NASA Contractor Report 182055

ICASE INTERIM REPORT 10

COMMUNICATION OVERHEAD ON THE INTEL iPSC-860 HYPERCUBE

Shahid Bokhari

NASA Contract No. NAS1-18605
May 1990

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665-5225

ICASE INTERIM REPORTS

ICASE has introduced a new report series to be called ICASE Interim Reports. The series will complement the more familiar blue ICASE reports that have been distributed for many years. The blue reports are intended as preprints of research that has been submitted for publication in either refereed journals or conference proceedings. In general, the green Interim Report will not be submitted for publication, at least not in its printed form. It will be used for research that has reached a certain level of maturity but needs additional refinement, for technical reviews or position statements, for bibliographies, and for computer software. The Interim Reports will receive the same distribution as the ICASE Reports. They will be available upon request in the future, and they may be referenced in other publications.

Robert G. Voigt
Director

Communication Overhead on the Intel iPSC-860 Hypercube*

Shahid H. Bokhari

Department of Electrical Engineering

University of Engineering & Technology, Lahore, Pakistan

and

ICASE, NASA Langley Research Center

Hampton, Virginia

Abstract

Experiments have been conducted on the Intel iPSC-860 hypercube in order to evaluate the overhead of interprocessor communication. It is demonstrated that (1) contrary to popular belief, the distance between two communicating processors has a significant impact on communication time, (2) edge contention can increase communication time by a factor of more than 7, and (3) node contention has no measurable impact.

*Research supported by the National Aeronautics and Space Administration under NASA contract NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science & Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23665-5225.

1 Introduction

A key measure of the power of a multiple computer system is the interprocessor communication overhead. Despite major improvements in design and technology, the time to communicate a datum from one processor to another remains one or more orders of magnitude greater than the time to access it on the same processor. This time is influenced by the type of interconnection network used as well as the strategy used for routing messages.

This report presents the results of experiments carried out on the recently introduced Intel iPSC-860 hypercube. This machine is based on the powerful i860 microprocessor and uses circuit-switched communications through a hypercube interconnection. The experiments described below were developed on the 32 node iPSC-860 at ICASE and carried out on the 128 node iPSC-860 at NASA Ames Research Center.

The results of these experiments permit the following major conclusions to be made about this machine.

1. Contrary to popular belief, the time required to communicate between two nodes *does* depend on the number of intervening hops on both machines. Although this variation can be neglected for very large messages, it cannot be ignored for short messages.

2. Edge contention (the sharing of a communication link by two or more paths) leads to severe overhead for all message sizes. This can increase the time to communicate by a factor of more than seven.

3. Node contention (the sharing of a node by two or more paths) has no measurable impact.

2 The Intel iPSC-860 hypercube

The interconnection network of a 32 node hypercube is shown in Figure 1. The labeled vertices hanging from each vertex of the network represent processors of the hypercube. Two vertices in the network are connected if and only if the binary representations of their labels differ in exactly one bit. An important feature of interprocessor communications in the Intel hypercube is circuit switching. When two nodes wish to communicate, a dedicated

path is set up between them. Messages then flow through this path without involving intervening processors. The path between source and destination is determined by the 'e-cube' routing algorithm: starting with the right hand side of the binary label of the source processor, we move to the processor whose label most closely matches the label of the destination processor.

Since the routing algorithm is fixed, we can encounter *edge* and *node* contention. Edge contention is the sharing of an edge (i.e. a communication link) by two or more paths. Similarly, node contention is the sharing of a node.

Figure 1 illustrates paths from 0 to 31 (solid), 2 to 23 (dashed) and 14 to 11 (dotted). The *lengths* of these paths (the *distance* between source and destination) are 5, 3 and 2 respectively. The paths $0 \rightarrow 31$ and $2 \rightarrow 23$ share the edge 3–7, while the paths $0 \rightarrow 31$ and $14 \rightarrow 11$ share node 15.

3 Overview of Experiments

In the following sections we will describe results of experiments that evaluate the impact of path length, edge contention and node contention on communication time. Experimental data are presented in plots. Each plot is accompanied by a brief explanation of what it depicts and a discussion on the possible causes and consequences of any unusual phenomena.

Every point in every plot is an average of at least 100 runs for small (< 1000 byte) messages. This was done to improve the 1 msec. resolution of the clock supplied with the machine. For larger messages, proportionately fewer runs were made in order to save time.

4 Impact of Path length

The basic technique to measure the communication time between processors src and dest is to start the clock on src and then invoke the csendrecv (...,dest,...) routine to send out a message and wait for a reply. On dest a pair of consecutive crecv(...) and csend(...,src,...) is used to echo this message back. This is repeated n times for the reason stated above and the clock is then stopped. The time for a unidirectional transmission is computed to be $clock/(2 \times n)$.

4.1 Message size 0–200 bytes

Figure 2 shows the time required to communicate messages of length 0–200 bytes between processors that are $1, 2, \cdots, 7$ communication links apart. The time required to send a 0 byte message to a neighboring node (i.e. distance 1 away) is about 67 μsec. (this represents the absolute minimum for any communication operation on this machine). The time to communicate a 0 byte message over the maximum distance of 7 is 131 μsec. Inspection of these plots reveals that they are linear, parallel and evenly distributed from 0 to 100 bytes. The communication time increases at about 10 μsec. per communication link. This is a far from negligible variation: the time required to send a 4-byte floating point number distance 7 away is nearly double the time to send it to a neighboring node.

There is a sharp jump in the curves at message length 100 bytes. This is caused by a change in the communication protocol. Messages of length ≤ 100 bytes are sent without checking to see if there is buffer space in the destination processor to store the incoming message. For messages of length > 100 bytes a check is first made. This requires an additional round-trip message and accounts for the step at 101 bytes.

Beyond 100 bytes, the plots are again linear, parallel and evenly distributed. The spacing changes to about 30 μsec. per link and remains constant (see below). The time to communicate a 101 byte message over distance 7 is again about double the time for distance 1.

The time (in μsec.) to communicate a message of length m bytes over distance d is $t = 65 + 0.425m + 10.0d$, for $0 < m \leq 100$ and $t = 147 + 0.390m + 30.5d$ for $m > 100$. Note that the time for zero byte messages is slightly below what would be predicted by these expressions.

4.2 Message size 200–1000 bytes

The plot of Figure 2 is extended beyond 200 bytes in Figure 3. It can be seen that the times are linear, uniform and parallel. Since the distance effects remain constant, they account for a smaller percentage of total time as the message size increases. Nevertheless, the impact of distance can be as much as 20% for 1000 bytes.

4.3 Message size 1000–10000 bytes

The impact of distance becomes smaller as the message size increases to 10000 bytes. Figure 4 shows that at 10000 bytes the impact of distance is still a clearly measurable 5%. The plots remain parallel and evenly spaced, but there are now well-defined plateaus at about 2000 byte intervals. On the iPSC-860, communication between the i860 processors and the communication network is through 4000 byte FIFOs* which interrupt when they are half full or half empty. These FIFOs are responsible for the plateaus.

5 First edge contention experiment

In this experiment we established communications between nodes $0 \rightarrow 127$, nodes $6 \rightarrow 111$, nodes $4 \rightarrow 79$, and nodes $7 \rightarrow 15$. The routings between these two pairs of nodes are as follows.

$$0 \rightarrow 1 \rightarrow 3 \rightarrow 7 \rightarrow 15 \rightarrow 31 \rightarrow 63 \rightarrow 127 \quad (1)$$

$$6 \rightarrow 7 \rightarrow 15 \rightarrow 47 \rightarrow 111 \quad (2)$$

$$4 \rightarrow 5 \rightarrow 7 \rightarrow 15 \rightarrow 79 \quad (3)$$

$$7 \rightarrow 15 \quad (4)$$

This pattern is illustrated in Figure 5. It can be seen that all messages use link $7 \rightarrow 15$. This is the only edge that has multiple paths through it and no node has multiple paths through it.

To compare communication times with and without edge contention, we ran precisely the same procedures that were used in Section 4. That is, we sent messages from each source to each destination and echoed them back. Because of this, the results of this experiment have to be viewed cautiously. Firstly, since our experiments use a send-receive approach, it is important also to verify that the return paths (i.e. $127 \rightarrow 0$,$111 \rightarrow 6$, $79 \rightarrow 4$ and $15 \rightarrow 7$) are mutually edge and node disjoint and thus do not disturb the experiment. This is indeed the case. Secondly, the times that we measure are those for round trip communication between nodes 0 and 127. These messages encounter contention in the forward direction and no contention in the return direction. When these timings are divided by 2, this gives

*David Scott, personal communication.

us the average time for transmission, given that contention occurs only in the forward direction. The unidirectional transmission time, *with* contention would be twice the times shown below, minus the time for unidirectional transmission without contention. The plots given in this section are thus lower bounds on the contention overhead. In Sections 6 and 7 we describe experiments that measure the unidirectional transmission time directly.

