

Measured Performance of an Ethernet Local Network

by John F. Shoch and Jon A. Hupp

February 1980

© Xerox Corporation 1980

Abstract: The Ethernet communications network is a broadcast, multi-access system for local computer networking, using the techniques of *carrier sense* and *collision detection*. Recently we have measured the actual performance and error characteristics of an existing Ethernet installation which provides communications services to over 120 directly connected hosts.

This paper is a report on some of those measurements -- characterizing "typical" traffic characteristics in this environment, and demonstrating that the system works very well. About 300 million bytes traverse the network daily; under normal load, latency and error rates are extremely low and there are very few collisions. Under extremely heavy load -- artificially generated -- the system shows stable behavior, and channel utilization remains above 97%, as predicted.

CR Categories: 3.81.

Key words and phrases: Local computer networks, Ethernet system, carrier sense multiple access with collision detection (CSMA/CD), packet switching, measurements.

XEROX

PALO ALTO RESEARCH CENTER

3333 Coyote Hill Road / Palo Alto / California 94304

1. Introduction

One of the most attractive architectures for a local computer network is a shared, multi-access bus with distributed control. This approach was originally articulated in the design of the Ethernet communications network, in which all of the host machines share access to a single, passive coaxial cable [Metcalf & Boggs, 1976].

A number of interconnected Ethernet system installations have been in operation for several years, providing communications service for many hosts and supporting a wide variety of applications. The networks have always functioned very well, and probably represent the largest collection of local networks in use today. The intent of the present study was to further our understanding of this system by characterizing the actual behavior of one of these installations in normal use: traffic volume, packet sizes, error rates, efficiency, and more. In addition, we have constructed a test environment that allows us to artificially generate extreme amounts of traffic -- well beyond the expected operating conditions -- in order to help assess performance under heavily loaded conditions.

Successful use of a local computer network depends on much more than the specific communication medium: applications are using higher level protocols, including a complete architecture for internetwork communication among many different systems [Boggs, *et al.*, 1980]. Those topics, however, are beyond the range of the current study, which is directed primarily at the behavior of the Ethernet system itself. A preliminary version of these results was reported in [Shoch & Hupp, 1979]; more detailed discussion will be found in [Shoch, 1980].

In the sections which follow, we review the Ethernet local network, describe the experimental environment and test methodology, measure the system under normal operating conditions, and then measure the behavior with artificially high loads.

2. Review of the Ethernet principles

The approach used in the Ethernet system traces its roots back to the radio-based Aloha packet switching network, developed at the University of Hawaii [Abramson, 1970]. In the Aloha network, terminals equipped with packet radios shared a common multi-access radio channel. There was, however, no central controller allocating access to the channel; instead, a *random access* procedure was used, in which each terminal independently decided when to transmit. With this simplest *pure Aloha* scheme, two terminals may transmit at the same time producing a collision; after a collision, each terminal will have to wait for a random interval and then retransmit. The attraction of this approach is the elimination of any central control; as the load increases, however, the maximum possible utilization of a pure Aloha channel is only 18% [Abramson, 1977].

Unlike the pure Aloha scheme, when an Ethernet station wishes to send a packet a *carrier sense* mechanism is first used, forcing the station to *defer* if any transmission is in progress. If no other station is transmitting, the sender can begin immediately, with zero latency; otherwise, the sender

waits until the packet has passed before transmitting.

It is possible that two or more stations will sense that the channel is idle and begin transmissions simultaneously, producing a *collision*. Each sender, however, continues to monitor the cable during transmission, and can provide *collision detection* when the signal on the cable does not match its own output. In that case, each station stops transmitting, uses a *collision consensus enforcement* procedure to ensure that all other colliding stations have seen the collision, and then stops. A retransmission is then scheduled for some later time.

To avoid repeated collisions, each station waits for a random period of time before retransmitting. To avoid overloading the channel, thus making it unstable, the range of this retransmission interval is increased under times of heavy load, using a *binary exponential backoff* algorithm.

Taken together, these mechanisms represent the Ethernet random access procedure, also referred to as *carrier sense multiple access with collision detection*, or CSMA/CD. This access procedure can be applied to any suitable broadcast channel, such as radio, twisted pair, coaxial cable, fiber optics, diffuse infrared, or others. Coaxial cable is well suited for use in constructing a local computer network -- a system designed to provide high-bandwidth communications for machines within a distance of approximately 1 kilometer.

The shared component of an Ethernet local network consists of a single coaxial cable, typically strung in a meandering fashion through a building, perhaps in the ceiling or under a raised floor. As shown in Figure 1, individual computers, or *stations*, connect to the cable with the use of a CATV-style *tap*; a small *transceiver* is connected at the tap, with a cable running down to the *interface* which might be located in the station. The use of a passive medium, and the lack of any active elements in the shared portion, combine to help provide a very reliable and flexible system. This approach also provides for easy reconfiguration of stations: machines can be moved -- disconnected from one point and reattached at another -- without any need to take down the network. (For further details on the design of the Ethernet system, see [Metcalf & Boggs, 1976; Metcalfe, *et al.*, 1977; Shoch, 1979, 1980; and Crane & Taft, 1980].)

All of these mechanisms are combined to increase the probability that an Ethernet system will correctly deliver a client's packet. The network gives its best effort, but -- like any other low-level communications subsystem -- the Ethernet communications network cannot provide a 100% guarantee of delivery. In a complete network architecture, suitable *packet protocols* will be layered on top of this local network, to provide further reliability.

3. Experimental environment

The principles of the Ethernet system were first realized in the *experimental* or *prototype* system developed at the Xerox Palo Alto Research Center: using regular coaxial cable, running at 2.94 Mbps. There are now many of these networks in use in locations throughout the United States, tied

together with the Pup internetwork architecture [Boggs, *et al.*, 1980].

The particular local Ethernet network installation chosen for these measurements is one of the oldest in existence: it spans about 1800 feet (550 m.) and connects over 120 machines. These machines include large numbers of single-user stand-alone computers (mainly the Alto computer [Thacker, *et al.*, 1980]), two time-sharing servers running the Tenex operating system, numerous shared printers and file servers, as well as several internetwork gateways.

Applications which use the network include file transmission to the printers, file transfer to one of the storage systems, specialized multi-machine programs, access to shared data bases, remote diagnostics, down-line loading of programs, terminal access to the time-sharing machines, and others.

To conduct the measurements, a series of specialized test and monitoring programs has been built to assess the behavior of the network. The broadcast nature of an Ethernet system makes it particularly well suited for the passive collection of measurements: an individual station can sense the state of the cable and -- by operating its interface in a *promiscuous* mode -- can receive all of the packets passing by. (We thus avoid many of the Heisenberg problems: no processing is done at the source or destination to collect statistics, and no communications bandwidth is used to report them.)

Most of the measurements in the next two sections were collected using this passive technique, watching packets that go by on the cable.

4. Reliability and packet error rates

One of the major objectives of any local network is to provide reliable communications facilities, reflected both in the continued availability of the network itself, and in the lowest possible error rate as seen by individual hosts. Overall reliability was one of the most important considerations in the development of the Ethernet network. Thus, the only shared component in the communications subsystem is the passive coaxial cable; the shared portion has no electronics, no active components, no switches, no power being supplied through the line, and no power supplies.

In practical operation, there have been very few system-wide failures; almost all of these are caused by human error of some sort -- for example, removing the terminator at the end of the cable. On one occasion, a lightning strike generated a large surge in the building ground, disabling a number of improperly installed transceivers. Modified installation procedures and a minor circuit change have now significantly improved lightning resistance.

These system-wide failures are easy to control, and overall reliability is very high. Individual packets, however, are subject to damage when being transmitted, and may arrive in error. The design of the Ethernet system does not attempt to guarantee 100% error-free operation, and applications may use suitable error control procedures, such as packet acknowledgments and

retransmissions. But, for sustained high performance, one would certainly want to minimize the occurrence of packet errors.

