PERFORMANCE OF STANDARD AND MODIFIED NETWORK PROTOCOLS IN A REAL-TIME APPLICATION

Mostafa A. Bassiouni Computer Science Department
Michael Georgiopoulos Electrical/Computer Engineering
Ming-Hsing Chiu Computer Science Department
Ratan K. Guha Computer Science Department

University of Central Florida
Orlando, FL 32816 USA

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. The self-healing nature of real-time networked simulation has been found to allow for a modification based on discarding old packets whenever new state updates become available. The performance benefits obtained by implementing this modification at both the application and data-link layers are presented for two medium-access network protocols: ETHERNET and GBRAM. An analysis of a phenomenon, called the greedy node problem, in distributed simulation networks is presented.

I. INTRODUCTION

Recent breakthroughs in several computer-related technologies have made possible the interconnection of large number of real-time simulators via local area networks. We shall use the terms "distributed simulation" and "simulation networks" interchangeably to denote the networking of a large number of real-time simulators for the purpose of training [IEEE93]. 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 video screens of their vehicle, as well as those of other vehicles in its sight range. It is the responsibility of the underlying LAN to provide each simulator with a reliable and fast mechanism to send and receive the information pertaining to the simulated activities.

In previous work, we examined some of the problems facing the design and implementation of effecient simulation networks, e.g., data reduction of simulator's traffic [BASS90, BASS94], on-the-fly decompression algorithms [BASS95a], performance evaluation of real-time transfer syntax schemes [BASS95b] and network design alternatives [BASS89]. In this paper, we examine other networking aspects of distributed simulation and present the results of a performance evaluation study for standard and modified medium-access protocols suitable for distributed simulation.

II. NETWORK SYSTEM CONFIGURATION MODELS

We shall consider two network configurations having bus-based topologies. The first configuration is an ETHERNET network which uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) [ANSI85a]. The second configuration uses GBRAM which is a non-contention protocol that avoids collision by a virtual token-passing mechanism [CHLA89, LIU81]. An

0-7803-3873-1/97 $10.00 © 1997 IEEE

ETHERNET LAN is a popular implementation choice for interconnecting real-time vehicle simulators. The Generalized Broadcast Recognizing Access Method (GBRAM) 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 of ETHERNET. 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 0, the node immediately to its right is assigned the index value 1, and so on. Under the GBRAM protocol, every node in the network perceives the channel state as consisting of cycles of scheduling and transmission periods. 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 M nodes on the bus with index values 0 through M-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 $j \oplus 1$ where $\oplus$ indicates modulo addition using base M (i.e., node 0 follows node M-1). If node $j \oplus 1$ has a packet, it will transmit it (after a certain delay) and the scheduling period is thus terminated. Otherwise, the scheduling period continues and the next node (i.e., node $j \oplus 2$) detects after certain delay that the channel is still idle and therefore transmits a packet if it has one. The equations governing the times at which different nodes are scheduled to transmit can be found in [CHLA89, LIU81].

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. 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 [ANSI85b].

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. These control nodes are more active than the normal vehicle simulators. For the ETHERNET protocol, this creates a problem known as the greedy node effect which is analyzed below.

III. THE GREEDY NODE PROBLEM

The ETHERNET protocol uses an exponential back-off policy to resolve packet collisions. After a given packet collides for the j^{th} time, the node trying to send it delays its retransmission for $R \times \tau$ seconds, where τ is the end-to-end bus 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 likelyhood of capturing the network for transmission. Below, we analyze a LAN with two active nodes: G the greedy node and N the normal (or nongreedy) node.

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 τ. With a probability of 0.25, node G delays by 0 and node N delays by τ. 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. 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, τ, 2τ or 3τ while node G (with one unsuccessful attempt) will delay retransmission by either 0 or τ. 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. This 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. A formal analysis is given below.

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:

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

$$= \sum_{k=1}^{2^{\min\{g,10\}}} \frac{2^{\min\{n,10\}} - k}{2^{\min\{g+n,\, g+10,\, 20\}}}$$

The probability that another collision will occur is given by

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

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

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

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

The numerical values of the above probabilities for different values of g and n are given in Table 1.

One of the reasons for the choice of GBRAM in our study is that it is a bus-based protocol that eliminates the adverse effect of the greedy node on other neighboring simulators. 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 gets a fair chance to transmit.

IV. COMPARISON OF STANDARD AND MODIFIED PROTOCOLS

Detailed simulation models (written in Concurrent C) have been used to gain insight into the performance of simulation networks under the two network protocols described earlier. Both ETHERNET and GBRAM utilize the same bus topology 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 [ANSI85a]. 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).

Figure 1 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 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 1 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 2 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 2.

V. Modified Protocols

One of the features that we considered in our tests is unique to simulation networks. This feature allows for an optimization to reduce the load on the access medium. An explanation of this optimization is given below.

