

**A Network Access Protocol for Hard Real-Time
Communication Systems**

Özgür Ulusoy

BILKENT UNIVERSITY

**Department of Computer Engineering
and
Information Science**

Technical Report BU-CEIS-95-03

To appear in "Computer Communications"

A Network Access Protocol for Hard Real-Time Communication Systems

Özgür Ulusoy

Department of Computer Engineering and Information Science

Bilkent University

Bilkent, Ankara 06533, TURKEY

e-mail: oulusoy@bilkent.edu.tr, fax: (90)(312)266-4126

Abstract

Distributed hard real-time systems are characterized by communication messages associated with timing constraints, typically in the form of deadlines. A message should be received at the destination before its deadline expires. Carrier sense multiple access with collision detection CSMA/CD appears to be one of the most common communication network access schemes that can be used in distributed hard real-time systems. In this paper, we propose a new real-time network access protocol which is based on the CSMA/CD scheme. The protocol classifies the messages into two classes as ‘critical’ and ‘noncritical’ messages. The messages close to their deadlines are considered to be critical. A critical message is given the right to access the network by preempting a noncritical message in transmission. Extensive simulation experiments have been conducted to evaluate the performance of the protocol. It is shown that the protocol can provide considerable improvement over the virtual time CSMA/CD protocol proposed for hard real-time communication by Zhao et al. [11].

Keywords: Real-time communication systems, network access protocol, carrier sense multiple access, time-constrained messages.

1. Introduction

In this paper, we propose a network access protocol for hard real-time communication systems. Each message in a hard real-time communication system is associated with a timing constraint, typically in the form of a deadline. The message needs to be received at the destination before its deadline expires; otherwise (i.e., if it cannot be delivered within its deadline), it is considered as a lost message and is dropped from the system. The principal performance consideration is the fraction of messages that are received at the destination within their deadlines, rather than the average delay or throughput of the messages as in traditional communication systems. Examples of applications requiring real-time communication include the stock market trading, industrial process management, network management, air traffic control, and packetized voice transmission. In the stock market trading, for instance, a task can be initiated at a remote site to learn the price of a particular stock at that site. If the message carrying the result of the task cannot be delivered to the task’s origin in a specified time period, it should be ignored, because conditions in the stock market can change very fast.

The most common communication network used in distributed hard real-time systems is the multiple access network [11]. In a multiple access network, messages are transmitted on a shared channel. Only one message can be successfully transmitted over the channel at any

time. Among the various multiple access communication protocols, carrier-sense multiple access with collision detection (CSMA/CD) appears to be one of the most popular schemes. Under a CSMA/CD protocol, every station that wants to transmit a message must first listen to the channel to see whether any transmission is in progress [1]. If so, the transmission of the message is deferred until the end of the current transmission. Although the channel can be sensed, collisions of messages cannot be completely avoided due to the propagation delay of the channel. When a collision is detected, the stations abort transmission and the collided messages are retransmitted at a later time.

The real-time network access protocol proposed in this paper is based on the CSMA/CD scheme. A brief description of the protocol is as follows. A message close to its deadline is considered to be ‘critical’. A critical message has the right to preempt a ‘noncritical’ message in transmission. The idea here is to estimate if a message can afford to wait until the current transmission on the channel terminates; if not, the message in transmission is preempted. Thus, the message with approaching deadline is given a chance to arrive at the destination station within its deadline. The expectation here is that the preempted noncritical message can still have enough time to be delivered before its deadline expires. The protocol aims to increase the fraction of satisfied message deadlines.

A detailed simulation model of a real-time communication system was employed in the evaluation of the protocol. Two other network access protocols (the traditional CSMA/CD and the virtual time CSMA/CD) were also considered for comparison purposes. Various simulation experiments were conducted to study the relative performance of the protocols under different communication and real-time environments. The performance metric used was the fraction of messages transmitted within their deadlines. The proposed protocol was observed to provide significant performance benefits over the other protocols.

The next section summarizes the recent work on real-time communication network access protocols. Section 3 describes the proposed real-time network access protocol together with the communication system adopted in our study. The performance experiments and results are presented in Section 4. Section 5 concludes the paper and outlines future directions of this work.