There are several mechanisms available to detect and discard a damaged packet. When a station receives a packet, the interface checks that an integral number of 16-bit words has been received and that the 16-bit cyclic redundancy checksum (CRC) is correct. If either of these conditions is not met, the packet is considered damaged and is normally discarded.

With a passive listener set to receive every packet on the net, our current machines initially reported one damaged packet in about 6000, but there was wide variance among machines. Further examination has indicated that most of these errors reflect problems in the receiver section of the interface itself, and the packets were actually well-formed on the cable. Experimentation with a revised interface has produced normalized packet error rates of about 1 in 2,000,000 packets. Since these error rates are so low, packets received in error have been excluded from all of the results reported below.

We should note that there is a much more significant cause of lost packets which has nothing to do with network performance: failure to keep a receiving interface running properly. Some networking systems have proposed use of an interface which will receive a packet and then have a lengthy *refractory period* or *blind spot* before the receiver is restarted; these devices have a long *receiver-to-receiver turnaround time*. During these intervals, any packets sent to this station may be correct on the cable, but entirely missed by the station. Interface turnaround times require careful attention in the design of network equipment (see [Shoch, 1980]); but this is a problem beyond the scope of the local network itself.

5. Performance characteristics under normal load conditions

5.1. Overall traffic characteristics

In normal use, this single Ethernet system carries about 2.2 million packets in a 24-hour period, totalling almost 300 million bytes. (For comparison, this traffic roughly corresponds to about half of the volume carried through the Arpanet on an average day.) Not surprisingly, the load is very light at night and is heaviest during the regular daytime hours, with a slight dip around lunchtime; see Figure 2, showing network load over a 24-hour period, sampled over fairly long intervals (six minutes each).

5.2. Utilization

Again measured over a full 24-hour day, this traffic represents an extremely modest usage of the net: ranging from about 0.60% to as much as 0.84%. During shorter periods, however, maximum utilization in the busiest interval is much higher: about 3.6% over the busiest hour, 17%

over the busiest minute, and 37% in the busiest second. Figure 3 represents the load observed over four minutes on the Ethernet network, with individual samples summed over one second intervals; note that the full range of this figure would be contained within just one sample in the previous figure.

These results help to verify our design assumption that computer communications applications tend to produce a "bursty" pattern of requests; what we observe on the shared Ethernet channel is the aggregation of some number of independent sources of "bursty" traffic.

5.3. *Packet length*

Packets sent through the system exhibit a bimodal distribution of packet length: most of the packets are short ones (containing terminal traffic, acknowledgments, etc.), but most of the total volume is carried in the large packets (often containing file transfer traffic) -- see Figure 4. This is similar to some of the measurements reported for the Arpanet [Kleinrock & Naylor, 1974], but very different from the distributions frequently used in analytical models of networks.

To some extent, the upper and lower limits on the length of packets represent artifacts of the various implementations. Almost all of the traffic consists of Ethernet packets carrying encapsulated internetwork packets, or Pups [Boggs, *et al.*, 1980]. Thus, the minimum sized packet with no data would usually include the Ethernet and Pup headers, while acknowledgments and packets with only 1 or 2 bytes of data would be just a bit larger. At the other extreme, software considerations usually impose an effective upper limit on the size of a packet; depending on the particular system, this ranges from about 200 to 540 bytes.

The mean packet length is about 122 bytes, and the median is about 32 bytes.

5.4. *Source-destination traffic patterns*

On a given day, over 120 hosts use this Ethernet installation -- nearly every machine known to be connected. The extent of this utilization ranges from a fraction of a percent to over 25% of the observed network traffic. Nearly 1300 different source-destination pairs communicate during this period; on the average, each host sends packets to more than 10 other hosts, but some of this traffic is concentrated to and from specialized servers (see the following section).

Figure 5 shows the source-destination traffic matrix for a fairly typical day. A heavily used server both sends and receives packets from many other hosts; each server appears in the figure as a pair of broken lines, intersecting on the diagonal and roughly symmetric about that point.

This matrix only indicates sources and destinations on the local Ethernet system, but some of these servers are gateways to other networks. Thus, they represent the path to some larger number of internetwork addresses, and there are many more internetwork source-destination pairs.

5.5. *Server traffic patterns*

As suggested from the source-destination traffic matrix, many of the packets are going to or from specialized servers: time-sharing systems, gateways, information servers, file systems, and printers.

Over one typical day, these identifiable servers sent about 69% of the packets, and received about 73%.

5.6. *Inter-packet arrival times*

The distribution of inter-packet arrival times over a 24-hour period is shown in Figure 6. The mean inter-packet arrival time is 39.5 ms., with a standard deviation of 55.0 ms. and a median of 8.5 ms.

A fair portion of the current traffic consists of request/response transactions with a server. With low utilization, it is not uncommon to have this exchange take place with no intervening packets being transmitted; some of the spikes in the inter-packet arrival time distribution represent the turnaround times of these servers.

Figure 7 indicates the cumulative distribution of inter-packet arrival times: it shows that 50% of the packets are followed by the next packet within 10 ms., 90% of the packets are followed by the next packet within 64 ms., and 99% within 183 ms.

5.7. *Latency and collisions detected by a sender*

With the current level of traffic, it is not surprising that most attempts to send a packet succeed on the first transmission -- there are no collisions and no need to backoff and retransmit.

To measure this result, a test program has been run which periodically wakes up and tries to transmit a packet; under normal load, it indicates that 99.18% of the packets make it out with zero latency, 0.79% of the packets are delayed due to deference, and less than 0.03% of the packets are involved in collisions.

5.8. *Overhead*

In addition to useful data, there are several forms of overhead which impact Ethernet communications: headers and error checking on every packet, as well as entire packets carrying acknowledgments, routing tables, or other ancillary information.

The overhead due specifically to Ethernet encapsulation is only 4 bytes; this represents about 3% of an average length packet. The standard internetwork protocol header consumes another 22 bytes, or about 18% of an average packet. Thus, data fields of all packets represent about 79% of the bits carried. It is worth noting that overhead accounts for less than 5% of the maximum length packet used for high-volume data transfers.

To get a precise measure of the actual data bits sent by user protocols one would need to exclude acknowledgment packets, and perhaps model both ends of the end-to-end process. As an approximation, however, we can obtain a reasonable estimate by measuring the amount of data carried inside data packets of the most common protocols. This will generally provide a conservative estimate of user traffic, counting as overhead all of the packets used to set up a connection, error packets, boot loader packets, and all of the specialized packet types exchanged by other user protocols. (This procedure may also double-count the data in any packets which are actually retransmitted by the source; in general, that number is very small.)

Using this alternative approach, about 69% of all the bits carried through the Ethernet system have been classified as user data, leaving a total of about 31% encompassing all forms of overhead.

5.9. *Intranet vs. internet traffic*

About 72% of all of the packets seen have both their source and destination on this network -- they are intranet packets. About 28% of the packets are internet traffic, coming from or going to another network via an internetwork gateway. The presence of the two large time-sharing systems on this particular network accounts for much of this traffic, as users access these hosts from distant locations. In addition, it is possible for a local Ethernet system to serve as a transit network for traffic originating on one network and destined for another; with the current topology, however, we see few such "transit packets" on this particular network.

6. Performance characteristics under high load conditions

The foregoing discussion has considered network performance under normal operating conditions, reflecting the current demands placed upon the network by the existing set of applications. Further growth of new systems will increase the load on the net, and the system should certainly be able to accommodate short term bursts of very high load. One of the considerations in designing an application would be the performance of an Ethernet system as the load increases dramatically.

Initial modelling of the Ethernet system approach [Metcalf & Boggs, 1976] has indicated that the system should continue to function very well as the load goes up. The following measurements serve to verify that prediction.

In a high load situation, each individual host can be set to generate packets at a specified rate, generally less than the channel capacity, and independent of the rate at which these packets are actually sent through the channel. The total offered load then represents the sum of these individual offered loads, and may be less than 100% of the channel capacity, or more than 100% (in which case some of the transmit queues overflow, and packets are lost). This load can be varied by changing the number of hosts, or the load generated per host. (In section 7 below, we explore a bit

further the special case in which each host can be viewed as a continuously queued source, always trying to transmit a packet, and generating a new one as soon as this one is sent. Thus, each host can be viewed as attempting to present a load of 100%. In this case, it is only the number of hosts attempting to transmit which is of interest; it is not meaningful to talk about increasing the load generated by each host.)

