A Collision Free Protocol for LANs Utilizing Concurrency for Channel Contention and Transmission

Eric Chi-Wah Wong

A Thesis

in

The Department

of

Electrical Engineering

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

March 1985

ABSTRACT

A Collision Free Protocol for LANs Utilizing Concurrency for Channel Contention and Transmission

Eric Chi-Wah Wong

In the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Scheme, the ready station avoids collisions by listening to the carrier due to another user's transmission. However, collisions still happen due to the factor of propagation delay. These collisions introduce a limitation on system throughput even with the introduction of backoff algorithms.

In this work, a new protocol is designed to eliminate collisions by introducing the ideas of parallelism and prescheduling in order to enhance the system utilization. In this new protocol, it is proposed that the channel contention period coincides with the previous packet transmission.

In this thesis, we provide the new collision free protocol and prove that it is correct and collision free. We also analyse this protocol and obtain the average utilization factor and packet delay for a network operating under such a protocol. Finally, we establish the validity of our theoretical results through simulation.

ACKNOWLEDGEMENTS

Many thanks to my supervisor, Professor N. Dimopoulos, for his patience and guidance to lead me to complete this thesis work.

I would like to show my gratitude to Miss Kimmy Chung for her help in finishing the drawings and tidying up the programs. I would also like to thank Eddie Wingrowiz, the Nguyen brothers and Kin-fun Li for their help in the computer plots when I was in Edmonton. Thanks to Madeleine Klein for her excellent and efficient typing and lastly to all my friends at Concordia University for their pressure relieving and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Multiprocessors and Local Area Networks

In recent years, the trend of research has been the quest for higher speed, greater computation power, better performance and more reliable computers at a lower cost. This may be possible by aggregating several low cost microprocessors for a single application. Several architectures have been implemented or proposed. Principally these architectures are either tightly coupled multiprocessor systems with an MIMD or SIMD structure, or loosely coupled computer networks. The following is a short introduction to multiprocessor systems and computer networks.

1.1.1 Multiprocessor Systems

The principal characteristics of a multiprocessor system is the ability of each processor to share memory with some or all the other processors in the configuration. The sharing capability is provided through an interconnection network between the processors and the memory modules, which logically looks like Fig. 1.1. The function of the switch is to provide a logical link between any processor and any memory module.

There are four design decisions in selecting the architecture of an interconnection network. First, it is the operation mode of the network communication which can be classified into three categories : synchronous, asynchronous or a combination of the two. Synchronous communication is the one where the information paths are established synchronously while asynchronous communication is required for multipro-

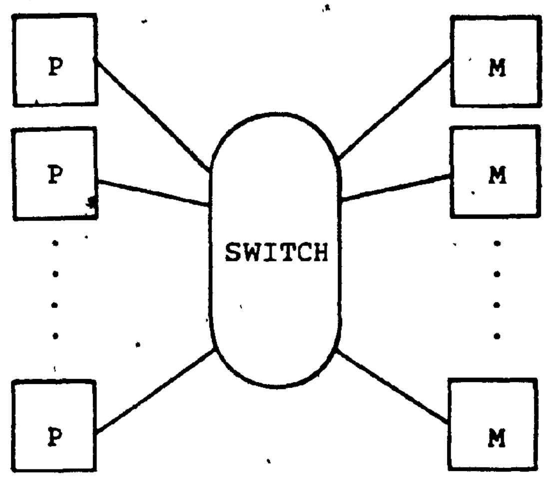


Fig. 1.1 Logical Organization of a Multiprocessor

cessing in which connection requested are issued dynamically. Second, the control strategy, which can be either distributed or centralized. This is to control the setting of the switching elements and interconnection links by the individual switching element or a centralized controller. The third item is the switching methodology which consists of two major switching strategies : circuit switching and packet switching. In circuit switching, a communication path is physically established from a source to a destination (e.g. procerssor to a memory). On the other hand, in packet switching, there is no direct electrical path between source and destination. Instead, the data is put in a packet and routed through the interconnection network. Generally, circuit switching is suitable for large data transmissions while packet switching is more efficient for short data messages. Eventually, the network topology is a key factor in deciding the system architecture. Network topology is characterised by the pattern of the interconnection among the nodes (i.e. switching points and edges representing the communication links) of the network. The differences in the interconnections may result in differences in the system architecture.

In most of the multiprocessor systems, there exist delays in the average memory access time due to contention. Some suggested solutions are to introduce cache memory or local memory for each processor. Note that in the centralized switching multiprocessor system, the number of interconnections is determined when the switch is designed. However, more extensibility and better reliability can be provided in the distributed switching multiprocessors. In such a scheme, a processor can access its local storage directly, but access to non-local storage must be routed through its switch.

A. Tightly Coupled System

A tightly coupled multiprocessor system is a system where processors have access to a common memory. This common memory provides a very fast data transfer medium as well as making it possible to share a common code.

The SIMD multiprocessor model (e.g. array processors), consists of several processors and parallel memory modules which operate synchronously under one control unit; such a model handles single instruction streams operating on multiple data streams. Examples are the Illiac IV [6] and Staran [4]. Another type of multiprocessors can handle multiple instructions and multiple data stream and hence is called a MIND multiprocessor system. Existing systems with such an architecture are HEP [26], C.mmp [33], Cm^* [27,28], data flow processor [8], and flow model processor [28].

B. Loosely Coupled System

Unlike the tightly coupled systems, the loosely coupled systems are those where processors communicate with nonlocal memories via input-output circuits. Thus the transfer of information requires input and output operations. The communication media are slower and may require a number of intermediate processes. However, the interconnection medium is more flexible, and generally the physical distance is greater than that of the tightly coupled systems. The interconnection medium coupling the system could be either parallel or serial.

The Local Area Network is one kind of loosely coupled system extended in a limited geographical area. Local Area Networks will be discussed in the next section.

1.1.2 Local Area Networks

A Local Area Network (LAN) is a communication network that provides for the interconnection of a variety of data communicating devices within a small area. A distinction between LANs and multiprocessor system is the degree of coupling (refer Section 1.1). Besides, most LANs use packet switching in communication.

The nature of a Local Area Network falls somewhere between a multiprocessor system and a long-haul data network. The classification of these multiple processor systems is governed by their physical size. It is given in Table 1.1 [29].

TABLE 1

INTER NODE DISTANCE		
0.1 m	Circuit Board	Data flow machine
1 m	System	Multiprocessor
10 m	Room	
100 m	Building	Local network
1 km	Campus	
10 km	City	Long haul network
100 km	Country	
1000km	Continent	Interconnection of
10,000 km	Planet	long haul networks

The key technical characteristics of a local area network depends on the transmission medium, topology and the channel access protocol.

Presently, most LANs employ bit-serial transmission over either twisted pair or coaxial cable (see Figure 1.2). Due to the simplicity and low interfacing cost, many LANs use baseband signaling. On the other hand, several commercial LANs use CATV broadband cable as the transmission medium because of its broad bandwidth that can support simultaneous transmission of data, voice and video [7]. In the future, one promising candidate for the local-area transmission medium is the optical fibre which can achieve a very high transmission rates in the order of Gigabits/sec (10^9 bps).

Network topology (as was defined in Section 1.1.1) is again characterised by the pattern of the interconnection among the various nodes (a controller is used to access the network) of the network. There are three popular topologies that are used in LANs (see Fig. 1.3). The star [18] topology is the most commonly used network architecture which has a central node for the control of its operation. The ring (or loop) [3,23] topology connects its nodes in a closed network and circulates messages in one direction. These messages are amplified and repeated at each node they pass through. The bus [22] topology is a long bus (a wire or several wires) that runs through each node and allows nodes to be added or removed without impairing the network. The trend in network topology seems to be the bus and loop and their variants (trees, double loops, etc.).

The determination of a transmission by a particular node is governed by a set of rules of procedure and formal interface format called protocol. The protocols that multiplex

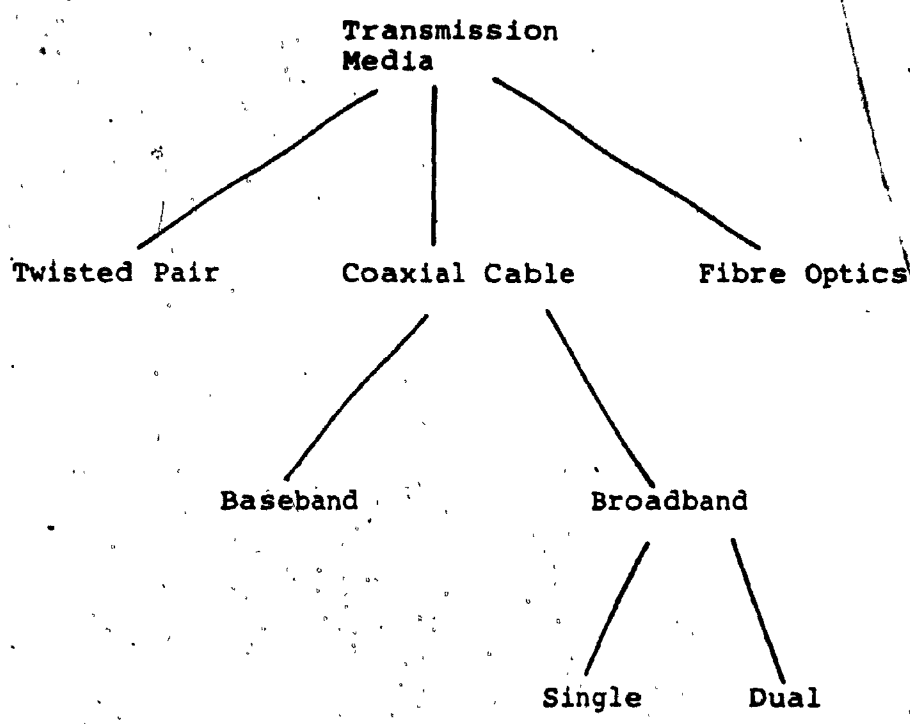
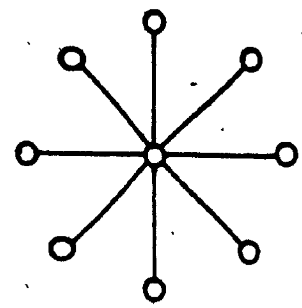
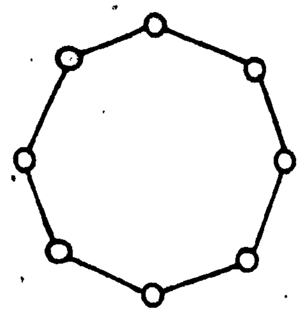


Fig. 1.2 Transmission Media used in Local Area Networks (LANs)

(a) star



(b) ring or loop



(c) tree

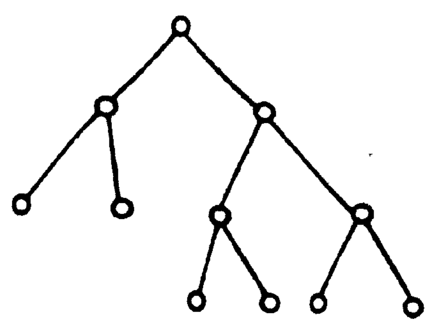


Fig. 1.3 Three Popular Topologies used in Local Area Networks (LANs)

node accesses to the shared communication channel are called the channel access protocols (CAPs). Most of the existing CAPs can be categorized in the following : fixed (or dedicated assignment), random assignment (or contention), and demand assignment protocols (refer Fig. 1.4) .

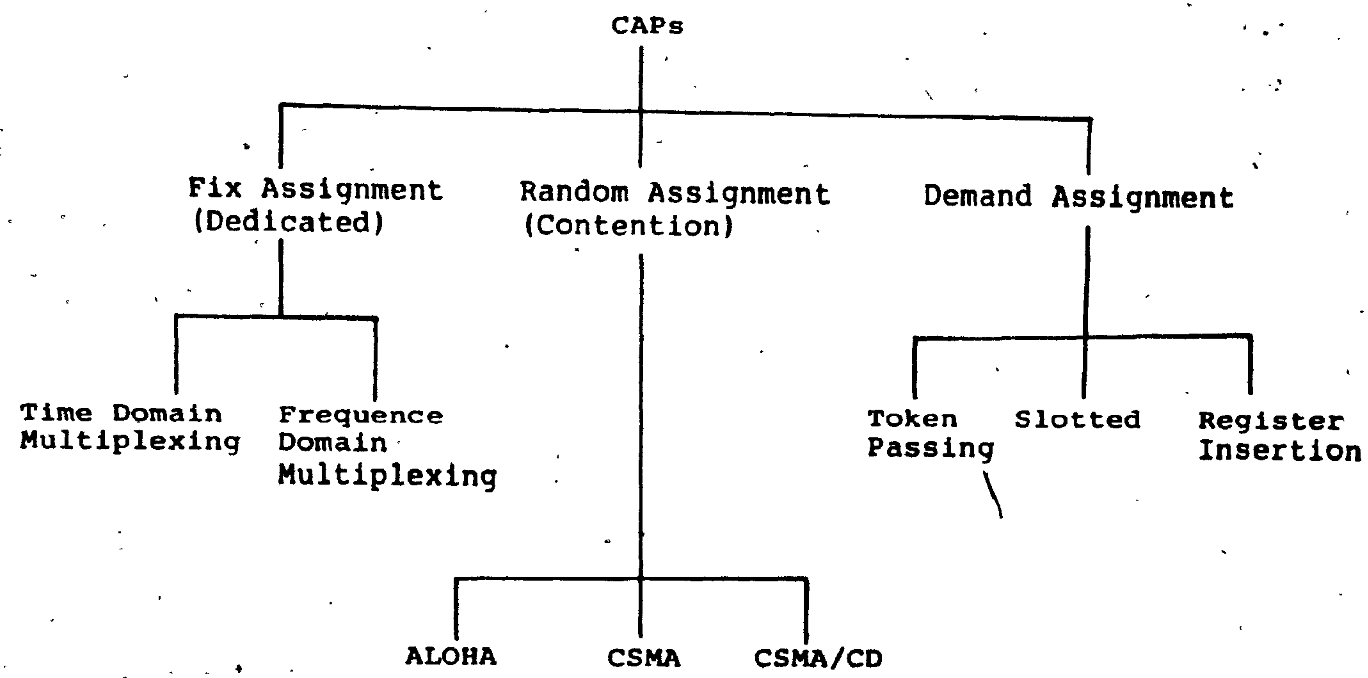
In this work, the main concern is put on the random assignment protocols.

1.2 Random Assignment (or Contention) Protocols

In random assignment protocols, there is no strict ordering of the stations contending for access to the channel. In the random access technique, a station is free to broadcast messages at a time determined by itself without any coordination with the other stations. Systems with a common channel shared among several stations will lead to conflicts. These systems are called contention systems.

ALOHA [1] is a simple protocol where stations transmit regardless of the state of the channel. Collisions occur due to the overlap of two or more packets. The Carrier Sense Multiple Access (CSMA) [16,17,19,31] protocol tries to reduce the potential from collisions by incorporating a listen-before-transmit feature. In this protocol, a station listens for a transmission (senses a carrier) and decides whether to transmit or wait corresponding to the state of the channel, being either idle or occupied. Collisions may still happen due to propagation delays and the lack of coordination between the stations.

The third kind of protocol is the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [31,32] which



CAPs - Channel Access Protocols

Fig. 1.4 Diagram showing the categories of the existing CAPs employs a listen-while-transmitting feature. This permits transmitting nodes to detect collisions while they are transmitting and to abort the transmission once a collision is detected. This improves the bandwidth due to the fact that that collisions are terminated as soon as they are discovered, and reduces the delay caused by collisions. However, the CSMA/CD cannot eliminate collision completely due to the existence of a finite signal propagation delay.

We can therefore see that the basic characteristics of CSMA and CSMA/CD require that each ready station sense the channel prior to transmission. A station will never transmit when it senses the channel busy. Tobagi and Kleinrock described two such protocols known as p-persistent and non-persistent CSMA [16]. The protocol we adopt in this work is a variant of the 1-persistent CSMA/CD (which is a particular case of p-persistent) in which a ready station senses the channel and performs the following procedure.

<u>1-persitent CSMA/CD</u>

1. If the channel is sensed idle, the station initiates transmission of the packet with propability one. As long as no collision is detected, transmission of the packet proceeds. If a collision is detected, transmission is aborted and the station re-schedules the transmission of the packet to some later time and repeats the algorithm.
2. If the channel is sensed busy, the station persists in sensing the channel until it becomes idle, and only then it transmits the message. The station transmits the message with probability one, hence the name 1-per-

sistent.

1.3 Scope of the Thesis

Our focal point of this thesis is to develop an efficient protocol for the H-Network. The H-Network is a part of the Homogeneous Multiprocessor System [10,12] which is shown Fig 2.1. The Homogeneous Multiprocessor is a tightly coupled MIMD architecture that is composed of two parts, namely the Homogeneous Multiprocessor proper and the H-Network. The Homogeneous Multiprocessor proper is designed for problems that can be modeled by a set of parallel processes operating independently of each other. These processes may require occasional exchange of information or control with their immediate neighbours. These applications included image processing through relaxation [34], digital filtering [2], and neural networks [9]. The principal role of the H-Network is to aid the communication between the Homogeneous Multiprocessor and its environment. It will carry user and file traffic in cooperation with front end and back end processors. Moreover, the network enables distant processors to communicate directly without the intervention of intermediate processors. The H-Network is a baseband local area network with a structure which resembles that of the Ethernet [22]. The original protocol followed here is the 1-persistent CSMA/CD. However, by having seperate pathways for data transmission, network acquisition and collision detection, and with the fact that the total physical length is of the order of 10 m, a high data rate ($\sim$7 M bytes/sec) is accomplished.

In order to achieve a better system performance, we introduce the back-off algorithms on the existing protocol. This further reduces the potential for collision. However, this is not the most efficient way to increase the bandwidth or to reduce the packet delay because collisions impose the limitations on the system performance. Thus, the urge for a collision-free protocol is required .

The principal goal of the thesis is to propose a new collision-free protocol for the H-Network. In this work, we will prove that the new protocol guarantees no collisions and we will also investigate its performance improvement over the original CSMA/CD with backoff. The improvement will be studied by comparing the performance of the original protocol to that of the new one based on the system throughput and delay analysis. The original protocol was designed by Kehayas [15] and analysed by Dimopoulos and Wong [14].

In Chapter 2, we present the structure of the Homogeneous Multiprocessors System. This includes the architecture of the multiprocessor proper and the H-Network. In addition, we will analyse the performance of the original protocol of the H-Network. The protocol is further simulated through PAWS [5] for comparison. We also present the performance study of CSMA/CD with backoff algorithm on the H-Network by using the same technique.

In Chapter 3, we present the design of the new collision-free protocol. The performance analysis of this new protocol is done in Chapter 4. Also, a simulation results of this new protocol are given. This is done by means of

PAWS.

Finally, a detailed comparison of the new collision-free protocol and the CSMA/CD protocols with and without backoff, is presented in Chapter 5.

CHAPTER 2

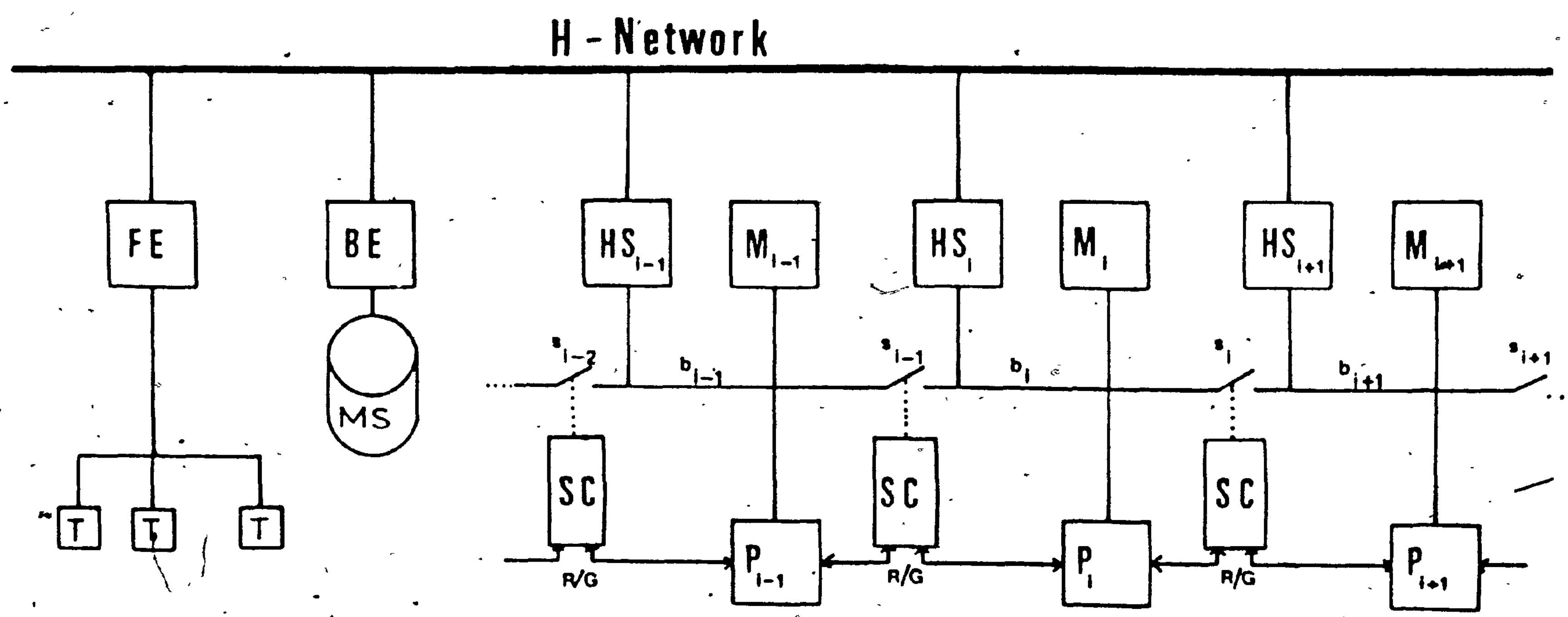
The H-Network Architecture and Protocol

2.1 The Homogeneous Multiprocessor System Overview

The Homogeneous Multiprocessor system is composed of two major parts; namely the H-Network [11] and the Homogeneous Multiprocessor proper [10,12], these are shown in Fig. 2.1.

The Homogeneous Multiprocessor proper is a tightly coupled MIMD structure which consists of k processing elements. Each processing element consists of a processor, a memory module and a local bus. In between every two processing elements, there is a switch called the interbus switch which, when closed, will connect two neighboring local buses in order to form an "extended bus" to provide a communication pathway to facilitate the interprocessor communication. Associated with each one of the switches, there exists a switch controller which will control the switch distributively. The algorithm given in [10] ensures that the operation of the switches are deadlock free and mutually exclusive.

The H-Network is a high speed distributed packet switching local computer network. It is an additional high speed data pathway that facilitates user interaction, control, and file transfer to and from the Homogeneous Multiprocessor as well as information exchange between distant processors. The H-Network is a baseband local area network with a structure resembling that of the Ethernet [22]. By employing parallel pathways for the data transmission, network acquisition and collision detection, we are able to enhance the system



M: Memory
P: Processor
FE: Front End
BE: Back End

MS: Mass Storage
T: Terminal
S: Bus Switch
SC: Switch Controller

HS: Network Station
R/G: Bus request/grant
b: Local Bus

Fig. 2.1 The Homogeneous Multiprocessor Organization and the H-Network throughput and achieve a higher data rate (~ 7 Mbytes/sec) as compared with those of the classical architectures.

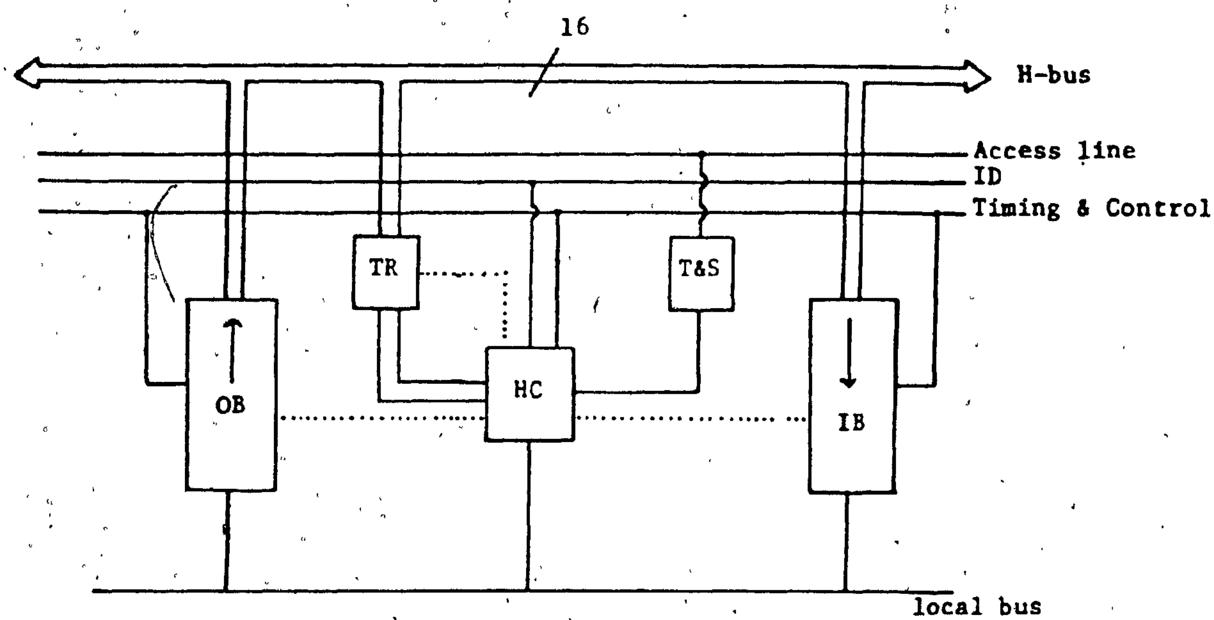
In this chapter, we will concentrate on the structure of the H-Network and its protocol. As for the Homogeneous Multiprocessor, references [10,12,14] give a very detailed description on the architecture and the performance analysis of the multiprocessor system.

2.2 Structure of the H-Network

The H-Network consists of four pathways together with Network Stations (called the H-Stations) that interface the network to each processor of the Homogeneous Multiprocessor proper. These pathways are the H-Bus, the Access Line (AL), the ID line plus the Timing and Control. These are shown in Fig. 2.2.

(a) The H-Bus - this is the high speed data channel which consists of 16 data lines in the present implementation. Data packets are tramsmitted over the H-Bus to facilitate the station-to-station data communication.

(b) The Access Line (AL) - this is a single contention line which is used by the H-stations during the network acquisition phase to ensure the mastership of the network. The AL can be in one of two logical states, either free or occupied that reflect the condition of the H-Network. A station will read and set the AL via a fast Test and Set (T & S) module within a window of vulnerability, a (less than 100ns). This short delay will upgrade the system utilization by providing a higher probability of success in gaining the mastership of the



OB: Output Buffer
IB: Input Buffer

T&S: Test and Set Circuit
HC: Station Controller

Fig. 2.2 The H-Station and the Channel Pathway network.

(c) The ID line - this is used for collision detection purposes. Each station in the network has its unique identification code which will be emitted over the ID line by the transmitting station.

(d) Timing and Control - this group of lines make possible the actual transmission and the packet acquistion through the H-Bus.

Each H-station (refer to Fig. 2.2) interfaces with the H-Bus via the input and output buffers. These buffers are fast FIFO's (128 x 16 bits/word) that are used as temporary storage to capture an incoming packet and to hold an out-going packet until the station controller achieves mastership of the network. Note that the maximun length of the packet (right now is 128 words) can be increased by expanding the capacity of the buffers.

2.3 Protocols of the H-Network

The H-Network protocol is composed of two layers, namely the lower layer protocol and the higher layer protocol. The lower layer is the actual transmission that provides a station-to-station packet delivery to support the higher layer protocol. The details can be found in reference [15].