5.1 First experiment: line plot

The results of the first contention experiment are shown in Figure 6 for messages of length 0 to 7000 bytes, in increments of 1 byte. The plot labeled 1 shows the times for path (1) alone, the plot labeled 2 shows the times for paths (1) and (2) simultaneously, and so on. Plot 1 corresponds to the path $0 \rightarrow 127$ alone and is thus equivalent to the uppermost plot in Figure 4. The plot labeled 2 shows the very significant impact of two paths sharing an edge. This is about 33% for 7000 byte messages.

Plots 3 and 4 of Figure 6 illustrate an interesting aspect of the iPSC-860 communication network that we have discovered. These plots are jagged and chaotic, but fall within very clearly defined envelopes. Over some ranges of message length, the time can vary from almost no overhead to maximum overhead within one byte (this accounts for the dark bands). The gross patterns of light and dark regions of these plots are reproducible form one run to another, although the precise peaks do not always match. It is important to note that despite the jaggedness, the times for 3 and 4 paths are always slightly great than the upper envelopes of the 2 and 3 path times, respectively. Thus the time for 6000–7000 byte messages over 4 paths is never less than $1.8\times$ the time for 2 paths.

5.2 First experiment: scatter plot

The values of Figure 6 are replotted in Figure 7 as a scatter plot. This brings out the distribution of points more clearly. It is evident that there is a cycle of period 2000 bytes in this plot. We conjecture that the waveform is caused by a complex interaction of the FIFO mentioned above and the messages that circulate between each source-destination pair.

Since we do not employ any form of synchronization in this experiment, there are slight differences in the relative times messages are launched from

sources, caused by the increasing lengths of messages. As a result there are some lengths for which messages are launched with so much relative delay that they miss each other altogether. This accounts for the jaggedness in these plots and cautions us that when looking for contention effects within a larger parallel application on the iPSC-860, we may sometimes not be able to detect any, because of fortuitous timing. Contention effects may then suddenly appear in an application when a change in the code causes just enough change in timings to make messages contend.

5.3 First experiment: detailed plot (0–200 bytes)

Degradation in performance due to edge contention can occur over all message sizes. Figure 8 magnifies the plot of Figure 6 over the range 0–200 bytes and shows some degradation at 80 bytes. We will demonstrate much greater degradation for small messages in Section 6.2.

5.4 First experiment: detailed plot (2000–2500 bytes)

Figure 9 affords a detailed view of the chaotic part of the envelope and also shows a small pulse at 2210 to 2221 bytes. This phenomenon does not occur at any other point on the curves and is probably due to the FIFO.

6 Second edge contention experiment

The second contention experiment uses the following eight paths which are depicted in Figure 10.

$$0 \to 1 \to 3 \to 7 \to 15 \to 31 \to 63 \to 127 \quad (1)$$
$$1 \to 3 \to 7 \to 15 \to 31 \to 63 \quad (2)$$
$$3 \to 7 \to 15 \to 31 \quad (3)$$
$$7 \to 15 \quad (4)$$
$$5 \to 7 \to 15 \to 79 \quad (5)$$
$$6 \to 7 \to 15 \to 47 \quad (6)$$
$$2 \to 3 \to 7 \to 15 \to 31 \to 95 \quad (7)$$
$$4 \to 5 \to 7 \to 15 \to 47 \to 111 \quad (8)$$

The second experiment has been designed to impose the maximum possible amount of contention on one edge. Thus we have 8 paths sharing edge $7 \rightarrow 15$. In this experiment we have other edges with smaller amounts of contention. This is in contrast with the first experiment, in which only one edge had contention, and all others were contention-free. This experiment was again run for message lengths of 0 to 7000 bytes but, in order to save time, we incremented by 25 bytes over most of this range. The communication procedures were changed so that the echoed message at the destination is of zero byte length. The time taken by this return message (132 μsec.) is neglected for large messages but indicated on the plots for short messages[†].

6.1 Second experiment: line plot

Figure 11 shows a set of line plots from the second experiment. The plot labeled 1 shows the time for path (1) ($0 \rightarrow 127$) alone, plot 2 shows the time for paths (1) and (2) simultaneously and so on. This plot shows the same overall features as Figure 6[‡] but the overhead due to congestion is more severe. The time required by eight contending messages of 7000 bytes is more than seven times the time required for one message.

6.2 Second experiment: detailed plot

Figure 12 magnifies Figure 11 over 0 to 200 bytes. The increments are of 1 byte, so that this figure can be compared directly with Figure 8. The horizontal line at the bottom of this plot is at 132 μsec. and depicts the time of the zero byte echo which, as stated above is included in the total time. It is evident from this figure that edge contention can create significant overhead for as few as 4 paths, even for zero byte messages. The time for zero byte messages can be more than quadrupled when 8 paths contend for an edge.

[†]It can be verified that the return paths are node and edge disjoint, so that our experiment is not disturbed by contention among the return messages.

[‡]We must bear in mind that the resolution of Figure 11 is far less than Figure 6, since we have incremented message sizes by 25 bytes instead of 1 byte.

7 Third contention experiment

In this experiment we employed the same communication pattern (Figure 10) as for experiment 2 of Section 6, but used the gsync() routine to fully synchronize all transfers. The gsync() routine is expensive—it takes about 1 msec.—and thus causes a loss of resolution at small message lengths. Nevertheless this experiment serves to validate the previous experiment: the times for long messages for both are in agreement. Figure 13 shows the plots for this experiment. The horizontal line at the bottom of the plot is the gsync() time.

8 Impact of nodal congestion

Figure 14 plot shows the timings obtained when four pairs of node communicate with each other such that one node has 8 messages passing through it. We have chosen the paths $7 \rightarrow 31$, $13 \rightarrow 47$, $14 \rightarrow 79$ and $9 \rightarrow 15$. These messages are routed by the communications hardware according to the 'e-cube' routing algorithm such that all pass through node 15. Each source-destination pair uses echoing, so that there are a total of 8 messages passing through node 15. No edge has more than one message passing through it in one direction. The lines in this plot are the times when 1, 2, 3 and 4 of the above mentioned paths are established simultaneously. All four lines are indistinguishable: there is no impact of nodal congestion on communication time.

9 Conclusions

The Intel iPSC-2[§] and IPSC-860 are among the first commercial examples of circuit switched machines. Since circuit switching provides very fast communications, it is generally felt that it eliminates most, if not all, of the inefficiencies caused by communication overhead. In particular, it is a common belief that in circuit switched machines the precise placement of subcomputations on processors is irrelevant, since communication overhead is negligible. Our measurements indicate that this is a mistaken belief since

[§]Some measurements of communication overhead on the iPSC-2 are described in [1].

the communication overhead is (1) not negligible and (2) extremely sensitive to placement.

We conclude that communication overhead on the iPSC-860 will limit performance on communication-bound algorithms. This overhead is caused both by distance effects (significant for small message sizes) and by contention (significant for all message sizes). We have also shown that in a specific application, contention effects may sometimes not be manifest because of fortuitous timing. They may suddenly appear, in great severity, when a small change in the code changes the timings.

It appears that many circuit switched or 'wormhole' routed machines will be built in the near future. It will be interesting to see the effects of edge contention on these machines. For the Symult-2010 mesh machine, results described in [2] indicate that edge contention can have similarly serious consequences.

Acknowledgements

I wish to thank Tom Crockett at ICASE for getting me started on using hypercubes. I am indebted to David Bailey for arranging access to the NASA-Ames iPSC-860 and to Leigh Ann Tanner, also at Ames, for her help in using the machine. Numerous discussions with my colleagues at ICASE have been very useful.

References

[1] Luc Bomans and Dirk Roose. Benchmarking the iPSC/2 hypercube. *Concurrency: Practice and Experience*, 1:3–18, 1989.

[2] S. Chittor and R. Enbody. Performance analysis of Symult 2010's interprocessor communication network. Technical Report CPS-89-19, Michigan State University, Department of Computer Science, 1989.