Upon a state change (due to movement or change in the vehicle's appearance), a simulator on the network broadcasts the value of its new state to other nodes on the LAN. Each new state results in the generation of a new packet at the application layer. The packet is then submitted to the data link layer in order to start the process of its transmission. In ETHERNET, for example, only one packet per node is delivered for transmission at a time. Other packets are normally queued up at the application level (i.e., at the node level) waiting for the end of the ongoing attempt of transmission.

Due to the self-healing nature of real-time simulated training, a modification can be introduced to relieve the congestion of the network at high traffic loads. In this modification, an old packet that has not yet been transmitted onto the network, can be simply discarded upon the generation of a newer packet (carrying more recent information about the location and appearance of the vehicle). This modification allows for the graceful recovery of network performance at high traffic loads without overly compromising the realism of the training exercise. The process of discarding old packets can be implemented in two ways: i) at the application layer, or 2) at the data-link layer. These two methods are discussed below.

In the application-layer case, the arrival of a new packet can be simply used to replace the previous packet (stored at the application layer) which holds a less recent state. The discarding of the old packet helps speed up the transmission of the latest state of a node. Notice that any packet already submitted to the data link layer (e.g., to the ETHERNET medium-access controller board) is not affected by new arrivals. This is because such packets are under the control of the medium-access protocol boards and are not accessible from the application layer.

The above modification has been incorporated in the ETHERNET model. Table 2 shows a typical test run of the standard ETHERNET (Version I) and the corresponding configuration when packets are discarded at the application layer (Version II). The results given in Table 2 are for an ETHERNET LAN driven by 80 simulators with an average network load of 6667 packets per second.

Discarding old packets at the data-link layer (due to the generation of newer packets) can further improve LAN performance at high traffic loads. We have investigated the impact of the data-link modification on LAN performance using the simulation models. Table 3 gives a comparison of ETHERNET performance with and without the above-mentioned modification at the data-link layer. The column labeled ETHERNET I gives the results for the standard ETHERNET implementation while that labeled ETHERNET II gives the results for the modified data-link protocol. All the results in Table 3 correspond to a LAN configuration with 100 nodes.

Table 4 gives a comparison of GBRAM performance with and without the data-link modification. The column labeled GBRAM I gives the results for the standard GBRAM implementation while that labeled GBRAM II gives the results for the modified data-link protocol. All the results in Table 4 correspond to a LAN configuration with 100 nodes.

VI. CONCLUSIONS

The contention-free protocol, GBRAM, demonstrates superior performance over the CSMA/CD counterpart for simulation networks with high traffic loads (i.e., 65% to 90% of bandwidth). The self-healing nature of real-time networked simulation permits the discarding of old packets when new state updates become available. The performance benefits obtained by discarding old packets at the application and data-link layers have been presented. An analysis of the greedy node problem for an ETHERNET bus with two active nodes was presented.

ACKNOWLEDGMENT

This work has been supported by the Army Research Office under Grant No. DAAH04-95-1-0250 and by earlier grant from STRICOM. The views and conclusions herein are those of the authors and do not represent the official policies of the funding agencies.

REFERENCES

[ANSI85a] ANSI/IEEE Standard 8802/3 "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specification" IEEE Computer Society Press, 1985.

[ANSI85b] ANSI/IEEE Standard 8802/4 "Token passing bus access method" IEEE Computer Society Press, 1985.

[BASS95a] Bassiouni, M. and Mukherjee, A. "Efficient decoding of compressed data" Journal of the American Society of Information Sciences (JASIS), John Wiley & Sons, Vol. 46, No. 1, 1995, pp. 1-8.

[BASS95b] Bassiouni, M. and Loper, M. "Applications of OSI standard transfer syntax in real-time networks" Journal of Computer Standards and Interfaces, Elsevier Publishing, Vol. 17, 1995, pp. 349-357.

[BASS94] Bassiouni, M.; Chiu, M. and Garnesy, M. "On-the-fly message filtering in a distributed simulation application" Proc. of IEEE Int. Phoenix Conference on Computers and Communications (IPCCC), April 1994, pp. 12-18.

[BASS90] Bassiouni, M. and Mukherjee, A. "Data compression in real-time distributed systems" Proc. of IEEE Global Telecommunications Conference (GLOBECOM), 1990, pp. 967-971.

[BASS89] 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.

[CHLA89] 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.

[IEEE93] IEEE Standard 1278-1993 "Information Technology- Protocols for Distributed Interactive Simulation Applications" available from the Institute for Simulation and Training, Orlando, 1993.

[LIU81] 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.

