Abstract

Ordinary service and exhaustive service are two major alternatives of scheduling policies considered in providing access to token ring networks. Results to date have shown that exhaustive service results in more delay to lightly loaded stations in asymmetric traffic while ordinary service wastes time in circulating the token after each transmission. This work presents a new token passing protocol, called adaptive service, in which the token holding time is dynamically changing; in this way, it provides a fair compromise between exhaustive and ordinary service. The simulation results show that in asymmetric traffic, adaptive service gives improvement on the local delay compared with exhaustive service and gives improvement on global delay compared with ordinary service. Also for symmetric traffic, it gives improvement compared to ordinary service. Moreover, it always provides superior throughput performance over ordinary service.

Performance Evaluation and Comparison of Ordinary, Adaptive and Exhaustive services in the Token Ring Network

Hong Wang

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

Title of Thesis: PERFORMANCE EVALUATION AND COMPARISON OF ORDINARY, ADAPTIVE AND EXHAUSTIVE SERVICE IN THE TOKEN RING NETWORK

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

Thesis Approval:

_______________________________ _______________
Dr. Constantine Manikopoulos , Advisor Date
Associate Professor

_______________________________ _______________
Dr. George Antoniou Date
Visiting Professor

_______________________________ _______________
Dr. Mengchu Zhou Date
Assitant Professor

VITA

Name: Hong Wang

Degree and date to be conferred: Master of Science in Electrical Engineering

Secondary education: Shanghai XiangMing high school

Collegiate institutions attended	Date	Degree	Date of degree
New Jersey Institute of Technology, Newark, NJ	1989-91	M.S.	1991
Shanghai University of Science and Technology, Shanghai, PRC	1985-89	B.S.	1989

Major: Electrical Engineering

Acknowledgments

I wish to express my sincere thanks to Dr. Constantine N. Manikopoulos for his valuable guidance and encouragement throughout the course of this thesis. I am specially indebted to him for his insightful and constructive criticisms at every stage of this thesis without which this thesis would have never been completed.

Last but not least I would like to thank all those who have helped me in one way or the other during the entire course of this thesis.

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Token Ring Local Area Network

Local area networks are an area of increasing importance in recent years. One of the more important protocols for a local area network is that of the token ring. In this protocol, a permit to transmit or "token" is circulated among the stations in the network, so that a station cannot transmit until it receives a token message. Normally, the stations are connected to a unidirectional bus. When stations have nothing to transmit, the token, consisting of a single valued token bit within the message frame header, circulates around the ring. When a station has a message, or messages, to transmit, it waits until it detects a message frame header with zero valued token bit passing its channel connection and sets the token bit to 1. It appends the message to be transmitted to the frame header. Changing the token to 1 implies that the channel is busy and that a message is circulating on the channel. When some other station senses a non-empty token it refrains from transmitting. All the stations on the network check the destination address of the frame. The message travels on the network and serves as an automatic acknowledgement to the sender. After the message has made a trip around the ring, the transmitting station purges the message off the frame, sets the token bit to zero (implying an empty token) and then passes the token to the next station for its use. The various versions of the protocol in the token ring network n differ in how many messages the station can put onto the ring before it is forced to retransmit the token. Two of the existing possibilities are ordinary service and exhaustive service which will be discussed later.

1.2 Performance Measures

Two measures of LAN performances are commonly used:

1) Message delay: Message delay is measured as the time elapsing since the message was queued at the sending station to the moment the entire message is successfully received at the destination.

2) Throughput: The throughput of the network is a measure in bits per second of the traffic being successfully transmitted between stations. Since packets can become corrupted in traveling from station to station, it is

customary to count only the error-free bits when measuring throughput. The value of throughput is normalized.

1.3 Purpose of the study

Several performance studies have already been presented in the literatures for modeling ordinary service [1,2] and exhaustive service [3,4,5]. Both ordinary and exhaustive services have several disadvantages for symmetric and asymmetric traffic. This is because as traffic changes they do not change their token holding strategies. Exhaustive service gives more local delay at lightly loaded stations, while ordinary service results in more global delay for the network as a whole.

The contribution brought by the results presented in this paper is that the new service, adaptive service, is proposed. Adaptive service dynamically changes token holding time as load on the network changes at different stations, at different time. And it is proved to be a very good compromise between the ordinary and exhaustive service for all types of traffic, different message lengths and different number of stations.

1.4 Outline

The outline of the rest of this thesis is as follows.

Chapter 2 presents a brief theory of the ordinary and exhaustive service and the principle of adaptive service. Chapter 3 is about the simulation model, which is concerned with different services.

Results of the simulation study are described in Chapter 4. Delay and throughput characteristics are plotted and analyzed.

Conclusions are presented in chapter 5.

Appendix A covers the input model for different types of traffic.

Chapter 2

Token Ring Network

2.1 Two Versions of the Token Ring Protocol

As already mentioned, two major protocol alternatives in accessing the token ring network are ordinary service and exhaustive service

Ordinary service: when the token reaches a station, only the first pending message (if any) is transmitted, before the station passes the token to the next station.

Exhaustive service: when the token reaches a station, all the pending messages are transmitted, before the station passes the token to the next station.

2.1.1 Delay Performance

In the ordinary service, a station can only transmit one message per token. All the stations have equal access to the transmission medium. But when the traffic is asymmetric, which means the traffic is not equally distributed among all the stations in the network, this scheduling policy reduces the total amount of useful bandwidth available for data transmission by circulating the token most of the time. In this way, every station gets more and more delays. This is especially true for the heavily loaded stations which suffer long waiting times in the queue. The lightly loaded stations are not hampered by comparison to heavily loaded stations, however heavily loaded stations, which may account for most of the traffic, are getting delays due to the required continuous token rotation. This increases both the local delay at heavily loaded stations and the global delay of the network.

With exhaustive service difficulties appear primarily for asymmetric traffic load. At the heavily loaded stations, it takes a very long time to empty the queues. This results in high local delay to the lightly loaded stations. Because most of the traffic is assigned to the heavily loaded stations and local delays at the heavily loaded stations are low, this service gives low global delay, however very high local delay result at the lightly loaded stations. This service is, thus, unfair to the lightly loaded stations.

2.1.2 Throughput Performance

In order to compare the throughput characteristics of the ordinary and exhaustive services, we assume that the stations always have a message, or messages, to transmit when they catch the token. It is further assumed that the same number of messages are transmitted on the network at each token rotation. Let m stand for the maximum number of messages received at their destinations during the simulation run, n stand for the number of stations in the network, p_{no} and p_{ne} stand respectively for the number of messages transmitted by the n^{th} station when holding the token in ordinary and exhaustive services. m and n remain the same values for the two services. In the ordinary service, p_{no} always equals one. The total token rotation times are $r_o=m/(\Sigma^{n-1}_{n=0}\ p_{no})=m/n$. In exhaustive service, p_{ne}= the number of messages pending at the station during the token holding time which is equal or greater than 1. So the total token rotation times are $r_e=m/(\Sigma^{n-1}_{n=0}\ p_{ne})\leq m/n=r_o$. The simulation time is proportional to the token rotation times. The more the token rotation times are, the smaller the throughput is. Theoretically, the exhaustive service has superior throughput performance over the ordinary service.

2.2 Proposed Model of Adaptive Service

The reason to introduce a new service, adaptive service, is that neither ordinary service nor exhaustive

2.2 Proposed Model of Adaptive Service

The reason to introduce a new service, adaptive service, is that neither ordinary service nor exhaustive service works well for all types of traffic. Ordinary service results in high global delay and low throughput while exhaustive service causes very high local delay at lightly loaded station in asymmetric traffic. Adaptive service provides a compromise between ordinary and exhaustive service for all types of traffic.

Now let us look at how the adaptive service works.

2.2.1 Delay Performance

In this service every station has a counter and a timer. Counter counts queued messages at the station. When the station passes the token to the next station it resets its timer. When the token returns the station compares the time elapsed for this latest token rotation with the ideal token rotation time. Here, ideal token rotation time is the time spent in rotating the token once on the network without transmitting any messages. So in this service the timer keeps track of the global activity of the network and the counter keeps track of the local activity at the station.

Now from the exhaustive service we know that if the station having a long queue keeps the token for more time it reduces the global message delay of the network. In other words, to reduce global delay of the network, token holding

$t^{j}_{ht,i}$= time to hold the token in i^{th} rotation at j^{th} station.

Q^{j}_{i} = queued packets in i^{th} rotation at j^{th} station.

t_{pac} = time to transmit a packet on the channel.

$P^{j}_{tok,i}$= the number of packets transmitted on the channel by j^{th} station when holding the token in i^{th} rotation.

We know that

$$P^{j}_{tok,i}= \frac{t^{j}_{ht,i}}{t_{pac}} \qquad (2.1)$$

and from above discussion

$$t^{j}_{ht,i} \propto Q^{j}_{i} \qquad (2.2)$$

from Equation (2.1) and (2.2)

$$P^{j}_{tok,i} \propto Q^{j}_{i} \qquad (2.3)$$

With regard to the disadvantages of the exhaustive service, as we discused earlier the station holding the token has to keep track of the recent activity on the network. It means, if other stations on the network become active, it has to reduce its token holding time (or packets per token).

A timer at every station keeps track of the recent activities on the network as following.

Let's assume that

$t^{j}_{p,i}$= time the j^{th} station passes the token to the $(j+1)^{th}$ station in i^{th} rotation. Let it be the starting time of i^{th} rotation.

$t^{j}_{r,i}$ = time the j^{th} station receives the token from the $(j-1)^{th}$ station in i^{th} rotation.

t_{ideal}= time taken by the token for one rotation without transmitting any messages.

$t^{j}_{t,i}$ = total time spent in i^{th} rotation for the j^{th} station.

$$t^{j}_{t,i}=t^{j}_{r,i} - t^{j}_{p,i} \tag{2.4}$$

$t^{j}_{l,i}$= time token arrived late in the i^{th} rotation for j^{th} station. Or, in other words, the time spent in transmitting packets on the channel in i^{th} rotation by other stations on the network. Evidently,

$$t^{j}_{l,i}=t^{j}_{t,i} - t_{ideal} \tag{2.5}$$

L^{j}_{i}= the number of packets the token arrived late in the i^{th} rotation for the j^{th} station. Thus,

$$L^{j}_{i} = \frac{t^{j}_{l,i}}{t_{pac}} \tag{2.6}$$

Here, L^{j}_{i} indicates the global activity of the network in units of packets. As L^{j}_{i} increases, the station has to reduce its token holding time or equivalently the number of packets to be transmitted per token. In this way by reducing $P^{j}_{tok,i}$ we can keep control in heavily loaded stations.

$$P^j{}_{tok,i} \propto \frac{1}{L^j{}_i} \tag{2.7}$$

From Equations (2.3) and (2.7)

$$P^j{}_{tok,i} \propto \frac{Q^j{}_i}{L^j{}_i} \tag{2.8}$$

Thus,

$$P^j{}_{tok,i} = K * \frac{Q^j{}_i}{L^j{}_i} \tag{2.9}$$

The parameter K has to satisfy that

$$1 \le P^j{}_{tok,i} \le Q^j{}_i \tag{2.10}$$

When $P^j{}_{tok,i} = 1$ the adaptive service becomes the ordinary service; when $P^j{}_{tok,i} = Q^j{}_i$ the adaptive service approaches the exhaustive service.

From the above equation we can see that if the numerator increases the global delay is reduced; if the denominator increases the local delay at other stations is reduced. In this way the adaptive service gives a compromise between network global delay and local delay at individual stations. Moreover, in this service each station has different token holding times and it changes dynamically as load changes at the station at different times.

What needs to be mentioned here is that $P^j{}_{tok,i}$ determines how many packets are transmitted per token, so this value should be converted into how many messages are transmitted per token. The reason of doing this will be discussed in Chapter 3.

2.2.2 Throughput Performance

According to the assumption we made in 2.1.2, the total token rotation time for the adaptive service is $r_a = m/\Sigma^{n-1}_{n=0} p_{na}$. P_{na} is the number of messages transmitted by the n^{th} station when holding the token in adaptive service. P_{na} has the value between P_{no} and P_{ne} ($P_{no} \leq P_{na} \leq P_{ne}$) according to equation (2.10). So r_a will be bigger than r_e and less than r_o. This leads to the conclusion that the exhaustive service has the highest throughput, ordinary service has the lowest throughput and adaptive service is in between these two.

Chapter 3

Configuration and Simulation Model

The simulation has been done by using the Local Area Network Simulation Facility (LANSF) [7]. The implementation of this simulation job can be described in three parts. They are input model, protocol code and performance measures(output file).

3.1 Input Model

The data set for the simulator consists of a number of logically separable parts. In order they are: time section, configuration section, traffic section, protocol-specific section and exit section.

3.1.1 Time Section

The time section specifies the number of indivisible time units(ITUs) in the experimenter time unit(ETU). In this simulation model the ETU is defined as 10^7 ITUs. Thus, it may correspond to a real second, under the assumption that the "clock" of our network runs at 10Mhz.

3.1.2 Configuration Section

The configuration section defines the network backbone in the following sequence:

1) Number of stations

2) Port allocation

3) Number of links

4) Port assignments

5) Distance matrix

In our model, the network consists of 8 stations which are numbered from 0 to 7. Every station has two ports: input port and output port, through which stations are interconnected by links. For 8 stations, 8 links form the ring type of network. The port transmission rate is 1bit/ITU. The distance between two stations is 10 ITUs. If the data rate of the ring is R mbps, a bit is emitted every 1/R μsec. With a typical signal propagation speed of about 200m/μsec [6], each bit occupies 200/R meters on the ring. This means, in our model, 10ITUs correspond to 200 meters.

3.1.3 Traffic Section

In LANSF there is a traffic generator called the client. The standard client is quite flexible and it seems that all practically interesting traffic patterns are covered by its capabilities.

The traffic pattern is specified as a set of message types, each message type representing a class of messages generated according to some specific rules. For each message type, we have to supply its description. The description is a sequence of parameters which must be in the following order:

1) Options,

2) Message inter-arrival time (if the message type is bursty, this is the inter-arrival time for messages within a burst),

3) Burst inter-arrival time, and

4) Burst size.

Combinations of different options generate nonburst or burst traffic. The last two parameters are only expected for a bursty message type. To generate nonburst traffic:

1) Message inter-arrival time may be exponentially or uniformly distributed, and

2) Message length may be exponentially or uniformly distributed.

For burst traffic the same options are available for message inter-arrival time and for the message length within a burst and for the burst itself:

1) Inter-arrival time may be exponentially or uniformly distributed, and

2) Burst size (the number of messages within a burst) may be exponentially or uniformly distributed.

Inter-arrival time explicitly defines the load on the network. As we decrease the inter-arrival time between the messages (or bursts) the load on the network increases. In this way, by changing inter-arrival time we can vary the load over a selected range. All simulations are done for a load range 1-7 mbps. Simulations for nonburst traffic are done with different exponential inter-arrival time and uniformly distributed message lengths from 2000 bits to 10,000 bits. For burst traffic the simulations are done with 10 ITUs exponential inter-arrival time between messages, uniformly distributed message lengths from 2000 bits to 10,000 bits, different exponential burst inter-arrival time for different load and uniformly distributed burst size of 20 messages.

Two more parameters are needed to complete the procedure of generating messages

1) sender of message, and

2) receiver of message.

By assigning weight to the stations, we can create symmetric or asymmetric traffic. The weight is a non-negative number which specifies the relative frequency of selecting a particular station as a sender of message. We will discuss symmetric and asymmetric in 3.1.5.

3.1.4 Protocol-specific and Exit Section

In this section, protocol-specific values like packet length, header and trailer information, token length, interpacket space and other necessary values are given. We have done simulation for 128 header bits and 32 trailer bits. Token length and interpacket space are specified as 24 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 unit, and

3) CPU time limit.

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

3.1.5 Quantify Asymmetry

We have done simulation for both symmetric and asymmetric type of traffic on the network. For symmetric traffic every station has same probability to be a sender, so that messages generated by the client are distributed evenly to all stations.

In order to evaluate the degree of asymmetry, for simplicity, we assume that one station in the network generates certain percent of the total traffic, and the rest of the traffic is evenly distributed among the other stations.

Let us define a parameter **a** which evaluates the degree of asymmetry as follow:

$$\mathbf{a}= \sqrt{\frac{\sum_{i=1}^{n} (R_i - \bar{R})^2}{n\bar{R}^2}} \quad , i<> k; \qquad (3.1)$$

where R_i is the load distributed to each station

$\bar{R}$ is the average load for each station

n is the total number of stations in the network

k is the station which makes the network asymmetric

When $R_i=\bar{R}$, **a**=0. This corresponds to the symmetric case.

When R_i decreases, $R_i<R$, **a** increases. The percentage of traffic which is distributed to station k increases.

When $R_i<<\bar{R}$, **a**--> $\sqrt{(n-1)/n}$. Almost 100% of the traffic is assigned to station k. If n is big enough **a**≈1.

Obviously, the bigger the value **a** is, the higher the degree of asymmetry is.

The Table 3.1 shows how the load distributions change when **a** changes. Station 0 is the special station which makes the network asymmetric in our simulation.

3.2 Protocol Codes for Simulation Model

In LANSF, the protocol is expressed by the program that consists of two C files. One file contains mainly declarations of user defined symbolic constants and another file contains code of different processes. Here, we are mainly interested in the code of channel access to transmit messages for different services.

a\St	St 0	St 1	St 2	St 3	St 4	St 5	St 6	St 7
a=0	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%	12.5%
a=0.1	22.3%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%	11.1%
a=0.2	31.2%	9.83%	9.83%	9.83%	9.83%	9.83%	9.83%	9.83%
a=0.3	41.6%	8.49%	8.49%	8.49%	8.49%	8.49%	8.49%	8.49%
a=0.4	49.9%	7.15%	7.15%	7.15%	7.15%	7.15%	7.15%	7.15%
a=0.561	65.0%	5.0%	5.0%	5.0%	5.0%	5.0%	5.0%	5.0%
a=0.6	68.6%	4.48%	4.48%	4.48%	4.48%	4.48%	4.48%	4.48%
a=0.7	78.0%	3.14%	3.14%	3.14%	3.14%	3.14%	3.14%	3.14%
a=0.8	87.3%	1.81%	1.81%	1.81%	1.81%	1.81%	1.81%	1.81%
a=0.9	96.7%	0.47%	0.47%	0.47%	0.47%	0.47%	0.47%	0.47%

Table 3.1 Traffic Distributions vs **a**

3.2.1 Brief Review of Three Services

In ordinary service each station transmits one message per token on the channel, if it has messages waiting in the queue.

In exhaustive service, the station transmits messages until its message queue is empty and then it releases the token to the next station.

In the adaptive service, according to equation 2.9, the number of packets transmitted per token of each station is defined by the formula

$$\text{packets per token}= \frac{\text{queued\_packet}}{\text{tok\_pac\_late}} \qquad (3.2)$$

in which queued_packet means the number of packets queued in the station which holds the token and tok_pac_late means the time, in terms of packets, spent in transmitting packets on the channel in the very last rotation of the token by other stations on the network while packets per token means that the number of packets transmitted per token by the station.

In the case of single-packet message, the least transmission unit per token is one packet. In the multi-packet message case, the least transmission unit per token is one message.

3.2.2 Ordinary and Exhaustive Service

The partial pseudo code of the program for ordinary service is as follows:

```
Case TRANSMIT_OWN_PACKET:
  get the length in bits of the first message in the queue;
  get the packets per token by converting the length into
  packets;
  if(any message is in the queue, then get the first packet,
    add header and trailer and store it in packet buffer)
  then
      begin
         transmit packet to the output port;
         continue at case PACKET_TRANSMITTED;
     end
  else
     continue at case PASS_TOKEN;
Case PASS_TOKEN:
  transmit token to the output port;
  reset the counter;
  continue at case TOKEN_PASSED;
Case TOKEN_PASSED:
  stop transfer at output port;
  continue at case INITIALIZE;
Case PACKET_TRANSMITTED:
  stop transfer at output port;
  release packet buffer;
  increase counter by one;
  if(counter equals to the packets per token)then
   wait for interpacket space and continue at case
   PASS_TOKEN;
  else
   wait for interpacket space and continue at case
   TRA_PK_AGN;
Case TRA_PK_AGN:
  if (any message is in the queue, then get the first, and
     header and trailer and store it in packet buffer)then
     begin
       transmit packet to the output port;
       continue at case PACKET_TRANSMITED;
     end
  else
    continue at case PASS_TOKEN;
```

Above code shows how ordinary service is implemented in LANSF. This process is suitable for both single-packet and multi-packet message case.

Exhaustive service has almost same type of code. Only one case is different.

```
Case PACKET_TRANSMITTED:
  stop transfer at output port;
  release packet buffer;
  wait for interpacket space;
  continue at case TRANSMIT_OWN_PACKET again;
```

In this way, the program will be in the loop until there is no more message in the queue at the station.

