Performance Characteristics of 2 Ethernets: an Experimental Study

Timothy A Gonsalves

Computer Systems Laboratory
Department of Electrical Engineering
Stanford University, Stanford, CA-94305

Abstract

Local computer networks are increasing in popularity for the interconnection of computers for a variety of applications. One such network that has been implemented on a large scale is the Ethernet. This paper describes an experimental performance evaluation of a 3 and a 10 Mb/s Ethernet. The effects of varying packet length and transmission speed on throughput, mean delay and delay distribution are quantified. The protocols are seen to be fair and stable. These measurements span the range from the region of high performance of the CSMA/CD protocol to the upper limits of its utility where performance is degraded. The measurements are compared to the predictions of existing analytical models. The correlation is found to range from good to poor, with more sophisticated models yielding better results than a simple one.

1. Introduction

In the past few years, local computer networks for the interconnection of computers and shared resources within a small area such as a building or a campus have rapidly increased in popularity. These networks support applications such as file transfers and electronic mail between autonomous computers, the shared use of large computers and expensive peripherals, and distributed computation. Typically, these networks have bandwidths in the range 0.1 - 10 Mb/s and span distances of 0.1 - 10 km. Local networks are also being interconnected to form multi-hop internets which allow communication over larger areas.

Many architectures have been proposed for local networks, and several have been implemented. The Ethernet [9] was one of the earliest to be implemented. Use of 3 Mb/s experimental Ethernets at several interconnected locations by a large community of users over several years and an experimental study of the performance of one under varying conditions have proved its merit for a variety of data transfer applications [11]. This has led to the introduction of a 10 Mb/s Ethernet as a commercial product [5].

Support and experimental facilities for this work were provided by the Xerox Palo Alto Research Centers. Additional support was provided by an IBM Graduate Fellowship, and by the Defense Advanced Research Projects Agency under Contract MDA 903-84-K-0249, monitored by the Office of Naval Research.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

This paper describes the results of experimental measurements of the performance of a 3 Mb/s Experimental Ethernet and a 10 Mb/s Ethernet. We present throughput, mean delay and delay distributions for the networks under artificially generated traffic conditions, making comparisons to the predictions of existing analytical models [8, 9, 10]. These results show the capabilities and limitations of the networks as well as the effects of increasing the channel speed. These experiments significantly increase the available base of data for the calibration and validation of analytical and simulation models of CSMA/CD networks.

In the next section we describe the principles of the Ethernet architecture and relevant details of the implementations used in this work, and we review prior work in this area. Section 3 describes the details of the experimental procedures. Results are presented and discussed in section 4. Section 5 is a summary of the paper.

2. Background and Previous Work

2.1. The Ethernet Architecture

The Ethernet [9] is an example of a packet-switched local network using a single shared bus or channel for communication among several hosts. The nature of the channel is such that only one packet can be transmitted at a time and every packet traverses the entire channel. This requires that all the hosts observe some protocol to gain control of the channel in an orderly and fair manner.

The Ethernet uses an access protocol called CSMA/CD. CSMA, or carrier sense multiple-access, is one of the distributed schemes proposed to efficiently utilize a broadcast channel [7]. In the CSMA scheme, a host desiring to transmit a packet waits until the channel is idle, i.e., there is no carrier, and then starts transmission. If no other hosts decide to transmit during the time taken for the first bit to propagate to the ends of the channel, the packet is successfully transmitted (assuming that there are no errors due to noise). However, due to the finite propagation delay, some other hosts may also sense the channel to be idle and decide to transmit. Thus, several packets may collide. To ensure reliable transmission, acknowledgements must be generated by higher level protocols. Unacknowledged packets must be retransmitted after some time.

In order to improve performance, the Ethernet protocol incorporates collision detection into the basic CSMA scheme, hence the abbreviation CSMA/CD. To detect collisions, a transmitting host monitors the channel while transmitting and aborts transmission if there is a collision. It then "jams" the channel for a short period to ensure that all other transmitting hosts abort transmission also. Each host then schedules its packet for retransmission after a random interval chosen according to some retransmission or back-off algorithm. The randomness is essential to avoid repeated collisions between the same set of hosts. The retransmission algorithm can

1Hereafter we use the term Ethernet to refer to both networks.
affect such characteristics as the stability of the network under high loads and the fairness to contending hosts.