The higher layer protocol is the one that is of major importance because it decides the synchronization of the stations and directs the physical transmission. Throughout this work, the emphasis will be on the design of the higher layer protocol only.

The H-Network protocol is divided into four phases: network acquisition, packet transmission, collision detection and re-transmission. When a station is ready, i.e. it has a packet to transmit, it has to gain the mastership of the network before it starts transmitting. The network acquisition is carried out by the H-station (through the T&S module) which follows a CSMA protocol on the AL. The H-station, upon a request for the mastership of the network, will continuously sense the AL. Once the AL is sensed idle, the H-station sets the AL busy and it proceeds to transmit the packet and enters the transmission phase. Simultaneously with the start of the packet transmission, the collision detection is initiated. The transmitting station releases its unique identification code to the ID line and at the same time, it listens back the code to test for collisions. A collision will occur when two or more stations sense the AL free and begin transmitting simultaneously. In such case, a collision will be sensed and the transmission will be aborted immediately. At the end of the transmission, the transmitting station will reset the AL from busy to idle again. In case of collision, the colliding station (having terminated the packet transmission) also resets the AL to free. Then it will either repeat the network acquisition for retransmission immediately, or it will execute a backoff algorithm (depending on the protocol with or without backoff). The performance of the network under both protocols will be analysed in Section 2.4.

2.4. Performance Analysis of the H-Network CSMA/CD with and without Backoff Protocols

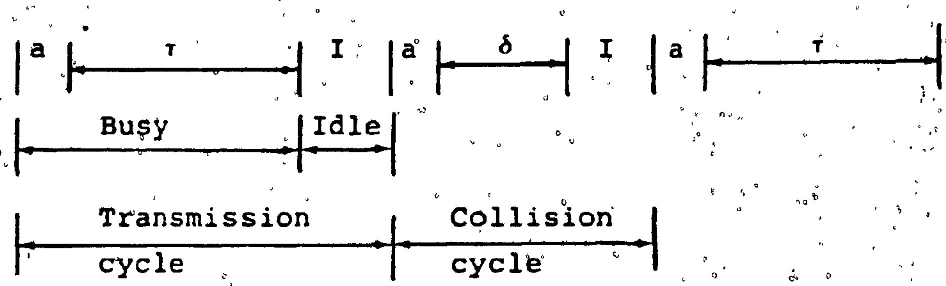
The criteria on the performance measure of a local area network are the system throughput (the average utilization factor) and the average delay (response time). The average utilization factor is defined as the average percentage of time that the network is successful transmitting. The response time, which is defined as message transmission delay from the time the packet arrives at the transmitting station, to the time it is successfully received at the destination station. These two parameters depend on the system architecture as well as the protocol being used.

In this section, we will give a performance evaluation on the H-Network. The work is divided into three parts. First, we present a worst case analytical study on the CSMA/CD protocol of the H-Network by following Dimopoulos' [14] and Tobagi's [16] papers. Secondly, we give a simulation on the performance of the H-Network through PAWS [5]. Finally, we will give the performance evaluation of the CSMA/CD protocol with backoff algorithms by using the same simulation technique.

2.4.1. A worst case study on CSMA/CD on H-Network

In the following analysis, we assume that there are N stations in the system and all these stations are ready to transmit all the time. Before we proceed any further, we would like to give a few definitions. First, the window of vulnerability, a, is defined as the maximum interval of time during which two or more stations may sense the network as idle. Second, the arrival time of a station is defined as the time at which a station requests the mastership of the H-Network. This is the time when the station is ready (i.e. it has a packet to transmit). Third, the transmission period (TP), τ , is defined as the time interval required for a packet to be transmitted. Next, the collision detection interval, δ is defined as the time interval needed for the detection of a collision and the abortion of the current transmission. And the period of latency, T, is defined as the constant finite period of time elapsing between successive attempts of a station to gain mastership of the H-Network. Finally, the retransmission interval d_i; i= 0,1,.....,k; is defined as the time period, a station must wait after it collides in the current busy (transmission or collision) period and it has participated in i collisions since its last successful transmission. In particular, in the case that no backoff is used, we have k=0 and $d_0 = 0$.

The interval of time seperating two consecutive regenerative points is known as a cycle. Hence we can see that the network alternates between the idle period and busy period which constitute a cycle (refer to Fig 2.3). When a busy period ends, all the stations attempt to get the mastership of the network exactly once within the latency period T. Because the H-Network follows the CSMA/CD protocol, the next busy period will be a transmission period (TP) only when the first arriving station is at least one window of vulnerability (or network end-to-end propagation delay), a ahead of the



a - the window of vulnerability

Fig. 2.3 H-Network Acquisition, Transmission and Collision Detection Cycle second requesting station; otherwise a collision will occur. The network acquisition, packet transmission and collision detection cycles are also outlined in Fig 2.3.

The average system utilization factor is defined as follows

$$S = \frac{\overline{U}}{\overline{I} + \overline{B}}$$

where $\overline{U}$ is the average successful transmission period, $\overline{I}$ is the average idle period and $\overline{B}$ is the average busy period. Following reference [14], we derived a simple expression for the worst case analysis of the utilization factor. Given N stations competing for mastership of the network and assume that the requesting time of a station is uniformly distributed in the latency interval [O,T], the average utilization period is given as

$$S = \frac{\tau \left(\frac{T-a}{T}\right)^N}{R+\tau \left(\frac{T-a}{T}\right)^N + \delta\left(1-\left(\frac{T-a}{T}\right)^N\right)} \tag{2.1}$$

where R is the delay interval that the H-Network is sensed idle at a particular time and a transmission is initiated R seconds after. Here we simplify the problem by assuming R = 0.

Since we have considered that there are always N stations active all the time in the H-Network, by applying Little's result [20], the average packet delay (response time) is a function of N which is given as :

$$D = \frac{N}{S}$$

hence, in this case we have

$$D = \frac{N\{R + \frac{T}{N+1} + \tau(\frac{T-a}{T})^N + \delta(1-(\frac{T-a}{T})^N)\}}{\tau(\frac{T-a}{T})^N} \tag{2.2}$$

In Kleinrock and Tobagi's paper [16], a Poisson distribution was used in analysing the performance of the CSMA/CD protocol. With the system parameters in the H-Networks, we can apply their result and model it as a Poisson process with parameter $G = N / T$. Hence the system utilization factor is given as

$$S = \frac{\tau e^{-(N/T)a}}{a + \tau e^{-(N/T)a} + \delta(1-e^{-(N/T)a}) + T/N} \tag{2.3}$$

A plot of the average utilization factor is given in Fig 2.4. The plot includes the results from Equation 2.1 (i.e. uniform distributed) a Poisson distribution from Equation 2.3, and a simulated result run by PAWS. The corresponding simulation model is given in the next section. In addition, a plot of the average packet delay is also presented in Fig 2.5.

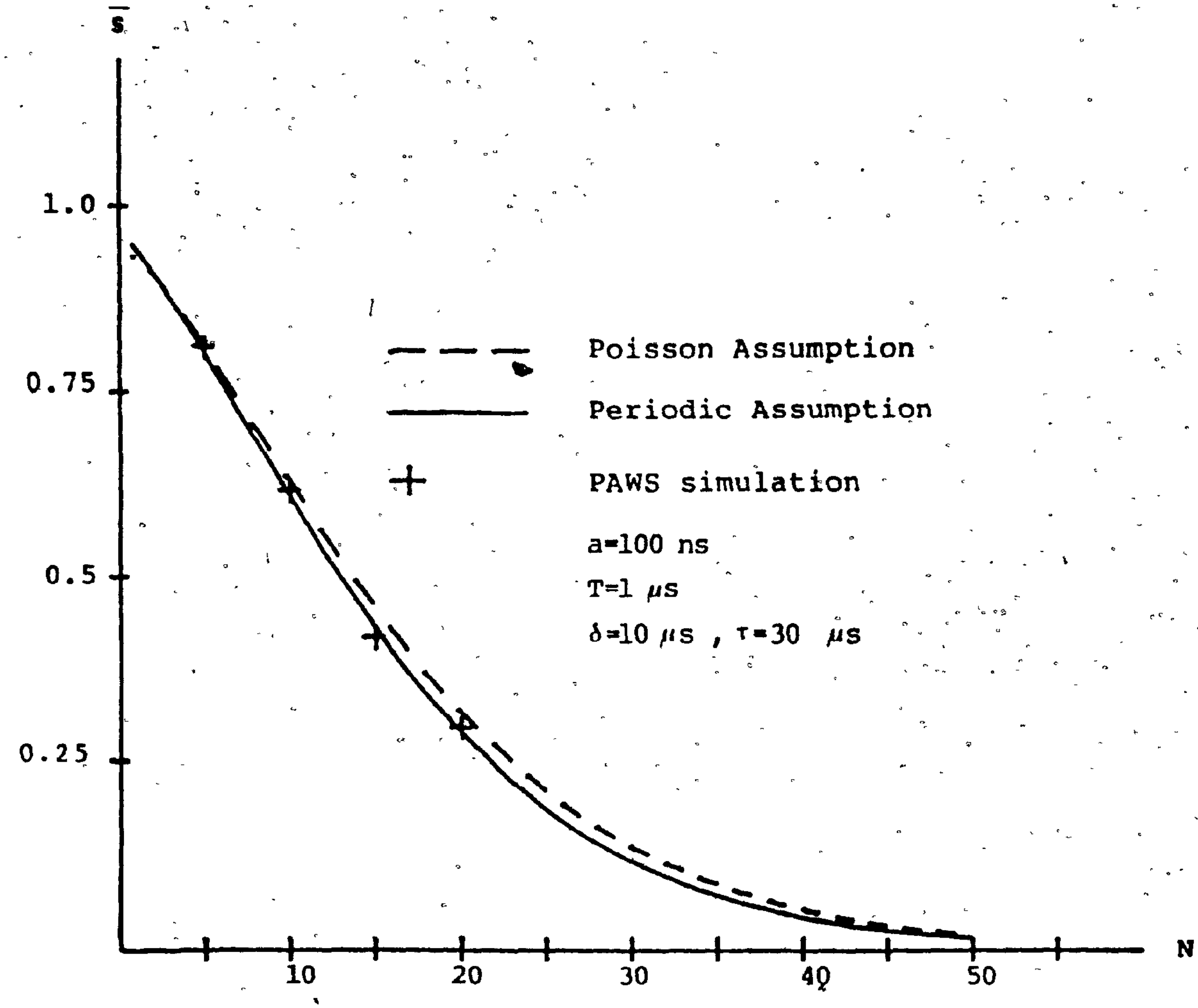


Figure 2.4 The average utilization factor of the H-Network CSMA/CD with no Backoff Algorithms.

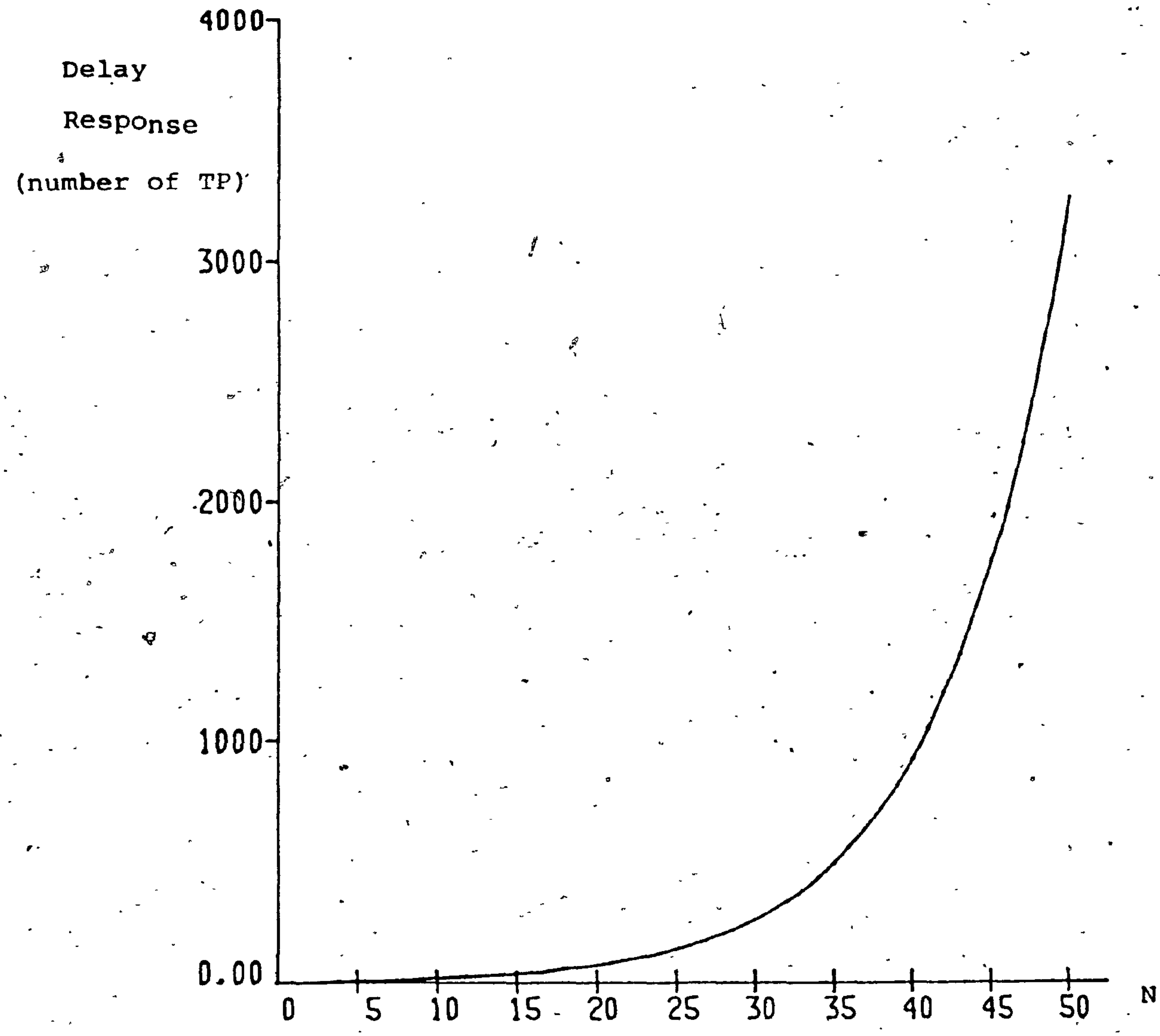


Fig. 2.5 Delay Response of the original H-Network Protocol

2.4.2 Performance Evaluation of H-Network CSMA/CD Protocol Through Simulation

In the following, a performance evaluation on the H-Network CSMA/CD (with no backoff) protocol is modeled by using PAWS [5]. A comparison of the analytical and simulated results is provided.

The H-Network protocol is firstly modeled by Information Processing Graphs (IPG) [5,25] which is a pictorial construct for modelling information processing systems. More explicitly, the IPG is used to describe pictorially the information flow as well as to specify the processing to be done at specific work stations.

These work stations, including central processors and controllers, are called nodes in the IPG. The model is then easily translated from the IPG to a PAWS simulation model. The corresponding IPG model is shown in Fig 2.6 and the PAWS simulation program is attached in Appendix A. The following is a brief description on the IPG model of the H-Network Protocol.

Packets are generated at source node MSOURCE from time to time. These packets will be processed at SETUP1 to fill up the queues at all STATION [I]. The purpose is to simulate the situation that all stations will have packets ready to transmit all the time. The ready time stamp is marked at node RDYTIME. The channel contention is simulated by nodes CNTRL1 and CNTRL2 while node WINDOW is to delay the packets for a period equivalent to a window of vulnerability. The de-

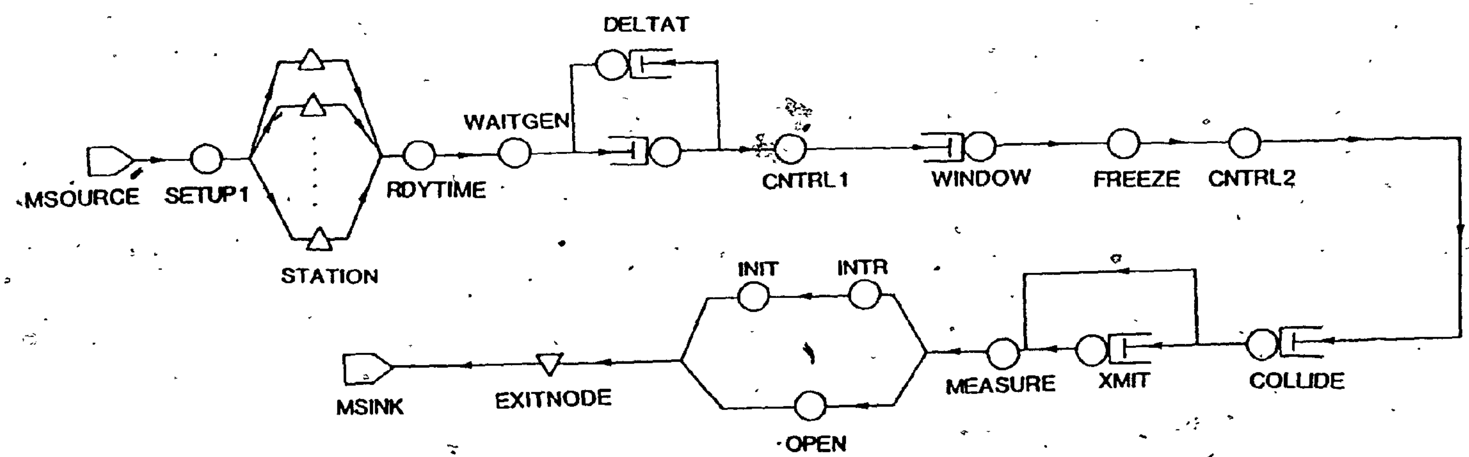


Fig. 2.6 IPG simulation model on H-Network Protocol without Backoff Algorithm

cision whether a packet is successfully transmitted is determined at CNTRL2 which will detect if there are more than one arrivals within the window of vulnerability. The nodes FREEZE and OPEN work in a complementary action. The FREEZE node will stop the packets flowing into node CNTRL1; this represents the CSMA/CD when a ready station arrives at an idle network, the channel will be sensed busy after a propagation delay. On the other hand, when a busy period ends, the channel will be freed and it will be available for contention. This action is simulated by node OPEN.

The colliding packets and the transmitted packet will be sunk in the node MSINK. The statistics are collected at node MEASURE and hence the utilization factor can be evaluated.

In the PAWS program, a very detail description on the function(s) of each node is given in Appendix A.

2.4.3 Performance Evaluation on H-Network CSMA/CD with Backoff algorithm

In both CSMA/CD with and without backoff protocols, if the previous busy period was a successful transmission, then all the N stations in the network will compete for the network at the end of the period. However, these two protocols can be distinguished when a collision occurs. In the case of CSMA/CD with no backoff, the collided stations are not penalised and are allowed to compete immediately when the collision detection terminates. On the other hand, in the CSMA/CD with backoff, the collided stations will have to wait a retransmission interval d_i (as it was defined in Section 2.3) before participating in the channel contention. The other non collided stations will attempt to acquire the network when the busy period ends.

The performance evaluation is again modeled by IPG (see Fig 2.7) and simulated by PAWS (refer to Appendix B). In reference [13], a detail analysis is given on the CSMA/CD with backoff on the H-Network.

The IPG model in the backoff case is similar to that of the no backoff situation except that the collided packets will flow into the nodes BACKGEN and BACKWAIT instead of the node MSINK. The corresponding retransmission delay interval d_i is set at node BACKGEN and the collided packet has to wait d_i seconds at node BACKWAIT.

In the simulation program, we use three types of backoff algorithms, namely the linear, the quadratic, and the combination of quadratic and linear. The corresponding average utilization factors are calculated and the results are given in Fig 2.8 to Fig 2.10.

2.5 Discussion

As shown in Fig 2.4, the theoretical results are close to the results using Poisson Assumption. A further validation is made by the PAWS simulation. The theoretical results show that the system utilization drops abruptly with the increase of the number of stations. In other words, there are many collisions when the traffic becomes heavy. This imposes the limitation on the system throughput which is caused by the collisions.

Fig 2.8 to 2.10 show that the system throuhput improves

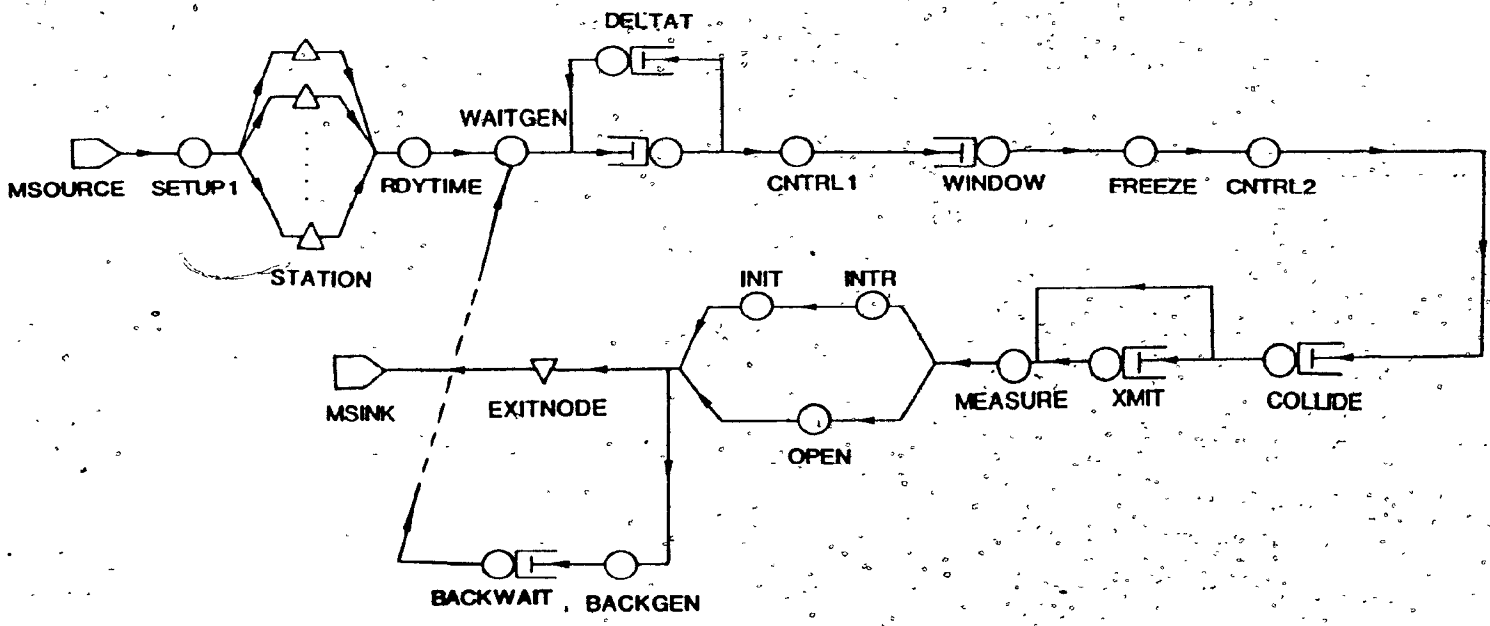


Fig. 2.7 IPG simulation model on H-Network Protocol with Backoff Algorithms

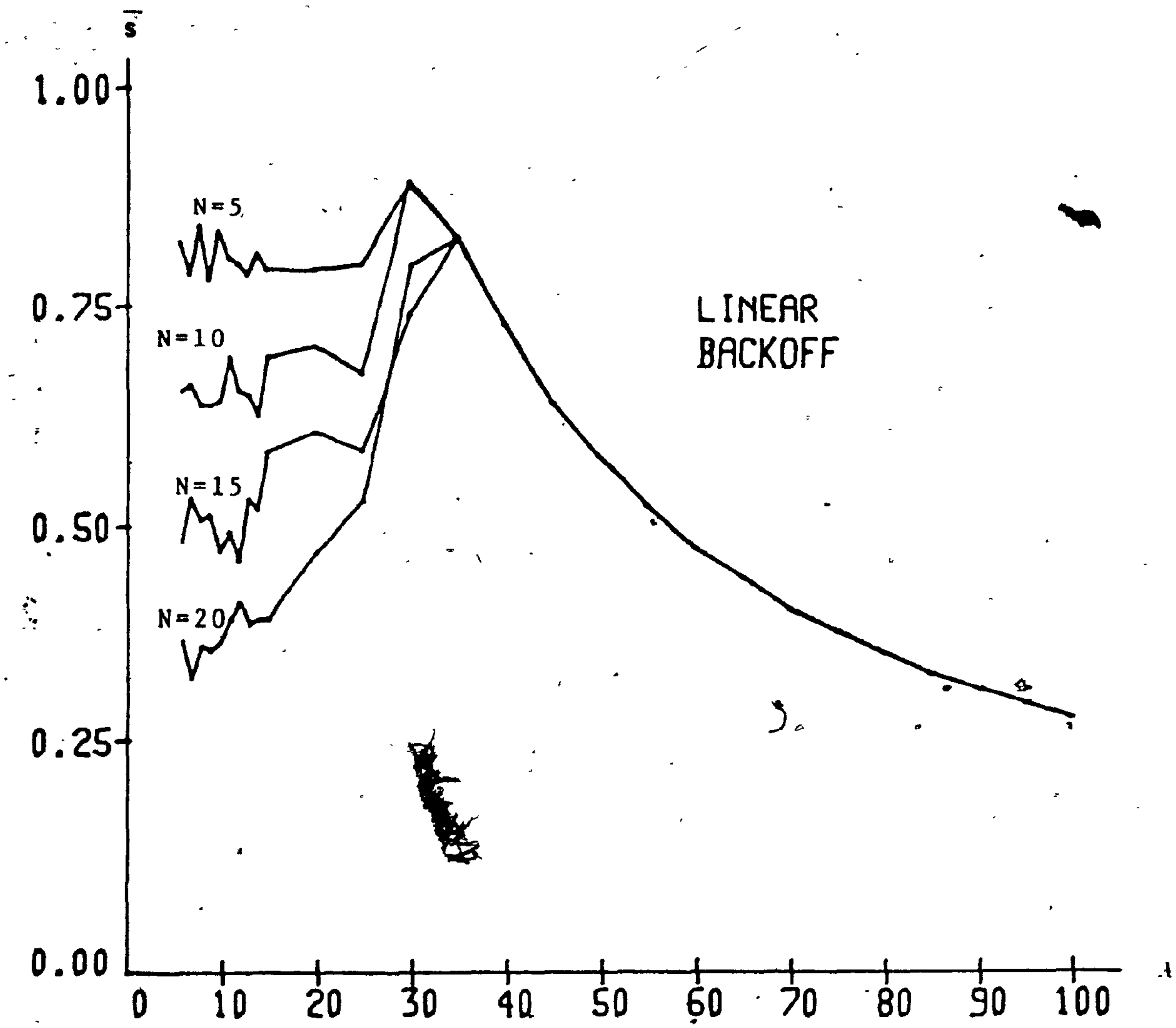


Figure 2.8 The average utilization factor $\bar{s}$ vs the average packet interarrival interval λ for N stations and a linear Backoff Algorithm.

