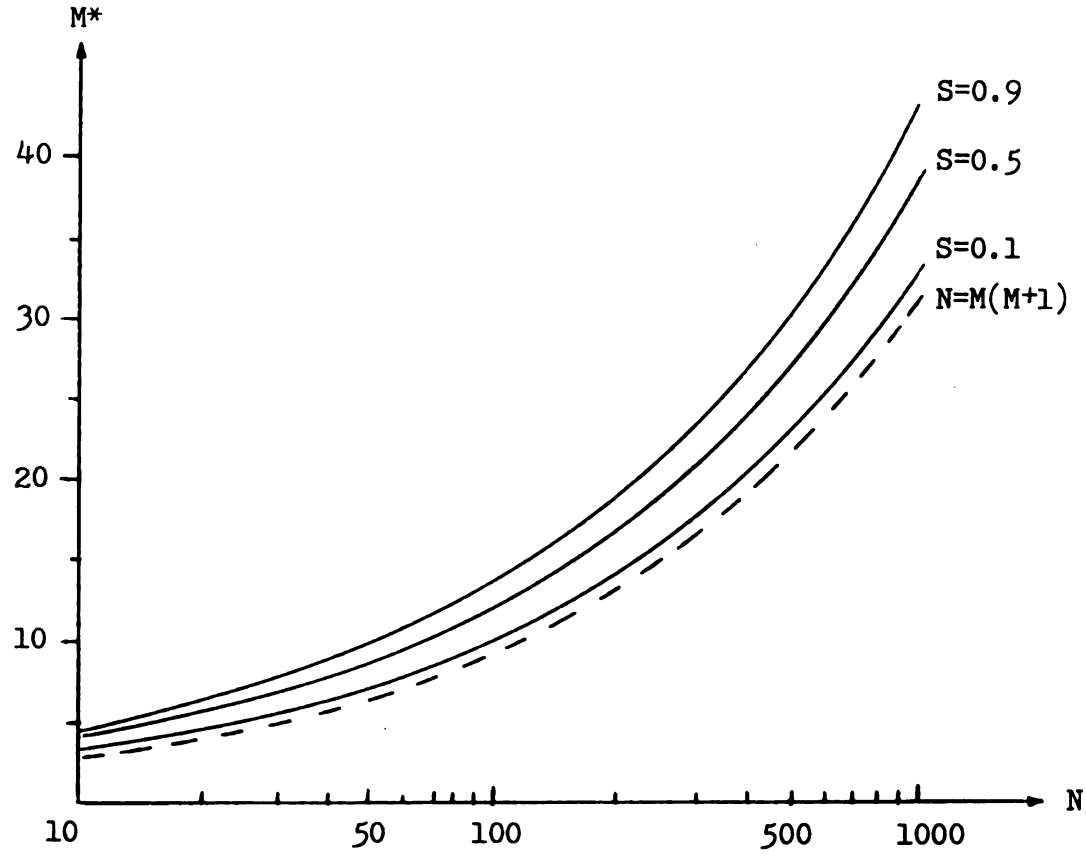


Figure 7-1. Optimal grouping for different channel loadings


Delay

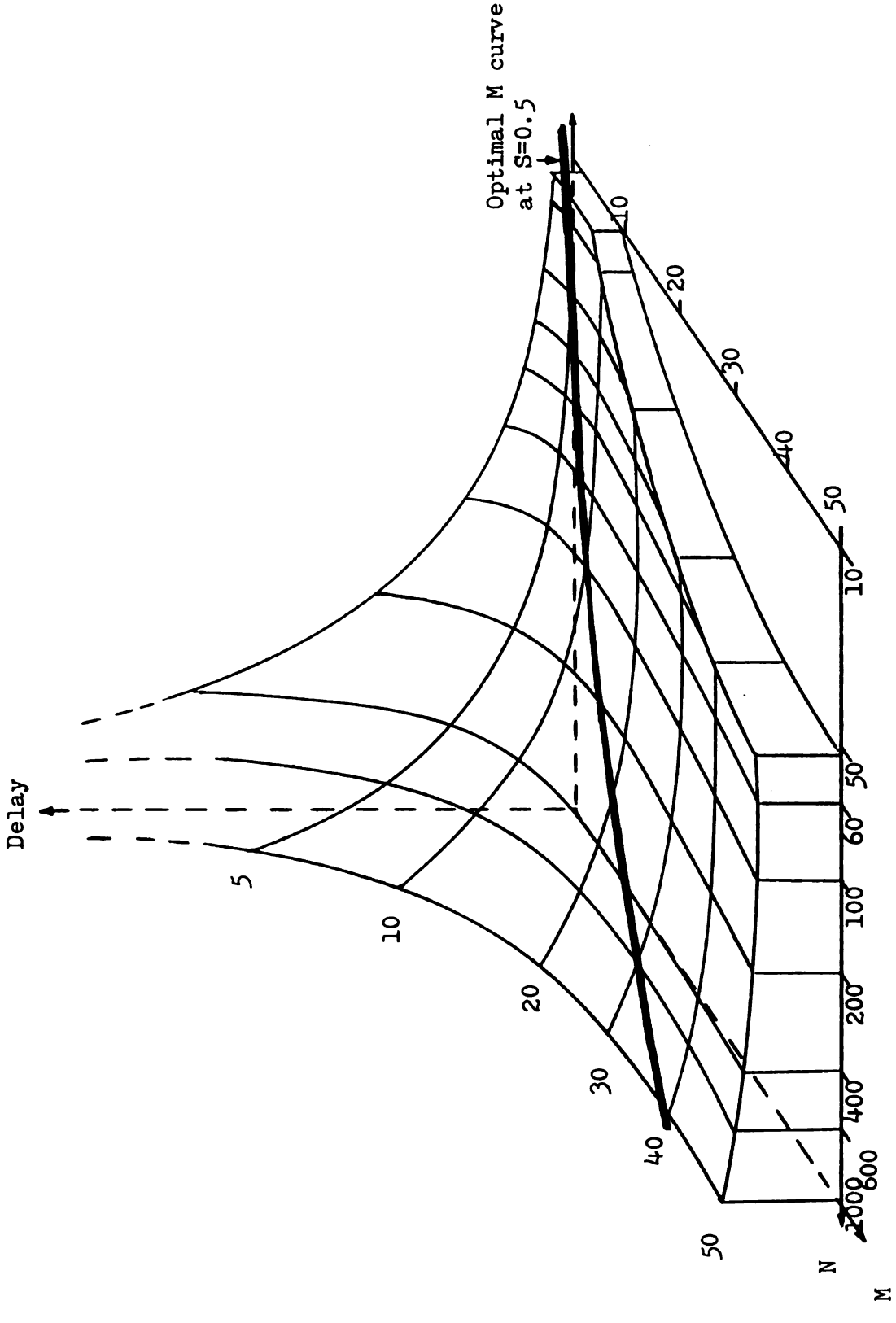


Figure 7-2. Queuing delays versus network groupings

CHAPTER 8

PERFORMANCE OF THE GBRAM PROTOCOL

Basically, SDAM and GBRAM share many common characteristics, such as the virtual-token concept, the use of carrier-sensing capability for conflict-free transmissions, and the decentralized controls, etc.. As a result, many of the performance issues bear strong resemblances, such as the effect of turnaround time and the effect of different or variable packet sizes, etc.. Hence they will not be repeated in this chapter. In the following, we shall concentrate on the throughput-delay performance of GBRAM under various bus propagation delays. For convenience, we assume that the carrier-sensing time of each node is included as a part of the time-slot (or mini-slot) in which the node senses the channel and attempts to transmit its packet.

8.1

eval
mode
simu
para

The
the
ex
an
ex
ch
in
ar
pr
sc
wi
d
t
m
c

8.1 The Throughput-Delay Performance of GBRAM

To verify the analysis in the previous chapter and to evaluate the performances of the GBRAM variants, simulation models using GPSS language have been developed and simulation runs were conducted with the following parameters:

N = 50 nodes (evenly spaced),

M = 7 groups, and

$\underline{a}$ = 0.01, 0.1, 0.5, and 1.0.

The simulation results are plotted in Figure 8-1 along with the analytic results obtained from equation (7.5). The extreme-prioritized GBRAM performs the best, while the fair and the extrem-fair GBRAM perform the worst. This is to be expected, since the former uses the least amount of changeover time while the latter do the opposite. However, in small to medium values of $\underline{a}$ ($\underline{a} \leq 0.1$), their performances are close to each other. It suggests that, while the prioritized scheme gives better performance than do the fair schemes, the fair GBRAM guarantees fairness to all nodes while offering competitive performance, and therefore deserves some special consideration. As $\underline{a}$ value increases, the trade-off between fairness and delay performance becomes more apparent, and measures for tuning the network configuration must be considered.

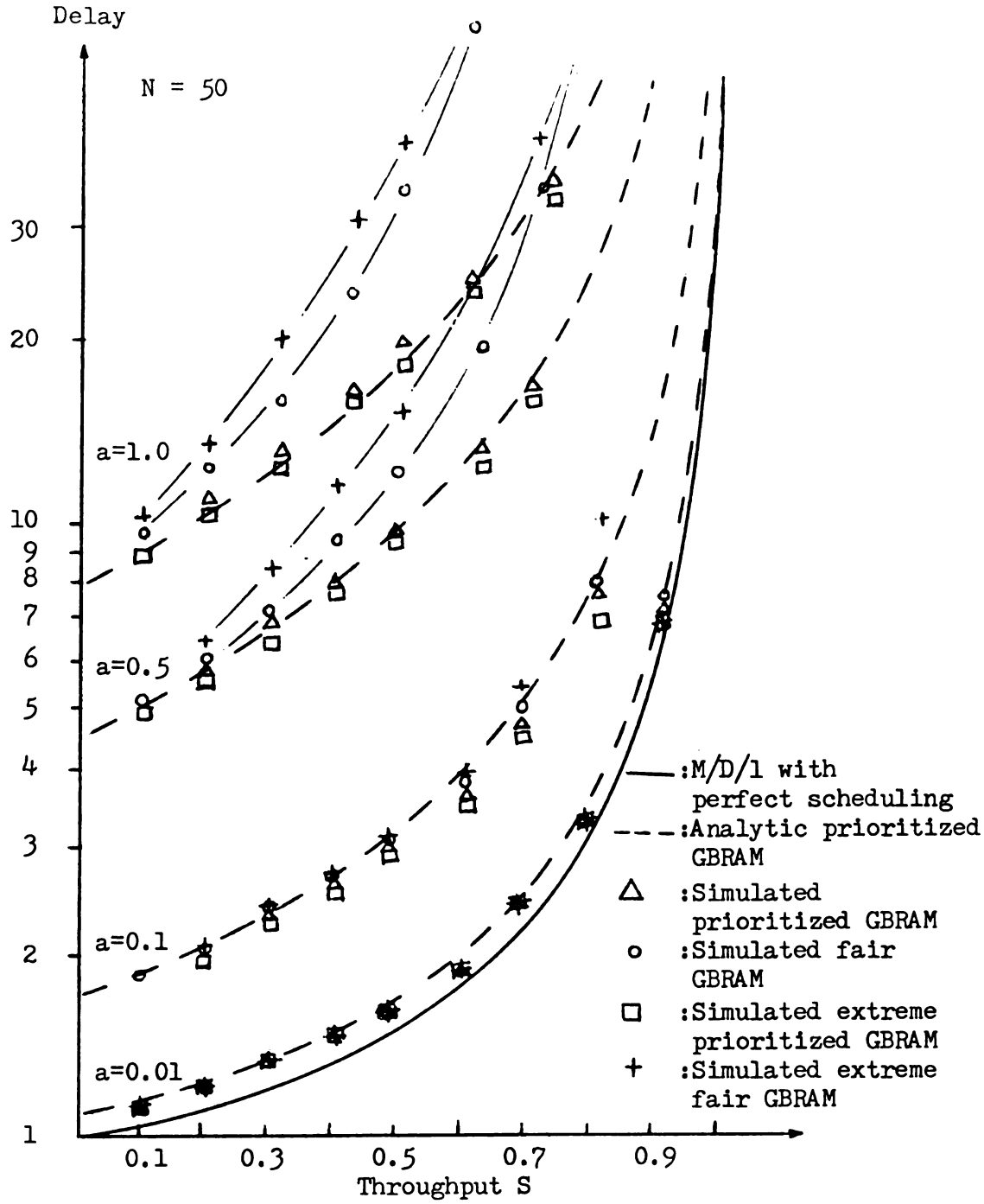


Figure 8-1. Throughput-delay performance of GBRAM

curv

For

1,0(

E(D)

and

perf

sig

$\underline{a}$=0.

