

AD-A241 262



Contract N61339-89-C-0043

February 6, 1990

Performance evaluation of

# Local area networks for real-time simulation

M. Bassiouni  
M. Georgiopolous  
J. Thompson

Institute for Simulation and Training  
12424 Research Parkway, Suite 300  
Orlando FL 32826

University of Central Florida  
Division of Sponsored Research

IST-CR-90-2

91-11416



REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		7 RESTRICTIVE MARKINGS NONE	
2a SECURITY CLASSIFICATION AUTHORITY N/A		3 DISTRIBUTION AVAILABILITY OF REPORT Approved for Public Release; Distribution is unlimited	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE N/A			
4 PERFORMING ORGANIZATION REPORT NUMBER(S) IST-CR-90-2		5 MONITORING ORGANIZATION REPORT NUMBER(S) IST-CR-90-2	
6a NAME OF PERFORMING ORGANIZATION Institute for Simulation and Training	6b OFFICE SYMBOL (if applicable)	7a NAME OF MONITORING ORGANIZATION Project Manager for Training Devices	
6c ADDRESS (City, State, and ZIP Code) 12424 Research Parkway, Suite 300 Orlando, FL 32826		7c ADDRESS (City, State, and ZIP Code) 12350 Research Parkway Orlando, FL 32826	
8a NAME OF FUNDING/SPONSORING ORGANIZATION DARPA/TTO	8b OFFICE SYMBOL (if applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N61339-89-C-0043	
8c ADDRESS (City, State, and ZIP Code) 1400 Wilson Blvd. Arlington, VA 22209		10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. PROJECT NO. TASK NO. WORK UNIT ACCESSION NO.	
11 TITLE (Include Security Classification) Performance Evaluation of Local Area Networks for Real-Time Simulation			
12 PERSONAL AUTHOR(S) Bassiouni, M.; Georgiopoulos, Michael; Thompson, Jack			
13a. TYPE OF REPORT TECHNICAL	13b. TIME COVERED FROM 4/89 TO 2/90	14. DATE OF REPORT (Year, Month, Day) February 6, 1990	15. PAGE COUNT 26
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES FIELD GROUP SUB-GROUP		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) LAN'S, Simulation Networks, SIMNET, LAN Performance, ETHERNET, Token Ring	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Recent advances in computer and communications technologies have made possible the interconnection of large number of real-time training simulators via local area networks. In this paper, we examine the networking aspects of distributed simulation, characterize the problems unique to this application, and present the findings of a comparison study for three different network access protocols suitable for distributed simulation. Specifically, these protocols are the carrier sense multiple access with collision detection (CSMA/CD), the token-ring protocol, and the virtual token-passing bus access method. The implications of the results of the comparison study and the insight gained from our research for improving real-time simulation networking are discussed.			
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Mike Carney		22b. TELEPHONE (Include Area Code) 407-390-4816	22c. OFFICE SYMBOL AIRC-PM-TND-EN

# Performance Evaluation of Local Area Networks for Real-Time Simulation

**M. Bassiouni**

*Department of Computer Science*

**M. Georgiopoulos**

*Department of Electrical Engineering*

**J. Thompson**

*Institute for Simulation and Training*

*University of Central Florida  
Orlando, FL 32816 USA*

*[Faint, illegible text, possibly bleed-through from the reverse side of the page]*

A-1

## Abstract

Recent advances in computer and communications technologies have made possible the interconnection of large number of real-time training simulators via local area networks. In this paper, we examine the networking aspects of distributed simulation, characterize the problems unique to this application, and present the findings of a comparison study for three different network access protocols suitable for distributed simulation. Specifically, these protocols are the carrier sense multiple access with collision detection (CSMA/CD), the token-ring protocol, and the virtual token-passing bus access method. The implications of the results of the comparison study and the insight gained from our research for improving real-time simulation networking are discussed.



## 1. Introduction

Recent breakthroughs in several computer/communications core technologies have made possible the interconnection of large number of real-time simulators (special purpose hardware) via local area networks. Two main applications/advantages of such networks are:

- (1) To provide a low-cost effective tool for the training of personnel in applications involving the interactions among mobile vehicles. Examples of such applications include training exercises for police forces, fire/ambulance services, and military combat fighting.
- (2) To provide an effective "test before you build" development tool to be used for evaluating proposed modifications in existing systems, as well as an aid in designing/developing new systems. Tactics and coordination strategies might also be simulated and evaluated before they are adopted in real-life.

We shall use the terms "distributed simulation", "simulation networks", and "real-time training networks" interchangeably to denote the networking of a large number of real-time simulators for the purpose of training [4, 15]. Each simulator consists of specialized hardware (a high-speed microcomputer, computer image generation subsystem, and sensor/control devices) bearing resemblance to the interior of the simulated vehicle (e.g., tank or police car). Each simulator has its own local copy of the database describing the simulated environment (e.g., city streets, buildings, terrain). As the crew of the simulated vehicle operate as they would in the real-life vehicle, the appropriate visual scenery is displayed on the CRT screens of their vehicle, as well as those of other vehicles in its sight range. It is obvious that the simulators participating in a training session must communicate with each other while carrying out the simulation. It is the responsibility of the underlying local area network (LAN) to provide each simulator with a reliable and fast mechanism to send and receive the information pertaining to the simulated activities.

The networking of real-time interactive simulation training systems departs from the traditional use of a computer network, whose function would normally be to provide sharing of computing resources among multiple users (nodes) on the network. When used to interconnect real-time simulators, the network is used almost exclusively for communication of process state information between the

simulators engaged in the training exercise.

There are many inherent limitations to using a network in this application. For example, as the number of simulators on the network and the workload per simulator increases, there may be a deterioration in throughput and a degradation of other network performance parameters. If data packet delays through the network become too large, for example, the effectiveness of a real-time training simulation may be overly compromised due to the time-critical response requirements in the simulation of true-to-life, action-requiring training scenarios. Depending upon the network communication protocol being used, there may also be an increase in the frequency of retransmitted and lost messages.

The Institute for Simulation and Training (IST) at the University of Central Florida has established a Network and Communications Technology Laboratory (NCTL/IST) dedicated to performing research for the purpose of enhancing the networking capabilities of distributed simulations. This laboratory houses a number of real-time simulators (different types of simulated ground and air vehicles) and is the center of several research activities/projects dealing with different aspects of real-time distributed simulation. In this paper, we report on the findings of an ongoing project focusing on the design and evaluation of local area networks for distributed simulation. Specifically, we examine the networking aspects of distributed simulation, characterize the problems unique to this application, and present the findings of a comparison study for three different network access protocol methods suitable for distributed simulation. The implications of the results of the comparison study and the insight gained from our research for improving real-time simulation networking are discussed.

## 2. Network System Configuration Models

Various choices exist for the implementation of a LAN [11, 16, 18, 19] (e.g., transmission medium, topology, access protocols, etc.) to interconnect simulation devices. In this paper, we present the results of a performance evaluation study for three network configurations having different topologies and protocol access methods. The first configuration has a bus topology and uses the ETHERNET protocol [14] which belongs to the class of Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocols [1]. The second configuration has a loop topology and employs a non-

contention protocol that avoids collision by a token-passing mechanism [3, 8, 17, 20]. The third configuration has a bus topology and employs a non-contention protocol that avoids collision by a virtual token-passing mechanism [7, 12]. A brief description of the three configurations is given below.

