8-31-1991

Performance of the extended token bus network with asymmetric load distribution

Hong Lin
New Jersey Institute of Technology

ABSTRACT

The Token Bus, IEEE 802.4, provides high priority message options. The delay of high priority messages becomes an important criterion in the design and management of computer communication networks. A goal of the protocol is to allocate sufficient bandwidth and frequency of access to minimize of the high priority message delay. In this study, a simulation model of a priority based network, consisting of two token bus networks interconnected by a bridge is implemented. The load distribution among stations is asymmetric and the messages are divided into two priorities, high and low. Both fixed and variable token token holding times for each station to transfer high priority message, two different approaches are considered and implemented. In the variable case token hold may vary according to the local and global load. The simulation results show that the variable high priority token holding time approach can reduce the high priority message delay.

Performance of the Exteded Token Bus Network with Asymmetric Load Distribution

by

Hong Lin

Thesis submitted to the Faculty of the Graduate School of

the New Jersey Institute of Technology in partial fulfillment of

the requirements for the degree of

Master of Science in Electrical Engineering

1991

APPROVAL SHEET

PERFORMANCE OF THE EXTENDED TOKEN BUS NETWORK WITH ASYMMETRIC LOAD DISTRIBUTION

Name of Candidate: HONG LIN
MASTER OF SCIENCE
IN ELECTRICAL ENGINEERING, 1991

Thesis Approval:

Dr. Constantine Manikopoulos — Date
Associate Professor

Dr. Anthony Robbi — Date
Associate Professor

Dr. Mengchu Zhou — Date
Assistant Professor

VITA

HONG LIN

DATE OF BIRTH:

PLACE OF BIRTH:

EDUCATION:

Jan. 1990 - Aug. 1991	New Jersey Institue of Technology MSEE
Jan. 1988 - Dec. 1989	Staten Island College, CUNY, NY BSES

Dedicated to My Father

... for all his love

ACKNOWLEDGEMENT

I am very grateful to my graduate advisor Dr. Constantine Manikopoulos whose inspirations and guidance benefited me significantly; without his support this work could not have been finished.

I would also like to thank my friends and classmates at NJIT, who have given me their assistance and encouragement in many ways.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Local area networks(LANs) are being widely deployed and used in offices, business, and research because of the attractive solutions they provide for distributed resource sharing. However, LANs are limited in their geographical coverage, information-carrying capacity, and the number of users they can support. LAN traffic is expected to increase manifold over the next few years because of several trends: an increasing LAN user base, an increasing number of software applications that employ bandwidth-consuming graphical user interfaces, migration towards diskless workstations attached to file servers via a LAN, etc.

In order to accommodate gradual traffic growth, the simplest solution is to acquire another LAN (possiblely of the same kind as the first) and have the two LANs interconnected by a device that is generally referred to as a bridge. A bridge lies midway in complexity between the repeaters used in Ethernet and the gateway processor used between networks in internetworking environment.

There has been considerable activity in the study of the interconnection of similar and dissimilar LANs. A number of references have dealt with the design issues and performance analysis of various interconnected LANs. For instance, Ho and Mukherjee[7] present a token ring local area network capable of operating in multiple parallel segments via a central switch (called a bridge). It shows the performance difference between a single ring and segmented rings via bridges with different load. The performance of two token ring LANs , interconnected by a bridge, is analyzed by Bux[5]. An important issue is to obtain the advantages of interconnection without losing possible accuracy [4]. Bijan et al.[9] describe the performance degradation in heterogeneous local computer network interconnections.

Since the bridge contends for access to the LAN as a single station, one bridge may "represent" many nodes on another LAN. Thus, sometimes it is still very important for us to concentrate on a single subnet.

The IEEE 802.4 token-passing bus scheme[1] has emerged as the standard for factory floor local-area networks due to several major reasons such as its deterministic nature, reliability at peak loads, and its support of a message based priority mechanism. The priority mechanism handles multiple classes of data. It allocates the channel bandwidth among different priority classes by means of a set of timers.

Many approaches have been proposed to provide access control for token networks which support multi priority messages.

Jayasumana[10] describes a priority mechanism in which the protocol allocates the channel bandwidth among different priority classes of messages by means of a set of timers at each station. Tobagi[2] describes the throughput analysis of timer controlled token passing protocols under heavy loads. In his model, each station transmits the same amount of synchronous data when the token reaches the station, regardless of the time elapsed from the previous token arrival. This assumption implies that the synchronous throughput is proportional to the token speed.

Minet[8] evaluated the performances of two different protocols, SAE LTPB (Linear Token Pass Bus) and IEEE 802.4 token bus. The comparison has been accomplished by theoretical analysis and simulation experiment based on the different timer control mechanisms between these two protocols.

The same problem is studied by Valenzana[3] from a different point of view. In his model, each station transmits synchronous traffic depending upon the time spent in the previous token rotation, so that the amount of synchronous traffic is constant and does not depend on token rotation speed. Furthermore, transmission of high priority message also depends upon the channel bandwidth assigned to it.

1.2 Purpose of This Study

Since the simulation model is a extended token bus, two subnets connected by a bridge, there are two types of traffic. One is the inter-net or inter-network traffic which goes through the bridge. The other is the intra-net or intra-network traffic which only flows inside the subnet. In this work, we are only interested in the delay characteristics of the high priority messages which travel inside the intra-net and inter-net with two different timer control schemes.

Two different timer control schemes have been examined for the message delay analysis. One is the ordinary priority timer control, and the other is an adaptive priority timer control. In the first one, the token holding time for the highest priority message is fixed, whereas, in the second, it is varied based on the ratio of local load to global load.

The asymmetric load distribution among stations has been set up for the simulation. Eight different types of message delay have been evaluated, and these are:

1. Average global high priority message delay in intra-net
2. Average global high priority message delay in inter-net
3. Average lightly loaded high priority message delay in intra-net
4. Average lightly loaded high priority message delay in intra-net
5. Average global low priority message delay in intra-net
6. Average global low priority message delay in inter-net
7. Average lightly loaded low priority message delay in intra-net
8. Average lightly loaded low priority message delay in intra-net

The objectives of this study are as follows:

1. To study the performance of an extended token based Local Area Network with multiple priority messages by simulation.
2. To compare these two different services with different load distribution.

1.3 Outline

An outline of the rest of this thesis is as follows:

Chapter 2 presents a brief theory of token based Local Area Networks interconnected by a bridge. Chapter 3 describes in detail the simulation model for an extended LAN supporting two levels of priority messages. Chapter 4 plots the simulation results with discussions. And Chapter 5 provides the conclusion.

Appendix A gives out one of input models.

Chapter 2

Extended Token Based Local Area Networks with Priority Schema

In this section, the basic concepts of the IEEE 802.4 protocol and extended network are presented. And two different types of timer controls are also described with the discussion of the effect of asymmetric load distribution. Two subnets connected by a bridge is the model set up in our simulation. Due to the features of that kind model, the performance of each subnet which determined by the parameter setting of the subnet will mainly determine the performance of the whole network. Thus, we still concentrate on the study of the subnet.

2.1 Token Bus Network

Industrial communication systems have been growing at a rapid rate in order to fulfill the growing demands for computer aided management, engineering and manufacturing of products. The Manufacturing Automation Protocol (MAP), based on the OSI seven layer model, is the basic frame work for the design and development of computer communication protocols and equipment in support of industrial automation. The IEEE 802.4 token passing bus protocol defines the physical and the data link layers of the MAP using base/broadband communication over a bus topology. The token passing bus is a protocol which offers distributed and flexible medium assignment, robustness, and real-time capabilities using a timed token-passing scheme.

The Medium Access Control (MAC) layer uses a broadcast protocol where the task of initialization and maintenance of channel assignment is distributed equally among all the attached stations. The access to the shared physical medium is regulated by a special packet known as the token and only the current token holder can transmit on the bus. At the end of the transmission it passes the token to the next station in a logical ring which consists of all the stations in the network.

2.2 Priority Schemes with Timer Control

As we discussed previously, each station belongs to the logical ring over which circulates the token. Each station has two neighbors on the logical ring, a predecessor and a successor. A station receives the token from its predecessor. The token represents the right to transmit data messages. Upon completion of data transmission a station passes the token to its successor.

Since IEEE 802.4 targets real time message transportation, it provides the priority option for the message transmission control. With the priority option, the medium access control (MAC) sublayer of 802.4 offers four levels of priority classes, denoted by 0 (lowest priority), 2, 4, 6 (highest priority). When a station receives the token, it starts transmitting frames of its highest priority class. The rule governing the highest priority frames is that a station shall not transmit consecutive frames for more than some maximum time, called the high priority token holding time, noted as $HPTHT$. After sending the highest priority frames, it starts servicing the queue of the next access class. Each of the three lower access classes at a node is assigned target token rotation time, noted as $TTRT$. The station is allowed to send frames of that particular access class until the $TTRT$ expires. The fraction of bandwidth that will be allocated to each class is controlled by $TTRT$ of the access classes.

Although the timer controlled token passing protocol has interesting properties for allocating the bandwidth for each class of messages, it is difficult to use it in a real situation, because of its difficulty of setting the values of the target token rotation time for each class, and in particular the value of $TTRT$. If the $TTRT$ is too large, the maximum token rotation time becomes large, worsening the responsiveness of the network.

2.3 Extended Local Area Network

Individual local networks may be interconnected to form a bridged local network. The bridging of local networks may be desirable for a variety of reasons, including improved performance, signal quality, and availability, as well as connectivity when individual local networks are already in existence.

2.3.1 Properties of Bridge

Bridges connecting LANs have several useful properties:

1. Bridges isolate LANs from traffic which does not need to traverse that particular LAN.

2. LANs are limited in physical extent by either propagation delay or signal attenuation and distortion. Since the bridge is a store-and-forward device,

it forwards frames after gaining access to the appropriate LAN via the normal access method.

3. Some LAN architectures support a variety of physical media which cannot be directly connected at the physical layer. Bridges allow these media to co-exist in the same extended LAN.

4. It is possible to build a bridge serving its LANs which are dissimilar.

5. Because of physical layer limitations or stability and delay considerations, most LAN architectures have a practical limit on the number of stations on a single LAN. Since the bridge contends for access to the LAN as a single station, one bridge can "represents" many nodes on another LAN.

2.3.2 Desirable Characteristics

There are number of characteristics which an ideal extended LAN should possess. These include:

1. Only traffic generated by user stations should exist on the individual LANs (i.e. no traffic resulting from complex routing algorithms). Further, this traffic should traverse only those LANs necessary to best reach its destination.

2. The bridge should not cause duplicate frames to be delivered to the destination.

3. The combination of LANs and bridges should not alter the frame ordering as transmitted by the source station.

4. In the LAN environment, users expect high throughput and low delay. The extended LAN should preserve these characteristics.

5. Frames should not be allowed to exist in the extended LAN for an unbounded time.

2.4 Two Different Schemes of Timer Control

Timer controlled token passing schemes allocate the bandwidth for each class by assigning a token holding time to each class of messages. In addition to $TTRT$, the target token rotation time, there is another important parameter, the high priority token holding time, which is assigned to the highest priority message. Obviously, the way on which the timer parameters assigned to each class, will directly affect the message delay of that class, and indirectly those of other classes.

In this study, two classes of priority messages, high priority and low priority messages are assumed and two different schemes have been considered for controlling the transmission of high priority messages. These two schemes are named as ordinary timer control and adaptive timer control.

2.4.1 Ordinary Timer Control

In the first approach, the high priority token holding time at each station is set to a constant value T_s and the target token rotation time for low priority traffic at each station is set to a constant value $T_{R(l)}$. A high priority token holding time ($HPTHT$) for a station represents a guaranteed amount of time that the station may use the channel every time it receives the token, and it loads the "token hold timer" with this value. Every station transmits high priority messages until either the token hold timer expires or the messages in the queue are exhausted. Whenever the messages are not exhausted, the station will attempt to transmit more messages, provided the token hold timer has not expired. If the token hold timer expires during the transmission of the current message, the station will continue message transmission until the whole of the current message has been sent before it transfers control to the next access lower priority class.

With the priority option, we can treat each class of message queue as a virtue station in a processing element. The Target Token Rotation Time ($TTRT$) is defined for each processing element to control the transmission of low priority traffic. The processing element measures the time it takes for the token to circulate around the ring. If the token returns to the queue – virtue station in less than the target token rotation time, the processing element is allowed to send messages of low priority until the $TTRT$ has expired. If the $TTRT$ has expired by the time the token returns, it is not allowed to send any low priority messages.

With the two classes of priority messages, the above assumptions lead to the following protocol rules for the model:

$$g_{c,i} = min(Tr_{h\_message}, HPTHT) \tag{2.1}$$

$$a_{c,i} = max(0, TTRT - TRT_i - g_{c,i}) \tag{2.2}$$

where

$g_{c,i}$ = time spent to transmit high priority message on the c^{th} visit of the token to the i^{th} station.

$Tr_{h\_message}$ = the transmission time spent on the highest priority message in the queue.

$a_{c,i}$ = time spent to transmit low priority message on the c^{th} visit of the token to the i^{th} station.

$HPTHT$ = time threshold for high priority message transmission.

$TTRT$ = time threshold for low priority message transmission.

TRT_i = token rotation time in the i^{th} visit for the virtual station of low priority class, and it is defined as the time interval between two consequent visits to that visual station.

2.4.2 The Effect of Asymmetric Load Distribution

In this subsection, the effect of asymmetric traffic on the performance of the network with priority options is discussed. An asymmetric traffic describes an uneven load distribution among stations attached to the network. It can be seen that for a certain time period, some stations of the network are lightly loaded while some are heavily loaded. Let's assume that there are an average P packets in the queue at the lightly loaded station and Q packets in the queue at the heavily loaded station when the station catching the token at moderate load. So that $P << Q$, for asymmetric traffic. When we treat the queue of each class message as virtual station, the definition of asymmetric load distribution above can be applied to the high priority message queues in the stations. In this study, we are interested only in the high priority message delay.

The ordinary timer control can guarantee allocation of a certain amount of bandwidth to high priority message with fixed high priority token holding time. However, since the amount of the messages of that class can be transferred in one token rotation are fixed, most of time is spent in rotating the token at moderate load. Therefore, with asymmetric load distribution, the size of high priority queues in highly loaded stations increases sharply. This increases overall delay (global delay) of high priority messages in the network. The more asymmetric the load distribution, the higher the global delay of the network.

2.4.3 Adaptive Timer Control

In order to reduce the high priority message delay of the network under asymmetric load distribution, we proposed a second scheme, adaptive timer control. Under the adaptive timer control, the $HPTHT$ for high priority message transmission is not constant. Instead, the high priority token holding time for each station is proportional to the ratio of the local load to the global load. The value of the local load is expressed by the time needed to transmit the high priority messages which are queued in the station while the value of global load is expressed by the time spent by the all other stations in the logical token ring to transmit their messages which include high priority as well as low priority messages. By this way, the adaptive timer control dynamically changes the token holding time for high priority messages at different stations at different times.

Then the equations above should be rewritten as:

$$g_{c,i} = min(Tr_{h\_message}, S * HPTHT) \tag{2.3}$$

$$a_{c,i} = max(0, TTRT - TRT_i - g_{c,i}) \tag{2.4}$$

where

S = the ratio of the local load to the global load.

Chapter 3

Configuration and Simulation of a Network Model

This chapter begins with the description of the model simulated. Then, it provides the assumptions of the simulated model. At last, how the measurement methods of the simulation experiments are presented.

3.1 Input Model

In LANSF [6] the input data set describes the simulated model by a number of logically separate sections. The data file starts with the time section followed by the configuration, traffic, protocol specific, and exit section. A sample input data file is given in Appendix A

3.1.1 Time Section

The time section specifies the number of indivisible time units (ITUs) in the experiment time units (ETUs). In our simulation model, channel (link) capacity is defined as 10 Mbps. In our model, for simplicity, ETU is defined as 10^7 ITU. Therefore, it is quite clear how to reference the other parameters calculated to ETU.

3.1.2 Configuration Section

The configuration section defines the network backbone as follows:

1. Number of stations
2. Port allocation
3. Number of links
4. Port assignment
5. Distance matrix

The simulation model diagram is shown in Fig. 3.1:

The model can be clearly described with the links and the stations inside the system, and the way on which the stations are attached to those links. The whole system has totally four links, link0, link1, link2, and link3. We define link0 and link3 as broadcast ether-type (coaxial cable like) links corresponding to a single, uniform, bidirectional communication channel. Whereas link1 and link2 are defined as unidirection links. All sixteen stations in the system are numbered from 0 to 15. The model simulated consists of two token bus subnets viewed as upper subnet and lower subnet in Fig. 3.1. These two subnets are interconnected by a bridge through two bridge stations, station7 and station8. There is one bidirectional port (input and output) for each station, by which, stations are connected to link 0 and link3 in their own subnet. Each bridge station has two additional unidirection ports which hooked to link1 and link2. One port is used for input and the other for output. Distances among stations are expressed as the time of propagation for a signal from one port to the other in the units of time needed in transmitting a single bit.

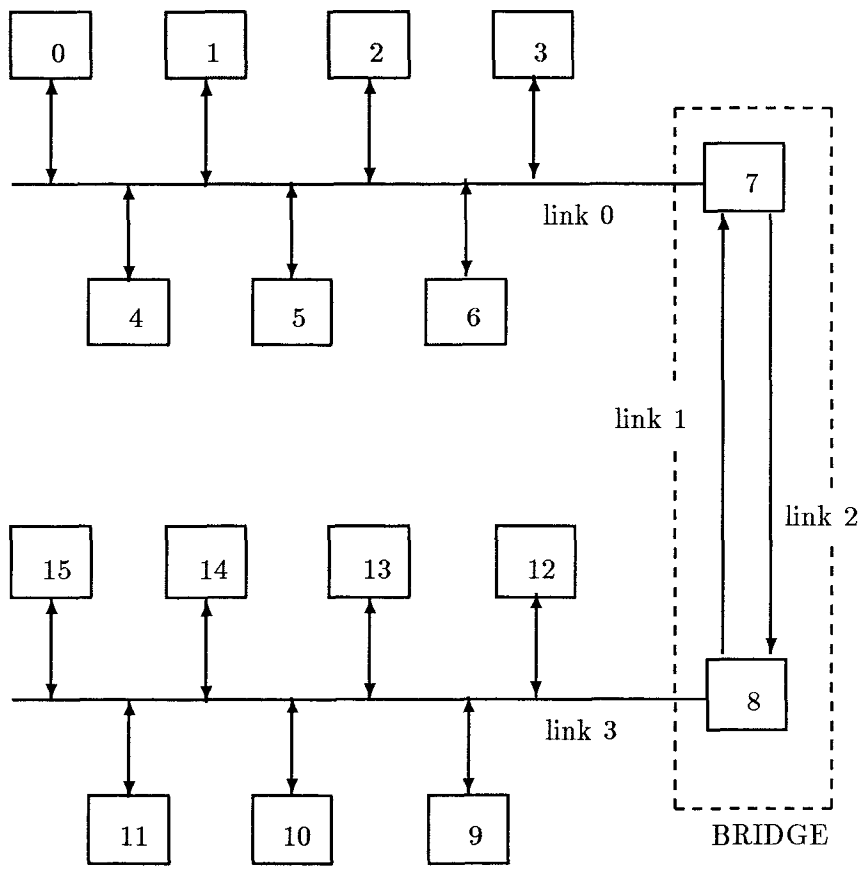

Figure 3.1: Simulation Model depicting the two LANs and the connecting bridge

3.1.3 Traffic Section

The traffic pattern is specified as a set of message types, each message type representing a class of messages generated according to the parameters. These parameters are:

1. Options

2. Interarrival time

3. Message length

4. Number of senders and receivers with their weight

The options determine if the message generated is burst or nonburst and message interarrival time follows uniform or exponential distribution. The actual load offered by a certain type of message on a given link is related to the interarrival time, message length, and the relevant senders defined in the message.

3.1.4 Protocol-specific and Exit Sections

In this section, protocol-specific values like packet length, header and trailer information, token length, interpacket space, target token holding time and high priority token holding time are defined. We have done simulations for fixed size of packets with 160 bits in the header and trailer. Token length and inter packet space are specified as 160 bits and 16 bits respectively.

The exit section describes the stop conditions for the simulation. Three limits can be declared to exit simulation.