Measurements of interest include the actual level of channel utilization, the stability of the system as the load increases (does the total network utilization dramatically decrease as the load increases?), and the fairness with which the channel is shared.

To help gauge this behavior under high load, and to help stress the capabilities of the system, we have constructed test programs which generate artificially high levels of traffic. Using a special control program, we use the network to find idle machines and then load them with a test program which can be set to produce a specified offered load to the network. Statistics are accumulated as to the number of packets successfully sent and at the end of a test period, after all the test machines are idle, the controlling machine can reach out and collect these statistics.

When a test program generates a packet for transmission, it is placed upon an output queue; actual transmission is performed by an output process, which also handles retransmissions due to collisions. The presence of the queue allows the sender to tolerate short term variations in the availability of the channel. If the total offered load exceeds 100%, however, it is clear that some of the traffic being generated will never be transmitted successfully, and it merely gets dumped off of the output queue.

One can start with a modest offered load, and then increase that level by either adding more machines or increasing the load being offered by each host. With ideal scheduling, one would like to see the total channel utilization increasing with the total offered load, up to 100% (see Figure 8). Beyond 100% load -- under *very* heavy load -- we would like to see the channel utilization remain at 100%, representing full use of the available capacity.

Few real system can perform this well. A pure Aloha channel gets 18% utilization at best and a slotted Aloha channel can realize 37% utilization. Both Aloha mechanisms can become unstable: utilization declines significantly as offered load increases [Abramson, 1977]. In general, one tries to use a control procedure that will maximize the utilization, while remaining stable as the load increases.

Actual test runs were made on the regular Ethernet system described above, usually at night, when there was very little other traffic. The high load results reported below were for full-size data packets (512 bytes), typical of most of the volume moved through the net.

6.1. *Maximum utilization*

As the total offered load increases from 0% to 90%, channel utilization matches it perfectly: all of the traffic gets out correctly. As the offered load moves above 90%, the channel utilization flattens out at a level above 96% (see Figure 9). This experiment has been run with even greater

loads, as high as 9000% (well beyond any reasonable operating region!); utilization continued to increase slightly, approaching 98% -- just 2% short of the ideal case.

6.2. *Stability*

As Figure 9 indicates, an Ethernet system under high load shows no instability: the throughput curve does not decline as the total offered load increases.

6.3. *Fairness*

The Ethernet control discipline is also very fair in its allocation of channel capacity: at 100% offered load (10 hosts at 10%/host) we observe a total utilization of 94% with individual throughputs ranging from 9.3 to 9.6%.

7. Performance characteristics with continuously queued sources

The preceding section described a general model in which both the load per host and the number of hosts could be varied, producing total offered load in the range of 0% to 150%. In this section we examine a bit more closely the special case in which every host is continually queued, trying to generate a load of 100% of the channel rate.

This corresponds to the analytical model developed in the original paper on the Ethernet communications network (see [Metcalfe & Boggs, 1976]); the actual measurement results are of use in trying to verify this model. In the ideal case, a single machine continuously transmitting would produce a utilization of 100%; with perfect scheduling, the addition of other machines would not cause the total utilization to decrease. But we do not have perfect knowledge nor scheduling, and as more users queue up to transmit, collisions take place, and the utilization decreases. The analytical results have predicted, however, that for reasonably large packet sizes the distributed control algorithm of the Ethernet would only cause the total utilization to decrease by a small percentage (see Figure 10a).

In practice, we can only approximate these test conditions, since the channel cannot be driven continuously by one host (due to non-zero transmitter-to-transmitter turnaround times). A single machine transmitting large 512 byte packets, for example, can offer a 95% load, but with smaller packet sizes the effective load becomes even lower. With the current population of machines, the most interesting (and feasible) measurements use from 5 to 64 hosts.

7.1. Maximum utilization

As predicted, the total utilization starts out very high, and decreases somewhat as additional hosts are added. The test results for various combinations of packet length and number of hosts are summarized in the table below (also see Figures 10a and 10b):

Ethernet utilization with continuously queued sources

Q = number of hosts, each generating 100% load

Q	Packet length P (in bytes)				
	512	128	64	6	4
5	97%	95%	94%	72%	----
10	97%	91%	89%	68%	58%
32	97%	90%	83%	64%	56%
64	97%	92%	85%	61%	54%

These measured results are very similar to the results predicted in the model used in the original paper on the Ethernet system. As expected, Ethernet utilization increases with packet size. For full-size data packets -- 80% of the typical traffic -- total utilization remains above 97%, even with 64 continuously queued hosts.

With small packets, any time lost to collisions and collision resolution becomes larger compared to the packet size, and total utilization decreases. Four bytes represents the smallest packet we can send through the Ethernet network, and utilization would decrease further if the packets were even smaller. Indeed, with even smaller packets the packet transmission time should approach the *collision interval* and network utilization approaches $1/e$, the maximum efficiency of a slotted Aloha network [Abramson, 1977].

7.2. Stability

As the table above shows, the Ethernet system remains stable even under extreme overload conditions. In experiments with as many as 90 hosts sending medium- to large-size packets (each offering up to a 95% network load) Ethernet utilization remains high, and shows no signs of suddenly decreasing or becoming unstable.

7.3. Fairness

With more than one continuously queued source, some of the traffic cannot be accommodated, and each station can get only some fraction of its nominally desired bandwidth (100% per host). In the full range of tests the system exhibits very good fairness to all of the machines. With 90 hosts sending, the average utilization per host is 1.1%, ranging from about .9% to 1.3%.

8. Concluding remarks

From these results, we can now begin to verify some important hypotheses about the performance of an Ethernet local network:

1. The error rates are very low, and very few packets are lost.
2. Under normal load, transmitting stations rarely have to defer and there are very few collisions. Those few collisions are resolved very quickly. Thus, the access time for any station attempting to transmit is virtually zero.
3. Under heavy load there are more collisions, but the collision detection and resolution mechanisms work well, and channel utilization remains very high -- almost 100%. In addition, channel sharing is quite fair.
4. Even under extreme overload, the Ethernet channel does not become unstable. (There have been several proposals for complex control schemes which have claimed higher utilization than the Ethernet approach, fairer allocation of the channel, or better performance [Shoch, 1979]. In light of the above results, these alternatives offer increased complexity with little potential benefit.)

With all of these characteristics, an Ethernet system remains a particularly attractive choice as an architecture for local communication.

9. Acknowledgments

The original design of the Ethernet system is due primarily to Robert M. Metcalfe and David R. Boggs; they have both been of great help in undertaking this project. We are also indebted to Edward A. Taft for his help in conducting this work.

10. References

[Abramson, 1970]

N. Abramson, "The Aloha system -- another alternative for computer communications," *AFIPS Conference Proceedings (FJCC 1977)*, 37, 1970, pp. 281-285.

[Abramson, 1977]

N. Abramson, "The throughput of packet broadcasting channels," *IEEE Transactions on Communications*, com-25:1, January 1977, pp. 117-128.

[Boggs, et al., 1980]

D. R. Boggs, J. F. Shoch, E. A. Taft, and R. M. Metcalfe, "PUP: An internetwork architecture," *IEEE Transactions on Communications*, com-28:4, April 1980.

[Crane & Taft, 1980]

R. C. Crane and E. A. Taft, "Practical considerations in Ethernet local network design," *Proc. of the 13th Hawaii International Conference on Systems Sciences*, January 1980, pp. 166-174.

[Kleinrock & Naylor, 1974]

L. Kleinrock and W. E. Naylor, "On measured behavior of the ARPA Network," *AFIPS Conference Proceedings (NCC 1974)*, 43, June 1974, pp. 767-780.

[Metcalfe & Boggs, 1976]

R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks," *CACM*, 19:7, July 1976, pp. 395-404.

