Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

MODELING AND ANALYSIS OF ETHERNET NETWORKS

John K. Jackman and D. J. Medeiros
Department of Industrial and Management Systems Engineering
The Pennsylvania State University
University Park, Pennsylvania 16802

ABSTRACT

A simulation model of an Ethernet local area network is described. The model is used to examine the performance of the truncated exponential backoff algorithm under conditions of heavy loading. At high loading levels the current backoff algorithm tends to generate a large queue of deferring nodes and thus perpetuates the collision interval. A new backoff algorithm is proposed which reduces the number of collisions on the network, resulting in shorter delay times. Comparitive results are presented for the current and proposed algorithms. At offered loads greater than 100%, a reduction in delay times of 10% and 25% is obtained for 512 and 64 byte packet lengths, respectively.

INTRODUCTION

The need for integration of intelligent devices in the work environment is forcing users to look to local area networks (LAN). These networks provide a means of communication between a wide variety of devices such as terminals, personal computers, mainframes, and printers, over a limited geographical area of 15 kilometers or less, with a transmission rate in the range of 3 to 50 megabits per second (Mbs). The proliferation of micro-computers and peripherals has been matached by the rapid increase in the number of LANs being offered by different vendors.

One of the most popular LANs is Ethernet, introduced in 1976 as a joint effort by Xerox, Digital Equipment Corp., and Intel [1]. The design is based on packet broadcast technology using a bus architecture. The access control protocol, which specified the rules for accessing the common bus so that contention is avoided, is considered to be the most significant design issue. Ethernet uses an access control known as carrier sense multiple access with collision detection (CSMA/CD).

CSMA/CD is a distributed form of control, eliminating the single point of failure problem. Each node on the network must listen to the bus and wait till the medium is idle before it can transmit its packet. The carrier sensing is also known as listen before transmitting (LBT). There is the potential for collisions between messages if two or more nodes try to transmit at the same time when the network becomes idle. During a transmission, the node will monitor the network for collisions. The collision detection is often termed listen while transmitting (LWT). If a collision is detected then the node will jam the network for a short period of time to insure that all nodes detect the collision.

A time delay is generated for each node involved in the collison from a probability distribution and the node starts the process over again. The time delay is a function of the frequency of collisions on the network. The advantages of the CSMA/CD are its relative simplicity, low cost, and provision of equal access for all nodes. The disadvantages are the inability to guarantee a response time and the degradation of throughput at high loading levels.

OBJECTIVES

A combined process and discrete event model has been developed for the simulation of Ethernet installations using the SIMAN [2] simulation language. The three main objectives of this model are to: 1) find a more efficient backoff strategy (the method of generating a time delay for colliding nodes), 2) develop a model which incorporates the spatial arrangement of nodes on the network, and 3) provide a real time graphics animation of the bus traffic.

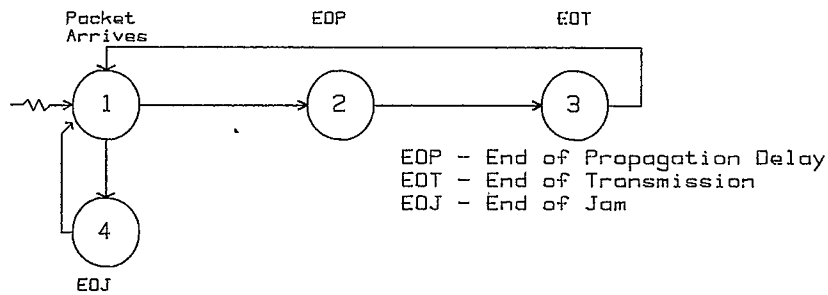
Previous studies made using different backoff algorithms for the CSMA/CD protocol did not find a significant improvement in performance [3,4]. Stuck has pointed out that the physical location of nodes on the network plays a critical role in network performance and has not been simulated up to this point [5]. The animation provides a means of verifying the protocols as well as finding shortcomings.