ETHERNET is a CSMA/CD protocol used for LANs with bus topology and is based on a distributed network control whereby each node on the common bus determines its own channel access time based on the information collected by monitoring the traffic activities on that bus. Figure 1 gives an example of an ETHERNET network configuration used to implement existing real-time simulation/training networks [15]. In this implementation, a number of nodes (typically eight) are connected through a multi-port transceiver to a single point on the coaxial cable (via a medium access unit). If a node on the bus has a packet ready for transmission, it first monitors the network to determine whether any transmission is in progress. If a transmission is in progress, the network bus is said to be "busy", otherwise it is "idle". If the node finds the bus busy, transmission of its data packet is deferred until the bus becomes idle. The node waits a certain minimum time interval (inter-frame gap) after the bus becomes idle and then begins the transmission of its packet. If multiple nodes attempt to transmit at the same time, their transmissions interfere with each other resulting in "packet collision". Once a transmitting node detects collision, it sends out a bit sequence referred to as a "jam signal". After the jam signal has been transmitted, the nodes involved in the collision schedule a retransmission attempt at a randomly selected time in the future. A packet is discarded if its transmission does not succeed (due to collisions) after sixteen consecutive transmission attempts. The performance of the ETHERNET protocol is directly related to how efficiently nodes avoid collisions and handle retransmissions. An ETHERNET LAN is the standard implementation currently used to interconnect real-time vehicle simulators at NCTL/IST.

Figure 2 gives a block diagram of the basic configuration of the token-ring LAN. Simply stated, a token-passing ring is a LAN with a loop topology in which a token (a unique bit sequence in a data packet) is passed around the network, in a round-robin fashion, from one node to the next. Contention for transmission is resolved by stipulating that only the node currently in possession of the token is allowed to transmit a frame, or a sequence of frames, onto the ring. When the transmission is finished,

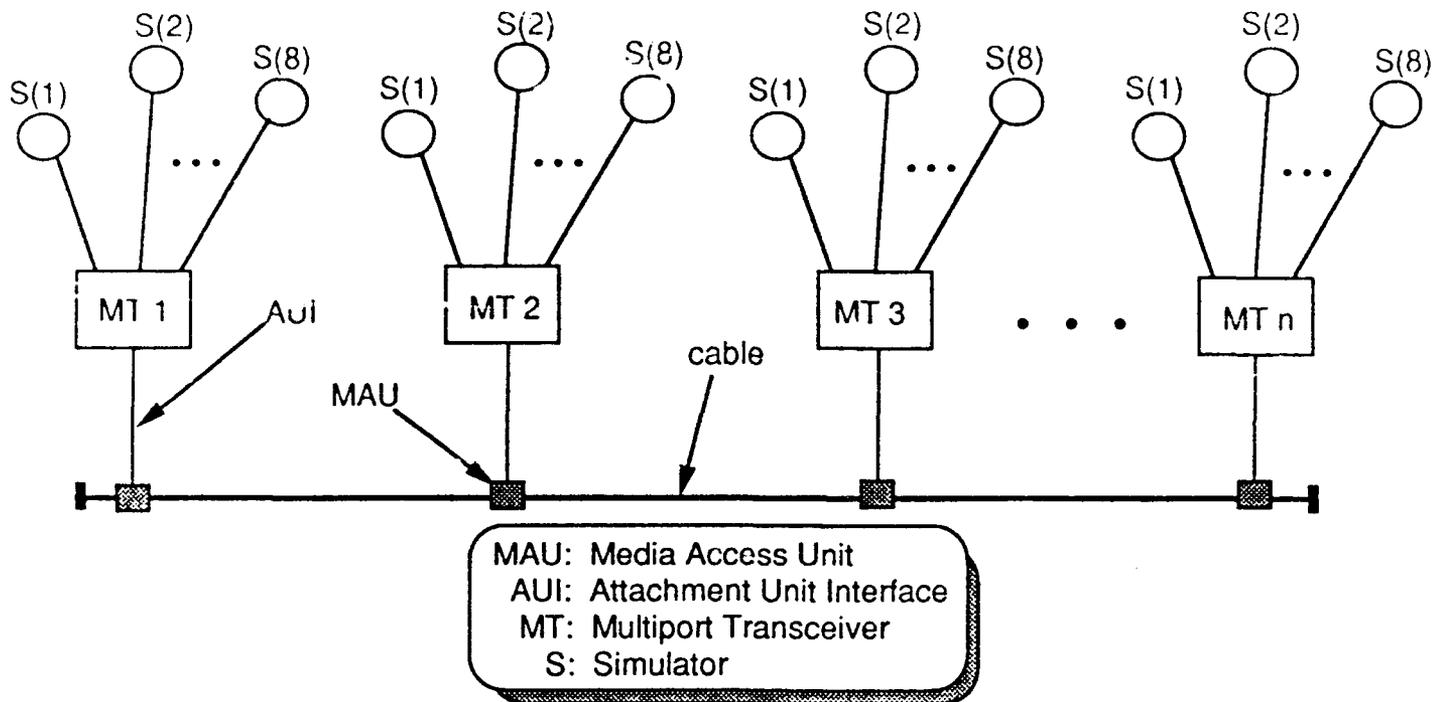


Figure 1. Bus Network Topology System Configuration

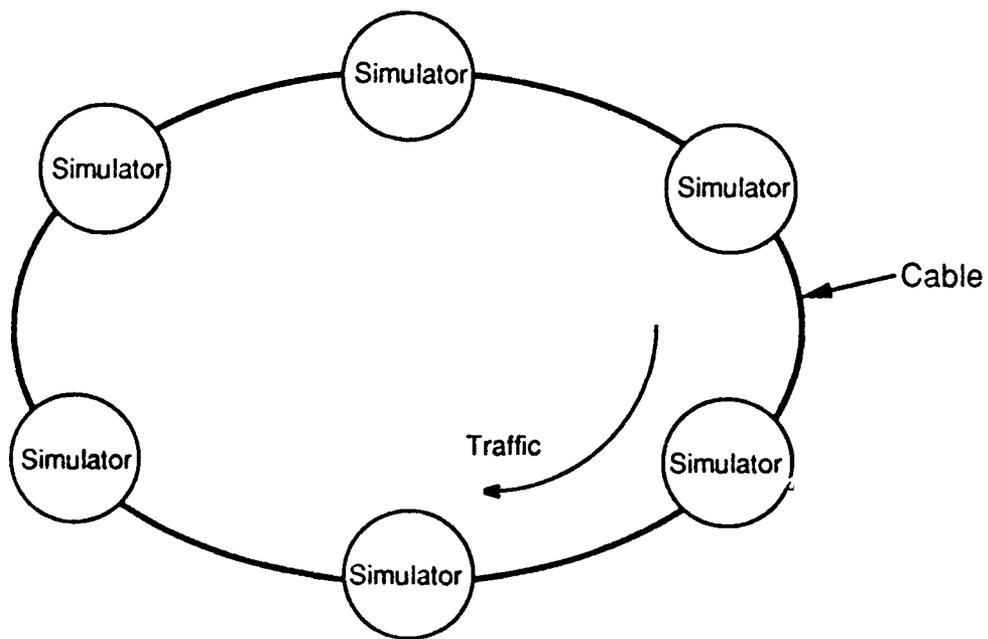


Figure 2. Ring Network Topology System Configuration

the token is passed to the node downstream which then gains the right to transmit. Since there is a single token on the ring, only one node can be transmitting at a time. Other (non-transmitting) nodes, however, continuously receive the bit stream, examine it and repeat it onto the network (i.e., place it on the medium to the next station). A station repeating the bit stream may copy it into local buffers or modify some control bits if appropriate. A prototype token-ring scheme for real-time simulators has been recently completed at NCTL/IST.

The Generalized Broadcast Recognizing Access Method (GBRAM) [12] is a bus-topology contention-free protocol based on a decentralized scheduling function that provides access to the network for each node on the bus at a unique time instant. The topology of GBRAM LANs is a bus similar to that shown in Fig. 1 for ETHERNET. A GBRAM implementation for real-time simulators is now underway at NCTL/IST.