[Metcalfe, et al., 1977]

R. M. Metcalfe, D. R. Boggs, C. P. Thacker, and B. W. Lampson, *Multipoint data communication system with collision detection*, United States Patent No. 4,063,220, December 13, 1977.

[Shoch, 1979]

J. F. Shoch, *An annotated bibliography on local computer networks (1st edition)*, Xerox Parc Technical Report SSL-79-5, and IFIP Working Group 6.4 Working Paper 79-1, October 1979.

[Shoch, 1980]

J. F. Shoch, *Design and performance of local computer networks*, in press, to appear 1980.

[Shoch & Hupp, 1979]

J. F. Shoch and J. A. Hupp, "Performance of an Ethernet local network -- a preliminary report," *Local Area Communications Network Symposium*, Boston, May 1979, pp. 113-125. Revised version presented at the *20th IEEE Computer Society International Conference (Compeon '80 Spring)*, San Francisco, February 1980.

[Thacker, et al., 1979]

C. P. Thacker, E. M. McCreight, B. W. Lampson, R. F. Sproull, and D. R. Boggs, *Alto: A personal computer*, Xerox Parc Technical Report CSL-79-11, August 1979. To appear in Siewiorek, Bell, and Newell (Eds.), *Computer Structures: Readings and Examples*, 2nd edition.

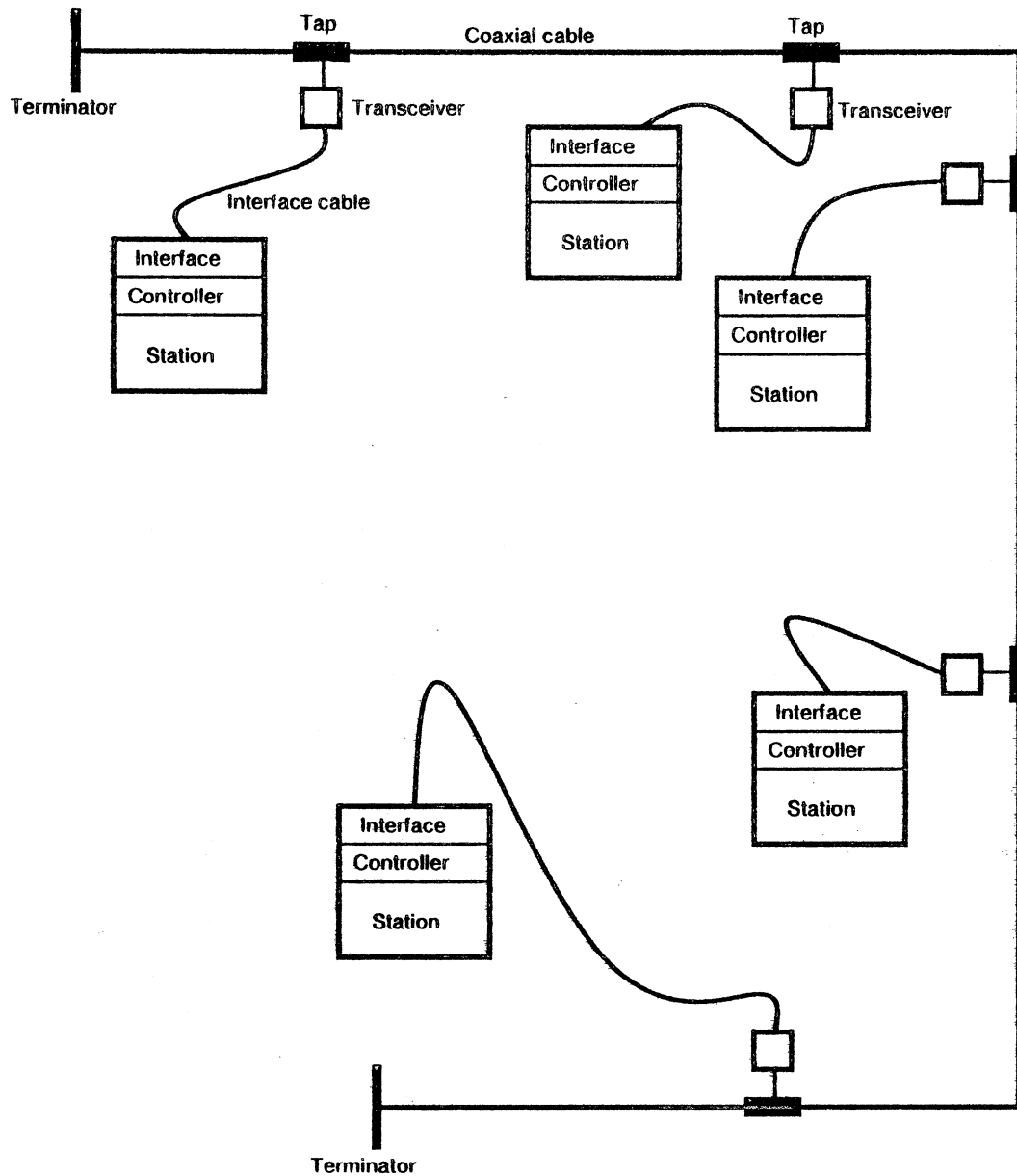
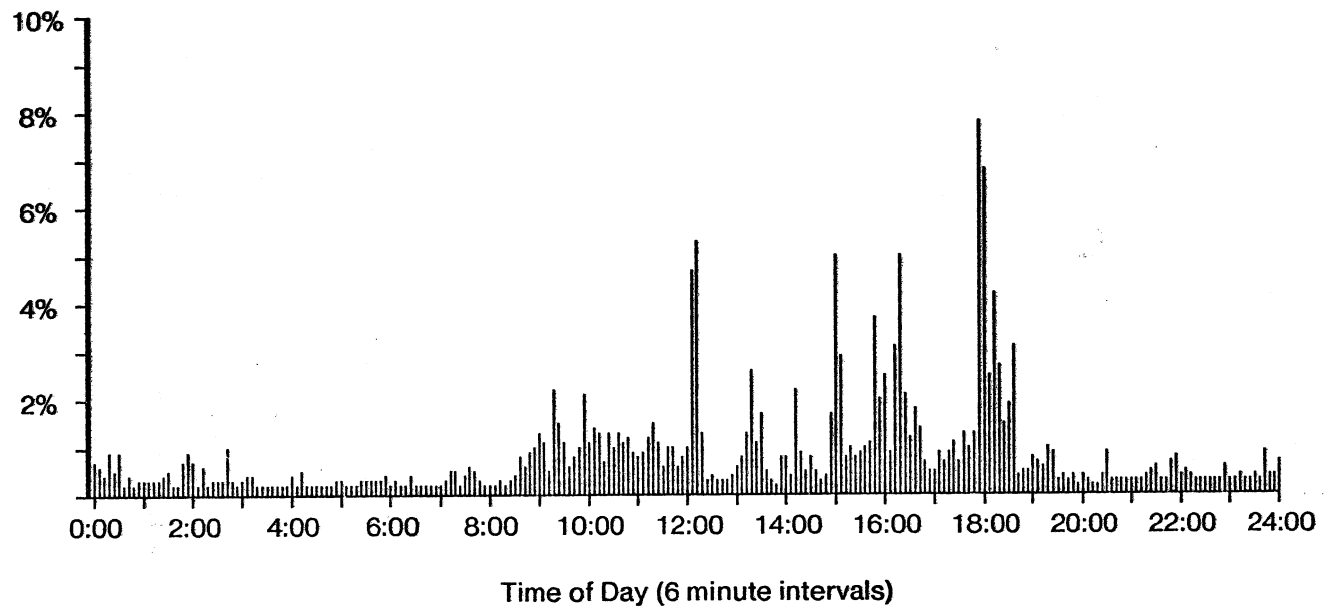


Figure 1: An overview of an Ethernet communications network (after [Metcalfe & Boggs, 1976]).