E(D

8.2

bus

bus

nod

(e.

gro

arb

hav

cha

int

ari

alg

For varying N and $\underline{a}$ values, GBRAM form a family of curves with a common asymptote at S=1.0 (see Figure 8-2). For small values of $\underline{a}$ ($\underline{a}$=0.01) GBRAM can easily support 1,000 nodes with low delays (for example, at S-->0, E(D)=1.314 as compared to 1.0 of M/D/1 perfect scheduling, and 6.005 of BRAM and MSAP). As $\underline{a}$ increases, the performance of the network degrades, but GBRAM still shows a significant improvement over BRAM (for N=200 and S-->0, at $\underline{a}$=0.1, E(D)=2.388 for GBRAM and 11.05 for BRAM; at $\underline{a}$=1.0, E(D)=14.884 for GBRAM, and 101.50 for BRAM).

8.2 Ease of Implementation and Network Reliability of GBRAM

Whereas the SDAM protocol is restricted to a branching bus topology, the GBRAM protocol can be implemented in both bus networks and radio-channel networks. Furthermore, the nodes in a group need not be sequenced in a fixed order (e.g., from left to right as in the SDAM protocol); and the groups of nodes can be placed on the network in any arbitrary order. This allows the network configuration to have much more flexibility and the nodes more mobility.

The scheduling algorithm described in the preceeding chapter is simple enough to be implemented in a simple interface unit: only carrier-sensing and some basic arithmatic operations are required to carry out the algorithm. There is no need to have any end-nodes to

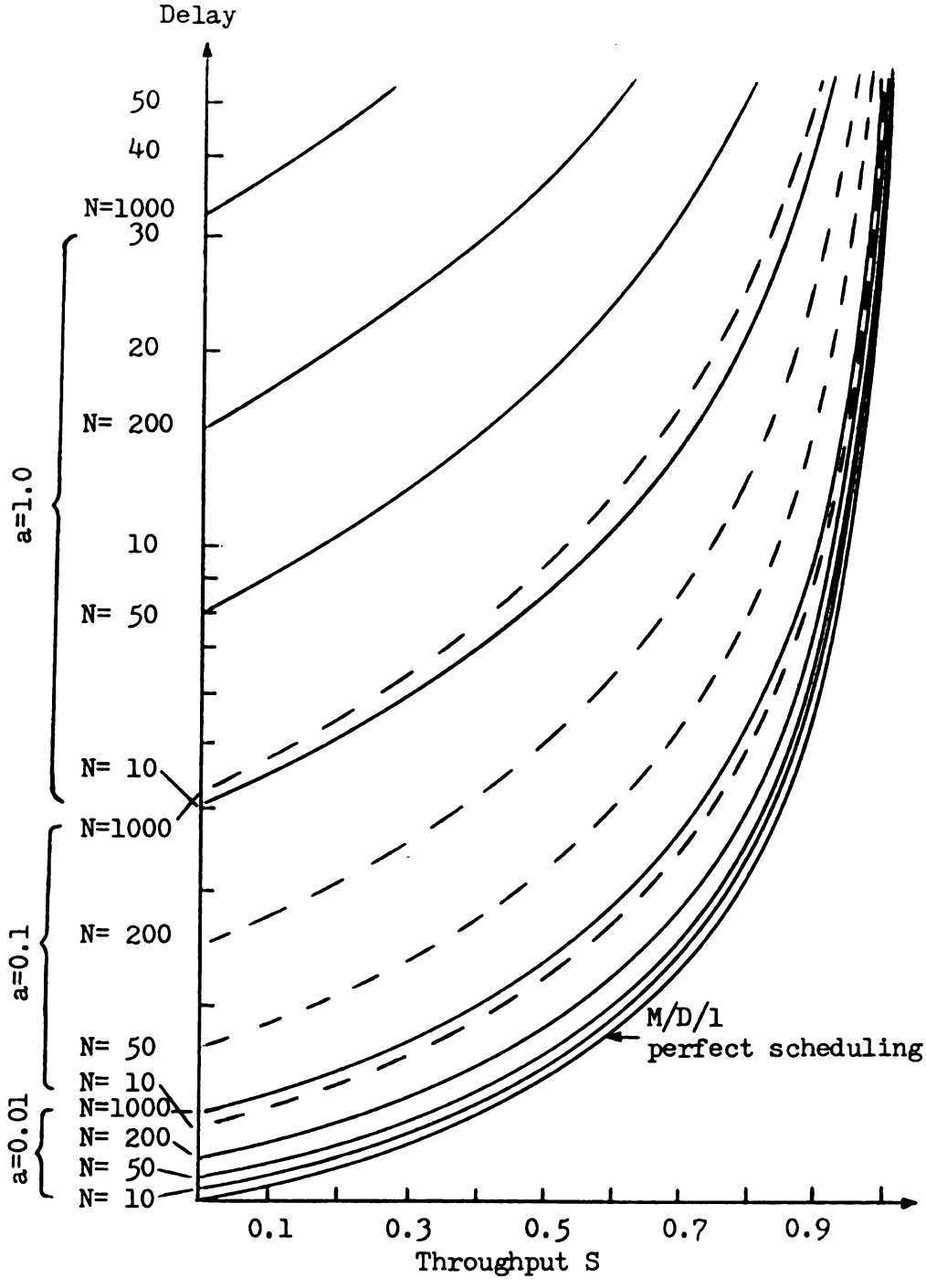


Figure 8-2. Delay performance of prioritized GBRAM

generate the token, although a set of monitor nodes is required for the network's initial setup and token recovery. This set of monitor nodes are assigned different priorities. If for some reason the token is lost in the network, then the node with the highest priority will generate a new token after a time-out period. Should this node fail, then the monitor node with the second highest priority will attempt to recover the token after another time-out, and so on. Hence the network reliability can be easily maintained.

CHAPTER 9

COMPARISON OF PERFORMANCES

In this chapter, we shall compare our SDAM algorithm with two other similar schemes: the Al scheme proposed by Eswaran et. al. [Esw79,Ham80], and the BID system proposed by Ulug et. al. [Ulu81]. These two schemes also attempt to minimize the network control's changeover time in a decentralized environment and still provide collision-free transmissions. Both their similarities and dissimilarities, as well as their performance characteristics, will be presented.

The performance of SDAM and GBRAM will also be compared with other popular protocols. Specifically, we will select two protocols: CSMA/CD [Tob79] and BRAM [Chl79a] (and MSAP[Kle77a]) for this comparison study for the following reasons:

1) both protocols have decentralized controls and easy

implementations;

2) both protocols are well-known and well-analyzed in the literature; and both give very good performance under various conditions;
3) both protocols have been used for comparing the performance of other protocols in their respective categories (i.e., random access techniques and token-passing techniques). Therefore they can be used as the yardsticks for comparing SDAM and GBRAM with other protocols not discussed here.

The M/D/1 optimal scheduling, which defines the best possible performance for a single-server system with Poisson arrivals and constant service times, will also be used to determine the absolute performance of the SDAM and the GBRAM protocols.

9.1 Comparison of SDAM and the A1 Scheme

In the A1 scheme, the network control is distributed. But it requires a separate logic control wire to propagate an one-way logic signal from one end of the bus to the other. The set of N nodes (or ports) are numbered sequentially from left to right. Associated with each port J is a flip-flop S(J), called the "send" flip-flop. This flip-flop is connected to the control wire to its right, as

shown in Figure 9-1. The signal P(J) tapped at the control wire to the left, on the other hand, is the inclusive OR of the send flip-flops of all ports to the left of port J.

If we denote by T the end-to-end bus propagation delay (plus a small fixed quantity), B(J) the busy/idle status of the bus as seen by port J, and R(J) the propagation delay along the control wire from port J to the right-most port, then the access control algorithm of A1 can be stated as follows:

1. Set S(J) to 1 when there is a packet to be sent.
2. Wait for a time interval R(J)+T.
3. Wait until B(J)=0 and P(J)=0; then begin transmission of the packets, simultaneously resetting S(J) to 0.

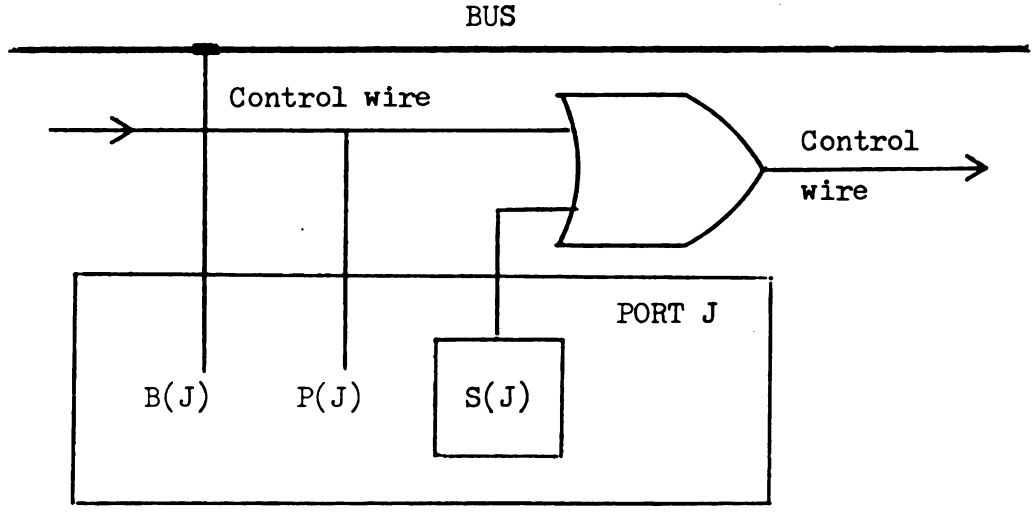


Figure 9-1. Port interface logic of the A1 scheme

The analysis in [Ham80] shows that the A1 scheme does indeed provide conflict-free channel accesses among the users, and is highly efficient when there are consecutive transmissions from different nodes. In extremely light loads, the control overhead for transmitting a packet at any port J is just R(J)+T, which is a significant improvement over those conventional decentralized demand-assignment schemes. However, a node on the right part of the bus may be blocked indefinitely from transmission if the nodes to its left alternately have packets to send [Ham80].