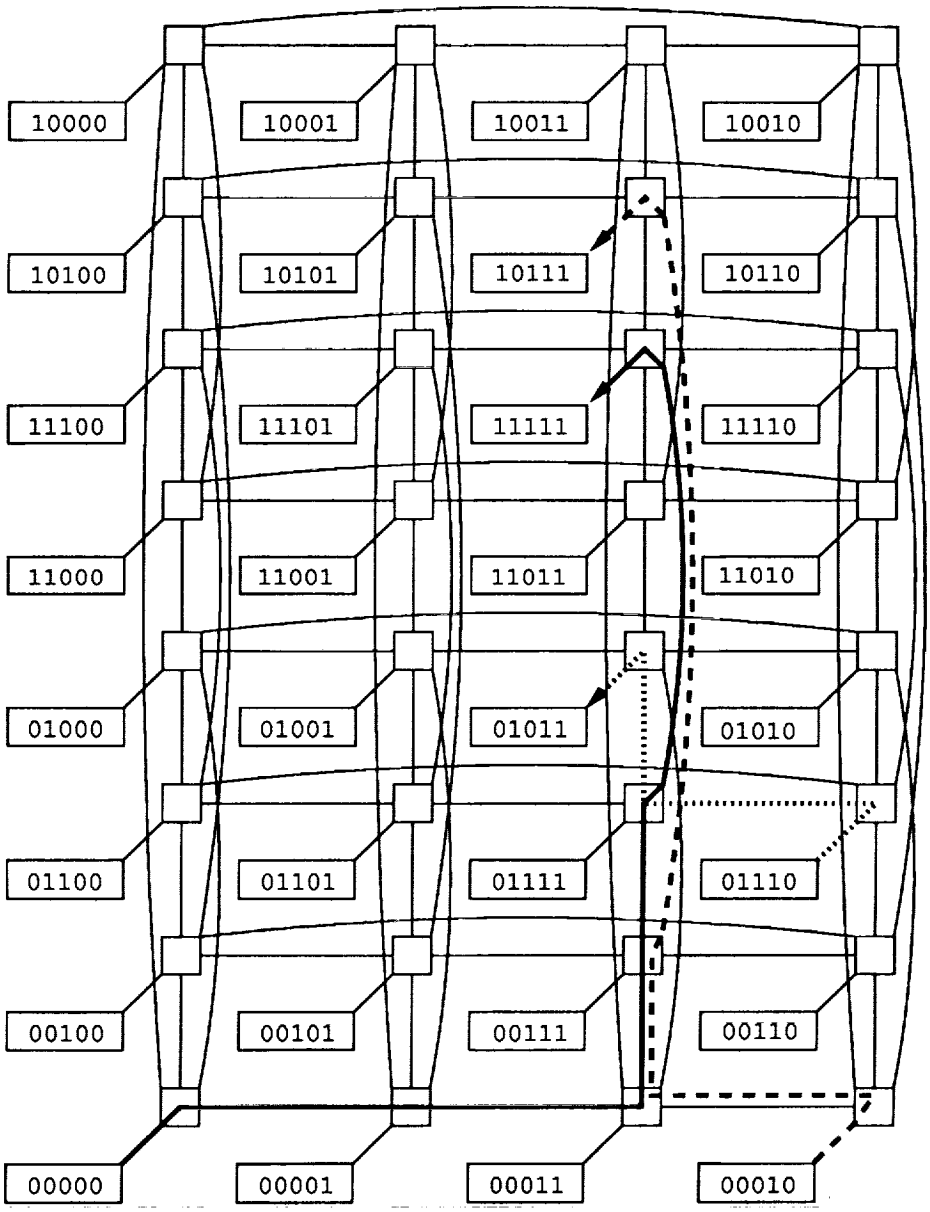

Figure 1: Interconnection network of a 32 node hypercube.

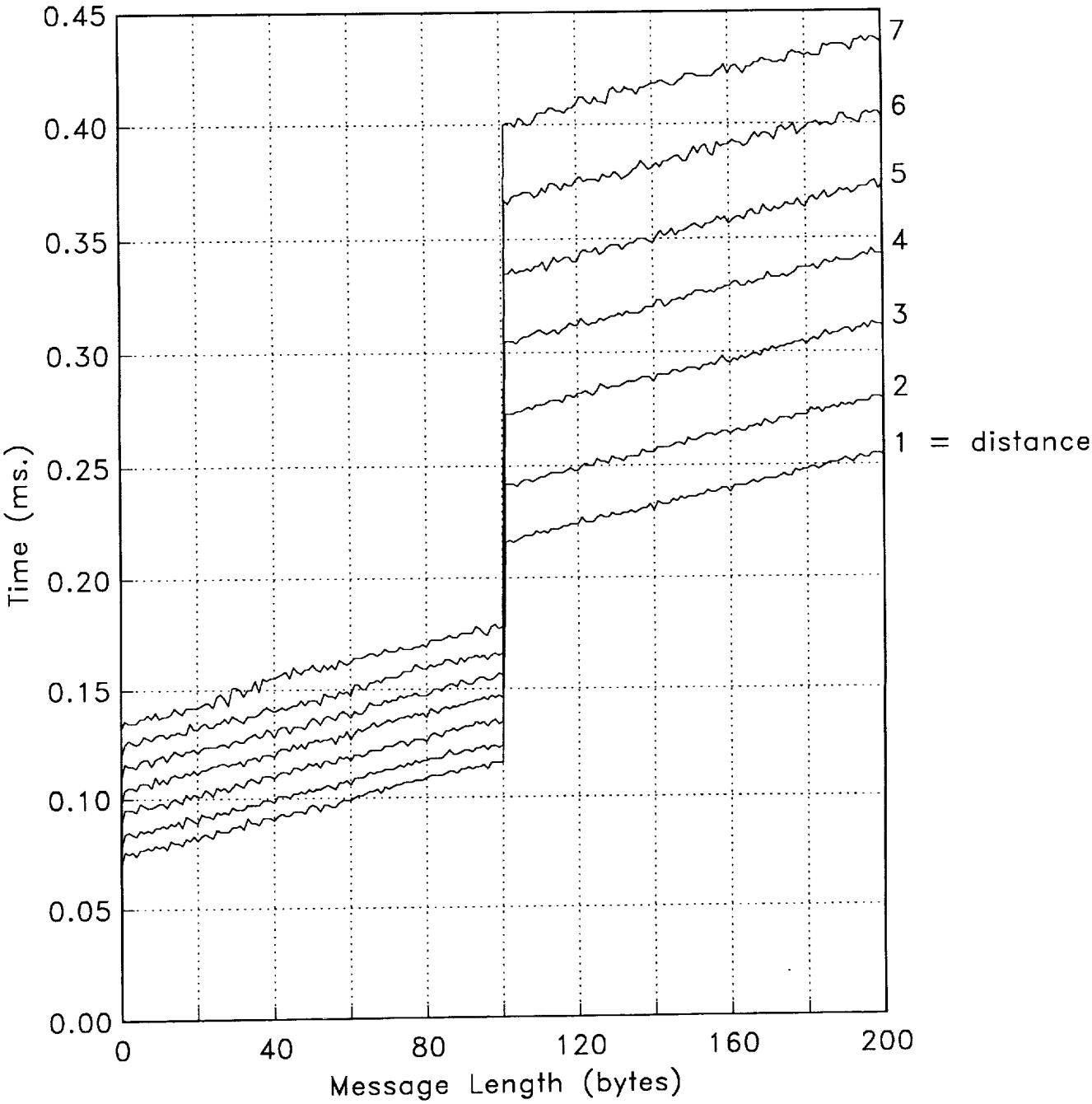

Figure 2: Impact of path length: 0–200 bytes.