### 2.2. A 3 Mb/s Ethernet Implementation

We summarize the relevant details of the 3 Mb/s Ethernet local network used in our experiments. This network has been described in detail earlier [4, 9]. The network used in our experiments has a channel about 550 metres long with baseband transmission at 2.94 Mb/s. The propagation delay in the interface circuitry is estimated to be about 0.25 μs [3]. Thus, the end-to-end propagation delay, \( T_p \) is thus about 3 μs. The retransmission algorithm implemented in the Ethernet hosts (for the most part, Alto minicomputers [13]) is an approximation to the binary exponential back-off algorithm. In the binary exponential back-off algorithm, the mean retransmission interval is doubled with each successive collision of the same packet.

Thus, there can be an arbitrarily large delay before a packet is transmitted, even if the network is not heavily loaded. To avoid this problem, the Ethernet hosts use a truncated binary exponential back-off algorithm. Each host maintains an estimate of the number of hosts attempting to gain control of the network in its load register. This is initialised to zero when the network is first scheduled for transmission. On each successive collision the estimated load is doubled by shifting the load register 1 bit left and setting the low-order bit to 1. This estimated load is used to determine the retransmission interval as follows. A random number, \( x \), is generated by ANDing the contents of the load register with the low-order 8 bits of the processor clock. Thus, \( x \) is approximately uniformly distributed between 0 and 2^n-1, where \( n \) is the number of successive collisions, and has a maximum value of 255, which is the maximum number of hosts on the network. The retransmission interval is then chosen to be \( x \) time units. The time unit should be no less than the round-trip end-to-end propagation delay. It is chosen to be 38.08 μs for reasons of convenience. After 16 successive collisions, the attempt to transmit the packet is abandoned and a status of load overflow is returned for the benefit of higher level software.

Thus, the truncated back-off algorithm differs from the binary exponential back-off algorithm in two respects. Firstly, the retransmission interval is limited to 9.7 ms (255 X 38.08 μs). Secondly, the host makes at most 16 attempts to transmit a packet.

### 2.3. A 10 Mb/s Ethernet Implementation

The 10-Mb/s Ethernet used is similar to the network described in the previous section with a few exceptions. The channel consists of three 500 metre segments connected in series by two repeaters. The end-to-end propagation delay, including delay in the electronics, is estimated to be about 30 μs (see pg. 52 in [5]). The truncated binary exponential back-off algorithm uses a time unit of 51.2 μs. The load estimate is doubled after each of the first 10 successive collisions. Thus the random number, \( x \), has a maximum value of 1023, yielding a maximum retransmission interval of 53.4 ms. A maximum of 16 transmission attempts are made for each packet.

### 2.4. Previous Work

The literature contains many theoretical and simulation studies of CSMA/CD networks with various traffic patterns [1, 8, 10, 14, 15]. However, there are few reported performance measurements of such networks. The first measurements were reported by Shoch [11] on the throughput of a 3 Mb/s Ethernet under normal and artificially generated data traffic. A concise treatment of part of that study appeared subsequently [12]. Gonivas [6] reported the measured measured delay-throughput characteristics of the same network under artificially generated voice traffic.\(^2\) Amer [2] describes measurement tools implemented on a 1 Mb/s Ethernet. Toenees [16] reported measurements on that network with up to 6 traffic generating stations.

Our experiments differ from those reported by Shoch and Hupp [12] in two respects. Firstly, we measure packet delays in addition to throughput and the other quantities measured in the earlier experiments. Delay is an important performance metric particularly for interactive applications. Secondly, we present results for a 10 Mb/s network as well as for the 3 Mb/s network reported upon in the earlier study.

### 3. Experimental Environment

In this section we describe the techniques used to measure the performance of the Ethernet under varying load conditions. First, we describe the experimental setup and procedures and then we characterize the traffic patterns and performance measures used.