Compared with the SDAM protocol, the A1 scheme is similar to the OE-SDAM, where the token is passed single-directionally from one end of the bus to the other. But unlike the A1 scheme, SDAM does not require a separate control wire, hence the propagation delay R(J) on the control wire can be eliminated. More importantly, OE-SDAM provides fair channel-access to each of the nodes; no user will be blocked from transmission indefinitely.

The performance analysis of the A1 scheme in [Ham80] did not take into account protocol overheads such as node's turnaround time and carrier-sensing time. No explicit throughput-delay relationships were available for a general local network, although a three-node network was used as a case study for the unfairness of the queuing delays mentioned earlier. A summary of the comparison between SDAM and the A1 scheme is listed in Figure 9-3.

9.

La
co
se
"i
ri
Th
wi
(s
ar
ma
BI

9.2 Comparison of SDAM and the BID System

The BID system is developed by Ulug et. al. at the GE Labs [Ulu81]. In this scheme, the nodes are connected to a common bus via bus interface units (BIUs), and are numbered sequentially from one end of the bus to the other. An "implicit" token is passed back and forth (called right-sweep or left-sweep) between the end nodes of the bus. This implicit token is actually the absence of packets within a certain time interval following a passed-by packet (see Figure 9-2). These setups, along with BID's algorithm, are very similar to the C-SDAM protocol. However, several major differences exist between SDAM and the BID system: (1) BID uses data collision to maintain several levels of

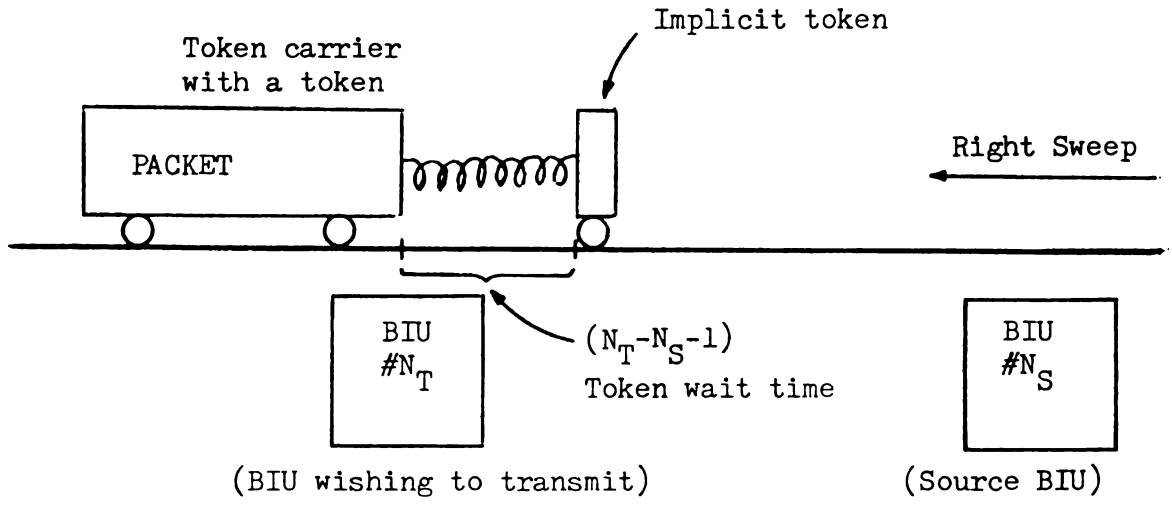


Figure 9-2. Implicit token of the BID system

priority among the packets; a higher-level packet can force a collision with an on-going lower-level packet, thus "rob away" the latter's channel access. (2) BID works only on a single bus (although this bus may be shaped in star-like or ring-like); and (3) BID does not make provisions for a single-directional token passing as does the OE-SDAM, which allows complete fairness among users on the bus.

The performance analysis of BID in [Ulu81] also make use of Konheim and Meister's polling formula (see equation (4.9) in Chapter 4). But they fail to observe that the polling sequence are different, and that the queuing delays are uneven for different users on the bus. Furthermore, they do not distinguish between the decoding time (or turnaround time, which does not cumulate during token passing) and the carrier-sensing/signal rising time, which will cumulate from node to node along the token-passing route. Consequently, a small fraction of time is wasted at each of the nodes. This could have a significant effect on network's performance if the user population is large (as have been pointed out in section 6.3, Chapter 6). A summary of the comparison between SDAM and the BID system can also be found in Figure 9-3.

	SDAM	A1	BID
Control Strategy	decentralized, virtual token	decentralized, control wire	decentralized, implicit token
Topology	branching or single bus	single bus	single bus
Data Collision	no	no	collision between different priority packets
Special Requirements	one or two end-nodes; carrier-sense; interface units	control wire; "send" flip-flop; port interface logic	two token-generators; carrier-sense; interface units
Token or Control Passing direction	either bi-directional or uni-directional	generally uni-directional	bi-directional
Overheads Considered	bus propagation delay; token initialization packet (TIP) time; node's turnaround time; carrier-sensing time	bus and control-wire propagation delay; others are ignored	bus propagation delay and node's decoding time only
Performance (Efficiency)			
High loads	very high	very high	very high
Low loads	high	high	high
Fairness of Access	fair to all nodes in OE-SDAM; unfair to middle nodes in C-SDAM	unfair to right side nodes; possibility of indefinite blocking	unfair to middle nodes

Figure 9-3. Comparison of SDAM, A1, and BID

9.3 Comparison of Throughput-Delay Performances

The throughput delay curve of various protocols with 50 users are compared in Figure 9-4. It shows that, for a small propagation delay (a=0.01), all of CSMA/CD, GBRAM, and SDAM perform very closely to the M/D/1 perfect scheduling, with CSMA/CD remains to have the lowest delay in light to medium loads. However, in high loads both SDAM's and GBRAM's performances exceed that of CSMA/CD due to their collision-free properties. As a increase (a=0.1), the performance of CSMA/CD degrades significantly [Tob79], while SDAM remains close to M/D/1, and GBRAM slightly behind SDAM (refer to Figure 9-5). Under such circumstances, even BRAM performs better than CSMA/CD in medium to high loads.

When the propagation delay is extremely large (a=1.0), SDAM can still provide an acceptable delay performance. But the performance for CSMA/CD deteriorates to that of the slotted ALOHA [Met76] (see Figure 9-6). Under any circumstances, both SDAM and GBRAM out-perform BRAM by far.

From Figures 9-4, 9-5, and 9-6, we see that the SDAM protocol is much less sensitive to the propagation delay than is the GBRAM protocol. However, one must bear in mind that the GBRAM protocol allows more flexibility in network connections (e.g., radio channel and mobile users). Consequently, the maximum propagation delay for a set of network nodes may be shorter for GBRAM than for SDAM, whose