In GBRAM, the nodes on the bus are ordered according to their physical location. Let us say that the leftmost node on the bus is assigned index value 1, the node immediately to its right is assigned the index value 2, and so on (i.e., nodes are assigned increasing integer values when scanning the bus from left to right). Under the GBRAM protocol, every node in the network perceives the channel state as consisting of cycles of scheduling and transmission periods. In general, the end of a transmission period designates the beginning of a scheduling period and the end of a scheduling period (in which there was no transmission) designates the beginning of another (new) scheduling period. The purpose of the scheduling period is to select the node that transmits next. As soon as a node starts transmission, the scheduling period is terminated and it is only after the end of the current transmission that a new scheduling period begins.

Assume that there are  $N$  nodes on the bus with index values 0 through  $N-1$  and let  $D(i,j)$  denote the delay (including propagational and circuit delays) needed for data to travel from node  $i$  to node  $j$ . Consider a scheduling period that starts when node  $j$  finishes transmission. The node that has the right to transmit next is node  $j +_N 1$ , where  $+_N$  indicates modulo addition using base  $N$  (i.e., node 0 follows node  $N-1$ ). If node  $j +_N 1$  has a packet, it will transmit it (after a certain delay as explained below) and the scheduling period is thus terminated. Otherwise, the scheduling period con-

tinues and the next node (i.e., node  $j+N/2$ ) detects after certain delay that the channel is still idle and therefore transmits a packet if it has one. In particular, if we let  $t$  denote the end of a scheduling period and if we assume that node  $j$  is the node that transmitted last prior to  $t$ , then an arbitrary node  $k$  is scheduled to transmit  $T(j,k)$  units of time after  $t$  where

$$T(j,k) = \begin{cases} \sum_{i=j+1}^k D(i-1,i) & \text{if } k > j \\ \left[ \sum_{i=j+1}^{N-1} D(i-1,i) \right] + D(N-1,0) + \left[ \sum_{i=1}^k D(i-1,i) \right] & \text{if } k \leq j \end{cases}$$

A node that has a packet to transmit initiates the transmission of the packet at its scheduled time instance, provided that the channel is sensed idle at that time. The above scheduling function ensures that GBRAM is a contention free protocol which avoids collision by scheduling different nodes at unique time instances. GBRAM is therefore considered to be a virtual token-bus protocol sharing the same general concept of explicit token-bus protocols [2].

### 3. Properties/Requirements of Distributed Simulations

The application of networks to interconnect real-time simulators has a number of characteristics and requirements. Recall that the main function of the LAN in this application is to communicate state update messages. When the state of a simulator changes (e.g., due to change in position or velocity, physical destruction, etc.) the simulator broadcasts a packet of type "state update". This message is delivered by the network to every other simulator or node on the network. Upon receiving a state update message, each simulator updates its own local database and displays any appropriate changes on its screens. To accomplish this function under real-time constraints, the design of a simulation LAN must satisfy the following requirements:

- \* The network must provide connectionless data transfer services (datagram services) that include point-to-point transfer, multicasting, and broadcasting.

- \* The transmission delay incurred by a packet should be minimal (far below 500 milliseconds [15]).
- \* The percentage of lost packets should be kept as close to zero as practicably possible. The impact of lost packets on the fidelity of the training exercise depends on the type/contents of the packet. For example, the loss of a single state update packet from a slowly moving vehicle can be usually tolerated and would not much degrade the animated imagery displayed by other simulators. This is because the simulator of this vehicle will soon broadcast another state update message after a small time interval and, hence, its coordinates in the database of other simulators will be corrected.

Due to the nature of simulation used for training, some nodes (simulators) on the network are more active than others. For example, there is usually a node in simulation networks used for the management/control of the entire training exercise. Also, two simulation LANs are sometimes connected together through gateways in order to enlarge the scope of training. Normally, such control nodes and gateways are much more active than the normal vehicle simulators. For the ETHERNET protocol, this creates a problem known as the greedy node effect which we analyze in the next section.

### 3.1. The Greedy Node Problem in Simulation Networks

The ETHERNET protocol uses an exponential back-off policy to resolve packet collisions. After a given packet collides for the  $j^{\text{th}}$  time, the node trying to send it delays its retransmission for  $R \times \tau$  seconds, where  $\tau$  is the end-to-end propagation delay (usually 51.2 microseconds) and  $R$  is an integer random number uniformly distributed in the range  $[0, 2^{\min\{j,10\}} - 1]$ . For example, if  $j = 5$  then  $R$  is uniformly distributed in the range  $[0,31]$  and if  $j = 12$  then  $R$  is distributed in the range  $[0,1023]$ . A packet is discarded after sixteen unsuccessful transmission attempts. During periods of high collision rates, this policy is biased in favor of the so called "greedy node" (i.e., a node which generates a large number of packets such that it quickly offers a new packet shortly after its current packet is successfully transmitted). After a collision has occurred, the greedy node has a higher likelihood of capturing the network for transmission. For example, consider a LAN with two dissimilar nodes: G the greedy node and N the normal (or nongreedy) node. In what follows, we present a typical scenario leading to the loss of a packet submitted by node N.

When two new packets from G and N collide for the first time, each of the two nodes will delay the retransmission of its packet by a random time interval equal to either 0 or  $\tau$ . With a probability of 0.25, node G delays by 0 and node N delays by  $\tau$ . In this case, node G transmits its packet successfully while node N will have to defer transmission until it detects a free channel for a period equal to the interframe gap (after G finishes transmission). Node N will then attempt to retransmit its packet at the same time that node G could also be attempting to transmit a new packet (note that the packet arrival rate in G is much higher than that of N). Thus the old packet of N and the new packet of G collide and node N (with 2 unsuccessful attempts) will delay retransmission by a random interval equal to 0,  $\tau$ ,  $2\tau$ , or  $3\tau$  while node G (with one unsuccessful attempt) will delay retransmission by either 0 or  $\tau$ . It follows that with a probability of  $5/8$ , node G will capture the bus and transmit its packet; with a probability of  $1/4$ , another collision will occur; and with a probability of  $1/8$ , node N will capture the bus and transmit its packet.

The above process can go on, each time the old packet of N collides with a new (or relatively new) packet from G. The latter node stands a better chance to transmit its packet and this process continues until the collision count for node N exceeds the limit and its packet is finally discarded by the ETHERNET protocol.

In general, if a packet from node N in its  $n^{th}$  retransmission collided with a packet from node G in its  $g^{th}$  retransmission, where  $g < n$ , then the probability that the greedy node G will capture the bus and transmit its packet successfully is given by:

$$\begin{aligned}
 P_G &= \sum_{k=0}^{2^{\min\{g,10\}}-1} \frac{1}{2^{\min\{g,10\}}} \times \sum_{j=k+1}^{2^{\min\{n,10\}}-1} \frac{1}{2^{\min\{n,10\}}} \\
 &= \sum_{k=1}^{2^{\min\{g,10\}}} \frac{2^{\min\{n,10\}} - k}{2^{\min\{g+n, g+10, 20\}}}
 \end{aligned}$$

The probability that another collision will occur is given by:

$$P_{coll} = \sum_{k=0}^{2^{\min\{g,10\}}-1} \frac{1}{2^{\min\{g,10\}}} \times \frac{1}{2^{\min\{n,10\}}}$$

$$P_{coll} = \frac{1}{2^{\min\{n, 10\}}}$$

The probability that the nongreedy node will acquire the bus is given by

$$P_N = 1 - P_G - P_{coll}$$

Table 1 gives the numerical values of the above probabilities for few sample values of  $g$  and  $n$ . From this table, it is obvious that the greedy node stands a better chance to capture the bus after a collision has occurred between its packet and a relatively older packet from a nongreedy node. The latter packet would therefore be forced to compete in the next round of transmission with a brand new packet from the greedy node, and its chance of successfully transmitting its packet gets slimmer as the number of its failed attempts increases.