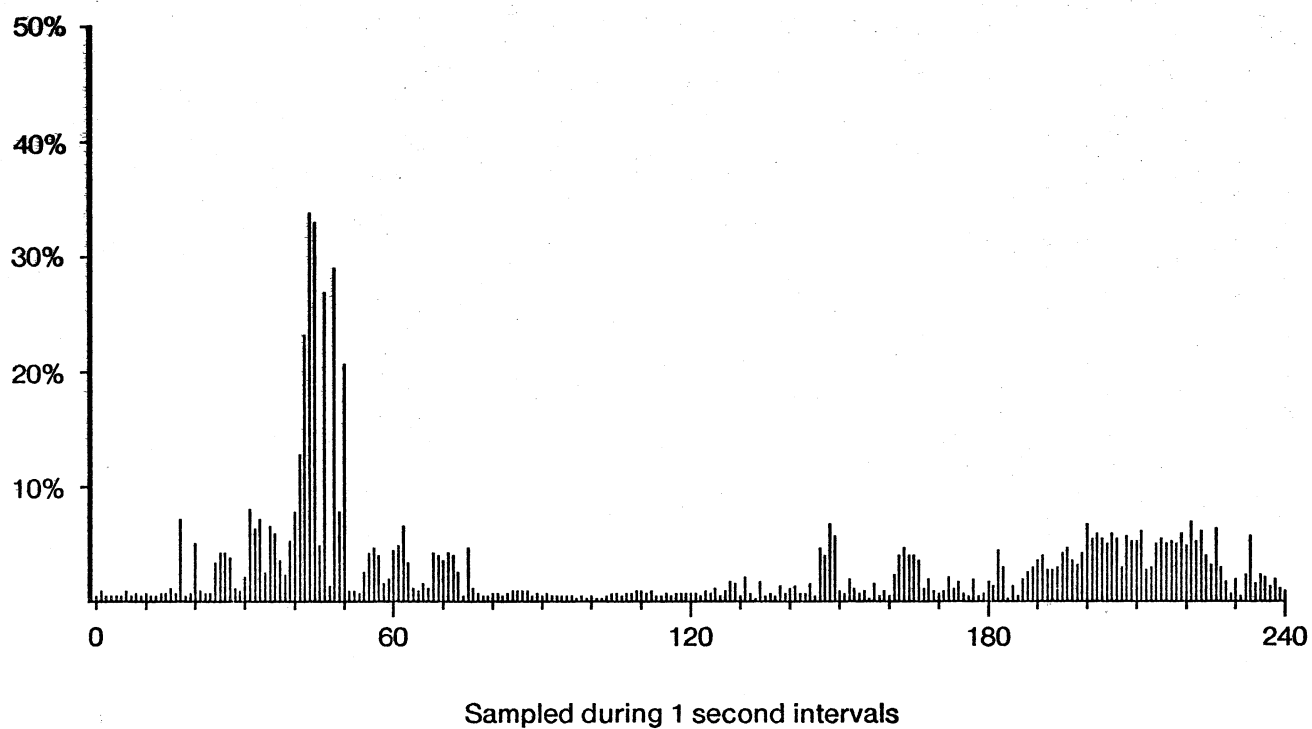


Max Load This Period = 7.9%

Min Load This Period = 0.2%

Average Load This Period = 0.8%

Figure 2: Ethernet load on a typical day, sampled over 6 minute intervals.



Max Load This Period = 32.4%

Min Load This Period = 0.2%

Average Load This Period = 2.7%

Figure 3: Ethernet load over a 4 minute period, sampled over 1 second intervals.

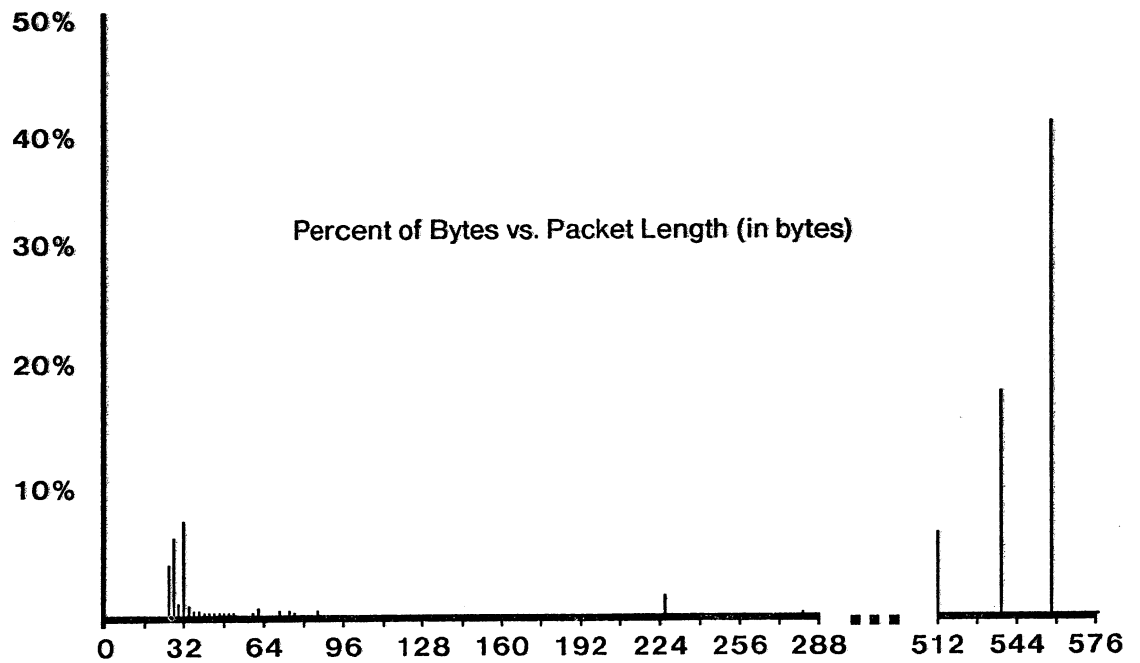
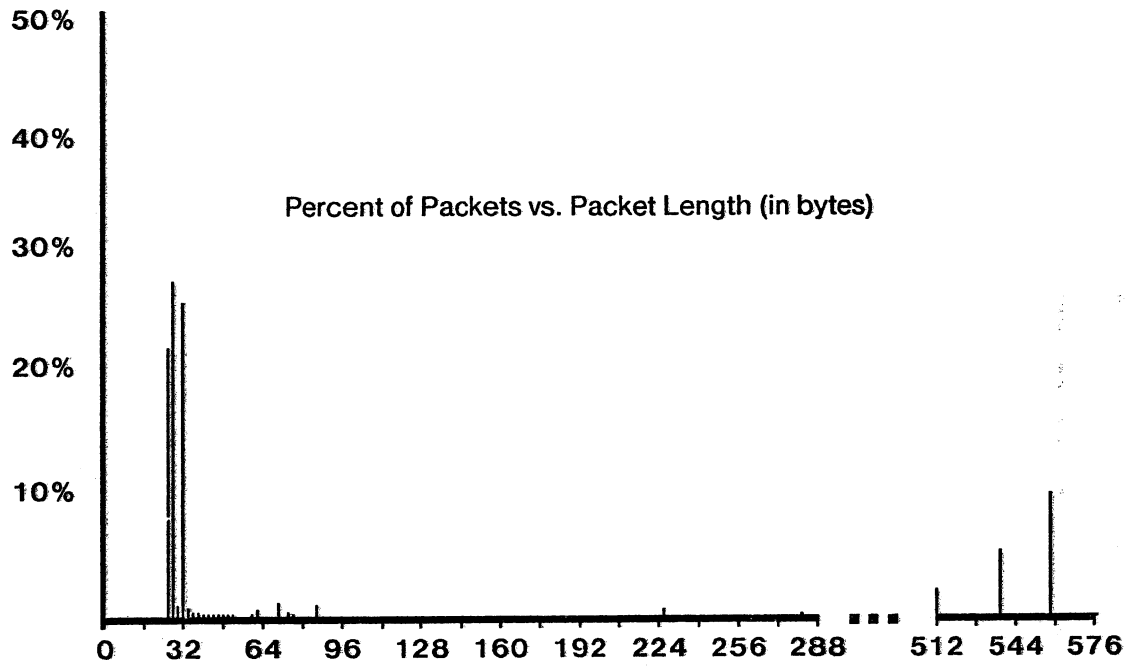


Figure 4: Distribution of packet lengths.