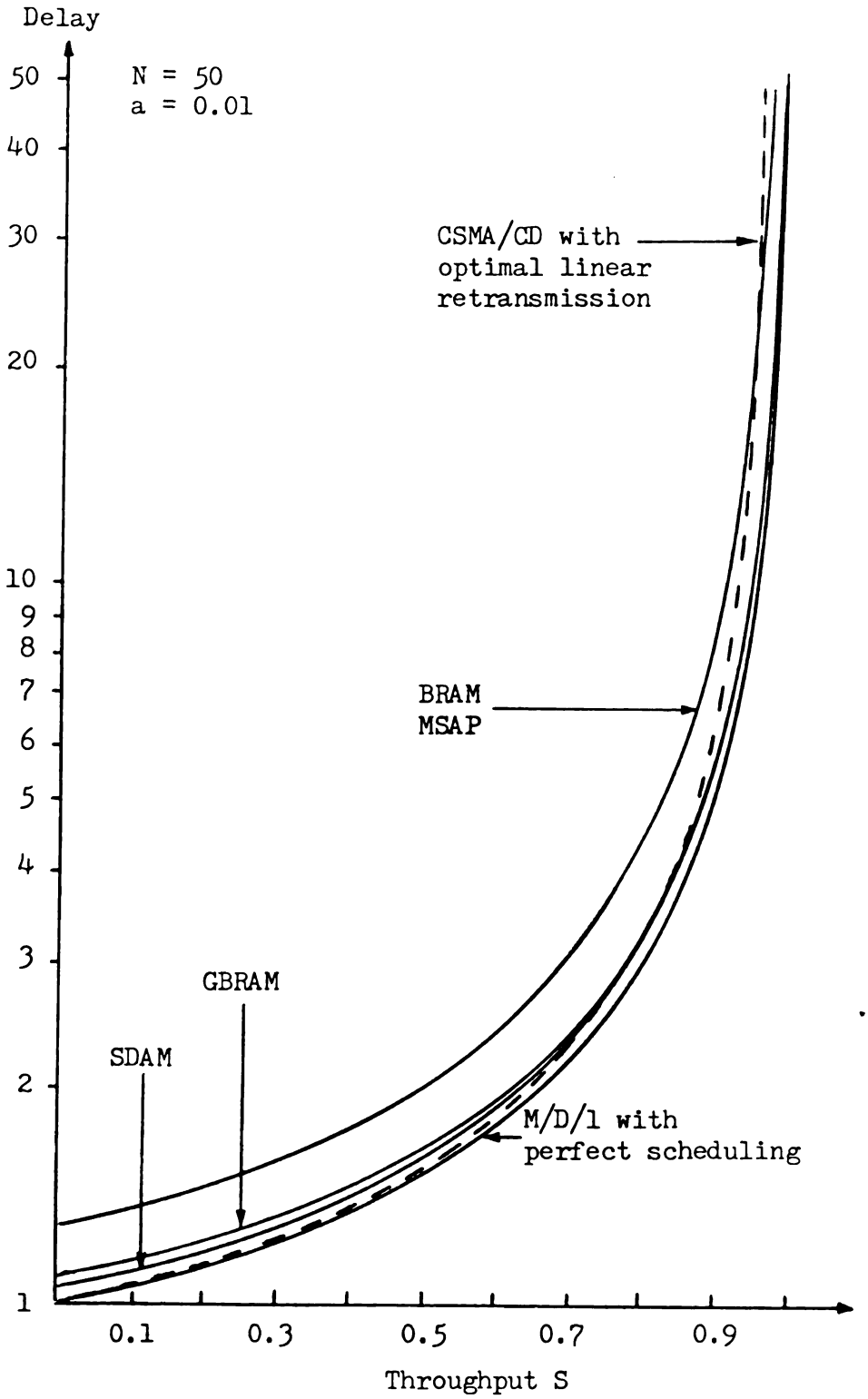


Figure 9-4. Comparison of delay performance at a=0.01

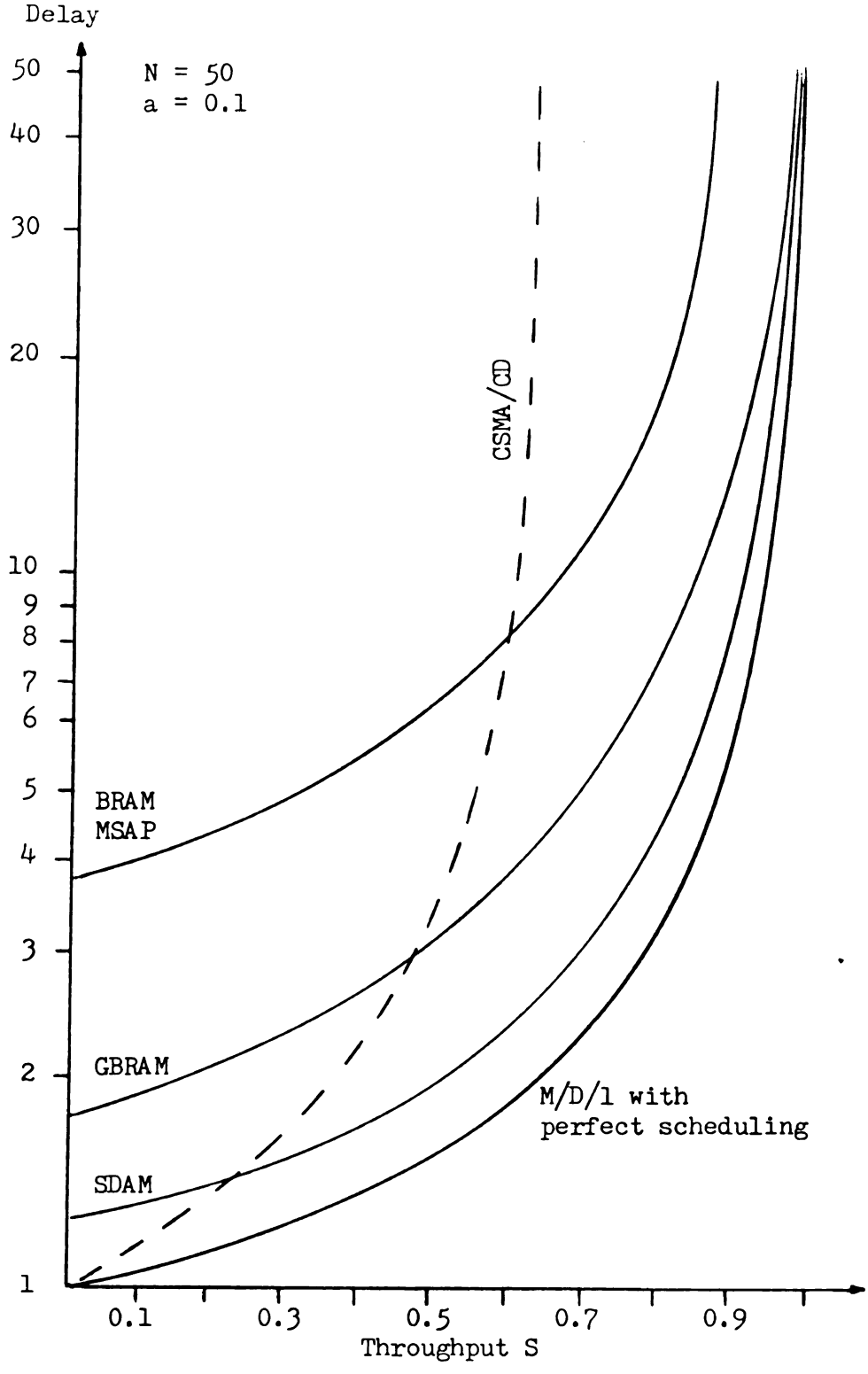


Figure 9-5. Comparison of delay performance at a=0.1

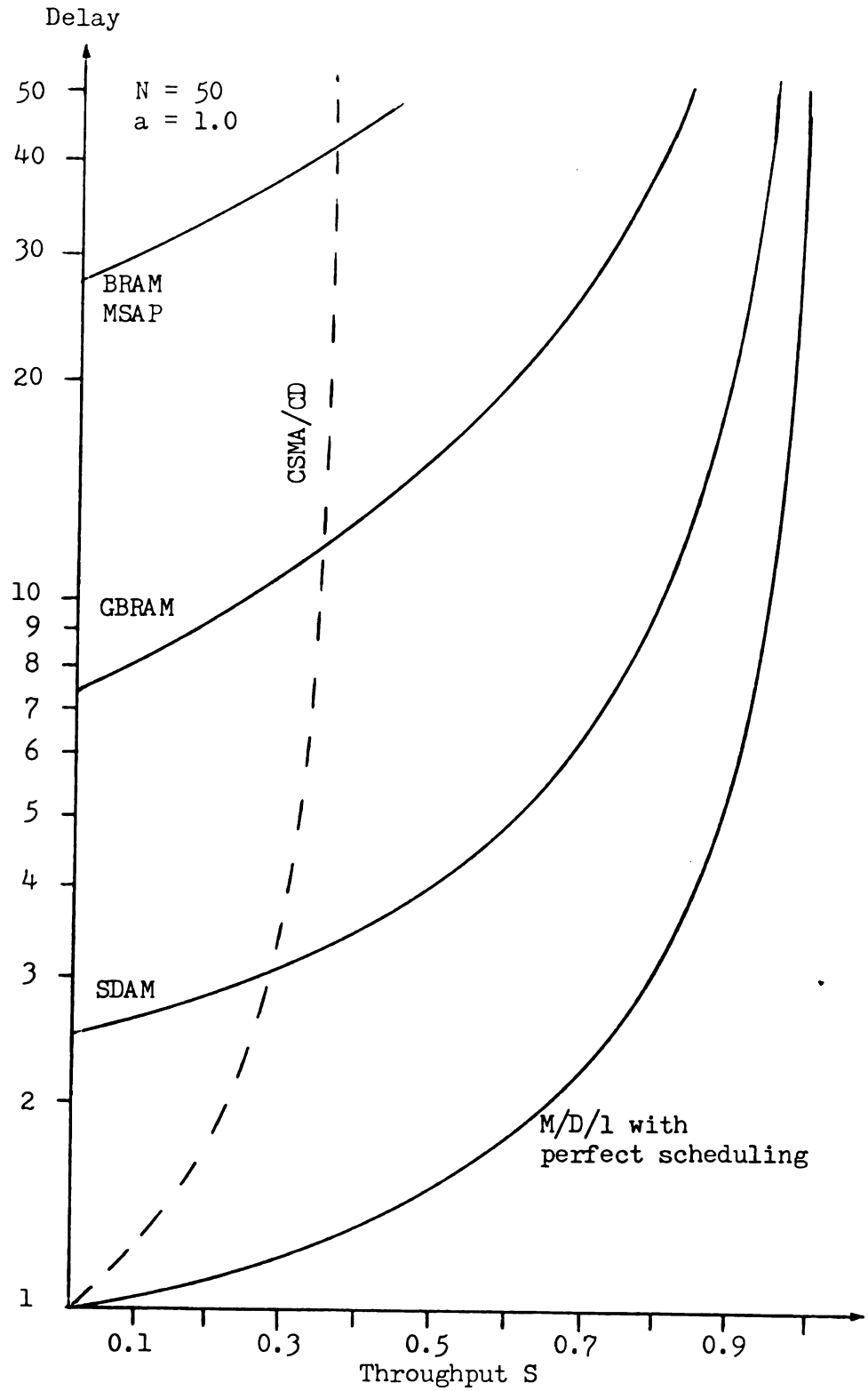


Figure 9-6. Comparison of delay performance at a=1.0

transmission paths are limited to cable bus connections.

9.4 Network Throughput and the Offered Traffic Load

The offered traffic load G of the network is defined as the total avarage number of packets available for transmission in the network. For the contention schemes, this load G is the packet arrival rate plus the rate at which packets re-enter the transmission queues because of data collisions. For the token passing schemes, since there is no data collision, we can see this G as the average number of packets in the system waiting for token to arrive and to be sent through the channel. In either case, we have $S \leq G$.

In a sense, the relationship between S and G indicates how efficiently the access scheme is removing packets from the queues and transmitting them through the network. The closer is G to S, the higher is the efficiency. This offered load G versus throughtput S is plotted in Figure 9-7. It shows that both SDAM and GBRAM are highly efficient throughout the entire spectrum of G, and remain stable (i.e., throughput does not degrade due to their collision-free properties) for high values of G.

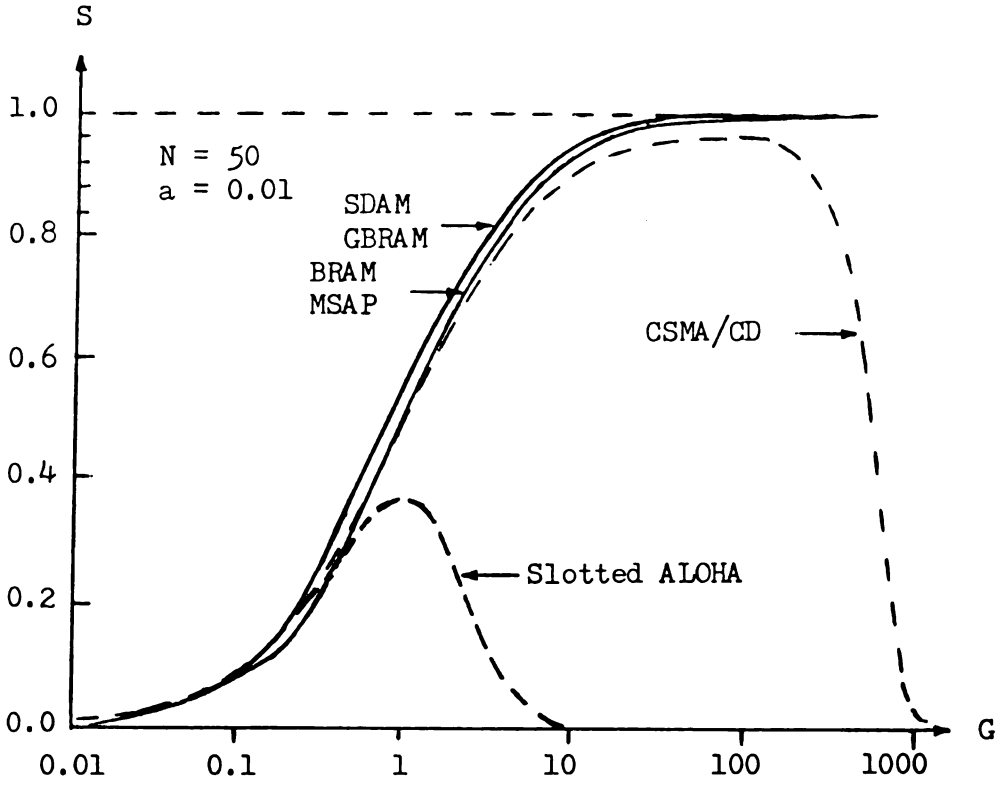


Figure 9-7. Offered load G versus throughput S

CHAPTER 10

SUMMARY, CONCLUSION, AND FUTURE RESEARCH DIRECTIONS

10.1 Summary and Conclusion

In recent years, we have witnessed a tremendous growth in the field of local area computer networking. With the advancement of new technologies (e.g., VLSI), the decrease in hardware costs, and the potential applications of local networks, it is only reasonable to expect a proliferation of local networks within the next decade.