1. Maximum number of messages

2. Virtual time limit

3. CPU time limit

We had done each simulation for total of 10,000 messages on the network.

3.2 Assumption of the Simulation Model

The software package LANSF is employed to simulate the network model. We assume that the network model simulated has no station fault, no modification of the logical ring, no token loss. Also, each station in each subnet is assumed to have the same characteristics. Based on the assumptions above, we set up the load distribution, the percentage load dedicated to each station, and provide the pseudo code for implement the second scheme described previously.

3.2.1 Load Distribution

At first, we define four basic types of messages as follows:

- L_{bu} is type of the message for which senders and receivers are in the upper subnet.
- L_{bl} is type of the message which has both senders and receivers in the lower subnet.
- L_{bgu} is type of the message which assumes that the senders are in the upper side of the subnet and the receivers are in the side subnet.
- L_{bgl} is type of the message which is characterized by senders in the lower side subnet and receivers in the upper side subnet.

Each of these types has both high priority messages and low priority messages. The first two items are termed as intra-network traffic while the other two are defined as inter-network traffic.

In a single subnet, there are two types of traffic load distributions, symmetric and asymmetric. *Symmetric traffic load* means that every station in the network generates approximately the same amount of messages. *Asymmetric traffic load* means that some of the stations in the network generate more messages to be sent than some other stations in the net. For simplicity, we only assume that one station in the subnet, for example in the left side token bus, station 0 generates n percent of total traffic load, and the remaining load is submitted equally by station i (i=1,2,..,6). In our simulation, the five possible load distributions are examined and they are listed in the following table:

	st0	st1	st2	st3	st4	st5	st6	st7
I	84%	1%	1%	1%	1%	1%	1%	10%
II	78%	2%	2%	2%	2%	2%	2%	10%
III	65%	4%	4%	4%	4%	4%	4%	11%
IV	54%	6%	6%	6%	6%	6%	6%	10%
V	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%

Table 3.1: Load Distribution on Stations

Obviously, the last one is a symmetric traffic distribution. Traffic load which originated from station 7 comes from the stations in the lower subnet.

3.2.2 Load on the link

In the simulation, the characteristics of the network are examined under a certain amount of the load on the network. Generally, we define the load of the network as the attempted load on the link of the network. As we previously described, the link0 and link3 carry the messages flowing inside the subnet whereas link1 and link2 move the messages through the bridge. In the most of cases, the amount of the messages through the bridge would be much smaller than the messages flow inside the subnet. Therefore, the load on the network set up for the simulation is defined as the load inside the subnet, that is the attempted load on the link0 (link3). The simulations are carried out under a load of link0 (link3) which is set to the following discrete numbers:

1, 2, 3, 4, 5, 6, 7 Mbits/s

3.2.3 pseudo code for adaptive service

Partial pseudo code of the program for implement the second scheme is shown as follows:

```
case GOT_TOKEN:
    Count total package length in high priority message queue
    Calculate the previous token rotate time for the virtue station
    if (timer is less than or equal to zero) then
        set high priority token timer to a fixed quantity HPTHT
    else
        deduct the ideal token rotation time from the token rotation time
        if (token rotation time larger than zero)
            adjust high priority token holding time proportion to the ratio of
            local load to global load
        else
            set high priority token timer to HPTHT with an addition of
            a given amount
        end if
    end if
end case
```

3.3 Performance Measurement

In the simulation experiments two kinds of delay measurements are carried out:

1. The *absolute message delay* of message M, denoted by $d_s(M)$, is measured as the time (in ETUs) elapsing since the message was queued at the sending station to the moment the entire message including its last packet is successfully received (`accepted`) at the destination.

2. The *absolute packet delay* of packet P, denoted by $d_p(P)$ is measured as the time (in ETUs) elapsing since the packet became ready to be transmitted (the queuing time is excluded) to the moment the entire packet is successfully received (`accepted`) at its destination.

To define the above-listed measures formally and to explain how the parameters of their distribution are computed, assume that we have a sequence of messages $M^1, \ldots, M^n$ and that message M^j consists of packets $P_1^j, \ldots, P_{k_j}^j$ with lengths $l_1^j, \ldots, l_{k_j}^j$, respectively. Let $l^j = \sum_{i=1}^{k_j} l_i^j$ denote the length of M^j. Message M^j was queued at the sender at time tq^j; its i'th packet P_i^j became ready for transmission at tt_i^j and was completely received by the target station at tr_i^j. The three delays mentioned above are calculated according to the following formulas:

$$d_s(M^j) = tr_{k_j}^j - tq^j, \tag{3.1}$$

$$d_p(P_i^j) = tr_i^j - tt_i^j. \tag{3.2}$$

The time when a packet becomes ready for transmission (tt_i^j) is determined as the maximum of the following two values:

1. the time when the buffer the packet is stored in was last released,

2. the time when the message the packet is acquired from was queued at the station.

The distribution parameters of the random variable representing the absolute message delay of multiple messages transmitted over the network are calculated assuming that the random variable consists of discrete samples, namely, the absolute message delays of particular messages. For instance, the absolute message delay for the n messages $M^1, \ldots, M^n$ is computed as:

$$d_s^a(M^1, \ldots, M^n) = \frac{\sum_{i=1}^{n} d_s(M^i)}{n}. \tag{3.3}$$

The absolute packet delay is interpreted in a similar way. Now we look at separate packets and the formula for determining the average delay is:

$$d_p^a(< P_1^j, \ldots, P_{k_j}^j >_{j=1}^{n}) = \frac{\sum_{j=1}^{n} \sum_{i=1}^{k_j} d_p(P_i^j)}{\sum_{j=1}^{n} k_j}. \tag{3.4}$$

For calculating the *absolute message delay*, it is assumed that messages are indivisible units and what only matters is the complete reception of an entire message. One can imagine a file transfer protocol in which it is illegal to use the initial portion of a partially received long file until its last packet, e.g. containing the global checksum, arrives in "good shape". In such a case, although some parts of the message being transmitted arrive at the recipient before the last packet, we cannot assume that they have been received until we get the entire message.

Finally, in certain real-time applications it is important to know how much time a station has to wait before it transmits successfully the first packet from its queue. Namely, if the station gets an urgent message to transmit, it can give it the highest priority and move it to the front of the queue. Therefore, in such applications, we need not be concerned with the message queuing time: what only matters is how fast stations are able to get their packets through. This property of a network is described by the *packet delay* which excludes the queuing time and assumes that packets are indivisible.

The standard data type `STATISTICS` offers a number of attributes which correspond to certain observed distribution parameters of random variables. These parameters are computed automatically whenever the random variable is updated. In particular, for each of the delay measures discussed above we get the following distribution parameters:

1. the number of samples (e.g. the number of packets for the packet delay, the number of bits for the weighted message delay);
2. the minimum observed delay,
3. the maximum observed delay,
4. the mean observed (average) delay,
5. the variance and standard deviation of the delay.

Chapter 4

Analysis of Simulation Results

In this chapter, we add to the description of the simulation model in the first section. Plots of the results of the simulation experiments are discussed in the second. All plots show the message delay characteristics for various loads with different load distributions.

4.1 Simulation Parameters

The simulation model has been presented in the previous chapter. There are two parameters we are going to present here. One is the size of the message, and the other is the size of the packet. The upper limit of the message size is 8192 bits and the lower limit is 2048 bits. For the packet size, we assume it may be varied from 2048 bits down to 512 bits instead of being fixed. In this way, the channel can be utilized more efficiently. The packet size described above does not include the header and trailer.

4.2 Discussion of the Simulation Results

At the beginning, we categorize all plots of the simulation results into two basic types, intra and inter-net. Each one can be separated further into four different types:

1. Lightly loaded station high priority message delay
2. Lightly loaded station low priority message delay
3. Global high priority message delay
4. Global low priority message delay

Each message delay above includes four different load distributions which are described in Table 3.1. The item V in Table 3.1 is symmetric load distribution , so it has four types of message delays:

1. Intra-net high priority message delay
2. Intra-net low priority message delay
3. Inter-net high priority message delay
4. Inter-net low priority message delay

Figs. 4.1 to 4.4 show the high priority message delays in lightly loaded stations vs. the load on the network. The percentages of the load dedicated by station 0 (or station 15) are varied from 84% to 54%. From these figures, we can see that the message delay is lower with the adaptive timer control than with the ordinary timer control. The difference is caused by the different way to allocate the bandwidth to all three types of messages, high priority messages, low priority messages, and token (or control messages). In one token rotation with the ordinary timer control, each station gets a fixed length of time, if needed, to send out high priority messages. So the bandwidth provided to high priority messages is limited. But with adaptive time control, when the global load (excluding the load in the local high priority message queue) is low and the local load in a high priority message queue is high, the station can get more time units to transmit high priority messages in order to reduce the message delay by decreasing the unnecessary time to rotate the token.

Figs. 4.5 to 4.8 show the low priority message delays in lightly loaded stations. In all these figures, we can see that when the load on the network is low, from 1 Mbits/s to 4 Mbits/s, the delays with two different timer control schemes are almost the same. With increasing load, from 5 Mbits/s to 7 Mbits/s, the delay of low priority messages with the adaptive timer control is higher than the one with ordinary timer control. This can be explained by the mechanisms of timer control schemes previously described. If the load is high, the station will get fewer chances to empty its low priority message queue in one token rotation. Thus, the delay of low priority messages is increased by waiting for the next token arrival to empty the queue.

Figs. 4.9 to 4.12 show the global high priority message delays. They show the same results as the previous four figures but with the difference that the percentage of reduced delay is varied while the load distribution in the network is changed. In the interval from 1 Mbits/s to 4 Mbits/s of network load, there is almost no variation in percentage of message delay reduction. When the load becomes high, from 5 Mbits/s to 7 Mbits/s, the percentage of the reduction in message delay has increased.

Figs. 17 to 20 show the inter-net high priority message delays in lightly loaded stations while Figs. 25 to 28 show the inter-net global high priority message delays. These figures describe the same delay characteristics as Figs. 1 to 4 and Figs. 9 to 12.

Fig. 29 shows the intra-net high priority message delay vs. the load. The results with ordinary timer control are quite close to the ones with adaptive timer control because the load is evenly distributed. Fig. 30 shows the intra-net low priority message delay vs. the load. Fig. 31 shows the inter-net high priority message delay vs. the load. From this picture, we can see that the high priority message delay is reduced when the adaptive timer control is used.

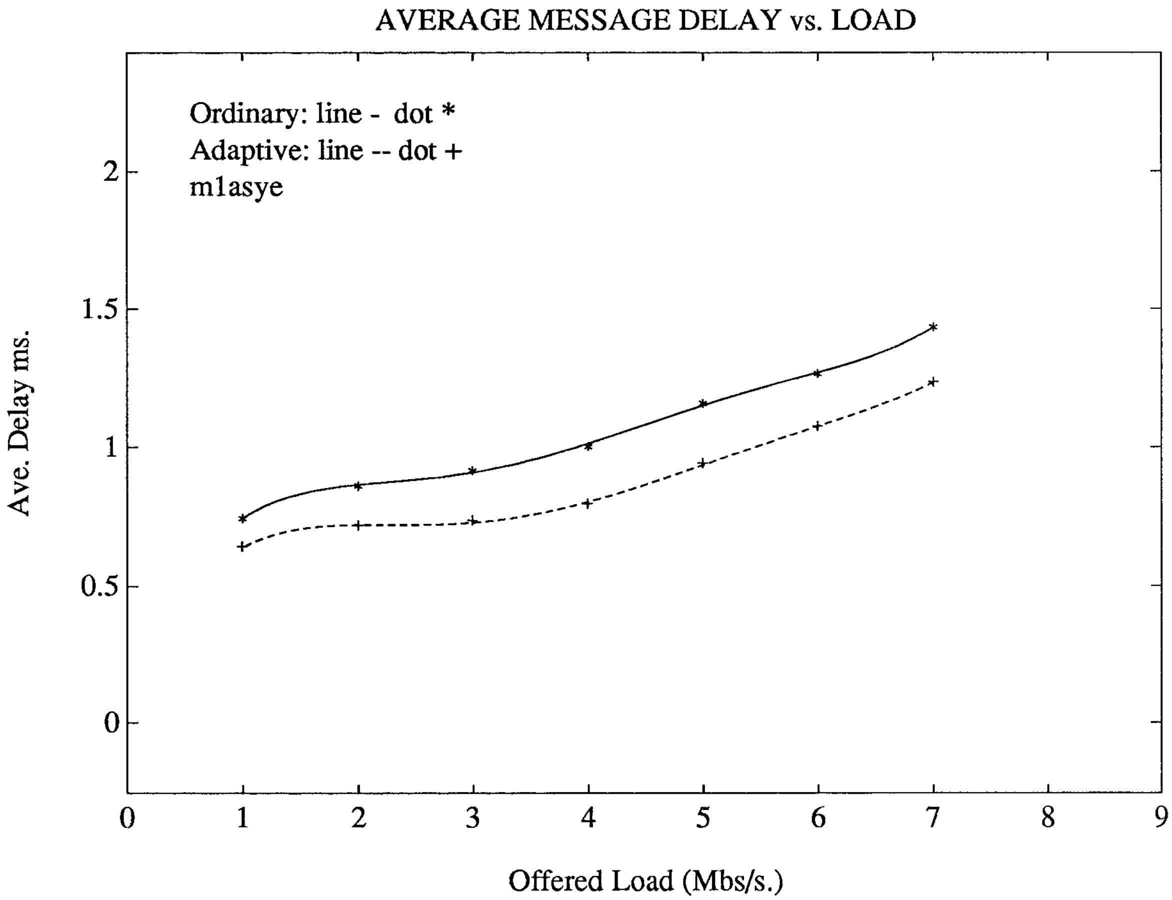

Figure 4.1: Intra High Priority Message Delay in Lightly Loaded Station with the 84% Load Distributed by Station 0

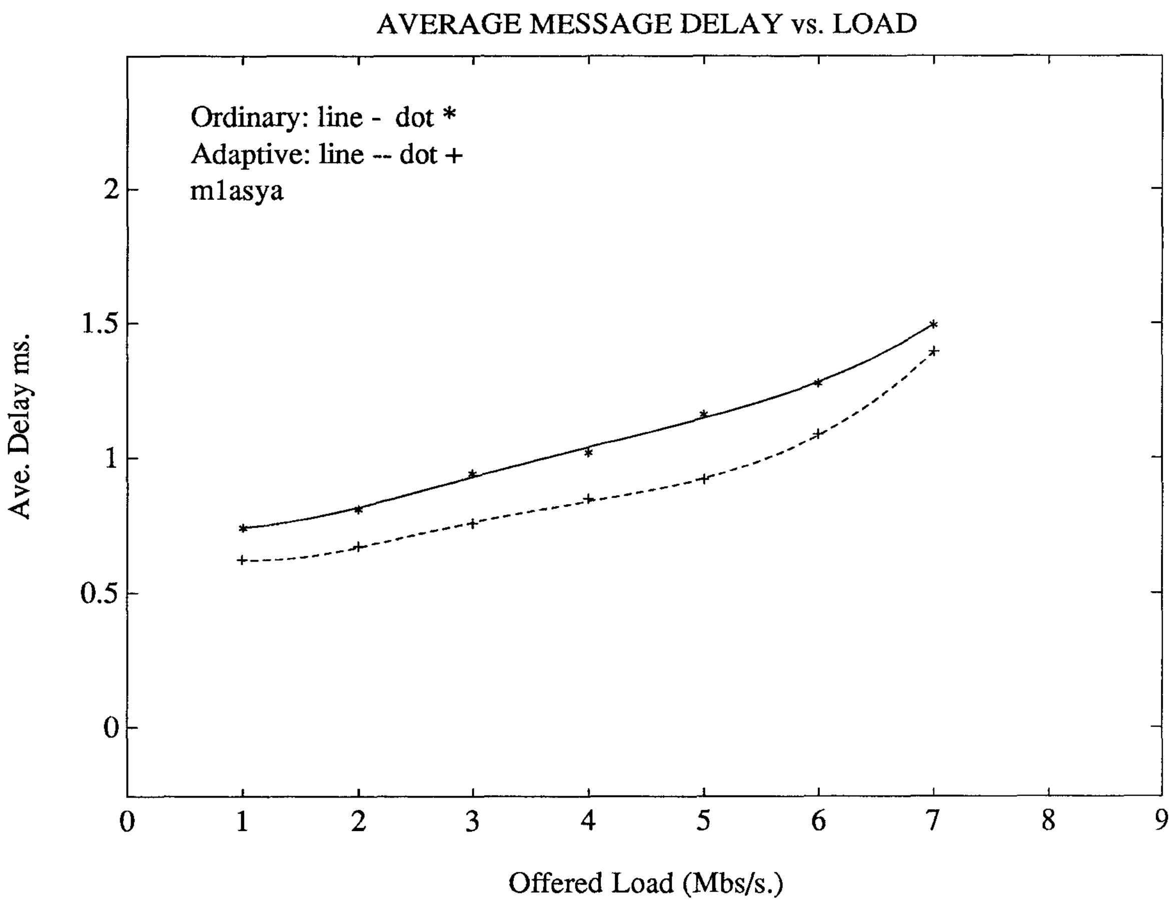

Figure 4.2: Intra High Priority Message Delay in Lightly Loaded Station with the 78% Load Distributed by Station 0

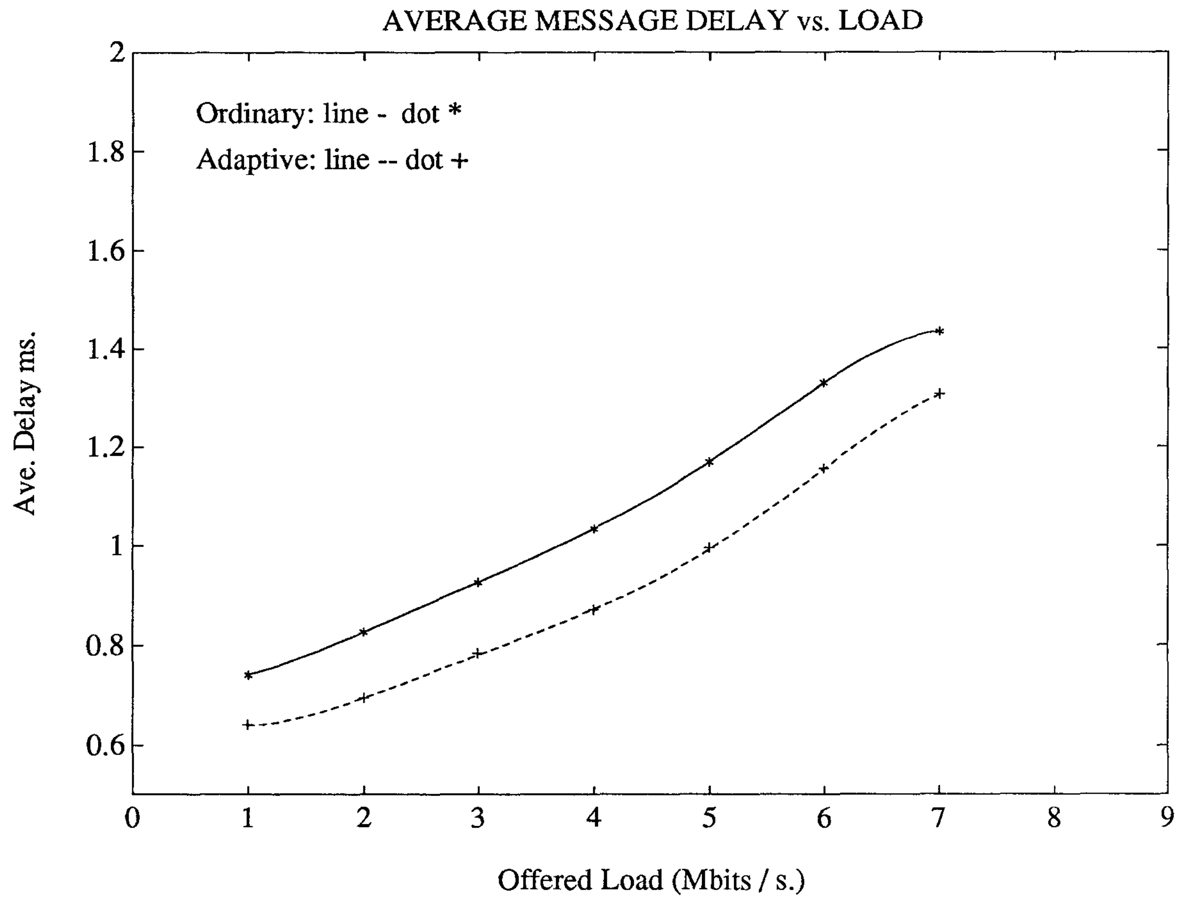

Figure 4.3: Intra High Priority Message Delay in Lightly Loaded Station with the 65% Load Distributed by Station 0