#### 3.1. Experimental Setup and Procedures

To set up and run an experiment we use a special control program, running on a host on the network, to find idle hosts on the network and load a test program into each [12]. The controller is then used to set parameters describing the traffic pattern to be generated by the test programs. Next, the test programs are started simultaneously. They generate traffic and record statistics for the duration of the run. At the end of the run, the statistics are collected from the participating hosts by the control program.\(^3\) The duration of each run is typically 60 seconds. Our tests show that there is no significant variation in statistics for run times from 10 to 600 seconds.

Measurements on the 3 Mb/s Etbernet [11] and our informal observations of traffic on the 10 Mb/s Ethernet indicate that at night the normal load rarely exceeds a small fraction of 1% of the network capacity. Thus, it is possible to conduct controlled experiments with specific traffic patterns and loads on the networks during the late night hours.

#### 3.2. Traffic Patterns and Performance Metrics

Each of the N hosts is assumed to have one packet buffer. After completion of transmission of a packet, the host waits for a random period, \( T_{idle} \) before the next packet is queued for transmission in the buffer. The mean idle period and packet length, \( P \), and their distributions are set for each host for the duration of each run. Note that this corresponds to a central server queueing model, with the network being represented by the central server. The offered load of host \( i \), \( G_i \), is defined to be the fraction of \( C \), the network bandwidth, that the host uses if it is the only active transmitter on the network. Thus:

\[
G_i = \frac{T_p}{(T_p + T_{idle})}
\]

where \( T_p = P/C \) is the average transmission time of a packet. The total offered load, \( G \), is given by \( \sum_{i=1}^{N} G_i \). In our experiments we use a homogeneous population of hosts. Packet lengths are fixed and packet generation times are uniformly distributed random variables. We ignore packets lost due to collisions which the transmitter cannot detect ( [11], p. 72) and due to noise since these errors have been shown to be very infrequent [12]. That is, we assume that if a packet is successfully transmitted it is also successfully received.

Thus, all the participating hosts in an experiment are transmitters of packets. We compute throughput, \( \eta \), as the ratio of the entire packet, except for a 6-byte header and checksum\(^4\), is useful data. Thus, our results represent upper bounds on performance since many actual applications will include additional protocol information in each packet. Packet delay, \( D \), is defined to be the time from when the packet is generated, that is, the time at which it is queued for network

---

\(^2\) The Ethernet used in these two studies is the 3 Mb/s Ethernet on which this work was performed.

\(^3\) On the 10 Mb/s Ethernet, owing to differences in software, loading of the test program was done manually. The rest of the control was automated as described.

\(^4\) We assume 6 bytes of overhead in the case of the 10 Mb/s Ethernet.
4. Results

In this section we present the results of our experiments. First, we discuss the performance of the 3- and 10-Mb/s Ethernets separately. Next we make some comparisons between the two sets of experiments. Finally, we compare our measurements to the predictions of some existing analytical models. In all experiments, fixed length packets were used. The inter-arrival times of packets at each host were uniformly distributed random variables.

1.3 Mb/s Experimental Ethernet

Figs. 4.1 - 4.7 display various aspects of the performance of the 3 Mb/s Ethernet.

Utilisation: Fig. 4.1 shows the variation of total throughput, \( \eta \), with total offered load, \( \varphi \), for \( P = 64 \), 128 and 512 bytes. For \( \varphi \) less than 80-90%, virtually no collisions occur and \( \eta \) is equal to \( \varphi \). Thereafter, packets begin to experience collisions and \( \eta \) levels off to some value directly related to \( \varphi \) after reaching a peak, \( \varphi_{\text{max}} \). For short packets, \( P = 64 \), this maximum is about 80%, for longer packets, \( P = 512 \), it is above 95%. The network remains stable even under conditions of heavy overload owing to the load-regulation of the back-off algorithm. (These curves are similar to the ones obtained by Shoch and Hupp [12].)
Table 4.1 3 Mb/s Ethernet: Successfully transmitted packets as a fraction of total packets

<table>
<thead>
<tr>
<th>$P$ bytes</th>
<th>$G$ %</th>
<th>Successful Packets % Total Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>64</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>99.8</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>96.1</td>
</tr>
<tr>
<td>128</td>
<td>64</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>99.7</td>
</tr>
<tr>
<td></td>
<td>850</td>
<td>88.3</td>
</tr>
<tr>
<td>512</td>
<td>64</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>99.9</td>
</tr>
<tr>
<td></td>
<td>880</td>
<td>58.2</td>
</tr>
</tbody>
</table>

Successful Packets %

Delay: Fig. 4.2 shows the delay-throughput performance for the same set of packet lengths. In each case, for $G$ less than $G_{max}$, the delay is approximately equal to the packet transmission time, i.e., there is almost no queuing delay for access to the network. As the throughput approaches the maximum, the delay rises rapidly to several times the packet transmission time owing to collisions and the associated back-offs.