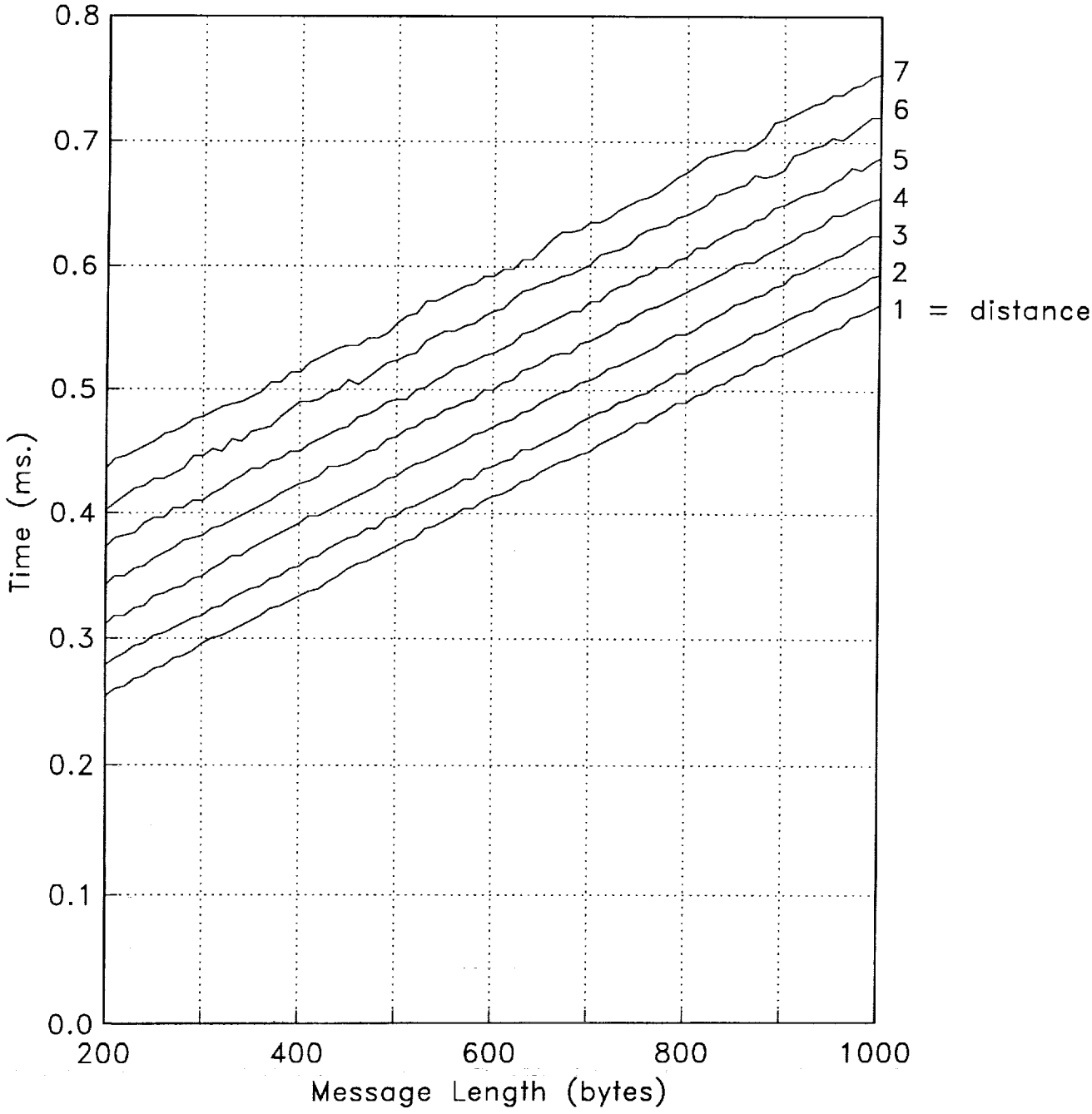

Figure 3: Impact of path length: 200–1000 bytes.

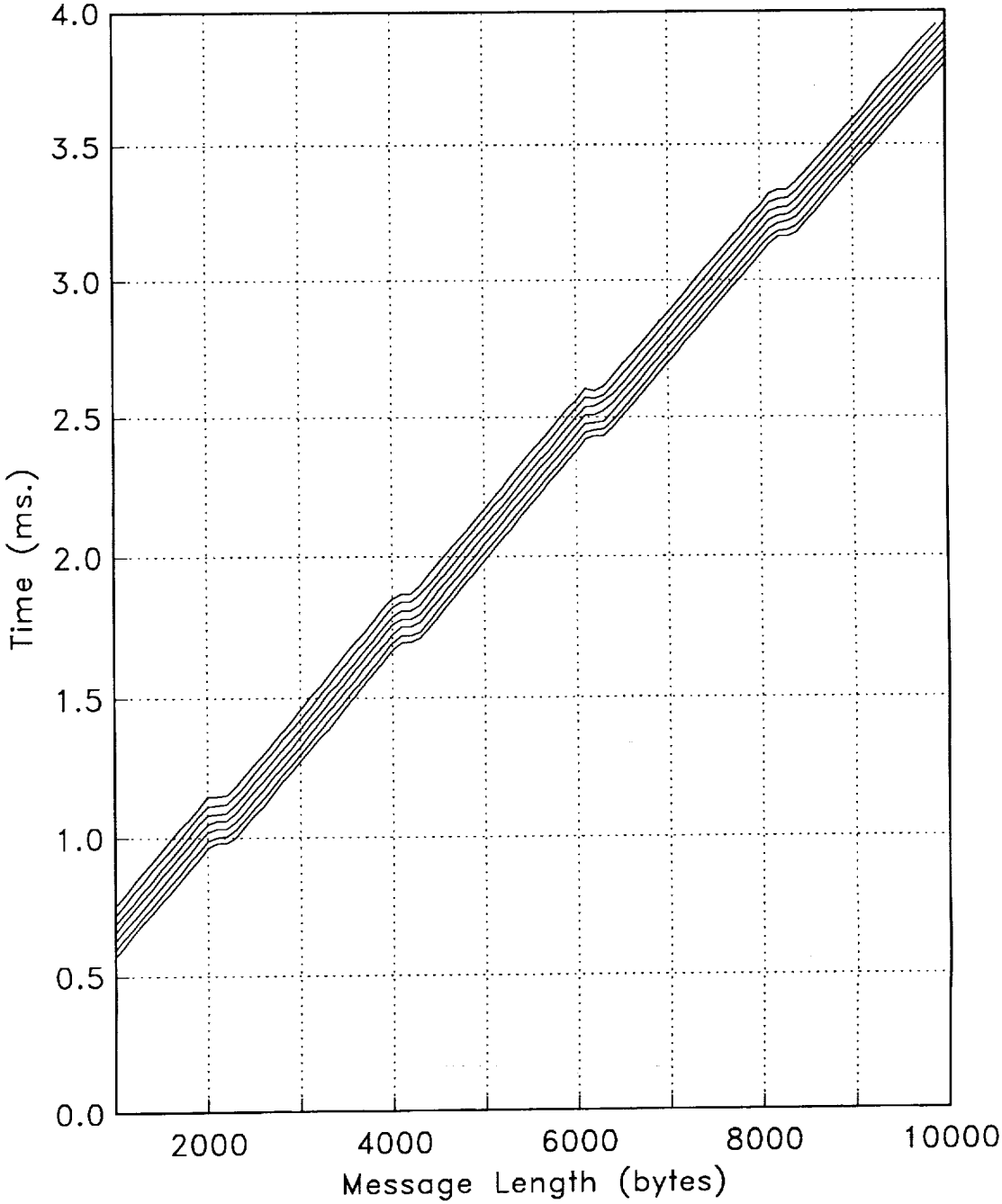

Figure 4: Impact of path length: 1000–10000 bytes.

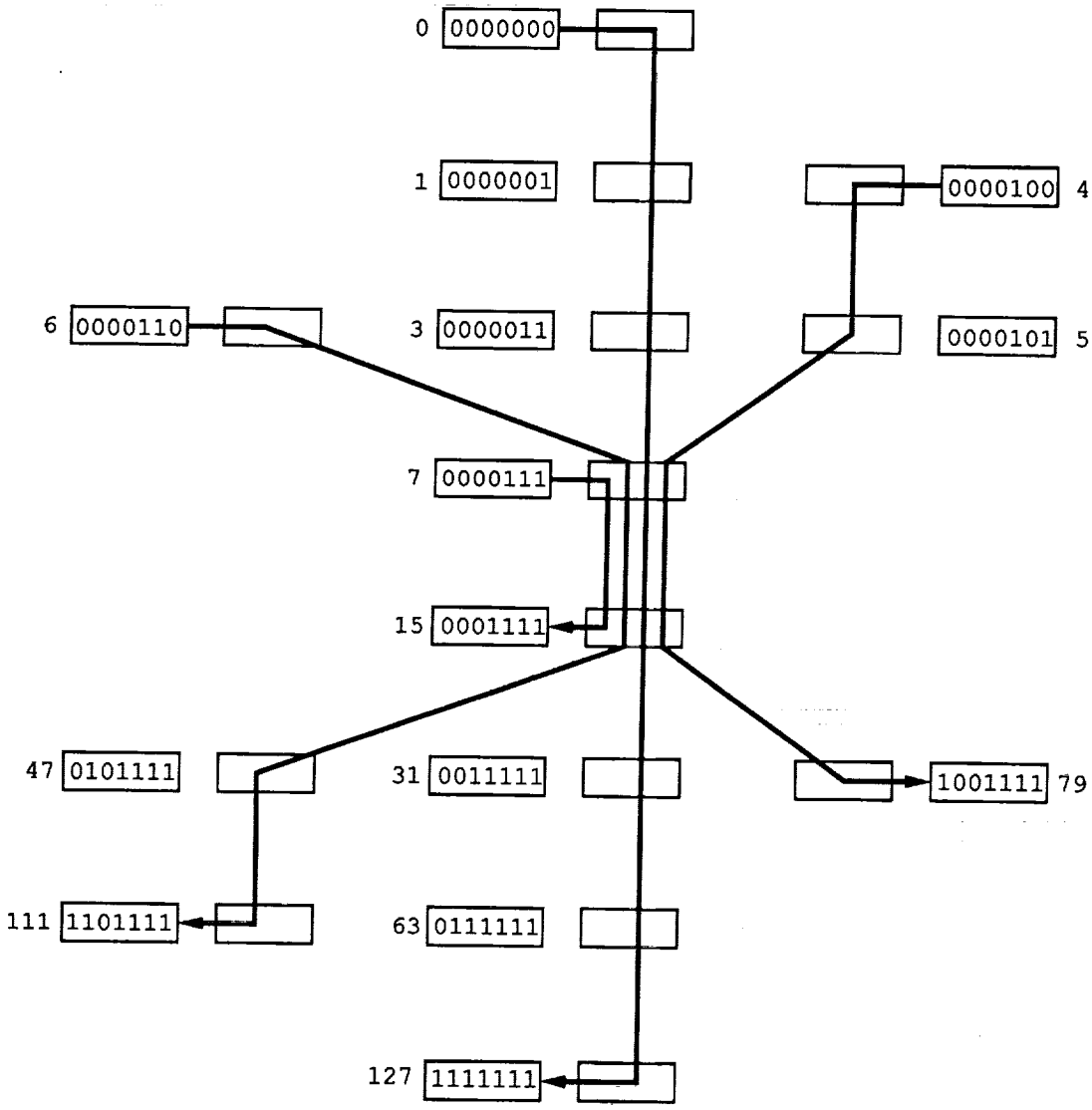

Figure 5: Communication pattern for first edge contention experiment.