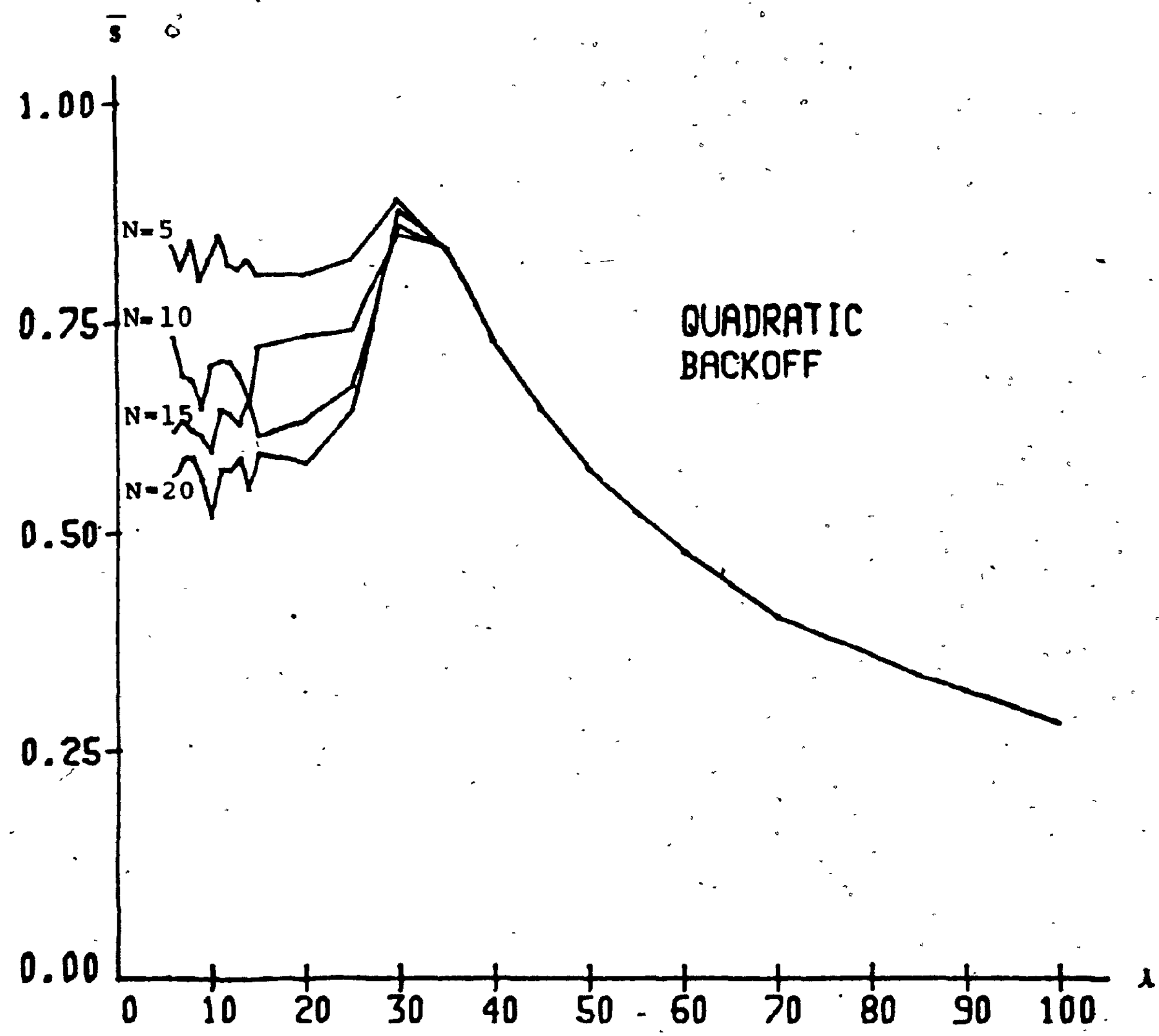


Figure 2.9 The average utilization factor $\overline{s}$ vs the average packet interarrival interval λ for N stations and a quadratic Backoff Algorithm.

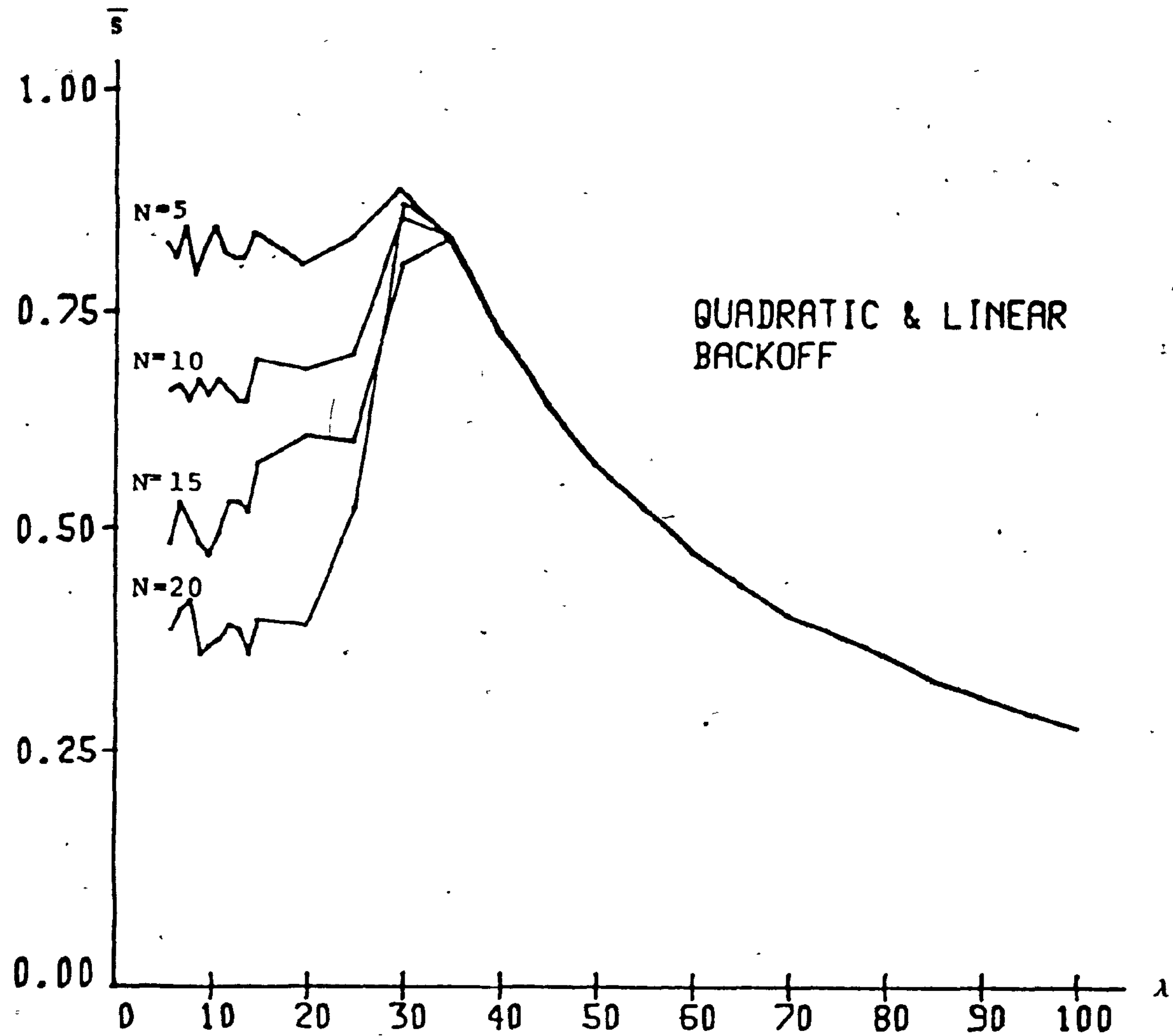


Figure 2.10 The average utilization factor $\overline{s}$ vs the average packet interarrival interval λ for N stations and a quadratic and linear Backoff Algorithm.

with the introduction of backoff algorithms. However, this improvement is not very effective especially when the traffic is heavy, i.e. the packet interarrival interval is small. Among three backoff schemes, the quadratic backoff is the most effective.

CHAPTER 3

An Introduction to a New Collision Free Protocol

3.1 Design Objective

In this chapter, we will present a new protocol for the H-Network. As we have seen, in the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) scheme, the ready station avoids collisions by listening to the carrier due to another user's transmission. However, collisions still happen due to the finite signal propagation delay. It is because it takes at most one end-to-end propagation delay, a, for the transmitted packet to reach all devices. A collision may occur if another transmission is initiated before the current one is sensed, and it then will take at most an additional propagation delay before interference reaches all devices. These collisions introduce a limitation on system throughput even with the introduction of backoff algorithms. The results were shown in Chapter 2.

In order to enhance the system performance, the ideas of parallelism and pre-scheduling are introduced. A simple method is to have the channel contention period undergo while a transmission is taking place. The contention period is the time interval during which a ready station competes with the other ready stations to gain the mastership of the channel (H-Bus). The period begins when a ready station arrives on an idle network, or when the previous contention period has finished (but nobody has acquired the mastership of the network for the next TP).

However, these two activities, channel contention and transmission, have to occur without interference from each other; otherwise errors may arise in transmission and channel contention. This is quite easily implemented in the structure of the H-Network as it employs a parallel data-highway for data transmission and a seperate control line (the AL) for channel contention. The functions of these lines and the network architecture has been given in Chapter 2. The system utilizations factor of the network may achieve its maximum only when collisions are eliminated. Thus a new CAP (Channel Access Protocol) will be introduced to govern the activities in the network so that no collisions will appear even in heavy traffic. The main principle of the protocol is given in the next section.

3.2 The Pebble Pool Paradigm

Suppose there is a resource (the channel) shared among a number of users. Only one user can utilize the resource at any one time. Everyone has one pebble at hand. This pebble is the token (key) which allows the user to acquire the resource. There is a pool which is public for everybody to put their pebbles in. Whenever anyone wants to use the resource, he has to check the pool first. If there are any pebbles inside it (i.e. somebody else is trying to acquire or has already acquired the resource), he is blocked (or backlogged), thus, he has to reschedule and try again at some later time. If the pool is empty, a user will perform the following actions: wait for a period of time; place his pebble in the pool, wait for a second period of time and recheck the pool again. The first waiting period is to provide an interval of time for all the other attempting users to enter the competition. The second waiting period is of vital importance. After this second waiting interval, the number of pebbles inside the pool will represent the actual number of attempting users. This is useful when a user rechecks the pool. If he finds that there is only one pebble (his own pebble) left inside the pool, it means that he is the only contending user, thus he will gain the mastership of the resource for the next period. As a consequence, this user will start using the resource immediately* after the current resource utilization period has finished. Otherwise, if there are more than one pebbles in the pool, he waits for a third period of time, takes back his pebble and tries again later on. The purpose of the third waiting period is to slow down the frequency of the acquisition of other attemting users. This is especially useful when the traffic is heavy.

In the case that a user succeeds in gaining the mastership of the resource, he will retrieve his pebble from the pool as soon as he starts using the resource. At that time, the other users (no longer backlogged) will repeat the same procedure and compete for the resource for the next utilization period.

---

* In the actual situation, there may be a small delay from the time he finds that he is the one with the pebble in the pool to the time he starts using the resource.

In the H-Network, the single resource H-Bus is shared by all stations. This channel can only be accessed by one station at a time. Placing the pebble in the pool corresponds to the TEST-AND-SET actions on the AL which controls the mastership of the network by a single station. Here there is a slight hardware modification on the AL. We introduce an analogue signal on AL such that this signal reflects the total number of pebbles in the pool. These modifications will be given in Appendix C.

The new protocol guarantees that no two stations will overlap in their transmission. This will be analysed in Chapter 4. In addition, the transmission periods and contention periods proceed in parallel. At the end of a contention period, at most one station has acquired the network. Thus collisions are eliminated and in cases that the total contention period is less than a transmission period, idle periods are also eliminated with a corresponding increase to the network utilization factor. Again, this will be discussed in Chapter 4.

3.3 Structure of the New Protocol

The New Collision Free protocol can be divided into four phases; namely the decision phase, the transmission phase, the ready-wait phase and the backlog phase. In Section 3.2, we have described the main idea of the protocol. In the following section we will give the detailed activities of the protocol of the H-Network.

When a station is ready (i.e. it has a packet to transmit), it has to gain the mastership of the channel (H-Bus) before it starts transmitting. Firstly, it checks whether the network is in the midst of a contention period by sensing the AL. If the AL is sensed busy, this indicates that one or more stations are currently contending for the channel and thus it is blocked. It has to reschedule and retry at some later time. If the AL is sensed idle, then no one is contending * the network and therefore the station is free to start its own contention period. Thus, it waits for an interval of time and then deposits its pebble by setting the AL to busy. This will definitely block any other fore-coming ready stations after at most an additional propagation delay, a. Afterwards, this station has to wait for a second period of time before it re-checks the AL . At that moment, if more than one station are still contending for the bus, this station fails to get the mastership of the H-Bus ; it then resets the TEST-AND-SET signal on the AL (i.e. withdraws its pebble) after it has waited a third period of time. Then this station repeats the attempt to gain mastership of the network some time later.

However, if the station is the only one left contending for the channel, it will gain mastership of the network during the next transmission interval. For this to happen, the channel (H-Bus) must be freed from the current transmission. The successful station will therefore wait until there are no

---

* Another contention period may have started less than a network propagation interval before, but due to propagation delays this information has not reached the current station yet.

more transmissions on the bus before it will transmit its own packet. Simultaneously with the start of transmission, it releases the AL by resetting the TEST-AND-SET signal. The purpose is to let the other stations compete for the channel (H-Bus) in the next transmission period (TP).

We see therefore that a station ready to transmit goes through the following four phases :

(1) the decision phase: during which it decides whether it will be allowed to transmit during the next transmission period or be backlogged;

(2) the transmission phase: during which one successful station transmits;

(3) the ready-wait phase: a user has acquired the channel for the next TP and it will start its transmission (i.e. enter the transmission phase) when the current transmission is terminated;

(4) the backlog phase: which it is entered from the decision phase in case that the contending station did not acquire the channel for the next transmission period.

These four phases are shown in Figure 3.1.

Note that the detection of the pebble pool and the deposition of the pebble in the pool are the only activities involving on the AL. One hardware implementation is to have a detection scheme on the AL . This line is common to all stations. It is tied to an open collector buffer and all the TEST & SET controllers in each station (refer Chapter 2). The amount of current that sinks into the buffer

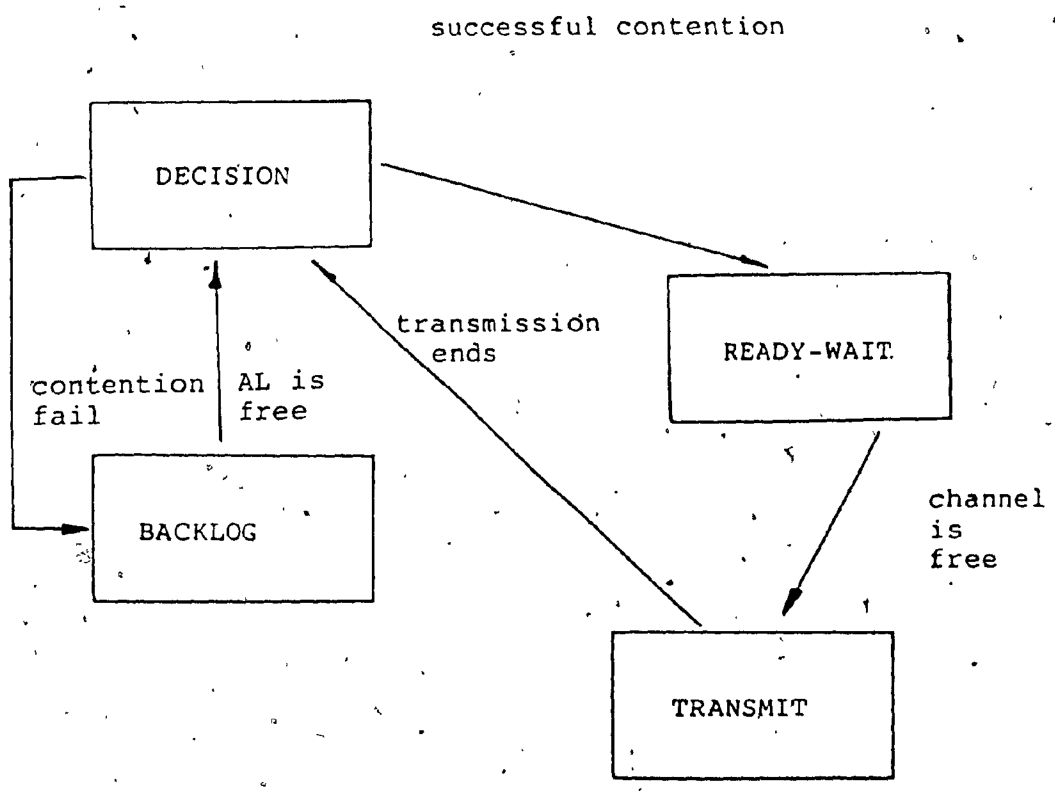


Figure 3.1 State diagram showing the four different phases of the new collision-free protocol

will be proportional to the number of TEST & SET signals being set on the AL . Therefore we can detect the number of contending stations by sensing the value of the current. There are three thresholds on this signal that correspond to three important states of the network :

(1) there is no contending stations

(2) there is only one contending station; and

(3) there are more than one contending stations.

In Figure 3.2, we give the detailed pictorial description on the activities inside the decision phase (the pebble pool paradigm). Note that the deposition of the pebble in the pool is the activity involving the seting and sensing of the AL. The proposed hardware implementation modifications are given in Appendix C.

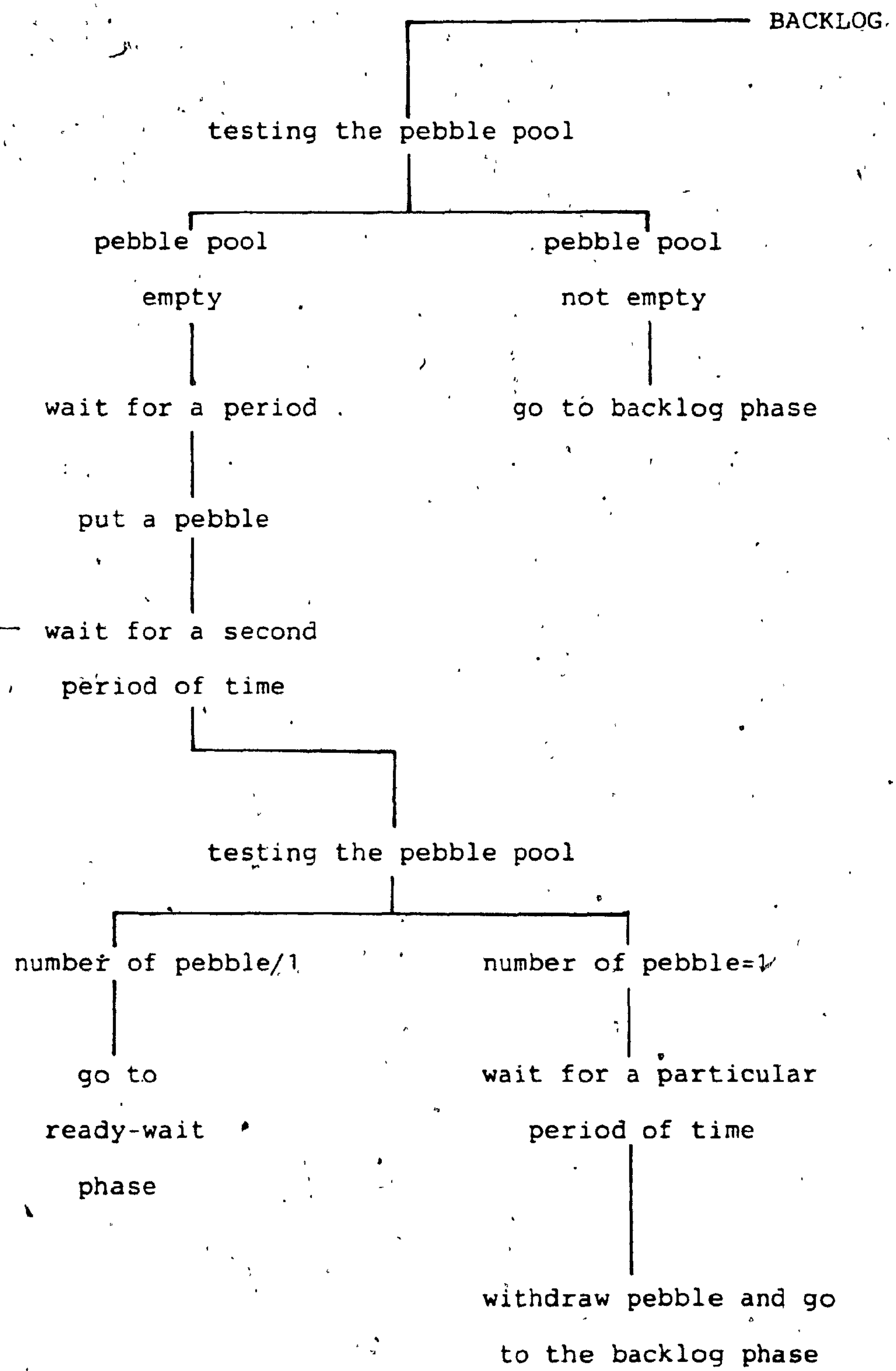


Figure 3.2 Diagram showing the activities in the decision phase of the Pebble Pool Paradigm

CHAPTER 4

Performance Analysis on the New Protocol

4.1 Introduction

In this Chapter, our major objective is to evaluate the average utilization factor in order to investigate the effectiveness of the new protocol. To simplify the problem, we introduce some notations and definitions in Section 4.2.

The study will utilize the definition of the average utilization factor given as

$$S = \frac{\bar{U}}{\bar{I}+\bar{B}}$$

which was described in Chapter 2.

In Section 4.2, we also present the relationship between the first waiting period (W_1) and second waiting period (W_2) such that the protocol will be free from collision. The corresponding proof is given in Section 4.3.

Before we calculate the average utilization factor, the probability of successful transmission has to be determined. Section 4.4 gives the conditions for successful transmission while the calculations are done in Section 4.5.

Due to the structure of the network protocol, the transmission period is fixed but the length of the contention period is variable. This variable length depends on the number of arrivals of ready stations within the first waiting period (W_1) after the first station has checked and found that the pebble pool (AL) is empty.

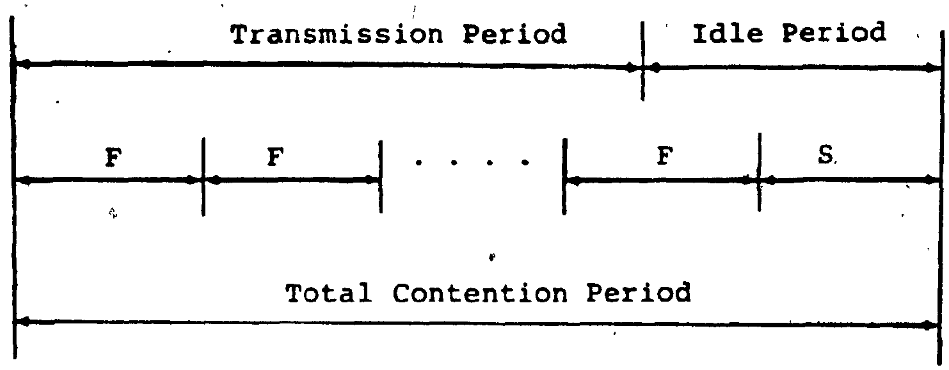
All ready stations will continuously compete for the channel until there is one station which has acquired the mastership of the network in the next TP (we call this a successful contention). A successful (unsuccessful) contention period is the contention period after which one (none) of the ready stations has gained the mastership of the channel.

The total contention period is the total time interval elapsed from the moment the first contention period started to the moment that a successful contention has been accomplished, i.e. it is the sum of all the unsuccessful contention periods plus the final successful contention period (refer to Fig. 4.1).

If it happens that the total contention period is longer than the transmission period, then this will result in the existence of idle periods even with a continuous heavy traffic. Since there are no collisions, the busy period will be the transmission period itself, thus the existence of the idle period will affect the system utilization factor directly. The details will be given in Section 4.5.

4.2 Notations and Definitions

Let W_{i1} be the first waiting period, i.e. the interval elapsed from the moment the i^{th}-contending station detects that the pool is empty to the moment it deposits its pebble in the pool. Denote by W_{i2} the second waiting interval which corresponds to the time interval elapsed from the moment the i^{th} station places its pebble into the pool to



F - Unsucessful Contention Period
S - Sucessful Contention Period

Fig. 4.1 Relationship between Total Contention Period, Transmission Period and Idle Period the moment it rechecks the pool. Finally, denote by W_{i3} the time interval elapsed from the moment the i^{th} station tests the pool to the moment it retrieves its pebble in case it failed to obtain mastership of the network.

Let us assume that at time t_0 the channel is idle and that only one station S_0 is ready at this instant. Fig. 4.2 shows the timing diagram of the activities of this station. t_0 is the ready time, i.e. the time at which the station has a packet to transmit. W_{01} and W_{02} will be the two waiting times.

Clearly, the station S_0 sets the AL at time $t = t_0 + W_{01}$. Due to the propagation delay, a, all the other ready stations will sense the AL as being set the latest at time $t = t_0 + W_{01} + a$. Let us call this instant t_{BLOCK}. Thus $t_{BLOCK} = t_0 + W_{01} + a$ which depends on the value of t_0. In other words, if a station S_0 senses the AL idle at time t_0, then all the other stations will definitely be blocked after an interval of $(W_{01}+a)$. Suppose there is another station S_1 becoming ready at time instant t_1 where t_1 is between t_0 and t_{BLOCK}, S_1 will still sense the AL idle at this moment (remember that the AL becomes set at $t = t_{BLOCK}$). Again S_1 will put its pebble at $t = t_1 + W_{11}$. Thus, in order for a station to make sure that all the other contending stations put their pebbles in the pool (before anyone starts checking the pool), each station has to wait for a period of time. This is the waiting period W_{i2}, for $i = 0,1,\ldots,N-1$. We will determine now the minimum value of W_{i2} which will

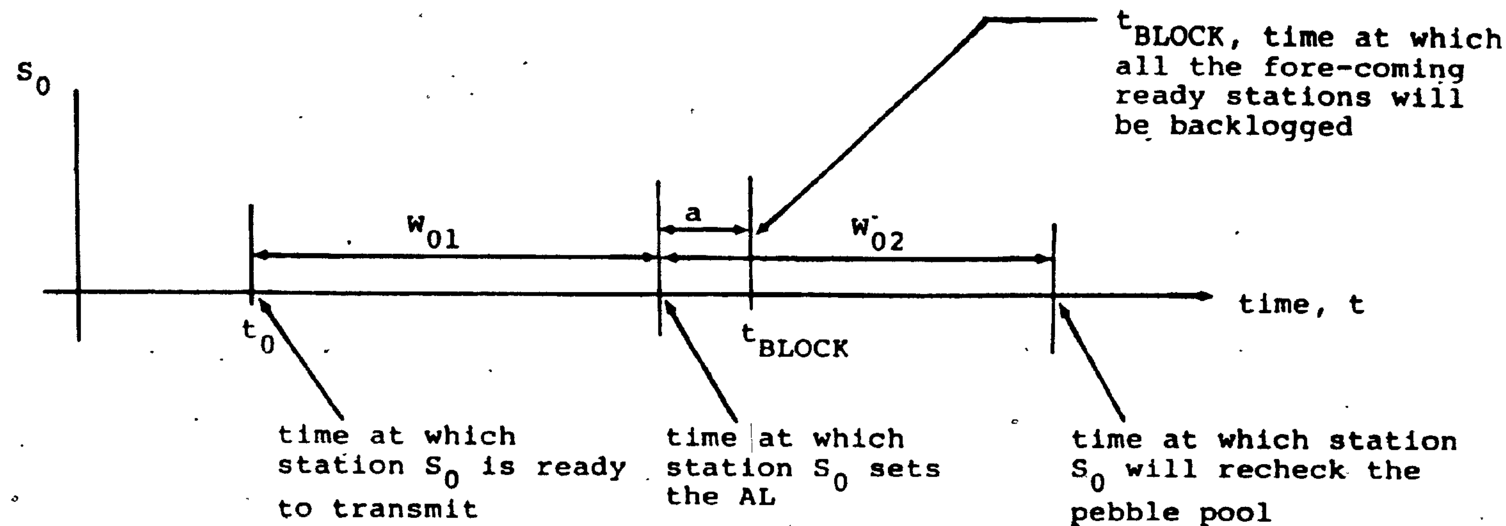


Fig. 4.2 Timing diagram showing the two waiting times of a ready station in the Decision Phase

ensure all the stations that sensed the AL as being idle, had enough time to set this line busy (put their pebbles).