MODEL DESCRIPTION

The model contains both the process and the discrete event orientations. Collison handling occurs in a discrete event written in FORTRAN. The remainder of the model is a block diagram depicting the flow of packets through the network. For reasons of brevity, the model is presented in Figure 1 as an event diagram.

The model includes a transmit queue where a copy of the packet being transmitted resides until the propagation delay is over, at which point a successful transmission takes place. A second queue contains any packets which have collided so that future collisions during the propagation delay time can be detected. If the packet in the transmit queue is involved in a collision, it is removed from that queue and placed in the collision queue. A third queue contains packets that have deferred because they sensed that the bus was busy.

The simulation begins with the generation of the network configuration. Nodes are initialized by assigning a node number and a location on the network in terms of propagation delay from one end of the network. The current model assigns distances from a uniform distribution with a minimum of zero



Event 1: Packet Arrives		
Step	Condition	Action
1	ALWAYS	Create a packet
2		Initialize packet
3	If Bus busy AND no	Carrier Sensed so put
4	packets in collison	in Waiting queue
5	queue or transmit	
6	queue	
7	ELSE	Schedule EOP
8	If collision THEN	Schedule Packet Arrives
9		Schedule EOJ
10	ELSE	Seize the Bus
11		Put in Transmit Queue
12	ALWAYS	RETURN

Event 2: End of Propagation Delay		
Step	Condition	Action
1	If copy in collision	Remove & destroy copy
2	queue	Dispose of entity
3		RETURN
4	If matching packet in	Schedule EOT
5	Transmit Queue	RETURN

Event 3: End of Transmission		
Step	Condition	Action
1	ALWAYS	Release Bus
2		Schedule Packet Arrives
3		Collect Statistics
4	If Packets deferring	Schedule Packet Arrives
5	ALWAYS	RETURN

Event 4: End of Jam		
Step	Condition	Action
1	ALWAYS	Increment # of Collisions by 1
2		Increment Attempts
3		Release the Bus
4	IF Packets deferring	Schedule Packet Arrives
5	IF Attempts>16 THEN	Record Fatal Error
6	ALWAYS	Schedule Packet Arrives
7		RETURN

Figure 1 CSMA/CD Event Diagram

and maximum equal to the maximum propagation delay.

Each node generates packets using an exponential interarrival time. After each transmission the packet is recycled by assigning a new packet length from a distribution and scheduling a new arrival time. The packet arrives at the bus (described as event 1 in Figure 1) and listens to determine if another node is transmitting. This is accomplished by checking the status of the bus and looking in the transmit and collision queues. If the node detects that the bus is busy, then the packet defers and is sent to a queue to wait until the bus becomes idle.

If the packet accesses the bus, a copy of the packet waits for a time representing the maximum propagation delay. At the same time, the actual packet seizes the bus, where it may be involved in a collision. If a collision occurs the network is jammed for an additional 4.8 microseconds then the bus is released (event 4).

After the propagation delay, transmission continues for the packet which has seized the bus if it has not been involved in a collision (event 2). After the transmission is finished (event 3), the packet releases the bus and a new packet arrival is generated. In addition, packets which had been deferring are rescheduled for transmission.

The collisions are handled by a FORTRAN routine which contains the backoff algorithm. If this is the sixteenth transmission attempt of the packet then an error has occured and the packet is not retransmitted. Otherwise, a delay is calculated using the appropriate backoff algorithm and the packet is rescheduled to transmit after the dalay.

COMPARISON WITH PREVIOUS DATA

Shoch [6] conducted performance testing of an existing Ethernet system and measured the utilization of the bus using artificially induced traffic. Each node was continuously queued with transmissions so that the node would be ready to transmit immediately after a successful transmission. A comparison of Shoch's data with results from our model shows good agreement for packet lengths of 512 bytes and some differences for lengths of 6 bytes (see Figure 2). The transmission rate and propagation delay were not specified in the reference but were assumed to be 3Mbs and 16 microseconds, respectively. The propagation delay assumption may be the cause of the differences with the short packet lengths.