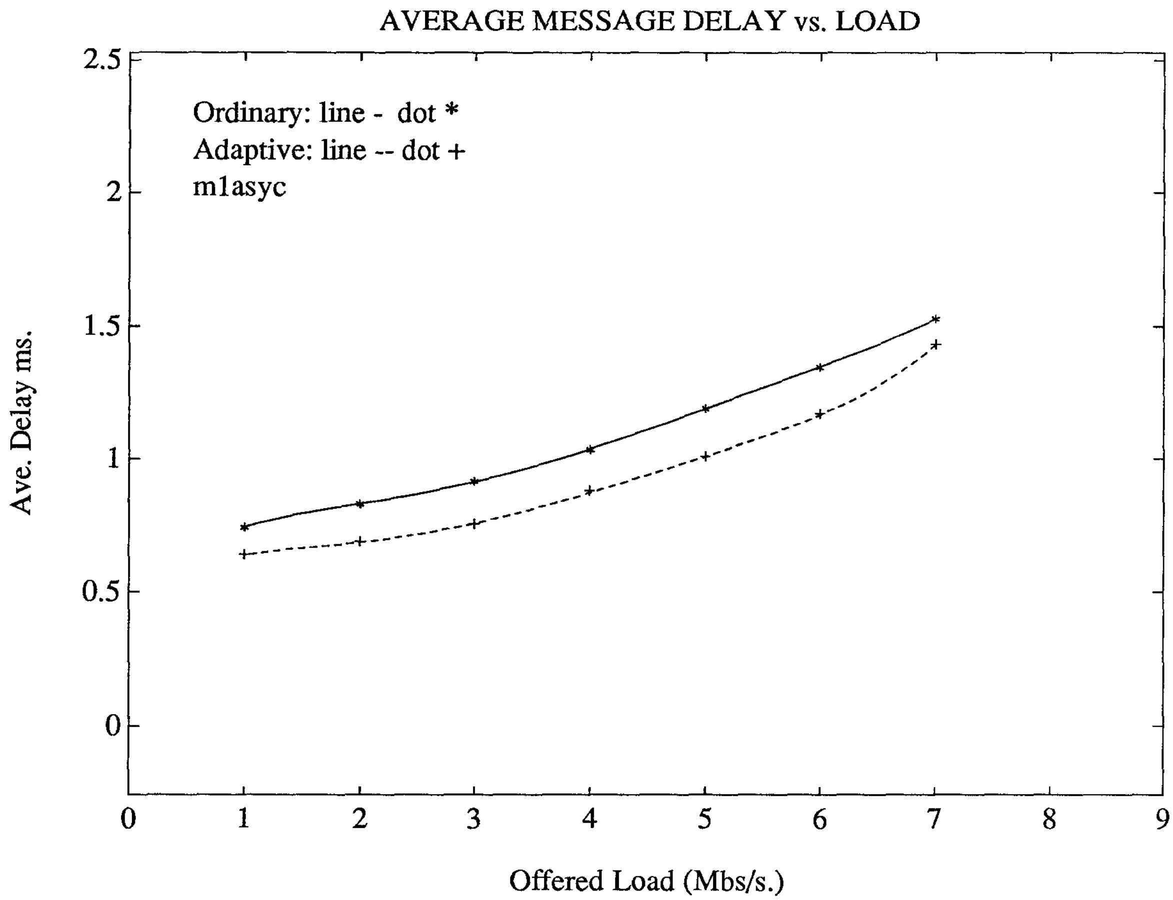

Figure 4.4: Intra High Priority Message Delay in Lightly Loaded Station with the 54% Load Distributed by Station 0

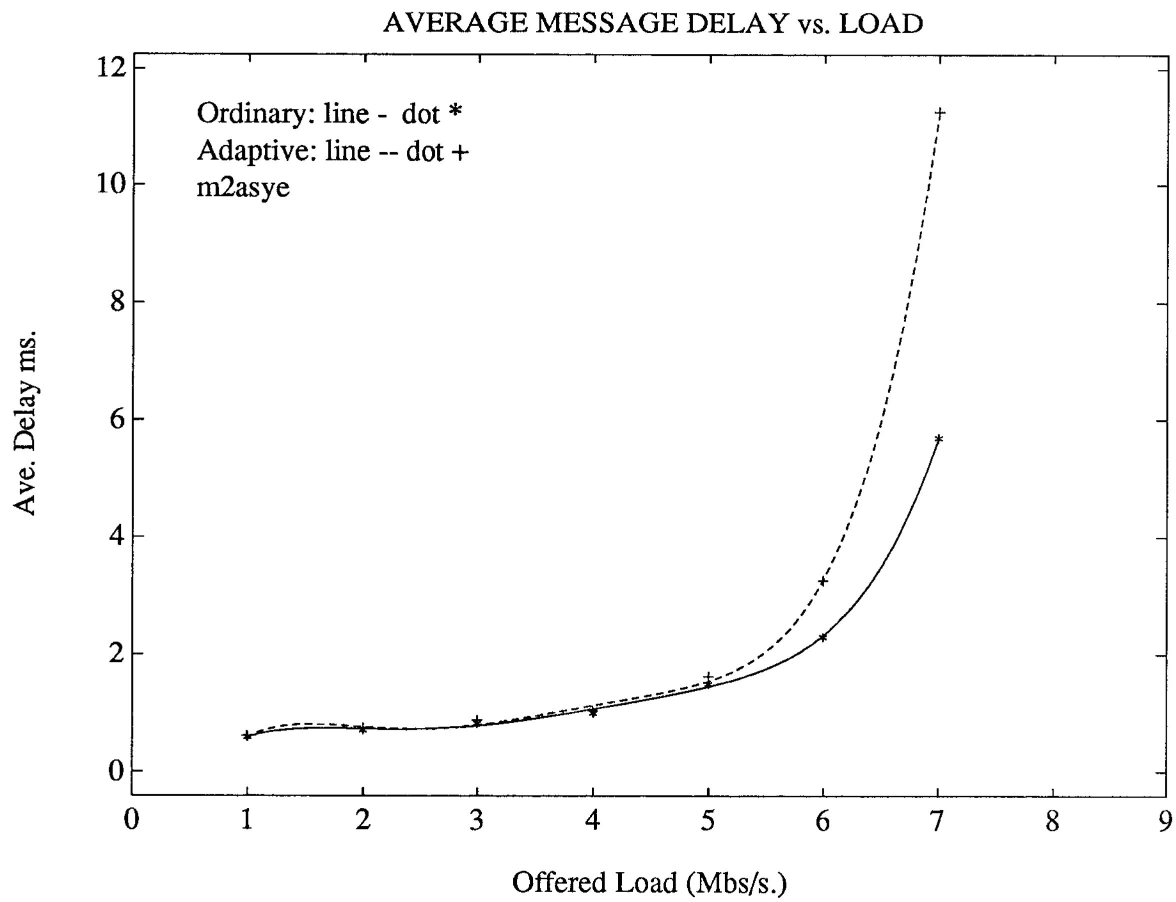

Figure 4.5: Intra Low Priority Message Delay in Lightly Loaded Station with the 84% Load Distributed by Station 0

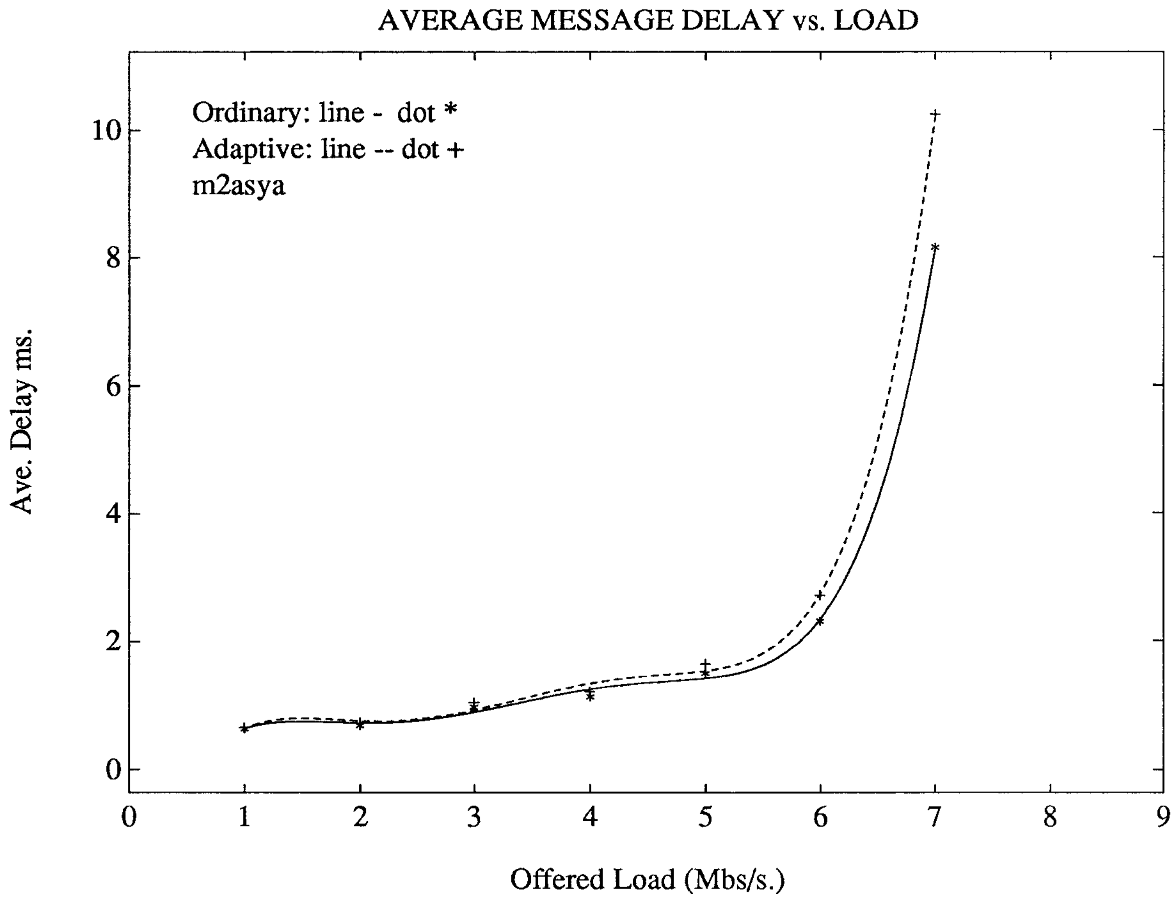

Figure 4.6: Intra Low Priority Message Delay in Lightly Loaded Station with the 78% Load Distributed by Station 0

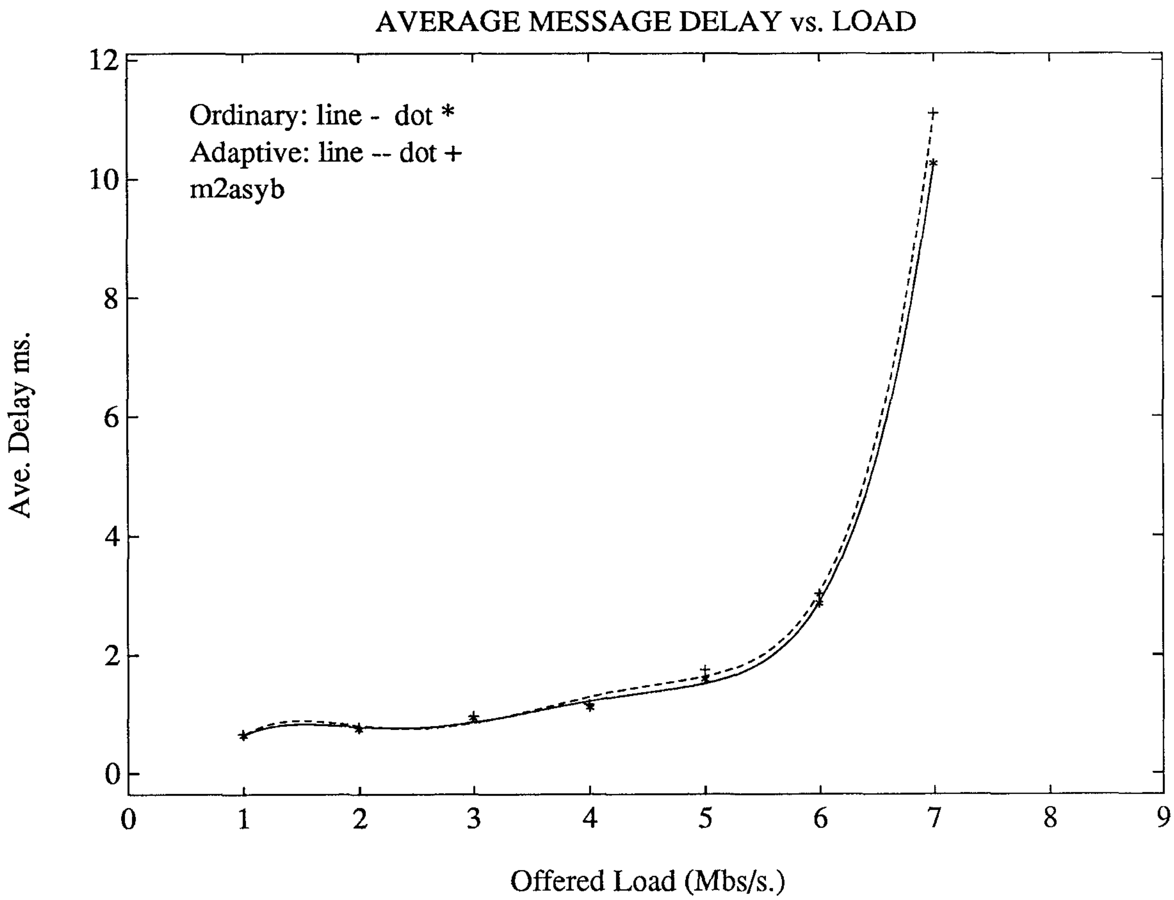

Figure 4.7: Intra Low Priority Message Delay in Lightly Loaded Station with the 65% Load Distributed by Station 0

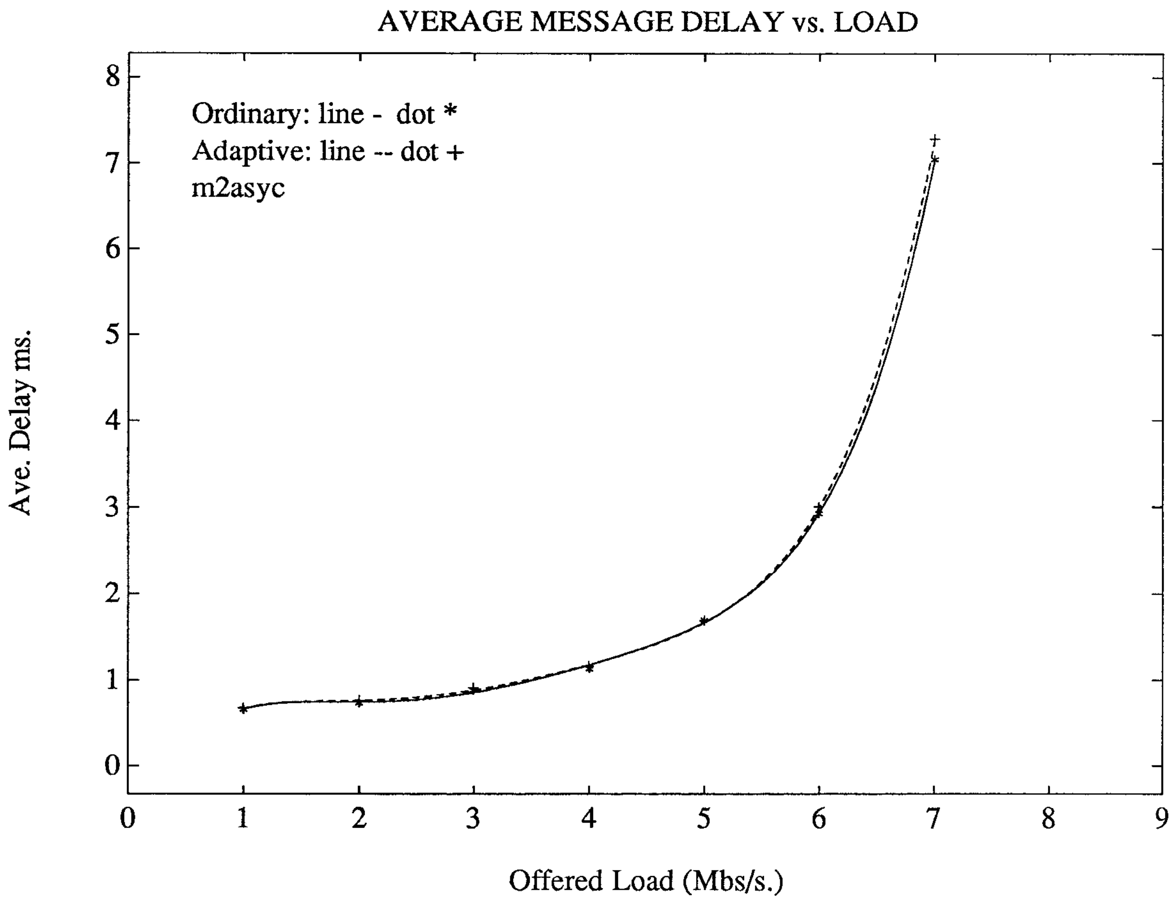

Figure 4.8: Intra Low Priority Message Delay in Lightly Loaded Station with the 54% Load Distributed by Station 0

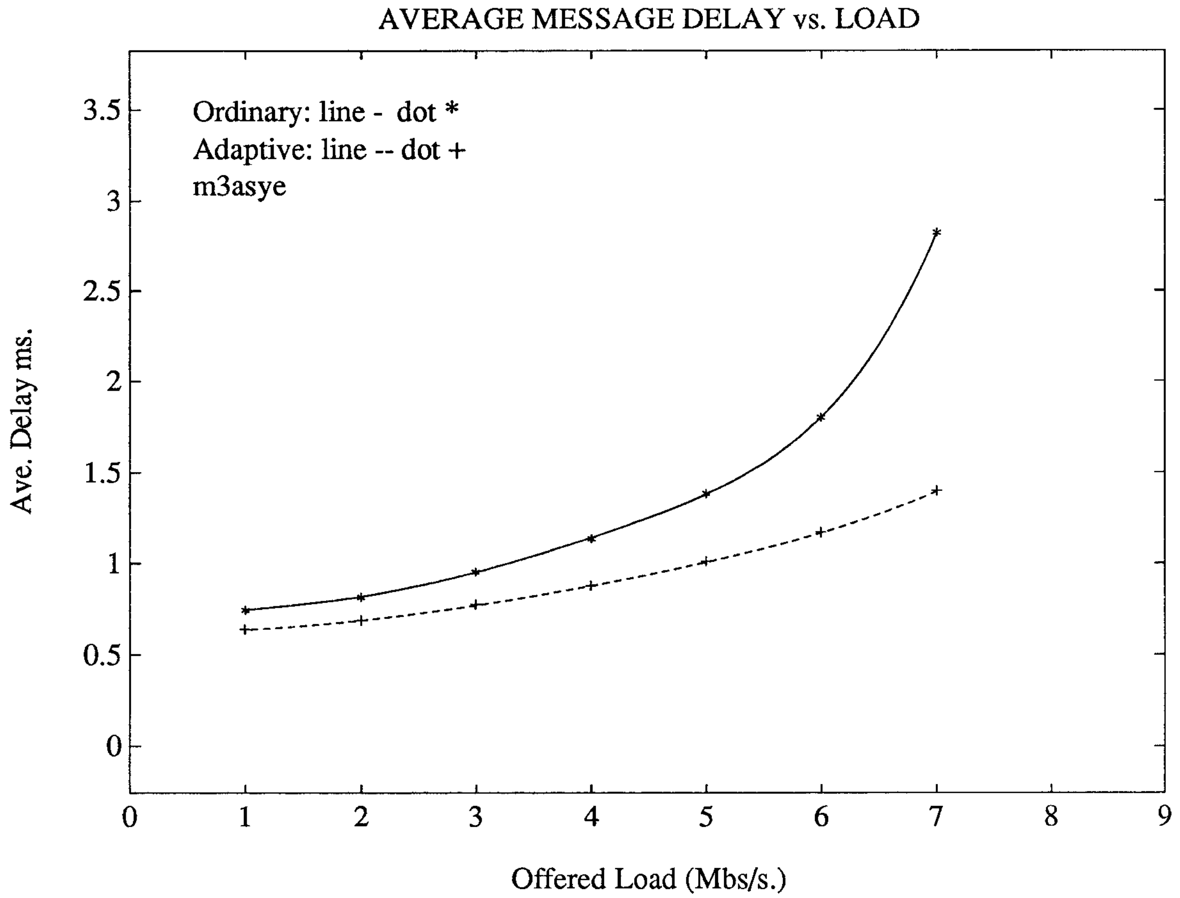

Figure 4.9: Intra Global High Priority Message Delay with the 84% Load Distributed by Station 0