Consider the limiting case where S_0 is the first ready station at time t_0 and S_i is the last ready station at time $t_{BLOCK} - \delta t$ where $\delta t \to 0$. All the other ready stations will be backlogged as they will recognize AL as being set for all times $t > t_0 + W_{01} + a$. Station S_0 puts its pebble in the pool exactly at $t = t_o + W_{01}$ while S_i will put its own at $t = t_i + W_{i1}$. All the stations in the network realize that S_i has put its pebble in the pool at $t = t_i + W_{i1} + a$ (due to the propagation delay, a, over the transmission medium). Also station S_0 will examine the pebble pool at time $t = t_0 + W_{01} + W_{02}$ in order to determine whether it is allowed to transmit or not. For S_0 to detect the existence of S_i's pebble in the pool, S_i has to place its pebble in the pool at least a network propagation delay before S_0 checks it. Hence we have

time at which S_0 examined the pebble pool for transmission	$>$	time at which S_i put its pebble plus a network propagation delay

or $\quad t_0 + W_{01} + W_{02} \quad > \quad t_i + W_{i1} + a$

i.e. $t_0 + W_{01} + W_{02} \quad > \quad t_0 + W_{01} + a - \delta t + W_{i1} + a$

since $\delta t \to 0$, we have

$$W_{02} > W_{i1} + 2a$$

W_{i1}, in general, is different for different stations, hence the limiting value for W_{02} is

$$W_{02} > 2a + \max_{i} \{W_{i1}\} \text{ for } i = 0,1,..,N-1 \tag{4.1}$$

where N is the number of stations in the network.

In order to simplify our calculations, we assume that W_{i2} is the same for all stations, therefore we have

$$W_{i2} = W_2 > 2a + \max_{i} \{W_{i1}\} \text{ for } i = 0,1,\ldots,N-1$$

Recall that our major objective is to eliminate collision so that the system utilization and throughput of the H-Network will improve.

In the next section we provide a proof of the correctness of the proposed protocol, i.e. a proof that our protocol provides for at most one station to be the master of the network at any particular time.

4.3 Proof of Correctness of the Transmission Protocol

The major objective in this section is to prove that the proposed protocol is collision free. The proof is by means of contradiction.

Let us assume that there are two or more stations transmitting at the same time, in other words, these stations have followed the protocol suggested and all of them became masters of the network at the same time. This situation can only happen when two or more stations detect the presence of only their own pebble in the pool.

In short, these stations have gone through the following sequences:

1. they are ready and have entered the decision phase after having examined the pebble pool;
2. finished waiting for W_{i1}, then put their pebbles in the pool;
3. examined the pebble pool again after having finished waiting for a period W_{i2} and each one of them found only one pebble in the pool;
4. these stations enter the ready-wait phase and wait for the next transmission period.

Assume that there are two ready station this analysis can be extended to N, where $N > 2$. S_0 is the first ready station with ready time being t_0, waiting times being W_{01} and W_2 respectively. Let S_1 be the other station with t_1, W_{11} and W_2 representing the ready time, and the waiting times.

As from Chapter 3, S_1 will be able to place its pebble provided that it senses the AL before this line is set by S_0. Otherwise S_1 will be blocked and it will go to the backlogged phase (refer Fig. 4.3). Thus we have

$$t_1 < t_0 + W_{01} + a \qquad (4.2)$$

then station S_1 places its pebble in the pool at time

$$t = t_1 + W_{11}$$

However, S_0 will only notice this event after another period

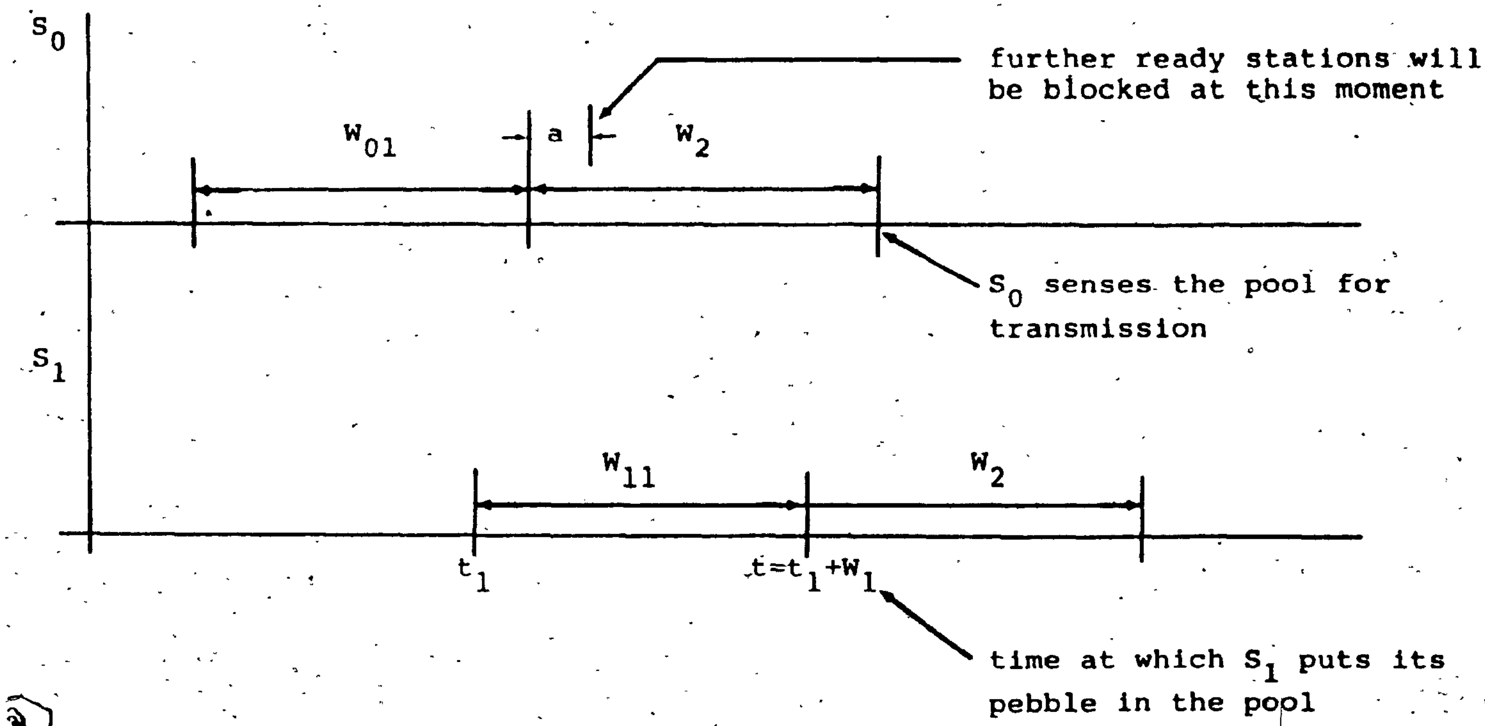


Fig. 4.3 Timing diagram showing the time interval that a second ready station can enter the contention after the arrival of the first station

of propagation delay. As it happens, S_0 senses the pebble pool at time $t = t_0 + W_{01} + W_2$. If S_0 were not able to detect the pebble placed by S_1 then the following must happen

$$\begin{array}{l} \text{time for } S_0 \text{ to} \\ \text{sense the pebble pool} \\ \text{before transmission} \end{array} - \begin{array}{l} \text{time for } S_1 \\ \text{to put} \\ \text{pebble} \end{array} < a$$

i.e. $(t_0 + W_{01} + W_2) - (t_1 + W_{11}) < a$

or $(W_{01} - W_{11}) + t_0 + W_2 < t_1 + a$

and from equation (4.2), we obtain

$$(W_{01} - W_{11}) + t_0 + W_2 < t_1 + a < (t_0 + W_{01} + a) + a$$

i.e. $W_{01} - W_{11} + t_0 + W_2 < (t_0 + W_{01} + a) + a$

i.e. $W_2 < W_{11} + 2a$ (4.3)

hence (4.3) is a necessary condition for stations S_0 and S_1 to collide, i.e. a collision will happen provided that $W_2 < W_{11} + 2a$. However, $W_2 > \max_i \{W_{i1}\} + 2a > W_{11} + 2a$, a contradiction because of (4.3).

From the analysis above, we can conclude that the suggested protocol eliminates collision.

4.4 Conditions for Successful Transmission

In this section, we will find the possible conditions necessary for a station to gain mastership of the network after one contention period. For simplicity, let $W_{01} = W_{11} =$

$W_{N-1,1} = W_1$ and the waiting time before testing for transmission is W_2 for all stations. There are two situations that a station may successfully occupy the channel:

1. this station is the only ready station that wants the channel for a certain period of time, or
2. this station competes with other ready stations and under some conditions it succeeds in getting the channel for the next transmission period.

These situations and conditions will be discussed in details below. For the first case, if the channel is idle, upon the arrival of one ready station, the channel will definitely be occupied by this station in the next TP (Transmission Period) provided that there are no arrivals during the next (W_1+a) seconds. The corresponding probability will be calculated in Section 4.5. For the second situation, there are more than one ready stations competing for the next TP. The protocol allows all the non-blocked competing stations to place their pebbles in the pool unless they are blocked. The station that finds out that there is only one pebble left (clearly the last arriving station), will succeed in getting the next TP. This is possible since all the failing stations have withdrawn their pebble from the pool. We will consider a two station competing situation first, and then we will discuss the general case involving more stations.

Figure 4.4 shows the timing diagram of these two

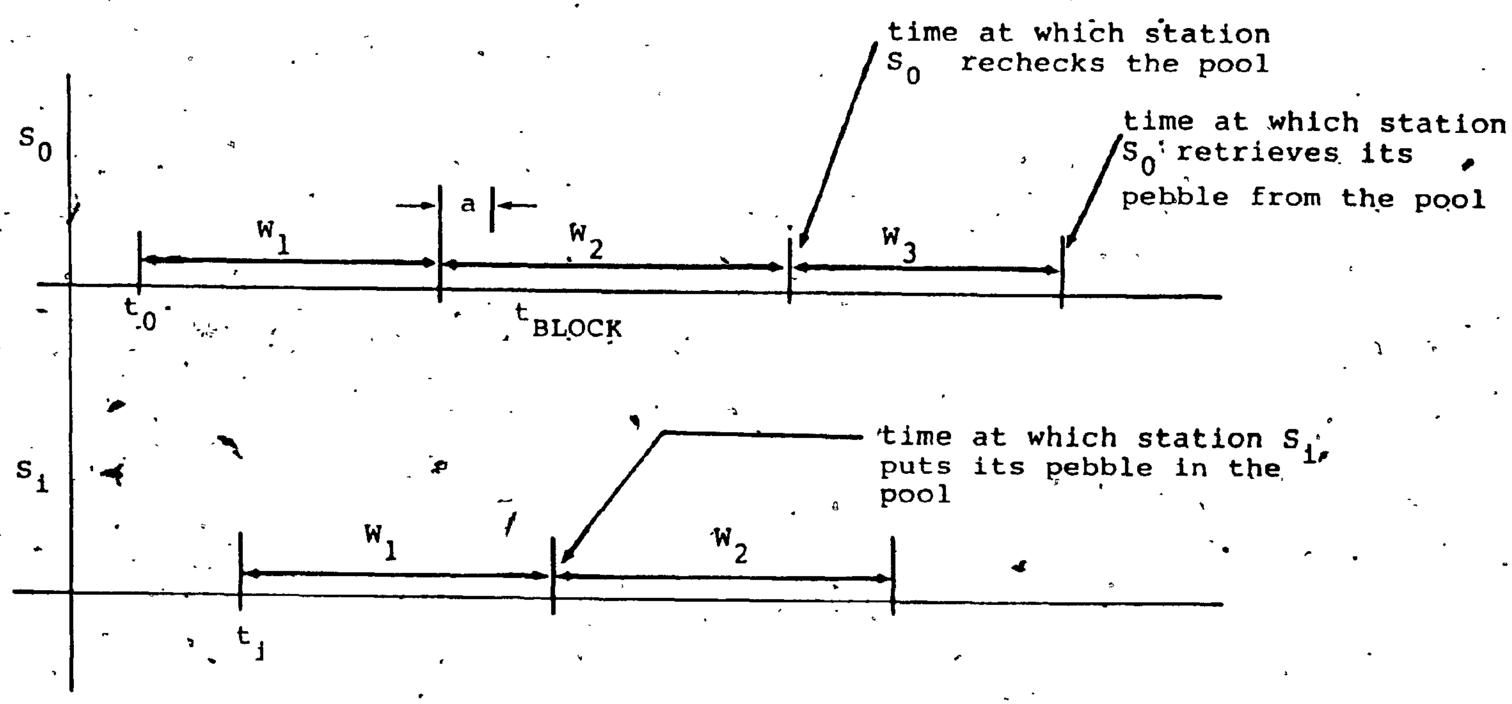


Fig. 4.4 Timing diagram showing the contention activities of station S_0 and station S_i

stations. S_0 is the first ready station at time t_0 and S_1 is the second ready station at t_1. Station S_0 puts its pebble in the pool at time $t = t_0 + W_1$ while station S_1 puts its own at time $t = t_1 + W_1$. Station S_0 will recheck the pool at time $t = t_0 + W_1 + W_2$. At that moment, it will realize that there is more than one pebble in the pool, and as a consequence, S_0 will take back its pebble after an interval of time W_3. However, due to the propagation delay, station S_1 will only know that station S_0 has taken back its pebble after a time $t = t' + a$ where $t' = t_0 + W_1 + W_2 + W_3$. If station S_1 senses the pool (testing for transmission) before time $t = t' + a$, it will notice that there is more than one pebble inside the pool. Naturally station S_1 fails to get the next TP, it waits for a time period W_3, takes back its pebble and finally switches to the backlog state.

In order for station S_1 to acquire the channel, we have the following condition:

time at which station S_1 tests the pool for transmission	>	time at which station S_0 has withdrawn its pebble from the pool plus a network propagation delay

or $\quad t_1 + W_1 + W_2 \quad > \quad t' + a$

i.e. $\quad t_1 + W_1 + W_2 \quad > \quad t_0 + W_1 + W_2 + W_3 + a$

i.e. $\quad t_1 - t_0 - W_3 - a \quad > \quad 0 \qquad (4.4)$

Also, for station S_1 to be able to put the pebble in the pool, it must become ready before it is blocked (at time t_{BLOCK}). Hence

$$t_0 < t_1 < t_0 + W_1 + a \tag{4.5}$$

From condition (4.4) and (4.5), we have the following

$$t_0 + W_3 + a < t_1 < t_0 + W_1 + a \tag{4.6}$$

In other words, the arrival time of station S_1 must be at least at a time interval (W_3+a) after the ready time of station S_0 but not exceed the blocking time of station S_0. Also, expression (4.6) implies that $W_1 > W_3$ is a necessary condition for the station S_1 to gain the mastership of the bus.

Consider now the case of more than two arrivals. Let station S_i be the last arriving station while sation S_{i-1} is the last but one. Station S_0 is the first ready station which governs the blocking time t_{BLOCK}. There is an arbitrary number k ($k = 0, 1, 2, \ldots, N-2$) ready stations between the ready times of station S_0 and station S_{i-1} (refer to Figure 4.5). These stations are not important because only the last arriving station is given the possibility to be successful in acquiring the channel.

Clearly equation (4.6) can be applied here. In order for the last station S_i to get the mastership of the channel, $t_i - t_{i-1} > W_3 + a$ where t_i and t_{i-1} are the arrival times for station S_i and S_{i-1} respectively. In short, the last but one station has to be at least in an

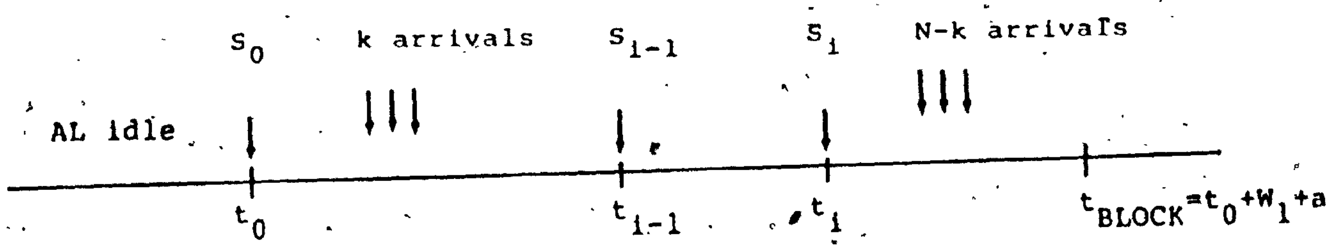


Fig. 4.5 Timing diagram showing the arrival times of the ready stations in the N stations contending situation

interval (W_3+a) seconds ahead of the last arriving station in order to leave enough time for the last ready station to gain the mastership of the channel. Figure 4.5 shows the timing diagram of the situation.

Note that the last but one station S_{i-1} may be actually the first ready station S_0 itself. This is exactly like the two station competing case that we have analyzed earlier.

4.5 Probability of Success in getting the next TP and the Determination of Existence of Idle Periods

In this section, we will determine the probability of getting a successful next TP (the corresponding conditions have been analysed in Section 4.4). Furthermore, we will determine whether there exist any idle periods. The final goal is to calculate the average system utilization factor which was defined in Chapter 2 as follows:

$$\bar{S} = \frac{\bar{U}}{\bar{I}+\bar{B}}$$

where $\bar{U}$ is the average successful transmission period, $\bar{I}$ is the average idle period and $\bar{B}$ is the average busy period.

We assume that there is always a constant finite period of time elapsing between successive attempts of a station to gain mastership of the channel (H-BUS). We denote this period of latency by T. Assume also that there are always N competing ready stations in the network. Thus, the probability distribution of attempting the trial (to gain the mastership of the channel) is continuous and uniformly distributed in [0,T]. We adopt one more notation below in order to clarify our calculation. Denote by $P_k(t_2-t_1)$ the probability that there are k arrivals, k=0, 1, 2,...,N-1 in an interval $[t_1, t_2]$, where $t_1, t_2 \in [0,T]$ and $t_2 > t_1$; i.e.

$$P_k(t_2-t_1) = P_r \text{ [k arrivals in the time interval } t=t_2-t_1;\ t_1,t_2 \in [0,T],\ t_2 > t_1]$$

where

$$k = 0,\ 1,\ 2,\ldots,\ N-1$$

Let us consider the first case where only one station becomes ready on an idle channel. Denote by t_0 the arrival time of station S_0; recall that W_1, W_2 are the time intervals that elapse between the sensing of the AL to the setting of AL, and from this instant to the instance of re-sensing of the AL respectively. The probability that station S_0 gains the mastership of the network (this is the first successful condition that was described in Section 4.4) is given as:

$$P = \text{Pr [success in gaining the mastership of the network after a contention period]}$$

$$dP_{s1}(t) = \text{Pr [exactly one arrival at } t_0 \in [t,t+dt] \text{ and no arrivals in the interval } [t,\ t+W_1+a];\ t \in [0,T]]$$

$$dP_{s1}(t) = \sum_{I=0}^{N-1} P_{k=0}(W_1+a)P_{k=1}(dt);$$ where I is the ID number* of the successful station

* Each station in the network has its own ID number for communication purposes.

$$= N(\frac{dt}{T}) \left\{ H(\frac{T-t-W_1-a}{T}) \right\}^{N-1}$$

where

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x > 0 \end{cases}$$

Therefore the probability of success in getting the mastership of the network after the first contention period is the sum of all the probabilities of success in getting the mastership of the network with $t \in [0,T]$. Thus the probability of getting a successful next TP (in the first situation) is given as:

$$P_{s1} = \int_0^T N(\frac{dt}{T}) \left\{ H(\frac{T-t-W_1-a}{T}) \right\}^{N-1}$$

$$= \frac{N}{T^N} \int_0^{T-W_1-a} dt \; (T-t-W_1-a)^{N-1}$$

$$= (\frac{T-W_1-a}{T})^N \qquad (4.7)$$

Similarly, in the second case, where two or more stations compete for the channel, denote by t_0 the time at which the first station arrives. Denote by t_i and t_{i-1} the times at which the last and the last but one station arrive. t_0, t_{i-1} are random variables that are uniformly distributed in interval [0,T] where T is the latency period (as was defined earlier). Fig. 4.6 shows the timing diagrams of these stations.

The probability of success in getting the mastership of the network after one contention period (station S_0 through S_{i-1} fail in getting the network, section 4.4 gives the

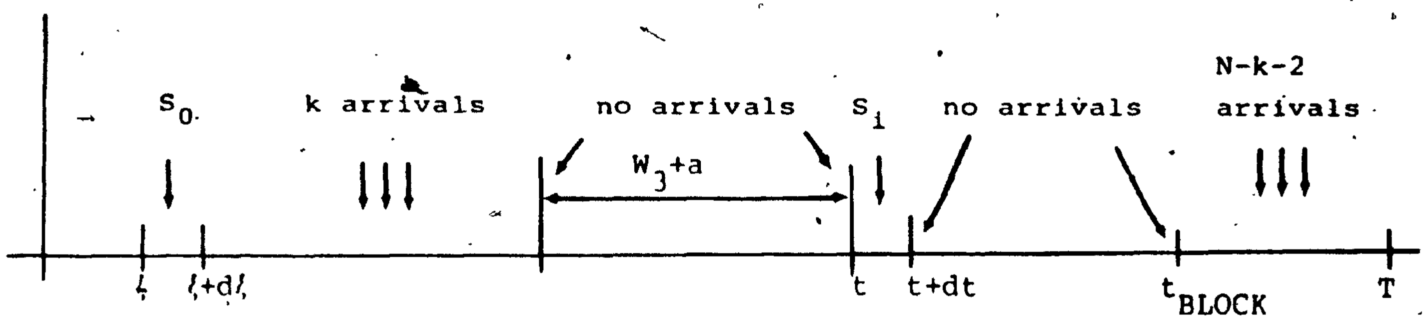


Fig. 4.6 Timing diagram showing the sucessful contention situation in case (a)

details, while station S_i will get the mastership of the network for the next TP) can be given as:

P = Pr [the last arriving station, S_i succeeds in acquiring the network for next TP after one contention period]

However, the calculation can be further divided into two cases:

a) $0 < t_0 < T - W_1 - a$

$$\begin{aligned} dP_{s2}(\xi,t) = \{Pr\ [&\text{exactly 1 arrival at } t_0 \in (\xi,\xi+d\xi) \\ &\text{exactly 1 arrival at } t_i\ (t,t+dt); \\ &\text{no arrivals at } [\xi+d\xi,\ t-W_3-a] \text{ and} \\ &\text{no arrivals at } [t+dt,\ t_{BLOCK}]; \\ &t_{BLOCK} > t > \xi + d\xi + W_3 + a;\ t_{BLOCK} = \\ &\xi + W_1 + a \text{ and } t \in [0,T],\ \xi \in [0,T-W_1-a]\} \end{aligned}$$

$$= \sum_{I=0}^{N-1} \{P_{k=1}(d\xi)\cdot P_{k=0}(W_3+a)\cdot P_{k=0}(t_{BLOCK}-t-dt) \cdot P_{k=1}(dt)\};$$

where I is the ID number of the successful station

$$=(N)\ \{P_{k=1}(d\xi)\cdot P_{k=0}(W_3+a)P_{k=0}(t_{BLOCK}-t-dt) \cdot P_{k=1}(dt)\}^{**} \tag{4.8}$$

Before we proceed any further, let us examine the range of values for t_i. The successful station S_i must not be blocked by S_0; thus, its arrival time t_i must be less than t_{BLOCK}, i.e.

**We assume that the transmitting station also competes for the network mastership for the next TP, thus a uniform distribution of N stations is attained.

$t_i < t_{BLOCK}$ or $t_i < t_0 + W_1 + a$; also it must satisfy condition (4.6) (refer to section 4.4) to get the mastership of the network, i.e. $t_i - t_{i-1} > W_3 + a$. Therefore, for $t_{i-1}=t_0$, the bounds of t_i with respect to t_0 are $t_0 + W_3 + a < t_i < t_0 + W_1 + a = t_{BLOCK}$. Hence, refer to Fig. 4.6, with $t_0 \in [\xi, \xi+d\xi]$, $t_i \in [t, t+dt]$; $dt, d\xi \to 0$, and rewriting eqution 4.8, we have

$$dP_{s2}(dt) = N \{P_{k=1}(d\xi)\cdot P_{k=0}(W_3+a)\cdot P_{k=0}(t_{BLOCK}-t-dt) \cdot P_{k=1}(dt)\}$$

As shown in Fig. 4.6, there are no arrivals in three intervals: $[0,\xi]$, $[t-W_3-a,t]$ and $[t+dt, t_{BLOCK}]$ for a successful contention period to occur. Since there are N stations competing for the channel, S_0 and S_i are the key stations in the contention. Therefore if there are k (where k=0, 1, 2,..., N-2) arrivals in the interval $[\xi+d\xi, t-W_3-a]$ then there must be (N-k-2) arrivals in the interval $[t_{BLOCK},T]$. Obviously,

$$dP_{s2}(\xi) =$$

$$N(\frac{d\xi}{T}) \int_{\xi+W_3+a}^{t_{BLOCK}} (N-1)\frac{dt}{T} \sum_{k=0}^{N-2} \binom{N-2}{k}\left(\frac{t-W_3-a-\xi}{T}\right)^K \binom{N-k-2}{N-k-2}\left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2}$$

$$=(\frac{N}{T})(\frac{d\xi}{T}) \int_{\xi+W_3+a}^{t_{BLOCK}} (N-1) \sum_{K=0}^{N-2} \frac{(N-2)!}{(N-2-k)!k!}\left(\frac{t-W_3-a-\xi}{T}\right)^K \left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2} dt$$

$$=\left(\frac{N}{T}\right)\left(\frac{d\xi}{T}\right)(N-1)\sum_{k=0}^{N-2}\frac{(N-2)!}{(N-2)!k!}\int_{\xi+W_3+a}^{t_{BLOCK}}\left(\frac{t-W_3-a-\xi}{T}\right)^{k}$$

$$\cdot\left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2} dt$$

$$=\frac{N(N-1)}{T}\left(\frac{d\xi}{T}\right)(N-2)!\sum_{K=0}^{N-2}\frac{1}{(N-2-k)!k!}\left(\frac{T}{k+1}\right)\left(\frac{t-W_3-a-\xi}{T}\right)^{k+1}$$

$$\cdot\left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2}\Bigg|_{\xi+W_3+a}^{t_{BLOCK}}$$

$t_{BLOCK} = \xi+W_1+a$, and therefore

$$dP_{s2}(d\xi)=\frac{N!}{T}\left(\frac{d\xi}{T}\right)\sum_{K=0}^{N-2}\frac{T}{(N-2-k)!(k+1)!}\left(\frac{\xi+W_1+a-W_3-a-\xi}{T}\right)^{k+1}$$

$$\cdot\left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2}$$

$$=\frac{N!}{T}\left(\frac{d\xi}{T}\right)\sum_{k=0}^{N-2}\frac{T}{(N-2-k)!(k+1)!}\left(\frac{W_1-W_3}{T}\right)^{k+1}\left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2}$$

Hence in the case of two or more ready stations competing for the network and with $0 < t_0 < T- W_1 - a$, the probability of success in getting the mastership of the network is the sum of all the probabilities with time stamp for the first arriving station $t_0 \in [\xi;\ \xi + d\xi]$, where $\xi \in [0,T]$.