Shoch [6] and Tobagi [7] found that the performance of CSMA/CD degrades as the packet length is decreased. Our model indicates the same kind of behavior as illustrated for the truncated binary exponential backoff algorithm in Figure 3. Packet lengths of 512 and 64 bytes were chosen to represent the bimodal packet length distribution determined by Shoch. A maximum utilization of 90% was obtained with a transmission rate of 10 Mbs and a round trip propagation delay of 51.2 microseconds. The offered load was generated by increasing the number of nodes on the network with each node having exponential interarrival time. Data from Shoch's experiment gives a maximum utilization of 96% at the same packet length.

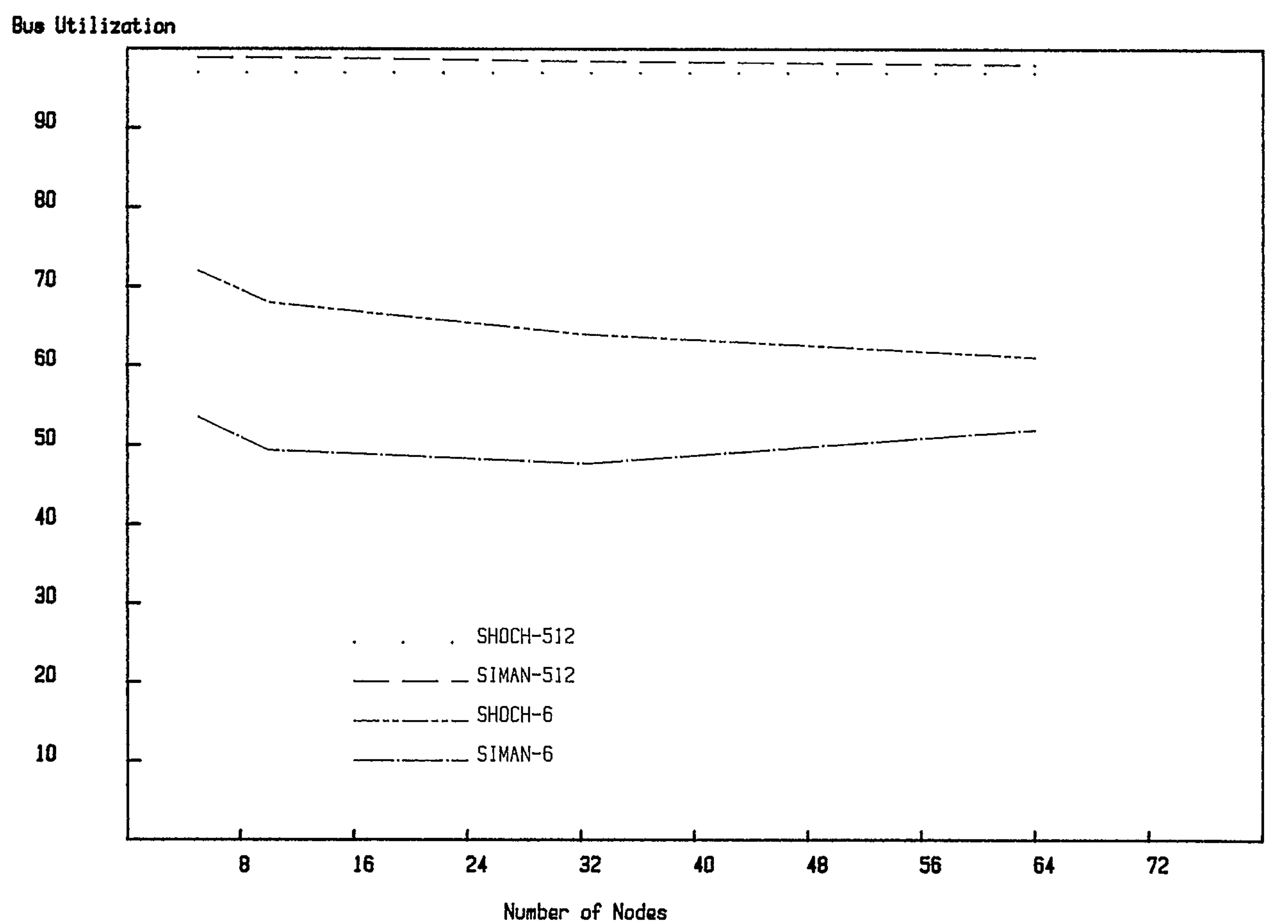


Figure 2 Comparison with Shoch's Data

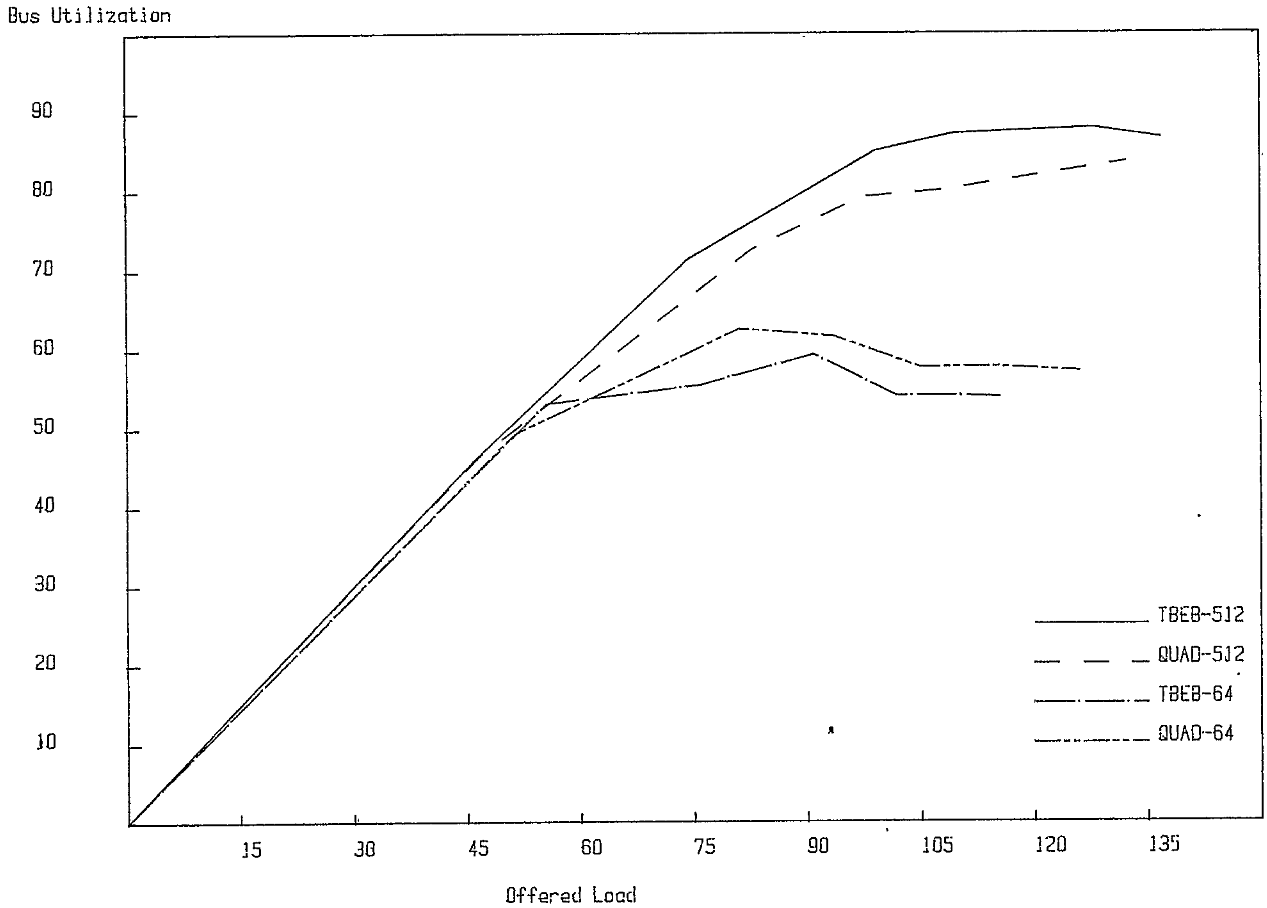


Figure 3 TBEB vs QUAD

MEASURES OF PERFORMANCE

The main measures of system performance for LANs have been the utilization of the bus and the delay time. An inverse relationship exists between bus utilization and packet delay time for the CSMA/CD protocol. Factors which influence these measures are packet length, propagation delay, slot time, and the backoff algorithm. The main concern of the actual network end user is not the utilization of the bus, but the delay encountered when trying to access the bus.

Longer packet lengths provide better bus utilization but increase the delay time. As would be expected, large propagation delays give lower bus utiliztion and longer delay times. The slot time and backoff algorithm are critical factors in system performance because the retransmission interval is usually a multiple of the slot time. This slot time is chosen as the maximum round trip propagation delay. In the Ethernet protocol it is specified as 512 bits or 51.2 microseconds.