In this paper we outlined some classifications of the many networks and protocols that have been proposed or developed. The focus was then directed to the most popular multi-access/broadcast techniques implemented on bus networks. More specifically, we have been interested in bus networks with decentralized controls for their network

performance and reliability. It was noted, however, that a serious trade-off exists between high-load efficiency and network control overhead: the random-access techniques are very efficient in light traffic, but not in high loads; the demand-assignment schemes, on the other hand, are very efficient for a small number of users under high traffic loads, but they introduce large control overheads in light loads. Additionally, both these types of protocols are sensitive to the network's propagation delay.

Two new multi-access virtual-token protocols were then proposed to solve the above problems: the shortest-delay access method (SDAM) and the group broadcast recognizing access method (GBRAM). The former minimizes the changeover time between two consecutive transmissions from different nodes; and the latter groups users into clusters, and utilizes a two-level scheduling function to reduce the control overhead. Both these protocols are decentralized and conflict-free.

The analysis in this thesis shows that SDAM and GBRAM perform closely to the M/D/1 perfect scheduling in a normal operating environment (e.g., a few hundred nodes within one mile's vicinity). The effects of several protocol overheads such as the node's turnaround time and carrier-sensing time are also assessed. The results indicate that SDAM and GBRAM are relatively insensitive to the number of users and the bus propagation delay. The non-exhaustive transmission

option of these protocols also guarantees a response time upper-bound for each node. Finally, due to the simplicity of their algorithms, SDAM and GBRAM are robust and easy to implement.

In summary, the SDAM and the GBRAM protocols do more than just combining the advantages of the token-passing schemes and the random-access techniques. Their high-efficiency, low-overhead, and conflict-free properties make them particularly suited for real-time applications and mixed voice/data traffics, either in a bursty mode or in a regular pattern of data flow. The tolerance for large user-population and large propagation delay (especially of the SDAM protocol) should also facilitate the expansion of the current local network scope into a wider range, no longer limited to just a few kilometers in distance. The GBRAM protocol can also be implemented into a radio network, which allows the users a higher mobility.

10.2 Some Topics for Future Research

So far we have examined two newly proposed virtual-token passing protocols on bus networks. The analysis presented in this paper have helped us understand the characteristics and potentials of these protocols. However, there are many more performance and implementation issues that are worth looking into. For example:

--we have been assuming Poisson arrivals of the packets for the simplicity of our analysis. In practice the arrival pattern need not be Poisson; and different traffic patterns (e.g., voice packets) could significantly affect network's performance. How well do SDAM and GBRAM handle different traffic patterns other than Poisson arrivals?

--we have seen a limited analysis of network performance under mixed-size packets. What is the effect of the variable-size packets to network's performance in general?

--pertaining to the above two issues, how do we collect and characterize the workloads imposed on a local network?

--both SDAM and GBRAM do not make provisions for handling multi-priority messages. The BID system described in Chapter 9 uses forced collisions to maintain several levels of priority. But we feel this collison policy could lead to an unnecessary waste of bandwidth and a degradation of overall delay performance. Is it possible to device a "smart" scheduling algorithm or a dynamic non-exhaustive transmission bound so as to effect a more efficient priority scheme?

--the user nodes under both protocols are numbered according to their physical locations. Is there a

good algorithm for associating a set of logical addresses to the node number, so that this node number is transparant to the user and to the inter-network messages, and that these logical addresses can be used in multi-cast references?

--both SDAM and GBRAM employ baseband signalling on a single common channel. Is it possible to implement these protocols onto a broadband network for a more efficient utilization and sharing of the bandwidth? How about on a fiber-optics network?

--we have assumed the existence of a set of network (or bus) interface units, which function as network decouplers and buffers for the user nodes. What kind of buffer management is required, and how large a buffer is needed for these interface units?

--what are the other elements that are needed for the actual implementation of either SDAM or GBRAM into an operational network?

The above list is just a few of the many topics which remain to be investigated. The answers to these questions will help us determine the feasibility of the protocols we just proposed, and ultimately become the building blocks for a future prototype network.

BIBLIOGRAPHY

BIBLIOGRAPHY

[Abr70] N. Abramson, "The Aloha System -- Another Alternative for Computer Communications," Fall Joint Computer Conference, NCC 1970, pp. 695-702.

[Abr73] N. Abramson, "The Aloha System," in N. Abramson and F.F. Kuo, eds., Computer Communication Networks, Prentice-Hall, 1973.

[Abr77] N. Abramson, "The Throughput of Packet Broadcasting Channels," IEEE Trans. Comm., Vol. COM-25, No. 1, Jan. 1977, pp. 117-128.

[Acr76] J. Acree and A. Lynch, "Ring Architecture Supports Distributed Processing," Data Communications, March/April 1976, pp. 51-55.

[Agr77] A. Agrawala et. al., "Analysis of an Ethernet-like Protocol," Proceedings of the Computer Networking Symposium, NBS, Dec. 1977.

[Agr78] A. Agrawala and R. Larsen, "Efficient Communccation for Local Computer Networks -- Coordinated Access Broadcast," U. of Maryland, Dept. of Computer Science, Technical Report TR-639, Feb. 1978.

[Ais75] H. Aiso et. al., "A Minicomputer Complex -- KOCOS ('Keio-Oki' Complex System)," Proc. of the 4th Data Communications Symposium, 1975.

[Alm79] G.T. Alms and E.D. Lazowska, "The Behavior of Ethernet-like Computer Communications Networks," TR No. 79-05-01, Dept. of Comp.Sci., U. of Washington, Apr. 1979.

[Alt78] J. Altaber, "Real-time Network for the Control of a Very Large Machine," Local Networking, NBS Special Publication 500-31, April 1978, pp. 5-6.

[And72] R.R. Anderson, J.F. Hayes, and D.N. Sherman, "Simulated Performance of a Ring-Switched Data Network," Trans. Comm., Vol. COM-20, No. 3, June 72, pp. 576-591.

[And75a] D.R. Anderson, "The EPIC-DPS: A Distributed Network Experiment," EASCON '75 Record, Sept. 1975, pp. 121A-121G.

[And75b] G.A. Anderson and E.D. Jensen, "Computer Interconnection Structures: Taxonomy, Characteristics, and Examples," ACM Computing Surveys, Vol. 7, No. 4, Dec. 1975.

[And78] E.W. Anderson and E.E. Newhall, "A Microprocessor-Based Controller For A Loop Switching System," ICC'78, Vol. 2, pp. 24.4.1-6.

[Ara79a] E. Aramis et. al., "ADCS : ARAMIS Distributed Computer System," 1979 Computer Networking Symp., pp. 139-151.

[Ara79b] E. Aramis, J.P. Cabanel, R. Dssouli, R.D. Sazbon, "ORION: A New Distributed Loop Computer Network," 1979 Computer Networking Symp., pp. 164-168.

[Ash75] R.L. Ashenhurst and R.H. Vonderohe, "A Hierarchical Network," DATAMATION, Feb. 1975, pp. 40-44.

[Bal76] J.E. Ball et. al., "RIG, Rochester's Intelligent Gateway: System Overview," IEEE Transactions on Software Engineering, Vol. SE-2, No. 4, Dec. 1976, pp. 321-328.

[Bas72] H.R. Baskin et. al., "PRIME - A Modular Architecture for Terminal-Oriented Systems," AFIPS Conference Proceedings, Vol. 40, 1972 Spring Joint Computer Conference, May 1972, pp. 431-437.

[Ber75] R.G. Berglund, "Understanding SDLC," Modern Data, Feb.-Sept. 1975.

[Ber80] H.V. Bertine, "Physical Level Protocols," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp. 433-444.

[Bib79] K.J. Biba and J.W. Yeh, "FordNet: A Front-End Approach to Local Computer Networks," Proc. LACN Symp., May 1979, pp. 199-214.

[Bil78] R.W. Bilec, D.A. Lutzky, and J.J. Peterka, "Simulating a Local Computer Network," 3rd Conference on Local Computer Networking, U. of Minn., Oct. 1978.

[Bin75a] R. Binder, "A Dynamic Packet-Switching System for Satellite Broadcast Channels," ICC'75, Vol. 3,

pp. 41-1 - 5.

[Bin75b] R. Binder, N. Abramson, F. Kuo, A. Okinaka and D. Wax, "ALOHA Packet Broadcasting - A retrospect," AFIPS Conference Proceedings, Vol. 44, 1975 National Computer Conference, pp. 203-215.

[Bli77] B.B. Bliss, W.A. Counterman, and E.A. Mackey, "Proposal for a Ring Network 'IDANET'", Proc. Conf. on A Second Look at Computer Networking, U. of Minn., Oct. 1977.

[Bor78] F. Borgonovo and L. Fratta, "SRUC: A Technique for Pakcet Transmission on Multiple Access Channels," Proc. Int'l Conf. Computer Communications, Kyoto, Japan, 1978.

[Cai78] G.D. Cain and R.C.S. Monling, "MININET: A Local Network for Real-time Instrumentation and Control," Proc. 3rd Conf. on Local Computer Networking, U. of Minn., Oct. 1978.

[Cap79] J.I. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," IEEE Trans. Inform. Theory, Vol. IT-25, Sept. 1979, pp. 505-515.

[Car77] R.J. Carpenter and R. Rosenthal, "A Local Network for the National Bureau of Standards," In Local Area Networking, Report of a Workshop held at the National Bureau of Standards, Aug. 1977, NBS Special Publication 500-31.

[Car78] R.F. Carpenter, J. Sokol, Jr. and R. Rosenthal, "A Microprocessor-based Local Netwrok Node," COMPCON 78 Fall, Sept. 1978, pp. 104-109.

[Chr73] R.D. Christman, "Development of the LASL Computer Network," COMPCON 73 Digest, Feb. 1973, pp. 239-242.

[Chl79a] I. Chlamtac, W.R. Franta, and K.D. Levin, "BRAM: The Broadcast Recognizing Access Method," IEEE Trans. on Communication, Vol. COM-27, No. 8, Aug. 1979, pp. 1183-1190.

[Chl79b] I. Chlamtac, W.R. Franta, P.C. Patton, and W. Wells, "Performance Issues in Local Computer Networks," Technical Report 79-16, Computer Science Department, U. of Minnesota, Minneapolis.

[Chl80] I. Chlamtac, "Issues in Design and Measurement of Local Area Networks," Proc. CMG-XI International

Conference on Computer Performance Evaluation, Dec. 1980, pp. 32-34.

[Chr77] G. Christensen, "Data Truck Contention in the HYPERchannel Network," Proc. A Second Look at Computer Networking, U. of Minn., Oct. 1977,.

[Cla78] D.D. Clark, K.T. Pogran, and D.P. Reed, "An Introduction to Local Area Networks," Proceedings of the IEEE, Vol. 66, No. 11, Nov. 1978, pp. 1497-1517.