Figs. 4.3 - 4.5 show the histograms of the cumulative delay distributions for low, medium and high offered loads for $P = 64$, 128, 512 bytes. The delay bins are logarithmic. The labels on the X-axis indicate the upper limit of each bin. The left-most bin includes packets with delay $\leq 0.57$ ms, the next bin, packets with delay $\leq 1.18$ ms, and so on. The ordinate is the number of packets expressed as a percentage of all successfully transmitted packets. Table 4.1 gives the fraction of the total packets generated that were successfully transmitted.

We see that for $G = 64\%$, the delay of all packets is approximately the minimum, i.e., there is little queuing for network access. Even with $G = 100\%$ most packets suffer delays of less than 5 ms. Under heavy load conditions, however, only about 75% of the packets have delays of less than 5 ms, with the remainder suffering delays of up to 80 ms.

Fairness: To investigate the fairness of the protocol to contending hosts, we examine variations in performance metrics measured by individual hosts with increase in $G$. In Fig. 4.6 we plot the normalized mean of the individual throughputs vs. $G$ for $P = 64$ and 512 bytes. The vertical bars indicate the normalized standard deviation, i.e., the coefficient of variation. Also shown are the maximum and minimum individual throughputs. For low $G$, there is little variation in individual throughput. Under overload, the variation increases but remains less than $\pm 10\%$.

Fig. 4.7 contains similar plots for the mean delay per packet measured by each host. The variations with $G$ are seen to be similar to those in Fig. 4.6. Thus the protocol is seen to be fair to all contenders. (This has been noted in other experiments [16, 12]).

The metrics show slightly higher variation for $P = 512$ than for $P = 64$ bytes. During the successful transmission of a 512 byte packet a larger number of hosts are likely to queue for access to the network. Thus, at the end of the transmission all these hosts will attempt to transmit and will collide. The time for resolution of the collision is dependent on the number of colliding hosts and hence may be expected to be longer for $P = 512$ than for $P = 64$.

Discussion: The 3 Mb/s Ethernet is found to achieve high throughput for the range of packet sizes considered. The protocol is fair and stable under overload.

For $G < 100\%$, i.e., most practical situations, the delay is within a small multiple of the packet transmission time, $T_p$. Under such conditions, the network could support real-time traffic with delay constraints satisfactorily. However, at heavy load, the stability of the protocol is achieved at the expense of large delays, two orders of magnitude larger than $T_p$ for a fraction of the packets. The majority of the packets still suffer relatively minimal delays.

Our results do not contradict the analytical predictions [15] that CSMA/CD becomes unstable at sufficiently high offered loads. The analysis predicts that instability sets in when the probability of a packet transmission attempt in $2Xp$ reaches approximately unity. Given the maximum retransmission back-off of 0.7 ms and $r_p$ of 3 ms, this occurs when there are more than 1000 hosts continuously attempting to transmit, much larger than the number of hosts in our experiments. For a longer network, this number would be proportionately lower.
4.2. 10 Mb/s Ethernet

Figs. 4.8 - 4.14 display various aspects of the performance of the 10 Mb/s Ethernet. The results shown were obtained using 30 - 38 transmitting hosts. In all cases, fixed length packets were used and the inter-packet times were uniformly distributed random variables.

Utilization: Fig. 4.8 shows the throughput as a function of total offered load, G, for P ranging from 64 to 5000 bytes. The shape of the curves is similar to the corresponding curves for the 3 Mb/s Ethernet. However, maximum throughput varies from 25% for P = 64 bytes (the minimum allowed by the Ethernet specifications), to 80% for P = 1500 bytes (the maximum allowed), to 94% for very long packets of 5000 bytes. We note that for each curve, for G below the knee point, the throughput is approximately equal to G. Even under conditions of heavy overload, the network remains stable.