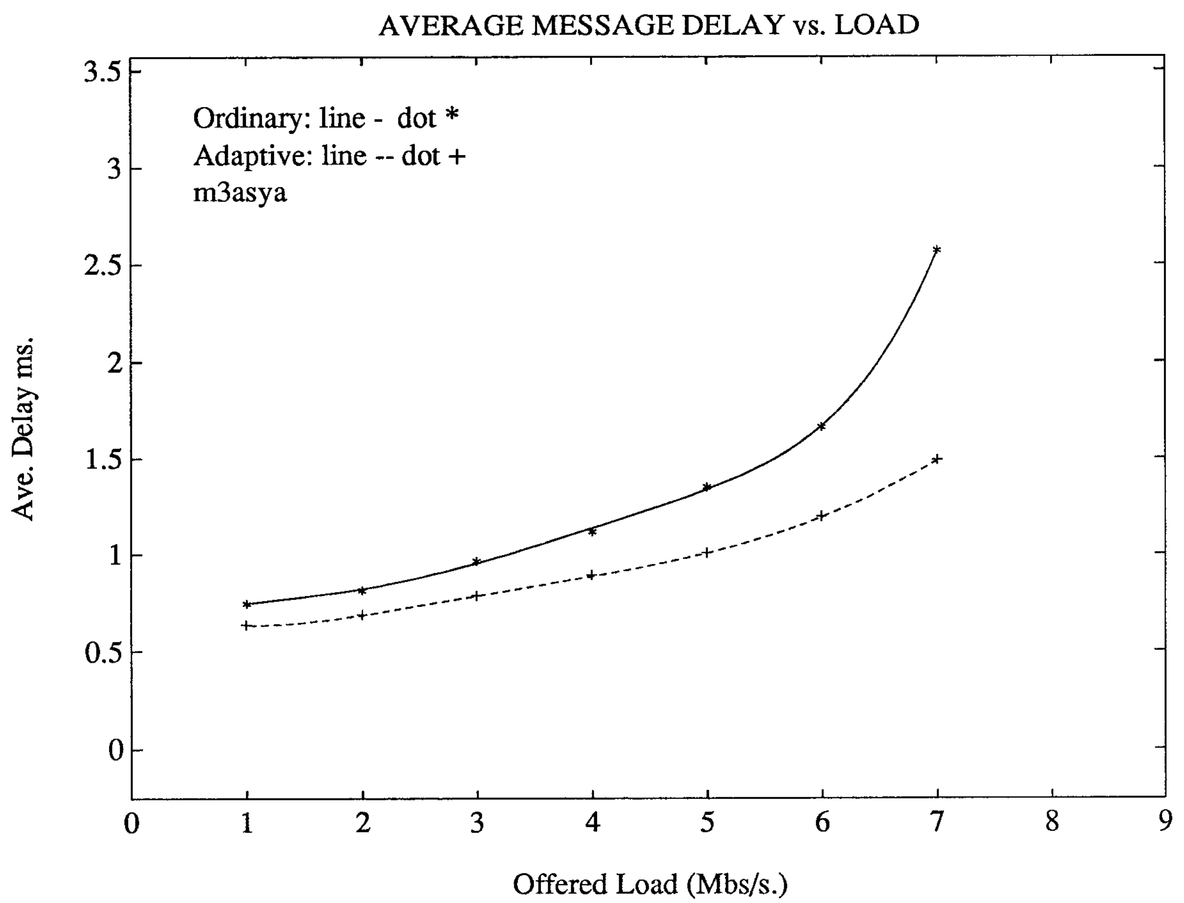

Figure 4.10: Intra Global High Priority Message Delay with the 78% Load Distributed by Station 0

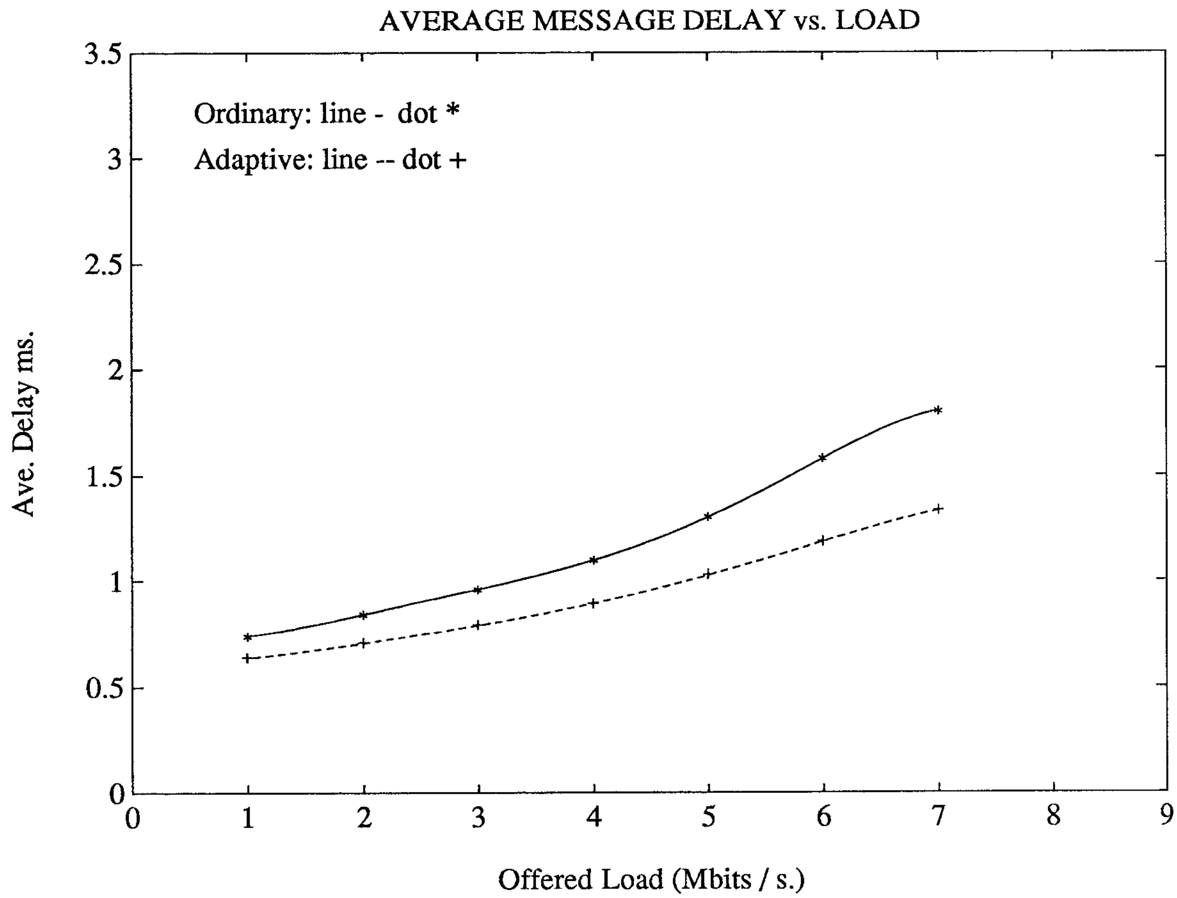

Figure 4.11: Intra Global High Priority Message Delay with the 65% Load Distributed by Station 0

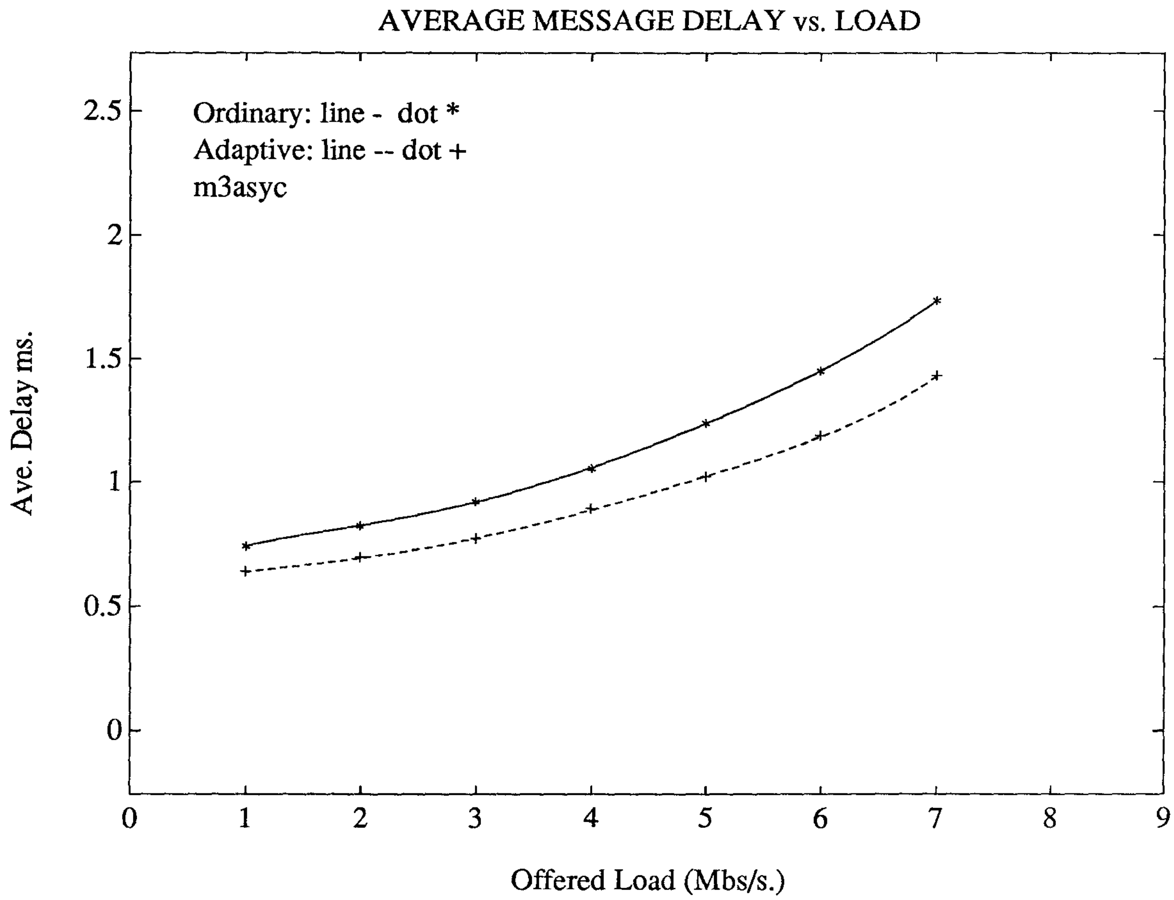

Figure 4.12: Intra Global High Priority Message Delay with the 54% Load Distributed by Station 0

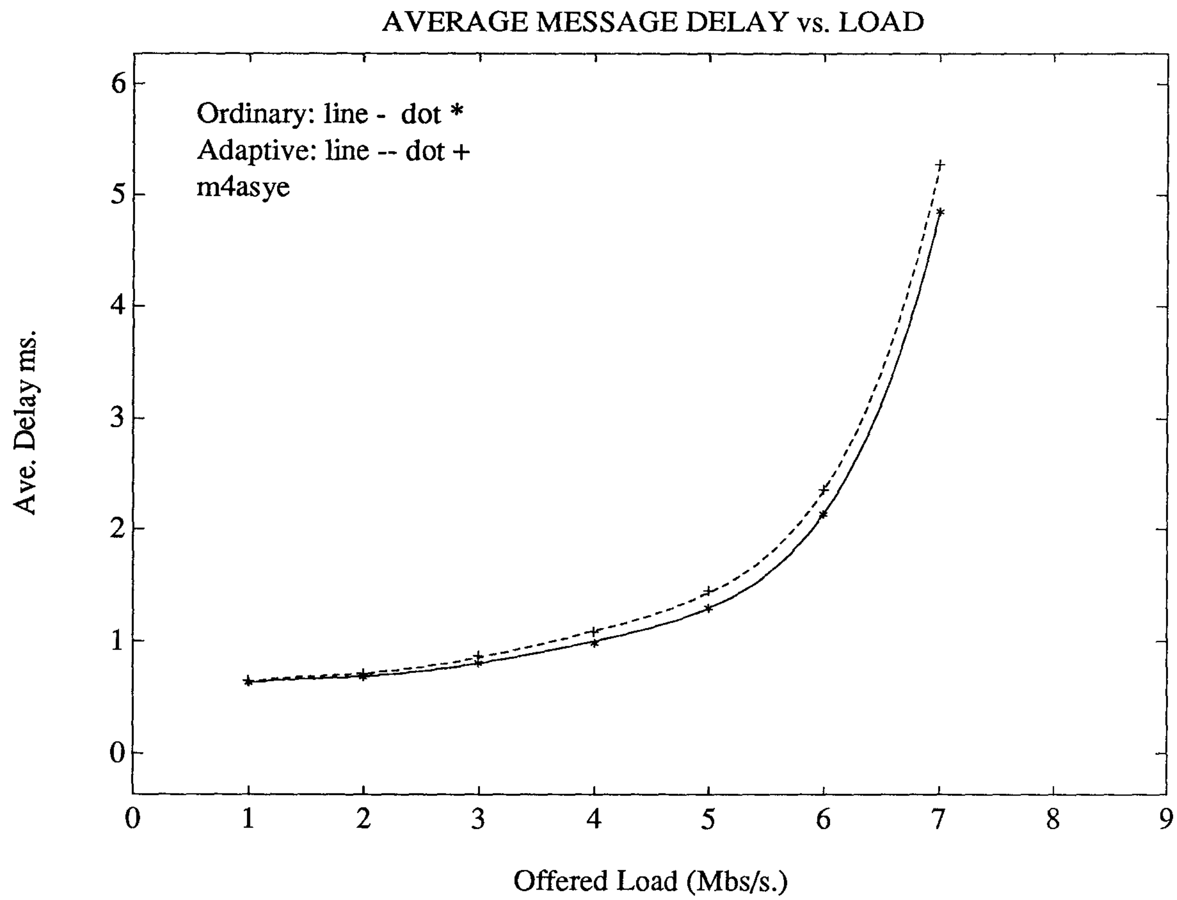

Figure 4.13: Intra Global Low Priority Message Delay with the 84% Load Distributed by Station 0

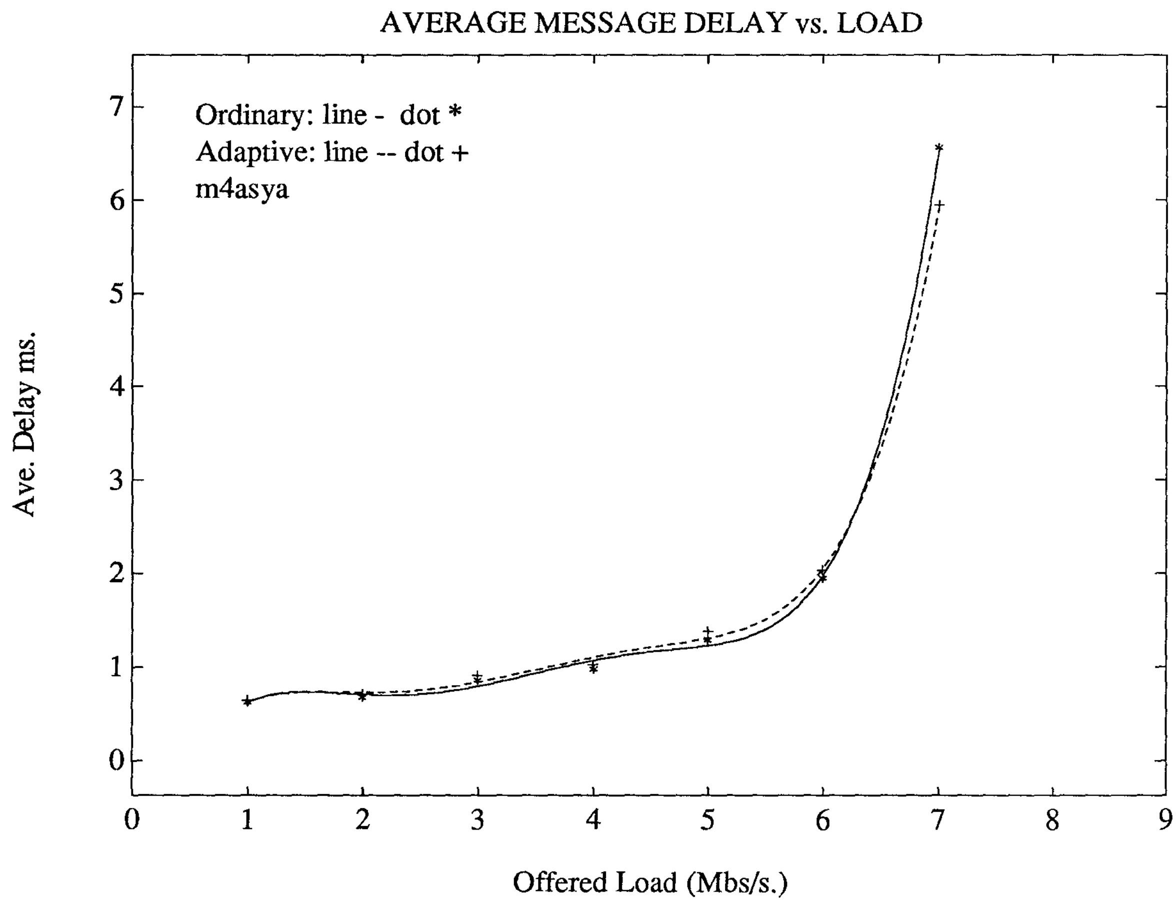

Figure 4.14: Intra Global Low Priority Message Delay with the 78% Load Distributed by Station 0