Table 1: Channel Probabilities after Collision

$g$	$n$	$P_G$	$P_{coll}$	$P_N$
1	2	0.625000	0.250000	0.125000
1	4	0.906250	0.062500	0.031250
1	8	0.994141	0.003906	0.001953
1	12	0.998535	0.000977	0.000488
2	4	0.843750	0.062500	0.093750
2	8	0.990234	0.003907	0.005859
2	12	0.997559	0.000977	0.001465
3	4	0.718750	0.062500	0.218750
3	8	0.982422	0.003906	0.013672
3	12	0.995605	0.000977	0.003418
4	6	0.867188	0.015625	0.117188
4	8	0.966797	0.003906	0.029297
4	12	0.991699	0.000977	0.007324

One of the factors for the choice of the token-ring and GBRAM protocols as alternative designs for

simulation networks is that they both eliminate the adverse effect of the greedy node on other neighboring simulators. In token-ring LANs, the free token circulates around the ring and gives each node a fair chance to capture the token and transmit a packet. Similarly, GBRAM imposes a certain order by which each node is scheduled to transmit. Since this order depends on the identity of the node which transmitted last, each node under GBRAM also gets a fair chance to transmit.

#### 4. The Simulation Models

Detailed simulation programs were used to study the suitability of implementing a simulation network using the token-ring and GBRAM protocols, and to predict the performance of the three schemes (ETHERNET, token-ring, and GBRAM) when used to support a large number of real-time simulators. In this section, we give a high-level description of the simulation models used in evaluating and predicting the performance of real-time simulation networks for both the bus and the ring configurations. The simulation models are written in Concurrent-C (an extension of the C language with concurrent programming facilities based on the "rendezvous" concept). The powerful synchronization and concurrency aspects of Concurrent-C [9] have provided us with a notationally convenient and conceptually elegant tool for modeling the parallel activities of the vehicle simulators and the underlying networking layer.

The process interaction model of Concurrent-C has been used in our simulation models to map the different entities and activities of the real-time network to its corresponding Concurrent-C processes. Figure 3 gives a block diagram showing the interactions among the different Concurrent-C processes used to simulate the real-time simulation network. Typically, eight simulators connect to the coaxial transmission cable at a single point via a multi-port transceiver. Each of the simulators is modeled as a "Simnode" process. A "Busnode" process for each point of contact is created to accept and transmit local traffic from any one of the eight nodes (simulators), as well as retransmit any external messages arriving at this node.

#### 4.1. Bus-Topology Simulation Model

The following process types are the major generic entities used in the simulation of the ETHERNET bus configuration. Similar entities have been used for the bus-based GBRAM protocol.

- \* Process **Simnode** is used to represent a vehicle simulator (node) on the network. This process is the source of local traffic. It generates packets according to a specified input method such as using traces of real data or stochastically generated interarrival times (e.g., exponential, uniform, fixed with jitters, etc.). Upon the arrival of a local packet, the Simnode process makes a request to the corresponding Busnode process in order to transmit the new packet. At this point, the Busnode process checks for a carrier flag. If the flag has been off for at least the interframe gap, the Simnode process can proceed with its transmission. If the carrier flag is on, the Simnode process must wait for the interframe gap then retry its request. When a collision is detected during transmission, the Simnode process sends a jam signal and increments the collision counter for this packet. This is followed by invoking a backoff algorithm for retransmission [1]. In ETHERNET, a packet is discarded after 16 unsuccessful attempts for transmission.
- \* Process **Lserver** is used to implement and control the flow of traffic (packets and jam signals) in the direction from right to left for each node. A process of this type is created for each node.
- \* Process **Rserver** is analogously defined for traffic flowing in the direction from left to right.
- \* Process **Busnode** is used to represent the point of contact of each simulator with the ETHERNET bus. A process of this type is created for each such point of contact on the bus. The Busnode process acts like a server ready to accept requests from the local Simnode processes, the Lserver processes or the Rserver processes. The Busnode is responsible for detecting collisions and it continuously monitors the carrier flag to see if it is busy. In the case of a collision, the Busnode process calls the Scheduler to awaken the transmitting Simnode process which then stops the transmission and sends the jam signal.
- \* Process **Scheduler** (not shown in Fig. 3) is used to time order events and control the sequencing of the entire simulation. Delays in the simulated network (such as transmission delays) are handled