Delays: Fig. 4.9 shows the delay-throughput performance for the same set of packet lengths. Again, the curves have similar shapes to the curves for the 3 Mb/s network. For G below the knee points, the delay is minimal, whilst above the knee points, it rises sharply. The knees in this case are less pronounced, especially for larger P.

Figs. 4.10 - 4.12 show the histograms of the cumulative delay distributions for low, medium and high offered loads for P = 64, 512 and 1500 bytes. Table 4.2 gives the fraction of the total packets generated that were successfully transmitted. For low loads, delay is minimal for P = 512 and 1500 bytes. For P = 64, though, even at G = 19% delays range up to 10XT. At high loads, for all packet lengths, the majority of packets suffer moderately increased delays,
while a fraction suffer very high delays, up to 0.5 s, for $P = 512$ and
1500, about 75% or all packets suffer delays $\leq 10 T_p$. For $P = 64$,
75% of packets suffer delays $\leq 15 T_p$.

We examine variations in performance metrics measured by individual hosts with increase in $G$. In Fig. 4.13 and 4.14 we plot the normalized means of the individual throughput and delays respectively vs. $G$ for $P = 64$ and 512 bytes. The vertical bars indicate the normalized standard deviation, i.e., the coefficient of variation. Also shown are the maximum and minimum individual throughputs.

The metrics show higher variation than in the 3 Mb/s case, in the range $\pm 35\%$. This may be attributed to the larger retransmission periods in the 10 Mb/s back-off algorithm (ref. Sections 2.2 and 2.3). Also, the dependence on $G$ is less marked. Contrary to the 3 Mb/s case, the variation is slightly lower here for the larger packets size.

With short packets, e.g. 64 bytes, $T_p$ becomes comparable to $T_p$ and hence the maximum throughput is low. With long packets, we see the high throughputs achieved with the 3 Mb/s net. Thus, packet lengths of the order of 64 bytes on a 10 Mb/s net approach the limit of utility of the Ethernet protocol. This does not imply that such short packets should not be used. A study of traffic on a typical local computer network shows that approximately 20% of the total traffic is composed of short packets, whilst the remainder of 80% consists of long packets [12]. The 10 Mb/s Ethernet studied could support a high throughput with such a traffic mix although it cannot do so with only short packets.

Packet delay is minimal for low to moderate loads. However, for short packets the delay increases even at fairly moderate $G$. At higher $G$, a fraction of packets suffer delays ranging from the minimum up to 0.5 s.

For the loads used the network is fair and stable. Instability predicted by analytic studies will occur with about 1500 hosts continuously attempting to transmit.

4.3. Comparison of the 3 and 10 Mb/s Ethernets

In this section we examine the effect of the difference in bandwidth between the two Ethernets on performance. The differences in some important parameters, such as network length, in the two cases should be borne in mind.

Utilisation: Fig. 4.15 shows the throughput, $\eta$, as a function of total offered load, $G$, for several packet lengths for the 2 networks. For $P = 64$ bytes, the throughputs of the two networks are almost equal. For longer packets, the 10 Mb/s network exhibits substantially higher throughput. The throughput increases less than linearly with increase in bandwidth. This is shown in Table 4.3 in which the ratio of the absolute throughput at 10 Mb/s to that at 3 Mb/s is given for several values of $P$.

Table 4.3
<table>
<thead>
<tr>
<th>$P$, bytes</th>
<th>$G$</th>
<th>Successful Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$%$</td>
<td></td>
<td>Total Packets</td>
</tr>
<tr>
<td>64</td>
<td>19</td>
<td>100.0</td>
</tr>
<tr>
<td>38</td>
<td>100.0</td>
<td></td>
</tr>
<tr>
<td>1900</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>30</td>
<td>100.0</td>
</tr>
<tr>
<td>90</td>
<td>99.9</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>98.7</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>30</td>
<td>100.0</td>
</tr>
<tr>
<td>90</td>
<td>99.7</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>93.3</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2 10 Mb/s Ethernet: Successfully transmitted packets as a fraction of total packets