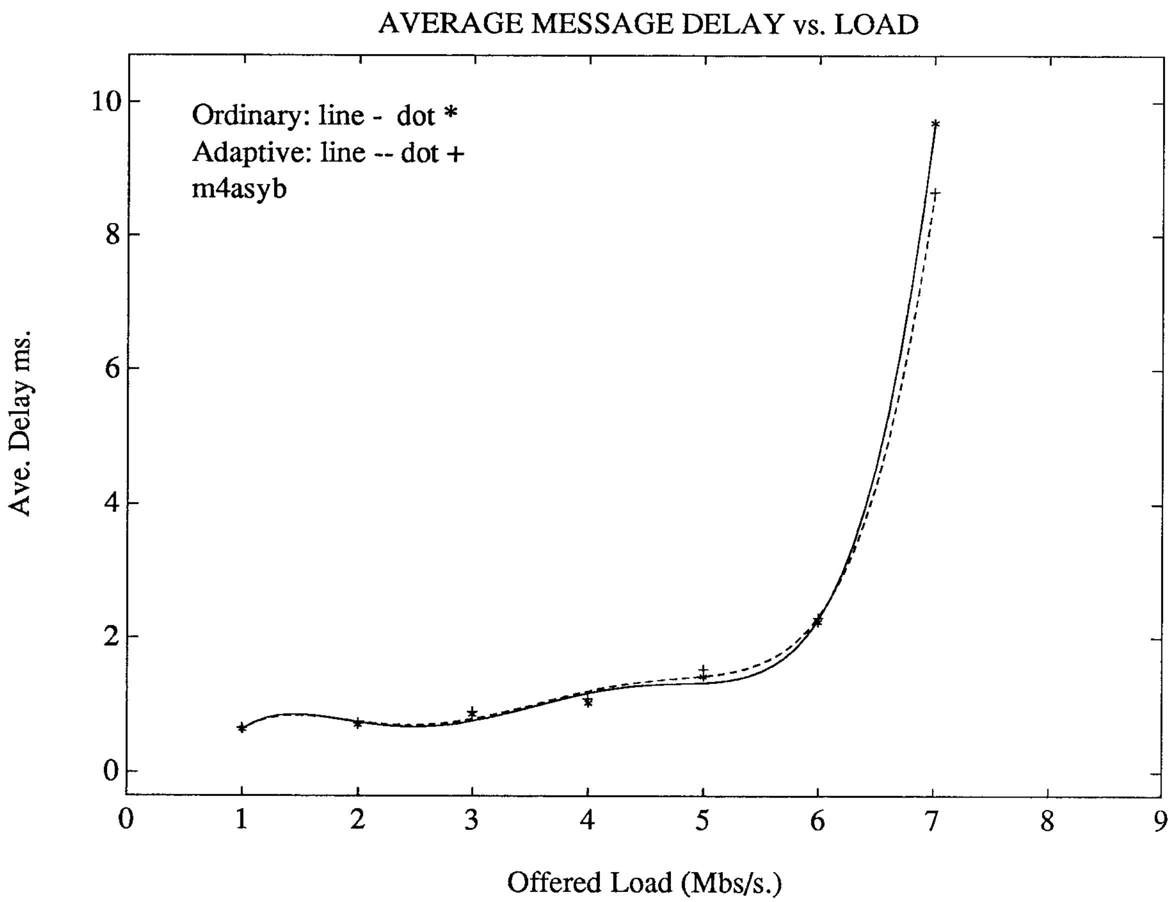

Figure 4.15: Intra Global Low Priority Message Delay with the 65% Load Distributed by Station 0

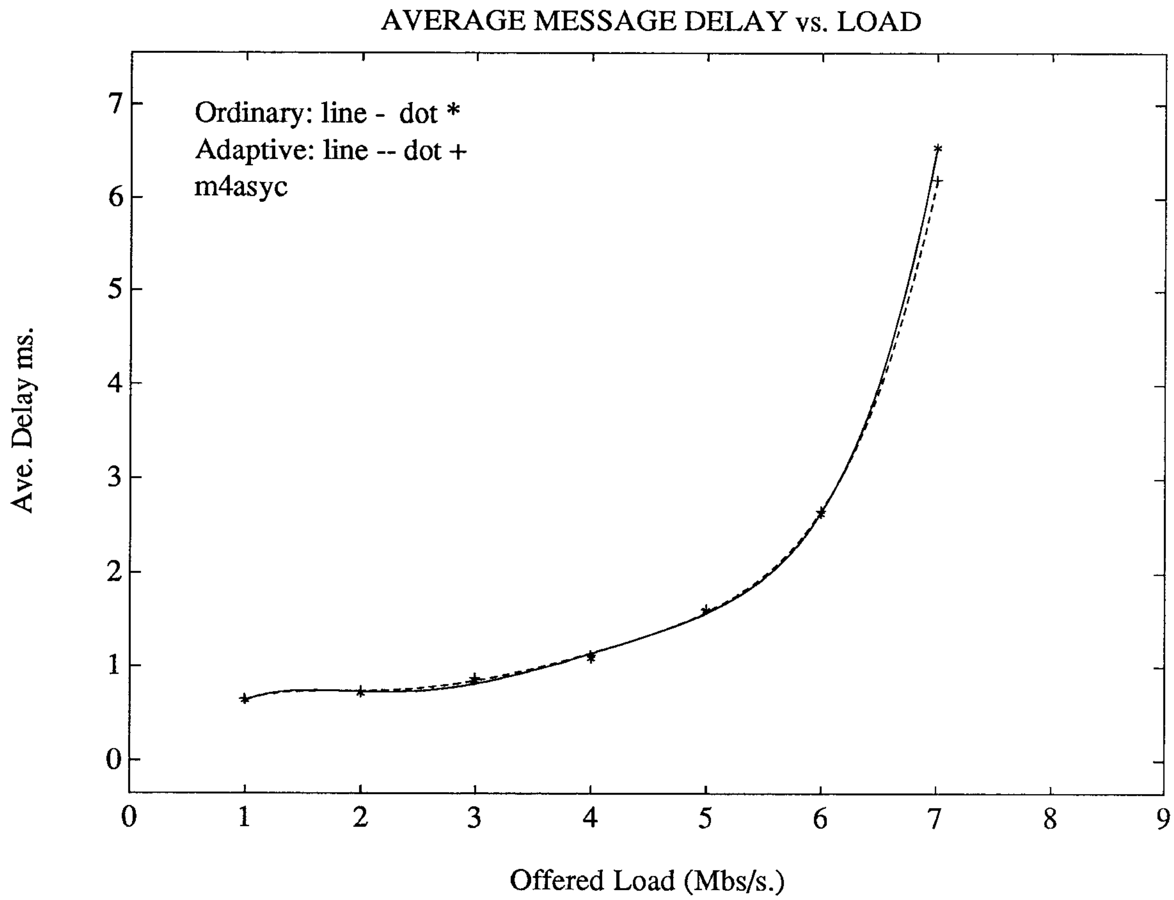

Figure 4.16: Intra Global Low Priority Message Delay with the 54% Load Distributed by Station 0

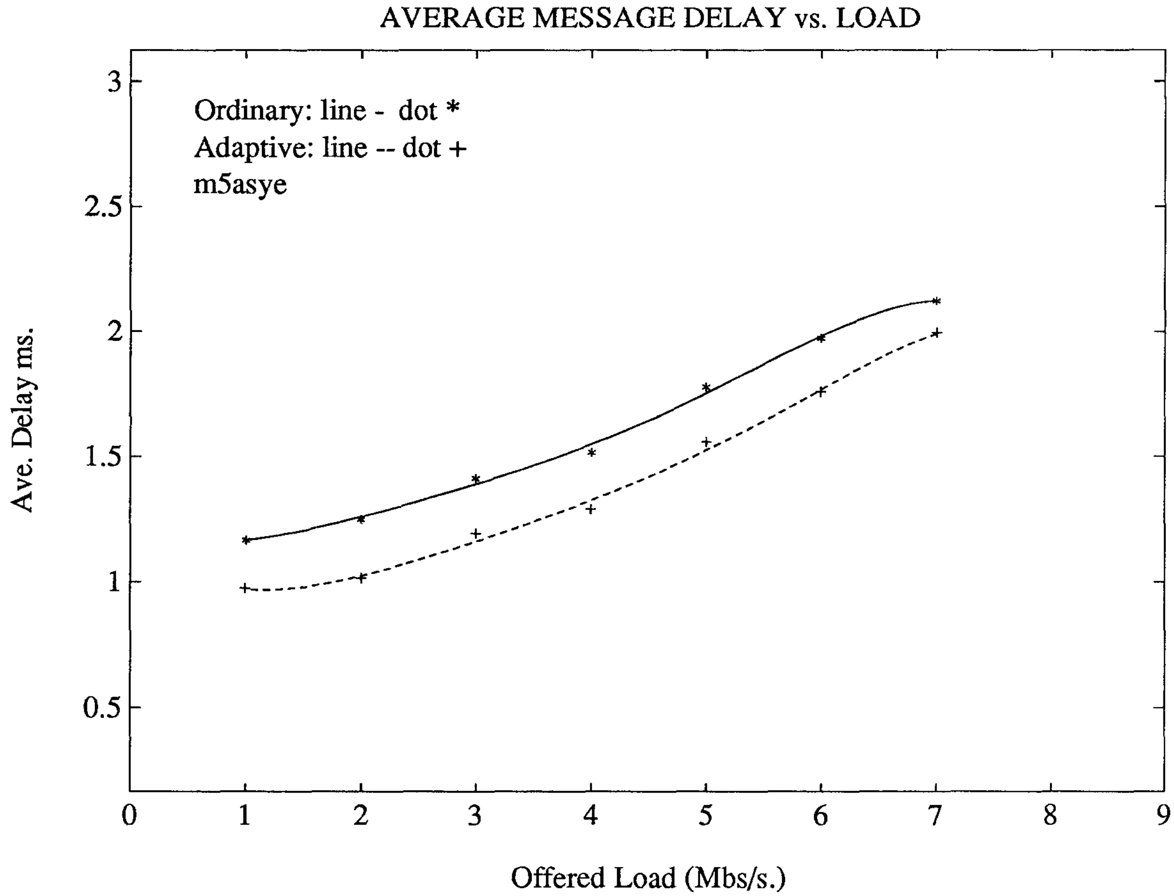

Figure 4.17: Inter High Priority Message Delay in Lightly Loaded Station with the 84% Load Distributed by Station 0

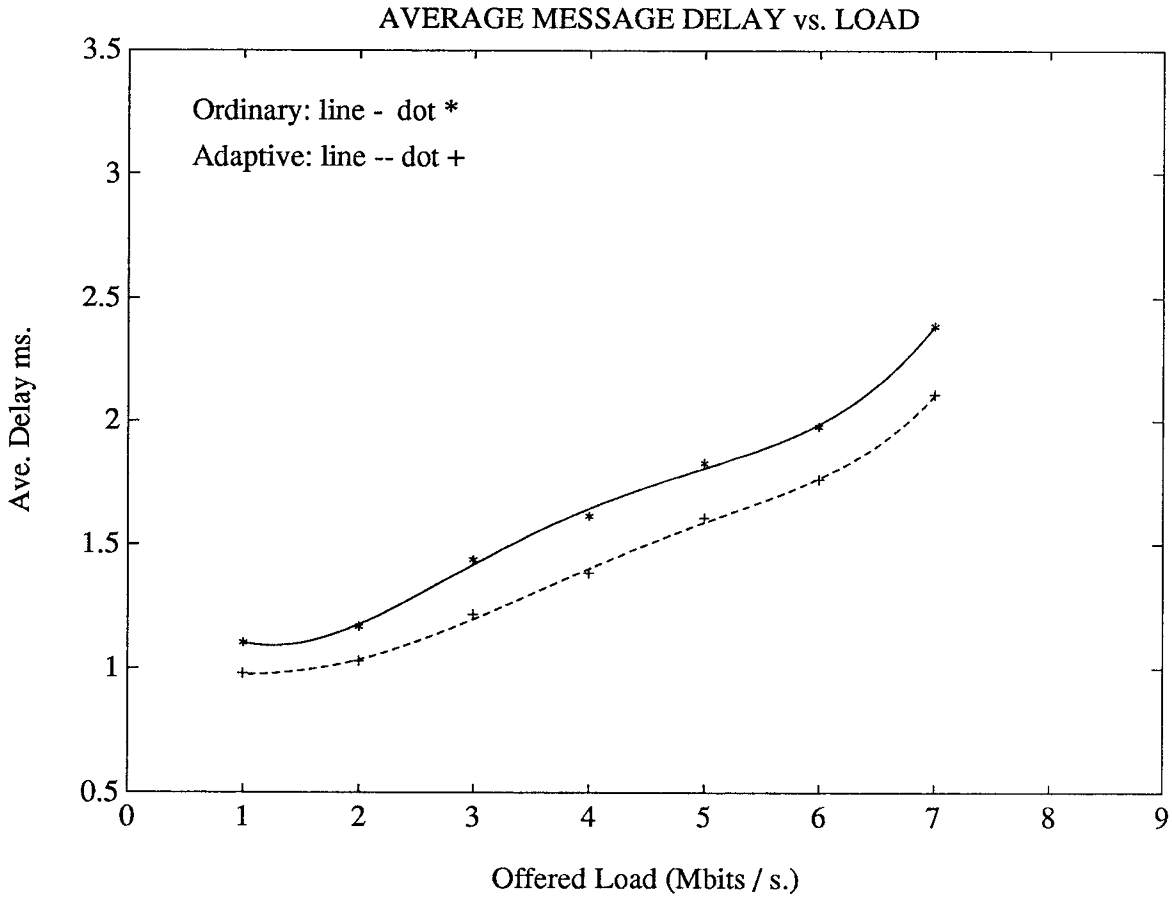

Figure 4.18: Inter High Priority Message Delay in Lightly Loaded Station with the 78% Load Distributed by Station 0

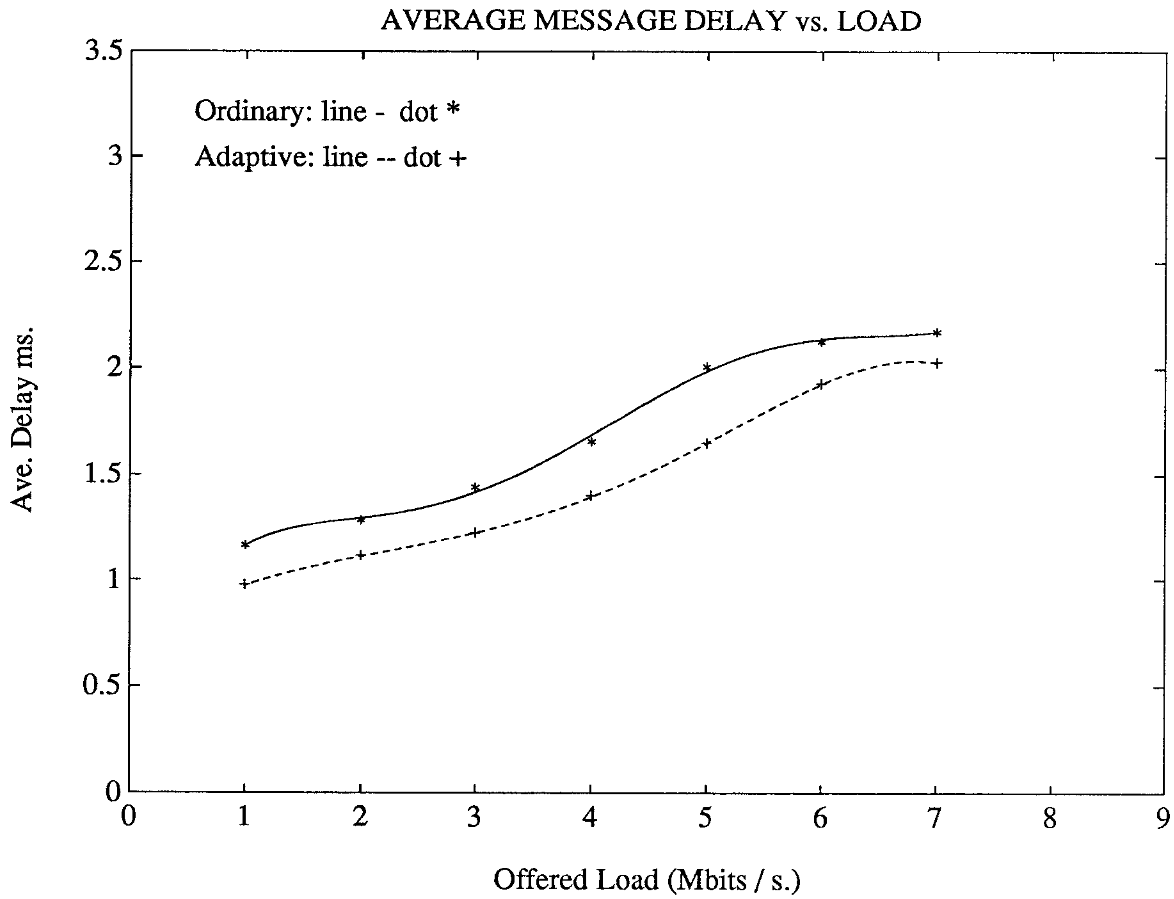

Figure 4.19: Inter High Priority Message Delay in Lightly Loaded Station with the 65% Load Distributed by Station 0

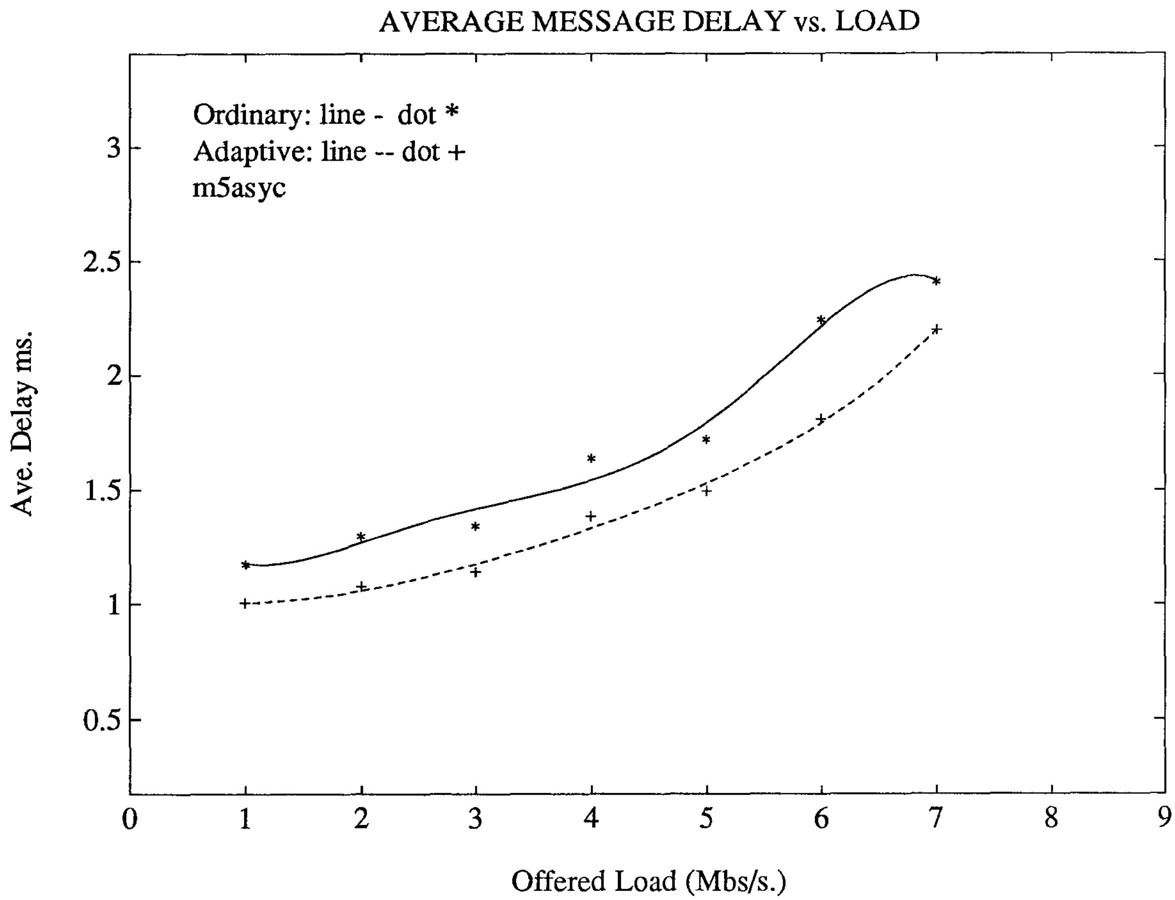

Figure 4.20: Inter High Priority Message Delay in Lightly Loaded Station with the 54% Load Distributed by Station 0