[Cof72] E.G. Coffman, Jr., L.A. Klimko, and B. Ryan, "Analysis of Scanning Policies for Reducing Disk Seek Times," SIAM J. Comput., Vol. 1, No. 3, Sept. 1972.

[Cof73] E.G. Coffman, Jr. and P.J. Denning, "Operating Systems Theory," Prentice-Hall, Englewood Cliffs, N.J., 1973, Chapter 5.

[Con76] H. Connell, "A Multi-Minicomputer Network for Optical Moving Target Indication," COMPCON, fall 1976.

[Cot77] I.W. Cotton and H.C. Folts, "International Standards for Data Communications: A Status Report," Proceedings of the Fifth Data Communications Symposium, Sept. 1977.

[Cot80] I.W. Cotton, "Technologies for Loacl Area Computer Networks," Computer Networks 4, North-Holland Publishing Company, 1980, pp. 197-208.

[Cov77] G.J. Coviello, O.L. Lake and G.R. Redinbo, "SYSTEM DESIGN IMPLICATIONS OF PACKETIZED VOICE," ICC 1977, Vol. 3, pp. 38.3-49 to 53.

[Cro73] W. Crowther et. al., "A System for Broadcast Communication: Reservation Aloha," Proc. of the 6th Hawaii International Conference on System Science, Jan. 1973.

[Cro75] W.R. Crowther et. al., "Issues in Packet Switching Network Design," AFIPS NCC 75, Vol. 44, pp. 161-175.

[Cyp78] R.J. Cypser, Communications Architecture for Distributed Systems, Addison-Wesley, 1978.

[DeM76] V.A. DeMarines and L.W. Hill, "The Cable Bus in Data Communications," DATAMATION, Vol. 22, No. 8,

Aug. 1976, pp. 89-92.

[Den67] P.J. Denning, "Effects of Scheduling on File Memory Operations," AFIPS 1967 Spring Joint Computer Conference, pp. 9-21.

[DEC80] DEC, Intel, and Xerox Corp., "The Ethernet: A Local Area Network, Data Link Layer and Physical Layer Specifications," Version 1.0, Sept. 1980.

[Don74] R.A. Donnan and J.R. Kersey, "Synchronous Data Link Control: A Perspective," IBM Sys. J., 13:2, 1974.

[Don78a] J.E. Donnelley and J.W. Yeh, "Interaction Between Protocol Levels in a Prioritized CSMA Broadcast Network," 3rd Conf. on Local Networking, U. of Minn., Oct. 1978.

[Don78b] J.E. Donnelley and J.W. Yeh, "Simulation Studies of Round Robin Contention in a Prioritized CSMA Broadcast Network," 3rd Conf. on Local Networking, U. of Minn., Oct. 1978.

[Eag79] B.M. Eaglestone, "A Campus Network Based on ICL 2900 Series Protocol," Software--Practice and Experience, Vol. 9, 1979, pp. 959-967.

[Eis71] M. Eisenberg, "Two Queues with Changeover Times," Oper. Research 19, 1971, pp. 386-401.

[Eis72] M. Eisenberg, "Queues with Periodic Service and Changeover Time," Oper. Research 20, 1972, pp. 440-451.

[Esw79] K.P. Eswaran, V.G. Hamacher, and G.S. Shedler, "Asynchronous Collision-free Distributed Control for Local Bus Networks," IBM Research Report RJ2482, San Jose, Calif., 1979.

[Far69] W.D. Farmer and E.E. Newhall, "An Experimental Distributed Switching System to Handle Bursty Computer Traffic," Proceedings ACM Symposium on Problems in the Optimization of Data Communications Systems (Pine Mountain, Ga.), Oct. 1969, pp. 31-34.

[Far72a] D.J. Farber, "Netowrks: An Introduction," DATAMATION, April 1972, pp. 36-39.

[Far72b] D.J. Farber and K.C. Larson, "The System Architecture of the Distributed Computing Systems - The Communications System," Proceedings of the Symposium on Computer-Communications and

Teletraffic, New York: Polytechnic Press, 1972, pp. 21-27.

[Far73] D.J. Farber et. al., "The Distributed Computing Systems," COMPCON 73, Feb. 1973, pp. 31-34.

[Far75] D.J. Farber, "A Ring Network," DATAMATION, Feb. 1975, pp. 44-46.

[Fer75] M.J. Ferguson, "On the Control, Stability, and Waiting Time in a Slotted ALOHA Randon-Access System," IEEE Trans. Comm., Vol. COM-23, No. 11, Nov. 1975.

[Fer77a] M.J. Ferguson, "A Bound and Approximation of Delay Distribution for Fixed-Length Packets in an Unslotted ALOHA Channel and a Comparison with Time Division Multiplexing (TDM)," IEEE Trans. Comm., Vol. COM-25, No. 1, Jan. 1977, pp. 136-139.

[Fer77b] M.J. Ferguson, "An Approximate Analysis of Delay for Fixed and VAriable Length Packets in an Unslotted ALOHA Channel," IEEE Trans. Comm., Vol. COM-25, No. 7, July 1977, pp. 644-654.

[Fle73] J.G. Fletcher, "The Octopus Computer Network," DATAMATION, April 1973, pp. 58-63.

[Fle75] J.G. Fletcher, "Principles of Design in the OCTOPUS Computer Network," Proceedings ACM '75, Oct. 1975, pp. 325-328.

[Fol79] H.C. Folts, "Status Report on New Standards for DTE/DCE Interface Protocols," COMPUTER, Sept. 1979, pp. 12-19.

[Fol80] H.C. Folts, "X.25 Transaction-Oriented Features -- Datagram and Fast Select," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp. 496-500.

[For75] J.W. Forgie, "Speech Transmission in Packet-Switched Stored-and-forward Networks," AFIPS Conference Proceedings, 1975 NCC, 44, Anaheim, May 1975, pp. 137-142.

[For77a] J.W. Forgie and A.G. Nemeth, "An Efficient Packetized Voice/Data Network Using Statistical Flow Control," ICC 77, Vol. 3, pp. 38.2-44 to 48.

[For77b] P.J. Fortune, W.P. Lidinsky, and B.R. Zelle, "Design and Implementation of a Local Computer Network," ICC 77, Vol. 3, pp.46.3-221 to 226.

[Fra75a] S.C. Fralick and J.C. Garrett, "Technological Considerations for Packet Radio Networks," AFIPS-NCC75, pp. 233-243. lm 1

[Fra75b] A.G. Fraser, "A Virtual Channel Network," DATAMATION, Feb. 1975, pp. 51-56.

[Fra78] R. Frank and D. Tolmie, "The LASL Network Switch," Proc. 3rd Conf. on Local Computer Networks, U. of Minn., Oct. 1978.

[Fra79a] A.G. Fraser, "DATAKIT -- A Modular Network for Synchronous and Asynchronous Traffic," ICC '79 Conference Record, Vol. 2, June 1979, pp. 20.1.1-20.1.3.

[Fra79b] A. Franck et. al., "Some Architectural and System Implications of Local Computer Networks, " COMPCON 79 Spring, Feb. 1979, pp. 272-276D.

[Fra80] W.R. Franta and M.B. Bilodeau, "Analysis of a Prioritized CSMA Protocol Based on Staggered Delays," Acta Informatica 13, 1980, pp. 299-324.

[Fre78] H.A. Freeman, "Performance Evaluation Trends," COMPCON 78 Fall, Sept. 1978, pp. 396-398.

[Fre79] H.A. Freeman and K.J. Thurber, "Issues in Local Computer Networks," IEEE 1979 International Communications, pp. 20.3.1-20.3.5

[Fre80] H.A. Freeman, "Tutorial Notes: Introduction to Local Computer Networks," 5th Conference on Local Computer Networks, Minneapolis, Oct. 6-7, 1980.

[Fri77] I.T. Frisch, "Experiments on Random Access Packet Data Transmission on Coaxial Cable Video Transmission Systems," IEEE Trans. Comm., Vol. COM-25, No. 10, Oct. 1977, pp.1199-1203.

[Ger78a] M. Gerla and D. Mueller, "PACUIT: The Integrated Packet and Circuit Alternative to Packet Switching," IEEE 1978 COMPCON Spring, pp. 153-156.

[Ger78b] L.H. Gerhardstein et. al., "The Pacific Northwest Laboratory Minicomputer Network," Proc. Third Berkeley Workshop on Distributed Data Management and Computer Networks, Aug. 1978, pp. 144-158.

[Gfe78] F.R. Gfeller, H.R. Muller, and P. Vettiger, "Infrared Communication for In-house Applications," Proc. COMPCON Fall'78, IEEE Computer Society,

Sept. 1978.

[Git77] I. Gitman et. al., "Issues in Integrated Network Design," ICC'77, Vol. 3, pp. 38.1-36 to 43.

[Gor80] R.L. Gordon, W.W. Farr, and P. Levine, "Ringnet: A Packet Switched Local Network with Decentralized Control," Computer Networks 3 (1980), North-Holland, pp. 373-379.

[Gra72] J.P. Gray, "Line Control Procedures," Proc. of the IEEE, Vol. 60, No. 11, Nov. 1972.

[Gra75] J.P. Gray and C.R. Blair, "IBN'S Systems Network Architecture," DATAMATION, April 1975, pp.51-56.

[Gre77] M. Green, "A DoD Local Network: A Structured Implementation," Proc. A Second Look at Local Computer Networking, U. of Minn., Oct. 1977.

[Gru74] D.S. Grubb and I.W. Cotton, "Criteria for the Performance Evaluation of Data Communications Services for Computer Networks," National Bureau of Standards, Technical Note 882, Dec. 1974.

[Haf74] E.R. Hafner, Z. Nenadal, and M. Tschanz, "A Digital Loop Communication System," IEEE Trans. Comm., June 1974, pp. 871-881.

[Ham80] V.C. Hamacher and G.S. Shedler, "Performance of a Collision-free Local Bus Network Having Asynchronous Distributed Control," Computer Architecture 1980, Vol. 8, No. 3, pp. 80-87.

[Han79] L.W. Hansen and M.S. Schwartz, "An Assigned-Slot Listen-Before-Transmission Protocol for a Multiaccess Data Channel," IEEE Trans. Comm., Vol. COM-27, No. 6, June 1979, pp. 846-856.

[Hay71] J.F. Hayes and D.N. Sherman, "Traffic and Delay in a Circular Data Network," Second Symposium on Problems in the Optimization of Data Commnnications Systems, Oct. 1971, pp. 102-107.

[Hay78] J.F. Hayes, "An Adaptive Technique for Local Distribution," IEEE Trans. on Communication, Vol. COM-26, Aug. 1978.

[Hea70] F.E. Heart et. al., "The interface message processor for the ARPA computer network," AFIPS Spring Joint Computer Conf., 1970, pp. 551-567.

[Hin77] O.R. Hinton et. al., "Distributed Industrial Control Using a Network of Microcomputers," COMPCON 77 Fall, Sept. 1977, pp. 316-319.

[Hop78] A. Hopper, "Data Ring at Computer Laboratory, University of Cambridge," Local Area Networking, NBS Special Publication 500-31, April 1978, pp. 11-17.