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

Table 2. ETHERNET Statistics (80 Simulators)

Measure	Version I	Version II
Max. # of trans. attempts	16	14
Avg. # of trans. attempts	2.45	2.25
Utilization	74.3%	72.0%
Avg. packet delay	5.4980 millisec	1.0297 millisec

Table 3. Performance of Modified ETHERNET (100 nodes)

Traffic Load (1000 packets/sec)	ETHERNET I		ETHERNET II	
	Mean Delay (millisecond)	% Lost packets	Mean Delay (millisecond)	% Lost packets
1.52	0.137	0.0	0.136	0.03
3.02	0.179	0.0	0.175	0.20
4.52	0.444	0.0	0.295	0.88
6.02	5.900	0.038	0.665	4.94
7.52	57.100	1.271	1.200	18.22

Table 4. Performance of Modified GBRAM (100 nodes)

Traffic Load (1000 packets/sec)	GBRAM I Mean Delay (millisecond)	GBRAM I % Lost packets	GBRAM II Mean Delay (millisecond)	GBRAM II % Lost packets
1.52	0.319	0.0	0.318	0.104
3.02	0.368	0.0	0.363	0.430
4.52	0.451	0.0	0.436	1.060
6.02	0.609	0.0	0.554	2.260
7.52	0.975	0.0	0.754	4.460

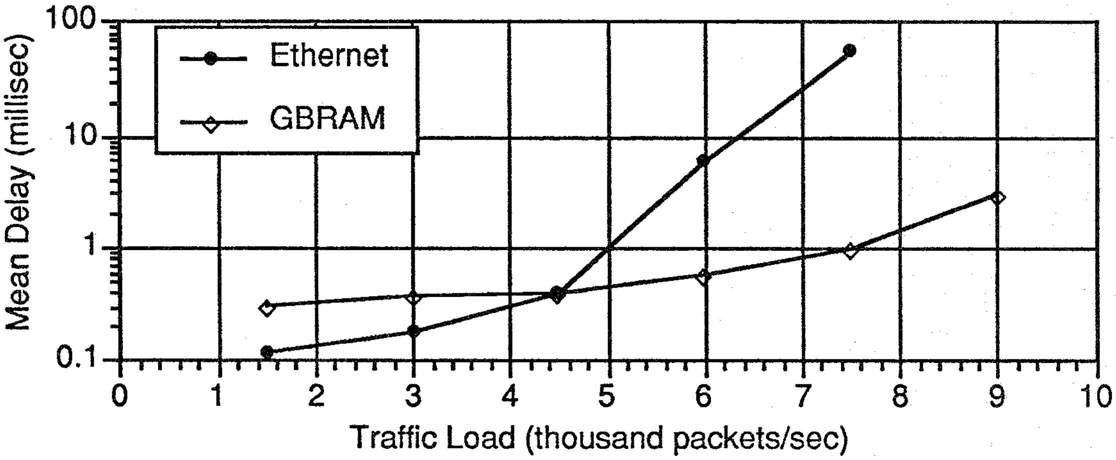

Fig. 1. Mean delay vs Traffic load (100 nodes)

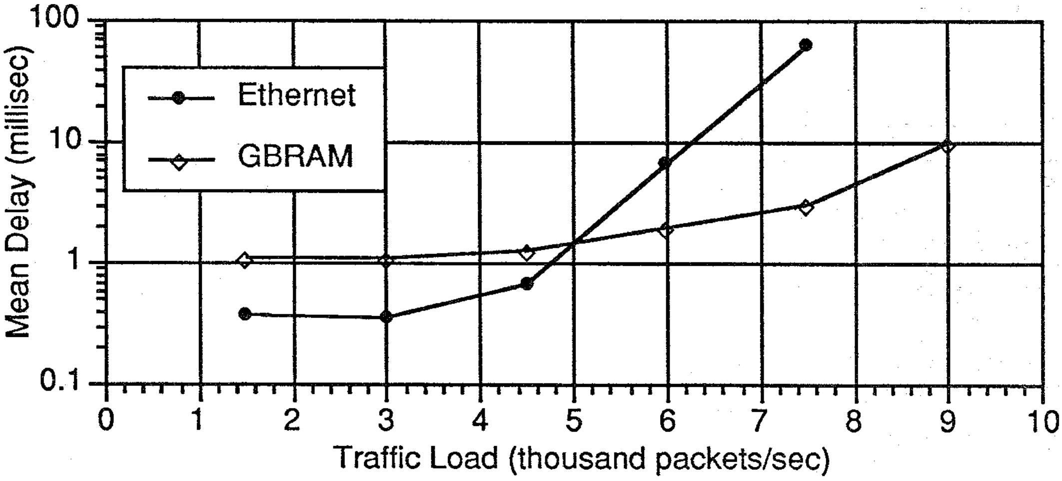

Fig. 2. Mean delay vs Traffic load (400 nodes)