3.2.3 Adaptive Service

The partial pseudo code of the program for adaptive service is as follows:

```
Case TRANSMIT_OWN_PACKET:
  if(timer is greater than zero) then
    begin
      get total token rotation time by deducting timer from
      current_time;
      get time token arrived late by deducting ideal
      rotation time from total token rotation time;
      get the number of packets per message ;
      get the number of total queued_packets in the queue;
     if(time token arrived late is zero) then
```

```
   begin
     if(queued_packet is less than 20)then

       packets per token = queued_packet;
     else
       packets per token = n packets,where n is greater
                           than 19 and n is the total
                           number of packets of m
                           messages. m is an integer;
     end
else
   begin
     convert time token arrived late in packets token
     arrived late;
     get packets per token (packet_send) by dividing
     queued_packet to packets token arrived late (k=1);
   if(packet per token is less than the number of
      packets  in the first message in the queue) then
      packets per token = the number of packets of the
      first message;
   else
       packets per token = n packets,where n is greater
                           than (packet_send-1) and n
                           is the total number of
                           packets of m messages. m is
                           an integer.
   end
```

```
    end
else
   one packet per token;
if(any message is in the queue, then get the first, add
  header and trailer and store it in packet buffer) then
  begin
    transmit packet to the output port;
    continue at case PACKET_TRANSMITED;
  end
 else
  continue at case PASS_TOKEN;
Case PASS_TOKEN:
  transmit token to the output port;
  continue at case TOKEN_PASSED;
Case TOKEN_PASSED:
  stop transfer at output port;
  reset counter;
  note current_time into timer;
  continue at case INITIALIZE;
Case PACKET_TRANSMITTED:
  stop transfer at output port;
  release packet buffer;
  increase counter by one;
  if(counter equals to packets per token) then
    wait for interpacket space and continue at case
    PASS_TOKEN;
  else
```

```
    wait for interpacket space and continue at case
    TRA_PK_AGN;
Case TRA_PK_AGN:
  if(any message is in the queue, then get the first, and
     header and trailer and store it in packet buffer)
    then
      begin
         transmit packet to the output port;
         continue at case PACKET_TRANSMITTED;
      end
  else
     continue at case PASS_TOKEN;
```

First part of the case TRANSMIT_OWN_PACKET calculates packets per token. In multi_packet message case, each station transmits at least one message, if it has messages waiting in the queue. So the station checks the value of packet_send. If packets per token is less than the length of the first message, it will be given the value of the first message length. If packets per token is larger than the length of the first message, it will be given the cumulative length of message 1, message 2 and up to message m. m satisfies the condition that the cumulative length is bigger than packet_send only once. This means that the cumulative length of m-1 messages will be less than packets per token. We also assume that for very first round every station can send one packet per token.

3.3 Performance Measures Produced by LANSF

Two important performance measures of a network are its delay measures and throughput information.

3.3.1 Delay Performance

Three delay measures used by LANSF are:

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 node to the moment the entire message (its last packet) is successfully received at the destination.

2. The weighted message delay of message M, denoted by $d_m(M)$, is calculated as the delay of single information bit measured (in ETUs) since the time M was queued at the sending station, to the moment when the packet containing that bit is successfully received at the destination.

3. The absolute packet delay of packet P, denoted by dp(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 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 $P^j{}_1, \ldots, P^j{}_k$ with lengths $l^j{}_1, \ldots\ l^j{}_k$, respectively. Let $l^j = \Sigma^{kj}{}_{i=1} l^j{}_i$ denote the length of M^j. Message M^j was queued at the sender at time tq^j; its i'th packet $p^j{}_i$ became ready for transmission at $tt^j{}_i$ and was completely received by the target station at $tr^j i$. The three delays mentioned above are calculated according to the following formulas:

$$d_s(M^j) = tr^k{}_{kj} - tq^j, \tag{3.3}$$

$$dm(Mj) = \frac{\Sigma\ {}^{kj}{}_{i=1}(tr^j{}_i - tq^j) * l^j{}_i}{l^j} \tag{3.4}$$

$$d_p(P^j{}_i) = tr^j{}_i - tt^j{}_i \tag{3.5}$$

The time when a packet becomes ready for transmission ($tt^j{}_i$) is determined as the maximum of the following two values:

- The time when the buffer the packet is acquired into was last released,

- 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 average absolute message delay for the n messages $M^1, \ldots,$ M^n is computed as:

$$d^a{}_s(M^1, \ldots, M^n) = \frac{\Sigma^n{}_{i=1}\ d_s(M^i)}{n} \tag{3.6}$$

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:

$$d^a{}_p(< P^j{}_1, \ldots, P^j{}_{kj} >^n{}_{j=1}) = \frac{\Sigma^n{}_{j=1} \Sigma^{kj}{}_{i=1}\ d_p(P^j{}_i)}{\Sigma^n{}_{j=1}\ k_j} \qquad (3.7)$$

With the weighted message delay, the situation is slightly more complicated. This measure is calculated individually for every information bit. Thus, in calculating the average weighted message delay the weighted delays of individual messages are further weighted by their lengths. In particular, the average weighted message delay of the n messages is determined by the formula:

$$d^a{}_m(M^1, \ldots, M^n) = \frac{\Sigma^n{}_{i=1}\ d_m(M^i) * l^j}{\Sigma^n{}_{i=1}\ l^j} \qquad (3.8)$$

In our simulation model, we compared the absolute message delay for different services. 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.

3.3.2 Throughput Performance

LANSF provides three throughput measures:

1. Global effective throughput, which is the ratio of the total number of information bits received at their destinations to the simulation time.

2. Receiver throughput of a link, which is computed as the ratio of the total number of bits received on the link to the simulation time.

3. Effective throughput of a link, which is the ratio of the total number of information bits successfully transmitted along the link to the simulation time.

We were interested in the global effective throughput for the performance comparison.

Chapter 4

Analysis of Simulation Results

Results of the simulation study are described in this chapter. For easier comparison, the delay and throughput characteristics are plotted.

4.1 Simulation Parameters

The simulation is done for 8 station network case. We tested symmetric and asymmetric traffic patterns with different combinations of message inter-arrival time and message lengths. The following are the exact traffic patterns we used in our simulation.

1. Symmetric (**a**=0) nonburst traffic with fixed message and packet length of 2000 bits.

2. Symmetric (**a**=0) nonburst traffic with varying message length from 2000 bits to 10,000 bits. The packet length is 2000 bits.

3. Symmetric (**a**=0) burst traffic with fixed message and packet length of 2000 bits and burst size of 20 messages.

4. Symmetric (**a**=0) burst traffic with varying message length from 2000 bits to 10,000 bits and fixed burst size of 20 messages. The packet length is 2000 bits.

5. Asymmetric nonburst traffic with fixed message and packet length of 2000 bits. See Table 3.1 for **a** and the traffic distributions.

6. Asymmetric nonburst traffic with varying message length from 2000 bits to 10,000 bits. The packet length is 2000 bits.

7. Asymmetric burst traffic with fixed message and packet length of 2000 bits and burst size of 20 messages.

8. Asymmetric burst traffic with varying message length from 2000 bits to 10,000 bits and fixed burst size of 20 messages. The packet length is 2000 bits.

The inter-arrival time of all the traffic patterns follows the exponential distribution, while the message length and the burst size follow the uniform distribution.

All the simulations are done with the channel capacity of 10 Mbits/sec.

4.2 Discussion of the Simulation Results

The discussion will follow the sequence of the input traffic patterns mentioned in 4.1. For performance comparison of the three services we chose parameter K=1 in adaptive protocol. The adaptive service performance vs. K is discussed separately in section 4.2.3.

4.2.1 Delay Performance Comparison

Figures 4.1 to 4.4 show the message delay vs. load for symmetric traffic patterns. When the load increases, the delay of ordinary service increases much faster than exhaustive service. And the adaptive service does provide a compromise between them. In the case of Figure 4.2, the adaptive service gives almost 19% improvement over the ordinary service at moderate load (5 Mbps).

Figures 4.5 to 4.16 show the message delay vs. load for asymmetric traffic patterns. Where **a** = 0.561 corresponds to the case that 35% of the total traffic of the network is distributed among 7 stations to form lightly loaded stations.

Among them, Figures 4.5, 4.8, 4.11 and 4.14 are the local message delay vs. load at lightly loaded stations. These graphs show that as the ~~l~~soad increases the delay curve of exhaustive service goes up much faster than the other two services because in exhaustive service the heavily loaded station keeps the token for more time and gives more delay to lightly loaded stations. In ordinary service each station can send only one message per token, so that every station keeps the token for the same amount of time. In this way lightly loaded stations are not getting any disadvantage from heavily loaded stations in ordinary service. In adaptive service the timer and the counter checks the recent activities on the network and does not allow the station to keep the token more time at heavily loaded stations. In case of Figure 4.14, the adaptive service gives as much as 40% improvement over the exhaustive service.

Figures 4.6, 4.9, 4.12 and 4.15 are the message delay vs loaded at heavily loaded station. Figures 4.7, 4.10, 4.13 and 4.16 are the global message delay vs. load. The graphs show that ordinary service gives more global delay and local delay at the heavily loaded station due to wasting time in token circulation after each transmission. The adaptive service gives as much as 60% and 71% improvement over the ordinary service at 5 Mbits/sec load in the cases shown in Figures 4.13 and 4.12 . The exhaustive service has the best delay performance which is what is expected.

Figures 4.17 and 4.18 give us a look of how global message delay changes with the parameter **a** at 5 Mbits/sec load. Generally, the adaptive service does give a compromise between the ordinary and exhaustive service.

4.2.2 Throughput Performance Comparison

Figures 4.19 to 4.26 show the throughput characteristics for the three services. These graphs reflect the conclusion we got in 2.4.2 that the exhaustive service gives the highest throughput , the ordinary service provides the lowest throughput for the network while the adaptive service behaves as a compromise between these two services.

We also notice that when the load is moderate (5 Mbps) or smaller there is no significant differences in throughput among the three services. This is because in our simulation model the traffic is assigned to the network gradually rather than instantly. When the load is low the ordinary service essentially has the capability to transmit all the messages in the queues, since most of the time only one message is there. So, ordinary, adaptive and exhaustive services take almost the same amount of time to exit the simulation under low load.

4.2.3 Delay Performance vs. K for Adaptive Service

If the network is not highly loaded there is no significant difference in throughput among the three services. So, we only discuss the delay performance vs. K (see equation 2.9) here.

From equations (2.9) and (2.10), we know when K is small enough the adaptive service will behave like the ordinary service; when K is big enough the adaptive service will perform in the role of the exhaustive service. The question is what value of K provides a better compromise between the ordinary service and exhaustive service for both local delay and global delay in asymmetric network.

Figures 4.27 to 4.34 show the delay vs. K with 5 mbps or 8 mbps load at **a**=0.561. As K increases the local delay of lightly loaded stations increases while the global delay and local delay at the heavily loaded station decreases. On the other hand, as K decreases the local delay of lightly loaded stations decreases while global delay and local delay at heavily loaded station increases. Obviously, there is no such a K with which the adaptive service can provide low local delay as well as low global delay. But we may find some Ks with which the adaptive service can provide a good compromise between the ordinary and the exhaustive services for both local delay and global delay. In figure 27, a good K is between 0.5 and 1.5. In figure 28, a good K is between 1 and 3. A K value between 0.5 to 1.5 will make adaptive service a good compromise in the case of figure 29. In figure 30, any K between 1 and 4 is good to choose. For higher traffic load (figures 4.31 to 4.34), a K value around 1.5 seems good for all traffic patterns. So we suggest to choose K around 1.5 for adaptive service in most cases.