2. Related Work

Recently, a considerable amount of work has been devoted to development and evaluation of protocols for time-constrained communication. Zhao et al. developed a virtual time CSMA/CD protocol [11] and a sliding window protocol [12] both implementing the minimum-laxity-first transmission policy. The virtual time CSMA/CD protocol was shown to perform better than the traditional CSMA/CD protocol. We compare the performance of our protocol to that of the virtual time CSMA/CD protocol in Section 4.3.

Strosnider et al. [9], and Yao et al. [10] provided extensions to the IEEE 802.5 Token Ring protocol to realize deterministic communication services. It was shown in [9] that the real-time performance of token ring networks can become very satisfactory by choosing a proper message size and a good priority assignment policy. The extended IEEE 802.5 protocol described in [10] approximates the optimal minimum-laxity-first transmission policy. Lee and Shen [3] described a new protocol for real-time communication in multiple token ring networks. The protocol attempts to minimize the late messages while maintaining a high

channel utilization. Ciminiera et al. [2] analyzed the behavior of timed token protocols in a double ring network. They also presented a method to compute the minimum guaranteed throughput for asynchronous messages. Shin and Hou [8] proposed an analytic evaluation model for time-constrained network access protocols. Ng and Liu [5] studied via simulation the real-time performance of token ring, token bus, and slotted ring protocols. It was shown that the slotted ring protocol performs much better than the other two protocols under various hard real-time communication environments. Schulzrinne [6] proposed some congestion control algorithms for reducing deadline violation in computer networks supporting real-time applications, and analyzed the performance of those algorithms. In [4], Malcolm and Zhao proposed a methodology for using the timed-token protocol in real-time communication. The timed-token protocol is a token-passing protocol in which each node receives a guaranteed share of the network bandwidth. The methodology provided in the paper can be used for testing the schedulability of real-time communication messages.

3. The Network Access Protocol

We assume that time is slotted and stations can only start transmitting messages at the beginning of each slot. Let τ denote end-to-end propagation delay of the channel. The length of a slot is considered to be equal to τ ; thus, a transmission at the beginning of a slot is recognized by all stations prior to the next slot. A collision can occur only between the messages that are transmitted at the same slot.

Each message M is characterized by its size S_M and deadline D_M . Laxity of a message specifies the maximum amount of time the transmission of the message can be delayed. Laxity of message M at time t is defined to be

$$L_M(t) = D_M - S_M - t$$

Messages are classified into two groups based on their laxities and a predefined threshold α . At a given time t , a message is considered to be *critical* if

$$L_M(t) \leq \alpha$$

otherwise (i.e., if $L_M(t) > \alpha$) it belongs to the class of *noncritical* messages. Laxities of messages are evaluated continuously, and thus, at any point in time a message from the class of noncritical messages can be transferred to the class of critical messages. If, during the evaluation of laxities, any message M in the message queue of any station is found to be late (i.e., $L_M(t) < 0$), then M is considered as a lost message and is dropped from the system.

α is a parameter of the protocol. The selection of α will be discussed in Section 4.2. It should be large enough to let a message be transmitted within its deadline when the channel is allocated to that message, and also be small enough to keep the size of the set of critical messages small at any moment in time.

When an idle channel is sensed, the transmission of a critical message can start at the beginning of the next slot. On the other hand, a noncritical message has to wait for the duration of the next slot, and its transmission starts at the beginning of the slot after next (if the channel is still idle). This scheme aims to give privilege to critical messages in accessing the channel and to prevent the collision of critical messages with noncritical ones.

The transmission of a critical message starts by broadcasting a short control message, called *notifier*, to all other stations to notify them that the channel is going to be occupied by a critical message. The notifier is actually a special bit pattern recognized by all stations. It is used to ensure that a critical message in transmission cannot be preempted.

The proposed real-time network access protocol works as follows:

- When a new message arrived is characterized as critical or when a noncritical message becomes critical, if the channel is idle the message is sent. Otherwise (i.e., the channel is busy) if the message currently being transmitted is not critical, it is preempted¹ to let the critical message be sent.
- Transmission of a noncritical message from a station is not possible as long as the channel is busy. Also, the message is not sent right after the channel becomes idle. The station should wait until the end of the next slot and if the channel is still sensed to be idle at that moment the message is sent.
- Following a collision, a noncritical message is suspended for the duration of a slot and if that slot is not occupied by a critical message² the noncritical message is retransmitted with probability P_i .
- If a critical message collides³ it is retransmitted immediately with probability P_i .