Thus we have,

$$P_{s2} = \int dP_{s2}(d\xi)$$

$$= \int_0^{T-W_1-a} \frac{N!}{T}\left(\frac{d\xi}{T}\right) \sum_{k=0}^{N-2} \frac{T}{(N-2-k)!(k+1)!} \left\{H\left(\frac{T-t_{BLOCK}}{T}\right)\right\}^{N-k-2} \cdot \left(\frac{W_1-W_3}{T}\right)^{k+1} d\xi'$$

$$= \left(\frac{1}{T}\right) \frac{N!}{T} \sum_{k=0}^{N-2} \frac{T}{(N-2-k)!(k+1)!} \left(\frac{W_1-W_3}{T}\right)^{k+1} \cdot \int_0^{T-W_1-a} \left\{H\left(\frac{T-t_{BLOCK}}{T}\right)\right\}^{N-k-2} d\xi$$

$$= \frac{N!}{T} \sum_{k=0}^{N-2} \frac{1}{(N-2-k)!(k+1)!} \left(\frac{W_1-W_3}{T}\right)^{k+1} \cdot \int_0^{T-W_1-a} \left(\frac{T-t_{BLOCK}}{T}\right)^{N-k-2} d\xi$$

$$= \frac{N!}{T} \sum_{k=0}^{N-2} \frac{1}{(N-2-k)!(k+1)!} \left(\frac{W_1-W_3}{T}\right)^{k+1} \cdot \int_0^{T-W_1-a} \left(\frac{T-\xi-W_1-a}{T}\right)^{N-k-2} d\xi$$

$$= \frac{N!}{T} \sum_{K=0}^{N-2} \frac{1}{(N-2-k)!(k+1)!} \left(\frac{W_1-W_3}{T}\right)^{k+1} \left(\frac{T}{N-k-1}\right) \left(\frac{T-\xi-W_1-a}{T}\right)^{N-k-1} \Bigg|_{T-W_1-a}^{0}$$

Let $\ell = k + 1$

therefore $\begin{cases} K = 0 & \rightarrow \ \ell = 1 \\ K = N - 2 & \rightarrow \ \ell = N - 1 \end{cases}$

Hence we rewrite the expression

$$P_{s2} = \frac{N!}{T^N} \sum_{\ell=1}^{N-1} \frac{1}{(N-\ell)!\ell!} (W_1-W_3)^{\ell} (T-W_1-a)^{N-\ell}$$

$$= \frac{1}{T^N} \sum_{\ell=1}^{N-1} \binom{N}{\ell} (W_1-W_3)^{\ell} (T-W_1-a)^{N-\ell}$$

$$= \frac{1}{T^N} \left\{ \sum_{\ell=0}^{N} \binom{N}{\ell} (T-W_1-a)^{N-\ell} (W_1-W_3)^{\ell} - (T-W_1-a)^N - (W_1-W_3)^N \right\}$$

$$= \frac{1}{T^N} \left\{ (T-W_1-a+W_1-W_3)^N - (T-W_1-a)^N - (W_1-W_3)^N \right\}$$

$$= \frac{1}{T^N} \left\{ (T-W_3-a)^N - (T-W_1-a)^N - (W_1-W_3)^N \right\}$$

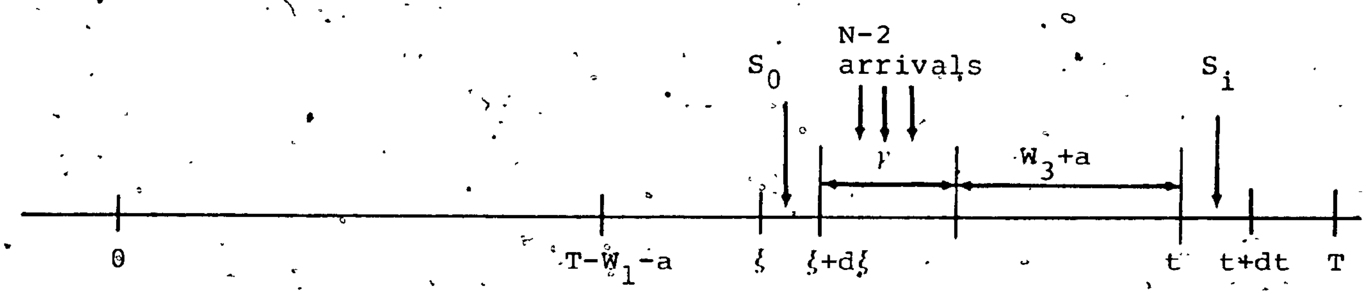
for further details, a derivation is given in Appendix D.

b) $T - W_1 - a < t_0 < T$

$$\begin{aligned} dP'_{s2}(\xi,t) = \sum \Pr \{ & \text{exactly 1 arrival at } t_0 \in (\xi,\xi+d\xi); \\ & \text{exactly 1 arrival at } t_i \in (t,t+dt); \\ & \text{no arrivals at } (\xi+d\xi,\ t-W_3-a) \text{ and} \\ & T > t > \xi + d\xi + W_3 + a;\ t \in [0,T]; \\ & \xi \in [T-W_1-a,\ T] \}^{***} \end{aligned}$$

Let us now determine the range of values for t_0 and t_i. Fig. 4.7 shows the situation for this case. Similarly, station ..

***Actually we should take care that there will be no arrivals after S_i, i.e. no arrivals in $[T, t_0+W_1+a]$. Based on our assumption on the uniform distribution in $[0,T]$, there are N arrivals only. If we slide the latency period, there should not be any arrivals in $[t_i, t_0+T]$, where $t_i<T$ and $W_1+a <T$, (i.e. $[T, t_0+ W_1+a]$ inclusive); otherwise this will violate our assumption.



$$\nu = t - w_3 - a - \xi$$

Fig. 4.7 Timing diagram showing the successful contention situation in case (b)

S_i must be arriving (W_3+a) seconds after S_0. There is no arrival after S_i, hence there must be (N-2) arrivals in $[\xi, t_i+W_3-a]$ (which is labeled as ν in Fig. 4.7). Thus with $t_0 \in [\xi, \xi+d\xi]$ and $t_i \in [t, t+dt]$, then $t \in [\xi + W_3 + a, T]$ and $\xi \in [T - W_1 - a, T - W_3 - a]$, therefore

$$dP'_{s2}(\xi) = N\left(\frac{d\xi}{T}\right)(N-1) \int_{\xi+W_3+a}^{T} \left(\frac{\nu}{T}\right)^{N-2} \left(\frac{dt}{T}\right)$$

$$= N(N-1)\left(\frac{d\xi}{T}\right)\left(\frac{1}{T}\right) \int_{\xi+W_3+a}^{T} \left(\frac{t-W_3-a-\xi}{T}\right)^{N-2} dt$$

$$= N(N-1)\left(\frac{d\xi}{T}\right)\left(\frac{1}{T}\right)\left(\frac{T}{N-1}\right)\left(\frac{t-W_3-a-\xi}{T}\right)^{N-1} \Bigg|_{\xi+W_3+a}^{T}$$

$$= N\left(\frac{d\xi}{T}\right)\left(\frac{T-W_3-a-\xi}{T}\right)^{N-1}$$

and

$$P'_{s2} = \frac{N}{T} \int_{T-W_1-a}^{T-W_3-a} \left(\frac{T-W_3-a-\xi}{T}\right)^{N-1} d\xi$$

$$= \frac{N}{T} \left(\frac{T-W_3-a-\xi}{T}\right)^{N} \left(\frac{T}{N}\right) \Bigg|_{T-W_3-a}^{T-W_1-a}$$

$$= \left(\frac{W_1-W_3}{T}\right)^{N}$$

The events in the first case as described earlier in this section and the second case just described above are mutually exclusive. Therefore, we can find the total probability of success in joining the mastership of the network after one contention period as the sum of probabilities of the two events.

i.e. $P_s = P_{s1} + P_{s2} + P'_{s2}$

$$= \left(\frac{T-W_1-a}{T}\right)^N + \frac{1}{T^N}\left\{(T-W_3-a)^N-(T-W_1-a)^N-(W_1-W_3)^N\right\}$$

$$+ \left(\frac{W_1-W_3}{T}\right)^N$$

$$= \left(\frac{T-W_3-a}{T}\right)^N \tag{4.9}$$

The probability of failure of any station in gaining the mastership of the channel after the contention period will be

P_f = Pr [no station gains the mastership of the bus after the contention period]

$= 1 - P_s$

In section 4.1, we have pointed out that an idle period may exist if the total contention period is longer than the transmission period. Our protocol assumes packets of fixed length (~ 128 bytes) at a constant data rate (~ 7M bytes/-sec); thus the transmission period is assumed to be constant and symbolized by τ (~ 30 μsec). As the contention for the mastership of the network for the next TP involves only the AL while the actual transmission takes place on the high speed H-BUS, these two events proceed concurrently; if it happens that the physical transmission has finished, yet there is no station succeeding in gaining the mastership of the network, then the H-BUS will be idle until one station gets control of it. Fig. 4.8 shows the timing diagram for this situation.

Suppose a ready station has gained the network mastership and it has entered the ready-wait state. Once the transmitting station has finished the transmission, this ready-and-waiting station will acquire the H-BUS immediately. Then it will commence its transmission (this is the period τ_2 as shown in Fig. 4.8), and at the same time, it will release the AL. This is to let the remaining stations compete for the mastership of the network for the next TP on the AL. However, if there are many consecutive unsuccessful contentions and the total contention period is longer than the transmission period, there will be a time interval during which all the stations will be competing for the network and no one will be transmitting, an idle period occurs. Note that once a successful contention period exists, then contentions are stopped until the next transmission has been initiated (and the AL will be released by the transmitting station). We proceed now to calculate the average length of the idle period.

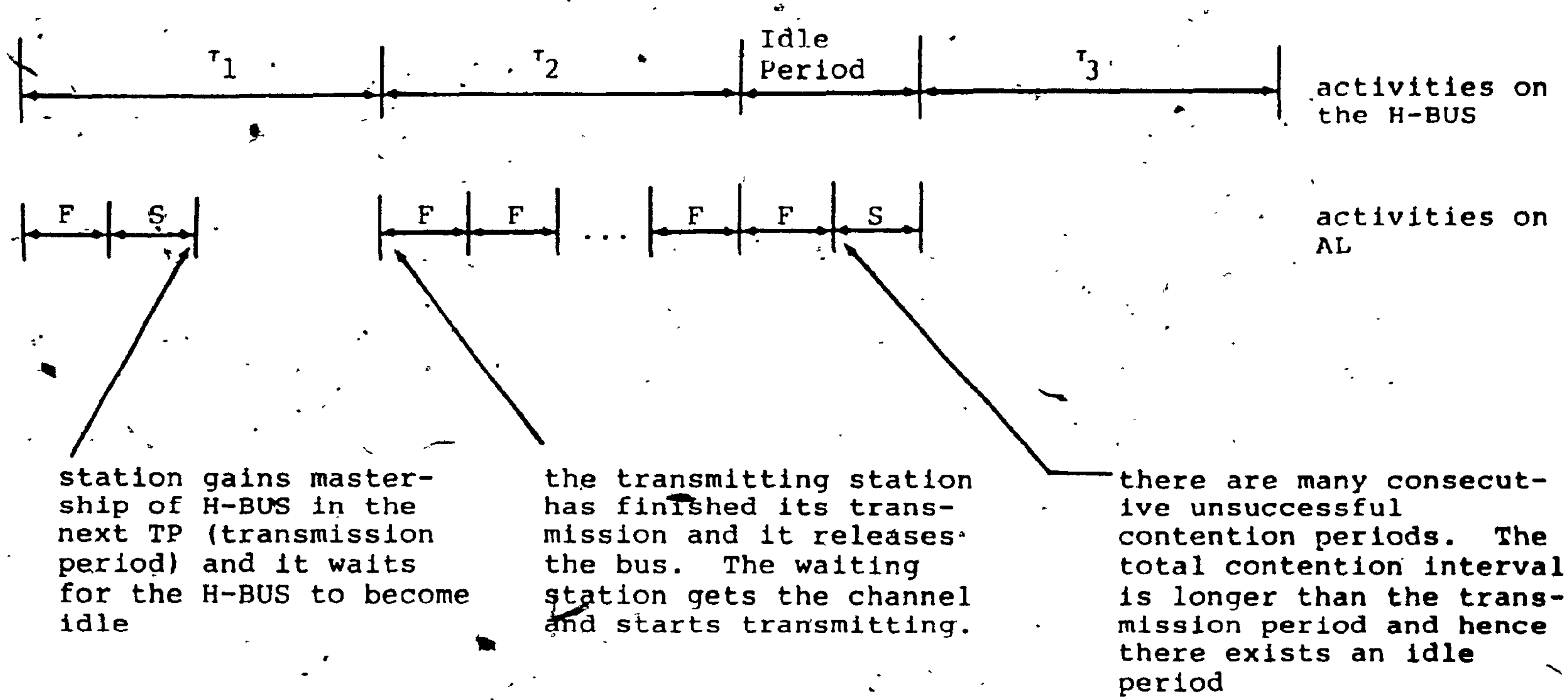


Fig. 4.8. Diagram showing the situation for the existence of an idle period

An idle period occurs when the length of the total contention period is longer than that of the transmission period; the length of the idle period then is given as the difference of the total contention period minus the transmission period. A total contention consists of (n-1) consecutive unsuccessful contention periods followed by one successful contention. The length of a contention period is a random variable which depends on the arrival of a ready station. Denote by x_i, $i = 1, 2, \ldots, n-1$ the length of the i^{th} unsuccessful contention period, and by y the length of the last successful contention period. Denote by C the length of the total contention period

i.e. C = total unsuccessful contention period + the successful contention period

$$= \sum_{i=1}^{n-1} x_i + y$$

The probability of existence of the idle period is given as

$$P = \Pr\,[\text{an idle period exist}]$$

$$= \Pr\,[\text{the total contention period} > \text{the transmission period}]$$

$$= P(C > \tau)$$

$$= P(n=1)P(y>\tau \mid n=1)+P(n=2)P(x_1+y>\tau) \mid n=2)$$

$$+\ldots+ P(n=k)P(x_1+x_2+\ldots+x_K+y>\tau \mid n=k)$$

$$= \sum_{K=1}^{B} P(n=k)P(\sum_{i=1}^{n-1} x_i+y>\tau \mid n=k) \text{ where } B \to \infty \qquad (4.10)$$

obviously,

$$P\left(\sum_{i=1}^{n-1} x_i + y > \tau \,\middle|\, n=k\right) < 1 \quad \text{and}$$

$$P = P(C > \tau) < \sum_{k=1}^{B} P(n=k); \; B \to \infty$$

Since $P(n=k) = P_f^{k-1} \cdot P_s$ and P_f, $P_s < 1$ then $\sum_{k=1}^{B} P(n=k) =$

$P_s \left(\frac{1}{1-P_f}\right) = 1; \quad B \to \infty$

hence

$$P = P(C > \tau) < 1$$

Since x_i's and y are determined by the arrival time of a station, they are defined by

$$x_i = t + W_1 + W_2 + W_3 + a$$

$$y = t + W_1 + W_2$$

where t is the arrival time of the ready station, and W_1, W_2, W_3 as well as a are constants. The arrival time, t, of the station is a random variable which is uniformly distributed in [0,T]; where T is the period of latency. Thus there exist a global maximum value of x_i (when t = T) and also a global minimum value of x_i (when t = 0) where

$$x_{min} = W_1 + W_2 + W_3 + a$$

$$x_{max} = T + W_1 + W_2 + W_3 + a$$

Similarly, the random variable y attains its minimum and maximum at

$$y_{min} = W_1 + W_2$$

$$y_{max} = T + W_1 + W_2$$

We can see that the distribution of y and x_i's differ by a constant $(W_3 + a)$. Let us assume the occurrence of the following special case: each of the total contention periods attains their maximum (i.e. x_{max} and y_{max}), then the number of contention periods such that an idle period just exists will be the minimum.

In other words, there should be a particular value of n (the number of contention periods), denoted as n_{min} such that if $n < n_{min}$, then there will be no idle period.

Now with the transmission period, τ, n_{min} can be easily determined by

$$n_{min} = [\frac{\tau - y_{max}}{x_{max}}] + 1 \qquad (4.11)$$

Where [g] is the largest integer that is less than or equal to the value of g. Rewriting equation (4.10), we have

$$\Pr\ [\text{an idle period exist}] = \sum_{k=1}^{B} P(n=k) P(\sum_{i=1}^{k-1} x_i + y > \tau \mid n=k)$$

with $B \to \infty$

From the analysis above, obviously

$$P(\sum_{i=1}^{k-1} x_i + y > \tau \mid n=k) = 0 \quad \text{for } k < n_{min}$$

Hence we can deduce the following

$$\Pr\ [\text{an idle period exist}] = \sum_{k=n_{min}}^{B} P(n=k) P(\sum_{i=1}^{k-1} x_i + y > \tau \mid n=k) \qquad (4.12)$$

Consider the expression in equation (4.12), $P(n=k)$ is the probability that k contention periods occured. This is quite easy to find as we have already found the probability of one contention period. Besides, the event that a second unsuccessful contention period is independent of the previous contention periods. So $P(n=k)$, the probability that there are $(k-1)$ unsuccessful contentions followed by a successful one, is a geometrical distribution with $P(n=k)=P_f^{k-1} \cdot P_s$ where P_f and P_s are the probabilities of an unsuccessful and successful period respectively.

For $P(\sum_{i=1}^{n-1} x_i+y > \tau \mid n=k)$, we have the following: x_i's and y are continuous random variables with $x_i \in [x_{min}, x_{max}]$ and $y \in [y_{min}, y_{max}]$. Yet they are independent because the length of a contention interval is independent of the length of the previous contention period(s). Now with

$$x_i = t + W_1 + W_2 + W_3 + a$$

$$y = t + W_1 + W_2$$

where t is a uniformly distributed random variable in the interval [0,T]

the expected mean and variance of x_i and y are (refer to Appendix E for details).

$$\eta_i = E[x_i] = \frac{T}{2} + W_1 + W_2 + W_3 + a$$

$$\sigma_i^2 = \sigma_{x_i}^2 = \frac{T^2}{12}$$

$$\eta_y = E[y] = \frac{T}{2} + W_1 + W_2$$
$$\sigma_y^2 = \sigma_{x_i}^2 = \frac{T^2}{12} \qquad (4.13)$$

By means of the Central Limit Theorem [4.24] with $z = \sum_{i=1}^{n-1} x_i + y$, then its mean η and variance σ^2 are given by

$$\eta = \eta_1 + \eta_2 + \ldots + \eta_{n-1} + \eta_y = \sum_{i=1}^{n-1} \eta_i + \eta_y$$

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_{n-1}^2 + \sigma_y^2 = \sum_{i=1}^{n-1} \sigma_1^2 + \sigma_y^2$$

and its density cna be approximated by

$$f(z) \simeq \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\eta)^2/2\sigma^2}$$

when n increases.

For the expression,

$$P(\sum_{i=1}^{n-1} x_i + y > \tau \,|n=k),$$

we can find the corresponding probability by

$$P(\sum_{i=1}^{n-1} x_i + y > \tau \,|n=k) = \int_\tau^\infty f(z)\, dz$$
$$\simeq \int_\tau^\infty \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\eta)^2/2\sigma^2} dz \qquad (4.14)$$

when n is sufficiently large.

In the following, let us assume that n is sufficiently large such that the Central Limit Theorem can be applied. Therefore, using equation (4.13)

$$\eta = \sum_{i=1}^{n=1} \eta_i + \eta_y$$

$$= (n-1)\ \eta_i + \eta_y$$

$$= n(\frac{T}{2} + W_1 + W_2) + (n-1)(W_3 + a) \qquad (4.15)$$

$$\sigma^2 = \sum_{i=1}^{n-1} \sigma_i^2 + \sigma_y^2$$

$$= \frac{nT^2}{12} \qquad (4.16)$$

hence with equation (4.14), we have

$$P = \Pr\ [\text{an idle period exists}]$$

$$= \sum_{k=n_{min}}^{\infty} P(n=k) P(\sum_{i=1}^{n=1} x_i + y > \tau \mid n=k)$$

$$= \sum_{k=n_{min}}^{\infty} (P_f^{k-1} \cdot P_s)(\frac{1}{\sigma\sqrt{2\pi}} \int_{\tau}^{\infty} e^{-(z-\eta)^2/2\sigma^2}\ dz)$$

$$= \frac{1}{2} \sum_{n_{min}}^{\infty} (P_f^{k-1} \cdot P_s)\ \text{erfc}\ (\frac{\tau-\eta}{\sqrt{2}\sigma}) \qquad (4.17)$$

where $\text{erfc}\ (\alpha) = \frac{2}{\sqrt{\pi}} \int_{\alpha}^{\infty} e^{-t^2}\ dt$

For the average idle period, we proceed with the following analysis. Given that there are n=k contention periods that occured, the average idle period is a conditional expectation which is given as

$$
\begin{aligned}
I_k &= E[I \mid n=k] \\
&= \int_{\tau}^{\infty} (z-\tau)\, P(C=z>\tau \mid n=k)\, dz \\
&= \int_{\tau}^{\infty} (z-\tau)\, \frac{1}{\sigma\sqrt{2\pi}}\, e^{-(z-\eta)^2/2\sigma^2}\, dz \\
&= \frac{\sigma}{\sqrt{2\pi}}\, e^{-(\tau-\eta)^2/2\sigma^2} + \frac{\eta-\tau}{2}\, \mathrm{erfc}\left(\frac{\tau-\eta}{\sqrt{2}\sigma}\right)
\end{aligned}
\tag{4.18}
$$

where $\mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_{x}^{\infty} e^{-t^2}\, dt$

The details are given in Appendix F. The average idle period is given as

$$
\begin{aligned}
\bar{I} &= E[I] \\
&= E[E[I \mid n=k]] \\
&= E[I_k] \\
&= \sum_{k=n_{min}}^{\infty} P(n=k) \cdot I_k \\
&= \sum_{k=n_{min}}^{\infty} P_f^{k-1} P_s \cdot I_k
\end{aligned}
\tag{4.19}
$$

Thus the average idle period is an infinite sum of series, if there exist an upper bound such that this bound is finite, then the average idle period exists and converges. The following is to find the existence of such a bound.

Now

$$\left|e^{-t^2}\right| < 1; \quad \text{for } -\infty < t < \infty$$

and

$$\frac{1}{2}\,\mathrm{erfc}(t) < 1; \quad \text{for } -\infty < y < \infty$$

hence by using equation (4.18) we have,

$$I_k = \frac{\sigma}{\sqrt{2\pi}}\, e^{-(\tau-\eta)^2/2\sigma^2} + \frac{\eta-\tau}{2}\,\mathrm{erfc}\,\left(\frac{\tau-\eta}{\sqrt{2}\sigma}\right)$$

$$< \frac{\sigma}{\sqrt{2\pi}} + \eta - \tau.$$

Now from (4.15) and (4.16)

$$B_k = \sqrt{\frac{k}{24\pi}}\,T + \left(\frac{T}{2} + W_1 + W_2 + W_3 + a\right) k$$
$$- (W_3 + a)$$

for $k > 1$, $k^2 > k$ or $k > \sqrt{k}$ and thus

$$B_k < \frac{kT}{\sqrt{24\pi}} + \left(\frac{T}{2} + W_1 + W_2 + W_3 + a\right) k$$
$$- (W_3 + a)$$
$$< C_1\{k - C_2\}$$

where the constants C_1 and C_2 depend on T, W_1, W_2, W_3 and a.

Hence,

$$I_k < B_k < C_1\{k - C_2\} \qquad \text{for some constants } C_1,\ C_2$$

from (4.19) therefore,

$$\bar{I} < \sum_{k=n_{min}}^{\infty} P_f^{k-1} \cdot P_s \cdot C_1 \{k - C_2\}$$

i.e. $$\bar{I} < C_1 \sum_{k=n_{min}}^{\infty} \{k\, P_f^{k-1} \cdot P_s - C_2\, P_f^{k-1} \cdot P_s\}$$

$$= C_1 \sum_{k=n_{min}}^{\infty} \{P_s P_f \frac{\partial}{\partial P_f} (P_f^{k})\} - C_1 C_2 \cdot P_s \sum_{k=n_{min}}^{\infty} P_f^{k-1}$$

$$= C_1 P_s \sum_{k=n_{min}}^{\infty} \frac{\partial}{\partial P_f} (P_f^{k}) - C_1 C_2 P_s \frac{P_f^{n_{min}-1}}{1 - P_f}$$

$$= C_1 P_s \frac{\partial}{\partial P_f} \left(\sum_{k=n_{min}}^{\infty} P_f^{k} \right) - C_1 C_2 P_s \frac{P_f^{n_{min}-1}}{1 - P_f}$$

$$= C_1 P_s \frac{\partial}{\partial P_f} \frac{P_f^{n_{min}}}{(1-P_f)} - C_1 C_2 P_f^{n_{min}-1}$$

$$= C_1 P_s \left(\frac{n_{min} P_f^{n_{min}-1} (1-P_f) + P_f^{n_{min}}}{(1 - P_f)^2} \right) - C_1 C_2 P_f^{n_{min}-1}$$

$= A_p(P_f)$; where A_p is a function of P_f

$\therefore \bar{I} < A_p$ for $k > 1$

for P_f different from one, the function $A_p(P_f)$ is real and finite. Now since the number of contentions is at least one, then obviously $\bar{I}$ is bounded and thus we can conclude that the average idle period exists and converges. The following is to calculate the average idle period by using the system parameters.

In our existing system, we assume a transmission period $\tau = 30\ \mu s$, latency period $T = 1\mu s$, the propagation delay $a =$ 100ns and probability of a successful contention period, which was given in equation (4.9) and rewritten again,

$$P_s = \left(\frac{T-W_3-a}{T}\right)^N ,$$ where N is the total number of stations in the system.

Obviously, the optimal probability of a successful contention period is when $W_3=0$. From equation (4.1), we have

$$W_2 > 2a + W_1$$

let $W_1 = a$,

hence $W_2 > 2a + W_1 = 3a$

let $W_2 = 3a$

thus, using equation (4.13)

$$x_{min} = W_1 + W_2 + W_3 + a \qquad = 500 \text{ ns}$$

$$x_{max} = T + W_1 + W_2 + W_3 + a = 1500 \text{ ns}$$

$$y_{min} = W_1 + W_2 \qquad = 400 \text{ ns}$$

$$y_{max} = T + W_1 + W_3 \qquad = 1400 \text{ ns}$$

By means of equation (4.11), we have

$$n_{min} = [\frac{\tau - y_{max}}{x_{max}}] + 1$$

$$= [\frac{30 - 1.4}{1.5}] + 1$$

$$= 20$$

Since $n_{min} = 20$ it is large enough for us to employ the Central Limit Theorem, therefore from equations (4.15) and (4.16), we have

$$\eta = n(\frac{T}{2} + W_1 + W_2) + (n-1)(W_3 + a)$$

$$= n - 0.1$$

$$\sigma^2 = \frac{nT^2}{12}$$

$$= \frac{n}{12}$$

Using equations (4.18) and (4.19), the average idle $\bar{I}$ period can be evaluated.

Now the system utilization factor is given by

$$\bar{S} = \frac{\bar{U}}{\bar{I} + \bar{B}}$$

where

$\bar{U}$ is the average successful transmission period;

$\bar{I}$ is the average idle period;

and $\bar{B}$ is the average busy period.

Since there are no collision periods, the average busy period equals the average transmission period. Thus

$\bar{B} = \bar{U} = \tau$ (30μs); hence

$$\bar{S} = \frac{\bar{U}}{\bar{I} + \bar{U}} \tag{4.20}$$

By using (4.18), (4.19) and (4.20) the corresponding system utilization factor is calculated. This is done by means of the computer program attached in Appendix F. In Fig. 4.9, a plot of the average system utilization factor together with the simulated results are given. Fig. 4.10 shows the average delay response of the New Protocol. The simulation model of this new protocol is run by PAWS again. The description of the IPG model is given in the next section.

4.6 Performance Evaluation on Collision Free Protocol through Simulation

In this section, a performance evaluation on the New Collision Free Protocol is again modeled by using PAWS. The simulated results are compared with the theoretical ones obtained in Section 4.5. Due to the current limitation on PAWS, only up to twenty stations could be simulated. In order to verify the theoretical results,a second set of parameters using a = 200 ns, T = 1μs, W_1 = 200 ns, W_2 = 600 ns and W_3 = 0 ns is also run.