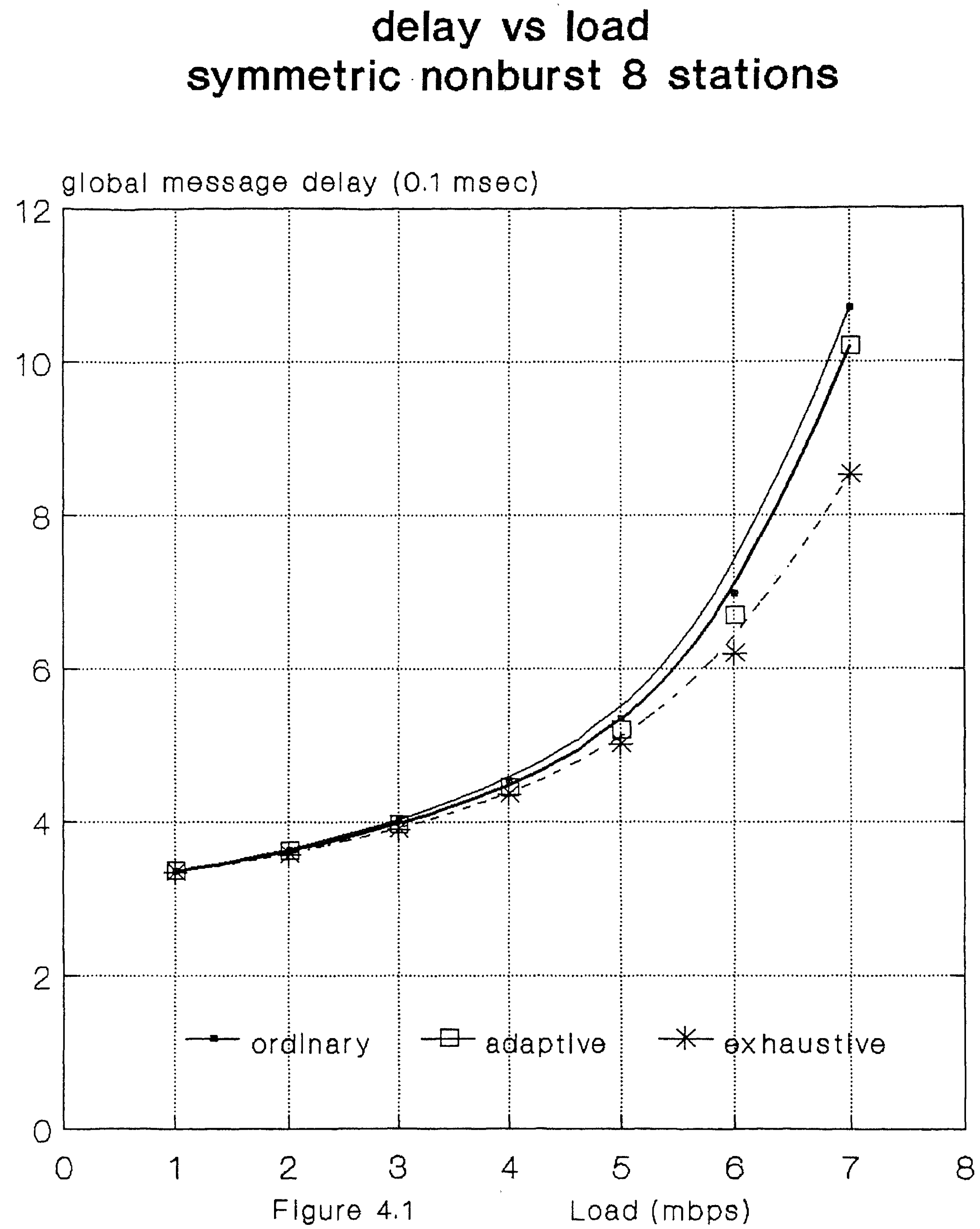


Figure 4.1

fixed message and packet length.
2000 bits/packet. a=0

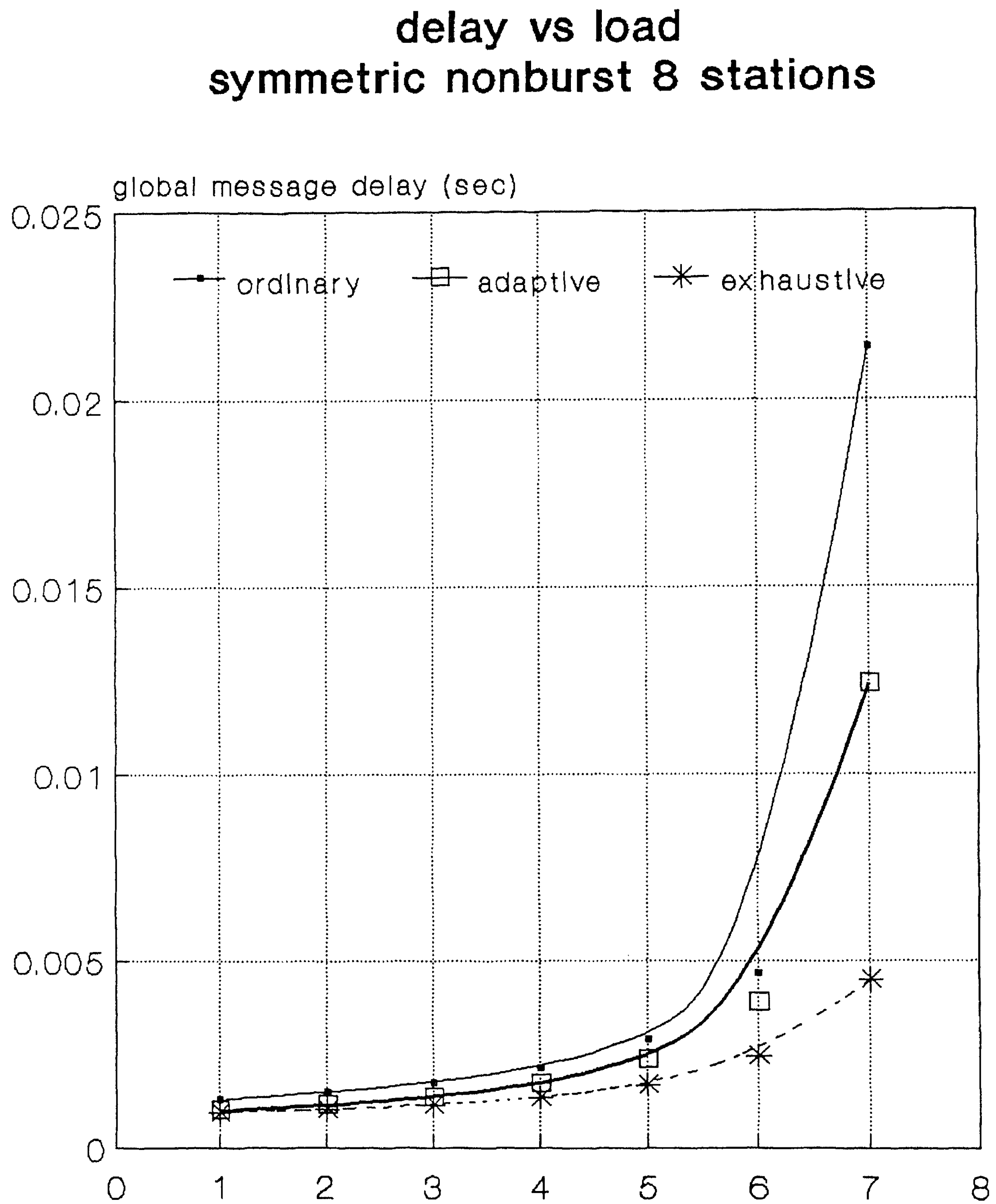


Figure 4.2

message length = 2000~10000 bits.
2000 bits/packet. a=0

delay vs load
symmetric burst 8 stations

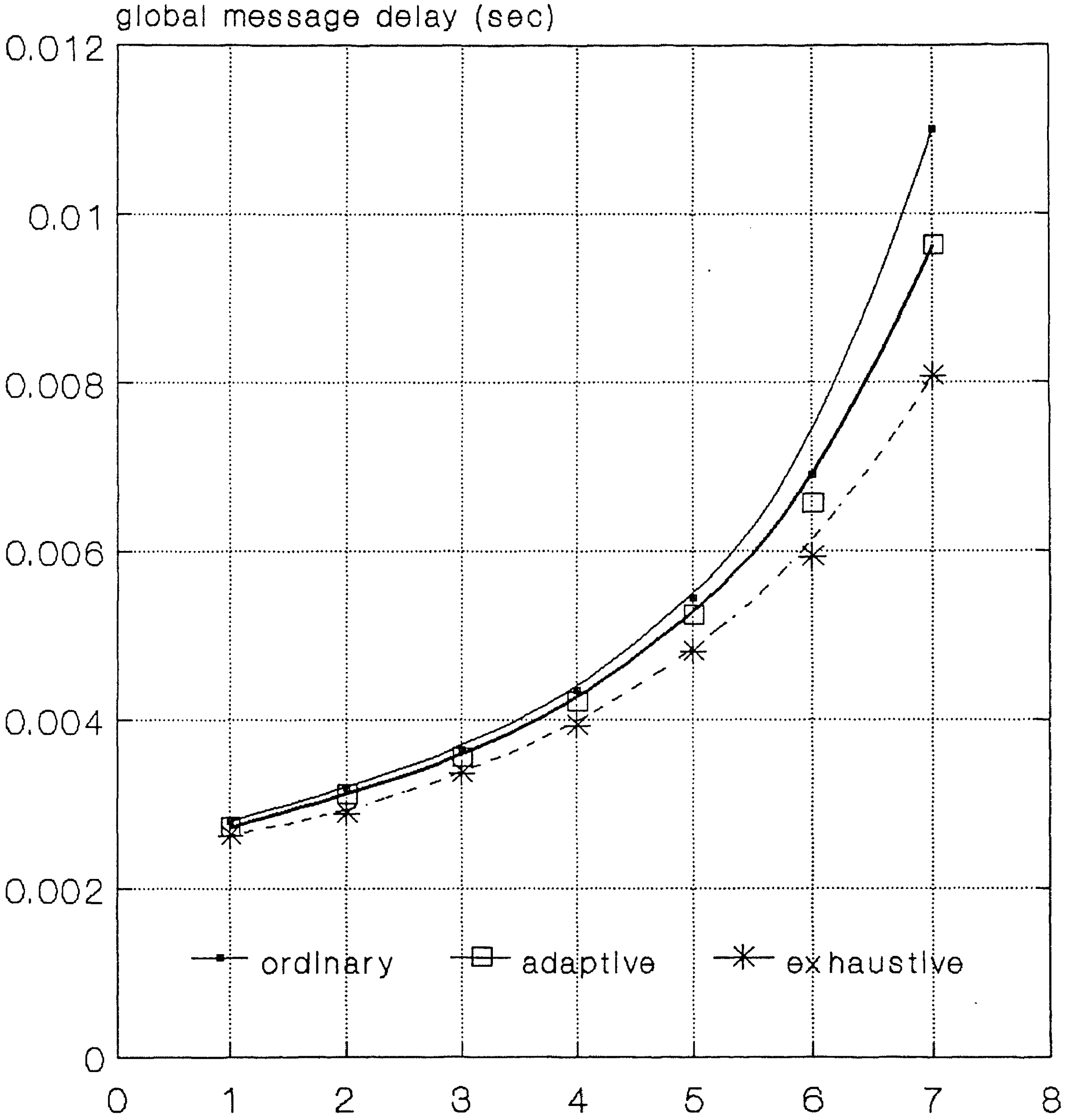


Figure 4.3

fixed message and packet length
2000 bits/packet. a=0

delay vs load
symmetric burst 8 stations

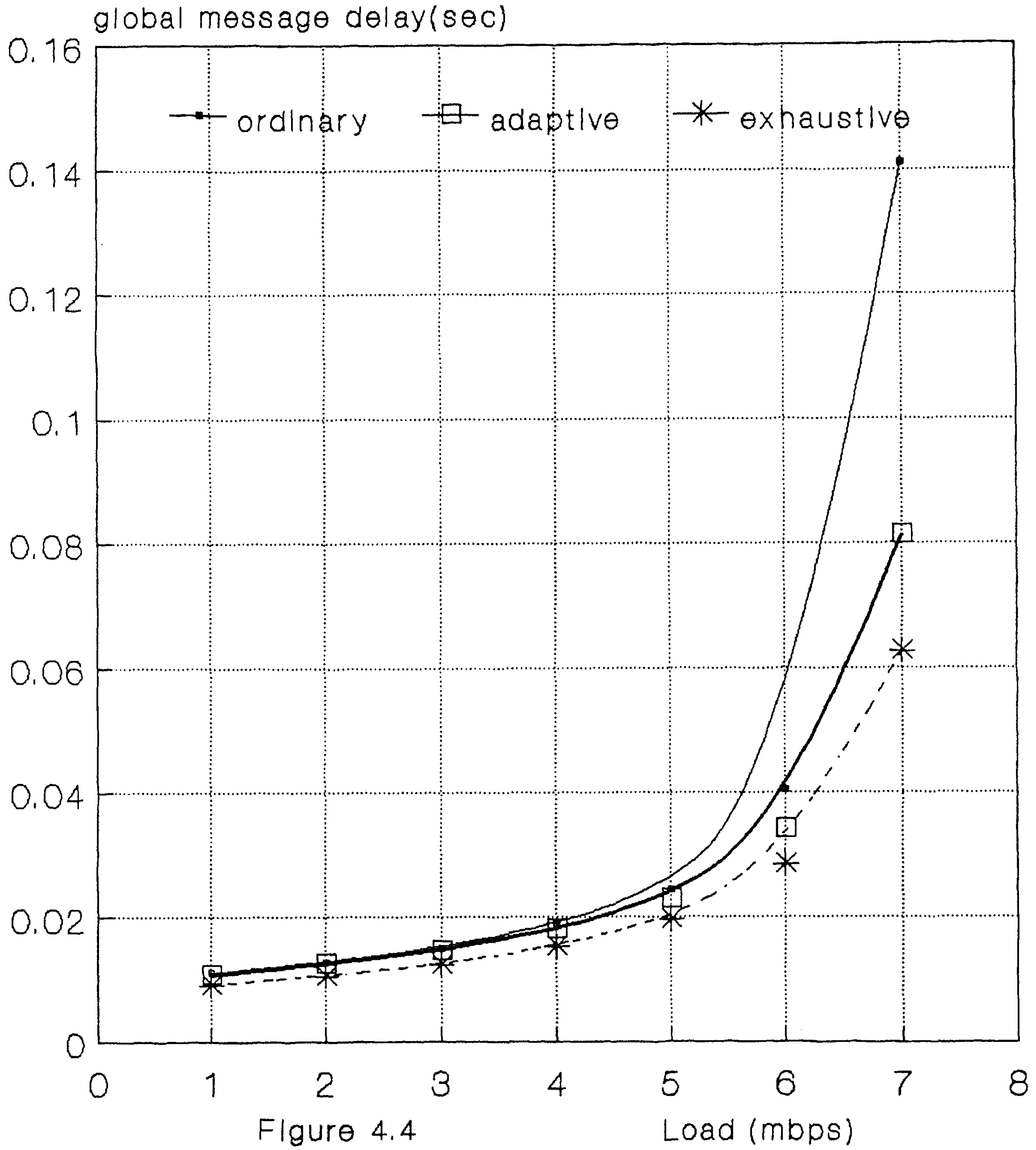


Figure 4.4

message length =2000~10000 bits
2000 bits/packet. a=0

delay vs load
asymmetric nonburst 8 stations
fixed message and packet length=2000bits

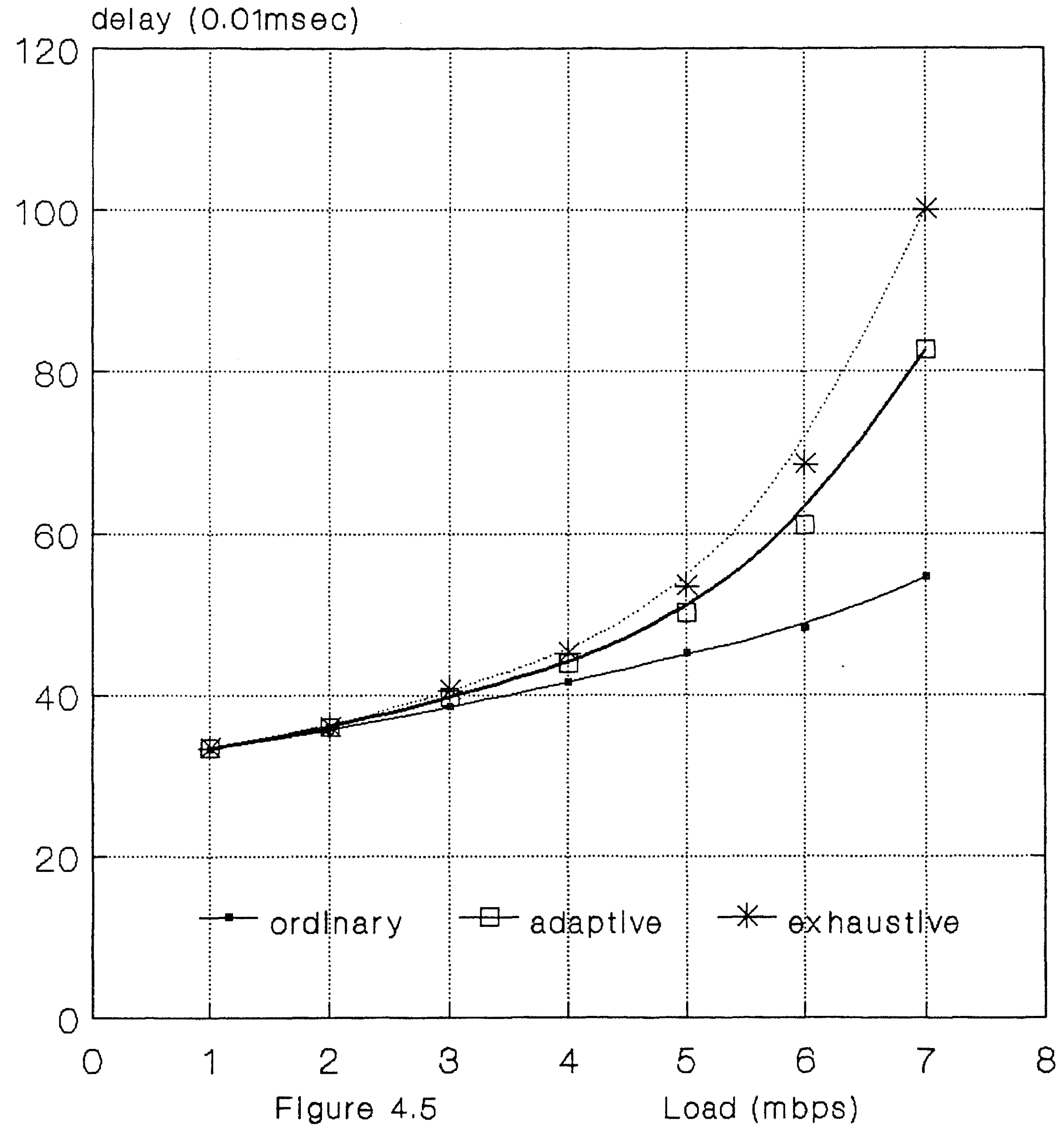


Figure 4.5

local message delay of lightly
loaed stations.2000bits/packet. a=0.561

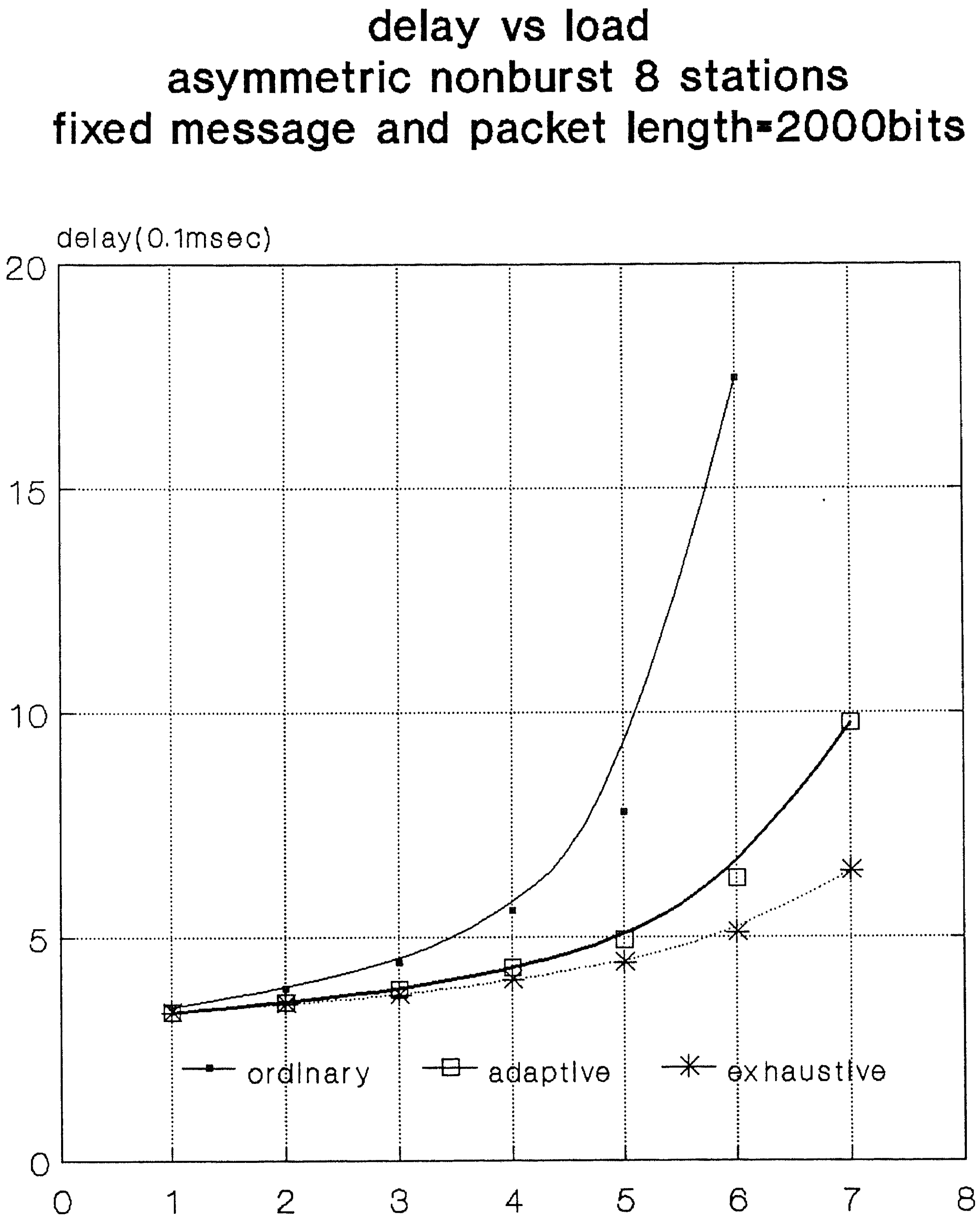


Figure 4.6

local message delay of heavily loaded station.2000 bits/packet. a=0.561

delay vs load
asymmetric nonburst 8 station
fixed message and packet length=2000bits