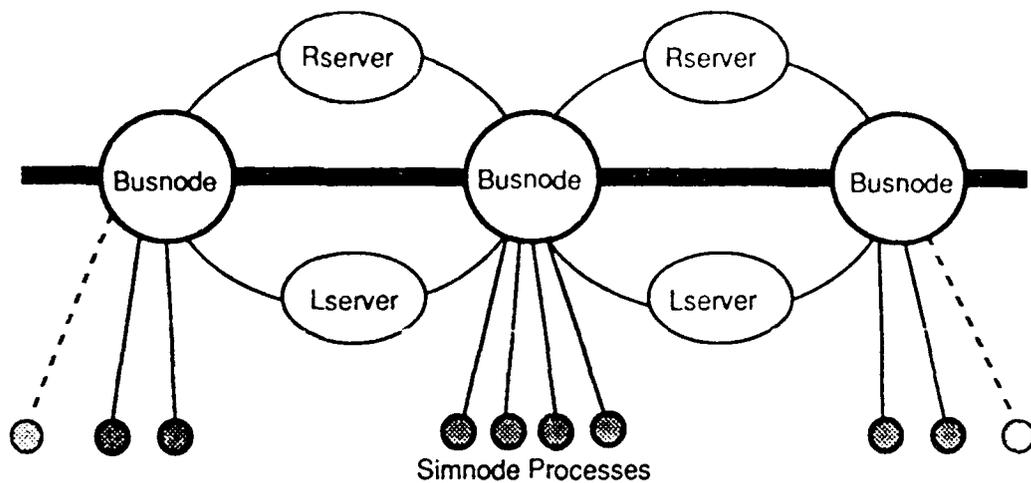


Figure 3. Simulation Model for Bus Topology Networks

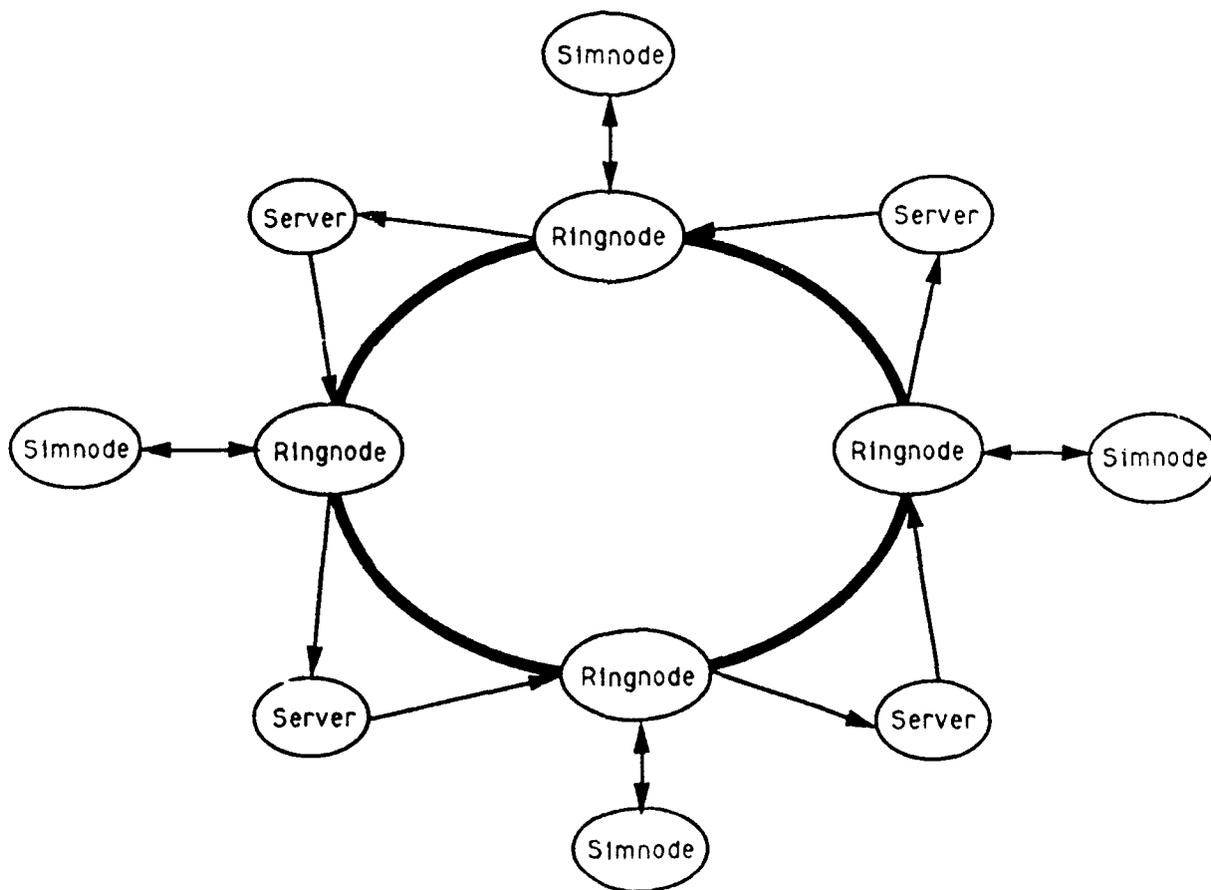


Figure 4. Simulation Model for Ring Topology Networks

by the Scheduler process which maintains the simulated clock and advances it appropriately. For each delay request from a process, the Scheduler determines the time when the process must be reactivated and saves this time in an "activation request" list. When all processes are waiting, the scheduler picks the next process to run, advances the simulated clock and reactivates the process. The simulated clock advances only when all processes are waiting; thus any (non-delay) computation done by a process takes place in zero simulated time.

#### 4.2. Ring-Topology Simulation Model

The simulation model for ring LAN topology is also written in Concurrent-C. A functional diagram of the simulation model for a real-time simulation network running under the token-ring protocol is shown in Figure 4. The ring simulation model uses the following two new process types (the Simnode and the Scheduler process types are the same as defined above for the bus model)

- \* Process **Ringnode** is used to monitor the ring traffic at the location of each node on the ring. A process of type **Ringnode** is created for each node on the ring. This process is responsible for implementing the token-based medium access protocol.
- \* Processes **Server** is used to simulate the (unidirectional) propagation of traffic and the corresponding delay between successive pairs of LAN nodes. A process of this type is created for each pair of adjacent nodes on the ring.

### 5. Performance Results

The main purpose of our comparative study is to investigate the suitability of implementing a simulation network using the token-ring and GBRAM protocols (as opposed to the existing ETHERNET implementations), and to predict the performance of these three schemes (ETHERNET, token-ring, and GBRAM) when used to support a large number of real-time simulators. In what follows, we discuss the main results of the study.

### 5.1. ETHERNET versus Token-Ring

The detailed simulation models described earlier have been used to gain insight into the performance of simulation networks under both the ETHERNET and token-ring protocols. Table 2 gives an example of some of the statistics collected in one test run when the ETHERNET configuration is driven by 80 simulators (similar appropriate statistics were collected for the token-ring scheme). It was assumed that the bus has a total of ten multi-port transceivers each of which is used to connect eight simulators to a single point on the coaxial cable.

**Table 2. Example of ETHERNET Statistics**

80 Simulators

Measure	Value
Average LAN load	5000 packets/sec
Packets discarded	0.0%
Max. Attempt	9
Avg. Attempt	1.47
Throughput	5000 packets/sec
Utilization	52.5%
Avg. Packet Delay	0.271 millisecond

For the token-ring model, the recreation of the "free token" onto the ring is assumed to follow the "early token release protocol" [5, 6]. According to this protocol, the transmitting station (the one which removed the free token from the ring) recreates the free token and puts it onto the ring immediately after it finishes transmitting its packet. This protocol results in better LAN throughput and smaller

packet delays than protocols that require the header (or the tail) of the transmitted packet to complete one cycle around the ring before the free token is recreated. Another factor affecting the performance of the token-ring scheme is the length of the free token. Although this length has been used as a variable in our various tests, all the results reported in this paper use a length of 24 bits (24 bits is the length used in many commercial token-ring implementations).

Since the ETHERNET and token-ring protocols are based on dissimilar topologies, the following assumptions were made in order to make the comparison as fair as practicable: i) The bandwidth of the ring is chosen to be identical to that of the ETHERNET bus (10 Mbits/sec), ii) Certain circuit delays in the ETHERNET multiport transceiver and the bus and ring attachments have been ignored in order to ensure that the comparison reflects the fundamental principles of the two schemes, iii) The speed of data propagation in both the ETHERNET coaxial cable and the ring links was assumed to be the speed of light.

The numerous simulation tests that we have conducted show that the throughput of simulation networks utilizing the ETHERNET protocol reaches a maximum of approximately 60-70% of the transmission medium bandwidth. The saturation in throughput is primarily due to the excessive collision rate that characterizes the behavior of ETHERNET at high network traffic loads. The token-ring configuration, however, has consistently yielded maximum throughput in the range 90-95% of the transmission medium bandwidth. The token-ring configuration uses a collision-free protocol and does not, therefore, suffer from the problem of declining throughput at high loads. Figure 5 shows the graph of throughput versus traffic load for the token-ring and ETHERNET LAN configurations with 80 nodes.

Our tests indicate that as the traffic load on the ETHERNET bus increases, the performance of the network deteriorates quickly resulting in significant increases in packet delays. With further higher loads, some packets are lost due to exceeding the limit of retransmission attempts, and the performance of ETHERNET rapidly collapses causing packet delays to become too large to be acceptable for real-time simulation. Token ring LANs, on the other hand, are much less sensitive to increased transmission rates compared to ETHERNET. Unlike collisions in ETHERNET, the overhead of token management in ring LANs does not result in throughput decline as the traffic load on the LAN increases. At high

loads, collision rates in ETHERNET become significant while the token-passing algorithm shows a more stable behavior. Since token rings are collision free, a maximum packet delay can be guaranteed for a given number of stations. Thus the real-time requirements of applications having high traffic loads (e.g., networks with large number of simulation/training devices) can be handled more gracefully using the contention-free token ring scheme.

Figure 6 shows the average packet delay versus LAN load for the token-ring and ETHERNET schemes. Although the token-ring has packet delays that are an order of magnitude better than those of ETHERNET, the above assumptions have undoubtedly contributed to widening the performance gap between the two models. In particular, using a "late free token release" policy and increasing the length of the free token would certainly result in an increase in the value of packet delays for token-rings and would therefore reduce the performance gap between the two configurations.

## 5.2. ETHERNET versus GBRAM

Both ETHERNET and GBRAM utilize the same bus topology (Figure 1) and therefore the same parameters (e.g., the time it takes to recognize that the channel is idle/busy) are used to compare the two schemes. The parameter values used in the tests reported in this paper are the maximum (worst-case) delays conforming to the IEEE 802 specifications as described in [1]. The length of the packet was chosen to be 1024 bits (which correspond to the size of the state update packet adopted in existing real-time simulation systems [15]).