P bytes G Successful Packets % Total Packets
---|--|--|--|
64 | 19 | 100.0 |
38 | 100.0|
1900 | 99.9|
512 | 30 | 100.0 |
90 | 99.9 |
300 | 98.7|
1500 | 30 | 100.0 |
90 | 99.7 |
300 | 93.3|

Table 4.3 Increase in $\eta$ with increase in $C$ from 3 to 10 Mb/s

<table>
<thead>
<tr>
<th>$P$, bytes</th>
<th>$\eta_{10}/\eta_{3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>1.05</td>
</tr>
<tr>
<td>512</td>
<td>2.45</td>
</tr>
<tr>
<td>1500</td>
<td>2.90</td>
</tr>
</tbody>
</table>

Table 4.3 Increase in $\eta$ with increase in $C$ from 3 to 10 Mb/s
4.4. Comparison with Analytical Predictions

We consider three analytical models: a simple model for computing maximum throughput [9], and two more sophisticated models for computing maximum throughput and delay-throughput characteristics [8, 15]. All the models assume that the channel is slotted, with slot-time related to the end-to-end propagation delay. Performance is expressed in terms of a parameter, \( \alpha \), equal to the ratio of the slot-time to the transmission time of a packet, i.e.,

\[
\alpha = \frac{r_s}{(P/C)}
\]

Further details of the analyses may be found in the papers cited.

First we consider the maximum throughput. From [9] we compute \( \eta \) with the number of hosts, \( N \), equal to 32 to correspond to our measurements (\( \eta \) does not vary much with \( N \)). We use the infinite population analysis of [15] to obtain \( \eta_{\text{max}} \) from the formula for \( \eta \) as a function of \( G \). Table 4.4 shows measured and computed values of maximum throughput for various values of \( P \) for the 3 Mb/s Ethernet. \( r_s \) is estimated to be 3 \( \mu \)s (see Section 2.2). Table 4.5 and 4.6 show corresponding sets of values for the 10 Mb/s Ethernet. The measured
Table 4.5 10 Mb/s Ethernet: Maximum Throughput, %

<table>
<thead>
<tr>
<th>P bytes</th>
<th>a</th>
<th>Measured</th>
<th>Metcalfe &amp; Boggs</th>
<th>Tobagi &amp; Hunt</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.22</td>
<td>26</td>
<td>55</td>
<td>51</td>
</tr>
<tr>
<td>200</td>
<td>0.072</td>
<td>62</td>
<td>79</td>
<td>72</td>
</tr>
<tr>
<td>512</td>
<td>0.028</td>
<td>72</td>
<td>91</td>
<td>87</td>
</tr>
<tr>
<td>1500</td>
<td>0.0098</td>
<td>86</td>
<td>97</td>
<td>96</td>
</tr>
<tr>
<td>5000</td>
<td>0.0029</td>
<td>95</td>
<td>99</td>
<td>99</td>
</tr>
</tbody>
</table>

Table 4.6 10 Mb/s Ethernet: Maximum Throughput, %

<table>
<thead>
<tr>
<th>P bytes</th>
<th>a</th>
<th>Measured</th>
<th>Metcalfe &amp; Boggs</th>
<th>Tobagi &amp; Hunt</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>0.28</td>
<td>26</td>
<td>49</td>
<td>46</td>
</tr>
<tr>
<td>200</td>
<td>0.092</td>
<td>60</td>
<td>75</td>
<td>68</td>
</tr>
<tr>
<td>512</td>
<td>0.036</td>
<td>72</td>
<td>88</td>
<td>83</td>
</tr>
<tr>
<td>1500</td>
<td>0.012</td>
<td>85</td>
<td>96</td>
<td>94</td>
</tr>
<tr>
<td>5000</td>
<td>0.0037</td>
<td>94</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>10000</td>
<td>0.0019</td>
<td>97</td>
<td>99</td>
<td>99</td>
</tr>
</tbody>
</table>