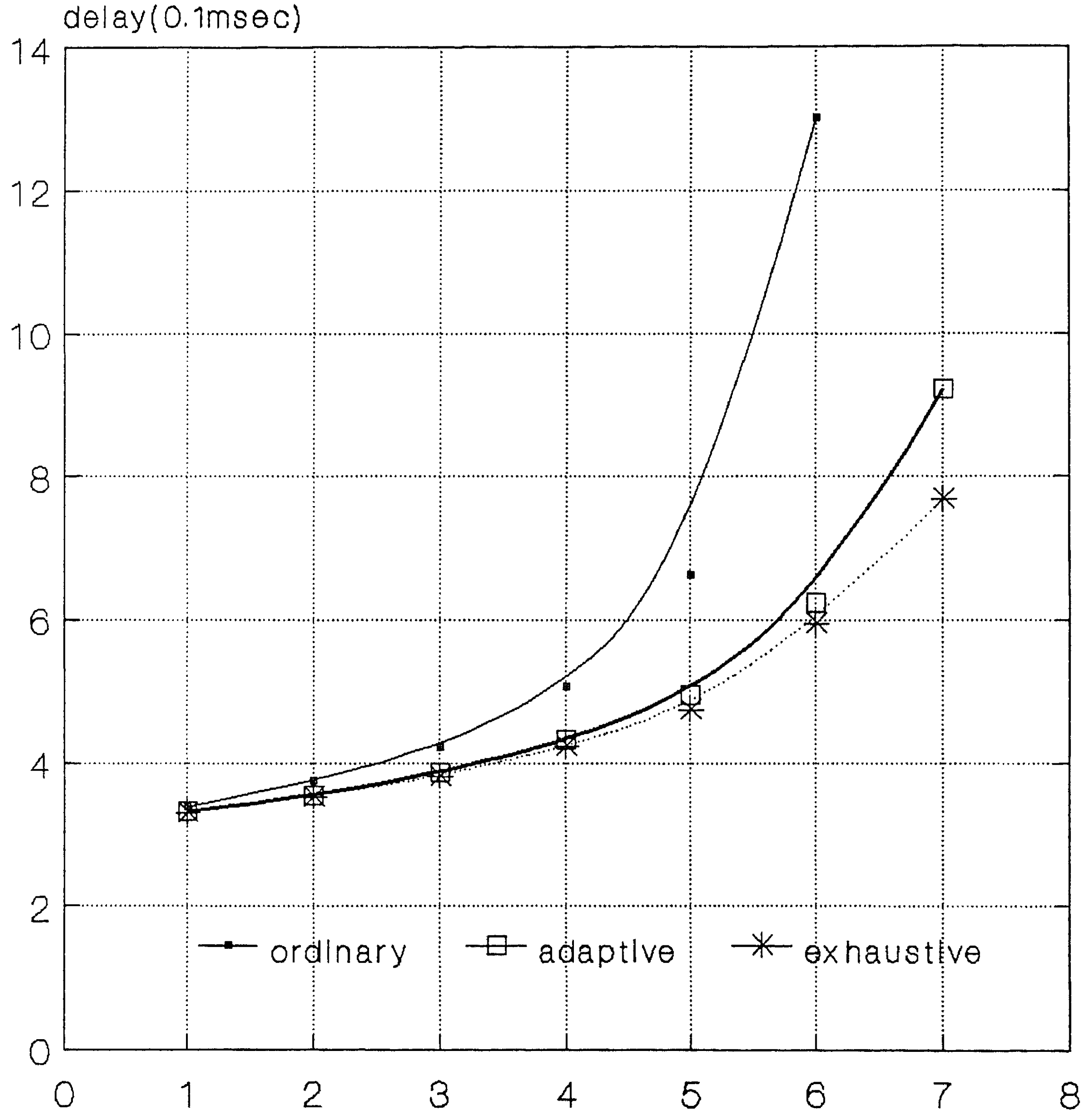


Figure 4.7

global message delay. 2000bits/packet.
a=0.561

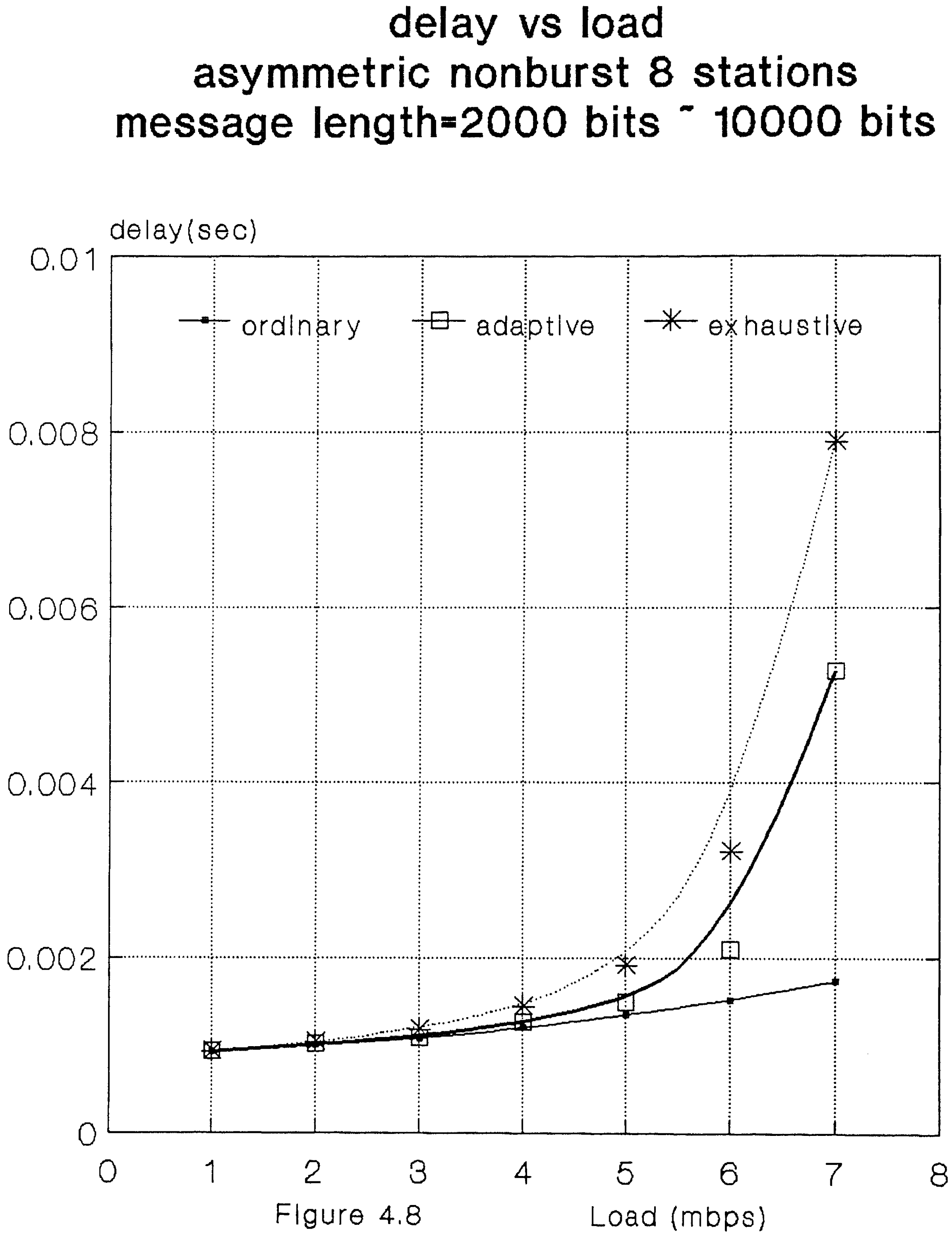


Figure 4.8

local message delay vs load at lightly loaded stations,2000bits/packet. a=0.561

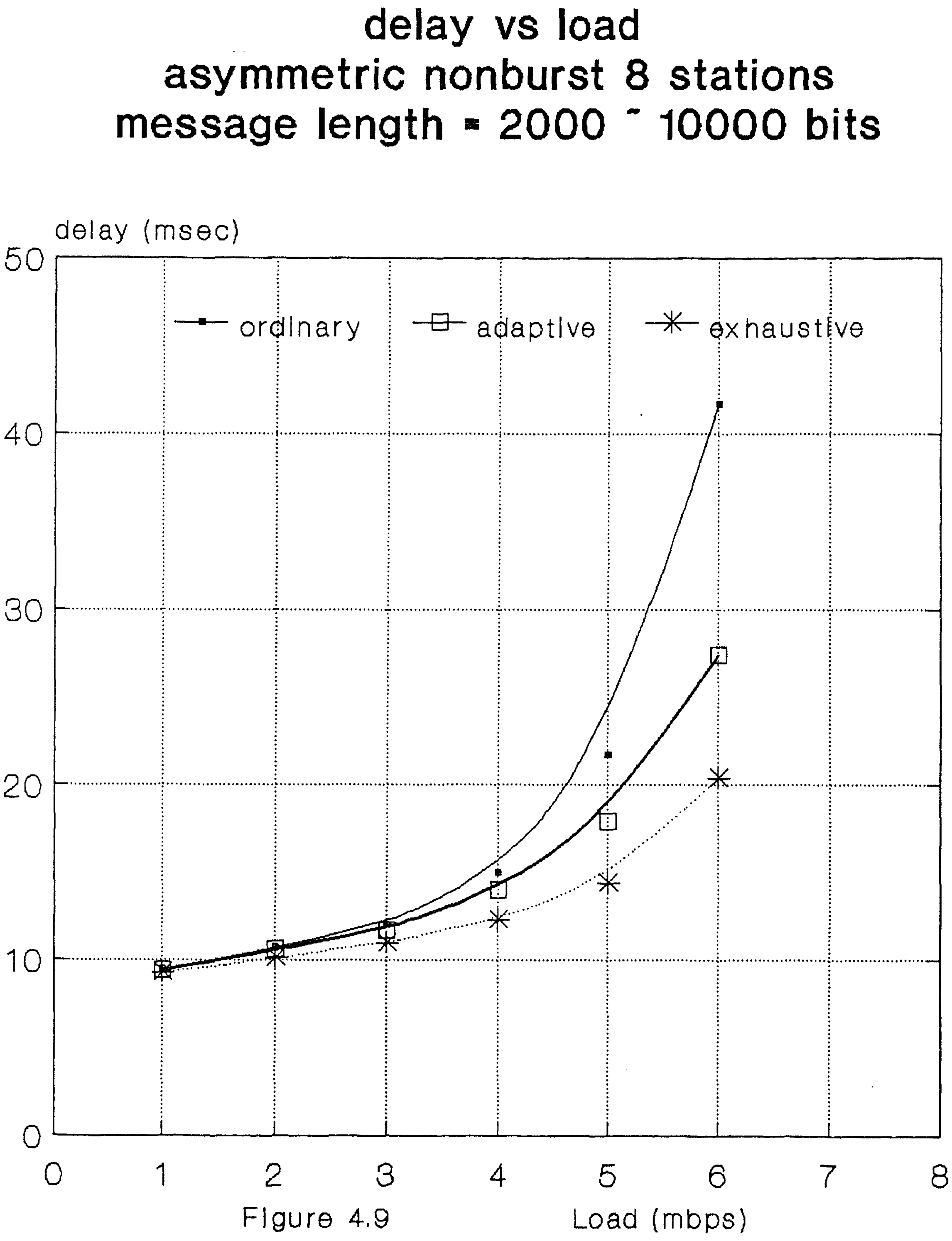


Figure 4.9

local message delay vs load at heavily loaded station. 2000 bits/packet.a=0.561

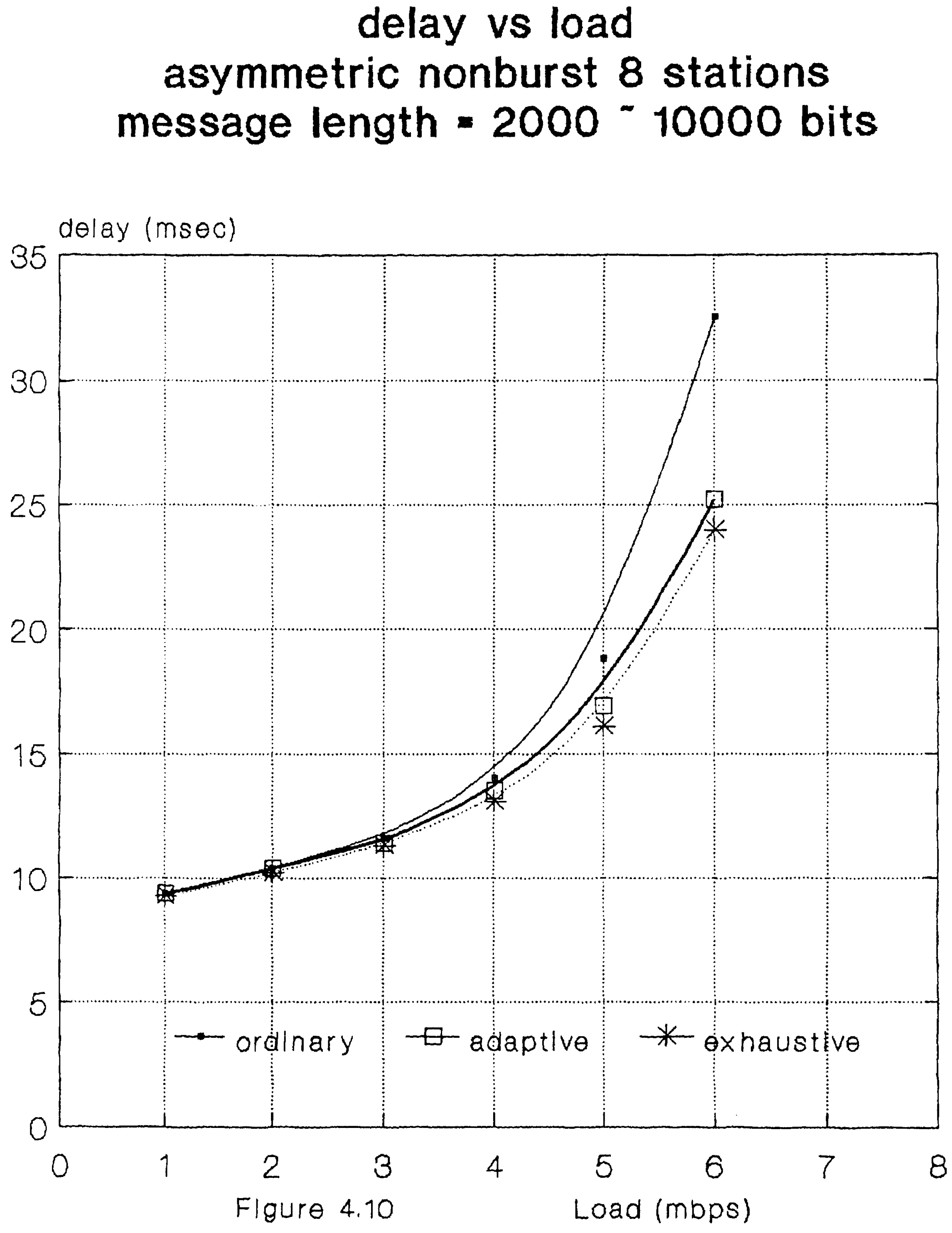


Figure 4.10

global message delay vs load.
2000bits/packet. a=0.561

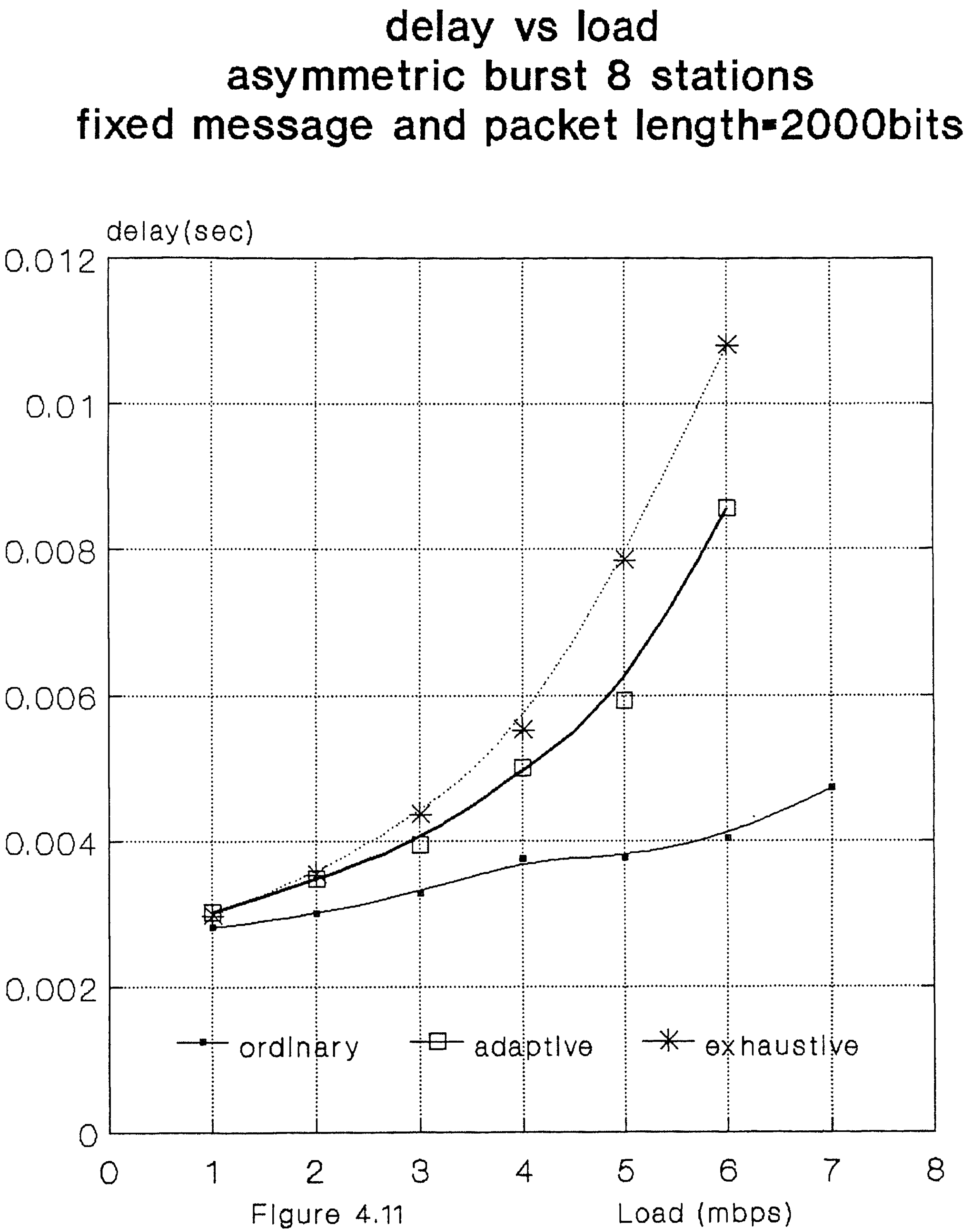


Figure 4.11

local message delay vs load at lightly loaded stations. a=0.561

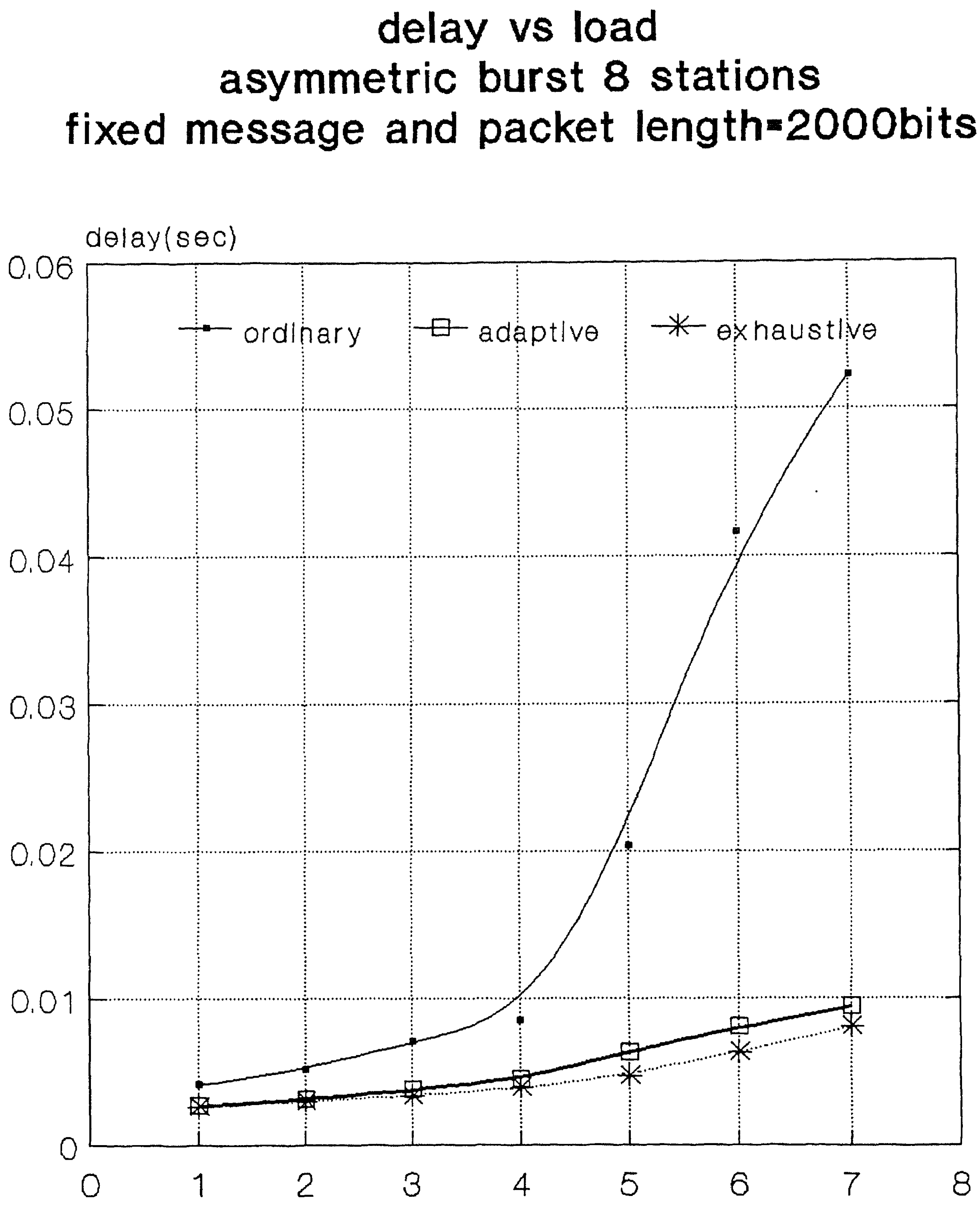


Figure 4.12

local message delay vs load at heavily loaded station. 2000 bits/packet.a=0.561

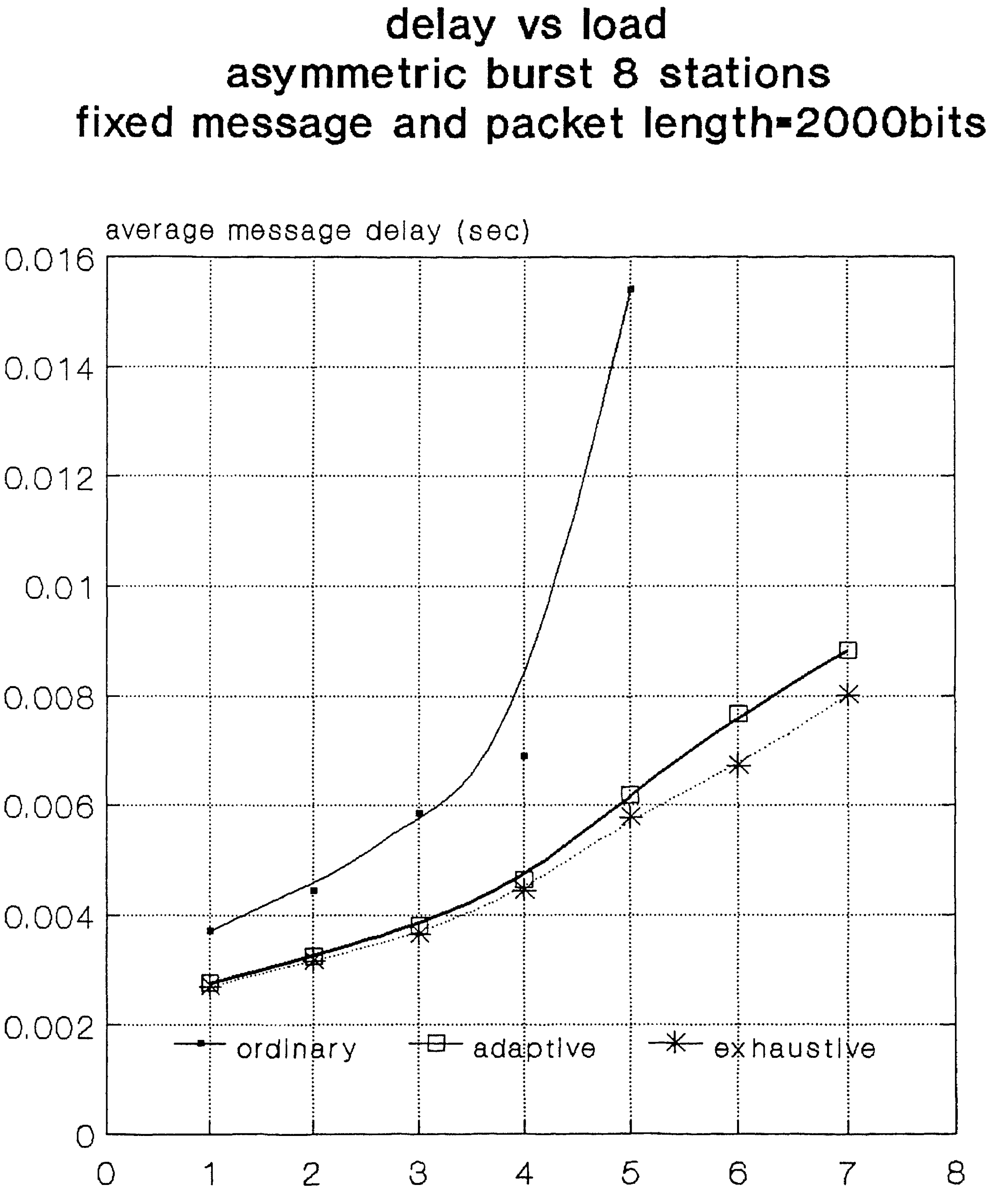


Figure 4.13

global message delay vs load
2000 bits/packet. a=0.561

delay vs load
asymmetric burst 8 stations
message length ▪ 2000 ˉ 10000 bits