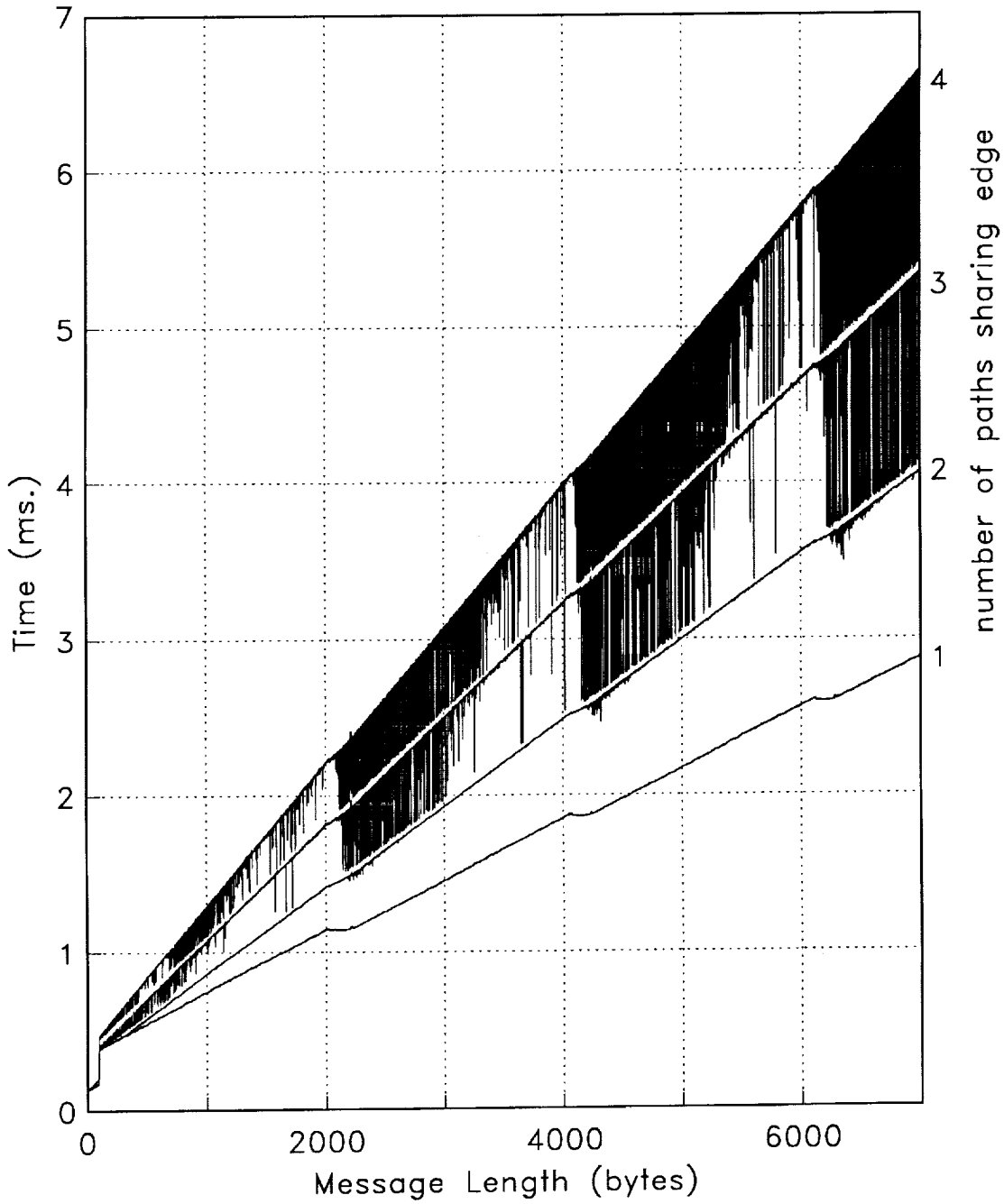

Figure 6: iPSC-860 : First contention experiment: line plot.

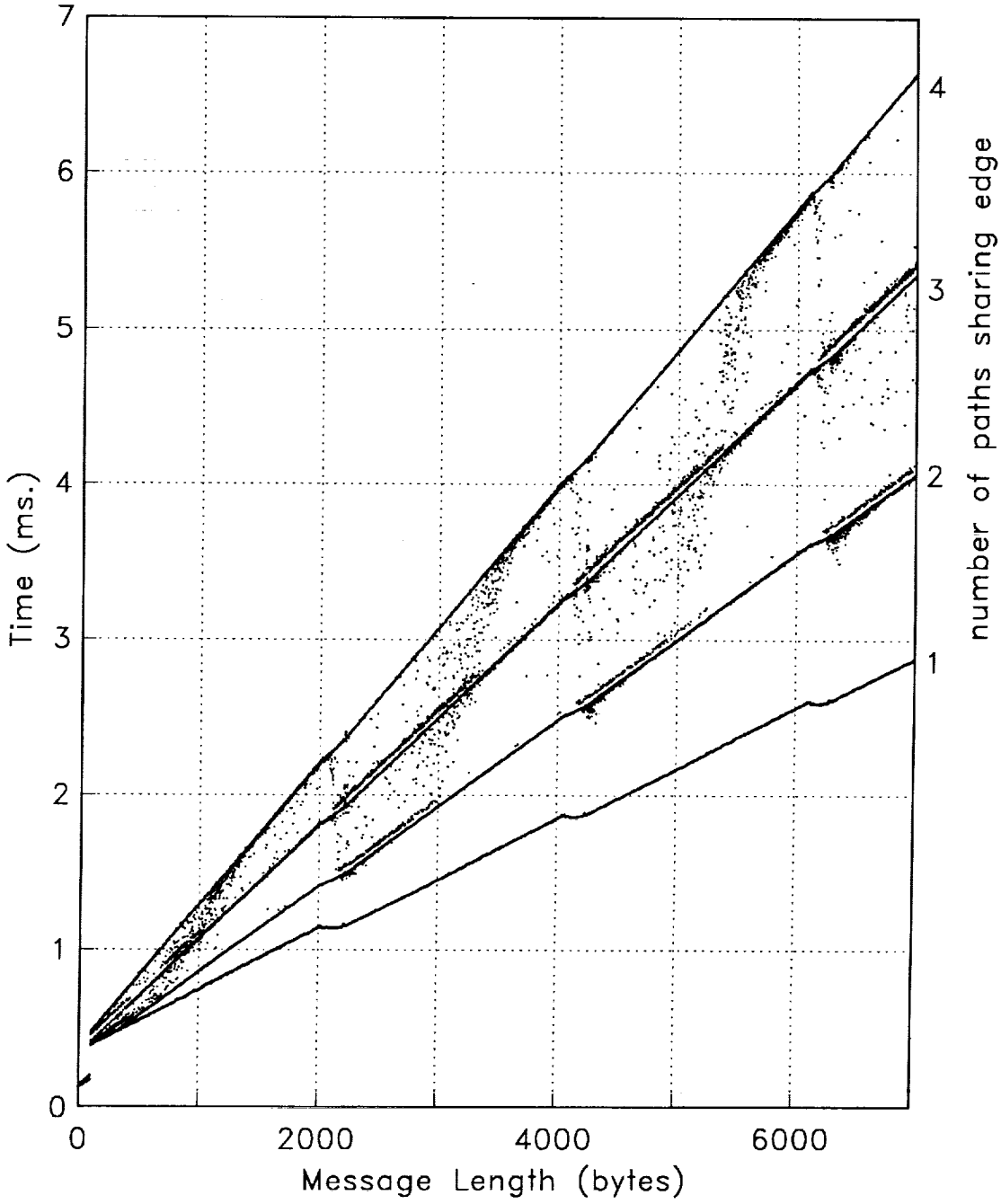

Figure 7: iPSC-860 : First contention experiment: scatter plot.