Figure 7 shows the average delay versus traffic load performance for the GBRAM and ETHERNET protocols with 100 nodes. We observe from this figure that for light traffic load, ETHERNET induces a delay approximately equal to the packet transmission time, i.e., there is almost no contention delay for access to the network. As the traffic load increases to medium loads, the delay rises to several times the packet transmission time due to collisions and the associated back-offs. While a node is incurring a back-off delay, it is not contending for network access. Thus, larger delays effectively reduce the instantaneous offered load and help maintain stability. Nevertheless, as the input traffic increases above a certain point, we observe an abrupt increase in ETHERNET delays due to the fact that at high loads

### Throughput vs Load

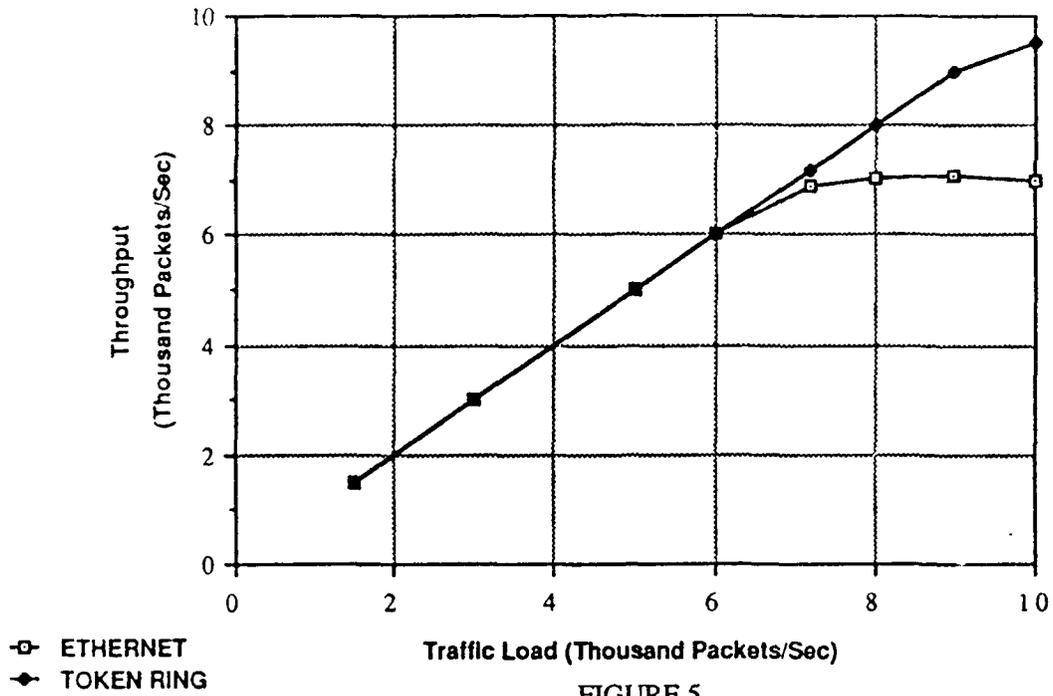


FIGURE 5

### Average Packet Delay vs Traffic Load

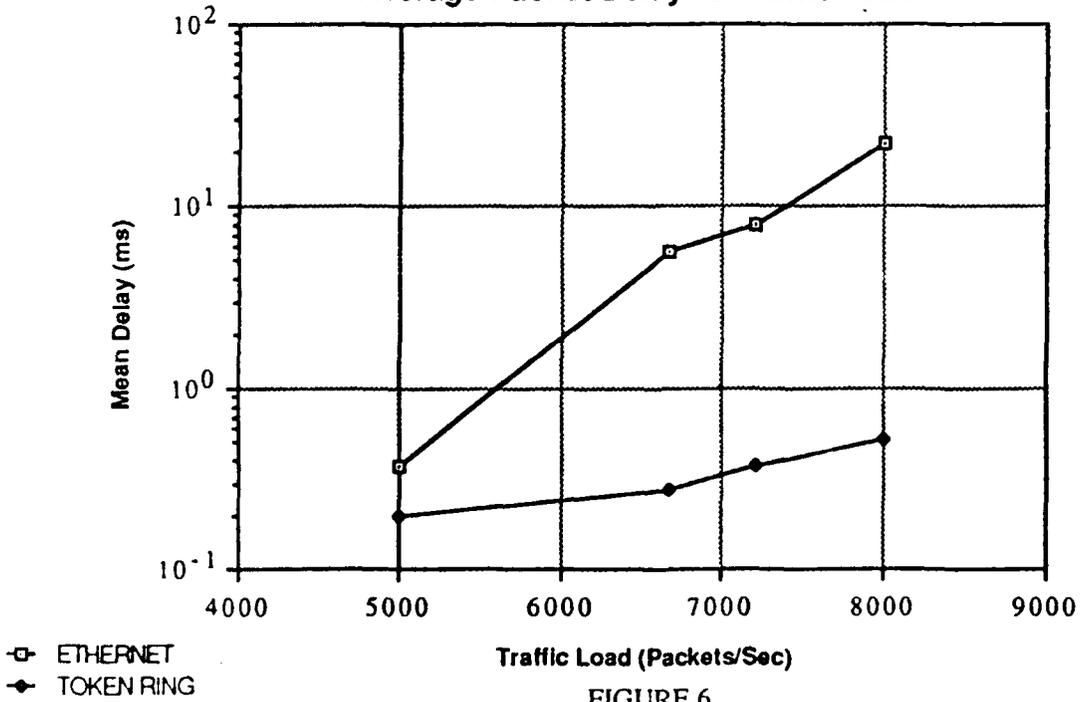


FIGURE 6

most nodes have more than one packet at a time awaiting transmission. While the "discarding of packets" feature of the ETHERNET protocol will generally guarantee relatively reasonable delays for the first packet in each queue, the second or third packet in the queue will experience larger delays. This results in the "blow-up" behavior of the ETHERNET protocol once the traffic load exceeds a certain limit. On the other hand, the GBRAM protocol exhibits a much more rational behavior. For light traffic loads, GBRAM induces a delay larger than the packet transmission time due to the fact that a packet may arrive at a node before its scheduling instance comes up. As expected, GBRAM is slightly worse than ETHERNET for light traffic loads. As the traffic increases, the performance of GBRAM becomes comparable with that of ETHERNET. For high traffic loads, GBRAM incurs smaller delays and it outperforms ETHERNET. This is because the deterministic nature of GBRAM avoids collision altogether. As a result, the channel is either idle or busy with successful transmissions. At high loads, all nodes are active most of the time. Hence, the channel is almost entirely occupied by successful transmissions (allowing us to accommodate a traffic load close to 100% of the bandwidth). It is worth noting that at traffic load of 9,000 packets/sec, GBRAM induces a delay smaller than that produced by ETHERNET at traffic load of 6,000 packets/sec. The cutoff point in Figure 7 occurs at a traffic load of approximately 4,500 packets/sec. Notice that even for traffic loads below this cutoff point, GBRAM exhibits a reasonable performance (i.e., delays smaller than 0.4 ms). Figure 8 shows the delay versus traffic load performance for GBRAM and ETHERNET with 400 nodes. Similar observations regarding the performance of the two protocols can be drawn from Figure 8.

Figures 9 and 10 show the histograms of the delay distribution for the GBRAM and ETHERNET protocols at traffic load of six thousand packets/second with 100 nodes and 400 nodes, respectively. The labels on the horizontal axis indicate the upper limit of each bin. For example, the leftmost bin corresponds to packets with delays smaller than or equal to 0.5 millisecond; the next bin corresponds to packets experiencing delays larger than 0.5 and smaller than or equal to 1.0 millisecond. We observe from Figures 9 and 10 that most of ETHERNET packets experience a delay smaller than 0.5 millisecond. On the other hand, a significant portion (larger than 10%) of ETHERNET packets experience delays larger than 3 millisecond for a 100-node LAN and larger than 5 millisecond for a 400-node