It is seen that both the models overestimate \( \eta_{\text{max}} \), the formula of Metcalfe & Boggs being less accurate than that of Tobagi & Hunt. The error is larger at smaller values of \( P \), i.e., at larger values of \( a \). Some factors affecting the accuracy of the analytical predictions are the accuracy of the estimation of \( r_p \) and the various assumptions made in the analyses. In particular, the assumption that transmission attempts are made only at the beginning of slots would lead to higher predicted throughputs. On the other hand, the assumption that every pair of hosts is separated by \( r_p \) would lower the predicted throughput. A third differences between the experimental and analytical models include the retransmission algorithm: Metcalfe & Boggs assume an optimum policy, Tobagi & Hunt assume an infinite retransmission delay; and the packet arrival process: the analyses assume Poisson arrivals, the measurements were made with uniformly distributed arrivals.

In Fig. 4.17, measured and predicted (\( \eta_0 \), Fig. 2) delay-throughput characteristics are plotted for \( a = 0.1 \) and 0.01. The measured curves for the 10 Mb/s Ethernet with \( a = 0.092 \) and 0.012 correspond to \( P = 200 \) and 1500 bytes respectively. For the 3 Mb/s Ethernet curve, \( P = 128 \) bytes.\(^7\) Considering \( a = 0.1 \), at high \( \eta \) the predicted and measured curves correspond closely. At low \( \eta \), however, the measured delay is much higher than predicted. This is due in part to the software overhead in the measurements - delays of a fraction of a millisecond could not be measured. Looking at \( a = 0.01 \), we note again the larger measured delays at low \( \eta \). At high \( \eta \), in the case of the 3 Mb/s Ethernet, the prediction varies from being optimistic to approximate correspondence. In the case of the 10 Mb/s Ethernet, however, the analysis consistently underestimates delay. Possible reasons for these discrepancies have been discussed above.

In summary, the simplistic formula of Metcalfe & Boggs overestimates the maximum throughput. The error is greatest for large \( a \). This, though, is precisely the region in which accurate estimation of performance is most important because network

\(^6\) As computed using the total packet length, including overhead. Throughputs shown are net, excluding overhead.

\(^7\) From the numerical results in [8] we note that the percentage changes in predicted values for small \( \Delta a \) is less than \( \Delta a / a \). Hence, we can make comparisons between measured curves with \( a = 0.012 \) and \( 0.008 \) and a predicted curve with \( a = 0.01 \), and between a measured curve with \( a = 0.002 \) and a predicted curve with \( a = 0.1 \).
performance in this region is poor. The more sophisticated analysis of Tobagi & Hunt yields more accurate though still optimistic predictions. Lam's delay-throughput predictions are seen to vary from approximately correct to optimistic. The accuracy is not consistent for a given network or a.

5. Conclusions

We have presented results of performance measurements on 3 and 10 Mb/s Ethernet with various packet lengths and offered loads ranging from a small fraction of network bandwidth to heavy overload. These experiments span the range from the region of high performance of CSMA/CD networks to the limits at which performance begins to degrade seriously. The former occurs when the packet transmission time is large compared to the round-trip propagation delay. The latter occurs when the two times are comparable in magnitude.

Packet delay is seen to be minimal over normal operating ranges of offered load. However, at saturation, the delay for some packets increases by orders of magnitude although the increase in average delay in much lower. Saturation occurs at lower values of offered load when the packet transmission time is comparable to the round-trip propagation delay. The protocol is fair to all contenders, with the 10 Mb/s Ethernet exhibiting somewhat higher variation in performance measured by individual hosts than the 3 Mb/s Ethernet.

Comparison of our measurements with the predictions of several analytical models indicate that the predictions tend to be optimistic, the more complex analyses yielding more accurate predictions than a simple model. The accuracy of the predictions is affected by the estimation of the propagation delay across the network.

6. Acknowledgements

Thanks are due to several people for their contributions to this work. John Shoch made available the measurement tools developed by himself and Jon Hupp for the 3 Mb/s Ethernet. Alfa Freier and Hal Murray provided assistance in the development of measurement tools for the 10 Mb/s Ethernet. Forrest Baskett, Yogen Dalal and Larry Garlick provided advice at various stages of the experiments. Fouad Tobagi's technical and editorial comments were helpful.

References