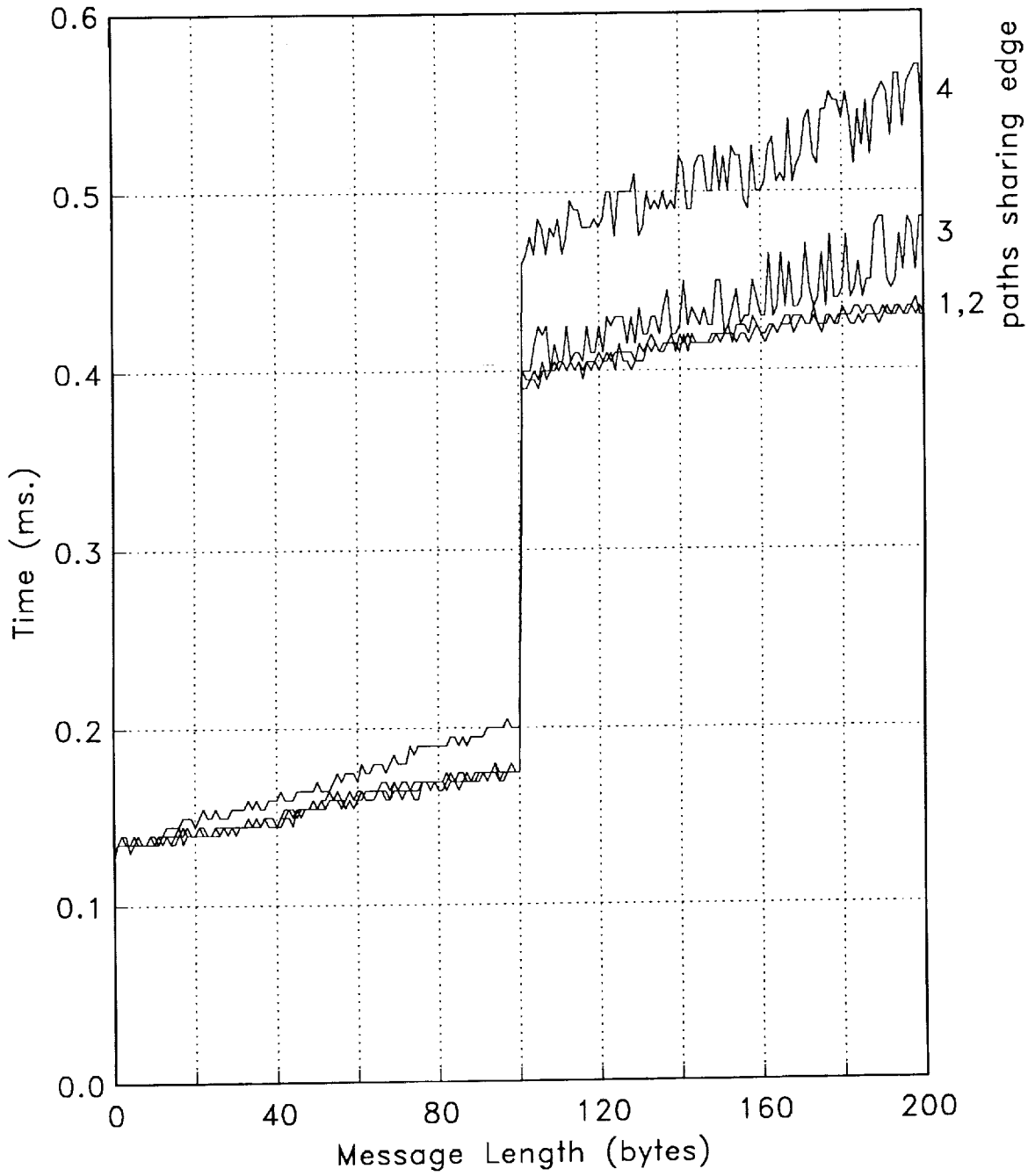

Figure 8: First contention experiment: detailed plot.

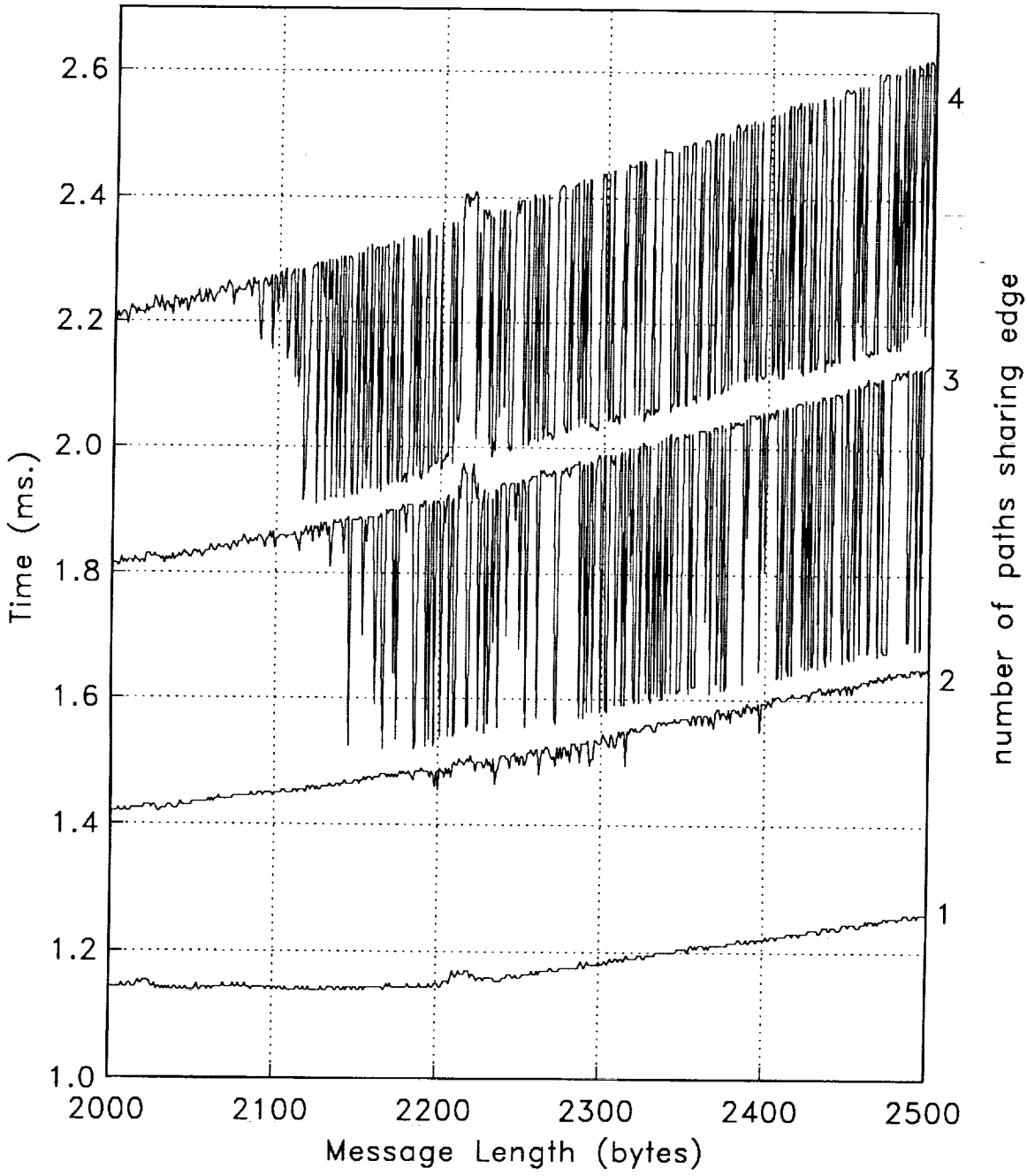

Figure 9: First contention experiment: detailed plot.