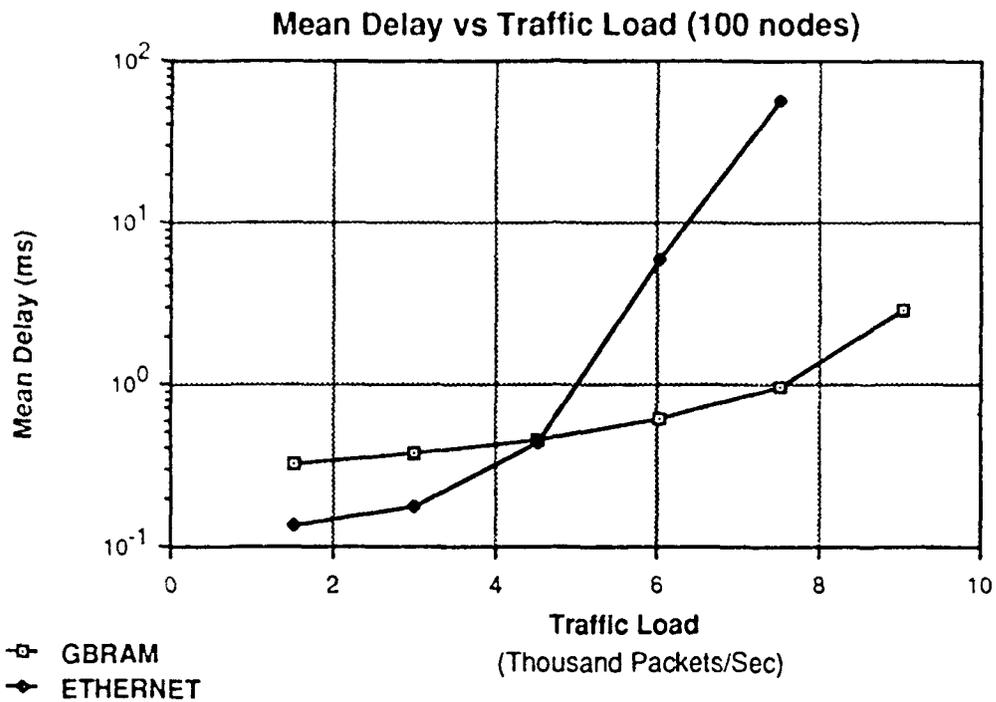


FIGURE 7

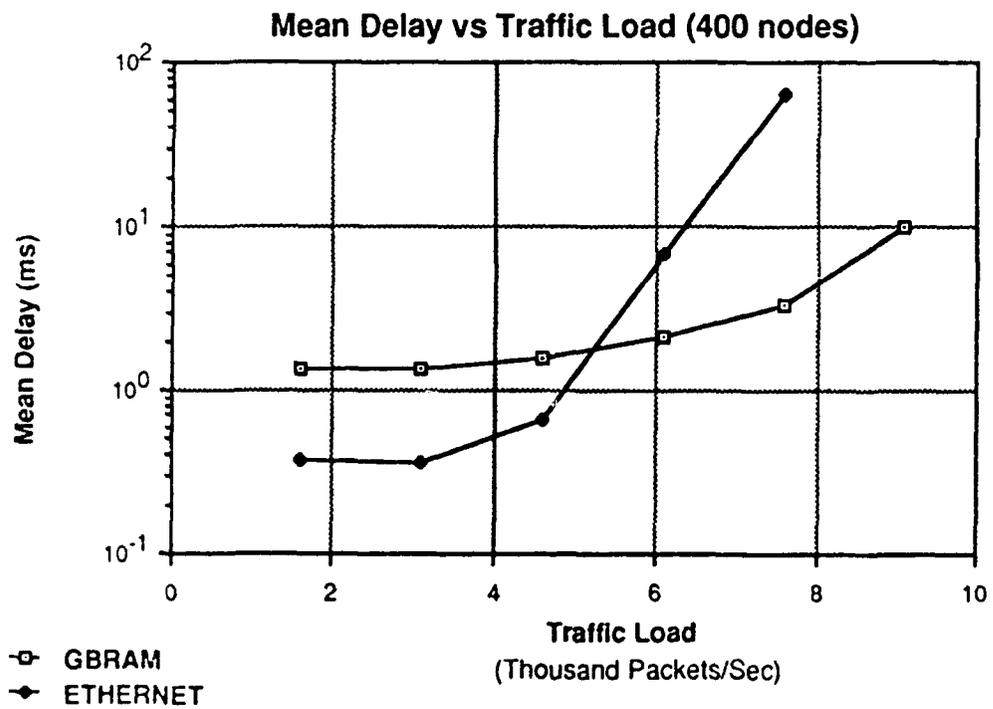


FIGURE 8

LAN. For GBRAM, only 0.2% of the packets experience delays larger than 3 millisecond for the 100-node LAN and 2.0% of the packets experience delays larger than 5 millisecond for the 400-node LAN. Overall, it can be seen that the delays induced by GBRAM are more evenly distributed than those of ETHERNET.

## 6. Other Considerations

In addition to the comparative results obtained by the simulation models, many other factors need to be considered when choosing and/or designing a medium access protocol for simulation networks. Below, we discuss three important considerations relevant to the design of real-time simulation networks.

1) Unlike ETHERNET and GBRAM, token rings provide a priority-based scheme for packet transmission across the network. In the ANSI/IEEE 802.5 ring implementation [3], the passing token has three bits indicating the current priority level of the ring (this gives 8 priority levels: 0 is the lowest priority and 7 is the highest priority). A station that captures the token, can only transmit packets whose priority is equal to or higher than the priority of the passing token. The 802.5 protocol also provides mechanisms that enable stations to request/change the priority of the passing token. In simulation networks, this capability might be used to assign priorities to the different types of network messages in order to optimize real-time performance at peak load conditions.

2) Because of its point-to-point connection property, rings readily accommodate the use of optical fiber as a transmission medium. In addition to offering, reduced size/weight and enhanced safety features, optical fiber also offers very high signal bandwidth. One very promising implementation of ring networks using optical fiber is the Fiber Distributed Data Interface (FDDI). FDDI [10, 13] is a 100 Mbits/sec token ring LAN protocol that is rapidly becoming accepted as the premier high speed LAN standard [13]. With its embedded extensibility to support even higher speeds (500 to 1000 Mbits/sec), FDDI is poised to become the dominant high-end LAN of the 1990's. The paradigm for FDDI topology is known as a "dual counter-rotating ring of trees". The physical layer topology consists of independent, full-duplex, point-to-point physical connections, while the logical layer consists of one or two rings. The FDDI MAC (medium access control) protocol provides data services similar to those of the IEEE

### Percentage of Good Packets vs Delay

(100 nodes: Traffic Load=3,000 packets/sec)

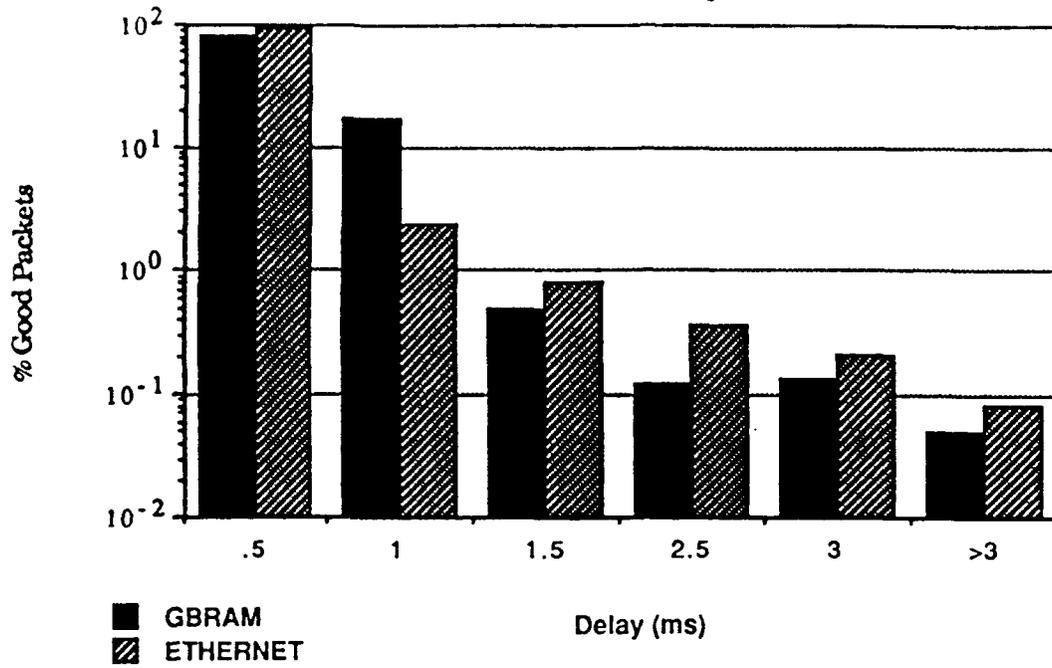


FIGURE 9

### Percentage of Good Packets vs Delay

(400 nodes: Traffic Load=6,000 packets/sec)