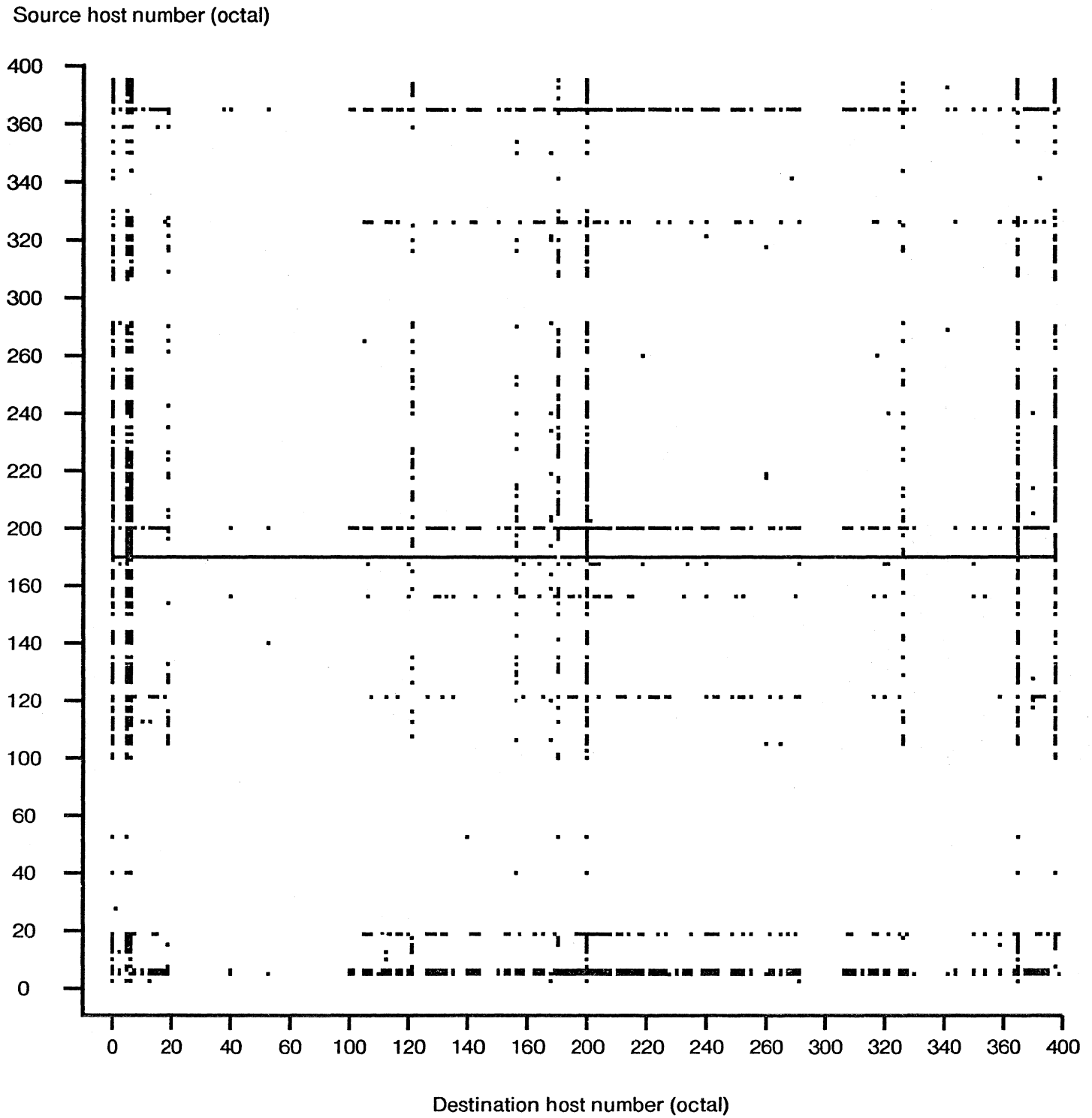


Figure 5: Source-destination traffic matrix.

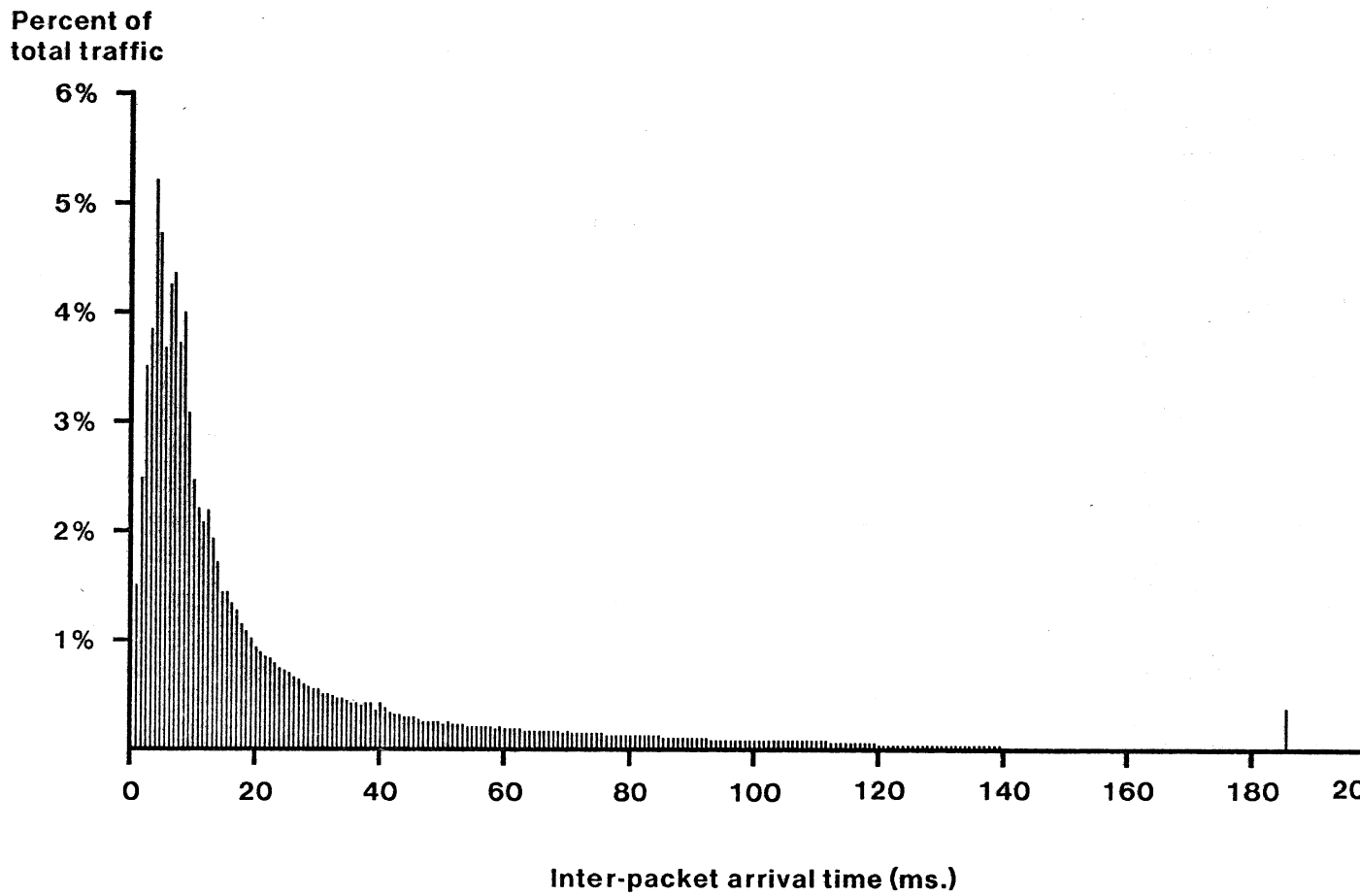


Figure 6: Distribution of inter-packet arrival times.

