Proceedings of the 1985 Winter Simulation Conference
D. Gantz, G. Blais, S. Solomon (eds.)

PERFORMANCE EVALUATION OF ETHERNET AND HYPERBUS LOCAL AREA NETWORKS USING COMPUTER MODELING

David S. Jennings and Aaron H. Konstam
Trinity University
San Antonio, TX 78284

ABSTRACT

The local area networks Ethernet and HYPERbus were simulated using the GPSS program language. Measurements of performance were network stability, messages transmitted per unit time, and the number of transmission attempts required per message. The simulations produced these results. Both networks were stable at the normal, 90 and 100 percent loads. At the 90 and 100 percent loads, Ethernet transmitted between 90.31 and 99.16 percent of the expected number of messages per unit time. HYPERbus under the same conditions transmitted 97.52 and 103.87 percent of the expected number of messages per unit time. Ethernet required up to 13 transmission attempts for some messages, with an average of 2.061 attempts per message. HYPERbus transmitted all of its messages on the first attempt.

INTRODUCTION

Local area networks are becoming the most common way of connecting a series of workstations together in order to allow them to communicate with each other and share common resources. Therefore it is becoming more and more important for installers of these systems to be able to evaluate the ability of competing local area networks using different communication protocols to handle the desired message loads.

The performance of the local area networks (LAN) Ethernet and HYPERbus are investigated in this study. Each LAN uses a different message transmission control protocol. The differences between these protocols, and how their operation affects the performance of their respective networks were investigated. Computer modeling was used to simulate the operation of these networks. The simulation programs gather statistics on three measures of performance: network stability, messages transmitted per unit time, and number of attempts required for successful transmission of each packet.

The network's stability is the relationship between the minimum message processing time and the actual processing time. If the amount of time required to process a message approaches infinity, then the stability of the system is compromised [1].

The number of messages a network can transmit during a given amount of time is a measure of the network's message capacity.

The number of times a message must be transmitted before being received at its destination is the one statistic that cannot be calculated from the current published research on these two LANs. The measure of attempts required can only be gathered by observing an actual system or through system simulation.

System simulation is used in this project for three reasons. First, there are no published methods to predict accurately the expected packet collision distribution for a specific load on either of these two networks. Simulation provides a technique to "transmit" packets, count the collisions and gather statistics on the retransmissions.

Second, there are no published studies of the Networks Systems Corporation's HYPERbus system outside of its corporate laboratories. Study and simulation of this network affords the opportunity to gain insight into the design, operation, and performance capabilities of this system.

Finally, simulation offers a convenient method for observing network stability under various loads.

The access technique used by a local area network best describes the operation of that network [1]. Both Ethernet and HYPERbus use a variation of contention based protocols to direct channel access.

The success of a contention network depends almost entirely on the access algorithm's ability to (1) detect that the network is not busy and allow a transmission, or (2) detect the network is busy and delay a transmission, and (3) resolve contention on the network when it occurs. Through careful design, contention schemes can utilize the bus very efficiently [1]. The two contention scheme variations used by Ethernet and HYPERbus are Carrier Sense Multiple Access with Collision Detection, and Carrier Sense Multiple Access with Collision Avoidance respectively.

The simulations in this study were all written in the GPSS language.

PRIOR LOCAL AREA NETWORK STUDIES

Ethernet was developed by Xerox Corporation and formally introduced by Metcalf and Boggs [2] in their 1976 paper. The network design stems from the Aloha Network's packet collision and retransmission algorithm for contention resolution [2]. The design was also founded on the premise that traffic patterns on the Ethernet would be "bursty", meaning that data packets occur rarely and with a sporadic frequency [3]. An actual system is unlikely to be driven at 90 or 100 per cent of its capability [3]. Because packets normally occur at such great intervals in time compared to the operating speed of the network, the access protocol uses a "best effort" approach to packet transmission.