TRUNCATED BINARY EXPONENTIAL BACKOFF

The truncated binary exponential backoff (TBEB) algorithm determines a backoff time after a collision as a sample from a discrete uniform distribution between 0 and some maximum value. The value is dependent on the number of attempts at transmission, as fo. ɔws:

K = min (10, nu ·er of attempts)
n = uniform (0, 2^K-1)
backoff = n x slot time

The basis for this algorithm is that the number of transmission attempts reflects the load on the system.

The strength of the CSMA/CD protocol is also its weakness. The capability of distributed control limits the cooperation betwen nodes. At low loading levels the backoff algorithm provides a FIFO access rule for the bus. As the loading increases the access becomes more LIFO than FIFO. Using a graphical animation of the system, it was noted that a packet arriving for the first time has a better chance of accessing the bus than those which are experiencing delays due to collisions. The obvious answer to this problem would be to minimize the collisions. A related problem is the large queues formed by packets which have been deferring at high loading levels. As soon as the bus becomes idle, the majority of these nodes collide. The backoff algorithm should attempt to minimize the number of nodes waiting for the bus.

QUAD BACKOFF

A number of different backoff algoriths were implemented in an attempt to reduce the number of collisions that takes place at a given loading level. The most promising of the alternatives uses the number of attempts raised to the fourth power. This becomes the maximum for the discrete uniform distribution. The backoff is calculated as follows:

k = min (5, Attempts)
n = Uniform (0, k^4)
backoff = n x slot time

This algorithm increases the interval of the uniform distribution at a faster rate than TBEB, and thus decreases the number of repeated collisions.

Comparative data for the two backoff algorithms was obtained using packet lengths of 512 and 64 bytes. The transmission rate was 10 Mbs with a round trip propagation delay of 51.2 microseconds as specified in the Ethernet standard.

The bus utilization shows some improvement for the new algorithm over the TBEB with a packet length of 64 bytes (Figure 3). However this is reversed for the longer packet. The main advantage of the new backoff is the shorter delay time obtained for both long and short packets.

At a loading level of 115% the QUAD backoff has a delay time 25% less than the current TBEB backoff (Table I). In order to illustrate the change in

Backoff Algorithm	Delay Time (microseconds)	95% CI
TBEB	1.76×10^3	0.17×10^3
QUAD	1.31×10^3	0.17×10^3

Table I. Delay, 64 Byte Packets, 115% Load

delay time with respect to the offered load, the percentage of packets that had a delay time less than or equal to 5 slot times was plotted versus offered load (Figures 4 and 5). It should be noted that at loading levels less than 100%, the backoff algorithms appear to be equivalent. However at levels greater than 100% the difference becomes significant. For 64 byte packets, approximatey 25% more packets successfuly access the bus within 5 slots. For the longer packets the difference is on the order of 10%. The reduced delay time can be accounted for by the smaller number of collisions observed for the new backoff algorithm.