4. Performance Experiments

4.1. Simulation Model

A simulation model was developed to evaluate the performance of the proposed protocol. The basic characteristics of the model were set to be similar to that of Zhao et al. [11] to make our results compatible with theirs. The simulation program was written in CSIM [7], which is a process-oriented simulation language based on the C programming language.

The assumptions of our simulation model can be listed as follows:

- Message arrivals to each station follow a Poisson process with an average arrival rate of λ messages/time-unit.
- The size of a message (i.e., the total time needed to transmit the message) is exponentially distributed with a mean of S time units. The laxity of a message M is chosen uniformly from the interval $(0, F * S_M)$ where the parameter F is called the laxity factor and S_M specifies the actual size of M . Total number of stations in the communication system is specified by the parameter N . The system load L is determined by the following formula:

$$L = \lambda * S * N$$

¹To realize preemption, the transmission of the message can be interfered by a short noise burst on the channel, so that its station will stop sending the message.

²Note that, a preemption is also considered as a collision, and following a preemption only a critical message is transmitted.

³A critical message can only collide with another critical message.

τ	Propagation delay	1 time unit
S	Mean message size	20 time units
N	Number of stations	20
$1/\lambda$	Mean interarrival time	800 time units
F	Laxity factor	3
P_i	retransmission probability after a collision	0.5

Table 1: Simulation Model Parameter Values

Default values assumed for the system parameters are provided in Table 1. Those values produce a system load of $L = 0.5$. The performance experiments presented in the following subsections discuss the results obtained under different system loads by varying the related parameter values.

- The size of the buffers used to hold messages at stations are infinite; thus, no message loss is experienced due to buffer overflows.
- The communication channel is assumed to be error-free. Therefore, there is no loss of messages and no retransmission is required. Issues such as reliability, fault recovery, and retransmission in real-time communication systems are beyond the scope of this paper.

The performance metric considered in our evaluations is *success_ratio* which is determined by the following formula:

$$success_ratio = \frac{messages_transmitted}{messages_transmitted + messages_dropped}$$

where *messages_transmitted* denotes the number of messages successfully transmitted, and *messages_dropped* denotes the number of late messages dropped from the system.

In the following experiments, the ‘independent replication’ method was used to validate the results by running each configuration 25 times with different random number seeds and using the averages of the replica means as final estimates. Each run continued until 1000 messages were generated at each data station. 90% confidence intervals were obtained for the performance results. The width of the confidence interval of each data point is within 4% of the point estimate. The mean values of the performance results were used as final estimates. The following subsections discuss only statistically significant performance results.

4.2. Sensitivity of the Protocol to Parameter α

In this experiment, we evaluated the sensitivity of the protocol to the value of α . The range of α values tested under different system loads was (1, 30). Figure 1 presents the *success_ratio* results for particular system loads $L = 0.1, 0.5, \text{ and } 1.0$ (these system loads were obtained by setting the average interarrival time of messages ($1/\lambda$) to 4000, 800, and 400 time units, respectively)⁴. At all load levels tested, the protocol’s reaction to the change

⁴Varying the mean message size (S) or the number of stations (N) are the other possible ways of changing the system load.