The system provides its best effort to transmit the packet on the first attempt without suffering a collision. During normal operations the packet will transmit successfully on the first attempt. If it does fail, the system is provided with mechanisms to retransmit the packet [2]. Shoch and Hupp concluded, in 1980, in a measured performance study of an Ethernet system, that: under normal conditions the error rate is very low and very few packets ever collide with each other; under normal conditions ready nodes rarely have to defer to passing transmissions; under high loads more collisions occur, the collisions are successfully retransmitted by the collision resolution mechanisms, and overall channel utilization remains very high; and the channel is fairly shared by all stations [3].

Shoch and Hupp also discovered that network utilization was linear as the load increased from 0 to 90 per cent of the theoretical maximum load. As the load increased beyond 90 to 100 per cent of the maximum, utilization flattened out at a level just above 96 per cent [3].

The only available prior study of the HYPERbus local network is a Network Systems Corporation internal study performed by Larry Swan [4].

Swan concluded that the critical factor affecting network efficiency was data packet length. As the packet size becomes larger the overall efficiency of the system increases because the amount of time spent transmitting the packet becomes larger in relation to the fixed delay period of the time slot. When the packets are smaller the overall efficiency drops but the number of packets that can be transmitted per unit time increases.

PRIOR SIMULATION STUDIES OF ETHERNET AND HYPERBUS

Ethernet has been previously modeled in several studies. Sauer and MacNair simulated the operation of a LAN which used the CSMA/CD access protocol like Ethernet's in their book Simulation of Computer Communication Systems. They determined that the protocol worked well at distributing the use of the bus and that the system would suffer few collisions under normal loads [5].

Another study was performed at Digital Equipment Corporation by Marathe and Hawe to investigate the usefulness of employing an Ethernet as a timesharing network for a large university. This study concluded that an Ethernet was more than adequate to provide the level of support needed in a timesharing system [6].

There are no available prior simulation studies of HYPERbus.

ACCESS PROTOCOLS

The acquisition protocol used by Ethernet is Carrier Sense Multiple Access with Collision Detection or CSMA/CD. The transmission sequence begins with the interface board signaling its transceiver that it has a ready packet. Regardless of whether it has a ready packet or not, a node's transceiver is always sensing or "listening" to the cable for the carrier signal of a passing data packet. When the ready transceiver does not detect the carrier of a passing data packet it assumes that the cable is not currently busy and initiates transmission. If no other nodes begin transmitting during the amount of time it takes for the the first bit of the current transmission to propagate to the maximum length of the cable, the current transmission is said to have acquired the cable and it will transmit successfully. No node will begin transmitting while hearing the carrier signal of a passing transmission [2].

In the event a node has a packet ready to transmit and the transceiver detects the carrier signal of a passing data packet, the ready node will defer transmitting its packet until the current packet has gone by, and an additional amount of time called the interpacket gap has expired. This gap can vary from 9.6 to 10.6 microseconds [7].

If two nodes begin transmission within a period of time less than the end-to-end propagation delay time of the system, then those two transmissions will interfere with each other and are said to have suffered a collision. When the two nodes which are colliding detect their collision, they continue transmitting for a short period of time, from 3.2 to 4.8 microseconds, which will guarantee that all other nodes will sense the collision. A 10 MHz squarewave signal is generated by the "colliding" transceivers and is recognized by the other nodes as the collision signal [7].

The two nodes will then wait for a random amount of time before retransmitting the data packets. The wait time is a integral multiple of the slot time of the Ethernet system. The slot time is 51.2 microseconds [7]. The multiplier is a random integer, k, in the range 0 ¼ k ¼ 2^n, where n is minimum (x,10) and x is the most recent number of attempts to transmit that packet.

The method of allocating transmission space on HYPERbus bus cable is called Carrier Sense Multiple Access with Collision Avoidance, or CSMA/CA. Design goals of the protocol are: to transmit data packets at high speeds on the bus; and to avoid contention and associated collisions on the bus, keeping expected error rate near zero [8].