Fig. 4.11 shows the IPG model of the New Collision Free Protocol. Packets are generated at source node MSOURCE in order to fill up the queue at all Stations [I].

The node WAITGEN generates the random arrival time of the packet from a particular station. The nodes W1 and W2 represent the waiting times. On the other hand, WINDOW1

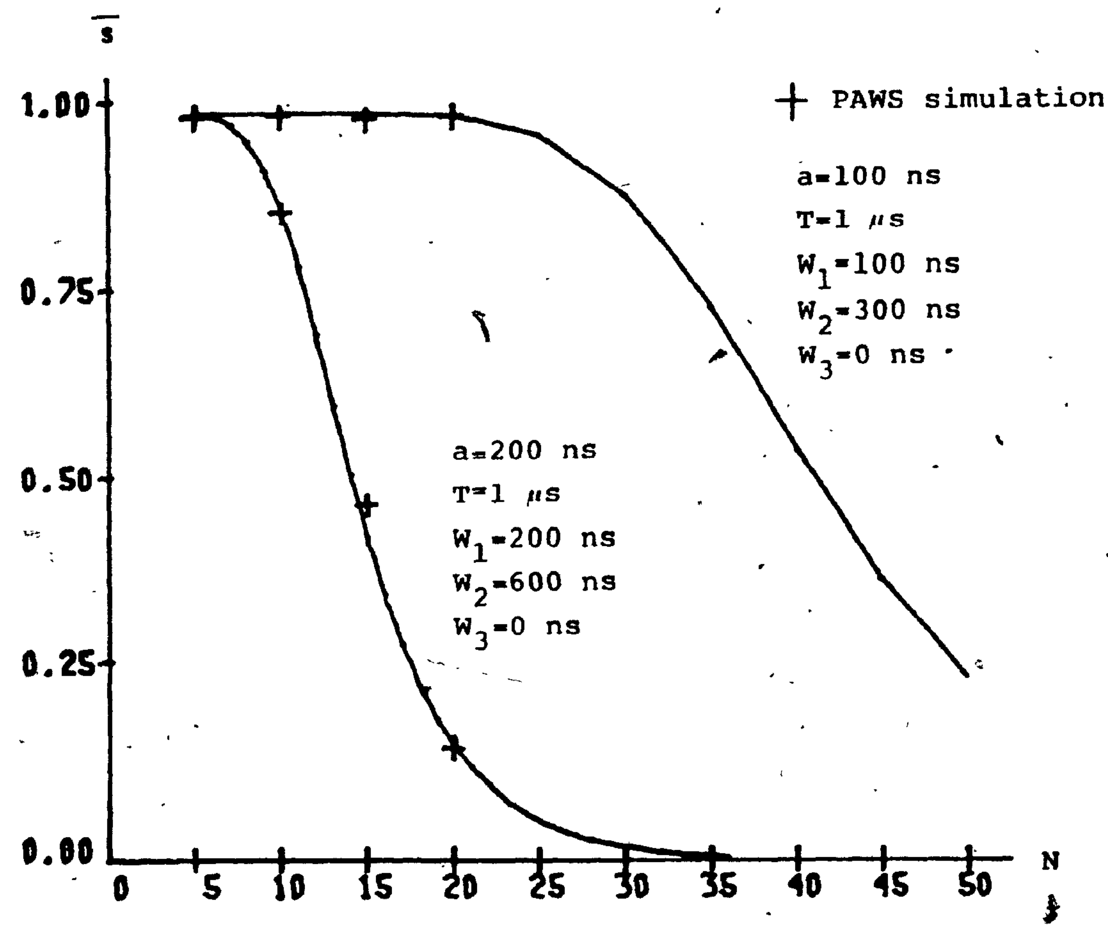


Figure 4.9 Utilization factors of the new Collision Free Protocol vs number of stations, N.

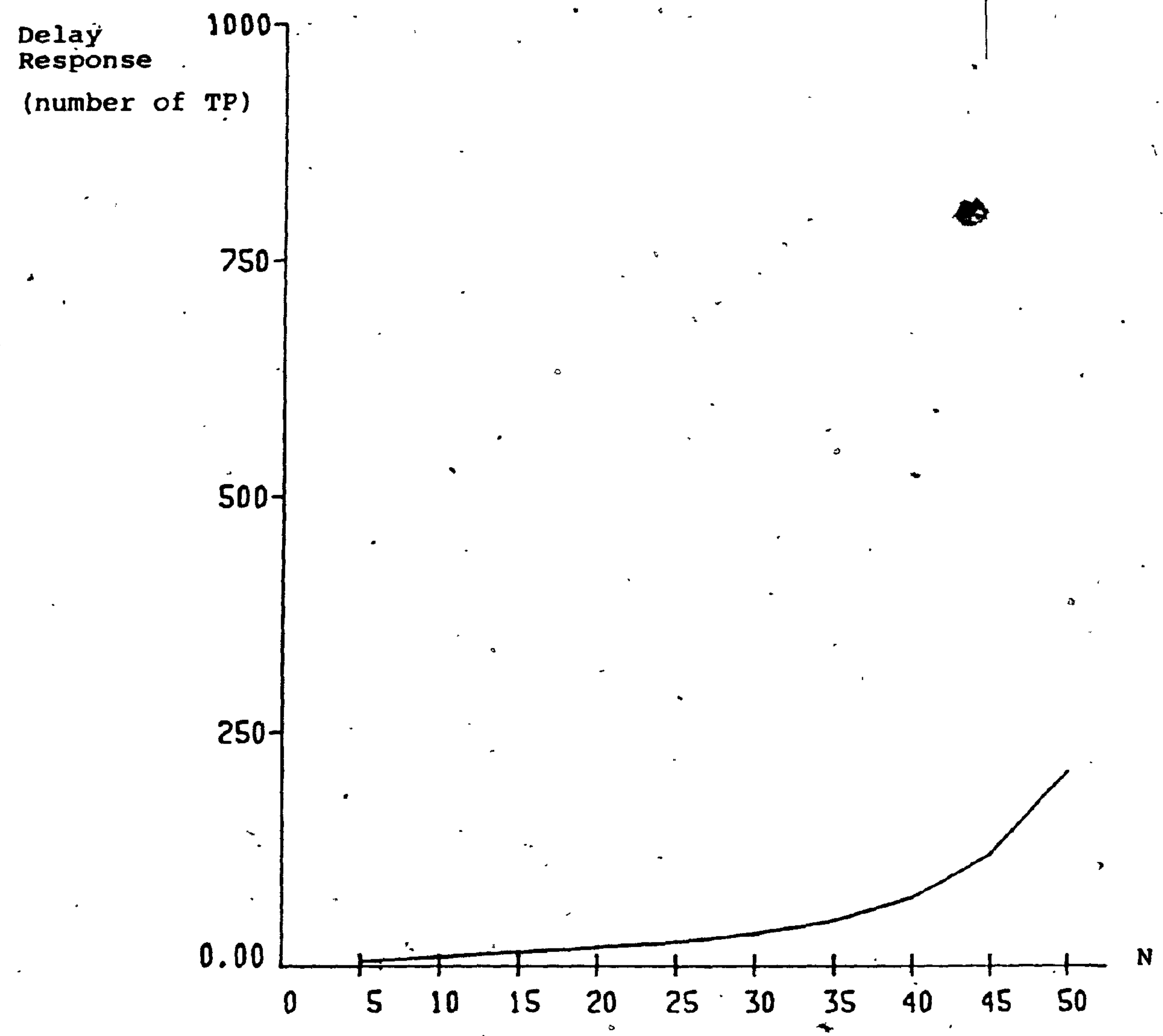


Fig. 4.10 The Delay Response of the New Collision Free Protocol

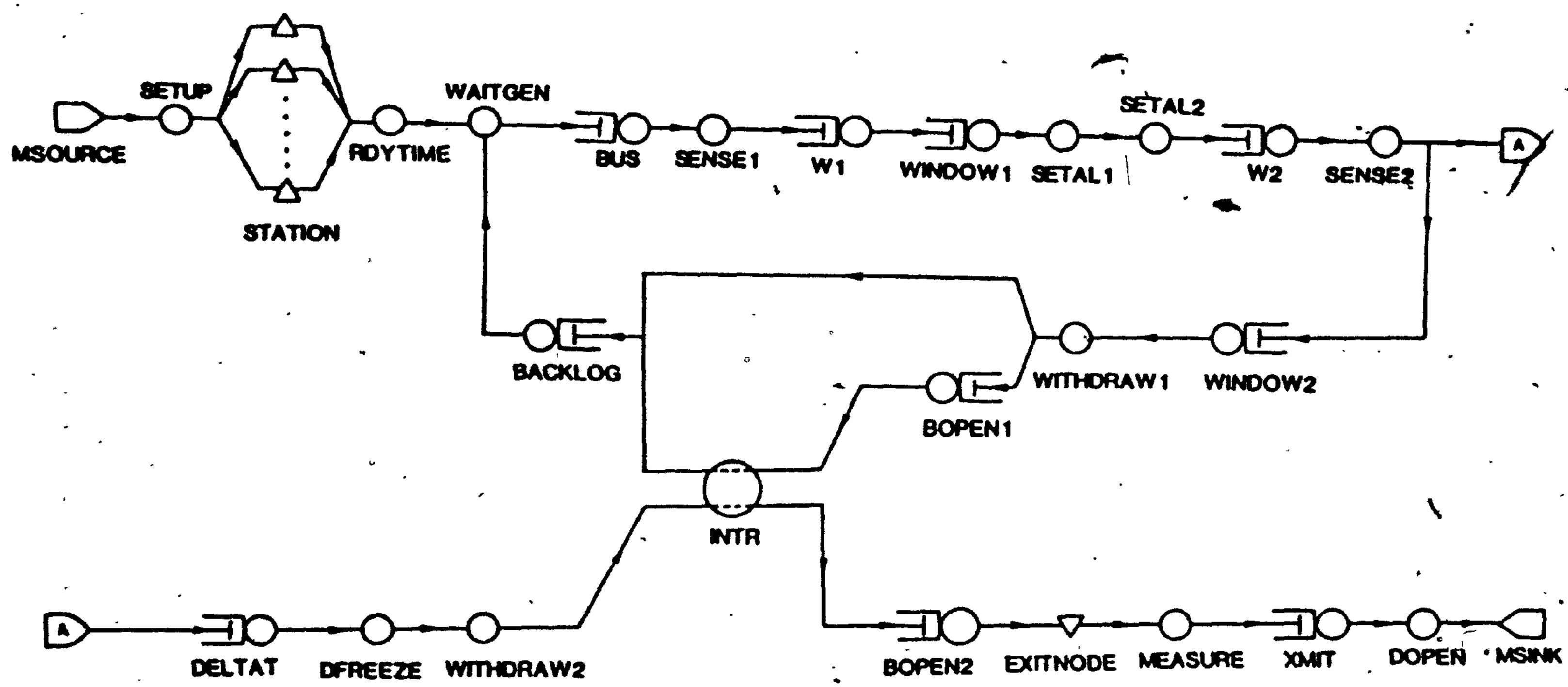


Fig. 4.11 IPG simulation model of the new Collision-Free Protocol

and WINDOW2 are equivalent to the network propagation delay. Nodes SENSE1 and SENSE2 are used for testing the pebble pool while SETAL1 and SETAL2 are to close the BACKLOG gate and to lay the pebble in the pool respectively. The node WITHDRAW is to simulate the unsuccessful stations at which their pebbles are retrieved. After then these packets will go to the gate BACKLOG.

On the other hand, the successful packet will go to gate DELTAT and wait for the current transmission, which occurs at node XMIT, to be transmitted. When a transmission finishes, the gate of DELTAT will be open, the packet that has been waiting will leave this node and close the gate right away. The purpose is to avoid two packet overlapping. Before this successful packet starts transmitting, it will retrieve the pebble at node WITHDRAW2. A new contention is triggered by opening the gate at node BACKLOG which is controlled at node BOPEN2. Another node BOPEN1 also controls the gate BACKLOG. This is when no station wins the contentions, all the packets will be waiting at node BACKLOG until the last unsuccessful station opens the gate and starts another new contention. The node INTR is to flush the packets that are left at the node BUS so that there are always N stations competing in any contention. The corresponding PAWS simulation program is given in Appendix G.

4.7 Discussion

As we can see from Fig. 4.9, the theoretical results agree closely with the simulated ones. This reflects the validity of the collision Free Protocol. The system utilization factor is relatively constant at one and starts cutoff at N=30. This shows the new collision Free Protocol improves a great extent in the system utilization factor when compared to the results obtained in Chapter 2. More discussions will be found in Chapter 5.

Similarly, we can see the delay response is much better in this new protocol than the old schemes. In Fig. 4.10, the plot shows that the delay time gets significant when N=30.

CHAPTER 5

Summary and Discussion

In this thesis, we have introduced a New Collision Free Protocol in which has a contention period coincides with the previous packet transmission. We have also proved that this protocol is capable of filtering out collisions. A detailed analysis is done to evaluate the corresponding system utilization factor and the results were shown in Fig. 4.9 and Fig. 4.10. In addition, two sets of parameters were calculated and further verified by using the PAWS simulation package.

Moreover, a hardware implementation is proposed for the H-Network to utilize such a Collision Free Protocol. In the future, our intention is to build the network and verify the protocol. Presently, the theoretical results are promising and they have shown a tremendous improvement over the standard CSMA/CD. The system utilization factors as well as the delay responses of the original H-Netowrk CSMA/CD protocol (Fig. 2.4, Fig. 2.5) and the new Collision Free Protocol (Fig. 4.9, Fig. 4.10) are re-plotted in Fig. 5.1 and Fig. 5.2 for comparison.

Obviously, in Fig. 5.1, the average utilization of the New Collision Free Protocol is far better off than the old protocol.

In the Collision Free Protocol, the utilization factor maintains a relatively high efficeincy (~ 100%) up to 30

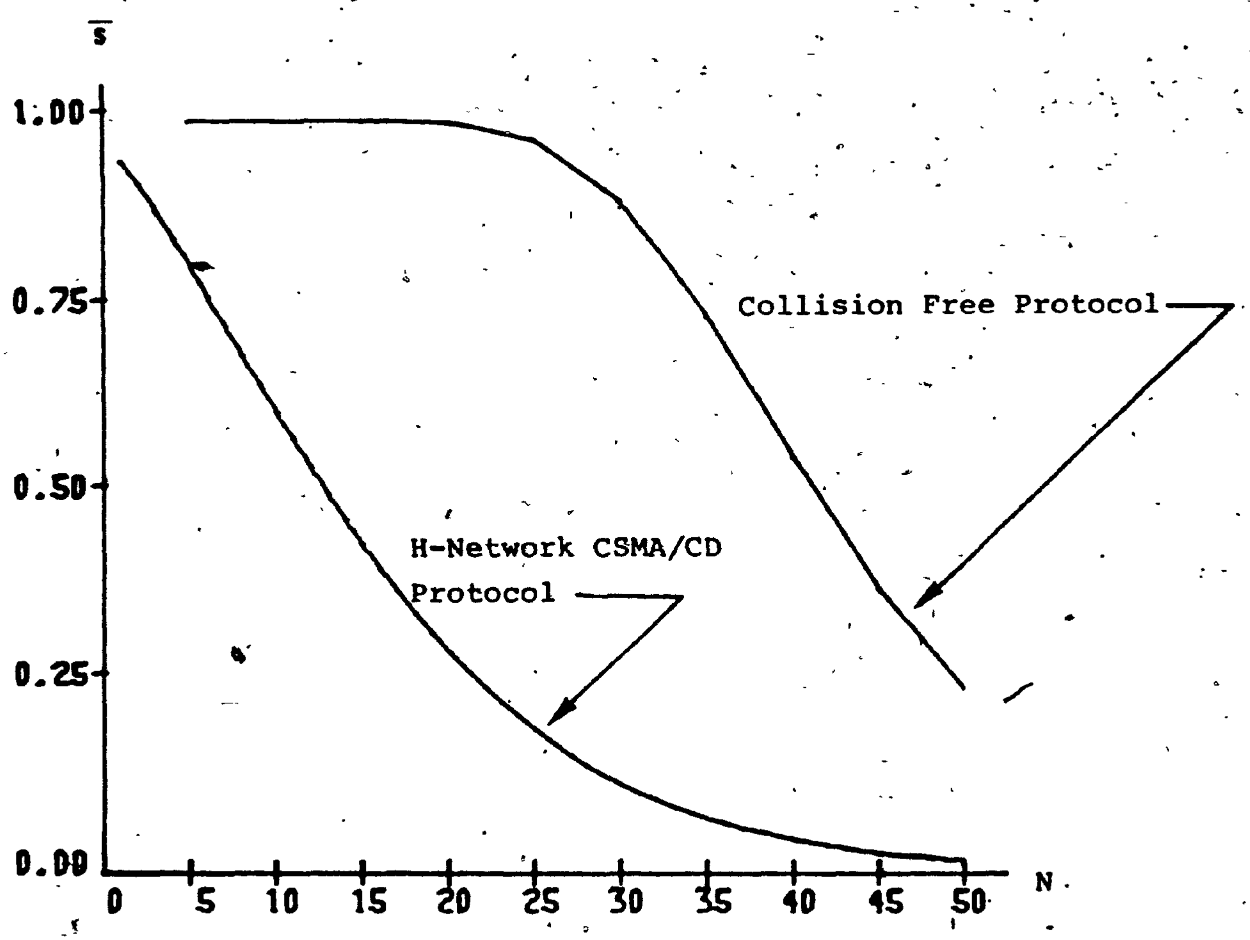


Figure 5.1 Comparison of the system utilization factors between the H-Network CSMA/CD Protocol and the new Collision Free Protocol vs number of stations, N.

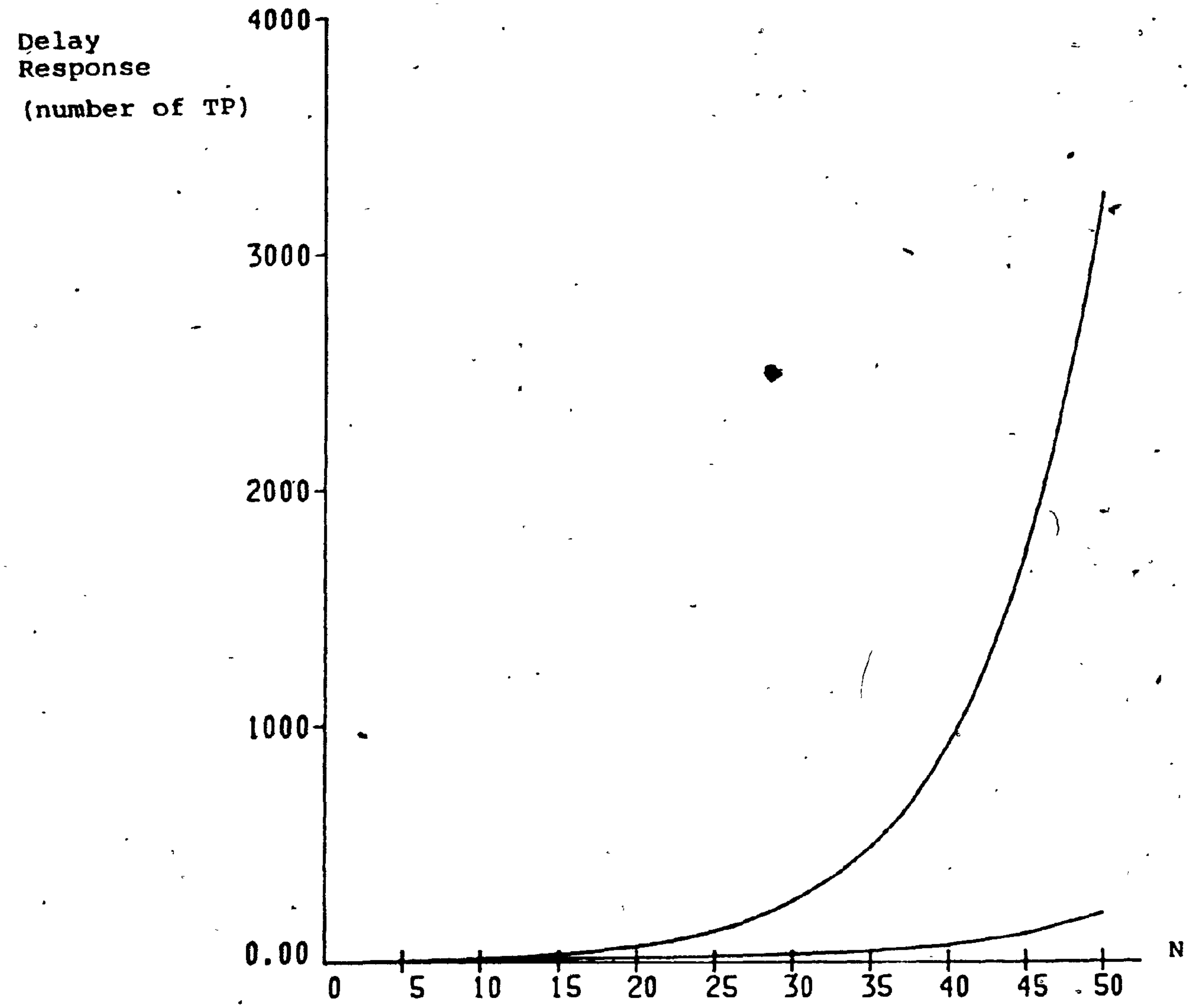


Fig. 5.2 Comparison on the average Delay Response between the original H-Network Protocol and the New Collision Free Protocol

stations. Afterwards, it drops gradually as the existence of the idle period become more probable. On the other hand, the old protocol experiences many collisions when the traffic starts getting heavy (when N > 5), and the system utilization falls exponentially due to collisions.

Also, the new protocol has a much smaller packet delay as compared to the old protocol.

To conclude, we developed an effective network contention protocol and provided the link level and transport level for the H-Network.

REFERENCES

[1] Abramson, N., "The Aloha System", AFIPS Conf. Proc., Vol. 37, pp. 281-285, FJCC, 1970.

[2] Ahmad, M.O., et al., "Ladder Realization of a Class of Two-dimensional Analog Voltage Transfer Functions with Application to Wave Digital Filters", Archiv for Electronik and Ubertrangungstechnik, Vol. 33, No. 2, pp. 81-85, 1979.

[3] Andrews, D.W. and G.D. Shultz, "A Token-Ring Architecture for Local-Area Networks: an update," IBM Corporation Communication Products Division, Research Triangle Part, North Carolina, U.S.A.

[4] Batcher, K.E., "The Flip Network in STARAN," Proc. 1976 Int'l Conf. Parallel Processing, pp. 65-71, Aug. 1976.

[5] Berry, R. and et. al., "PAWS 2.0 Performance Analyst's Workbench System," Information Research Associates, Austin, Texas, 1982.

[6] Bonknight, W.J. and et. al., "The Illiac IV System," Proc. IEEE, pp. 369-388, April 1972.

[7] Data Communication, pp. 61-79, McGraw Hill Inc., 1980.

[8] Dennis, J.B., "Data Flow Supercomputers," Computer, Vol. 13, No. 11, pp. 48-56, Nov. 1980.

[9] Dimopoulos, N., "Organization and Stability of a Neural Network Class and the Structure of a Multiprocessor System," Ph.D. thesis, University of Maryland, 1980.

[10] Dimopoulos, N., "The Homogeneous Multiprocessor Architecture - Structure and Performance Analysis," Proc. of the 1983 Int'l Conf. on Parallel Processing, pp. 520-523, August, 1983.

[11] Dimopoulos, N. and D. Kehayas, "The H-Network A High Speed Distributed Packet Switching Local Computer Network," Proc. on MELECON'83 Mediterranean Electrotechnical Conf. Athens, Greece, p. A01,02, May, 1983.

[12] Dimopoulos, N., "On the Structure of the Homogeneous Multiprocessor," IEEE Trans. on Computers, Vol. C-34, No. 2, pp. 141-150, Feb. 1985.

[13] Dimopoulos, N., and C.W. Wong, "Performance Evaluation of the H-Network Through Simulation," European Meeting, Digitech, July 1984.

[14] Dimopoulos, N., K.L. Li, and C.W. Wong, "Simulation and Performance of the Homogeneous Multiprocessor," Summer Computer Simulation Conf., Boston, M.A., July 1984.

[15] Kehayas, D., "The H-Network - A Packet Switching Local Area Network," Major Technical Report, Master of Electrical Engineering, Concordia University, 1983.

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

[17] Kleinrock, L. and F.A. Tobagi, "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, pp. 1417-1433, Dec. 1975.

[18] Kleinrock, L., "Queueing Systems Volume II: Computer Application," Wiley Interscience, pp. 301-302, 1976.

[19] Kleinrock, L. and F.A. Tobagi, "Packet Switching in Radio Channels: Part IV - Stability Considerations and Dynamic Control in Carrier Sense Multiple-Access," IEEE Trans. on Communications, Vol. COM. 25, pp. 1103-1120, Oct. 1977.

[20] Little, J., "A Proof of the Queueing Formula L = λW," Operation Research, Vol. 9, pp. 383-387, Mar-Apr, 1961.

[21] Lundstrom, J. and G. Bonnes, "A Controllable MIMD Architecture," Proc. 1980 Int'l Conf. Parallel Processing, pp. 19-27.

[22] Metcalfe, R.M. el al., "Ethernet: Distributed Packet Switching for Local Computer Networks," Communications of the ACM, vol. 19, pp. 395-404, July 1976.

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

[24] Papoulis, A., Probability, Random Variables and Stochastic Processes, McGraw Hill, 1965.

[25] Sauer, C.H. and K. Mani Chandy, Computer Systems Performance Modeling, Prentice-Hall Inc., Englewood Cliffs, N.J. 1981.

[26] Smith, B.J., "A Pipelined, Shared Resource MIMD Computer," Proc. 1978 Int'l Conf. Parallel Processing, pp. 6-8.

[27] Swan, R.J. et al., "Cm* - A Modular, Multi-Microprocessor," Proc. NCC, pp. 637-644, 1977.

[28] Swan, R.J. et al., "The Implementation of the Cm* Multi-Microprocessor," Proc. NCC, pp. 217-223, 1977.

[29] Tanenbaum, A.S., Computer Networks, p. 5, Prentice-Hall Inc., 1981.

[30] "The Selection of Local Area Computer Networks," NBS Special Publication 500-96, pp. 35-50, Nov. 1982.

[31] Tobagi, F. A. et al., "Modeling and Measurement Techniques in Packet Communication Network," IEEE Proc., pp. 1423-1447, Nov. 1978.

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

[33] Wolf, W.A. and C.G. Bell, "C.mmp - A Multi-Miniprocessor," Fall Joint Computer Conference, pp. 756-777, 1972.

[34] Zucker, S.W. et al., "An Application of Relaxation Labelling to Line and Curve Enhancement," IEEE Trans. on Computers, vol. C-26, pp. 394-403 and 922-929, 1977.