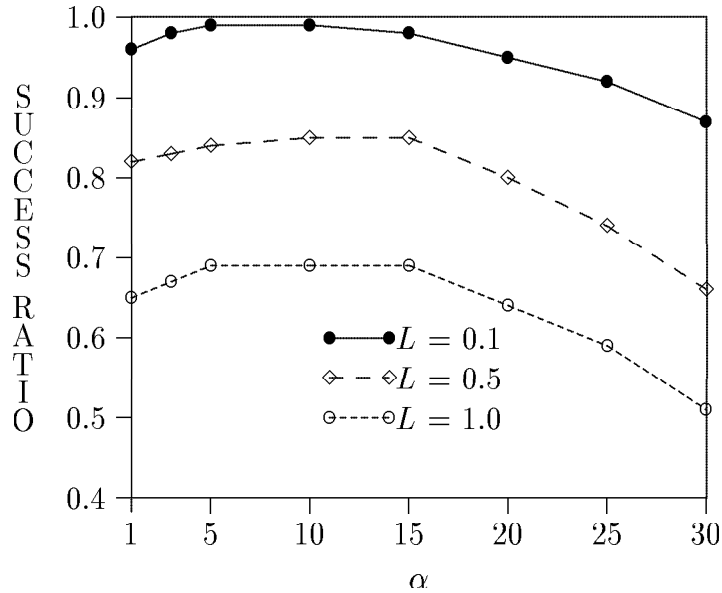


Figure 1: Performance of the protocol for different settings of α

in α is similar; increasing the value of α up to a certain point helps more messages meet their deadlines, but beyond that point the increase in α results in worse performance. For very small values of α (e.g., 1), when a message is designated as ‘critical’, the laxity of that message will likely be too small to complete its transmission before the deadline expires. The value of α needs to be as large as at least a few times the propagation delay (i.e., a few times the length of a slot), because before starting its transmission it is necessary first to preempt the currently transmitting message (if it is noncritical) and then to broadcast a notifier. As presented in the figure, the best performance is obtained when α takes a value in between 5 and 15. For larger values of α , increasing α results in a decrease in the performance. This can be explained by the fact that the number of critical messages in the system increases as the value of α is increased. Therefore, the probability of collision of critical messages also increases (since the source stations attempt to send them promptly), and the collisions cause more deadlines to be missed. The performance results provided in the following subsection were obtained by setting α to 10.

4.3. Sample Performance Results

We conducted various performance experiments to study the effects of systems parameters on the performance of the protocol. In this subsection, we present those experimental results that are most interesting and best illustrate the performance of the protocol. Two other protocols selected to be used for comparison purposes are:

- CSMA/CD: The traditional carrier sense multiple access protocol with collision detection. This protocol does not make use of timing constraints of messages in scheduling accesses to communication channel.
- VTCSMA/CD-L [11]: The virtual time CSMA/CD-L protocol implements the minimum-laxity-first real-time transmission policy. In this protocol, each station maintains two clocks: a real-time clock and a virtual time clock. Whenever a station finds the channel

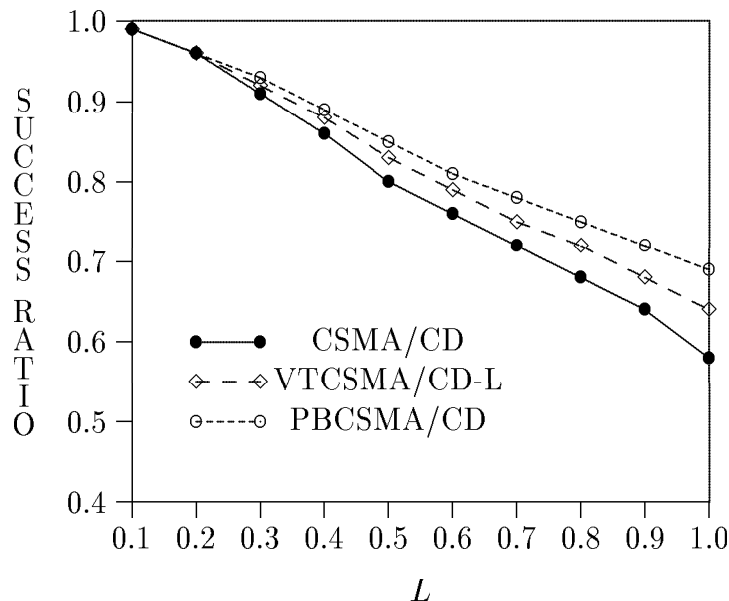


Figure 2: Performance of three protocols under varying load levels

to be idle, it resets its clock to equal the real clock. The virtual clock then runs at a higher rate than the real clock. A message is transmitted if the time on the virtual clock is equal to the LS value of the message which specifies the latest time to send that message. This protocol was shown to perform better than CSMA/CD in a wide range of real-time communication environments [11].