For the protocol to operate correctly each station on the network has a sequence number assigned to it during the system installation. One end of the bus is arbitrarily chosen as the end where the lowest sequence number station is located. The value of each sequence number is based on the total footage of cable from the previous station's interface unit to the next station's unit. The sequence number is coded into the contention switches of the bus jack unit for each station. The transmission window of each station exists because of the sequence number. The number will correspond to a memory location in a group of memory locations used as the contention resolution timer within each interface unit. During a contention time check, no station may begin transmitting until its time slot has been reached.

Each contention timer consist of 4095 memory locations. The timer value depends on the access time of the memory and the number or memory locations used (which in turn depends on the length of the system in feet). These locations are currently accessed at a speed of 512 nanoseconds, which results in the timer having a maximum value of 2.097 milliseconds [8]. On smaller (shorter) systems the timer value would be smaller.

The access protocol (CSMA/CA) operates in the following manner. When the system is first turned on after installation, all stations are quiet. The access protocol is waiting for the first station to attempt a transmission. This element of HYPERbus theory is linked to carrier sense. When a interface unit receives a ready packet from a processor it checks to see if the bus is busy. Since this is the first transmission on the system,the bus status bit is reset. The bus is clear and the interface unit starts its contention resolution timer. When the contention timer equals the slot time, that node begins transmitting. If no other nodes became ready to transmit during the the most recent transmission, the bus becomes quiet again and awaits the next transmission; if another node becomes ready to transmit during a current transmission it too will sense the cable. It will find the cable busy and will not start its contention timer. Instead, it will loop at this point in the algorithm, until the bus is sensed as not busy. When the bus is clear the contention timer is started and the interface waits for the timer to equal its slot time before starting transmission. The slot times are spaced to ensure that no collisions occur between nodes which became ready to transmit while the bus was busy.

SIMULATION PROGRAM DESCRIPTIONS

ASSUMPTIONS ABOUT THE MODELS

1. All packets arrive error free when ready to transmit. Packet error is negligible.

2. Packets arrive in the system according to the Poisson arrival pattern.

3. The network configuration adheres to the following limits:

   A. Total coaxial cable length is 500 meters.
   B. Number of stations: Normal loads = 120, High loads = 10

4. Propagation time equals 18.55 nanoseconds/meter [7,9].

For the HYPERbus simulation, the following assumption applies:

5. Contention time factors for the normal and 100% load are 289 and 219.9 respectively.

For the Ethernet simulation, four parameters affect the performance of the system. These are: (1) interarrival time of transactions in microseconds, (2) number of nodes on network, (3) distance between nodes, in meters, and (4) data packet size, in bytes.

Four parameters affect the operation of the HYPERbus model. They are: (1) interarrival time, (2) number of nodes, (3) contention time setting, and (4) packet length.

NORMAL LOADS

The value of these parameters for the normal load were derived from a study performed by Shoch and Hupp on an operational Ethernet system at Xerox's Palo Alto Research Center [3]. In their study of measured performance of Ethernet, they observed that during a normal day the average interarrival time of packets to the system was every 39.5 milliseconds. The overall length of the system was 550 meters and had 120 connected stations. The overall length includes 50 meters of transceiver drop cable at the two stations on each end of the coaxial cable. The coaxial cable length is 500 meters.

The data packets length distribution reflected that about 82 per cent of the data packets were only 32 bytes long and the remaining 18 per cent were about 558 bytes long.

For the performance simulation, interarrival time of each packet was generated using the Poisson arrival pattern with 39500 microseconds as the mean value of the function. There were 120 nodes on the system, each attached at approximately 4 meter intervals. The data packet length was assigned to each transaction according a discrete distribution function assigning 82 per cent as 32 bytes and 18 per cent as 558 bytes. Here is a summary of normal load parameters for both systems:

Packet Length	82 % @ 32 bytes, 18 % @ 558 bytes
Number of Nodes	120
Distance between Nodes	4 meters
Mean Interarrival Time	39500 microseconds
HYPERbus only	
Contention time factor	289 microseconds
Priority Range Fact	540

100 PERCENT LOAD

ETHERNET The parameter values for the Ethernet simulation under 100 per cent load were calculated as follows.