[Hsi80] P. Hsi and T. Lissack, "Local networks' consensus: High speed," Data Communications, Dec. 1980, pp. 56-66.

[Hue77] W. Huen et. al., "A Network Computer for Distributed Processing," COMPCON 77 Fall, Sept. 1977, pp. 326-330.

[IEE81] "Token Structure -- An Architecture," Token Working Group, IEEE Project 802, DLMAC Subcommittee, draft revision 10, July 10, 1981.

[IFI78] "A LOCAL COMPUTER NETWORK BIBLIOGRAPHY," IFIP WG 6.4 Repository, The IFIP working group (WG 6.4) on local computer netwrks, 1978.

[Inn75] D.R. Innes and J.L. Althy, "An Intra University Network," Fourth Data Communications Symposium, Oct. 1975, pp. 1-8 to 1-13.

[Jac69] P.E. Jackson and C.D. Stubbs, "A study of multiaccess computer communications," AFIPS Spring Joint Computer Conf., 1969, pp. 491-504.

[Jac77] I. Jacob et. al., "CPODA -- A Demand Assignment Protocol for SATNET," Proc. of the 5th Data Communications Symposium, Sept. 1977.

[Jen75] E.D. Jensen, "A Distributed Function Computer for Real-Time Control," Proc. of the 2nd Annual Symposium on Computer Architecture, Jan. 1975.

[Jen78] E.D. Jensen, "The Honeywell Experimental Distributed Processor - An Overview," Computer, Vol. 11, No. 1, January 1978, pp. 28-38.

[Kah75] R.E. Kahn, "The Organization of Computer Resources into a Packet Radio Network," AFIPS Conf. Proc., Vol. 44, May 1975, pp. 177-185. Reprinted in IEEE Trans. on Communications, Vol. COM-25:1, Jan. 1977, pp. 169-178.

[Kah78] R.E. Kahn et. al., "Advances in Packet Radio

Technology," Proc. IEEE, Vol. 66, Nov. 1978.

[Kai79] R.Y. Kain, W.R. Franta, and G.D. Jelatis, "CHIMPNET: a Network Testbed," Computer Networks 3, North-Holland, pp. 447-457.

[Kim75] S.R. Kimbleton and G.M. Schneider, "Computer Communication Networks: Approaches, Objectives, and Performance Considerations," ACM Computing Surveys, Vol. 7, No. 3, Sept. 1975, pp. 90-134.

[Kle74] L. Kleinrock and W.E. Naylor, "On measured behavior of the ARPA network," AFIPS NCC 74, Vol. 43, pp. 767-780.

[Kle75a] L. Kleinrock and S.S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," IEEE Trans. Comm., Vol. COM-23, No. 4, April 1975, pp. 410-423.

[Kle75b] L. Kleinrock and F. Tobagi, "Random access techniques for data transmission over packet-switched radio channels," AFIPS NCC 75, pp. 187-201.

[Kle75c] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I -- Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," IEEE Trans. on Communications, Vol. COM-23, No. 12, Dec. 1975.

[Kle76a] L. Kleinrock, Queuing Systems, Vol. I: Theory, and Vol. II: Computer Applications, Wiley-Interscience, NY, 1976.

[Kle76b] L. Kleinrock, "On Communications and Networks," IEEE Trans. on Computers, Vol. C-25, No. 12, Dec. 1976.

[Kle77a] L. Kleinrock and M. Scholl, "Packet Switching in Radio Channels: New Conflic-free Multiple Access Schemes for A Small Number of Data Users," Proc. ICC., June 1977, pp. 22.1-105 to 22.1-111.

[Kle77b] L. Kleinrock and Y. Yemini, "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication," ICC Conf. Proc., Chicago, Il., June 1977.

[Kon74] A.G. Konheim and B. Meister, "Waiting Lines and Times in A System with Polling," JACM, Vol. 21, no. 3, July 1974, pp. 470-490.

[Kon76] A.G. Konheim, "Chaining a Loop System," IEEE Trans. Comm., Vol. COM-24, No. 2, Feb. 1976, pp.203-210.

[Kuh79] R.C. Kuhns and M.C. Shoquist, "A Serial Data Bus System for Local Processing Networks," COMPCON 79 Spring Digest of Papers, Feb. 1979, pp. 266-271.

[Kun78] R.C. Kunzelman, "OVERVIEW OF THE ARPA PACKET RADIO EXPERIMENTAL NETWORK," IEEE 1978 COMPCON Spring, pp. 157-160.

[Lam75] S.S. Lam and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," IEEE Trans. Comm., Vol. COM-23, No. 9, Sept. 1975, pp.891-904.

[Lam77] S. Lam, "Delay Analysis of Time-Division Multiple Access (TDMA) Channel," IEEE Trans. on Comm., Vol. COM-25, Dec. 1977.

[Lam78] S.S. Lam, "A New Measure for Characterizing Data Traffic," IEEE Trans. Comm., Vol. COM-26, No. 1, Jan. 1978, pp. 137-140.

[Lam80] S.S. Lam, "A Carrier Sense Multiple Access Protocol for Local Networks," Computer Networks 4 (1980), North-Holland, pp. 21-32.

[Lee78] C.C. Lee and A.V. Pohm, "Interface Processor for High Speed Recirculating Data Netwrok," COMPCON 78 Fall, Sept. 1978, pp. 194-200.

[Len77] L. Lenzini and G. Sommi, "RPCNET, A NETWORK AMONG EDUCATION AND RESEARCH ORGANIZATIONS IN ITALY: CHARACTERISTICS AND STATUS," EUROCON'77, Vol. 2, pp. 3.1.3-1 to 12.

[Li 81] L. Li and H.D. Hughes, "Definition and Analysis of a New Protocol," Proc. 6th Conference on Local Computer Networks, Oct. 1981, pp. 21-29.

[Li 82] L. Li, H.D. Hughes, and L.H. Greenberg, "Performance Analysis of a Shortest-delay Protocol," Proc. 6th Berkeley Workshop on Distributed Data Management and Computer Networks, Pacific Grove, Calif., Feb. 16-19, 1982.

[Lid76] W.P. Lidinsky, "The Argonne Intra-Laboratory Network," Proc. Berkeley Workshop on Distributed Data Management and Computer Networks, May 1976, pp. 263-275.

[Lin78] K. Lin, "Design of a Packet-Switched Micro-Subnetwork," COMPCON 78 Fall, Sept. 1978, pp. 184-193.

[Liu78] M.T. Liu, "Distributed Loop Computer Networks," Advances in Computers, Vol. 17, (M. Rubinoff and M.C. Yovits, eds.) Academic Press, New York, pp. 163-221, 1978.

[Liu81a] T.T. Liu, L. Li, and W.R. Franta, "The Analysis of a Conflict-free Protocol Based on Node Clusters," Proc. 6th Conference on Local Computer Networks, Oct. 1981, Minneapolis.

[Liu81b] T.T. Liu, L. Li, and W.R. Franta, "A Decentralized Conflict-free Protocol, GBRAM, for Large-scale Local Networks," Proc. Computer Networking Symp., Dec. 1981, pp. 39-54.

[Lov79] R.A. Loveland, "PUTTING DECNET INTO PERSPECTIVE," DATAMATION, March 1979, pp. 109-114.

[Luc78] E.C. Luczak, "Global Bus Computer Communications Techniques," Proceedings, Computer Networking Symposium, National Bureau of Standards, Dec. 1978, pp. 58-71.

[Man76] W.F. Mann et. al., "A Network-Oriented Multiprocessor Front-End Handling Many Hosts and Hundreds of Terminals," AFIPS Conference Proceedings, Vol. 45, 1976 NCC, pp. 533-540.

[Man77] E.G. Manning and R.W. Peebles, "A Homogeneous Network for Data Sharing - Communications," Computer Networks, Vol. 1, No. 4, May 1977, pp. 211-224.

[Mar78] J.W. Mark, "Global Scheduling Approach to Conflict-free Multiaccess via a Data Bus," IEEE Trans. on Comm., Vol. COM-26, Sept. 1978.

[Mar81] J. Martin, Computer Networks and Distributed Processing: Software, Techniques, and Architecture, Prentice-Hall, 1981.

[McQ72] J.M. McQuillan and D.C. Walden, "The ARPA Network Design Decisions," Computer Networks, 1:5, Aug. 1972, pp. 243-289.

[McQ78] J.M. McQuillan, Understanding the New Local Network Technologies, BBN Report 3927, Sept. 1978.

[McQ79a] J.C. McQuillan, "Local Network Architectures," Computer Design, May 1979, pp. 18-26.

[McQ79b] J.M. McQuillan, "Local Network Technology and the Lessons of History," Proc. LACN Symposium, May 1979, pp. 191-197.

[Meh79] S.K. Mehra and J.C. Majithia, "A MODIFIED ETHERNET FOR MULTIPROCESSOR INTERCOMMUNICATIONS," 1979 Computer Networking Symposium, pp. 132-138.

[Mei77a] N. Meisner and D. Willard, "Dual-Mode Slotted TDMA Digital Bus," Proc. of the 5th Data Communications Conference, Sept. 1977.

[Mei77b] N.B. Meisner, D.G. Willard and G.T. Hopkins, "Time Division Digital Bus Techniques Implemented on Coaxial Cable," Proceedings of Computer Networking Symposium, NBS Dec. 1977, pp. 112-117.

[Met76] R.M. Metcalfe and D.R. Boggs, "ETHERNET: Distributed Packet Switching for Local Computer Networks," CACM, Vol. 19, No. 7, July 1976, pp. 395-404.

[Mil76] D.L. Mills, "An Overview of the Distributed Computer Networks," AFIPS Conference Proceedings, Vol. 45, 1976 National Computer Conference, pp. 523-531.

[Mil78] W. Miller, "INTERCONNECTING LOCAL NETWORKS," 3rd Conf. on Local Computer Networking, U. of Minn., Oct. 1978.

[Moc77] P.V. Mockapetris, M.R. Lyle and D.J. Farber, "On the Design of Local Network Inerfaces," IFIP Congress, Aug. 1977, pp. 427-430.

[Mow79] O.A. Mowafi and W.J. Kelly, "INTEGRATED VOICE/DATA PACKET SWITCHING TECHNIQUES FOR FUTURE MILITARY NETWORKS," 1979 Computer Networking Symposium, pp. 216-223.

[NBS78] "Local Area Networking," Report of a Workshop Held at the National Bureau of Standards, August 22-23, 1977. Ira W. Cotton, editor, U.S. Dept. of Commerce, National Bureau of Standards, April 1978.

[Nel78] D.L. Nelson and R.L. Gordon, "Computer Cells - A Network Architecture for Data Flow Computing," COMPCON 78 FALL, Sept. 1978, pp. 296-301.

[Par79] R. Pardo and M.T. Liu, "MULTI-DESTINATION PROTOCOLS FOR DISTRIBUTED SYSTEMS," 1979 Computer Networking Symposium, pp. 176-185.

[Pau78] W.R. Paulsen, M. Maranhao, and D.E. Thomas, "The Analysis of the Performance of Multi-Processor Architectures for SENET," 1978 Computer Networking Symposium, pp. 88-94.