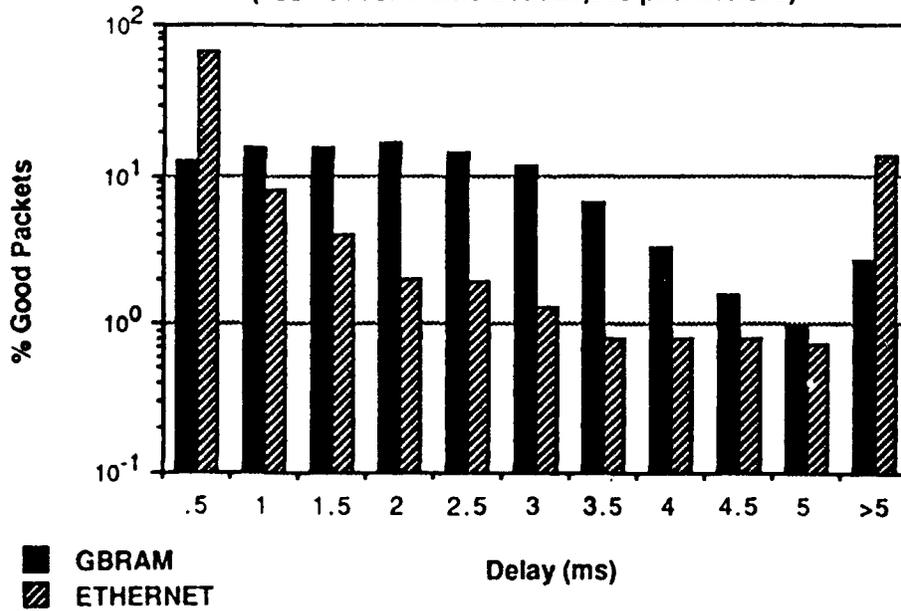


FIGURE 10

802.5 token rings. FDDI technology might eventually provide the simulation/training industry with powerful real-time LANs capable of interconnecting an unprecedented number of stations.

3) One main advantage of the bus structure (ETHERNET and GBRAM) over ring LANs is the reliability of network operation following a node failure. Bus-based LANs are resistant to node failures since the propagation of messages on the bus does not require the participation of any given node. A failure of a station on the ring structure, however, can bring the entire LAN down. This problem has been somewhat reduced by the increased reliability of today's ring chips and off-the-shelf ring attachments. Furthermore, FDDI rings use (optional) optical bypass switches in order to allow inactive (off-line) stations to pass the traveling data-carrying light waves directly from one neighbor to the next without active power.

## 7. Conclusions

In this paper, we have examined the problem of networking large number of real-time vehicle simulators and presented the results of a comparison study of three different network protocol access methods suitable for the networking of simulation training devices.

A local area network for real-time simulation is used almost exclusively for communication of process state information between the simulators engaged in the training exercise. One of the basic requirements of such a network is to provide small packet delays and to substantially reduce the data loss rate. Our simulations indicate that contention-free protocols (token-ring and GBRAM) demonstrate superior performance over CSMA/CD counterparts for simulation networks with high traffic loads (i.e., 65% to 90% of bandwidth). Compared to ETHERNET, both the token-ring and GBRAM protocols induce smaller packet delays, higher throughput, and eliminate the greedy node problem.

## Acknowledgements

This work has been supported by the U.S. Army's Program Manager for Training Devices (PM-TRADE) under Contract N61339-88-G-0002 Order 008 and the U.S. Defense Advanced Research Projects Agency (DARPA) under Contract N61339-89-C-0043. The views and conclusions in this document are those of the authors and do not represent the official policies of the funding agencies, the Institute for Simulation and Training, or the University of Central Florida.

The authors would like to thank J. Cadiz, M. Chiu, N. Christou, H. Chu, and S. Chatterjee for their technical assistance during the course of this project.

## References

- [1] ANSI/IEEE- International Standard 8802/3 "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specification" IEEE Computer Society Press, 1985.
- [2] ANSI/IEEE- Draft International Standard ISO/DIS 8802/4 "Token-passing bus access method" IEEE Computer Society Press, 1985.
- [3] ANSI/IEEE- International Standard 8802/5 "Token ring access" IEEE Computer Society Press, 1985.
- [4] Bassiouni, M.; Georgiopoulos, M. and Thompson, J. "Real time simulation networking: network modeling and protocol alternatives" Proceedings of the 11th Interservice/Industry Training Systems Conference (I/ITSC), November 1989, pp. 52-61.
- [5] Bassiouni, M. and Gupta, A. "Computing estimates of waiting times in local area networks with priority classes" Journal of Computer Systems Science and Engineering, Vol. 2, No. 4, October 1987, pp. 155-160.
- [6] Berry, R. and Chandy, K. "Performance models of token ring local area networks" Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, 1983, pp. 29-31.
- [7] Chlamtac, I.; Franta, W. and Levin, K. "BRAM the broadcast recognizing access method" IEEE Transactions on Communications, Vol. COM-27, No. 8, August 1989, pp. 1183-1189.
- [8] Dixon, R. ; Strole, N. and Markov, J. "A token ring network for local data communication" IBM system Journal, Vol. 22, 1983, pp. 74-62.

- [9] N. Gehani and Roome, W. "Concurrent C" Technical Report, AT & T Bell Laboratories, 1986.
- [10] Kolnik, I and Garodnick, J. "First FDDI local area network" Proceedings of 12th IEEE Conference on Local Computer Networks, 1987, pp. 7-11.
- [11] Kummerle, K. and Reiser, M. "Local area networks- major technologies and trends" in *Local Area Networks*, edited by K. Kummerle, J. Limb and F. Tobagi, IEEE Press, 1987, pp.2-27.
- [12] Liu, T.; Li, L and Franta, W. "A decentralized conflict-free protocol, GBRAM, for large scale networks" Proc. Computer Network Symposium, December 1981, pp. 39-54.
- [13] Marrin, K. "Emerging standards, hardware, and software light the way to FDDI" Computer Design, Vol. 28, No. 7, April 1989, pp. 51-57.
- [14] Metcalfe, R. and D.R. Boggs "Ethernet : Distributed Packet Switching for Local Computer Networks", Communications of ACM, Vol. 19, No. 7, 1976, pp. 395-403.
- [15] Pope, Arthur "The SIMNET network and protocols" BBN Report No. 7102, BBN Communications Corporation, MA, July 1989.
- [16] Stallings, W. "Local networks" ACM Computing Surveys, Vol. 16. No. 1, March 1984, pp. 3-42.
- [17] Strole, N. "A local communication network based on interconnected token-access rings: a tutorial" IBM Journal on Research and Development, Vol. 27, No. 5, 1983, pp. 481-496.
- [18] Tanenbaum, A. "Network protocols" ACM Computing Surveys, Vol. 13, No. 4, Dec. 1981, pp. 453-490.
- [19] Trooper, C. *Local Computer Network Technologies*. Academic Press, 1981.
- [20] Weaver, A. and Colvin, M. "A real-time messaging system for token-ring networks" Technical Report TR-88-08, Department of Computer Science, University of Virginia, 1988.