CONCLUSION

The current Ethernet protocol was compared to a new backoff algorithm which reduces the delay times for long and short packets while maintaining approximately the same bus utilization. The new backoff decreases the number of nodes waiting to access the bus, reducing the number of collisions when the bus becomes idle. The decrease in delay time depends on the loading level of the network. At offered loads of 100% or greater the difference is at lest 10% for long packets (512 bytes) and 25% for short packets (64 bytes). Loads of 100% or greater can occur on a network experiencing bursty traffic.

REFERENCES

1. Metcalfe, R. M. and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," J. ACM, Vol. 19, pp. 395-404.

2. Pegden, C. D., Introduction to SIMAN, Systems Modeling Corporation, State College, 1982.

3. Moura, J., Field, J.,Wong, J., "Evaluation of Collision Control Algorithms in Ethernet," Proc. 6th Data Comm. Symp., 1979.

4. Marathe, Madhav, "Design Analysis of a Local Area Network," IEEE Conference on Computer Networks, 1980, pp. 67-81.

5. Stuck, Bart, "Which Local Net Bus Access is most Sensitive to Traffic Congestion?," Data Communications, January 1983, pp. 107-120.

6. Shoch, J. and Jon Hupp, "Measured Performance of an Ethernet Local Network," J. ACM, Vol. 23, pp. 711-721.

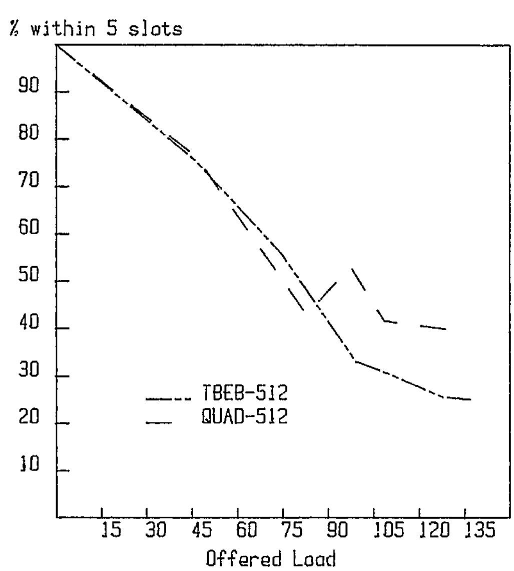


Figure 4 TBEB vs QUAD 512 Bytes

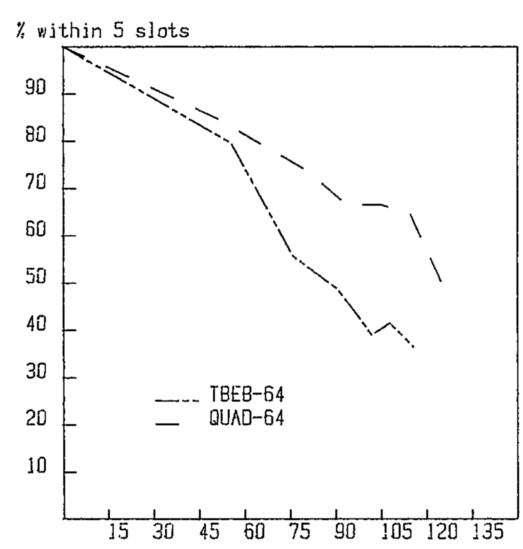


Figure 5 TBEB vs QUAD 64 bytes

7. Tobagi, F. and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," Proc. LACN Symp., 1979, pp. 217-243.