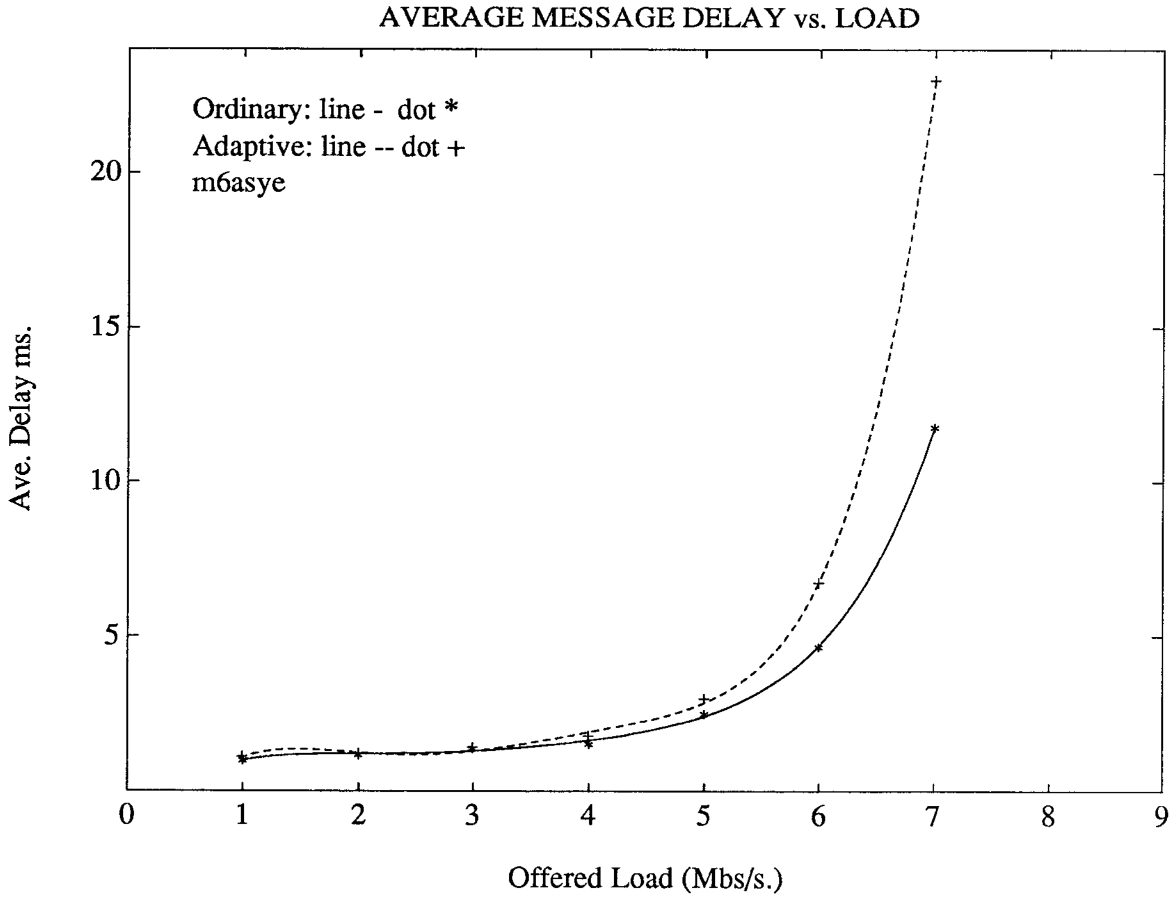

Figure 4.21: Inter Low Priority Message Delay in Lightly Loaded Station with the 84% Load Distributed by Station 0

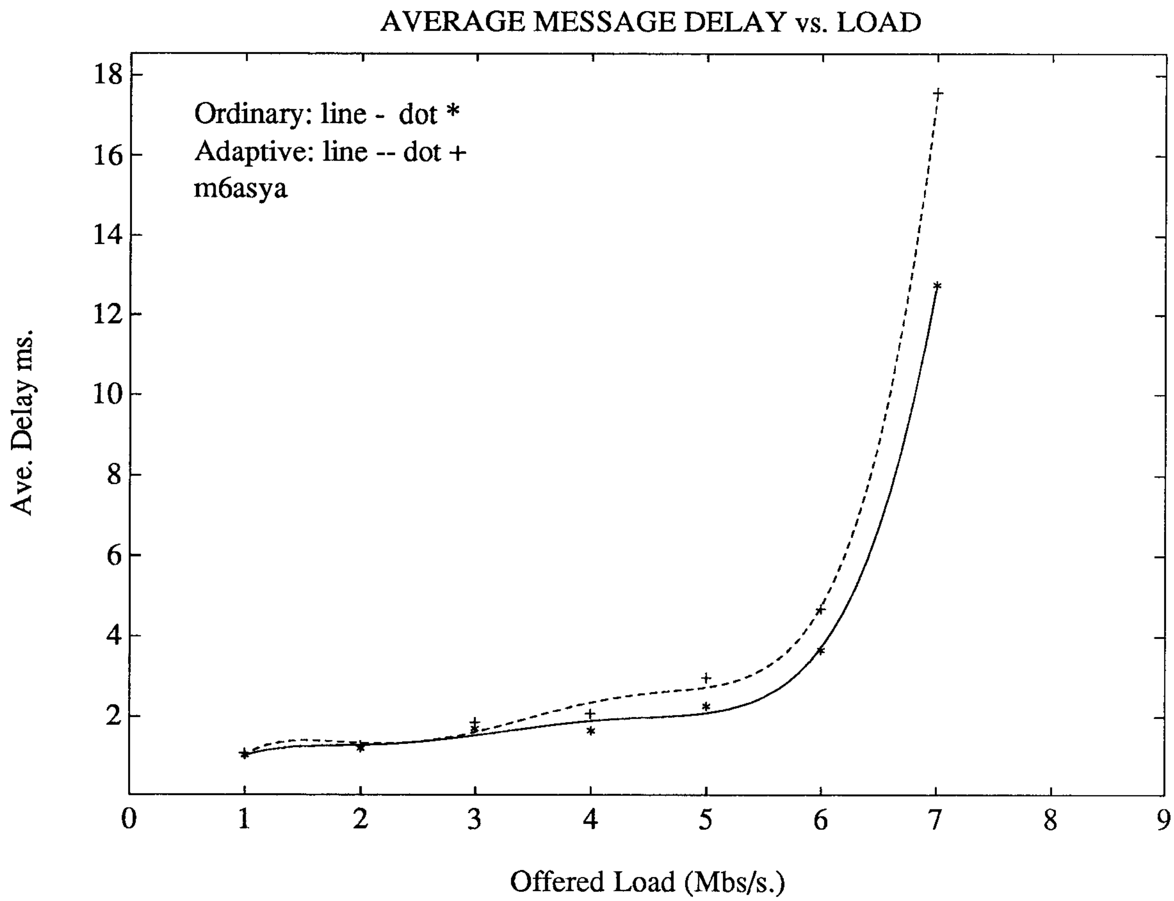

Figure 4.22: Inter Low Priority Message Delay in Lightly Loaded Station with the 78% Load Distributed by Station 0

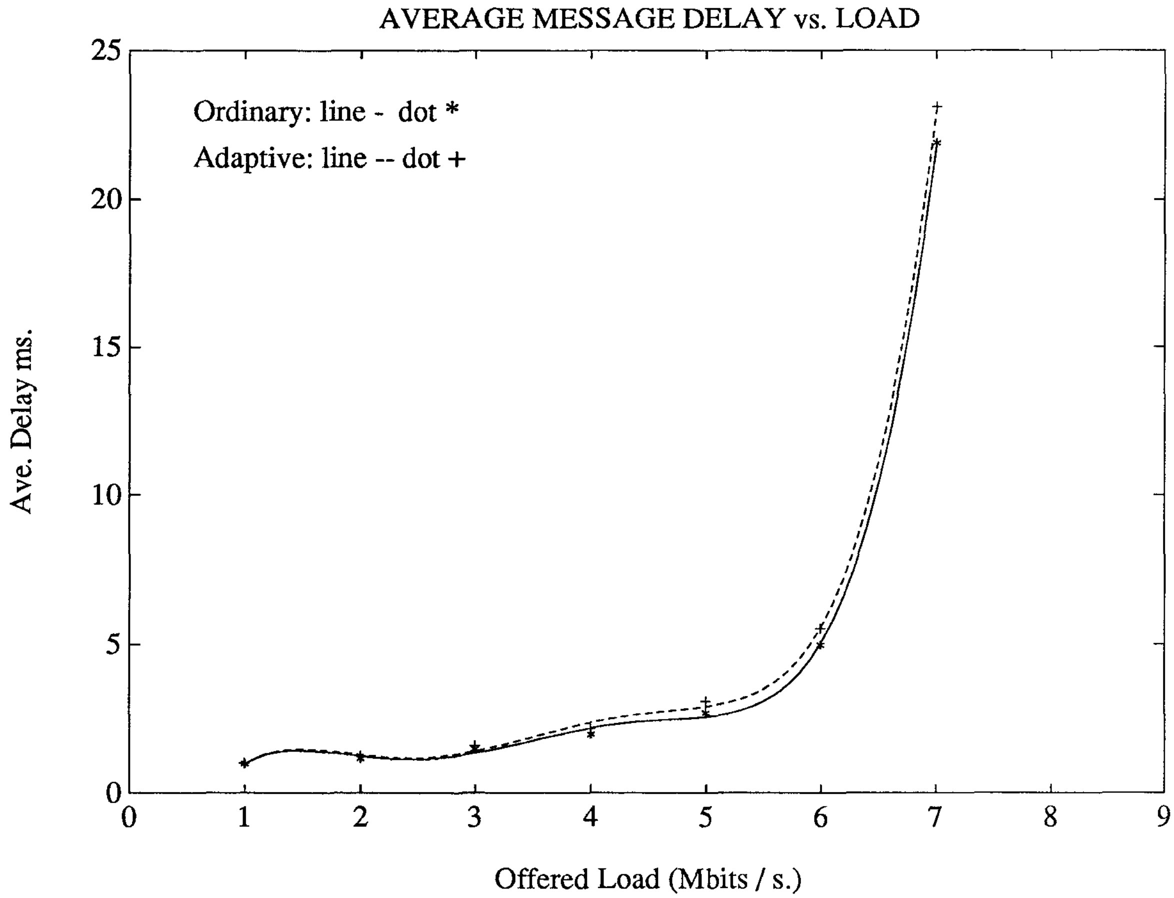

Figure 4.23: Inter Low Priority Message Delay in Lightly Loaded Station with the 65% Load Distributed by Station 0

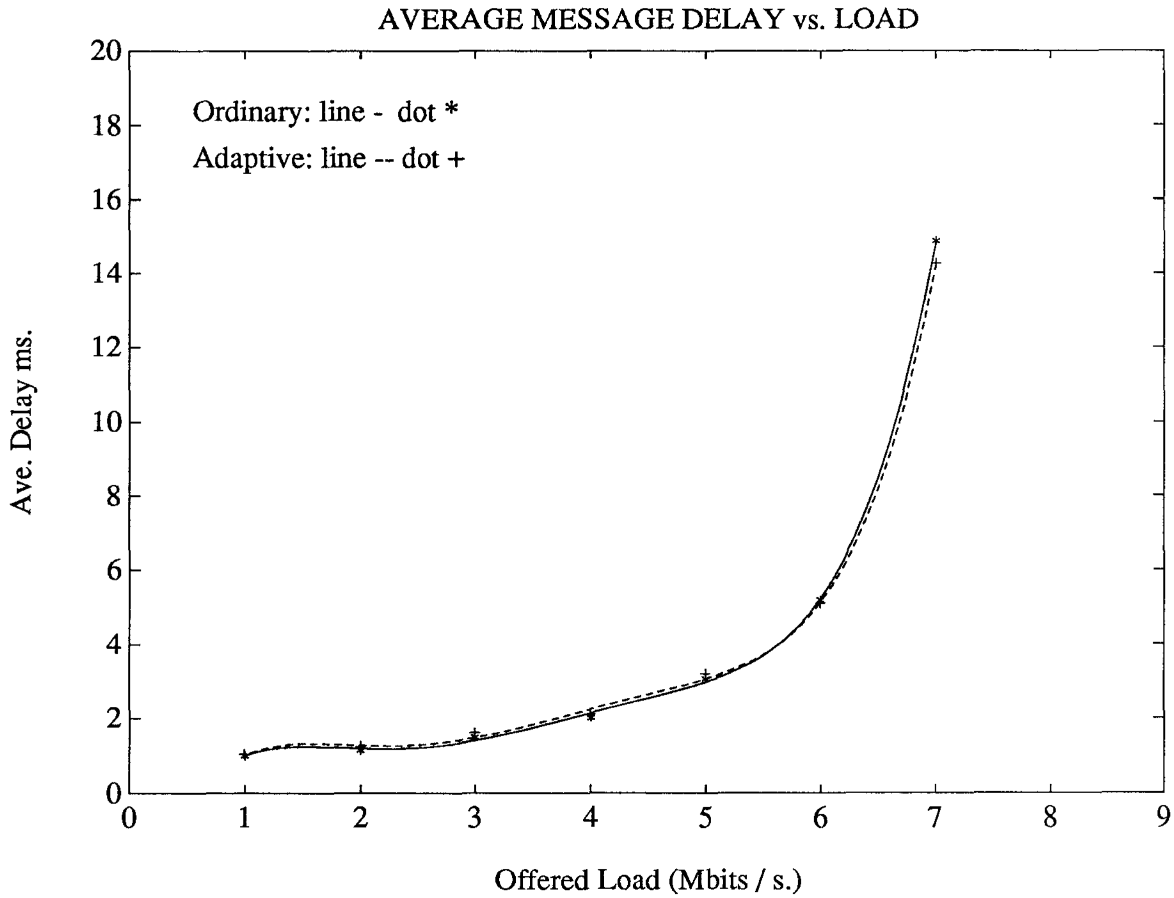

Figure 4.24: Inter Low Priority Message Delay in Lightly Loaded Station with the 54% Load Distributed by Station 0

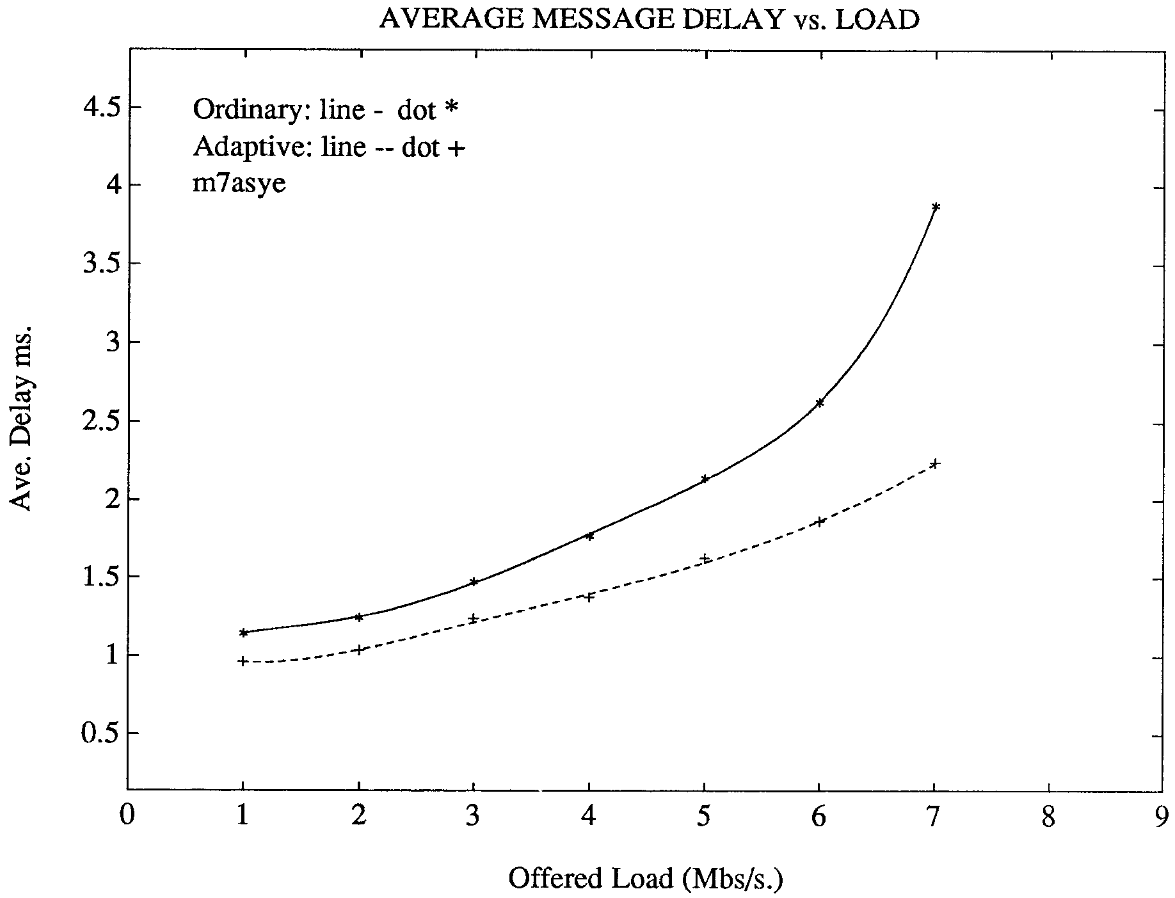

Figure 4.25: Inter Global High Priority Message Delay with the 84% Load Distributed by Station 0

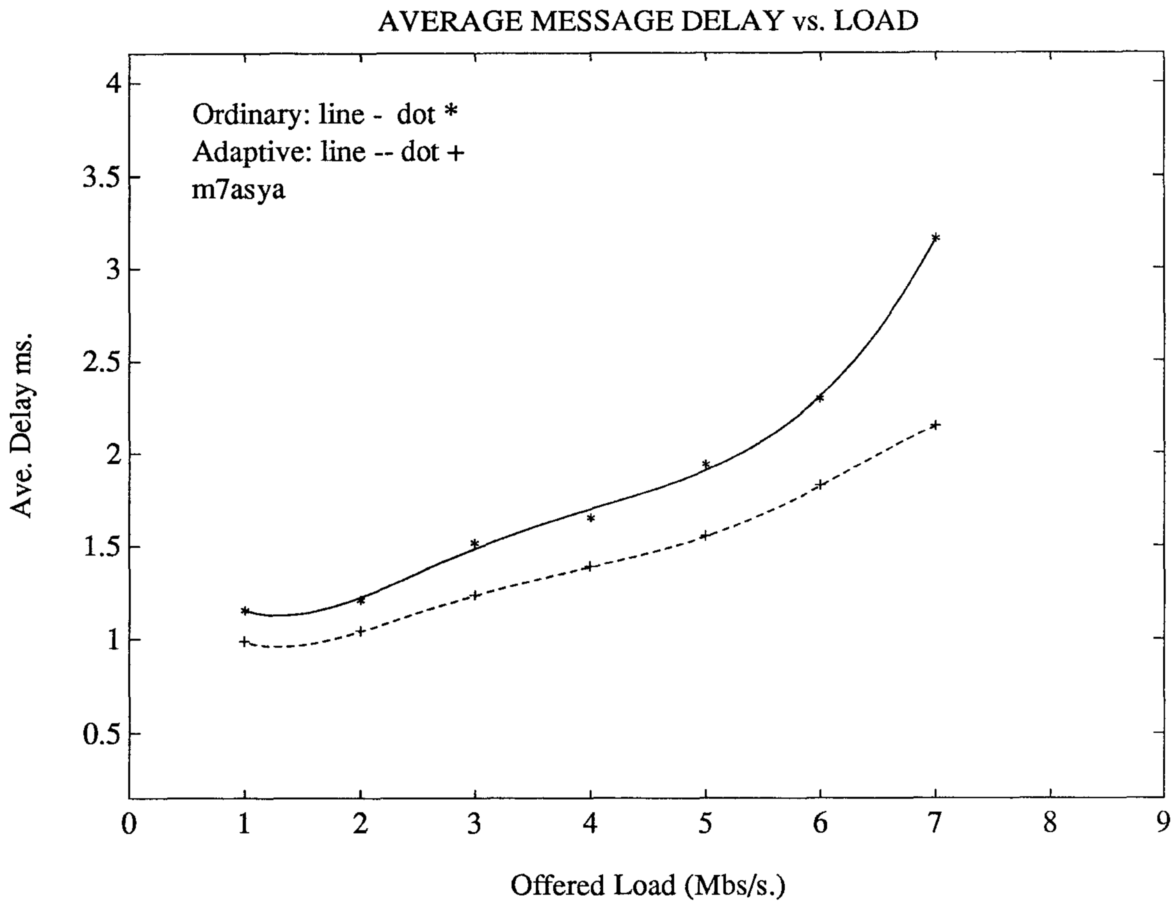

Figure 4.26: Inter Global High Priority Message Delay with the 78% Load Distributed by Station 0