Packet Length: Since the study performed by Shock and Hupp [3] the data packet size has been updated so that the minimum packet length is 72 bytes and the upper bound is 1526 bytes. The packet lengths for the performance runs under 100 per cent loads will be 72, 256, 512, 1526 bytes accordingly.

Number of Nodes and System Length: The system length is again 550 meters (500 meters of coaxial cable and 50 meters of drop cable) to keep the simulation limited to a single cable segment format. For the sake of simplicity the number of nodes used under high loads was set at 10. This provided an average inter-station separation of 55 meters.

Interarrival Times: The interarrival (IA) time of each transaction to the system is a function of how many transmissions are possible on the cable per second at a given packet length. To determine the IA time of packets to the system for a specific load, it is first necessary to calculate the time required to send one packet through the system.

The transmission speed of Ethernet is 10 million bits per second (10 Mbps). The network acquisition time for a single segment, 500 meter, network is approximately 10.8 [10] microseconds and the time used in the simulation for inter-frame spacing is 9.6 microseconds. The following equation will calculate the transmission time in microseconds required for any length data packet for a 500 meter Ethernet system.

$$T\_send = (\text{Packet length} * 0.8) + 10.8$$

where 0.8 is the factor for microseconds/ byte

This next equation allows the calculation of the interarrival time of data packets in a 500 meter system at a load factor of x:

T_ia = [(P * 0.8) + 10.8 + 9.6] / x

where

P = Packet length in bytes
x = load factor   0.0 -½ 1.0

This equation will calculate the minimum interarrival time of data packets, each of a common length, to an Ethernet so that no collisions will occur. Here is a summary of the high load parameter values:

Ethernet 100 Per Cent Load Parameters

Packet Length	Number of Nodes	Distance between Nodes	Mean Inter-arrival Times
72	10	55	78.00
256	10	55	225.22
512	10	55	430.00
1526	10	55	1241.20

HYPERbus The following equation allows the calculation of the IA time parameter values for the HYPERbus simulation [4].

T_ia = [(P * 0.8) + C_tm + 1110]/x

where P = packet size and C_tm = contention time factor

For a HYPERbus network with a Priority Range Factor Setting of 540 and an overall length less than 5000 feet the C_tm setting is 219.9 [4]. With this variable set it was then possible to calculate the following parameter values.

Packet Size	Mean IA Time	Number of stations	Contention time setting
72	1387.50	10	219.9
256	1534.70	10	219.9
512	1739.50	10	219.9
1526	2550.70	10	219.9
2048	2968.30	10	219.9
4096	4606.70	10	219.9

SIMULATION RESULTS

NORMAL LOAD RESULTS

Under normal operating conditions, the message traffic load on each of the networks is a very small percentage of their message capacity. At this level, the measure of messages per unit time becomes impossible to calculate, because the equation to predict the theoretical number of messages relies on all the messages being the same size. The normal load programs process packets whose lengths vary according to a probability distribution.

The systems should not suffer any instability at normal load levels, because messages are not arriving fast enough to tax the networks. For the most part, the systems were simulated at this load to help establish the validity of the models.

According to Shoch and Hupp's study, the normal load of an Ethernet is very low [3]. Over a 24-hour period the measured utilization presented a very modest usage of the network (i.e. between 0.60 and 0.84 per cent of the full capacity). At this modest, load collisions hardly ever occur and rarely will will a message have to defer before transmitting. The model runs produced results which are in line with this observed behavior.

All transmissions for both models were completed on the first attempt. On Ethernet there were no collisions and only 2 of the 500 transmissions had to defer. Also, of interest is the rather surprising measure of the amount of time required to simulate Ethernet at this load. It took 19.93 seconds of simulated real time for all these transactions to arrive and be transmitted. Shoch and Hupp observed that in a 24-hour period, a total of 2.2 million messages were sent on the Ethernet [3]. This works out to an average of 25.46 packets/sec. The simulation, using the Poisson arrival pattern with a mean of 39.5 milliseconds (86.4 * 10°3 /2.2 * 10°6), produced a packet per second rate of 25.08. The simulation operates within 98.51 per cent of the measured system using the Poisson arrival pattern, adding confidence to the arrival distribution scheme used and the model in general.