APPENDIX A

PAWS SIMULATION PROGRAM OF THE H-NETWORK PROTOCOL WITH NO BACKOFF ALGORITHM

```
!PROGRAM OBJECTIVE
!
!     This simulation program is to model the original protocol of
!the H-Network without backoff algorithms.
!The number of stations simulated is N=10. However,
!the number of stations, e.g.
!N=5, N=15,..etc, do not differ by much.  These will be pointed
!out in this program.
!
!

!DESCRIPTION
!
!     Packets are generated at source node MSOURCE with an inter-
!arrival time lamder (i.e MEAN in the program).  These packets
!will be processed at SETUP1 to fill up the competing stations-
!STATION[I].  At node RDYTIME, a time stamp is marked on the
!packet ready time.  The channel contention is controlled by
!nodes CNTRL1 and CNTRL2.  Node WINDOW is modeled to be the
!window of vulnerability (or propagaion delay).
!
!     Nodes FREEZE and OPEN work in a complementary action.
!The FREEZE node will stop the packets flowing into node CNTRL1
!this represents the CSMA/CD when a ready station arrives at
!an idle network, the channel will be occupied after at most a
!network propagation delay.  On the other hand, when a busy
!period ends, the channel will be freed and it will be available
!for contention again.  This is done at node OPEN.
!
!     The transmitted packets will be sunk into node MSINK and also
!for the collided packets .


DECLARE
   INTEGER TEMP, K, QWAIT, QLENGTH, COUNT, CRASH, INDEX;
   REAL    RTEMP, EXTIME, DT1, DT2, RTT1, RTT2, MEAN;

NODES MSOURCE
      SETUP1
      QSHUT
      QOPEN
      EXSINK
      STATION
      BUS
      WINDOW
      DELTAT
      CNTRL1
      CNTRL2
      WAITQEN
      RDYTIME
      INIT
```

```
        FREEZE
        COLLIDE
        XMIT
        MEASURE
        OPEN
        INTR
        EXITNODE
        MSINK
        ;

CATEGORIES  MESSAGE;
TOKENS      TOK;      !used to limit the packet to be sent
                      !is one per station

TOPOLOGY
    MSOURCE   SETUP1      (MESSAGE,ALL) 1.0;
    SETUP1    EXSINK      (MESSAGE,300) 1.0;
    SETUP1    QSHUT       (MESSAGE,301) 1.0;
    SETUP1    QOPEN       (MESSAGE,302) 1.0;
    QSHUT     EXSINK      (MESSAGE,ALL) 1.0;
    QOPEN     EXSINK      (MESSAGE,ALL) 1.0;
    SETUP1    STATION[LI[I]]  (MESSAGE,ALL) 1.0;
    STATION   RDYTIME     (MESSAGE,ALL) 1.0;
    RDYTIME   WAITGEN     (MESSAGE,ALL) 1.0;
    WAITGEN   BUS         (MESSAGE,ALL) 1.0;
    BUS       DELTAT      (MESSAGE,100) 1.0;
    BUS       CNTRL1      (MESSAGE,ALL) 1.0;
    CNTRL1    WINDOW      (MESSAGE,200) 1.0;
    DELTAT    WAITGEN     (MESSAGE,100) 1.0;
    WINDOW    FREEZE      (MESSAGE,200) 1.0;
    FREEZE    CNTRL2      (MESSAGE,200) 1.0;
    CNTRL2    COLLIDE     (MESSAGE,ALL) 1.0;
    COLLIDE   XMIT        (MESSAGE,202) 1.0;
    XMIT      MEASURE     (MESSAGE,202) 1.0;
    COLLIDE   MEASURE     (MESSAGE,200) 1.0;
    COLLIDE   MEASURE     (MESSAGE,100) 1.0;
    MEASURE   OPEN        (MESSAGE,100) 1.0;
    MEASURE   INTR        (MESSAGE,ALL) 1.0;
    INTR      INIT        (MESSAGE,ALL) 1.0;
    INIT      OPEN        (MESSAGE,ALL) 1.0;
    OPEN      EXITNODE    (MESSAGE,ALL) 1.0;
    EXITNODE  MSINK       (MESSAGE,ALL) 1.0;

DEFINE
    MSOURCE
        ! this is the source for the messages.  The inter-arrival
        ! rate (lamda) of a packet is constant which is MEAN.
        TYPE SOURCE
```

```
   REQUEST (MESSAGE,1) CONSTANT(MEAN);


SETUP1
   ! here we generate the packets to all the STATION[i] by
   ! assigning the phase to a random number.  The range of
   ! this number is from 1..N (In this case, N=10).  We also
   ! have LR[2] := packet generation time
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   GOTOIF 4         GB[7]
   LETEQ  GB[7]     TRUE
   LETEQ  TPHASE    301
   GOTO   3
 LABEL 4
   LETEQ  TPHASE    300
   LETEQ  K         0
 LABEL 1
   ADD    K         K       1
   GT     GB[10]    K       10 !*****
   GOTOIF 2         GB[10]
   LETEQ  QLENGTH   QL[STATION[K],MESSAGE]
   EQ     GB[9]     QLENGTH 10
   GOTOIF 1         GB[9]
   LETEQ  TPHASE    K
   GOTO   3
 LABEL 2
   GOTOIF 3         GB[8]
   LETEQ  GB[8]     TRUE
   LETEQ  TPHASE    302
   LETEQ  GR[7]     TIME
 LABEL 3
   LETEQ    LI[1]     TPHASE
   ;


STATION DIMENSION 10
   ! here the STATION[i] is of dimension N representing the
   ! contending stations.   The number of packets transmitted
   ! is one per station.  This is limited by the TOK of each
   ! station.  In each station, the packet transmission is by
   ! means of the FCFS queueing dicipline(QD).  However, since we
   ! use the INTERRUPT node feature, the interrupted packets
   ! will leave STATION[i] and feed back to STATION[i] in a
   ! random order.  Thus, here we use the PRIORITY SKIP for QD
   ! and LR[2] (the gen. time) to preserve them in a FCFS manner.
   TYPE ALLOCATE
   QUANTITY 1 TOK
   QD PRIORITY SKIP
   REQUEST (ALL,ALL) LR[2] CONSTANT (1) TOK;
```

```
BUS
   ! the packets have the time stamp to access the H-BUS in LR[1],
   ! which is uniformly distributed in [0,T], where T is the
   ! period of latency.  This time is generated at WAITGEN.
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,ALL) CONSTANT (LR[1]);

WINDOW
   ! this is to simulate the network propagation delay.  In our
   ! system, a=0.1 ms or 100 us.
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,ALL) CONSTANT (0.1);

DELTAT
   ! to give a very small delay time, delta t --> 0 so that there
   ! is a presedence for initialization of parameters in INIT before
   ! the transaction pass to WAITGEN.
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,100) CONSTANT (0.00001);

COLLIDE
   ! this is to smulate the collision detection interval.
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,ALL) CONSTANT (9.9);

XMIT
   ! this is to let the packet delay for the equivalent time
   ! interval as that for transmission.
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,202) CONSTANT(20.0);

CNTRL1
   ! this node works with node CNTRL2 to check for collision or
```

```
    ! successful transmission.  Packets pass this node will update
    ! the global counter COUNT which will be checked at CNTRL2.  If
    ! the COUNT = 1, that means there is only one packet arrived
    ! within its window of vulnerability; otherwise if COUNT > 1,
    ! we have a collision.
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    ADD      COUNT       COUNT        1
    ;


CNTRL2
    ! originally the phase of the packets is 200 which correspods to
    ! colliding transactions.  After checking that the COUNT is 1,
    ! the transaction is marked with phase 202 - successful trans-
    ! mission.  The colliding packets will increment the number of
    ! collisons for backoff purposes.
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    GT       GB[12]      COUNT        1
    GOTOIF   1           GB[12]
    LETEQ    TPHASE      202
  LABEL 1
    ADD      LI[2]       LI[2]        1        !update collision count
    LETEQ    GB[22]      FALSE
    GOTOIF   3           GB[23]
    LETEQ    GB[23]      TRUE
  LABEL 2
    LETEQ    GWAIT       GL[BUS,MESSAGE]
    ADD      GI[22]      GWAIT        COUNT
    EQ       GB[55]      GI[22]       10       !****
    GOTOIF   3           GB[55]
    ADD      GI[55]      GI[55]       1
  LABEL 3
    ;


RDYTIME
    ! this is to mark the ready time of the transaction
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    LETEQ    LR[3]       TIME
    LETEQ    LI[2]       1            !initialise collision count
    ;


WAITGEN
    ! this is to generate the random time for access the bus in the
    ! period of latency T; it also keep track the id of transactions
    ! compete for the bus.  The purpose is that these unscessful
```

```
   ! transactions will be interrupted by the transaction(s) that
   ! enters the INTR node
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    TPHASE      200
   LETEQ    LR[1]       UNIFORM(0.0,1.0)
   LETEQ    QI[INDEX]   TID
   ADD      INDEX       INDEX        1
   ;

INIT
   ! this is to initialise the counter INDEX back to value together
   ! with the corresponding TRANS-id QI[1] to QI[10] to 0; the
   ! counter INDEX is to keep track how many transaction has entered
   ! the competition for bus access.  Their trans-id is stored in
   ! the global array.
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    INDEX        1
   LETEQ    K            1
 LABEL 1
   LETEQ    QI[K]        0
   EQ       QB[98]        K         21
   GOTOIF   2            QB[98]
   ADD      K            K          1
   GOTO     1
 LABEL 2
   ;

MEASURE
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    COUNT        0
   NEQ      QB[1]        TPHASE          202
   GOTOIF   1            QB[1]
   ADD      QI[28]       QI[28]          1
   GOTO     2
 LABEL 1
   GOTOIF   2            QB[22]
   LETEQ    QB[22]       TRUE                     !avoid repeated calculate
   ADD      QI[29]       QI[29]          1 .      !colliding period
 LABEL 2
   ;

INTR
   ! this node is to interrput those transactions waiting
   ! in nodes BUS, WINDOW, COLLIDE, to simulate the collision
```

```
   ! event.  Those involve in collision will endup with same
   ! finish time and for those which are not will go back to
   ! WAITQEN again.
   TYPE INTERRUPT
   REQUEST (MESSAGE, ALL) TRANS:QI[1] 100
                          TRANS:QI[2] 100
                          TRANS:QI[3] 100
                          TRANS:QI[4] 100
                          TRANS:QI[5] 100
                          TRANS:QI[6] 100
                          TRANS:QI[7] 100
                          TRANS:QI[8] 100
                          TRANS:QI[9] 100
                          TRANS:QI[10] 100
                          TRANS:QI[11] 100
                          TRANS:QI[12] 100
                          TRANS:QI[13] 100
                          TRANS:QI[14] 100
                          TRANS:QI[15] 100
                          TRANS:QI[16] 100
                          TRANS:QI[17] 100
                          TRANS:QI[18] 100
                          TRANS:QI[19] 100
                          TRANS:QI[20] 100
                          ;

FREEZE
   ! this is to freeze the packets from flowing into BUS so that
   ! they will not interfere the bus activities once the window
   ! of vulnerability is over.
   TYPE SET
   REQUEST (MESSAGE, ALL) STATION[ALL] 0.0
                          BUS 0.0 ;

OPEN
   ! this is to resume the contention once the busy period is
   ! over.
   TYPE SET
   REQUEST (MESSAGE, ALL) STATION[ALL] 1.0
                          BUS 1.0 ;

MSINK
   ! this is the sink of all the transmitted packets.
   TYPE SINK;

QSHUT
   TYPE SET
```

```
      REQUEST (MESSAGE,ALL > STATION[ALL] 0.0;

   QOPEN
      TYPE SET
      REQUEST (MESSAGE,ALL > STATION[ALL] 1.0;

   EXSINK
      TYPE SINK;


   EXITNODE
      ! this is to return the token TOK back to the issuing station,
      ! so that the next packet in this station can compete for
      ! transmission.
      TYPE RELEASE
      REQUEST (MESSAGE,ALL > ALL TOK STATION[LI[1]];


   STATISTICS REPORT
      ! this is to collect statistics.
      RESPONSE SETUP1 MEASURE
      QL  STATION[ALL]
      ;


   RUN
      ! initialize the parameters and run with different inter-
      ! arrival time (lamda) - MEAN.
      LETEQ   MEAN    1.0
    LABEL 1
      LETEQ   COUNT   0
      LETEQ   INDEX   1
      RESET
      GO    10000.0   100.0
      PRINT QI[29]
      PRINT QI[28]
      DUMP
   EXIT
   ;

END;                          !end of simulated program
```

APPENDIX B

PAWS SIMULATION PROGRAM OF THE H-NETWORK PROTOCOL WITH BACKOFF ALGORITHMS

```
!PROGRAM OBJECTIVE
!
!     This simulation program is to model the original
!protocol of the H-Network with backoff algorithms.
!In this program, the backoff algorithms is the
!combination of the quadratic and linear, and the
!number of stations simulated is N=10. However,
!the other two backoff algorithms, quadratic
!and linear, are similar to this one.  Moreover,
!the number of stations, e.g.  N=5,N=15,..etc, do not
!differ by much.  These will be pointed
!out in this program.
!
!
!DESCRIPTION
!
!     Packets are generated at source node MSOURCE
!with an inter arrival time lamder (i.e MEAN in program).
!These packets will be processed at SETUP1 to fill
!up the competing stations STATION[I].  At node
!RDYTIME, a time stamp is marked on the
!packet ready time.  The channel contention is
!controlled by nodes CNTRL1 and CNTRL2.  Node WINDOW
!is modeled to be the window of vulnerability
!(or propagaion delay).
!
!     Nodes FREEZE and OPEN work in a complementary
! action. The FREEZE node will stop the packets
!flowing into node CNTRL1
!this represents the CSMA/CD when a ready station arrives at
!an idle network, the channel will be occupied after at most a
!network propagation delay.  On the other hand, when a busy
!period ends, the channel will be freed and it will be available
!for contention again.  This is done at node OPEN.
!
!     The transmitted packets will be sunk into node MSINK while
!the collided packets will be executing a backoff delay.  The
!delay is generated at node BACKGEN and exercised at node
!BACKWAIT.


DECLARE
   INTEGER TEMP,K,QWAIT,QLENGTH,COUNT,CRASH,INDEX;
   REAL    RTEMP,EXTIME,DT1,DT2,RTT1,RTT2,MEAN;

NODES MSOURCE
      SETUP1
      STATION
      BUS
      WINDOW
```

```
        DELTAT
        CNTRL1
        CNTRL2
        WAITGEN
        RDYTIME
        INIT
        BACKGEN
        BACKWAIT
        FREEZE
        COLLIDE
        XMIT
        MEASURE
        OPEN
        INTR
        EXITNODE
        MSINK
        ;

CATEGORIES   MESSAGE;
TOKENS       TOK;        !used to limit the packet to be sent
                         !is one per station

TOPOLOGY
   MSOURCE    SETUP1          (MESSAGE,ALL) 1.0;
   SETUP1     STATION[LI[1]]  (MESSAGE,ALL) 1.0;
   STATION    RDYTIME         (MESSAGE,ALL) 1.0;
   RDYTIME    WAITGEN         (MESSAGE,ALL) 1.0;
   WAITGEN    BUS             (MESSAGE,ALL) 1.0;
   BUS        DELTAT          (MESSAGE,100) 1.0;
   BUS        CNTRL1          (MESSAGE,ALL) 1.0;
   CNTRL1     WINDOW          (MESSAGE,200) 1.0;
   DELTAT     WAITGEN         (MESSAGE,100) 1.0;
   WINDOW     FREEZE          (MESSAGE,200) 1.0;
   FREEZE     CNTRL2          (MESSAGE,200) 1.0;
   CNTRL2     COLLIDE         (MESSAGE,ALL) 1.0;
   COLLIDE    XMIT            (MESSAGE,202) 1.0;
   XMIT       MEASURE         (MESSAGE,202) 1.0;
   COLLIDE    MEASURE         (MESSAGE,200) 1.0;
   COLLIDE    MEASURE         (MESSAGE,100) 1.0;
   MEASURE    OPEN            (MESSAGE,100) 1.0;
   MEASURE    INTR            (MESSAGE,ALL) 1.0;
   INTR       INIT            (MESSAGE,ALL) 1.0;
   INIT       OPEN            (MESSAGE,ALL) 1.0;
   OPEN       EXITNODE        (MESSAGE,202) 1.0;
   OPEN       BACKGEN         (MESSAGE,ALL) 1.0;
   BACKGEN    BACKWAIT        (MESSAGE,ALL) 1.0;
   BACKWAIT   WAITGEN         (MESSAGE,ALL) 1.0;
   EXITNODE   MSINK           (MESSAGE,ALL) 1.0;
```

```
DEFINE
   MSOURCE
       ! this is the source for the messages.  The inter-arrival
       ! rate (lamda) of a packet is constant which is MEAN.
       TYPE SOURCE
       REQUEST (MESSAGE,1) CONSTANT(MEAN);


   SETUP1
       ! here we generate the packets to all the STATION[i] by
       ! assigning the phase to a random number.  The range of
       ! this number is from 1..N (In this case, N=10).  We also
       ! have LR[2] := packet generation time
       TYPE COMPUTE
       REQUEST (ALL,ALL)
       LETEQ     LR[2]     TIME
       LETEQ     RTEMP     UNIFORM(1.0,10.999999) !depending on N
       LETEQ     TPHASE    FIX(RTEMP)
       LETEQ     LI[1]     TPHASE
       ;


   STATION DIMENSION 10
       ! here the STATION[i] is of dimension N representing the
       ! contending stations.   The number of packets transmitted
       ! is one per station.  This is limited by the TOK of each
       ! station.  In each station, the packet transmission is by
       ! means of the FCFS queueing dicipline(QD).  However, since we
       ! use the INTERRUPT node feature, the interrupted packets
       ! will leave STATION[i] and feed back to STATION[i] in a
       ! random order.  Thus, here we use the PRIORITY SKIP for QD
       ! and LR[2] (the gen. time) to preserve them in a FCFS manner.
       TYPE ALLOCATE
       QUANTITY 1 TOK
       QD PRIORITY SKIP
       REQUEST (ALL,ALL) LR[2] CONSTANT (1) TOK;


   BUS
       ! the packets have the time stamp to access the H-BUS in LR[1],
       ! which is uniformly distributed in [O,T], where T is the
       ! period of latency.  This time is generated at WAITGEN.
       TYPE SERVICE
       QUANTITY 1
       QD DELAY
       REQUEST (MESSAGE,ALL) CONSTANT (LR[1]);


   WINDOW
```

```
    ! this is to simulate the network propagation delay.  In our
    ! system, a=0.1 ms or 100 us.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,ALL) CONSTANT (0.1);

DELTAT
    ! to give a very small delay time, delta t --> 0
    ! so that there
    ! is a presedence for initialization of parameters
    ! in INIT before
    ! the transaction pass to WAITGEN.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,100) CONSTANT (0.00001);

COLLIDE
    ! this is to smulate the collision detection interval.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,ALL) CONSTANT (9.9);

XMIT
    ! this is to let the packet delay for the equivalent time
    ! interval as that for transmission.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,202) CONSTANT(20.0);

BACKWAIT
    ! this is to execute the backoff algorithm.  The waiting time
    ! is generated at BACKGEN and kept in LR[1].
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,ALL) CONSTANT (LR[1]);

CNTRL1
    ! this node works with node CNTRL2 to check for collision or
    ! successful transmission.  Packets pass this node will update
    ! the global counter COUNT which will be checked at CNTRL2.  If
```

```
    ! the COUNT = 1, that means there is only one packet arrived
    ! within its window of vulnerability; otherwise if COUNT > 1,
    ! we have a collision.
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    ADD       COUNT       COUNT         1
    ;

CNTRL2
    ! originally the phase of the packets is 200 which correspods to
    ! colliding transactions.  After checking that the COUNT is 1,
    ! the transaction is marked with phase 202 - successful trans-
    ! mission.  The colliding packets will increment the number of
    ! collisons for backoff purposes.
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    GT        GB[12]      COUNT         1
    GOTOIF    1           GB[12]
    LETEQ     TPHASE      202
  LABEL 1
    ADD       LI[2]       LI[2]         1          !update collision count
    LETEQ     GB[22]      FALSE
  LABEL 3
    ;

RDYTIME
    ! this is to mark the ready time of the transaction
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    LETEQ     LR[3]       TIME
    LETEQ     LI[2]       1                   !initialise collision count
    ;

WAITGEN
    ! this is to generate the random time
    ! for access the bus in the
    ! period of latency T; it also keep
    ! track the id of transactions that
    ! compete for the bus.  The purpose is that these unsccessful
    ! transactions will be interrupted by the transaction(s) that
    ! enters the INTR node
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    LETEQ     TPHASE      200
    LETEQ     LR[1]       UNIFORM(0.0,1.0)
    LETEQ     GI[INDEX]   TID
    ADD       INDEX       INDEX         1
```

```
;

INIT
   ! this is to initialise the counter
   ! INDEX back to value together
   ! with the corresponding TRANS-id
   ! GI[1] to GI[10] to 0; the
   ! counter INDEX is to keep track how many
   ! transaction has entered
   ! the competition for bus access.
   ! Their trans-id is stored in
   ! the global array.
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    INDEX      1
   LETEQ    K          1
 LABEL 1
   LETEQ    GI[K]      0
   EQ       GB[98]      K          21
   GOTOIF   2          GB[98]
   ADD      K          K          1
   GOTO     1
 LABEL 2
   ;

BACKGEN
   ! this is the node to implement the backoff algorithm,
   ! the existing method is by means of the quadratic backoff;
   ! we can also use the linear as well as the combination
   ! of the two.
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    CRASH      1                          !back off algor.
   LETEQ    LR[1]      0.05                       !LI[2] as counter
 LABEL 1                                          !for colliding freq.;
   MUL      LR[1]      LR[1]      2.0
   EQ       GB[20]     CRASH      LI[2]           !if it is finish
   GOTOIF   3          GB[20]                     !quit; else
   ADD      CRASH      CRASH      1               !keep multiplying
   GOTO     1
 LABEL 3
   ;

MEASURE
   TYPE COMPUTE
   REQUEST (ALL,ALL)
   LETEQ    COUNT      0
```

```
    NEQ     QB[1]      TPHASE        202
    GOTOIF  1          QB[1]
    ADD     QI[28]     QI[28]        1
    LETEQ   EXTIME     TIME                    !record EXTIME
    SUB     DT1        EXTIME        LR[2]     !calculate mean
    SUB     DT2        EXTIME        LR[3]     !and var.
    ADD     QR[3]      QR[3]         DT1
    ADD     QR[5]      QR[5]         DT2
    MUL     RTT1       DT1           DT1
    MUL     RTT2       DT2           DT2
    ADD     QR[4]      QR[4]         RTT1
    ADD     QR[6]      QR[6]         RTT2
    GOTO    2
  LABEL 1
    GOTOIF  2          QB[22]
    LETEQ   QB[22]     TRUE                !avoid repeated calculate
    ADD     QI[29]     QI[29]        1     !colliding period
  LABEL 2
    ;

INTR
    ! this node is to interrput those transactions waiting
    ! in nodes BUS, WINDOW, COLLIDE, to simulate the collision
    ! event.  Those involve in collision will endup with same
    ! finish time and for those which are not will go back to
    ! WAITGEN again.
    TYPE INTERRUPT
    REQUEST (MESSAGE,ALL) TRANS:QI[1] 100
                          TRANS:QI[2] 100
                          TRANS:QI[3] 100
                          TRANS:QI[4] 100
                          TRANS:QI[5] 100
                          TRANS:QI[6] 100
                          TRANS:QI[7] 100
                          TRANS:QI[8] 100
                          TRANS:QI[9] 100
                          TRANS:QI[10] 100
                          TRANS:QI[11] 100
                          TRANS:QI[12] 100
                          TRANS:QI[13] 100
                          TRANS:QI[14] 100
                          TRANS:QI[15] 100
                          TRANS:QI[16] 100
                          TRANS:QI[17] 100
                          TRANS:QI[18] 100
                          TRANS:QI[19] 100
                          TRANS:QI[20] 100
                          ;
```

```
FREEZE
   ! this is to freeze the packets from flowing into BUS so that
   ! they will not interfere the bus activities once the window
   ! of vulnerability is over.
   TYPE SET
   REQUEST (MESSAGE,ALL) STATION[ALL] 0.0
                         BUS 0.0 ;

OPEN
   ! this is to resume the contention once the busy period is
   ! over.
   TYPE SET
   REQUEST (MESSAGE,ALL) STATION[ALL] 1.0
                         BUS 1.0 ;

MSINK
   ! this is the sink of all the transmitted packets.
   TYPE SINK;

EXITNODE
   ! this is to return the token TOK back to the issuing station
   ! so that the next packet in this station can compete for
   ! transmission.
   TYPE RELEASE
   REQUEST (MESSAGE,ALL) ALL TOK STATION[LI[1]];

STATISTICS REPORT
   ! this is to collect statistics.
   RESPONSE SETUP1 MEASURE
   QL  STATION[ALL]
   ;

RUN
   ! initialize the parameters and run with different inter-
   ! arrival time (lamda) - MEAN.
   LETEQ  MEAN   15.0
 LABEL 1
   LETEQ  COUNT  0
   LETEQ  INDEX  1
   LETEQ  K      1
 LABEL 3
   LETEQ  QI[K]  0
   EQ     QB[98] K      30
   GOTOIF 2      QB[98]
```

```
      ADD     K       K      1
      GOTO    3
    LABEL 2
      RESET
      GO    5000.0  100.0
      PRINT QI[29]
      PRINT QI[28]
      ADD   MEAN   MEAN        5.0
      LEQ   QB[100] MEAN       100.0
      GOTOIF  1      QB[100]
      DUMP
   EXIT
   ;

END;                          !end of simulated program
```

APPENDIX C

HARDWARE MODIFICATION OF THE ACCESS LINE TO ENABLE THE PEBBLE POOL PARADIGM

The detection of the pebble pool and the deposition of the pebble into the pool are the actual activities involved on the AL (Access Line).

One hardware implementation is to have a detection scheme on the AL. This line is common to all stations. It is tied to an open collector buffer and all the TEST & SET controllers in each station (as shown in Fig. C-1). The amount of current that sinks into the buffer will be inversely proportional to the number of TEST & SET signals being set on the AL. Therefore we can detect the number of contending stations by sensing the value of the current. There are three thresholds on this signal that correspond to three important states of the network:

1) there is no attempting stations;
2) there is only one contending station; and
3) there are more than one contending stations.