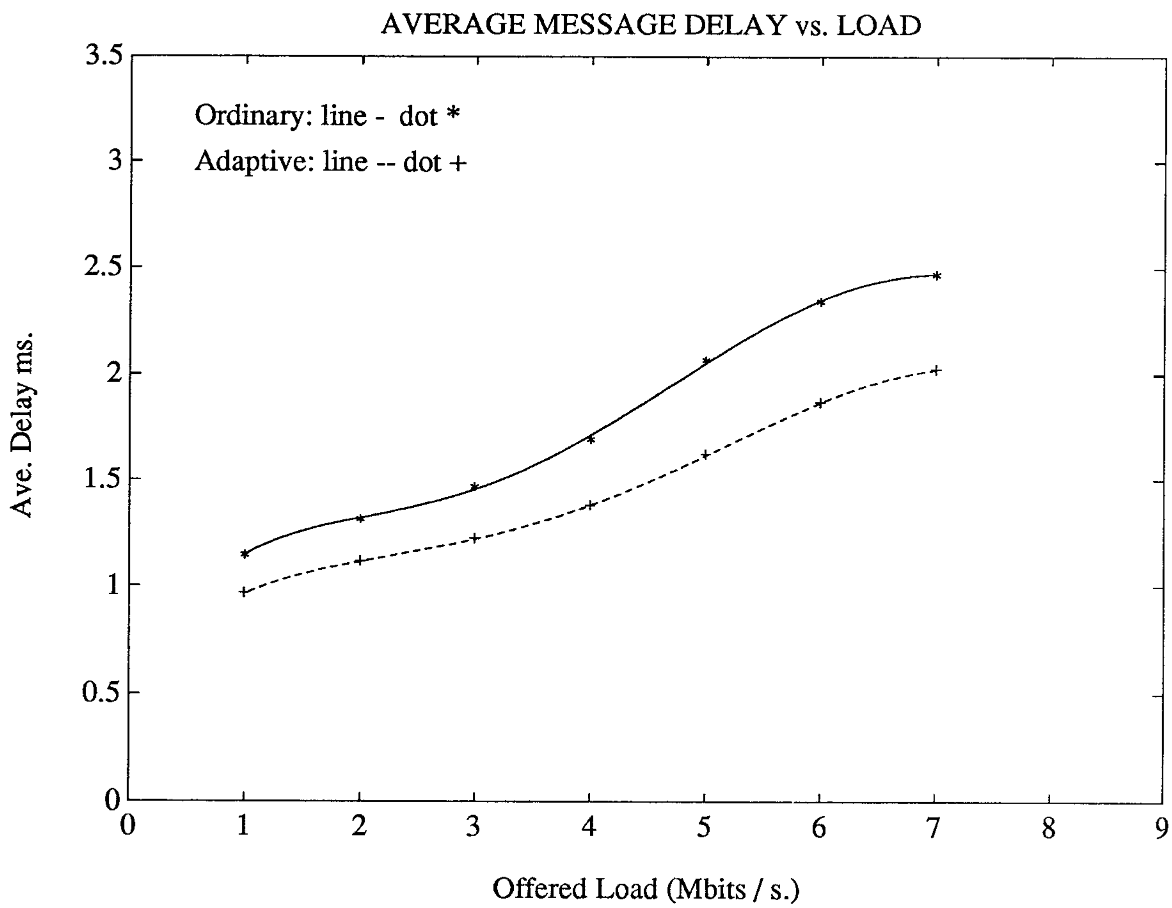

Figure 4.27: Inter Global High Priority Message Delay with the 65% Load Distributed by Station 0

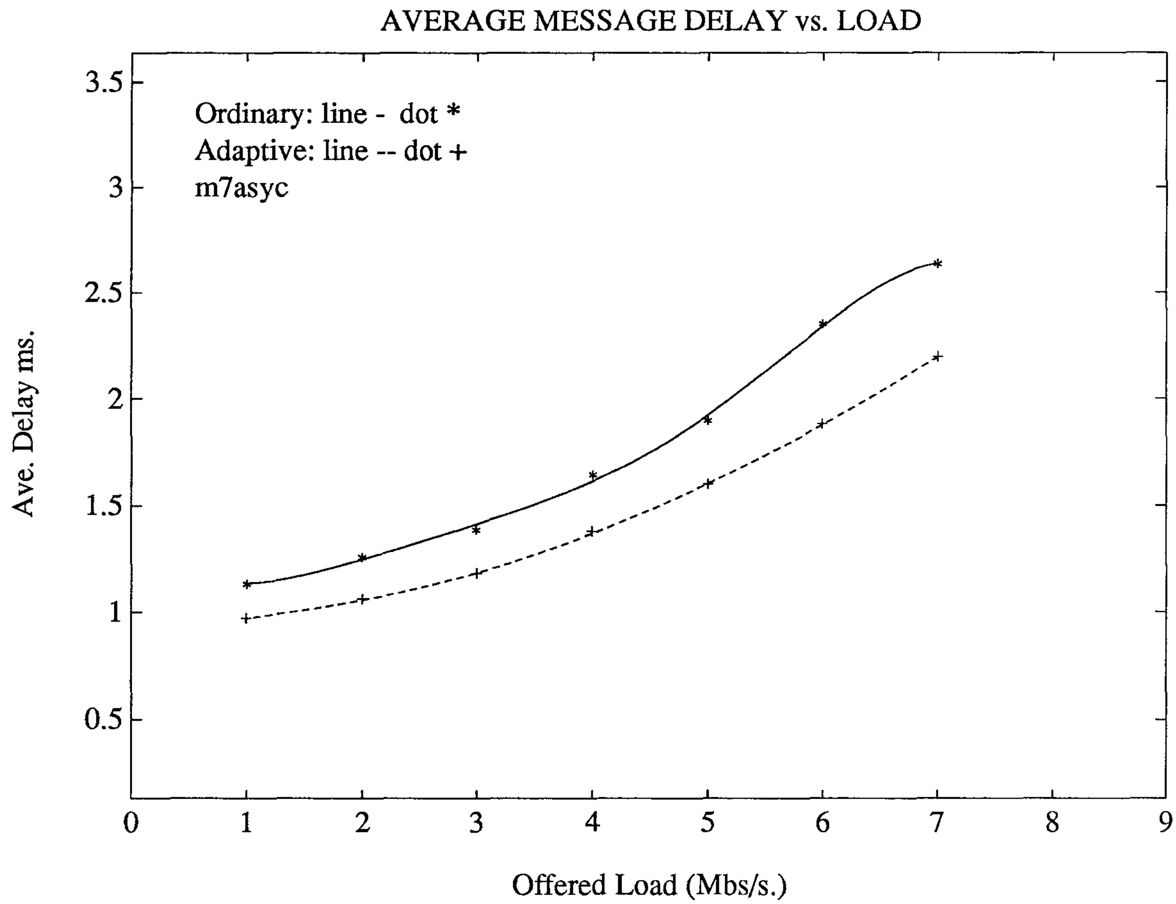

Figure 4.28: Inter Global High Priority Message Delay with the 54% Load Distributed by Station 0

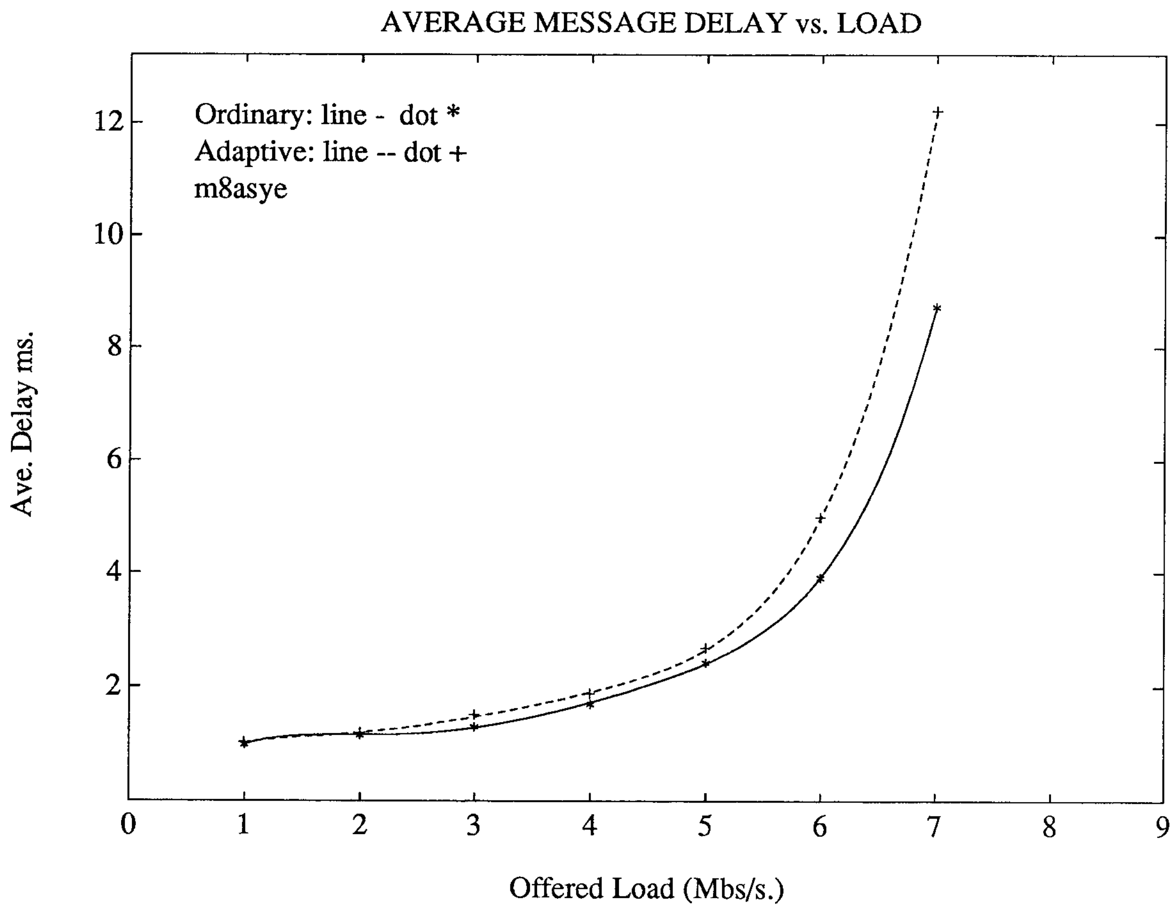

Figure 4.29: Inter Global Low Priority Message Delay with the 84% Load Distributed by Station 0

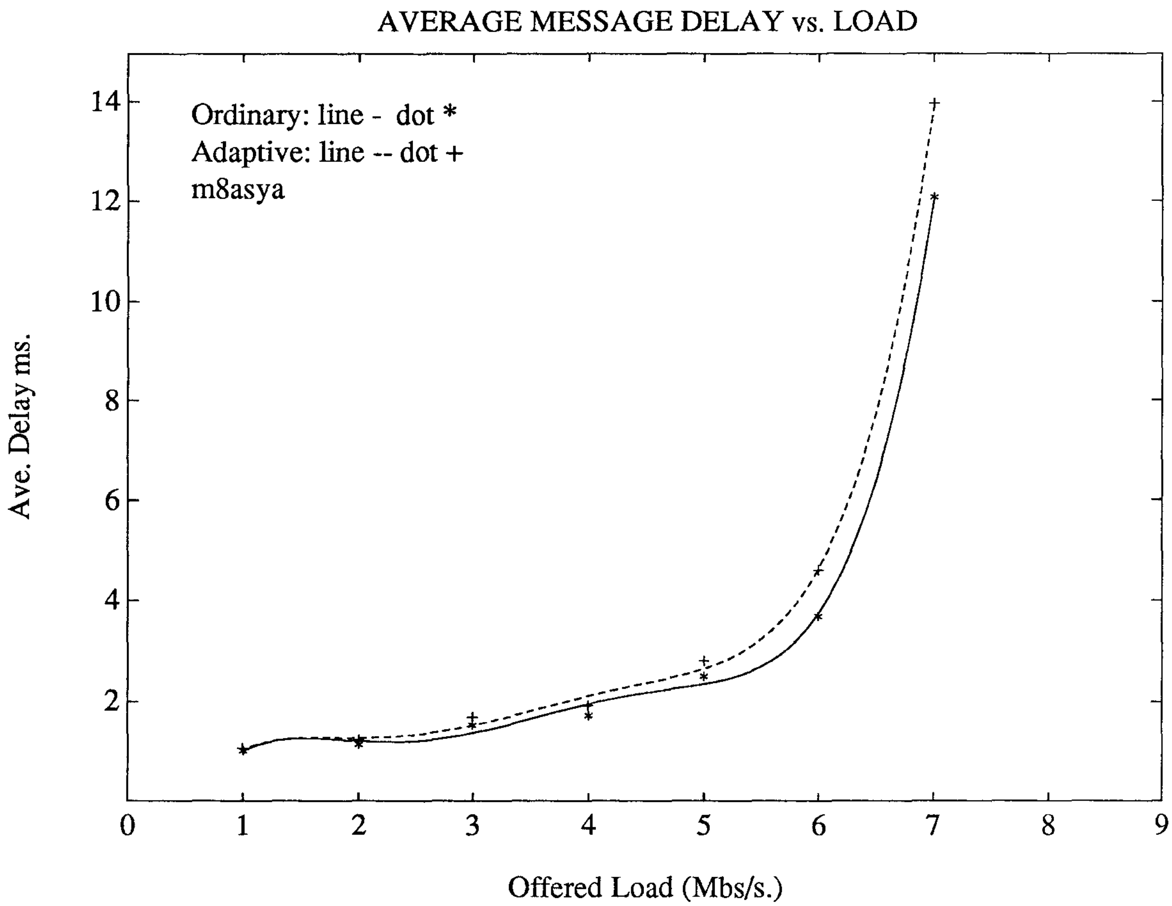

Figure 4.30: Inter Global Low Priority Message Delay with the 78% Load Distributed by Station 0

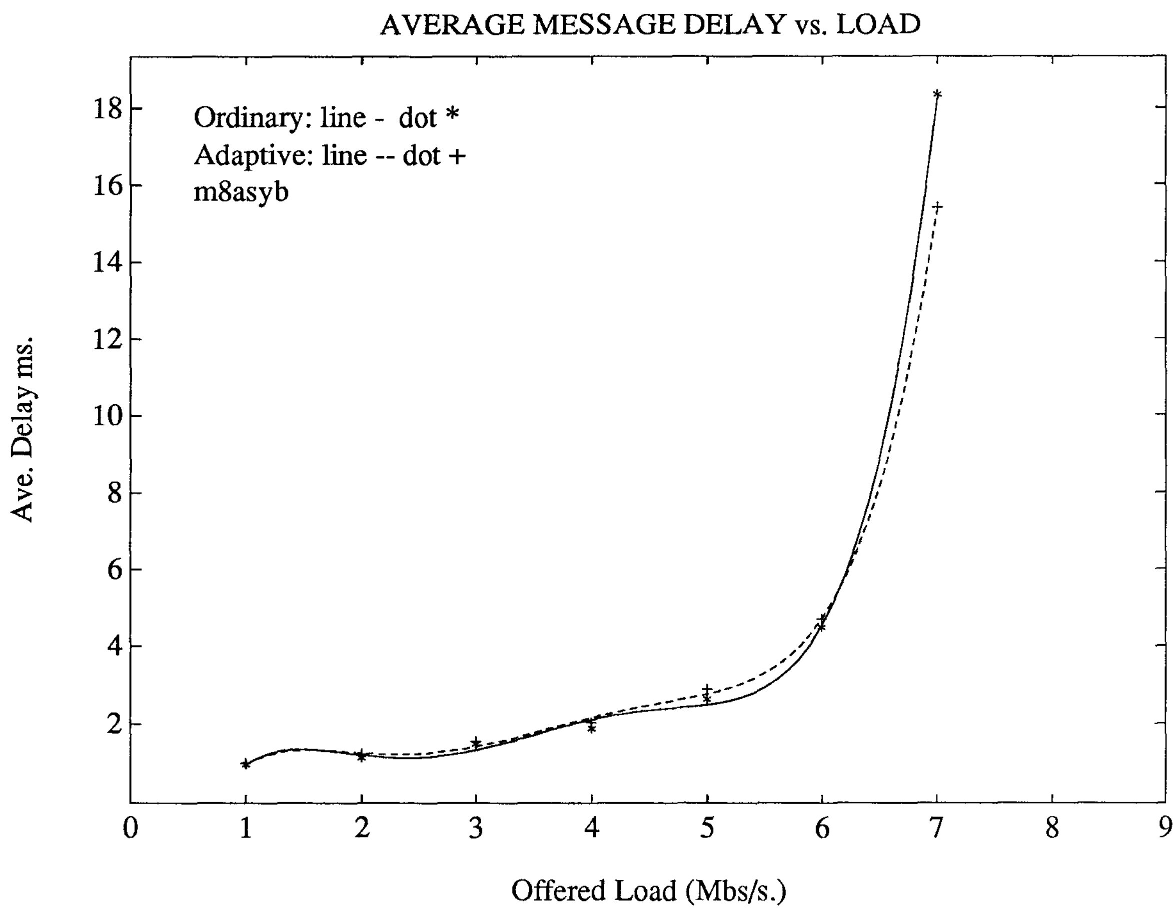

Figure 4.31: Inter Global Low Priority Message Delay with the 65% Load Distributed by Station 0

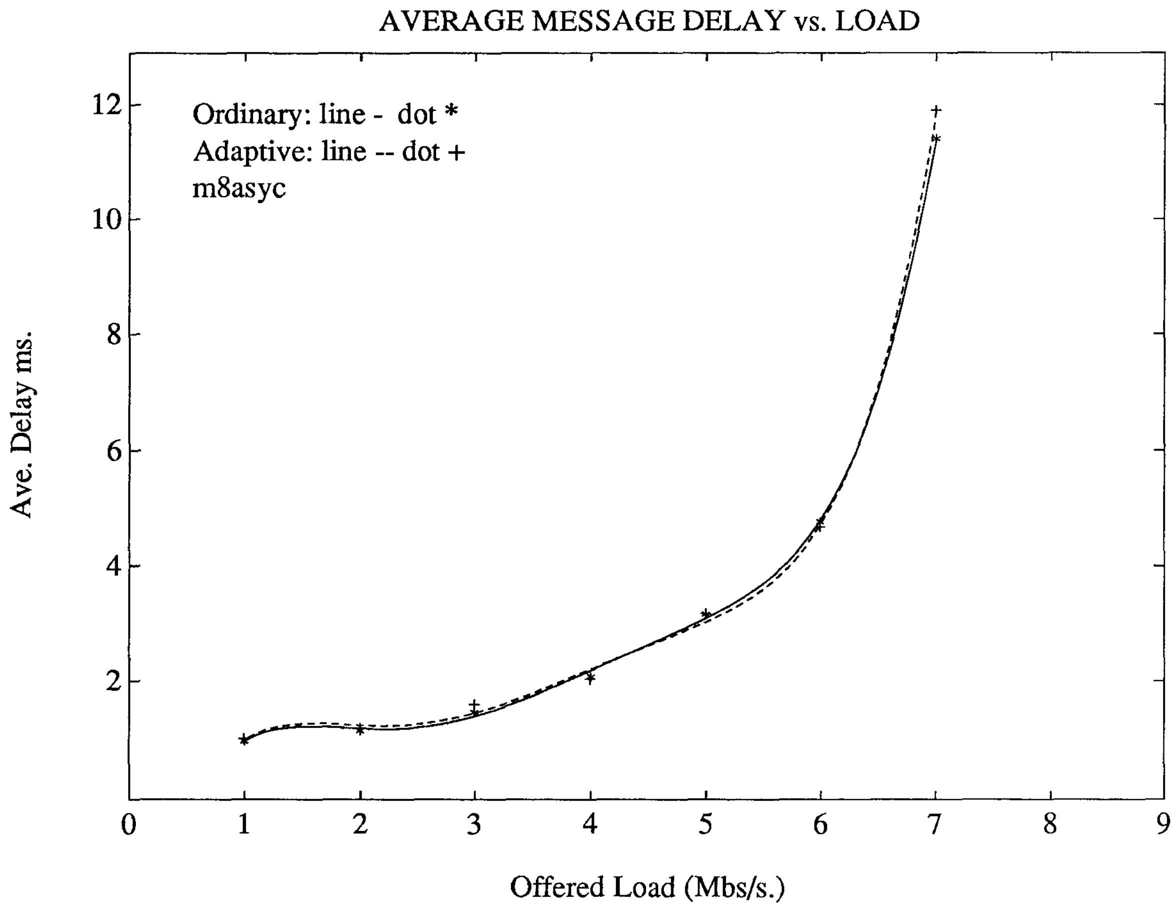

Figure 4.32: Inter Global Low Priority Message Delay with the 54% Load Distributed by Station 0