Figures 2 and 3 display the performance results of three protocols under varying load levels and different laxity sizes, respectively. PBCSMA/CD in both figures stands for our Preemption Based CSMA/CD protocol. In figure 2, the system load L is varied from 0.1 to 1.0 in steps of 0.1 by varying the value of the mean interarrival time ($1/\lambda$) from 4000 time units to 400 time units in steps of 400. Under low levels of message load all the protocols perform equally well. The difference between the performances of protocols starts to appear with a system load value of 0.3. Protocol VTCSMA/CD-L provides a considerable improvement over traditional protocol CSMA/CD especially under high levels of message load. This improvement is due to involving timing constraints of messages in scheduling channel access requests. The figure also shows that further improvement in real-time performance could be possible by employing our protocol PBCSMA/CD. The advantage of our protocol over VTCSMA/CD-L is that a message close to its deadline is given the right to access the communication channel regardless of whether the channel is currently busy or idle. Therefore, the message will have a chance to meet its deadline while the preempted message can still have enough time to be transmitted within its deadline. This simple policy results in more messages to satisfy their deadlines as displayed in the figure. With VTCSMA/CD-L, messages with very small laxities do not have a chance to be transmitted if the channel is currently being used, and thus they simply miss their deadlines. The performance results also show that the difference in performances of protocols PBCSMA/CD and VTCSMA/CD-L becomes greater as the message load increases.

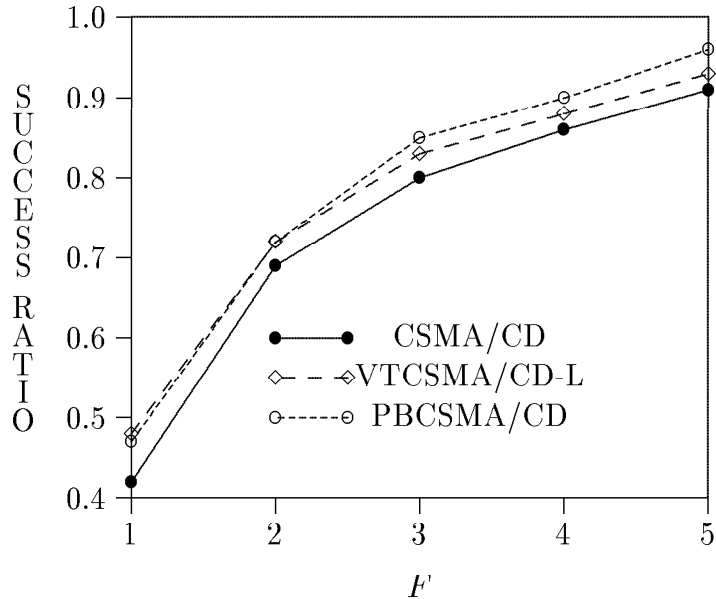


Figure 3: Performance of three protocols for different laxity sizes

The laxity of a message is the maximum length of time the message can wait to be transmitted and still satisfy its deadline. The parameter F , called laxity factor, determines the ratio of the laxity of a message to its transmission time. Message laxities are chosen uniformly from the interval $(0, F * S)$. Figure 3 presents the performance results of an experiment in which F was varied from 1 to 5 in increments of 1. The average system load value was set to $L = 0.5$ ($1/\lambda = 800$). The small values of F correspond to tight message deadlines. Not surprisingly, the performances of all protocols become better as the assigned deadlines get looser. Protocol CSMA/CD is again outperformed by both protocols VTCSMA/CD-L and PBCSMA/CD for different settings of parameter F . It can also be observed from the figure that PBCSMA/CD is preferable to VTCSMA/CD-L when the messages are associated with large laxities. For very tight deadlines (i.e., with $F = 1$ or 2), on the other hand, PBCSMA/CD cannot provide any performance advantage over VTCSMA/CD-L, because it is likely that multiple messages might be designated as critical at any time and although they have priority over noncritical messages as in VTCSMA/CD-L, they contend with each other for the channel access. Thus, a critical message is not always guaranteed to meet its deadline.

4.4. Impact of Preemption

The primary feature of our protocol PBCSMA/CD is that a critical message is allowed to preempt a noncritical message in transmission. In this subsection, we investigate the performance impact of the preemption feature of PBCSMA/CD in more detail.