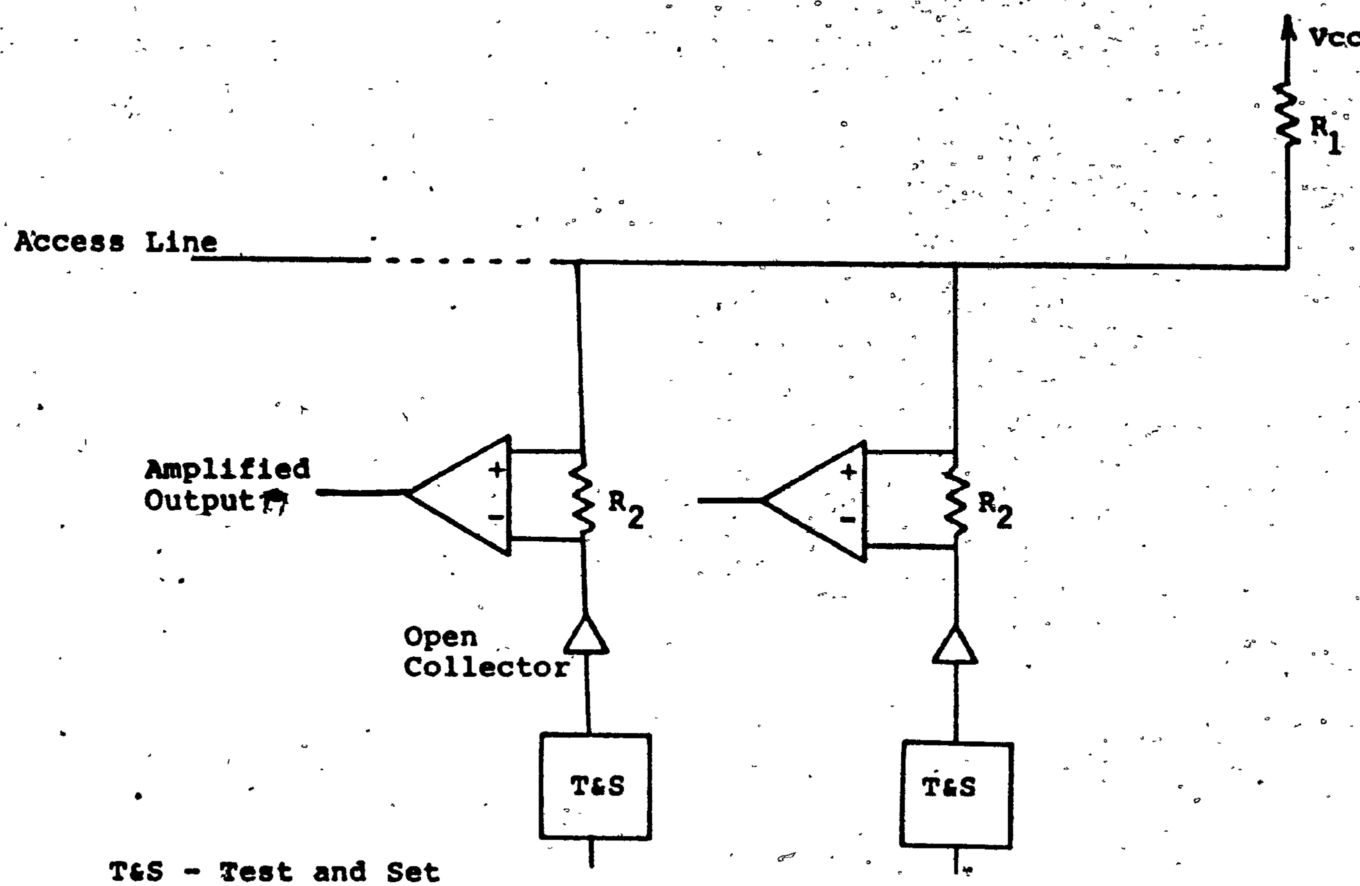


Fig. C-1 Hardware Modification on Access Line to enable Pebble Paradigm

APPENDIX D

MATHEMATICAL SIMPLIFICATION OF THE PROBABILITY OF SUCCESS IN CASE (a)

$$(p+q)^n = p^n+\binom{n}{1}p^{n-1}q+\ldots+\binom{n}{r}p^{n-r}q^r+\ldots+q^n$$

$$= \sum_{i=0}^{n} \binom{n}{i}p^{n-i}q^i$$

$$= p^n + \sum_{i=1}^{n-1} \binom{n}{i}p^{n-i}q^i+q^n$$

hence

$$\sum_{i=1}^{n-1} \binom{n}{i}p^{n-i}q^i = (p+q)^n - q^n - p^n$$

Now

let $p = T - W_1 - a$

$q = W_1 - W_3$

$$\sum_{\ell=1}^{N-1} \binom{N}{\ell}(T-W_1-a)^{N-\ell}(W_1-W_3)^{\ell}$$

$$= (T - W_1 - a + W_1 - W_3)^N \ (W_1 - W_3)^{\ell}$$

$$= (T - W_3 - a)^N - (T - W_1 - a)^N - (W_1 - W_3)^N$$

APPENDIX E

EVALUATION OF MEAN AND VARIANCE OF A SINGLE CONTENTION PERIOD

Now with

$$x_i = t + W_1 + W_2 + W_3 + a$$

where t is a random variable uniformly distributed in the interval $[0,T]$ and W_1, W_2, W_3 and a are constants.

hence

$$x_i = t + C_1 \qquad \text{where } C_1 \text{ is a constant}$$

$$\begin{aligned}
E[x_i] &= E\,[t + C_1] \\
&= E\,[t] + C_1 \\
&= \frac{T}{2} + C_1 \\
&= \frac{T}{2} + W_1 + W_2 + W_3 + a
\end{aligned}$$

$$\begin{aligned}
\sigma_{x_i}{}^2 &= E[x^2] - E^2[x] \\
&= E[(t + C_1)^2] - E^2[t + C_1] \\
&= E[t^2 + 2C_1 t + C_1{}^2] - (E[t] + C_1)^2 \\
&= E[t^2] + 2C_1\ E[t] + C_1{}^2 \\
&\qquad - E^2[t] - 2C_1\ E[t] - C_1{}^2 \\
&= E\,[t^2] - E^2[t]
\end{aligned}$$

Now

$$E[t^2] = \int_{-\infty}^{\infty} t^2 f_t \, dt$$

$$= \int_0^T t^2 \left(\frac{1}{T}\right) dt$$

$$= \frac{1}{T} \cdot \frac{t^3}{3} \Bigg|_0^T$$

$$= \frac{T^2}{3}$$

$$\sigma_{x_i}^{\ 2} = \frac{T^2}{3} - \left(\frac{T^2}{2}\right)$$

$$= \frac{T^2}{12}$$

APPENDIX F

MATHEMATICAL SIMPLIFICATION OF THE AVERAGE IDLE PERIOD GIVEN THAT THERE ARE n=k CONTENTIONS

$$I_k = \int_{\tau}^{\infty} (z-\tau) \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\eta)^2/2\sigma^2} dz$$

$$= \frac{1}{\sigma\sqrt{2\pi}} \int_{\tau}^{\infty} e^{-(z-\eta)^2/2\sigma^2} dz$$

$$- \frac{\tau}{\sigma\sqrt{2\pi}} \int_{\tau}^{\infty} e^{-(z-\eta)^2/2\sigma^2} dz$$

Let $t = \frac{z-\eta}{\sqrt{2}\sigma} \rightarrow dt = \frac{dz}{\sqrt{2}\sigma}$

and $t_1 = \frac{\tau-\eta}{\sqrt{2}\sigma}$

therefore,

$$I_k = \frac{1}{\sqrt{\pi}} \int_{\frac{\tau-\eta}{\sqrt{2}}}^{\infty} (\sqrt{2}\sigma \cdot t + \eta) e^{-t^2} dt$$

$$- \frac{\tau}{\sqrt{\pi}} \int_{\frac{\tau-\eta}{\sqrt{2}\sigma}}^{\infty} e^{t^2} dt$$

$$= \sqrt{\frac{2}{\pi}} \sigma \int_{t_1}^{\infty} t \, e^{-t^2} dt + \frac{\eta-\tau}{\sqrt{\pi}} \int_{t_1}^{\infty} e^{-t^2} dt$$

Now let

$$I_1 = \sqrt{\frac{2}{\pi}}\ \sigma \int_{t_1}^{\infty} t\ e^{-t^2}\ dt$$

let $\omega = t^2 \rightarrow d\omega = 2t\ dt$

$$I_1 = \sqrt{\frac{2}{\pi}}\ \sigma \int_{t_1}^{\infty} t\ e^{-t^2}\ dt$$

$$= (\tfrac{1}{2}) \sqrt{\frac{2}{\pi}}\ \sigma \int_{t_1^2}^{\infty} e^{-\omega} d\omega$$

$$= -\frac{\sigma}{\sqrt{2\pi}}\ e^{-\omega} \Big|_{t_1^2}^{\infty}$$

$$= -\frac{\sigma}{\sqrt{2\pi}} = [0 - e^{-t_1^2}]$$

$$= \frac{\sigma}{\sqrt{2\pi}}\ e^{-(\tau-\eta)^2/2\sigma^2}$$

Hence $I_k = I_1 + \frac{\eta-\tau}{\sqrt{\pi}} \int_{t_1}^{\infty} e^{-t^2}\ dt$

$$= \frac{\sigma}{\sqrt{2\pi}}\ e^{(\tau-\eta)^2/2\sigma^2} + \frac{\eta-\tau}{2}\ \mathrm{erfc}\ (\frac{\tau-\eta}{\sqrt{2}\sigma})$$

where $\mathrm{erfc}(y) = \frac{2}{\sqrt{\pi}} \int_{y}^{\infty} e^{-t^2}\ dt$

APPENDIX G

NUMERICAL PROGRAM TO CALCULATE THE AVERAGE IDLE PERIOD AND THE AVERAGE UTILIZATION FACTOR OF THE NEW COLLISION-FREE PROTOCOL

```
      PROGRAM SYS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DOUBLE PRECISION TEL,T,XMEAN,YMEAN,PS,PF,RMEAN,SIGMA,PI,S
      DOUBLE PRECISION PSUM,EBDD,E,TEMP1,TEMP2,P1,P2,P3,RIK,TTSQ
C
C.... PROGRAM TO CALCULATE THE AVERAGE UTILIZATION FACTOR OF THE NEW
C     COLLISION FREE PROTOCOL
C
C           I-MEAN = SUMMATION(PF**(K-1)*PS*IK)  FOR I=NMIN,INFINITY
C     AND
C           S-MEAN = U-MEAN / (U-MEAN + I-MEAN)
C
C           S-MEAN IS THE SYSTEM UTIL. FACTOR,
C           U-MEAN IS TEL
C
      T=1.0
      TEL=20.0
      A=0.2
      W1=A
      W2=2.0*A+W1
      W3=0
      XMAX=T+W1+W2+W3+A
      YMAX=T+W1+W2
      NMIN=INT((TEL-YMAX)/XMAX)+1
      XMEAN=T/2.+W1+W2+W3+A
      YMEAN=T/2.+W1+W2
      SIGSQ=T/12.
*
      PI=3.141592654
C
C
      WRITE(6,2)
2     FORMAT (1H1)
      WRITE(6,19) NMIN,XMEAN,YMEAN
19    FORMAT(1H ,///,'NMIN=',I3,2X,'MEANS X, Y :',2(F7.4,2X))
C
      READ (5,*) MULT
      EBDD=1.0E-50
C
      WRITE(6,10)
10    FORMAT(1H ,T10,'INPUT # OF STATIONS AND STEP',
     +       '(NFIRST,NLAST,NSTEP)')
C
      READ(5,*) NFIRST,NLAST,NSTEP
C
C.... CALCULATION OF THE AVERAGE IDLE PERIOD
C     AND SYSTEM UTIL. FACTOR FOR NS
C
      DO 250 NS=NFIRST,NLAST,NSTEP
         IF (NS.EQ.0) GOTO 250
         CALL FUNCT(NS,T,A,PS)
         PF=1.0-PS
         WRITE(6,222) NS,PS
222      FORMAT(1H ,///T10,'# OF STATIONS= ',I4,2X,'PS= ',F25.23)
C
         PSUM=0.
```

```
      TEMP1=0.
      TEMP2=0.
      E=1.E+15
      KOUNT=0
      DO 200  K=NMIN,10000000
         P1=PF**(K-1)*PS
         RMEAN=FLOAT(K-1)*XMEAN+YMEAN
         SIGMA=SQRT(FLOAT(K)*SIGSQ)
         TT=(1.0/SQRT(2.0))*(TEL-RMEAN)/SIGMA
         TTSQ=DBLE(TT*TT)
*
*        CHECK THE VALUE EXP(-TTSQ)
*
         IF (TTSQ.LE.540.0)THEN
            P2=(1.0/SQRT(2.0*PI))*SIGMA*DEXP(-1.0*TTSQ)
         ELSE
            P2=0.0
         ENDIF
         P3=(1.0/2.0)*(RMEAN-TEL)*ERFC(TT)
         RIK=P2+P3
         PSUM=P1*RIK+PSUM
         TEMP2=P1*RIK + TEMP2
*
*        FIND THE FINITE DIFFERENCES AFTER 200 TERMS
*
         KOUNT=KOUNT+1
         IF (KOUNT.EQ.200) THEN
            KOUNT=0
            IF(PSUM.LE.0.0) GOTO 199
            IF(((TEMP2/PSUM).LT.1.0E-50).OR.
     +            (TEMP2.LE.1.0E-100))GOTO 230
199         CONTINUE
            E=(TEMP2-TEMP1)*FLOAT(MULT)
*           WRITE (6,231)E
231         FORMAT(1H ,T10,'** E (INSIDE IF) **',E20.10)
            E=DABS(E)
            TEMP1=TEMP2
            TEMP2=0.
         ENDIF
C
C.... CHECK THE TERM SUM FOR CONVERGENCE
C
C
         IF (E.LT.EBDD) GOTO 230
200   CONTINUE
*
C
230   CONTINUE
      WRITE(6,233) K
233   FORMAT(1H ,T10,'# OF TERMS IS ', I10)
*
*
      WRITE(6,211) PSUM
211   FORMAT(1H ,T10,'THE AVERAGE IDLE PERIOD =',G25.15)
C
      S=TEL/(TEL + PSUM)
      WRITE (6,213) S
```

```
213     FORMAT(1H ,/T10,'THE SYS. UTIL. FACTOR IS ',F25.23)
C
250   CONTINUE
C
      STOP
      END
```

```
      SUBROUTINE FUNCT(NS,T,A,PS)
*****************************************************************
*     SUBROUTINE FUNCT(NS,T,A,PS)                               *
*     INPUTS : NS - # OF STATIONS, T - PERIOD OF LATENCY        *
*              A - PROPAGATION DELAY                            *
*     OUTPUT : PS - PROPABILITY OF A SUCESS CONTENTION          *
*****************************************************************
      DOUBLE PRECISION PS,T,FF,Y
C
C.... PROGRAM TO CALCULATE THE PROBABILITY OF A SUCESSFUL
C     CONTENTION  PERIOD
*
*     ACTUALLY THE PROBABILITY OF SUCCESS IS ((T-W3-A)/T)**NS
*     HOWEVER, IN OUR CASE W3=0.
*
      PS=((T-A)/T)**NS
C
      RETURN
      END
```

APPENDIX H

PAWS SIMULATION PROGRAM FOR THE NEW COLLISION-FREE PROTOCOL

```
DECLARE
    INTEGER TEMP, KK, NS, K, QWAIT, QLENGTH, COUNT, CRASH, INDEX;
    REAL    RTEMP, EXTIME, DT1, DT2, RTT1, RTT2, MEAN;

NODES MSOURCE
      SETUP
      QSHUT
      QOPEN
      STATION
      BUS
      RDYTIME
      WAITQEN
      BACKLOG
      SENSE1
      W1
      WINDOW1
      SETAL1
      SETAL2
      W2
      SENSE2
      WINDOW2
      WITHDRAW1
      BOPEN1
      INTR
      DELTAT
      DFREEZE
      WITHDRAW2
      BOPEN2
      XMIT
      DOPEN
      MEASURE
      EXITNODE
      MSINK
      ;

CATEGORIES  MESSAGE;
TOKENS      TOK;        !used to limit the packet to be sent
                        !is one per station

TOPOLOGY
    MSOURCE   SETUP       (MESSAGE, ALL ) 1.0;
    SETUP     MSINK       (MESSAGE, 300 ) 1.0;
    SETUP     QSHUT       (MESSAGE, 301 ) 1.0;
    SETUP     QOPEN       (MESSAGE, 302 ) 1.0;
    QSHUT     MSINK       (MESSAGE, 301 ) 1.0;
    QOPEN     MSINK       (MESSAGE, 302 ) 1.0;
    SETUP     STATION[LI[1]]  (MESSAGE, ALL ) 1.0;
    STATION   RDYTIME     (MESSAGE, ALL ) 1.0;
```

```
    RDYTIME    WAITGEN       (MESSAGE, ALL) 1.0;
    WAITGEN    BUS           (MESSAGE, 200) 1.0;
    BUS        SENSE1        (MESSAGE, 200) 1.0;
    BUS        BACKLOG       (MESSAGE, 100) 1.0;
    SENSE1     W1            (MESSAGE, 202) 1.0;
    SENSE1     BACKLOG       (MESSAGE, 200) 1.0;
    W1         SETAL1        (MESSAGE, 202) 1.0;
    SETAL1     WINDOW1       (MESSAGE, 202) 1.0;
    BACKLOG    WAITGEN       (MESSAGE, ALL) 1.0;
    WINDOW1     SETAL2       (MESSAGE, 202) 1.0;
    SETAL2     W2            (MESSAGE, 202) 1.0;
    W2         SENSE2        (MESSAGE, 202) 1.0;
    SENSE2     DELTAT        (MESSAGE, 202) 1.0;
    SENSE2     WINDOW2       (MESSAGE, 200) 1.0;
    WINDOW2    WITHDRAW1     (MESSAGE, 200) 1.0;
    WITHDRAW1  BACKLOG       (MESSAGE, 200) 1.0;
    WITHDRAW1  INTR          (MESSAGE, 201) 1.0;
    INTR       BOPEN1        (MESSAGE, 201) 1.0;
    BOPEN1     WAITGEN       (MESSAGE, ALL) 1.0;
    BACKLOG    WAITGEN       (MESSAGE, ALL) 1.0;
    DELTAT     DFREEZE       (MESSAGE, 202) 1.0;
    DFREEZE    WITHDRAW2     (MESSAGE, 202) 1.0;
    WITHDRAW2  INTR          (MESSAGE, 202) 1.0;
    INTR       BOPEN2        (MESSAGE, 202) 1.0;
    BOPEN2     EXITNODE      (MESSAGE, 202) 1.0;
    EXITNODE   XMIT          (MESSAGE, 202) 1.0;
    XMIT       MEASURE       (MESSAGE, 202) 1.0;
    MEASURE    DOPEN         (MESSAGE, 202) 1.0;
    DOPEN      MSINK         (MESSAGE, 202) 1.0;
    ;

DEFINE
    MSOURCE
        ! this is the source for the messages.  The inter-arrival
        ! rate (lamda) of a packet is constant which is MEAN.
        TYPE SOURCE
        REQUEST (MESSAGE, 1) CONSTANT(MEAN);
        ;

    SETUP
        ! here we generate the packets to all the STATION[i] by
        ! assigning the phase to a random number.. The range of
        ! this number is from 1..N (In this case, N=10).  We also
        ! have LR[2] := packet generation time
        TYPE COMPUTE
        REQUEST (ALL, ALL)
        LETEQ     LR[2]     TIME
        GOTOIF    4         GB[7]          !close the gate
        LETEQ     GB[7]     TRUE
```

```
  LETEQ    TPHASE    301
  GOTO     3
LABEL 4
  LETEQ    TPHASE    300              !let the packet goes
  LETEQ    K         0                !to the stations that require
LABEL 1
  ADD      K         K         1      !check all the stations
  GT       GB[6]     K         NS     !from 1 to N, in this case N=5
  GOTOIF   2         GB[6]
  LETEQ    QLENGTH   QL[STATION[K],MESSAGE]
  EQ       GB[9]     QLENGTH   15     !see if all stations each
  GOTOIF   1         GB[9]            !filled up with 10 packets
  LETEQ    TPHASE    K                !before simulation
  GOTO     3                          !
LABEL 2
  GOTOIF   3         GB[8]            !open gate and start the
  LETEQ    GB[8]     TRUE             !simulation
  LETEQ    TPHASE    302
  LETEQ    GR[7]     TIME             !mark the starting time
  !TRACE
LABEL 3
  LETEQ    LI[1]     TPHASE
  !LETEQ    RTEMP     UNIFORM(1.0,10.999999) !depending on N
  !LETEQ    TPHASE    FIX(RTEMP)
  !LETEQ    LI[1]     TPHASE
  ;


QSHUT
  ! this is to close the gate to set up the queues of packet
  ! in each STATION[i] before start the simulation
  TYPE SET
  REQUEST (MESSAGE,ALL) STATION[ALL] 0.0;

QOPEN
  ! this is to open the gate when the set up is ready
  TYPE SET
  REQUEST (MESSAGE,ALL) STATION[ALL] 1.0;


STATION DIMENSION 10
  ! here the STATION[i] is of dimension N representing the
  ! contending stations.  The number of packets transmitted
  ! is one per station.  This is limited by the TOK of each
  ! station.  In each station, the packet transmission is by
  ! means of the FCFS queueing dicipline(QD).  However, since we
  ! use the INTERRUPT node feature, the interrupted packets
  ! will leave STATION[i] and feed back to STATION[i] in a
  ! random order.  Thus, here we use the PRIORITY SKIP for QD
  ! and LR[2] (the gen. time) to preserve them in a FCFS manner.
```

```
    TYPE ALLOCATE
    QUANTITY 1 TOK
    QD PRIORITY SKIP
    REQUEST (ALL,ALL) LR[2] CONSTANT (1) TOK;


BUS
    ! the packets have the time stamp to access the H-BUS in LR[1],
    ! which is uniformly distributed in [O,T], where T is the
    ! period of latency.  This time is generated at WAITGEN.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,ALL) CONSTANT (LR[1]);


RDYTIME
    ! this is to mark the ready time of the transaction
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    LETEQ   LR[3]       TIME
    ;


WAITGEN
    ! this is to generate the random time for access the bus in the
    ! period of latency T; it also keep track the id of transactions
    ! that compete for the bus.The purpose is that these unsccessful
    ! transactions will be interrupted by the transaction(s) that
    ! enters the INTR node
    TYPE COMPUTE
    REQUEST (ALL,ALL)
    LETEQ   TPHASE      200
    LETEQ   LR[1]       UNIFORM(0.0,1.0)
    ADD     KK          KK          1
    LEQ     GB[79]      KK          21
    GOTOIF  1           GB[79]
    ADD     GI[79]      GI[79]      1
  LABEL 1
    LETEQ   GI[KK]      TID
    LETEQ   LI[2]       KK                  !keep the vector index
    ADD     GI[29]      GI[29]      1       !update participation count
    TRACE
    ;


BACKLOG
    ! this is the backlogged state that when the contending station
    ! senses the AL before laying the pebble or the station has
    ! failed in competing for the channel.
```

```
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,ALL) CONSTANT (0.00000001);


SENSE1
   ! this is when the station senses the AL before enter for
   ! contending for the channel
   TYPE COMPUTE
   REQUEST (MESSAGE,200)
   LETEQ     TEMP        LI[2]
   LETEQ     GI[TEMP]    0                !avoid being interrupted
   NEQ       GB[10]      COUNT     0      !if pebble pool empty
   GOTOIF    1           GB[10]           !No, backlog
   LETEQ     TPHASE      202              !Yes, keep going
 LABEL 1
   ;


W1
   ! the first waiting interval W1 is equal to the
   ! propagation delay, a
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,202) CONSTANT (0.1);


WINDOW1
   ! the window of vulnerability
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,202) CONSTANT (0.1);


SETAL1
   ! to lay the pebble in the pool
   TYPE SET
   REQUEST (MESSAGE,202) BACKLOG 0.0;

SETAL2
   !
   TYPE COMPUTE
   REQUEST (MESSAGE,202)
   ADD       COUNT       COUNT     1
   ;
```

```
W2
    ! second waiting period (2a+max{Wi1}) is 0.3, however
    ! the window of vulnerability is simulated at node WINDOW
    ! therefore the waiting period is 0.3 - 0.1 = 0.2
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,202) CONSTANT (0.2);

SENSE2
    ! recheck the pool again for transmission
    TYPE COMPUTE
    REQUEST (MESSAGE,202)
    EQ         GB[11]      COUNT     1       !if number of pebbles = 1
    GOTOIF     1           GB[11]            !Yes, go for transmission
    LETEQ      TPHASE      200               !else backlog
    GOTO       2
  LABEL 1
    ADD        GI[51]      GI[51]    1       !check transmitted packets
    LETEQ      TEMP        LI[2]
    LETEQ      GI[TEMP]    0
  LABEL 2                                    !when flushing the BUS
    ;

WINDOW2
    ! when the station fails to get the mastership of the network
    ! it retrieve its pebble.  Then only after a network pro-
    ! pagation delay, the other stations realise its
    ! pebble retrieval.
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,200) CONSTANT (0.1);

WITHDRAW1
    ! withdraw pebble
    TYPE COMPUTE
    REQUEST (MESSAGE,200)
    SUB        COUNT       COUNT     1       !retrieve pebble
    NEQ        GB[13]      COUNT     0       !if nobody gets
    GOTOIF     1           GB[13]            !channel
    LETEQ      TPHASE      201               !open BACKLOG gate
    ADD        GI[52]      GI[52]    1       !update count
    LETEQ      KK          0                 !reset interrupt index
  LABEL 1
    ;
```

```
BOPEN1
    TYPE SET
    REQUEST (MESSAGE,201) BACKLOG 1.0;

ANTR
    TYPE INTERRUPT
    REQUEST (MESSAGE,ALL) TRANS:GI[1] 100
                          TRANS:GI[2] 100
                          TRANS:GI[3] 100
                          TRANS:GI[4] 100
                          TRANS:GI[5] 100
                          TRANS:GI[6] 100
                          TRANS:GI[7] 100
                          TRANS:GI[8] 100
                          TRANS:GI[9] 100
                          TRANS:GI[10] 100
                          TRANS:GI[11] 100
                          TRANS:GI[12] 100
                          TRANS:GI[13] 100
                          TRANS:GI[14] 100
                          TRANS:GI[15] 100
                          TRANS:GI[16] 100
                          TRANS:GI[17] 100
                          TRANS:GI[18] 100
                          TRANS:GI[19] 100
                          TRANS:GI[20] 100
                          TRANS:GI[21] 100
                          ;

DELTAT
    ! small delay
    TYPE SERVICE
    QUANTITY 1
    QD DELAY
    REQUEST (MESSAGE,202) CONSTANT (0.000001);

DFREEZE
    TYPE SET
    REQUEST (MESSAGE,202) DELTAT 0.0;

WITHDRAW2
    ! the transmitting station will withdraw its pebble (the pool
    ! should be empty again) so that all the station can compete
    ! for the channel in the next TP.
    TYPE COMPUTE
    REQUEST (MESSAGE,202)
    SUB      COUNT      COUNT      1
    SUB      GI[29]     GI[29]     1      'avoid repeat count
```

```
  EQ        GB[12]     COUNT      0      !pebble pool empty
  GOTOIF    1          GB[12]            !if not, report error
  ADD       GI[99]     GI[99]     1      !update error count
 LABEL 1
  LETEQ     KK         0                 !reset INTR index
  ;

BOPEN2
   !
   TYPE SET
   REQUEST (MESSAGE,202) BACKLOG 1.0;

XMIT
   TYPE SERVICE
   QUANTITY 1
   QD DELAY
   REQUEST (MESSAGE,202) CONSTANT (29.999999);

DOPEN
   TYPE SET
   REQUEST (MESSAGE,202) DELTAT 1.0;

MEASURE
   ! to collect the statistics and to check whether there is
   ! always NS number of stations competing at any time
   TYPE COMPUTE
   REQUEST (ALL,ALL)
    LETEQ    GI[1]      QL[BUS,MESSAGE]
    LETEQ    GI[2]      QL[W1,MESSAGE]
    LETEQ    GI[3]      QL[WINDOW1,MESSAGE]
    LETEQ    GI[4]      QL[W2,MESSAGE]
    LETEQ    GI[5]      QL[BACKLOG,MESSAGE]
    LETEQ    GI[6]      QL[WINDOW2,MESSAGE]
    LETEQ    GI[7]      QL[DELTAT,MESSAGE]
    LETEQ    GI[8]      QL[XMIT,MESSAGE]
    LETEQ    INDEX      0
    LETEQ    GI[29]     0
  LABEL 1
    ADD      INDEX      INDEX        1   !find the total packets
    ADD      GI[29]     GI[29]       GI[INDEX]
    L        GB[13]     INDEX        8
    GOTOIF   1          GB[13]
   LETEQ    KK         0                 !should be in withdraw2
   EQ       GB[14]     GI[29]       NS
   GOTOIF   2          GB[14]            !perform testing
   ADD      GI[30]     GI[30]       1    !update error count
```

```
 LABEL 2
   ADD     GI[28]     GI[28]          1  !update # of xmit period
   ;


EXITNODE
   ! this is to return the token TOK back to the issuing station
   ! so that the next packet in this station can compete for
   ! transmission.
   TYPE RELEASE
   REQUEST (MESSAGE,ALL) ALL TOK STATION[LI[1]];


MSINK
   ! sink of all packets
   TYPE SINK
   ;


STATISTICS REPORT
   ! this is to collect statistics.
   RESPONSE SETUP  MEASURE
   QL  STATION[ALL]
   ;


RUN
   ! initialize the parameters and run with different inter-
   ! arrival time (lamda) - MEAN.
   LETEQ  MEAN   2.0
   LETEQ  NS     10
   !**** don't forget to change dimension STATION
   !**** don't forget to change queue length SETUP
   LETEQ  COUNT  0
   RESET
   GO     5000.0   100.0
    PRINT GR[7]             ! start time
    PRINT GI[28]            ! # of X-mission periods
    PRINT GI[30]            ! error count for # stations contending
    PRINT GI[99]            ! error count for empty pebble pool
   DUMP
EXIT
;

END;                        !end of simulated program
```