Percent of
total traffic

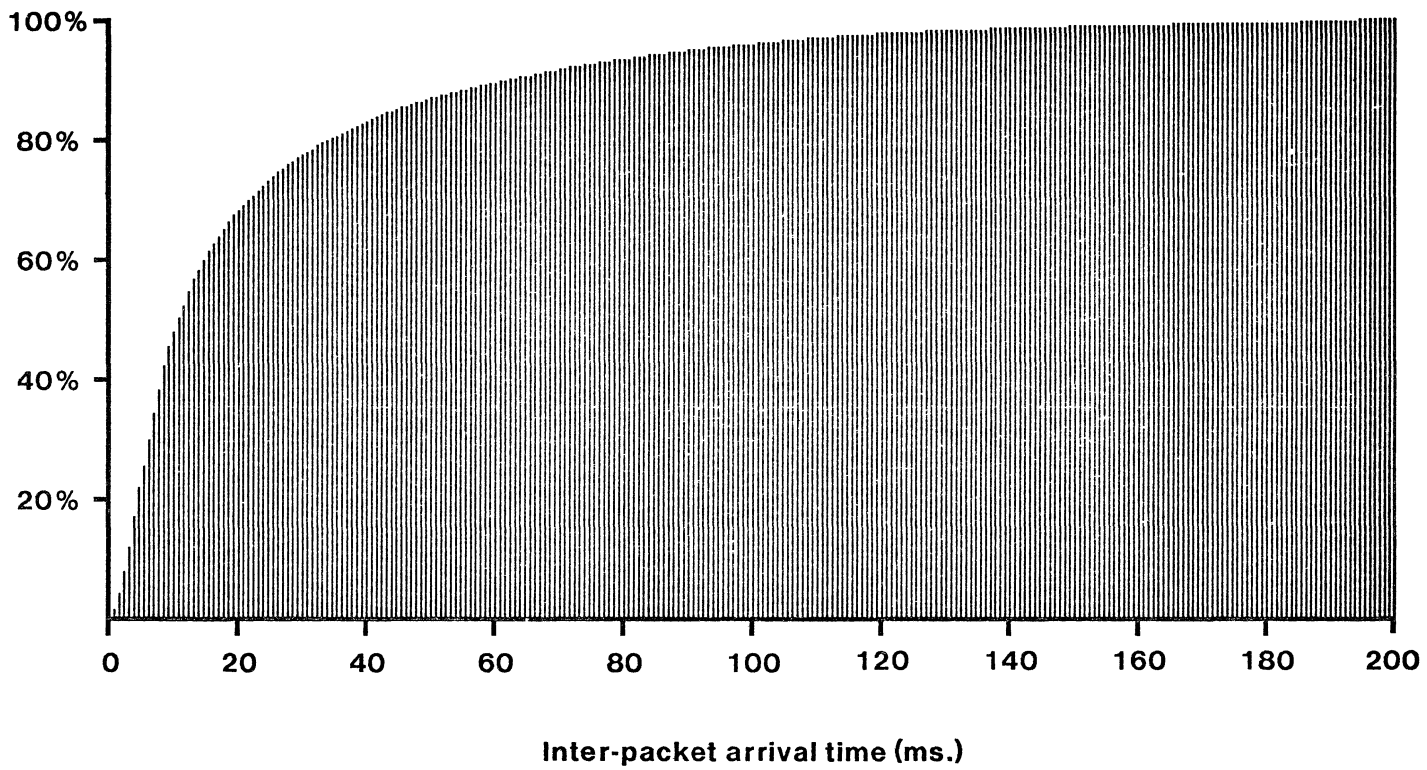


Figure 7: Cumulative inter-packet arrival times.

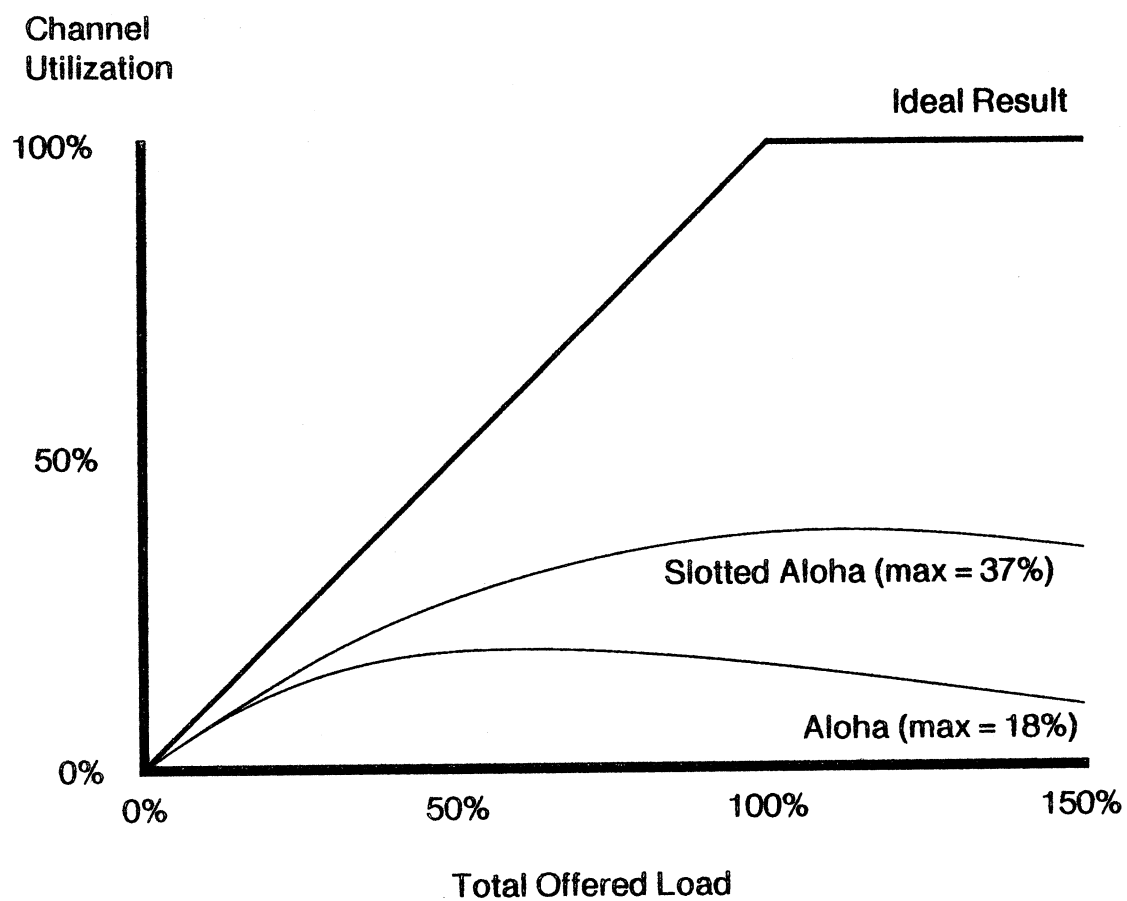


Figure 8: Utilization for several Aloha schemes.