No collisions were observed in the HYPERbus model, and only 20 of the 500 messages had to wait for their time slot before transmitting, meaning that 96.0 per cent of the packets were transmitted without delay.

The model transmitted an average of 24.24 packets per second, or 95.21 per cent of the expected packet/second rate.

SIMULATED HIGH LOAD OPERATIONS

It is at the higher loads that the performance measures become more meaningful. As the offered load increases toward the theoretical maximum the two models should continue to produce reasonable performance values.

Stability: Neither Ethernet or HYPERbus suffered instability at the 100 per cent load, as no nodes had to wait an unduly long time to transmit. Tables 1 and 2 reflect the overall stability of the networks.

Messages per unit Time: The Ethernet model exhibits performance which is very close to the theoretical values. Table 3 indicates that the Ethernet model transmits between 90.31 and 98.98 per cent of the expected number of messages per unit time. HYPERbus results for the same measure were between 97.52 and 98.91 per cent (Table 4).

Transmission Attempts: As activity on the network increases it is expected that more packets will collide and be retransmitted. Some of these packets will require several retransmission attempts before being successfully transmitted. Tables 5 and 6 show the distribution for the attempts required by the 500 transactions for the 100 per cent load for each network.

As the data packets became longer, the number of packets successfully transmitted by Ethernet on the first attempt became higher. Fewer collisions occurred on the first transmission attempt because the collision window represents a smaller percentage of the mean interarrival time for larger data packets. Under the 100 per cent load, less than half of the messages are transmitted successfully on their first attempt. Most of the colliding packets require

Table 1: Ethernet Minimum and Mean Transmission Completion Times as a Function of Packet Size at 100 % of Maximum Load

Packet Size	Min Completion Time	Mean Completion Time
72	68.4	909.869
256	215.6	841.157
512	420.4	1089.373
1526	1231.6	2659.733

Table 2: HYPERbus Minimum and Mean Transmission Completion Times as a Function of Packet Size at 100 % of Maximum Load

Packet Size	Min Completion Time	Mean Completion Time
72	1387.5	35695.101
256	1534.7	39440.460
512	1739.5	44647.699
1526	2550.7	65372.386
2048	2968.3	76027.375
4096	4606.7	117827.500

Table 3: Ethernet Transmissions per Unit Time for 100 % of Maximum Load

Packet Size	N_th	T_r	N_exp	Percent of Max
72	12820.51	43186	11577.82	90.31
256	4440.50	113765	4395.02	98.98
512	2325.58	218098	2292.55	98.58
1526	805.67	627322	797.04	98.93

Table 4: HYPERbus Transmissions per Unit Time for 100 % of Maximum Load

Packet Size	N_th	T_r	N_exp	Percent of Max
72	720.72	705120	709.10	98.39
256	651.59	779810	641.18	98.40
512	574.88	883968	565.63	98.39
1526	392.05	1296021	385.80	98.41
2048	336.89	1508399	331.48	97.52
4096	217.08	2340641	213.62	98.41

Legend for Tables 3 and 4

Number of packets Ethernet can transmit per second on a 500 meter network at the offered load x = N_th

N_th = §1000000/[(P * 0.8) + 10.8 + 9.6]† * 1.0

P = packet size in bytes

Calculated number of packets simulation transmitted on a 500 meter network per second = N_exp

N_exp = (500 * 10^6)/ T_r

T_r = value of relative clock at the end of the simulation program.

Table 5: Ethernet Packets Transmitted Successfully at 100 % of Maximum Load

Attempts Needed	Packet Length in bytes			
	72	256	512	1526
1	192	235	240	247
2	107	135	129	142
3	70	51	74	61
4	41	39	17	26
5	38	21	16	8
6	24	8	12	8
7	17	4	6	6
8	9	4	1	1
9	2	1	4	0
10	-	0	1	1
11	-	1	-	-
12	-	0	-	-
13	-	1	-	-
Average attempts per message	2.645	2.119	2.071	1.931

Table 6: HYPERbus Packets Transmitted Successfully at 100% of Maximum Load

Attempts Needed	Packet Length in bytes					
	72	256	512	1526	2048	4096
1	500	500	500	500	500	500
Average attempts per message	1	1	1	1	1	1

only a few retransmission attempts to be completed.