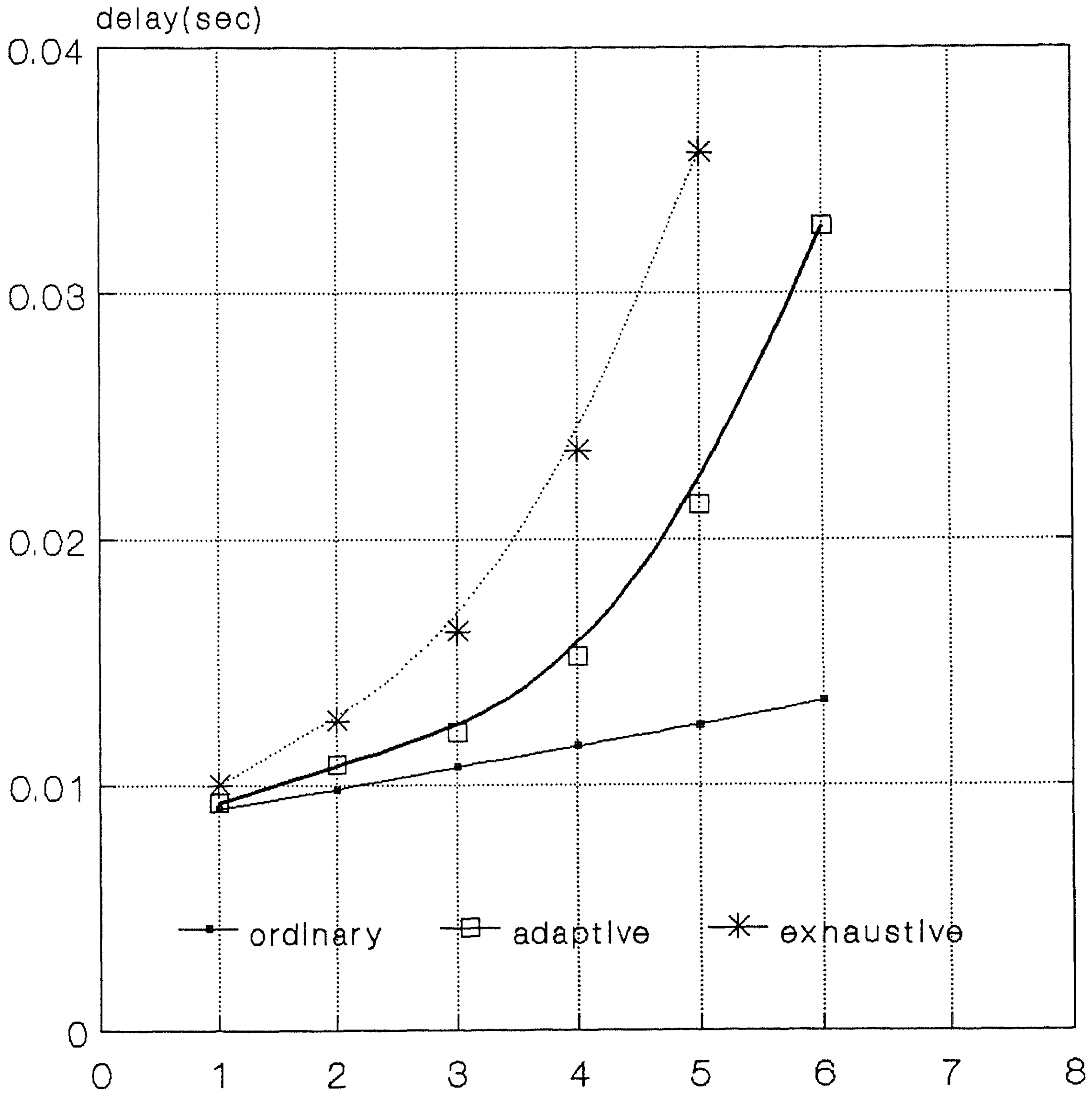


Figure 4.14

local message delady vs load at lightly loaded stations. 2000bits/packet.a=0.561

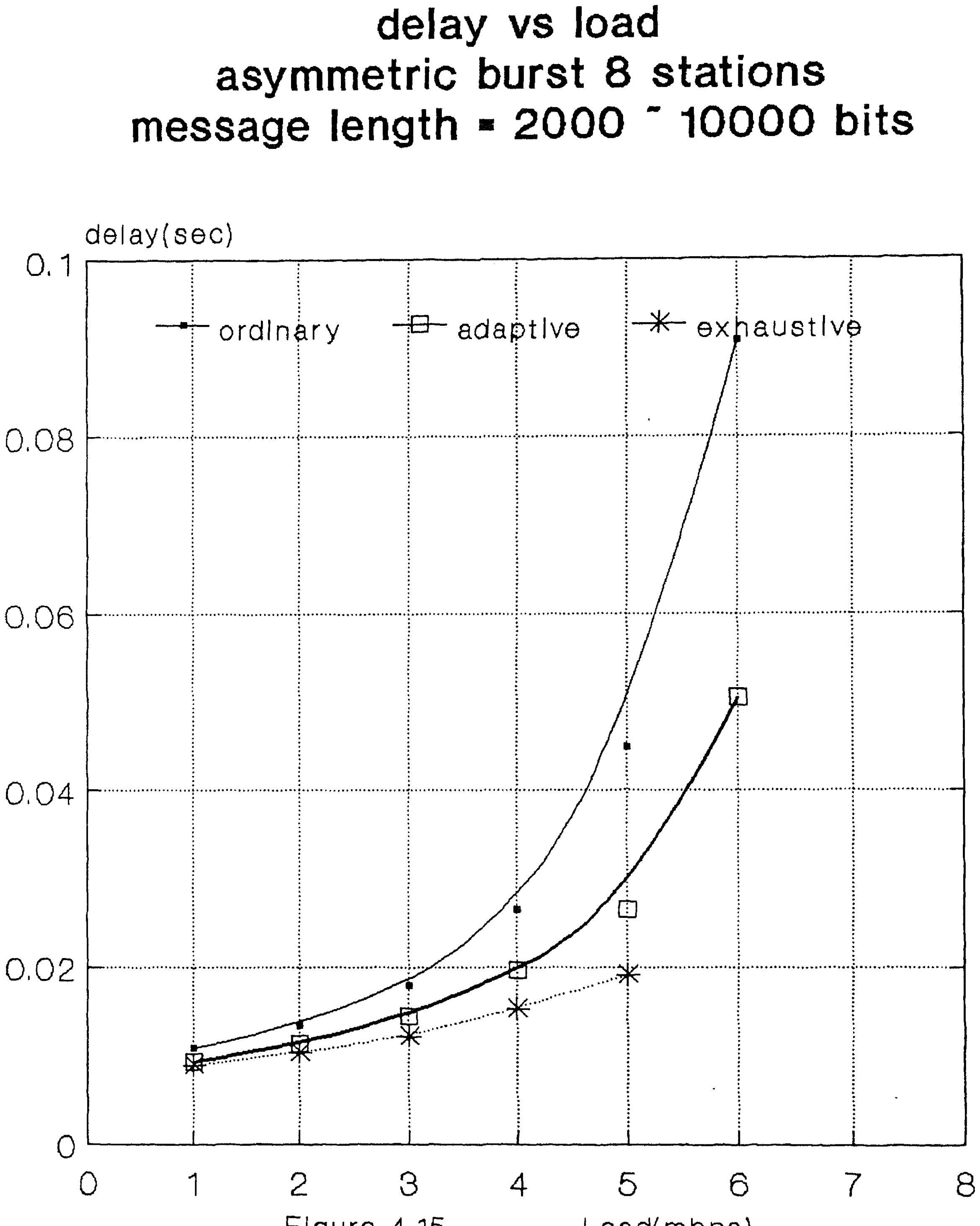


Figure 4.15

local message delady vs load at heavily loaded station.2000bits/packet.a=0.561

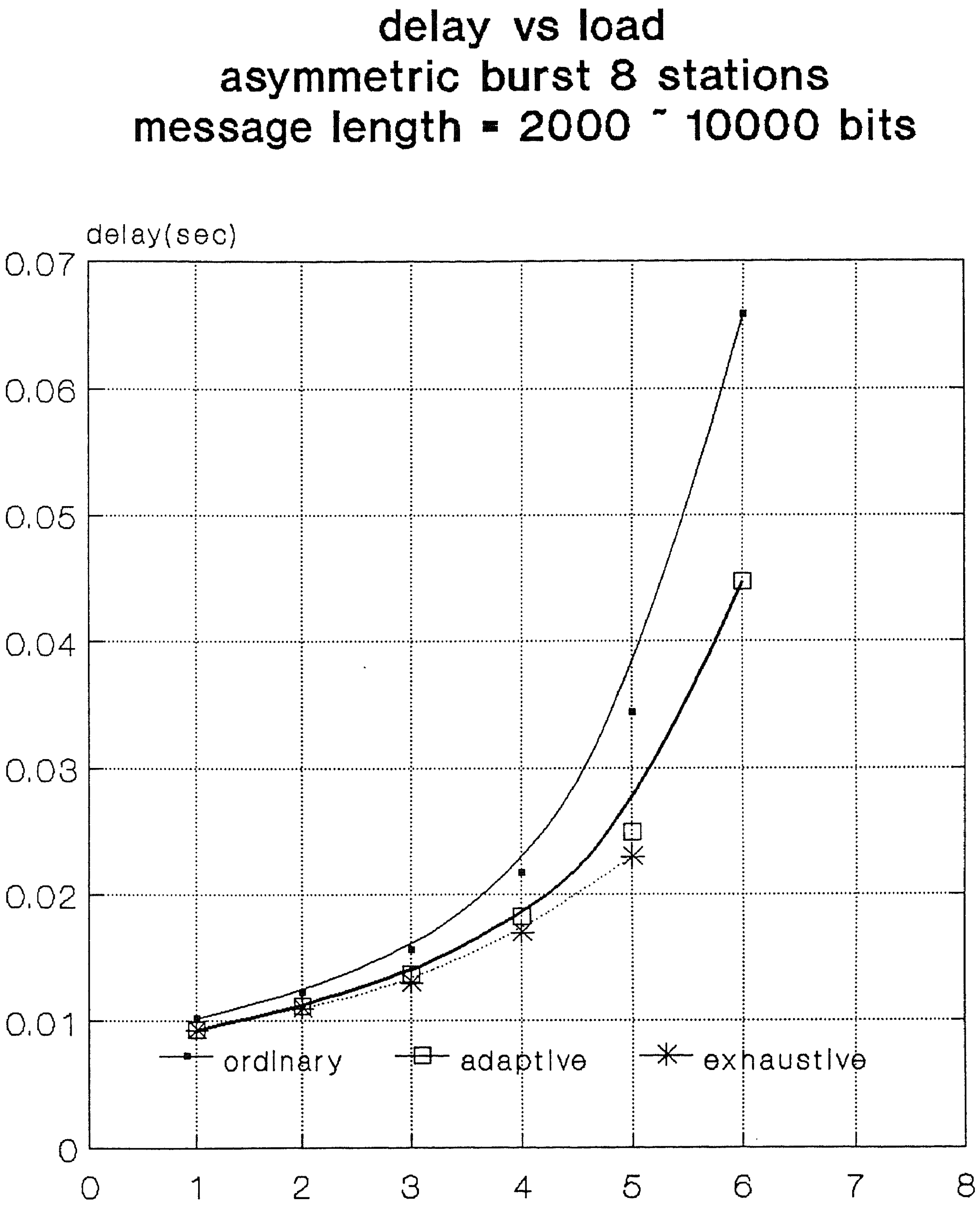


Figure 4.16

global message delay vs load.
2000bits/packet. a=0.561

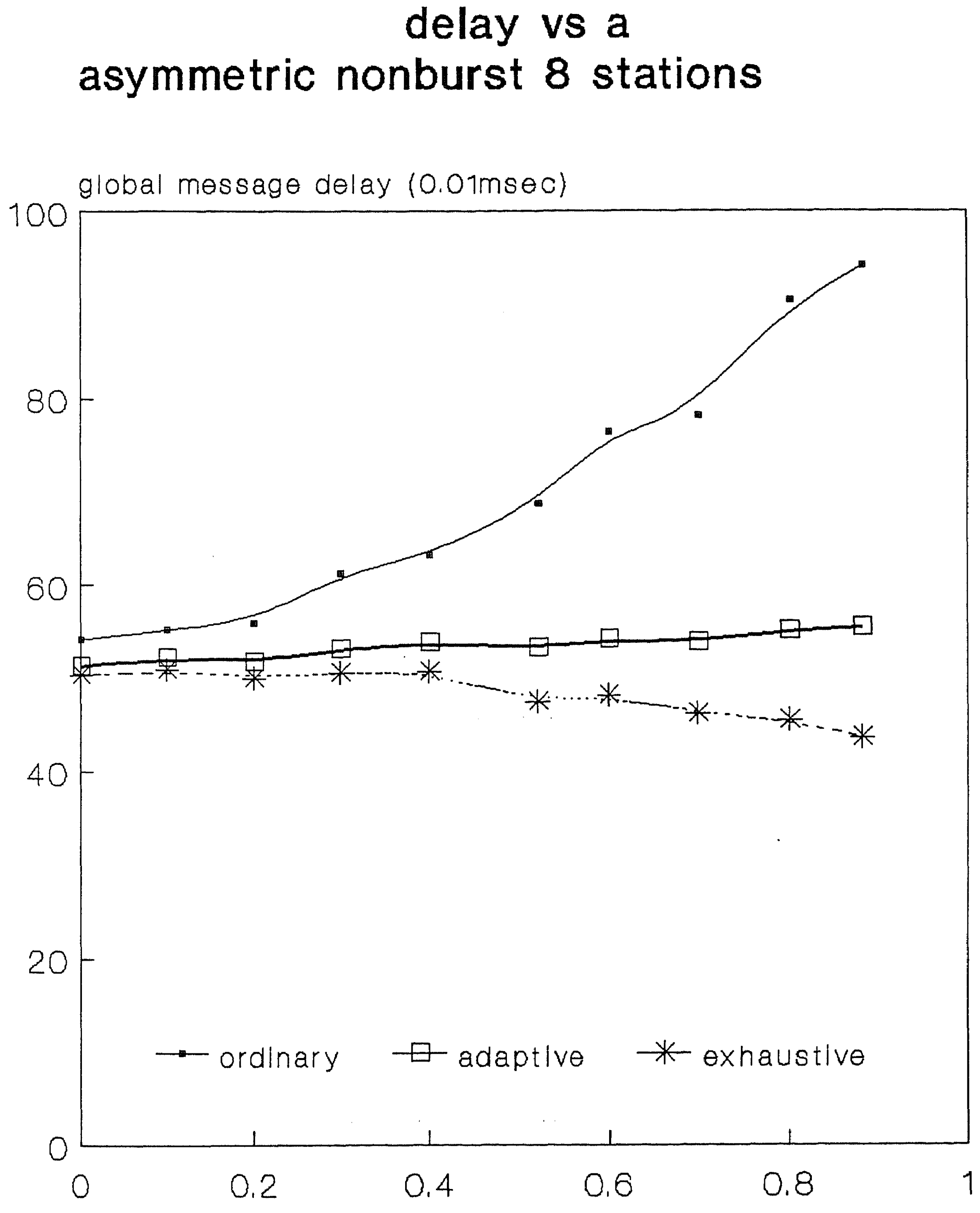


Figure 4.17

fixed message and packet length.
2000 bits/packet. at 5mbps load.

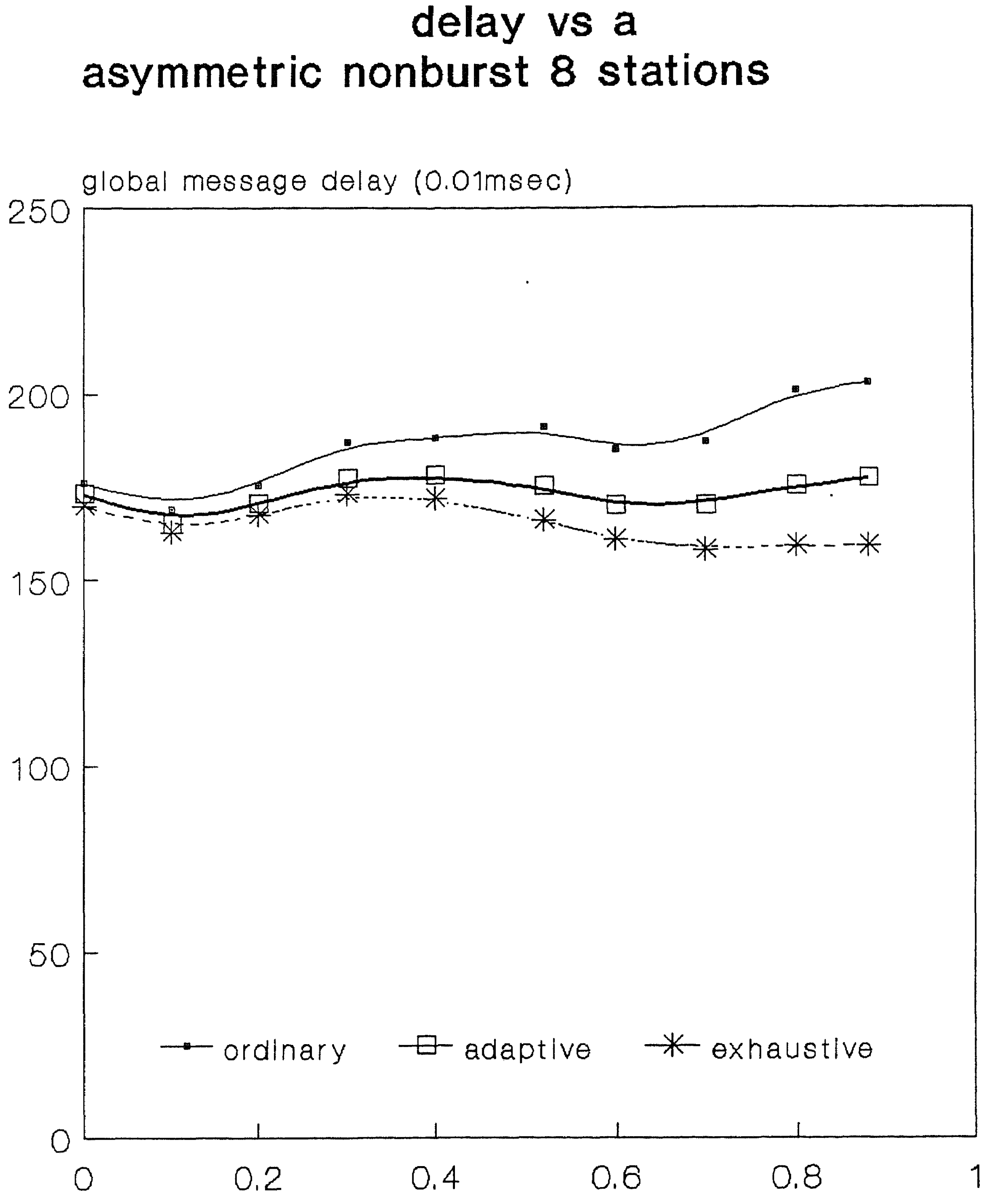


Figure 4.18

message length = 2000 ~ 10000 bits.
2000 bits/packet. at 5mbps load.

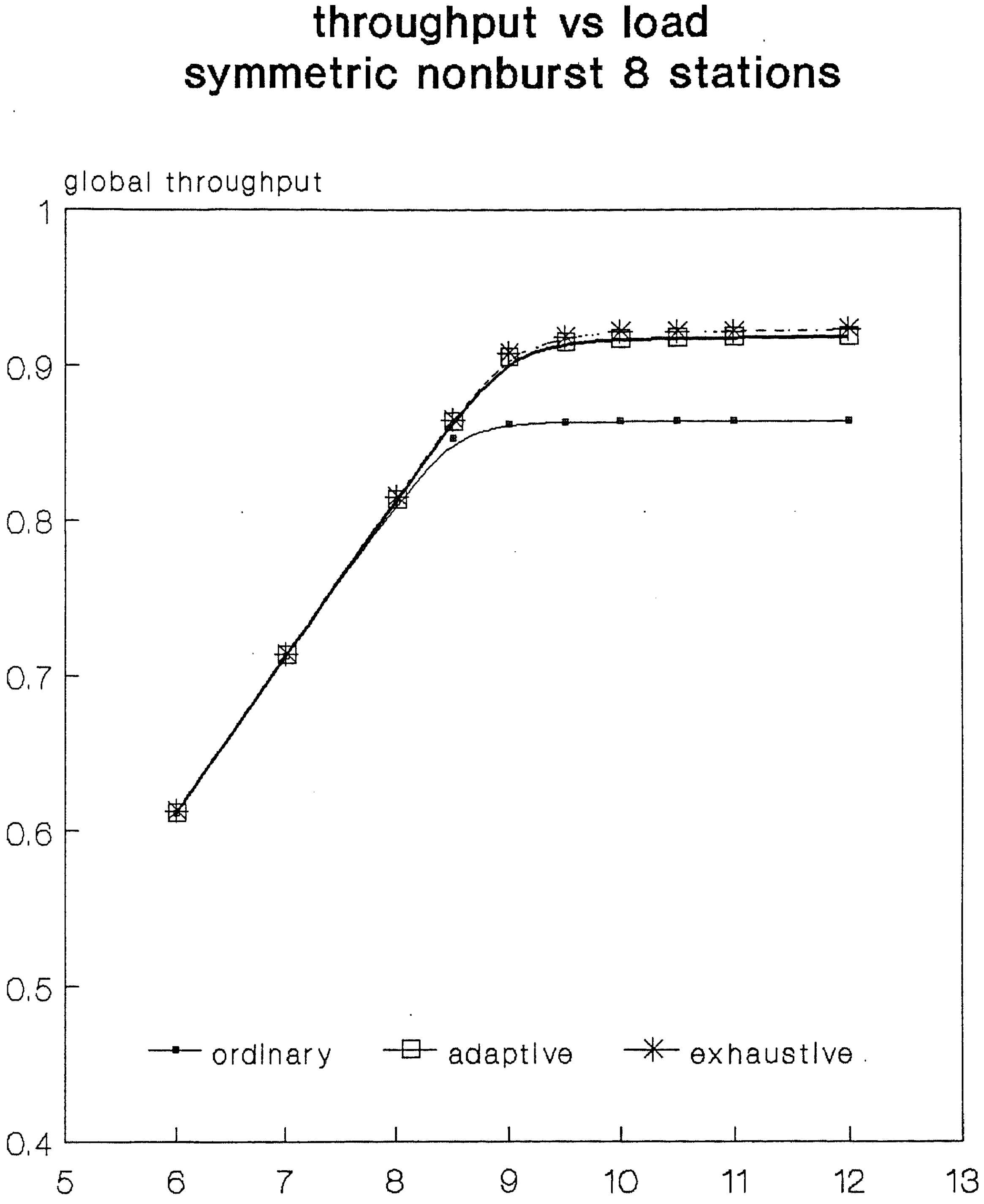


Figure 4.19

fixed message and packet length, 2000 bits/packet. a=0

throughput vs load
symmetric nonburst 8 stations

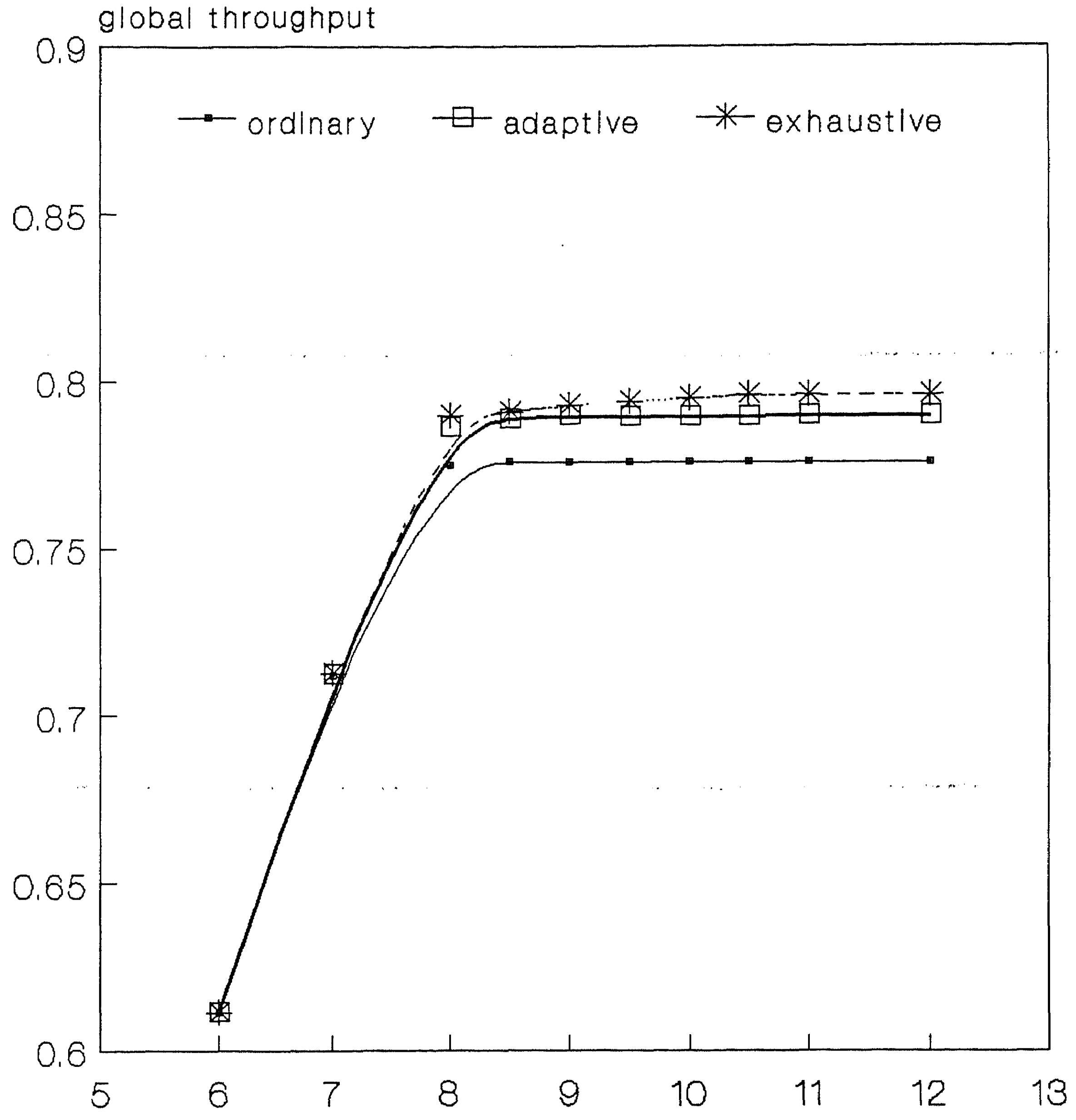


Figure 4.20

message length= 2000 ~ 10000 bits
2000 bits/packet. a=0.