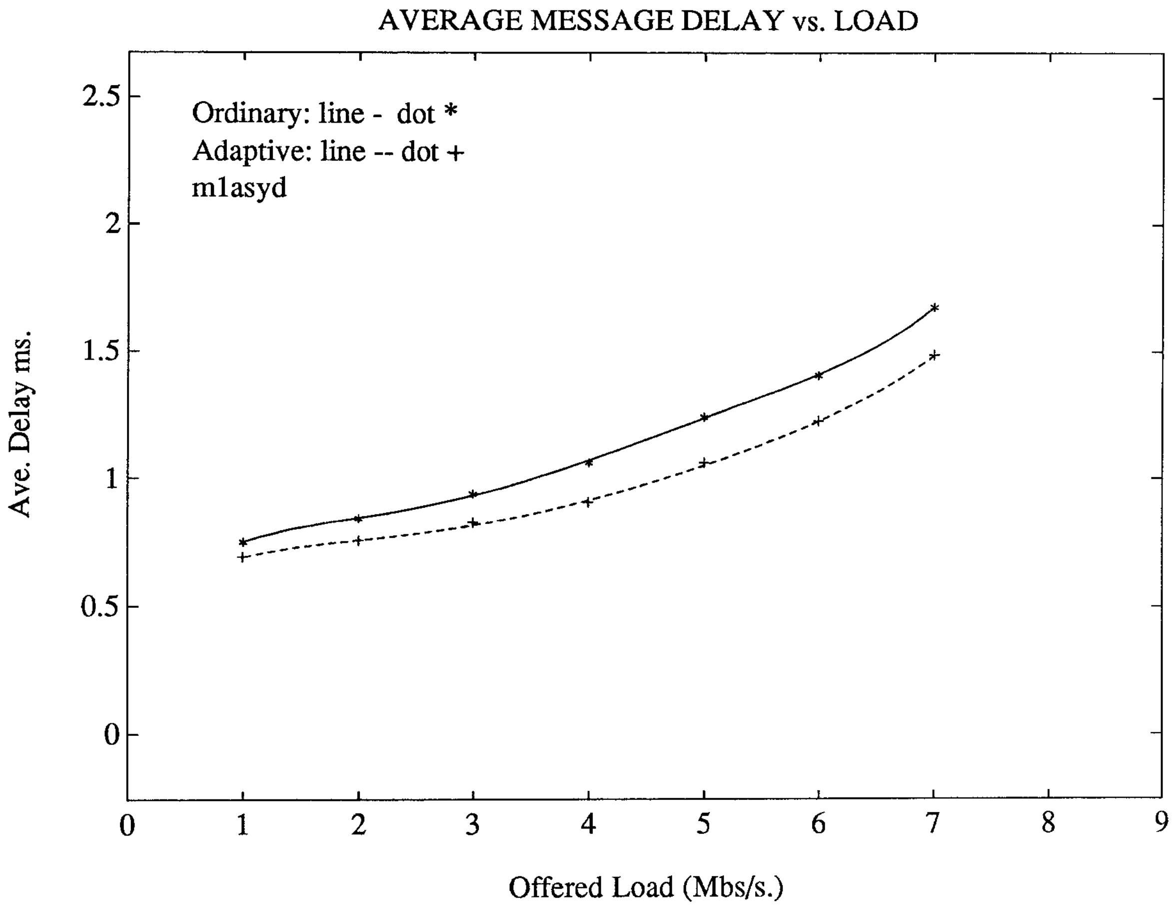

Figure 4.33: Intra Network High Priority Message Delay with Symmetric Load Distribution

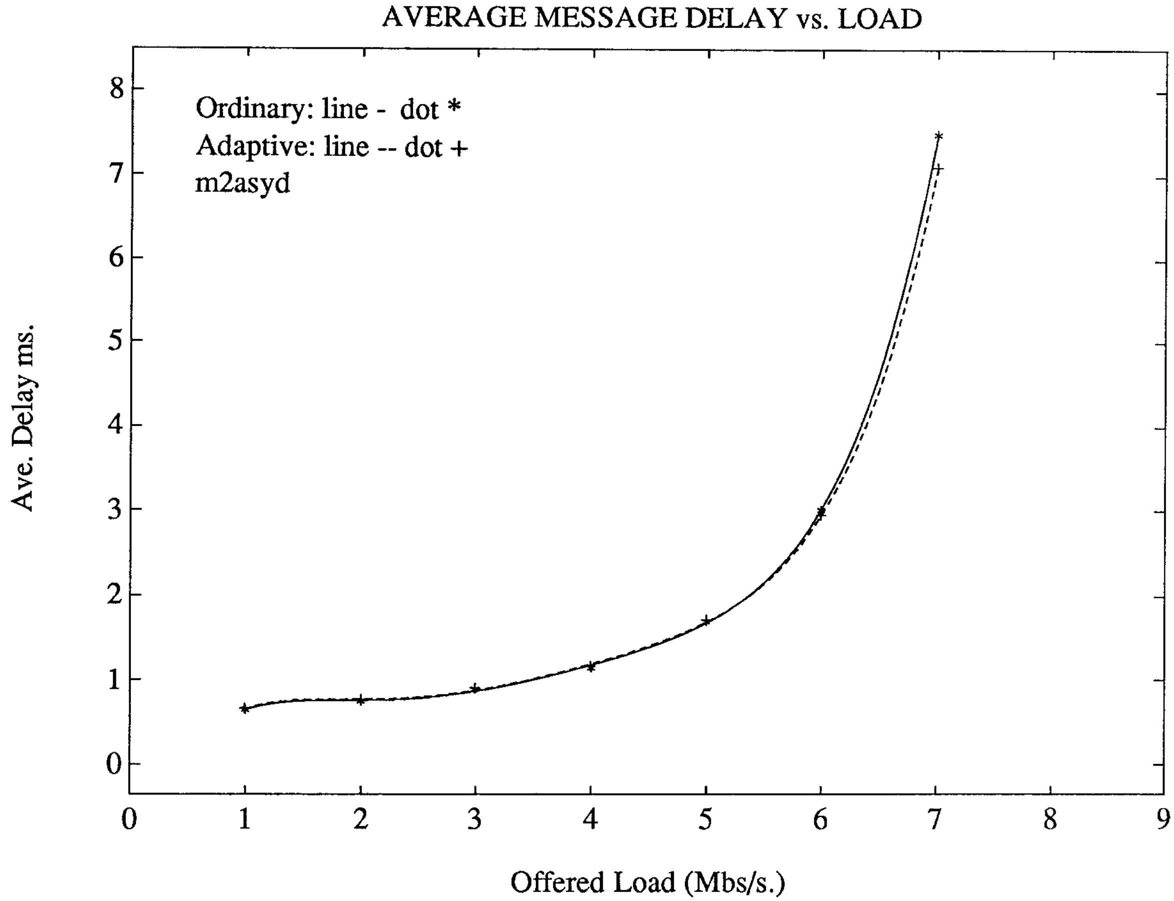

Figure 4.34: Intra Network Low Priority Message Delay with Symmetric Load Distribution

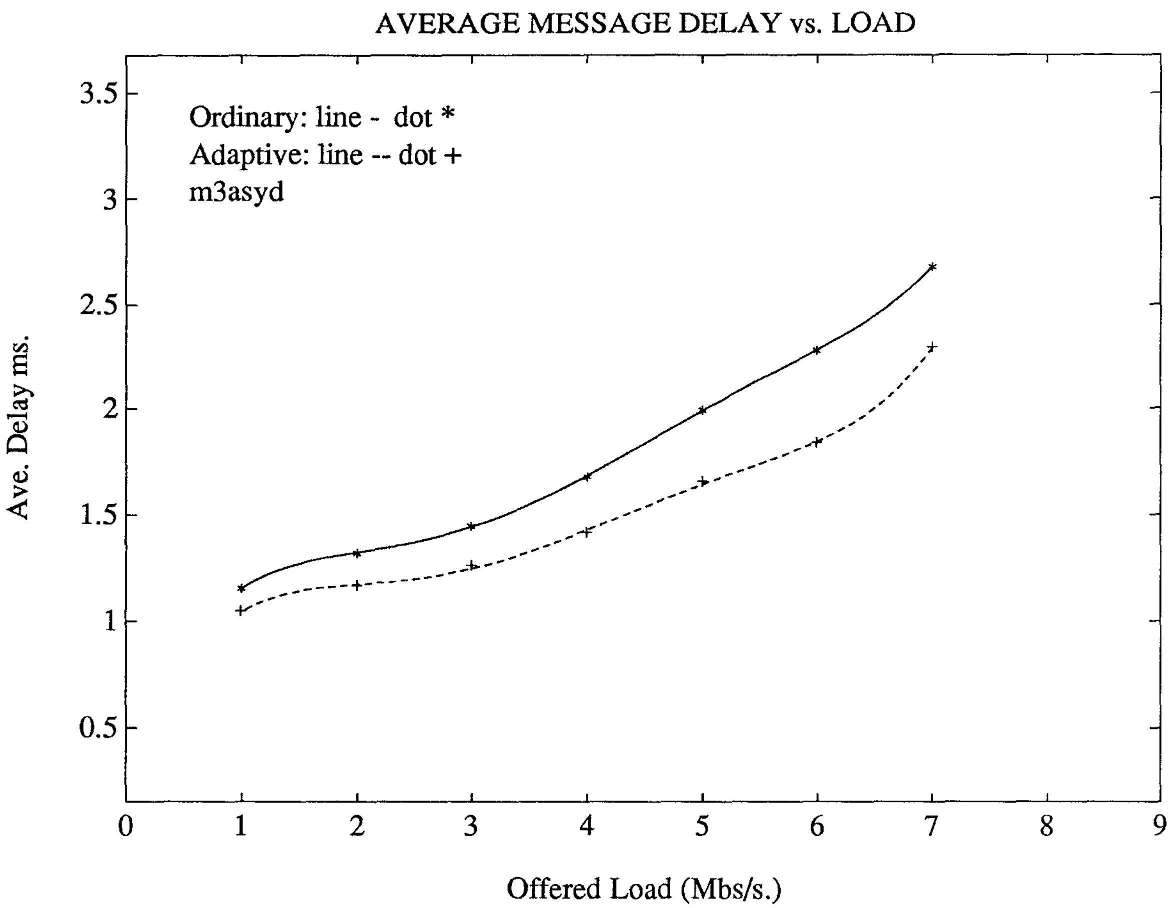

Figure 4.35: Inter Network High Priority Message Delay with Symmetric Load Distribution

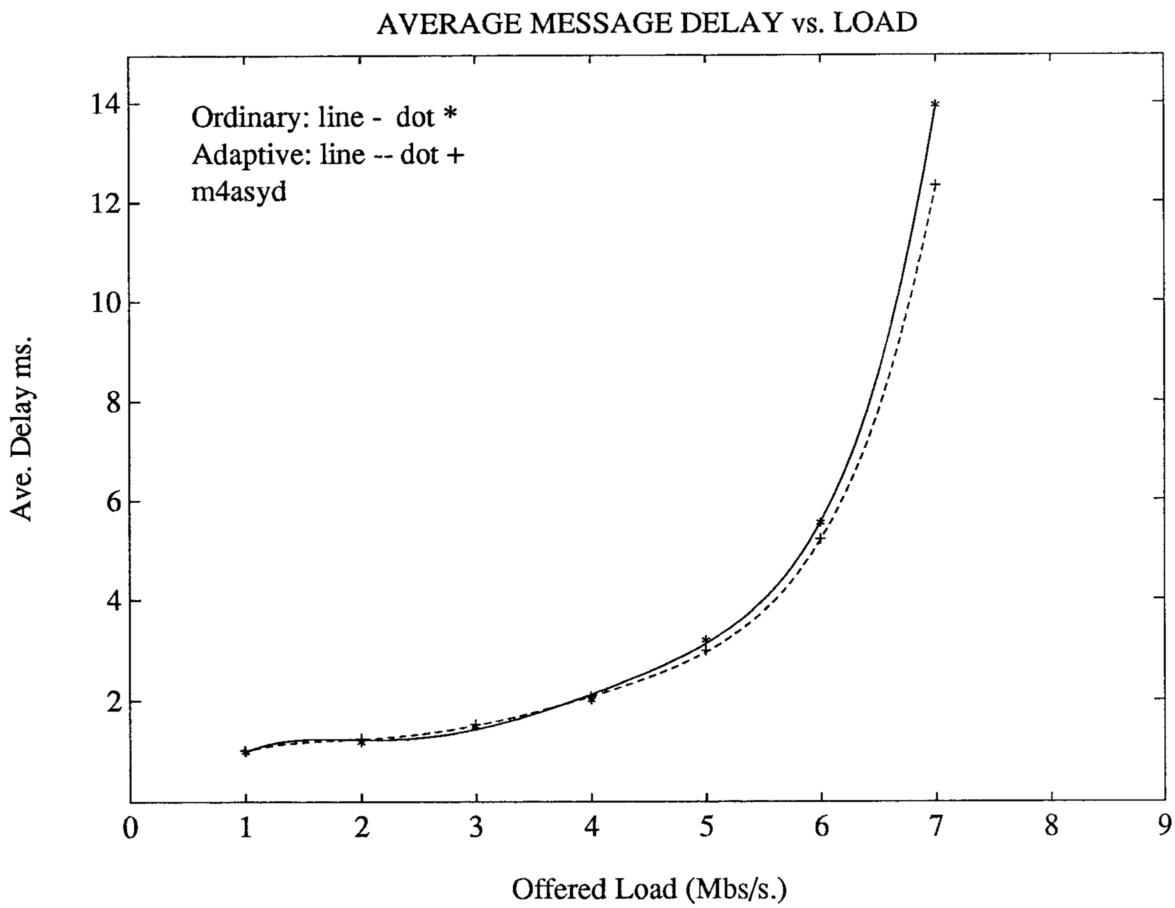

Figure 4.36: Inter Network Low Priority Message Delay with Symmetric Load Distribution

Chapter 5

Conclusion

A simulation study for the extended token bus network with priority based messages is presented. Two different schemes, ordinary and adaptive timer control for the transmission of high priority messages have been examined. The network is assumed to be homogeneous in terms of the characteristics of the sources connected to the network, but with asymmetric load distribution among the stations. The message delay for both intra and inter-net messages have been measured. Various results are plotted to illustrate the network performance under different load conditions.

By using a timer and a counter at each station, the adaptive timer control dynamically changes the high priority token holding time as the load on the network changes at different stations, at different time. For asymmetric load distribution, the adaptive timer control will reduce the high priority message delay in both the intra and inter-network domains. The difference between the ordinary and the adaptive timer control increases as the load distribution becomes more asymmetric.

Appendix A

Input Model

```
** Token bus **
*              **
* with bridge *
*              **

one etu                10000000 itu

Network configuration:

        Number of stations        16
        Ports:                    1/7 *Station 0-3 has 1 ports
                                  3   *Station 4 has 3 ports
                                  3   *Station 5 has 3 ports
                                  1/7 *Station 6-9 has 1 ports
        Buffers                   4/7
                                  4200
                                  4200
                                  4/7

Number of links         4

        *** Link 0 is broadcast bus interconnecting
        *** station 0,1,2,3,4
        Link type         0
        Archival time     120
        Number of ports   8     Ports:  0  0  1
                                        1  0  1
                                        2  0  1
                                        3  0  1
                                        4  0  1
                                        5  0  1
                                        6  0  1
                                        7  0  1

        Distance matrix:
```

```
@@ 01 02 03 04 05 06 07
   @@ 01 02 03 04 05 06
      @@ 01 02 03 04 05
         @@ 01 02 03 04
            @@ 01 02 03
               @@ 01 02
                  @@ 01
                     @@
```

```
*** Link 1 is a broadcast type bus interconnecting
*** station 5,6,7,8,9
Link type         0
Archive time      120
Number of ports   8     Ports:  8  0  1
                                9  0  1
                               10  0  1
                               11  0  1
                               12  0  1
                               13  0  1
                               14  0  1
                               15  0  1
Distance matrix:

@@ 01 02 03 04 05 06 07
   @@ 01 02 03 04 05 06
      @@ 01 02 03 04 05
         @@ 01 02 03 04
            @@ 01 02 03
               @@ 01 02
                  @@ 01
                     @@

*** Links 2 and 3 are unidirectional links interconnecting
*** station 4 and 5
Link type          3  (unidirectional channel)
Archival time      120
Number of ports    2    Ports:  7  2  1
                                8  1  1
Length (dist. matrix)  20

Link type          3  (unidirectional channel)
Archival time      120
Number of ports    2    Ports:  8  2  1
                                7  1  1
Length (dist. matrix)  20
```

Traffic:

```
Number of message type        16
```

*** Message type 0 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.007314286
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 1 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.007314286
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 2 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.007314286
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 3 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.007314286
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 4 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.002694737
Minimum length	1024
Maximum length	8192

Number of selection group 1
Number of flood groups 0

Number of senders 1, stations (0,1)
Number of receivers 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 5 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.002694737
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (0,1)
Number of receivers 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 6 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.002694737
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (15,1)
Number of receivers 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)

*** Message type 7 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.002694737
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (15,1)
Number of receivers 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)

*** Message type 8 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.051200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 9 (traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.051200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 10(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.051200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 11(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.051200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 6, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 12(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.019200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (0,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 13(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.019200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (0,1)
Number of receivers 7, stations (9,1) (10,1) (11,1) (12,1) (13,1) (14,1) (15,1)

*** Message type 14(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.019200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (15,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

*** Message type 15(traffic among station 0-3, intranetwork traffic)***

Options	1
Interarrival time	0.019200000
Minimum length	1024
Maximum length	8192
Number of selection group	1
Number of flood groups	0

Number of senders 1, stations (15,1)
Number of receivers 7, stations (0,1) (1,1) (2,1) (3,1) (4,1) (5,1) (6,1)

Protocol specific parameters:

Minimum packet length	512
Maximum packet length	2048
Frame (header + trailer) length	160
Token length	160
Packet space	16
Target token rotate time	10372
High priority token holding time	4096
Left bridge node	7
Right bridge node	8
Token idle time	1422

Bounds:

Maximum number of messages	10000
Virtual time limit	0
CPU time limit	0

Bibliography

[1] William Stallings, *Handbook of Local Network Standards* A Stallings/Macmillan Book, New York, NY 1987.

[2] Joseph W. Pang and Fouad A. Tobagi, "Throughput analysis of a timer controlled token passing protocol under heavy load", *Proc. of INFOCOM*, pp. 796-804, April 1988.

[3] A.Valenzano, P.Montuschi and L.Ciminiera, "On the behaviour of control token protocols with asynchronous and synchronous traffic" *Proc. IEEE INFOCOM*, pp. 348-355, 1989.

[4] Andrew S. Tanenbaum, *Computer Network* Prentice hall, Englewood cliffs, NJ, 07632, pp. 117-164, 1985.

[5] W. Bux and D.Grillo, "End to end performance in Local Area Networks with Bridges", *IEEE Transaction of Comm.*,Vol. 36, pp. 47-53, May 1988.

[6] Dr.Pawel Gburzynski and Dr.Piotr Rudnicki, "*The LANSF Protocol Modeling Environment*," Department of Computing Science, The University of Alberta, Edmonton, Alberta, Canada.

[7] Conrad K. Kwok and Biswanath Mukherejee, "A Multiple - Partition Token Ring Network" *Proc. of IEEE INFORCOM*, Vol.1, pp. 982-988, 1990.

[8] Pascale Minet, "Performance Issues with the SAE LTPB and IEEE 802.4 PROTOCOLS " *Proc. of IEEE INFORCOM*, Vol.1, pp. 982-988, 1990.

[9] Bijan Jabbari and Ya-Qin Zhang, "Performance degradation in Hetrogeneous LAN Interconnections" *Proc. of IEEE GLOBECOM*,Vol.2, pp. 919-923, 1989.

[10] Anura Jayasumana, "Performance Analysis of Token Bus Priority Scheme". *Proc. of IEEE INFOCOM*, pp. 46-54, March 1987.