One of the design goals of HYPERbus was to give each message a strong chance of being successfully transmitted on the first attempt regardless of the offered load, which keeps the error rate at zero [8]. The access protocol will only allow a collision during the network acquisition time of the first message transmission at the end of a quiet period. The observed worst case statistics that a collision will not occur is 99.9832 per cent. Table 6 shows that all 500 transmissions were completed on their first attempt.

CONCLUSION

Both networks are stable at the normal and 100 per cent loads. The Ethernet model transmits an average of 96.7 per cent of the theoretical maximum number messages per unit time. The HYPERbus model operates at an average of 98.25 per cent for the same measure. These figures indicate that the throughput remains high under simulated high load conditions. The Ethernet keeps throughput high by transmitting other ready messages while messages that collided are waiting to retransmit. The HYPERbus keeps the throughput rate high on a busy network by enforcing the collision avoidance algorithm.
Ethernet suffers many packet collisions at the high loads. One message required thirteen transmission attempts to reach its destination. Each message required an average of 2.061 transmission attempts, while each message transmitted on HYPERbus required only 1 attempt. A test run of HYPERbus was performed to observe system reaction with collisions. The load was 90 per cent with 5000 72-byte data packets transmitted. The Ethernet model was run under the same conditions. HYPERbus had 4 messages collide, Ethernet had 3063. It is clear that HYPERbus has the best algorithm for assuring first attempt delivery of messages on the defined network configurations.

REFERENCES

1. K.C.E. Gee, Local Area Networks., National Computing Centre, Manchester, 1982.
2. R.M. Metcalfe and D.R. Boggs, "Distributed Packet Switching for Local Computer Networks," Communications of the ACM, Vol. 19, number 7, July, 1976, pp. 395-404.
3. J. Shoch and J. Hupp, "Measured Performance of an Ethernet LAN," Communications of the ACM, Vol. 23,number 12, December, 1980, pp. 711-721.
4. L. Swan, HYPERbus Performance, An unpublished performance study from Network Systems Corporation,Minneapolis, March 1984.
5. C.H. Sauer and E.A. MacNair, Simulation of Computer Communication Systems., Prentice-Hall Inc., Englewood Cliffs, 1983.

6. Digital Equipment Corporation, Introduction to Local Area Networks., Digital Equipment Corporation, Maynard, 1982.

7. Digital Equipment Corporation., et al., The Ethernet. A Local Area Network, Data Link Layer and Physical Layer Specification, Version 2.0, Digital Equipment Corporation, Maynard, November, 1982.

8. Network Systems Corporation, HYPERbus System Internal Reference Manual., Network Systems Corporation, Minneapolis, April, 1984.

9. A. Goyne, discussions on Ethernet operation, October, 1984.

10. L. Oliver., et al., A Network Systems Tutorial, an unpublished training manual from Digital Equipment Corporation, Maynard, July, 1984.

DAVID S. JENNINGS
Received his Bachelor of Science degree in Electronic Technology from the University of Houston in 1982. He received a Masters of Science degree in Computing and Information Sciences from Trinity University in 1985. He is currently teaching in the Department of Computing Science at the University of Queensland in Australia.

Department of Computing Science
University of Queensland
St. Lucia, Queensland
Australia 4067
011-617-378-0436

AARON H. KONSTAM
Received his Bachelor of Science degree in Chemical Engineering from the Polytechnic Institute of Brooklyn in 1957. He received his PhD. in Chemistry from Pennsylvania State University in 1961. He is currently teaching in the Department of Computing and Information Sciences at Trinity University in San Antonio, Texas.

Department of Computing and Information Sciences
Trinity University
715 Stadium Drive
San Antonio, TX 78284
(512)-656-0355