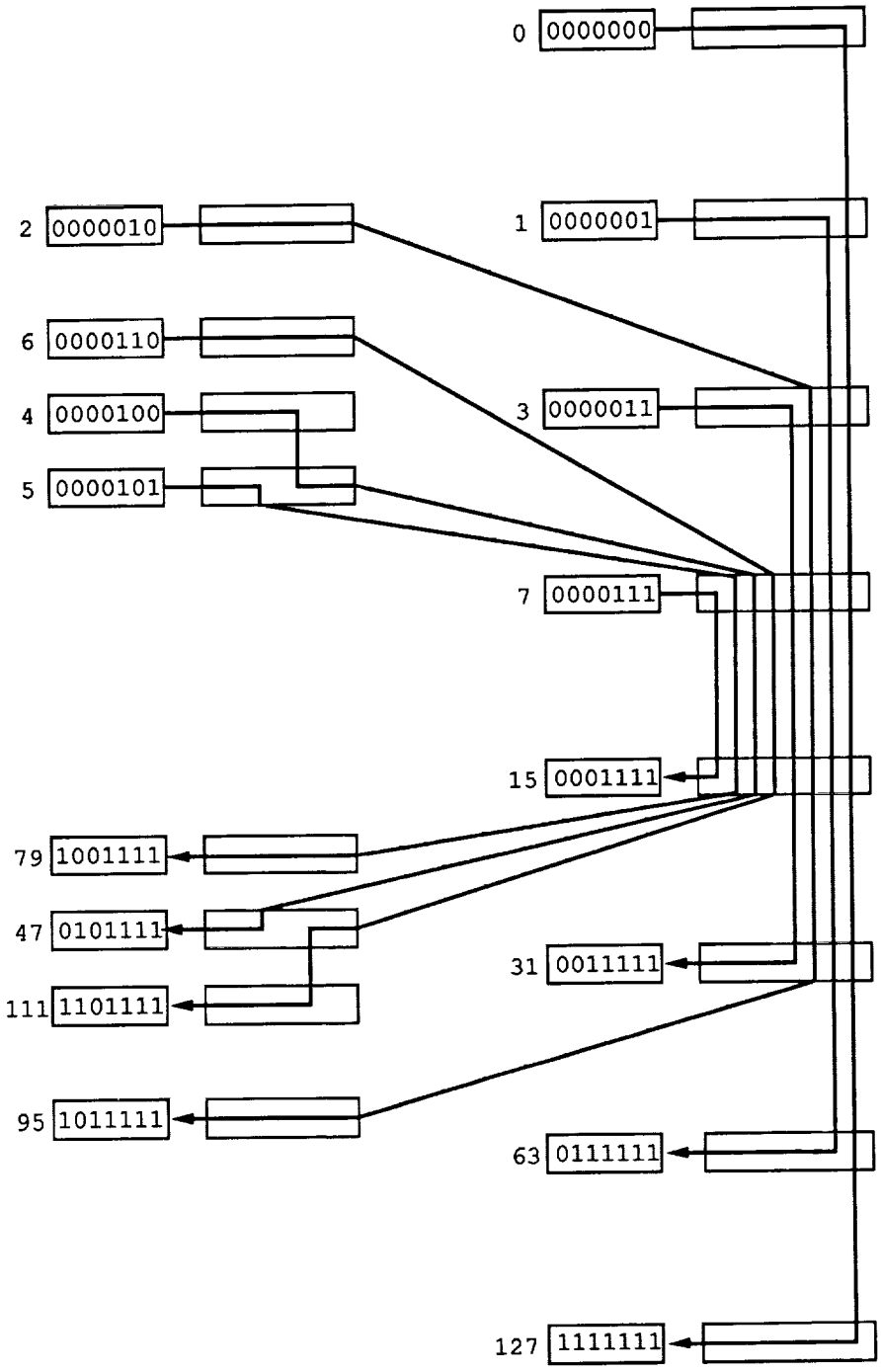

Figure 10: Communication pattern for 2nd. and 3rd. contention experiments.

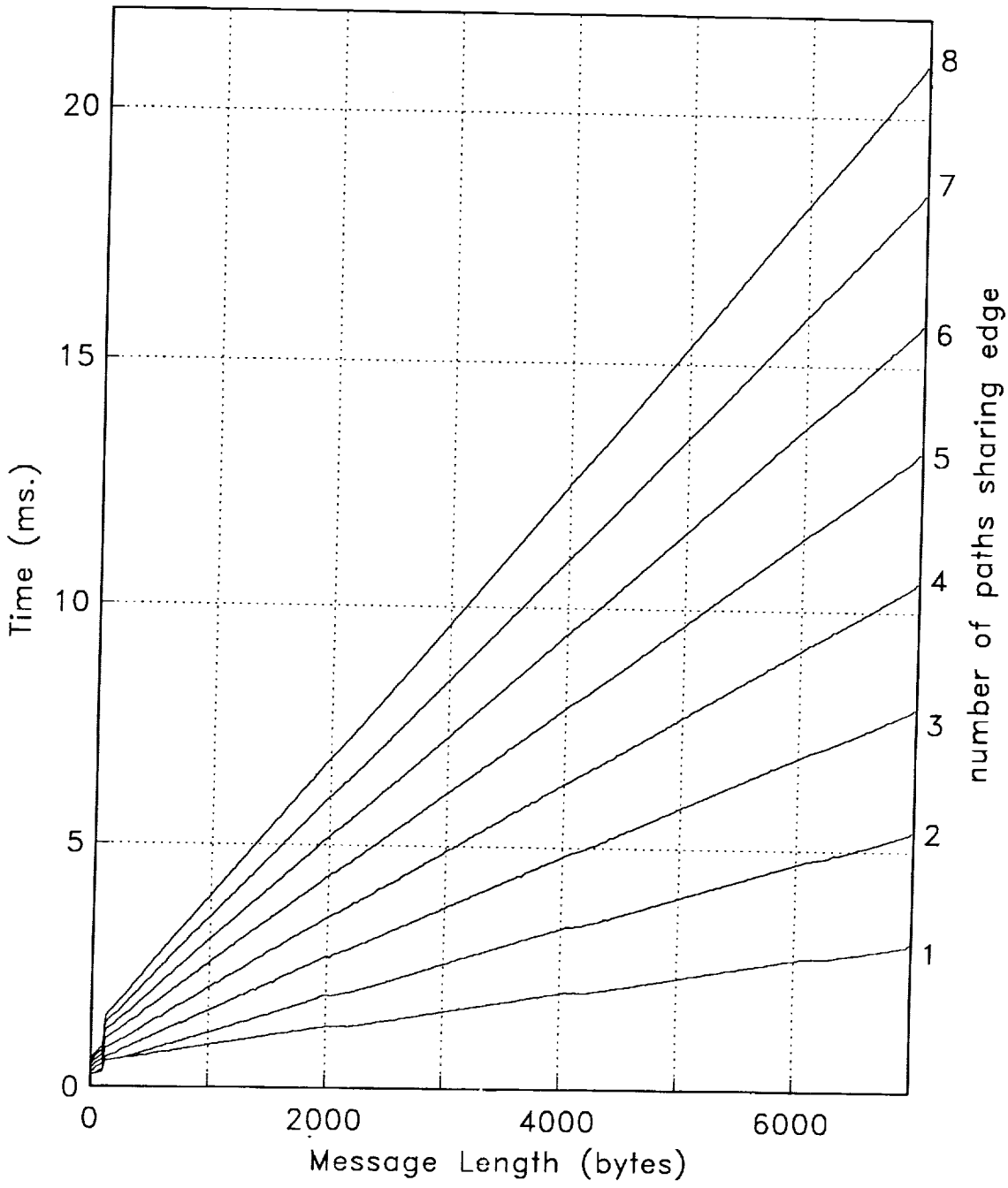

Figure 11: Second contention experiment: line plot.

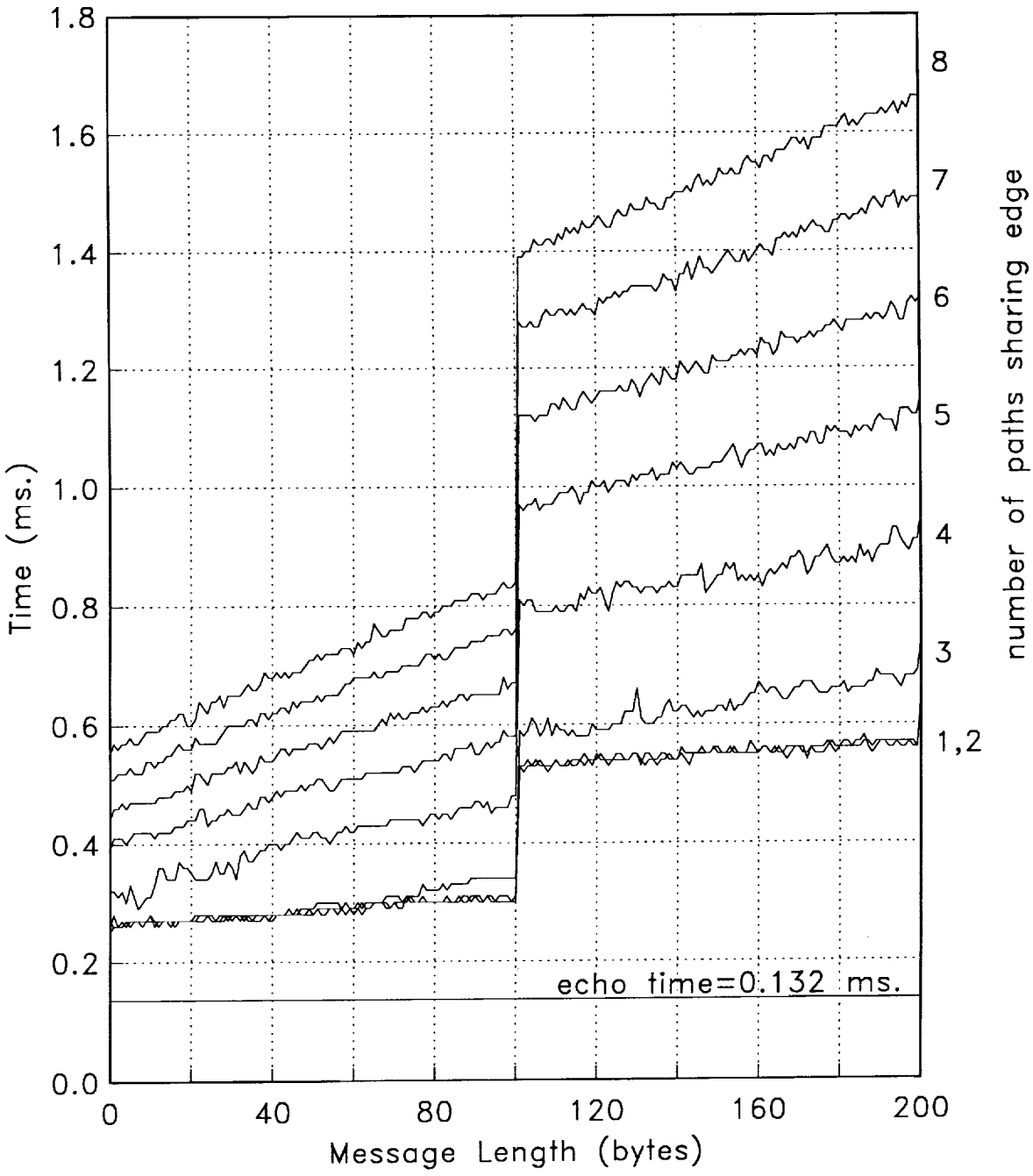

Figure 12: Second contention experiment: detailed plot.