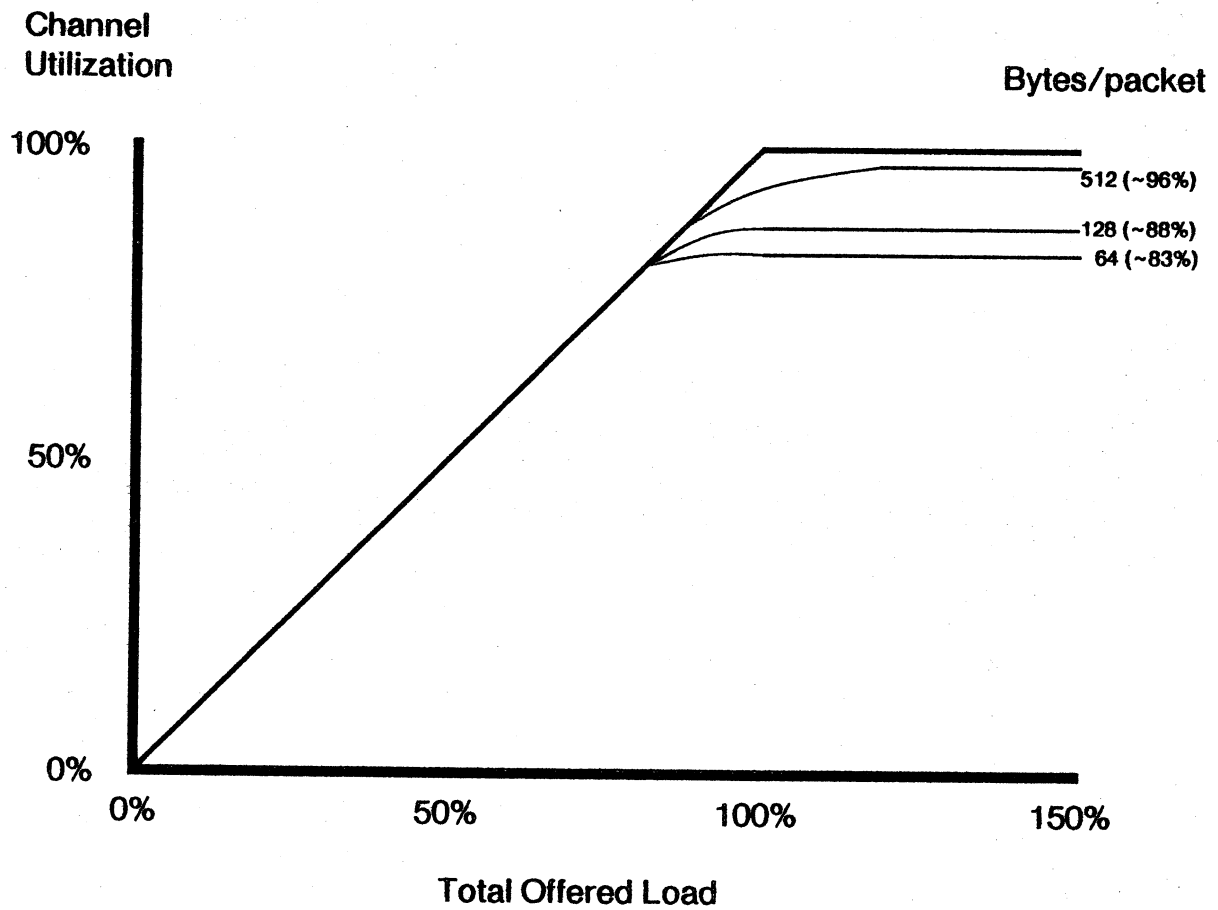


Figure 9: Measured utilization of the Ethernet network under high load.

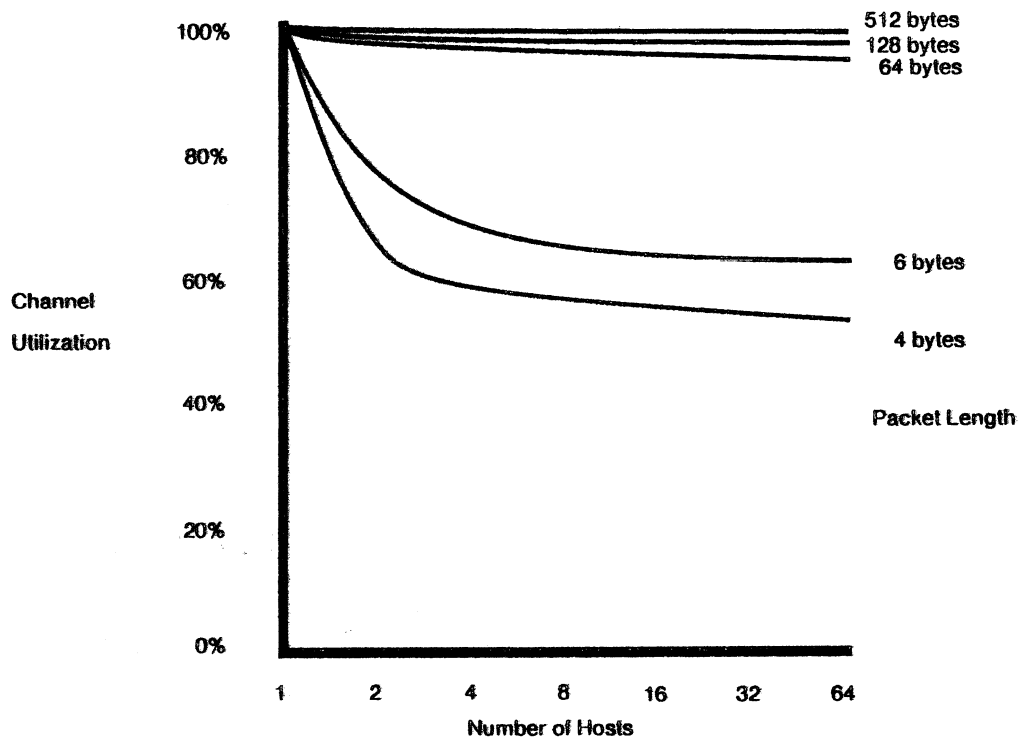


Figure 10a: Predicted utilization with continuously queued sources (recomputed from [Metcalfe & Boggs, 1976]).

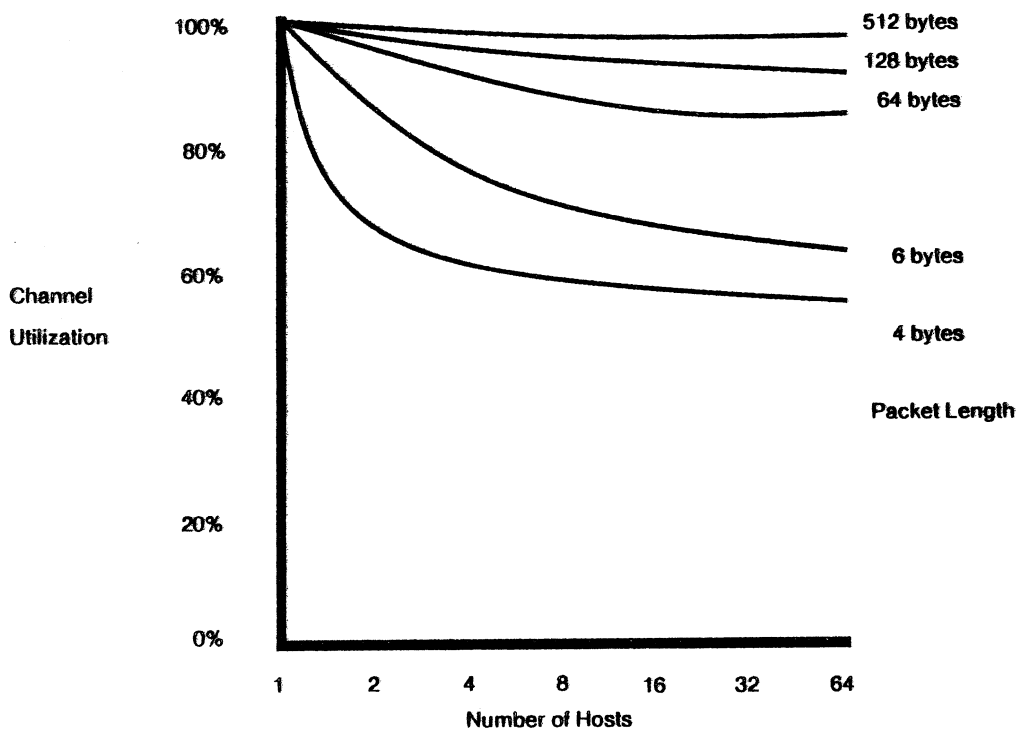


Figure 10b: Measured utilization with continuously queued sources.