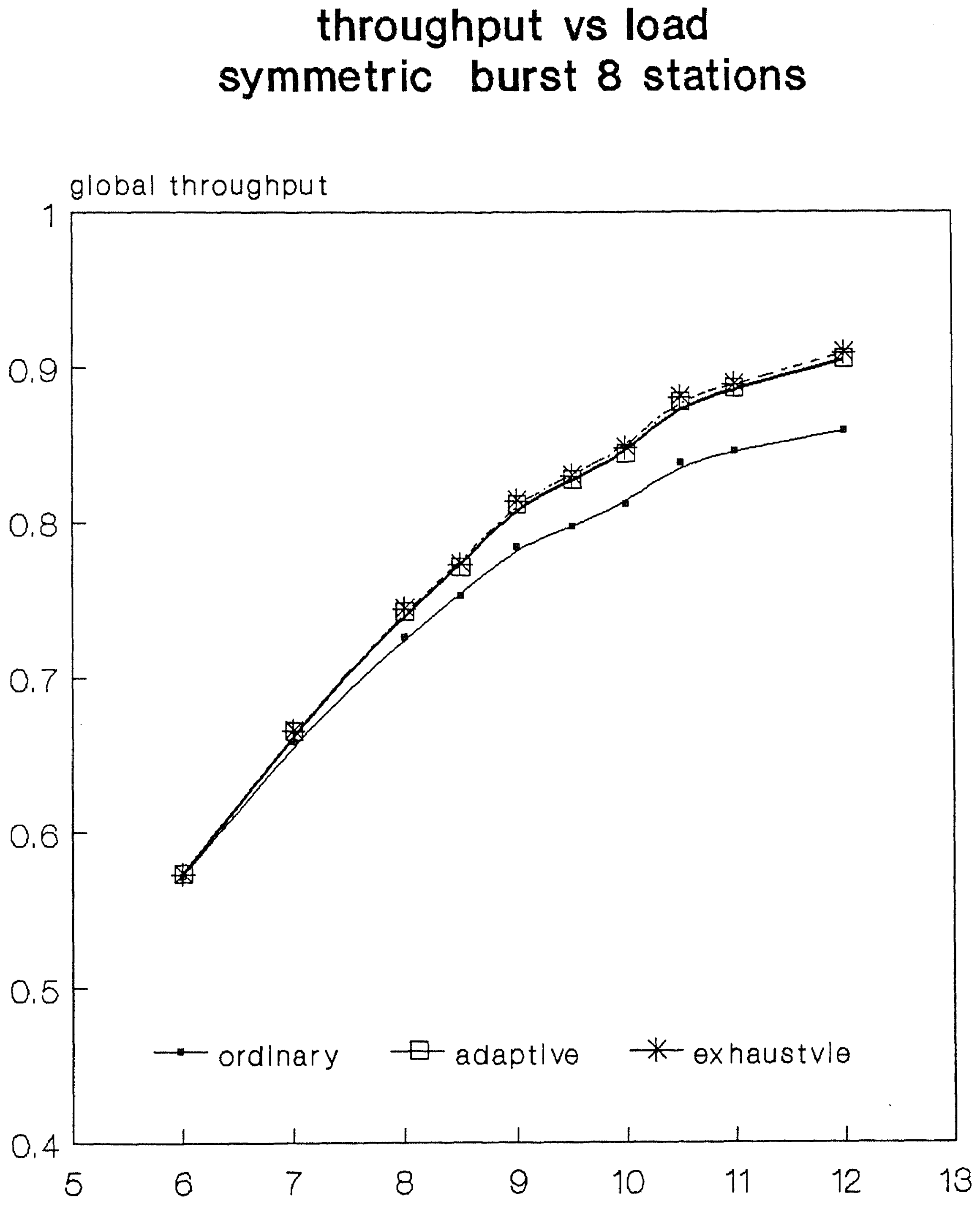
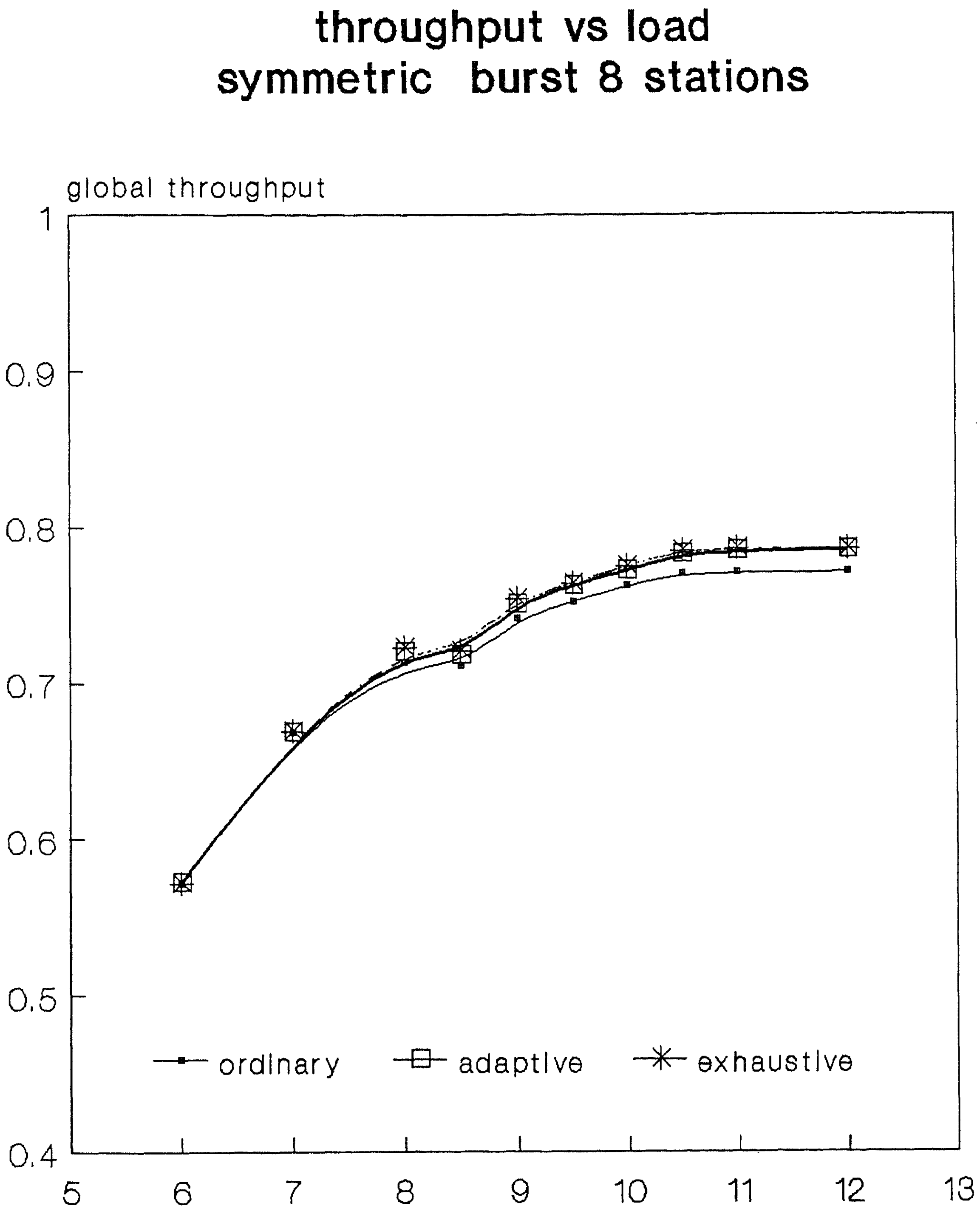