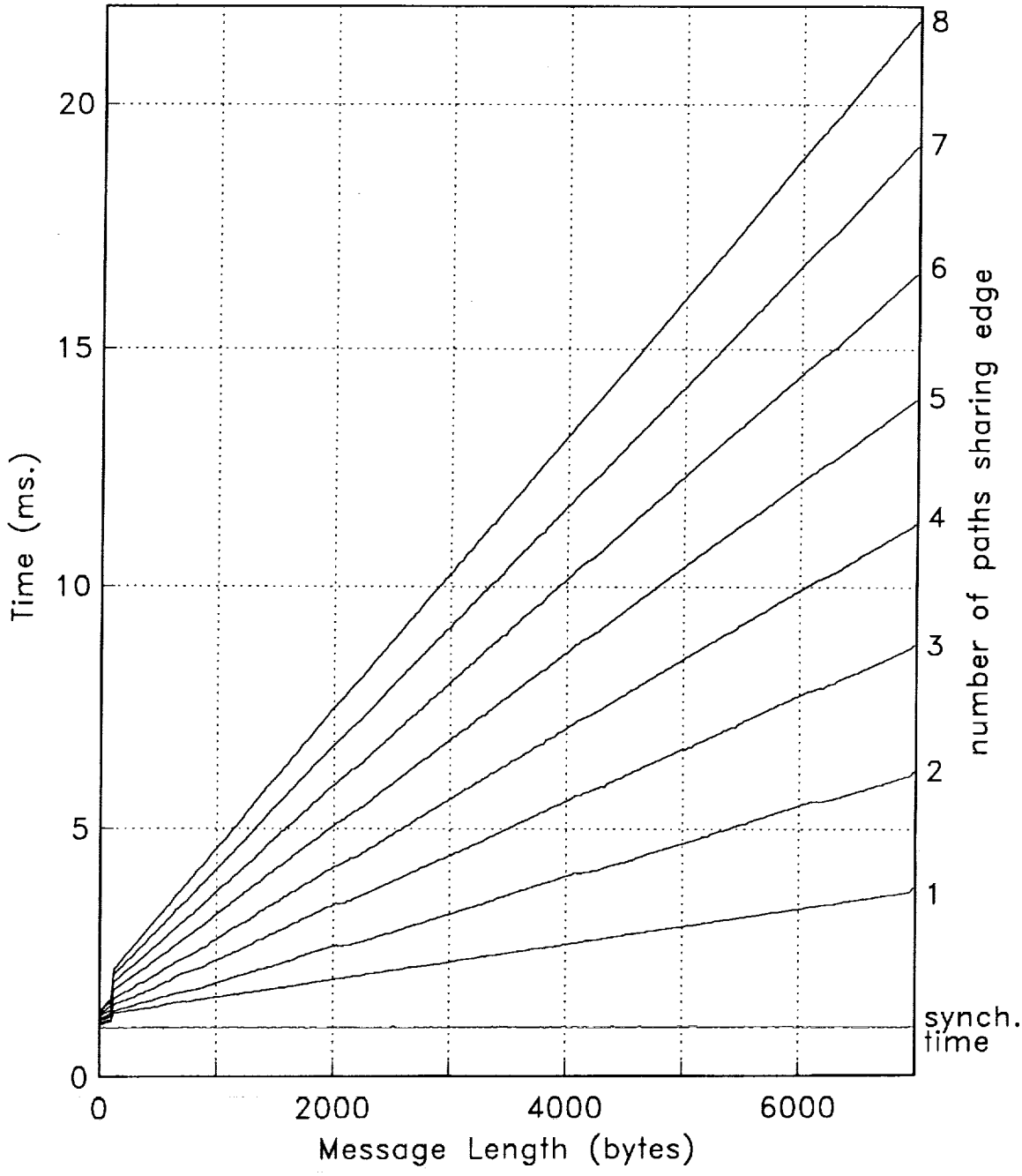

Figure 13: Third contention experiment: line plot.

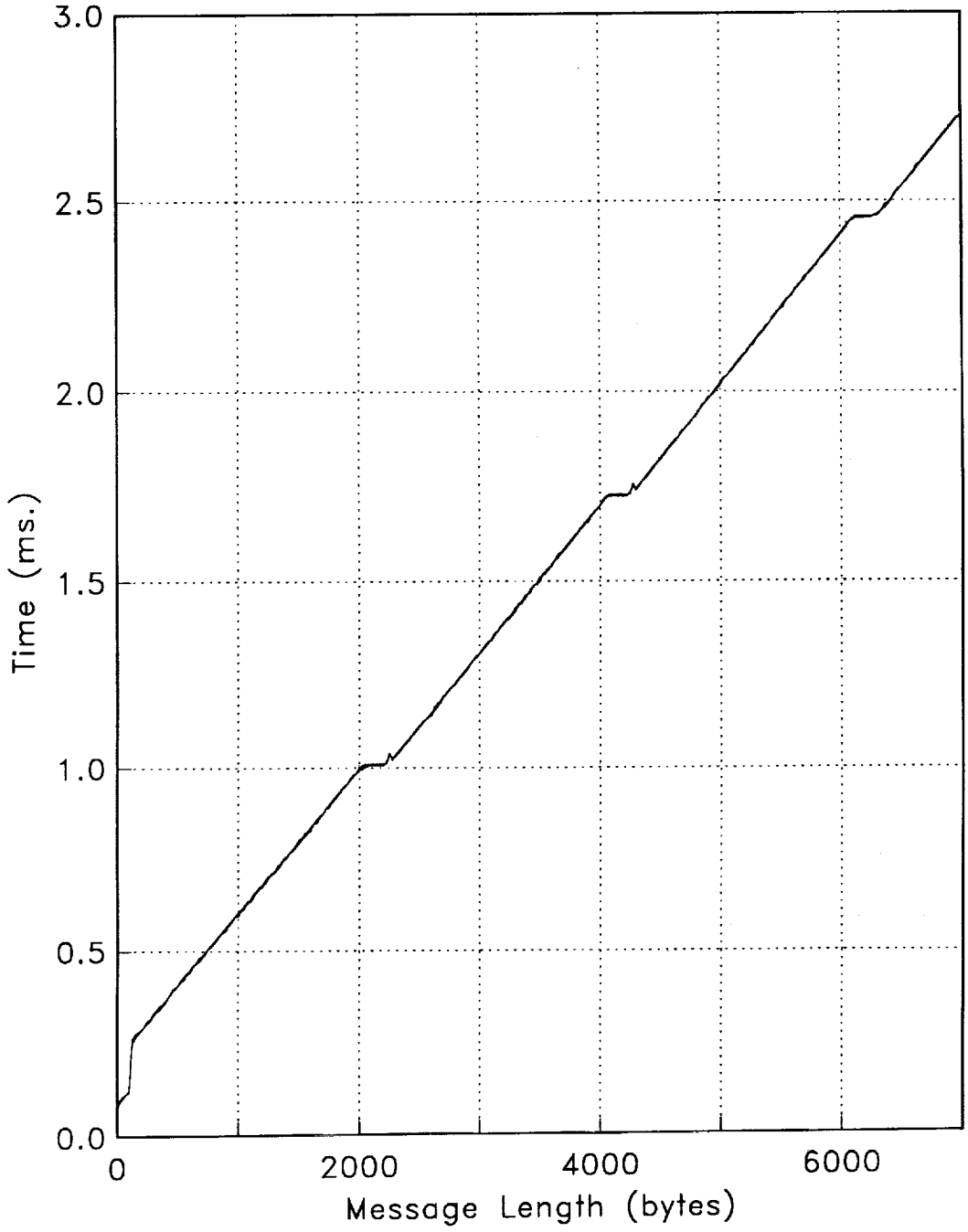

Figure 14: Nodal congestion experiment.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-182055 ICASE Interim Report No. 10		

4. Title and Subtitle	5. Report Date
COMMUNICATION OVERHEAD ON THE INTEL iPSC-860 HYPERCUBE	May 1990
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Shahid H. Bokhari	Interim Report 10
	10. Work Unit No. 505-90-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Final Report

16. Abstract

Experiments have been conducted on the Intel iPSC-860 hypercube in order to evaluate the overhead of interprocessor communication. It is demonstrated that (1) contrary to popular belief, the distance between two communicating processors has a significant impact on communication time, (2) edge contention can increase communication time by a factor of more than 7, and (3) node contention has no measurable impact.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
circuit-switching, hypercubes, Intel iPSC-860, contention	60 - Computer Operations and Hardware 62 - Computer Systems Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	27	A03

NASA FORM 1626 OCT 86