[Pie72] J.R. Pierce, "How Far Can Data Loops Go?", IEEE Trans. on Communications, Vol. COM-20, No. 3, June 1972, pp. 527-530.

[Pog78] K.T. Pogran, and D.P. Reed, "The MIT Laboratory for Computer Science Network," Local Area Networking, NBS Special Publication 500-31, April 1978, pp. 20-22.

[Pop79] R. Popsescu-Zeletin, D. Baum and B. Butscher, "A LOCAL COMPUTER NETWORK IN A RESEARCH ENVIRONMENT: THE HMINET," 1979 Computer Networking Symposium, pp. 158-163.

[Raw78] E.G. Rawson and R.M. Metcalfe, "Fibernet: Multimode Optical Fibers for Local Computer Networks," IEEE Trans. on Communications, July 1978, pp. 983-990.

[Rea75] C.C. Reames and M.T. Liu, "A Loop Network for Simultaneous Transmission of Variable-Length Messages," Second Annual Symposium on Computer Architecture, Jan. 1975, pp. 7-12.

[Rea76] C.C. Reames and M.T. Liu, "Design and Simulation of the Distributed Loop Computer Network .(DLCN)," Third Annual Symposium on Computer Architecture, Jan. 1976, pp. 124-129.

[Ric78] G. Ricart and A. Agrawala, "Dynamic Management of Packet Radio Slots," presented at 3rd Berkeley Workshop on Distributed Data Management and Computer Networks, Aug. 1978.

[Rob73] L. Roberts, "Dynamic Allocation of Satellite Capacity Through Packet Reservation," AFIPS Conference Proceedings, Vol. 42, June 1973.

[Rob75] L.G. Roberts, "ALOHA Packet System With and Without Slots and Capture," Comput. Comm. Review, Vol. 5, pp. 28-42, April 1975.

[Rod77] J. Rodgers, "Computer Networking with a Data Bus," Proc. of the 16th Annual Technical Symposium on

Systems and Software, NBS, June 1977.

[Ros73] S. Rosen and J.M. Steel, "A Local Computer Network," COMPCON 73 Digest, Feb. 1973, pp. 129-132.

[Rub77] I. Rubbin, "A Group Random-Access Procedure for Multi-Access Communication Channels," NTC'77 Conf. Record, Los Angeles, Dec. 1977, pp. 12:5-1 to 7.

[Rub79] I. Rubin, "Message Delays in FDMA and TDMA Communication Channels," IEEE Trans. on Comm., Vol. COM-27, No. 5, May 1979, pp. 769-777.

[Ryb80] A. Rybczynski, "X.25 Interface and End-to-End Virtual Circuit Service Characteristics," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp. 500-510.

[Sak77] T. Sakai et. al., "Inhouse Computer Network KUIPNET," Information Processing 77, B. Gilchrist, Editor, North-Holland Publishing Co., 1977, pp. 161-166.

[Sca74] R.A. Scantlebury and P.T. Wilinson, "THE NATIONAL PHYSICAL LABORATORY DATA COMMUNCIATION NETWORK," ICCC'74, No. 2, pp. 223-228.

[Sch73] J.W. Schwartz and M. Muntner, "Multiple-Access Communications for Computer Networks," in N. Abramson and F. Kuo, eds., Computer Communication Networks, Prentice-Hall, 1973.

[Sch77] J.W. Schwartz, "Computer-Communication Networks, Prentice-Hall, 1977.

[Sha78] R.R. Shatzer, "Distributed Systems/1000," Hewlett-Packard Journal, March 1978, pp. 15-20.

[She78] R.H. Sherman et. al., "Current Summary of Ford Activities in Local Networking," Local Area Networking, NBS Special Publication 500-13, April 1978, pp. 22-23.

[Sch78] M. Scholl and L. Kleinrock, "On a Mixed Mode Multiple Access Schemes for Packet-Switched Radio Channels," IEEE Trans. Comm., Vol. COM-27, No. 6, June 1979, pp. 906-911.

[Sho79] J.F. Shoch, "Design and Performance of Local Computer Networks," Ph.D Dissertation, Dept. of

Computer Science, Stanford University, 1979.

[Sho80a] J.F. Shoch and J.A. Hupp, "Performance of an Ethernet Local Network -- A Preliminary Report," COMPCON Spring 1980, pp. 318-322.

[Sho80b] J.F. Shoch and A.J. Hupp, "Measured Performance of an Ethernet Local Network," CACM, Vol. 23, No. 2, Dec. 1980, pp. 711-721.

[Spa79] O. Spaniol, "Modelling of Local Computer Networks," Computer Networks 3 (1979), North-Holland, pp. 315-326.

[Spe77] Sperry Univac, "AN/USQ-76 Converter - Switching System, Signal Data," Sperry Univac Product Description Bulletin, June 1977.

[Spr71] J.D. Spragins, "Analysis of Loop Transmission Systems," Second Symposium on Problems in the Optimization of Data Communications Systems, Oct. 1971, pp. 175-182.

[Spr78] J.F. Springer, "The Distributed Data Network, Its Architecture and Operation," COMPCON 78 Fall, Sept. 1978, pp. 221-228.

[Swa77] R.J. Swan et. al., "Cm* -- A modular, multi-microprocessor," AFIPS Conference Proceedings, Vol. 46, 1977 NCC, pp. 637-644.

[Szu78] E. Szurkowski, "A High Bandwidth Local Computer Network," COMPCON 78 Fall Sept. 1978, pp. 98-103.

[Tho75] J.E. Thorton et. al., "A new Approach to Network Storage Management," Computer Design, Vol. 14, No. 11, Nov. 1975, pp. 81-85.

[Tho79] J.E. Thornton, "OVERVIEW OF HYPERchannel," COMPCON'79 Spring, Feb. 1979, pp. 262-265.

[Thu72] K. Thurber et. al., "A Systematic Approach to the Design of Digital Bussing Structures," Proc. of the AFIPS FJCC, 1972.

[Thu79a] K.J. Thurber and H.A. Freeman, "Local Computer Network Architectures," COMPCON 79 Spring, Feb. 1979, pp. 258-261.

[Thu79b] K.J. Thurber and H.A. Freeman, "A Bibliography of Local Computer Network Architectures," Computer Architecture News, Vol. 7, No. 5, Feb. 1979,

pp. 22-27 and Computer Communication Review, Vol. 9, No. 2, April 1979, pp. 1-6.

[Thu79c] K.J. Thruber and H.A. Freeman, "Architecture Considerations for Local Computer Networks," Proc. 1st Int'l Conf. on Distributed Computing Systems, Oct. 1979, pp. 131-142.

[Tob74] F.A. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," Ph.D Dissertation, Computer Science Dept., U. of California, Los Angeles, Rep. UCLA-ENG 7499, Dec. 1974.

[Tob75] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II -- The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," IEEE Trans. on Communications, Vol. COM-23, No. 12, Dec. 1975.

[Tob76] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part III -- Polling and (Dynamic) Split-Channel Reservation Multiple Access," IEEE Trans. on Communications, Vol. COM-24, No. 12, Dec. 1976.

[Tob77] F.A. Tobagi and L. Kleinrock, "Packet Switching in Radio Chanels: Part IV -- Stability Considerations and Dynamic Control in Carrier Sense Multiple Access," IEEE Trans. Comm., Vol. COM-25, No. 10, Oct. 1977, pp.1103-1119.

[Tob78] F.A. Tobagi and L. Kleinrock, "The Effect of Acknowledgment Traffic on the Capacity of Packet-Switched Radio Channels," IEEE Trans. Comm., Vol. COM-26, No. 6, June 1978, pp. 815-826.

[Tob79] F.A. Tobagi and V.B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," Proc. LACN Symposium, May 1979, pp. 217-244.

[Tob80] F.A. Tobagi, "Multiaccess Protocols in Packet Communication Systems," IEEE Trans. on Communications, Vol. COM-28, No. 4, April 1980, pp. 468-488.

[Tok77] M. Tokoro and K. Tamaru, "Acknowledging Ethernet," COMPCON 77 Fall, Sept. 1977, pp. 320-325.

[Tro80] C. Tropper, "Models of Local Computer Networks," Mitre Corp. Report ESD-TR-80-111.

[Twe79] J.W. Tweedy, "COMPARATIVE ANALYSIS OF DISTRIBUTED SYSTEM DESIGN," 1979 Computer Networking Symposium, pp. 117-131.

[Tym71] L. Tymes, "TYMNET -- A Terminal-Oriented Communication Network," AFIPS Spring Joint Computer Conf., 38, 1971, pp. 211-216.

[Ulu81] M.E. Ulug, G.M. White and W.J. Adams, "Bidirectional Token Flow System," Proc. 7th Data Communication Symposium, Mexico City, Mexico, Oct. 1981, pp. 149-155.

[Wan78] J.F. Wanner, "WIDEBAND COMMUNICATION SYSTEM IMPROVES RESPONSE TIME," Computer Design, Dec. 1978, pp. 85-92.

[Wec76] S. Wecker, "DECNET: A BUILDING BLOCK APPROACH TO NETWORK DESIGN," NTC'76, pp. 7.5-1 - 4.

[Wec79] S. Wecker, "Computer Network Architecture," COMPUTER, Sept. 1979, pp. 58-72.

[Wes72] L.P. West, "Loop Transmission Control Structures," IEEE Trans. on Communications, Vol. COM-20, No. 2, June 1972, pp. 531-539.

[Wid76] L. Widdoes, "The Minerva Multi-Microprocessor," Proc. of the 3rd Annual Symposium on Computer Architecture, 1976.

[Wil73] D. Willard, "MITRIX: A Sophisticated Digital Cable Communications System," Proc. of the National Telecommunications Conference, Nov. 1973.

[Wil74] D.G. Willard, "A Time Division Multiple Access System for Digital Communication," Computer Design, June 1974.

[Wil75] M.V. Wilkes, "Communication Using a Digital Ring," Proceedings of the Pacific Area Computer Communication Network System Symposium, Aug. 1975, pp. 47-56.

[Wit78] L.D. Wittie, "MICRONET: A Reconfigurable Microcomputer Network for Distributed Systems Research," Simulation, Nov. 1978, pp. 145-153. Vol. 9, No. 2, April 1979, pp. 1-6.

[Wol78] J.J. Wolf and M.T. Liu, "A Distributed Double-Loop Computer Network (DDLCN)," Proceedings of the Seventh Texas Conference on Computing Systems,

Oct. 1978, pp. 6-19 to 6-34.

[Woo79] L.D. Wood et. al., "A Cable-Bus Protocol Architecture," Proceedings DATACOM '79, Nov. 1979.

[Wul75] W. Wulf and R. Levin, "A Local Network," DATAMATION, Feb. 1975, pp. 47-50.

[Yaj77] S. Yajima, et. al., "Labolink: An Optically Linked Laboratory Computer Network," Computer, Nov. 1977, pp. 52-59.

[Zim80] H. Zimmermann, "OSI Reference Model -- The ISO Model of Architecture for Open Systems Interconnection," IEEE Trans. Comm., Vol. COM-28, No. 4, April 1980, pp.425-432.