Figure 4.21

fixed message and packet length.
2000 bits/packet. a=0



throughput vs load
symmetric burst 8 stations

Figure 4.22

message length = 2000 ~ 10000 bits,
2000 bits/packet. a=0

throughput vs load
asymmetric nonburst 8 stations

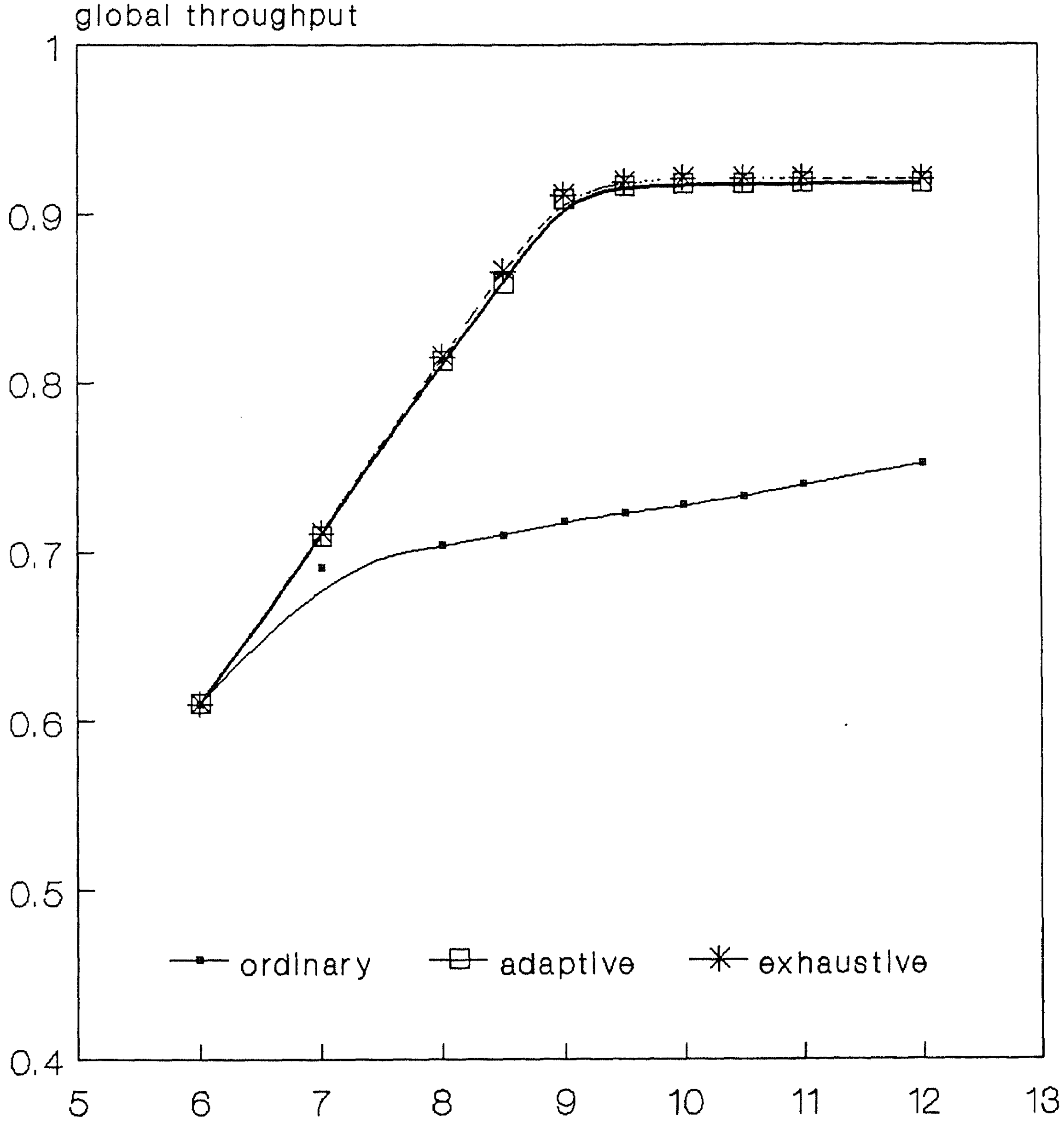


Figure 4.23

fixed message and packet length,
2000 bits/packet. a=0.561

throughput vs load
asymmetric nonburst 8 stations

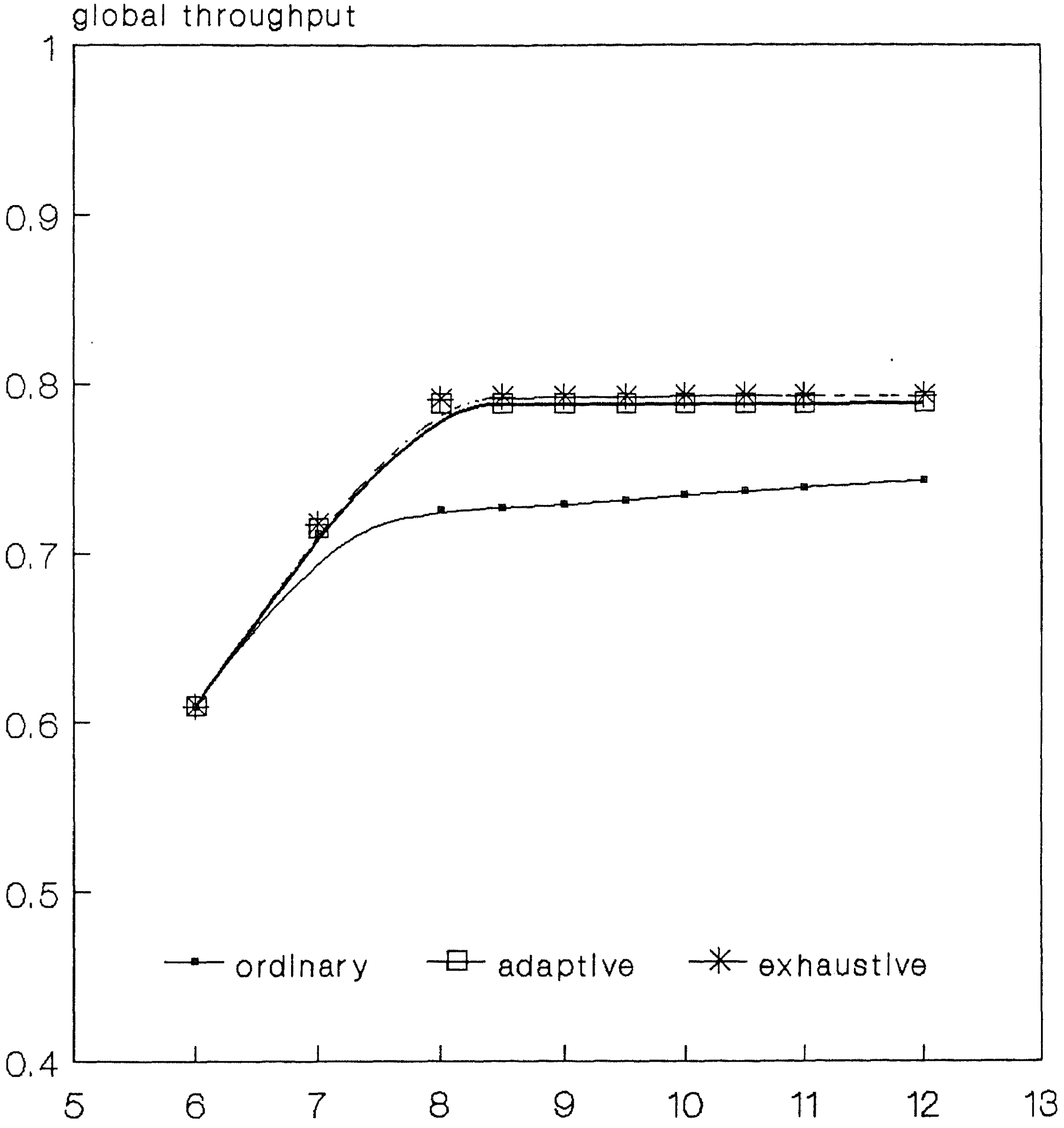


Figure 4.24

message length= 2000 ˜ 10000 bits
2000 bits/packet. a=0.561

throughput vs load
asymmetric burst 8 stations

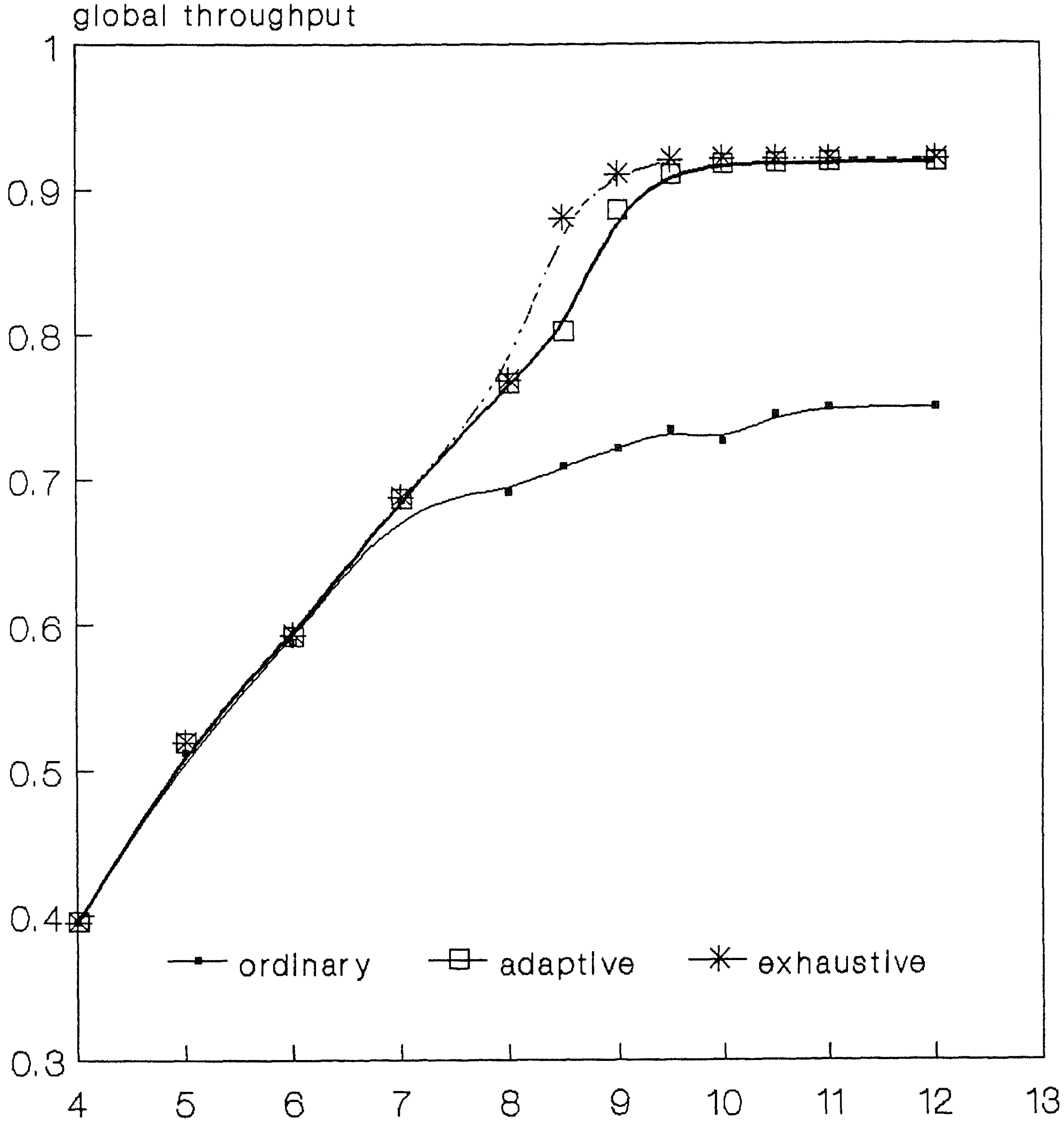


Figure 4.25

fixed message and packet length,
2000 bits/packet. a=0.561

throughput vs load
asymmetric burst 8 stations

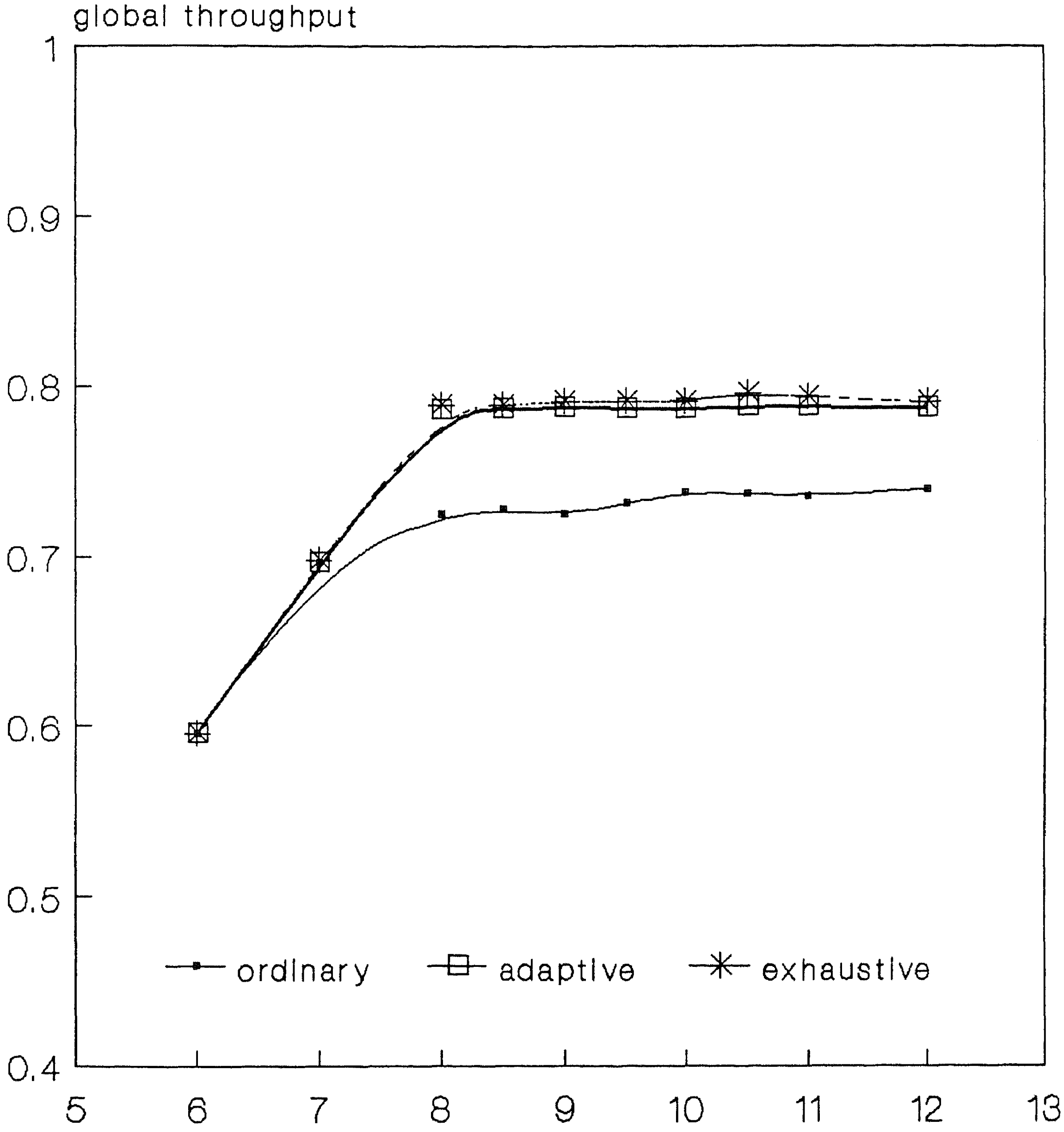


Figure 4.26

message length= 2000 ˜ 10000 bits
2000 bits/packet. a=0.561

delay vs k in adaptive service
asymmetric nonburst 8 stations
fixed message and packet length=2000bits

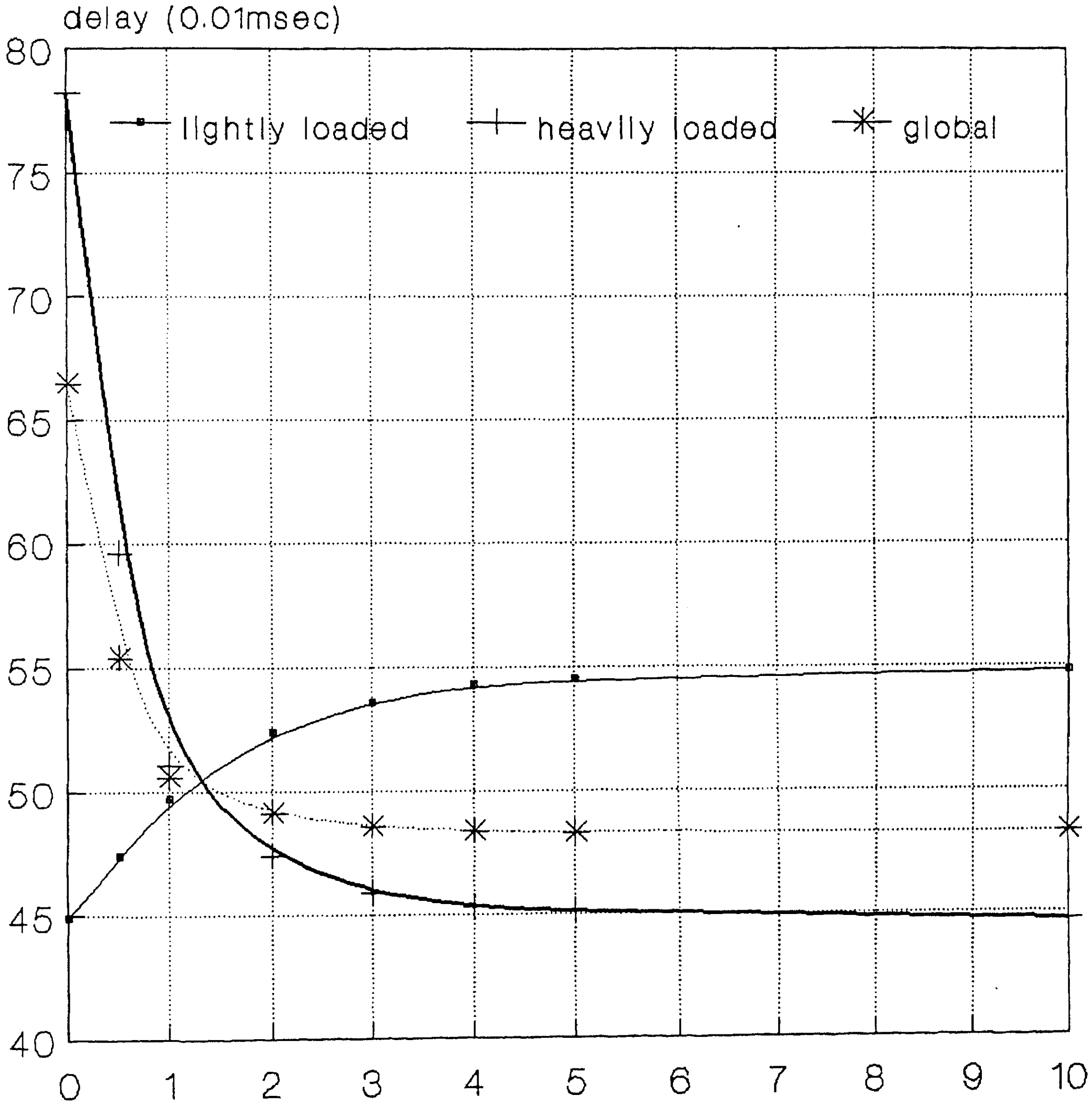


Figure 27.

a=0.561. 5 Mbps load.

delay vs k in adaptive service
asymmetric nonburst 8 stations
message length ▪ 2000 ˜ 10000 bits

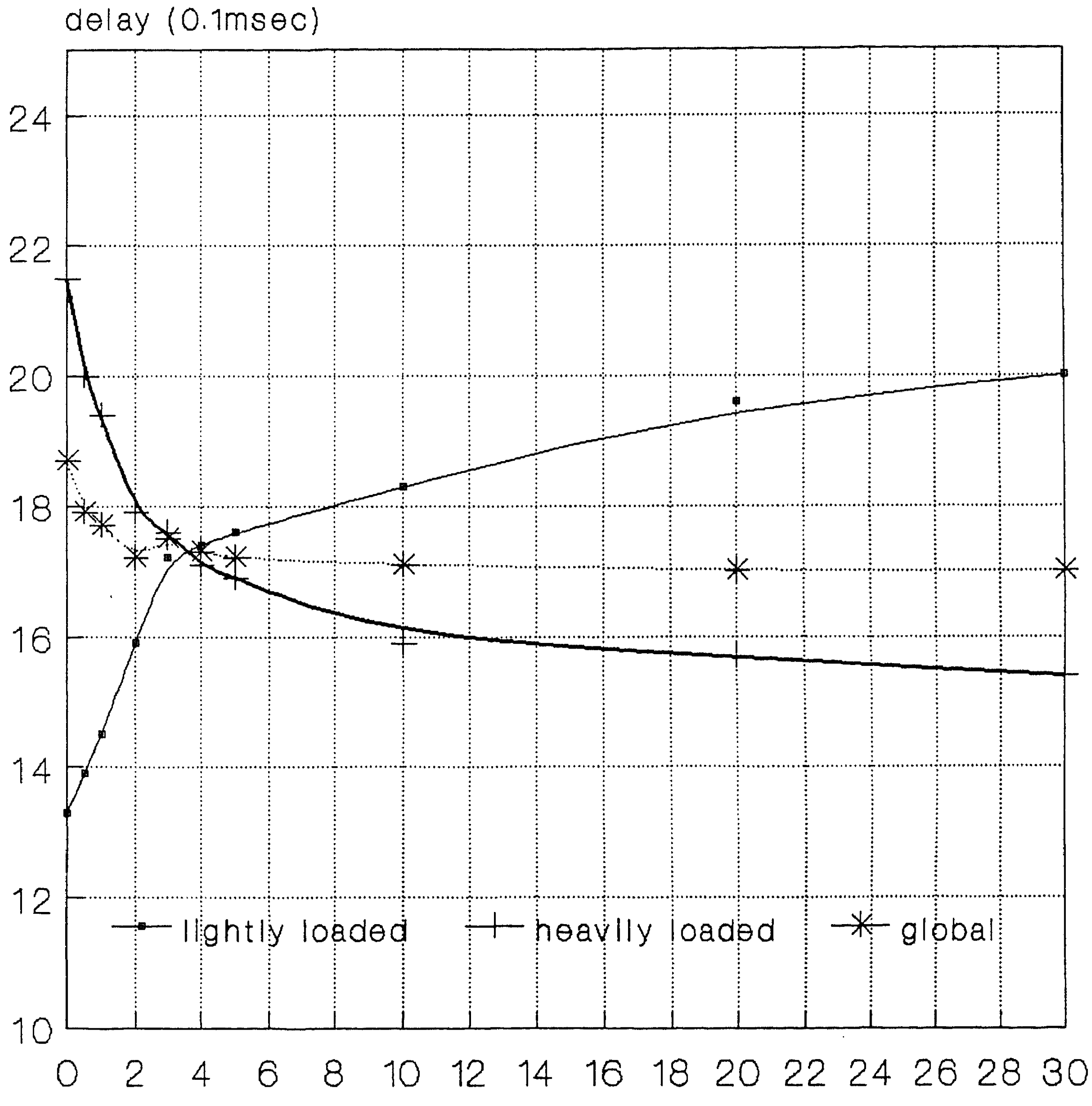


Figure 28.

a=0.561. 5Mbps load

delay vs k in adaptive service
asymmetric burst 8 stations
fixed message and packet length=2000bits

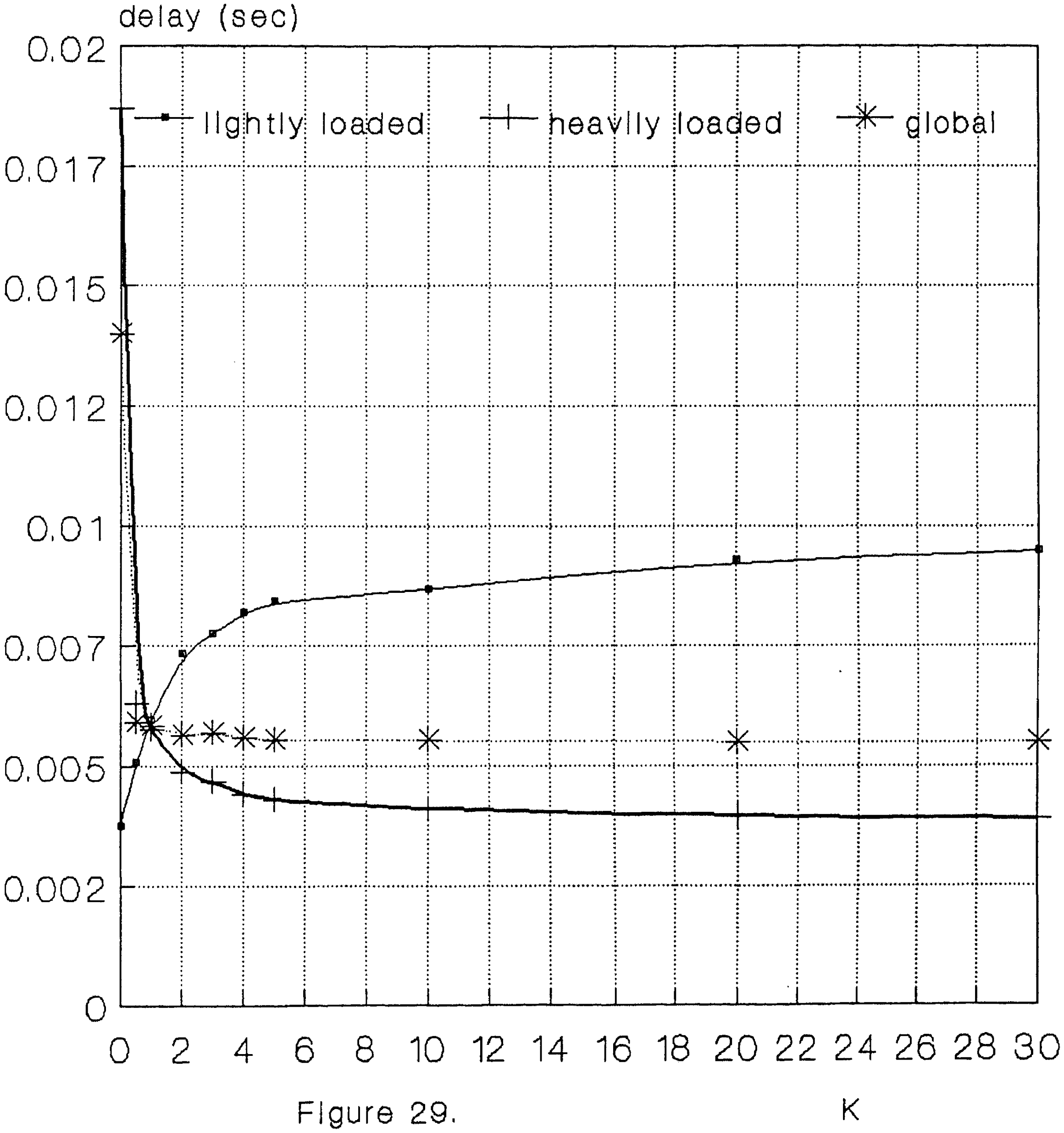


Figure 29.

a=0.561. 5 Mbps load

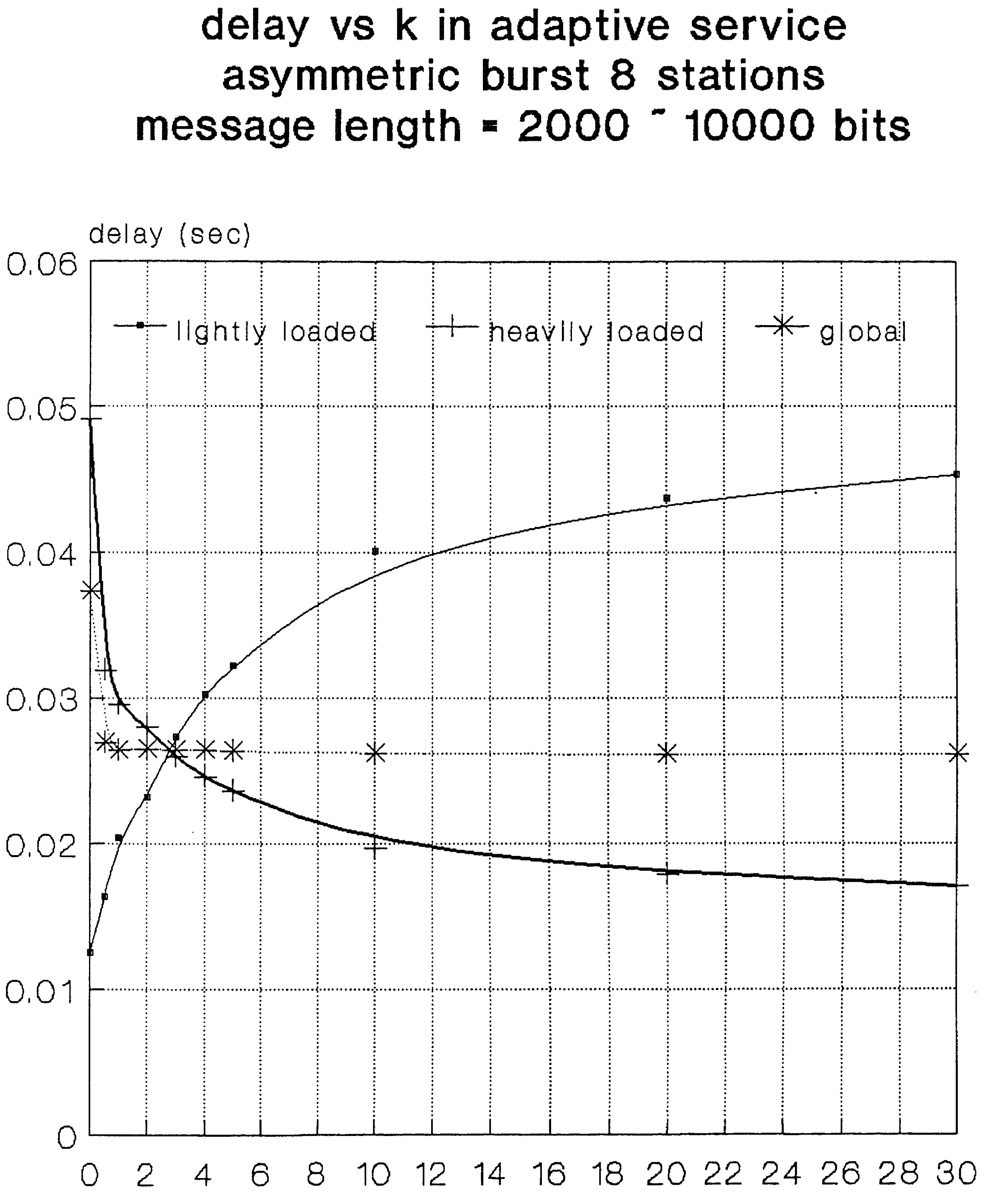


Figure 30.

a=0.561. 5 Mbps load.

delay vs k in adaptive service
asymmetric nonburst 8 stations
fixed message and packet length=2000bits

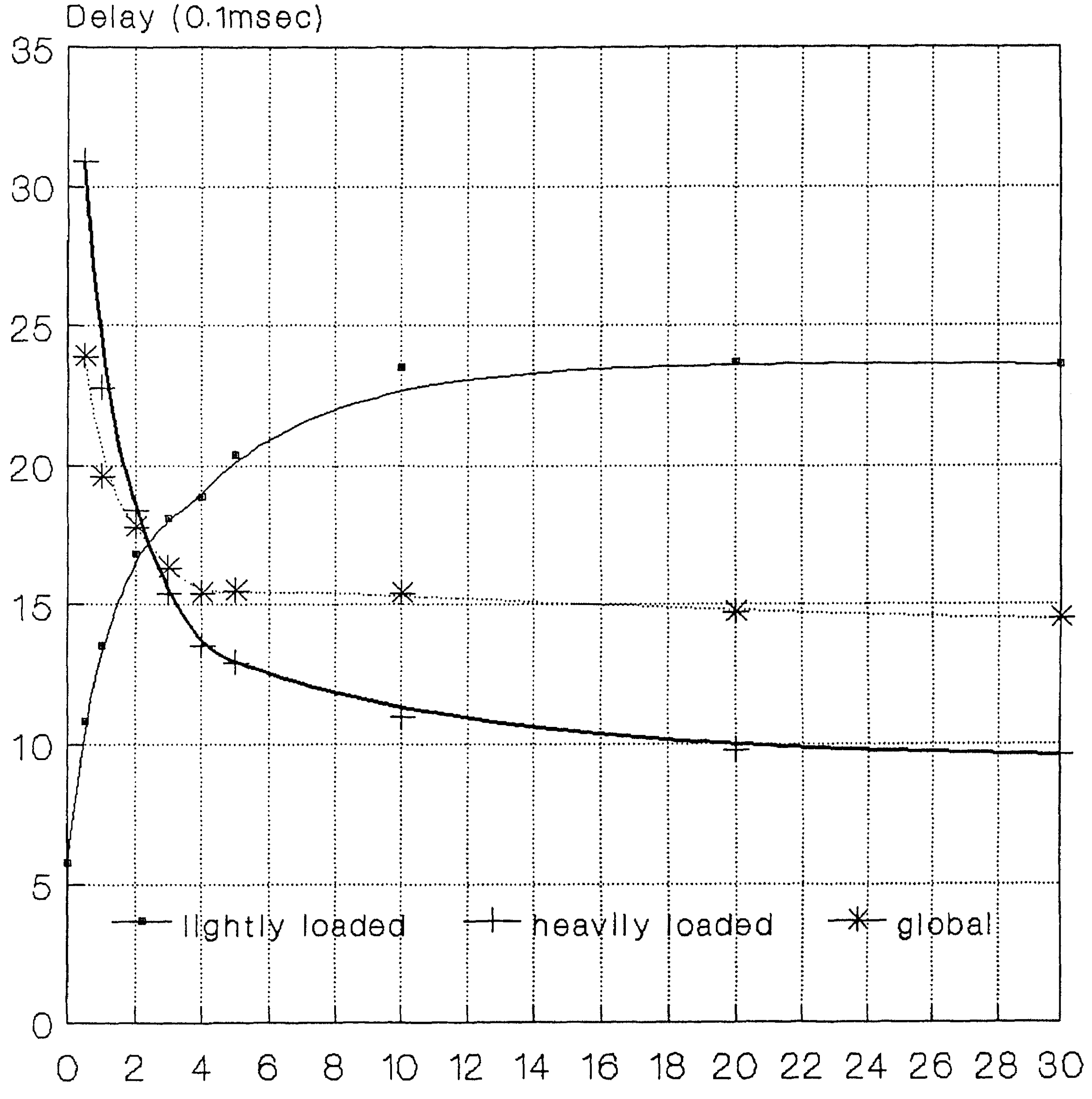


Figure 31

a=0.561. 8 Mbps load.

delay vs k in adaptive service
asymmetric nonburst 8 stations
message length ▪ 2000 ˜ 10000 bits

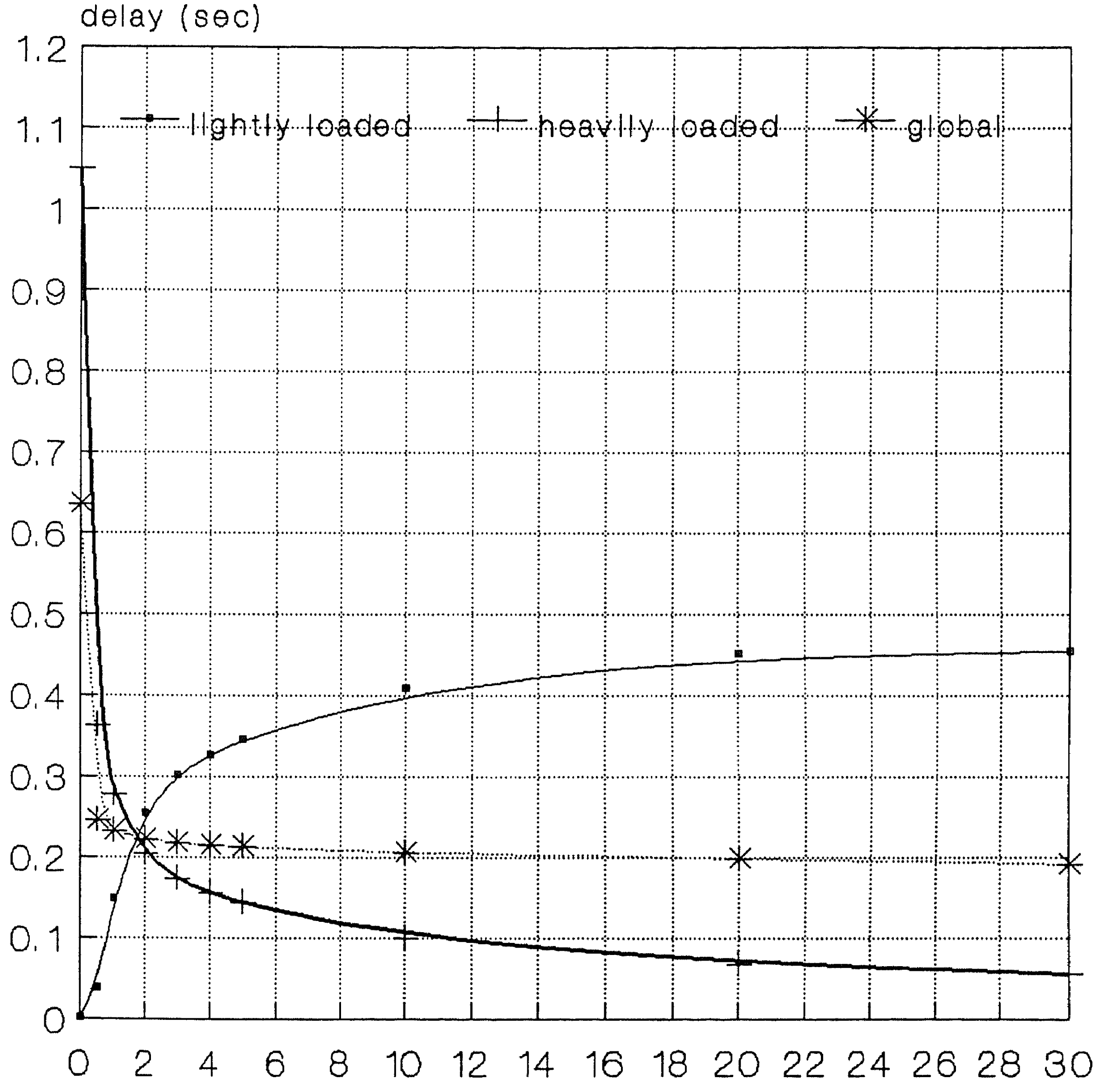


Figure 32.

a=0.561. 8Mbps load

delay vs k in adaptive service
asymmetric burst 8 stations
fixed message and packet length=2000bits

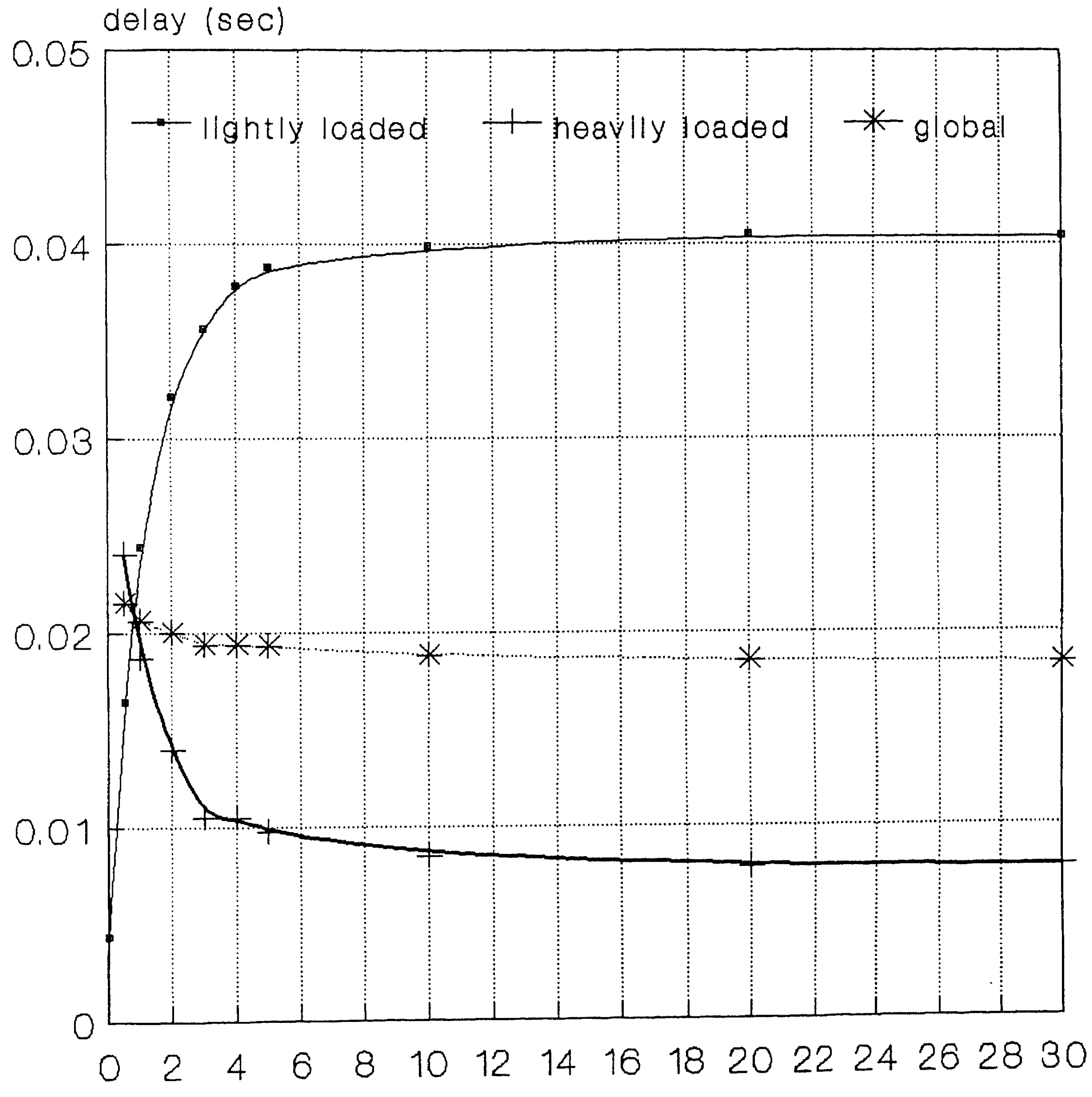


Figure 33.

a=0.561. 8 Mbps load

delay vs k in adaptive service asymmetric burst 8 stations message length = 2000 ~ 10000 bits

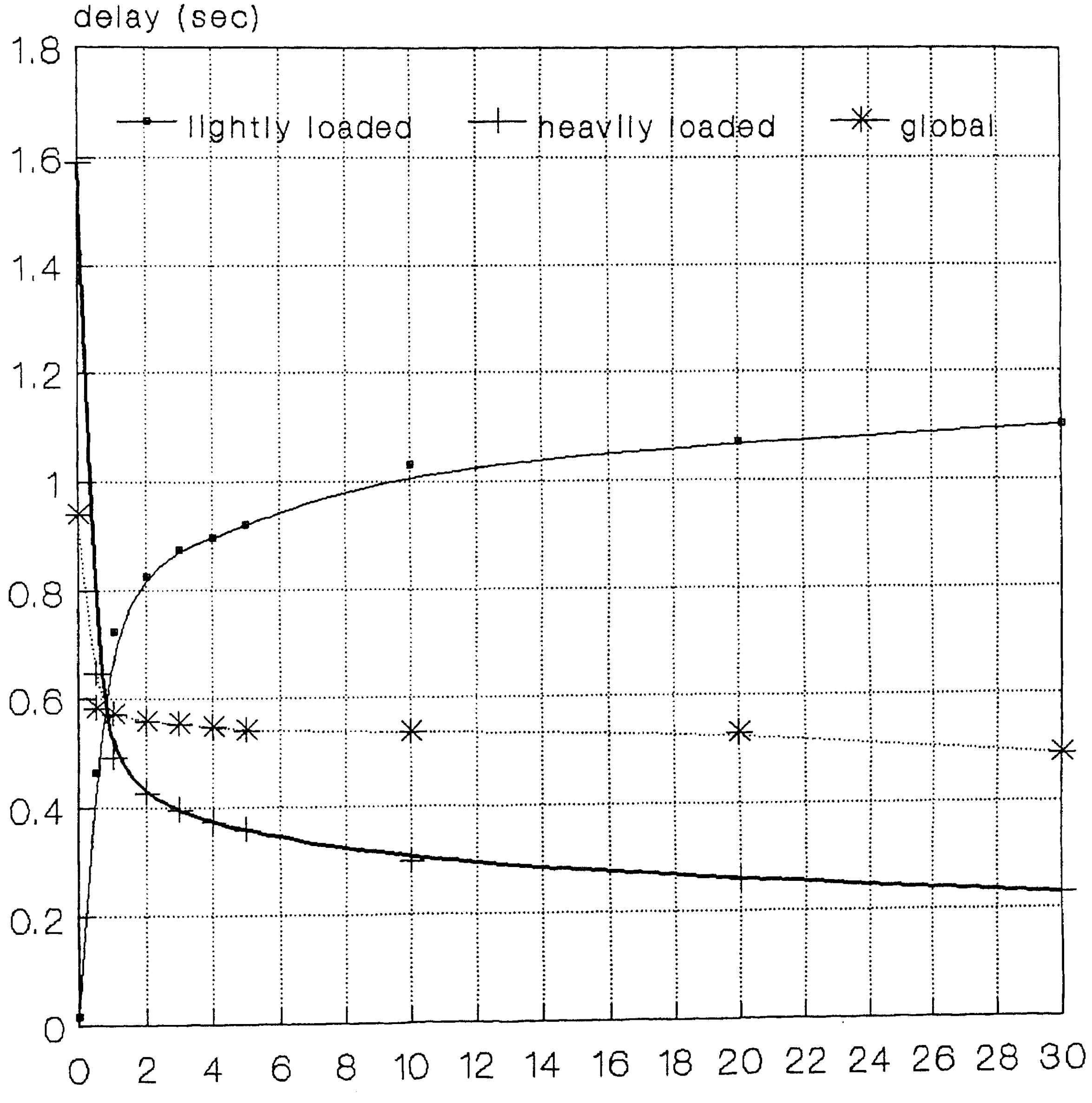


Figure 34.

a=0.561. 8 Mbps load.

Chapter 5

Conclusion

A study of different services in the token ring network is presented in this thesis. The main contribution of this thesis is that a new service, adaptive service, is proposed and its delay and throughput characteristics are compared with the ordinary and exhaustive service. By using timer and counter at every station, adaptive service keeps track of recent local and global activities on the network. It has been shown that in asymmetric traffic (**a**=0.561), adaptive service gives as much as 40% improvement at moderate load on the local delay compared with exhaustive service and gives as much as 71% improvement at moderate load on the global delay compared with ordinary service. Also, for symmetric traffic (**a**=0), the same adaptive service gives as much as 19% improvement at moderate load over ordinary service. And for all kinds of traffic, adaptive service has superior throughput performances compared with the ordinary service.

In general, it can be said that the adaptive service is a good compromise between ordinary service and exhaustive for all types of traffic patterns.

Appendix A

Input Model

Input model describes configuration and assumptions of the network model.

A.1 Input Model

```
1 ETU = 10,000,000 ITUs
Network Configuration:
Number of stations          8
Ports                       2/8  * each station has 2 ports
Number of links             8
* Links are unidirectionally  interconnecting the stations *
*Link 0
Archive time                120
Number of ports             2
Port assignment             0  1  1
                            1  0  1
Distance matrix (link length): 10
*Link 1
Archive time                120
```

```
Number of ports              2
Port assignment              1  1  1
                             2  0  1
Distance matrix (link length): 10
*Link 2
Archive time                 120
Number of ports              2
Port assignment              2  1  1
                             3  0  1
Distance matrix (link length): 10
*Link 3
Archive time                 120
Number of ports              2
Port assignment              3  1  1
                             4  0  1
Distance matrix (link length): 10
*Link 4
Archive time                 120
Number of ports              2
Port assignment              4  1  1
                             5  0  1
Distance matrix (link length): 10
*Link 5
Archive time                 120
Number of ports              2
Port assignment              5  1  1
                             6  0  1
```

```
Distance matrix (link length): 10

*Link 6

Archive time                120

Number of ports             2

Port assignment             6  1  1

                            7  0  1

Distance matrix (link length): 10

*Link 7

Archive time                120

Number of ports             2

Port assignment             7  1  1

                            8  0  1

Distance matrix (link length): 10

*Link 8

Archive time                120

Number of ports             2

Port assignment             8  1  1

                            9  0  1

Distance matrix (link length): 10

*Link 9

Archive time                120

Number of ports             2

Port assignment             9   1  1

                            10  0  1

Distance matrix (link length): 10

*Link 10

Archive time                120
```

```
Number of ports             2
Port assignment             10  1  1
                            11  0  1
Distance matrix (link length): 10
*Link 11
Archive time                120
Number of ports             2
Port assignment             11  1  1
                            0   0  1
Distance matrix (link length): 10
```

Symmetric nonburst traffic:

```
Number of message type                 1
** Message type 0 **
Number of message type                 1
     options                           =1
     Exponential interarrival time, uniformly distributed
     length
     Mean Inter-arrival time           =0.0004
     Minimum length                    = 2000
     Maximum length                    = 2000 (message length
     is fixed)
     Number of selection group         =1
     Number of flood group             =0
** Selection group 0 **
Number of senders 8,        stations (0,1) (1,1) (2,1) (3,1)
                                     (4,1) (5,1) (6,1) (7,1)
Number of receivers 8,      stations (0,1) (1,1) (2,1) (3,1)
```

(4,1) (5,1) (6,1) (7,1)

Protocol specific parameters:

Minimum packet length	=2000
Maximum packet length	=2000
Header	=128
Trailer	=32
Token length	=24
Token passing timeout	=2000000

Exit condition:

Maximum number of message	=10,000
Virtual time limit	=0
CPU time limit	=0

Different types of traffic has different traffic sections.

Symmetric burst traffic:

Number of message type	=1

** Message type 0 **

options	=1+4+8

Bursty traffic with:

- exponential burst interarrival time
- uniformly distributed burst size
- exponential message interarrival time within a burst
- uniformly distributed message length

Mean message interarrival time	=0.000001
Minimum message length	=2000
Maximum message length	=10,000 (varying

```
     message length)
     Mean burst interarrival          =0.008
     Minimum burst size              =20
     Maximum burst size              =20
     Number of selection group       =1
     Number of flood group           =0
** Selection group 0 **
Number of senders 8,         stations (0,1) (1,1) (2,1) (3,1)
                                      (4,1) (5,1) (6,1) (7,1)
Number of receivers 8,       stations (0,1) (1,1) (2,1) (3,1)
                                      (4,1) (5,1) (6,1) (7,1)
```

Asymmetric nonburst traffic:

```
Number of message type               2
** Message type 0 **
     Options                         =1
     Exponential interarrival time, uniformly distributed
     length
     Mean interarrival time          =0.001143
     Minimum length                  =2000
     Maximum length                  =2000 (message length
     is fixed)
     Number of selection group       =1
     Number of flood group           =0
** Selection group 0 **
Number of senders 1,         stations (0,1)
Number of receivers 8,       stations (0,1) (1,1) (2,1) (3,1)
                                      (4,1) (5,1) (6,1) (7,1)
```

```
** Message type 1 **
     Options                          =1
     Message interarrival time        =0.0006154
     Minimum length                   =2000
     Maximum length                   =2000 ( message length
     is fixed)
     Number of selection group        =1
     Number of flood group            =0
** Selection group 0 **
Number of senders 7,        stations       (1,1) (2,1) (3,1)
                                     (4,1) (5,1) (6,1) (7,1)
Number of receivers 8,      stations (0,1) (1,1) (2,1) (3,1)
                                     (4,1) (5,1) (6,1) (7,1)
```

Asymmetric burst traffic

```
Number of message type                2
** Message type 0 **
     Options                          =1
     Exponential interarrival time, uniformly distributed
     length
     Mean interarrival time           =0.06857
     Minimum length                   =2000
     Maximum length                   =10,000 ( varying
     message length)
     Number of selection group        =1
     Number of flood group            =0
** Selection group 0 **
```

```
Number of senders 1,       stations (0,1)

Number of receivers 8,     stations (0,1) (1,1) (2,1) (3,1)

                                    (4,1) (5,1) (6,1) (7,1)

** Message type 1 **

     Options                         =1

     Message interarrival time       =0.03692

     Minimum length                  =2000

     Maximum length                  =10,000 ( varying

     message length)

     is fixed)

     Number of selection group       =1

     Number of flood group           =0

** Selection group 0 **

Number of senders 7,       stations       (1,1) (2,1) (3,1)

                                    (4,1) (5,1) (6,1) (7,1)

Number of receivers 8,     stations (0,1) (1,1) (2,1) (3,1)

                                    (4,1) (5,1) (6,1) (7,1)
```

Bibliogrphy

[1] A.S.Sethi and T.Saydam. "Performance analysis of token ring local area networks," Computer Networks and ISDN Systems, vol. 9, pp. 191-200, 1985.

[2] G. Kimura and Y. Takahashi, "Diffusion approximation for a token ring system with nonexhaustive service," IEEE Jour. on Sel. Areas in Comm., vol. SAC-4, no. 6, pp. 794-802, Sept. 1986.

[3] M.J. Ferguson, "Computation of the variance of the waiting time for token rings," IEEE Jour. on Sel. Areas in Comm., vol. SAC-4, no. 6, pp. 775-782, Sept. 1986.

[4] D.Everitt, "Simple approximation for token rings," IEEE Trans. on Comm., vol. COM-34, no. 7, pp. 719-721, July. 1986.

[5] Ferguson, Michael J.;Aminetah, Yehuda J., "Exact results for nonsymmetric token ring systems," IEEE Tran. on Comm., vol. COM-33, pp. 223-231, March. 1985.

[6] Andrew S. Tanenbaum, "Computer Networks," Prentice Hall, Inc., Englewood cliffs, New Jersey,1988.

[7] Gburzynski, Pawel and Rudinki, Piotr, "The LANSF protocol modeling environment," Dept. of Computing Science, Uni. of Alberta, Alberta, Canada.