The following performance metric is used to determine the overhead of preemption:

$$preempt_oh = \frac{\text{Total number of slots wasted due to preemption}}{\text{Total number of slots simulated}}$$

Figure 4 displays the measured preemption cost under different system loads. As the system load gets higher, more messages compete for channel access leading to an increase in the

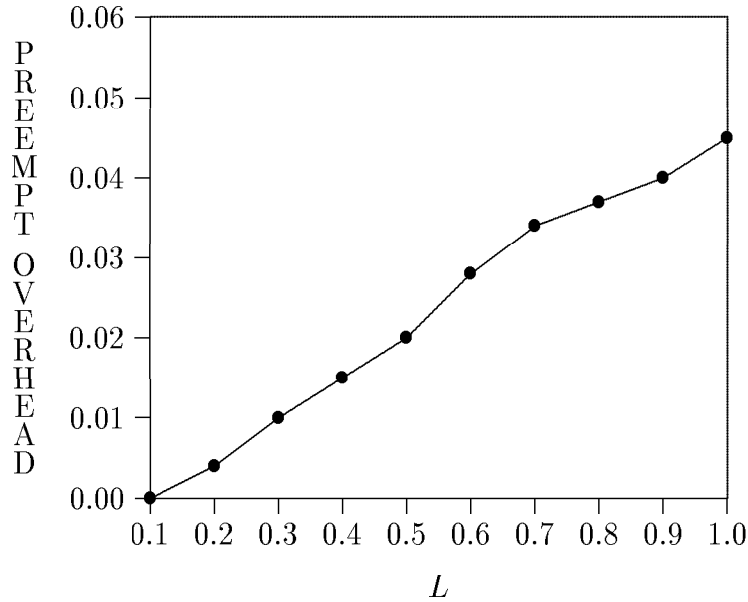


Figure 4: *preempt_oh* (fraction of wasted slots due to preemption) as a function of system load.

average waiting time of messages. Delaying a message before transmission narrows the distance to its deadline that might cause the message to become critical. An increase in the number of critical messages would mean more preemptions experienced and larger number of slots wasted due to those preemptions.

The concern that needs to be addressed at this point is that how the overhead of preemption observed for protocol PBCSMA/CD affects the performance of the system in terms of the satisfaction of message deadlines. To determine the performance impact of preemptions, we kept track of both the number of missed deadlines due to preemptions, and the number of satisfied deadlines with the help of preemption decision. The results are provided in Figure 5 as a function of the system load. As we just discussed, the number of preemptions increases as the system load becomes higher. Some of the preemptions might cause the preempted noncritical message to miss its deadline. However, the number of such messages is very few as can be seen from the figure. The majority of the preempted messages still have enough time to be delivered before their deadlines expire. On the other hand, most of the preemptions result in the transmission of critical messages within their deadlines. Thus, the performance benefits of preemption clearly outweigh its costs. As the level of system load increases, more messages benefit from the preemption.

5. Discussion

A communication system that supports time-critical applications is known as a real-time communication system. In this paper, we have proposed a network access protocol based on the CSMA scheme for real-time communication systems. The protocol classifies the messages into two classes as critical and noncritical messages. A message close to its deadline is considered to be critical, and it is given the right to access the network by preempting a noncritical message in transmission. The performance of the protocol has

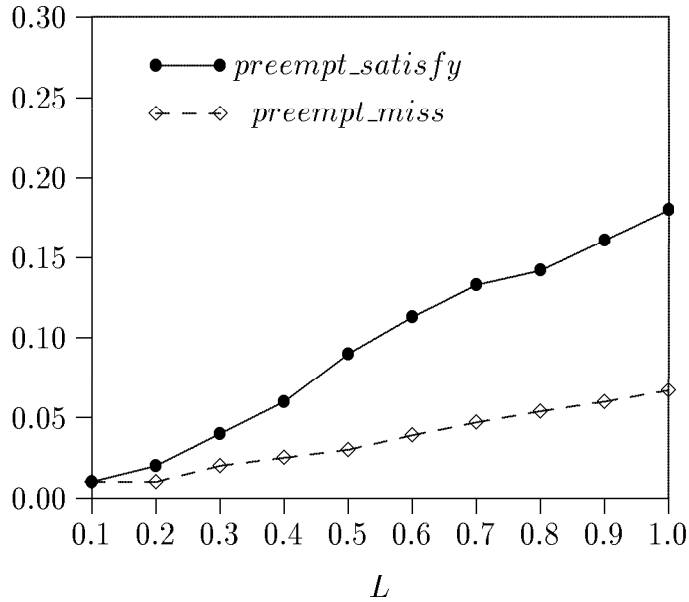


Figure 5: $preempt_miss$ (the fraction of missed deadlines due to preemption over all messages generated) and $preempt_satisfy$ (the fraction of satisfied deadlines with the help of preemption over all messages generated) as a function of system load.

been evaluated via a detailed simulation model. The simulation experiments have shown that the proposed protocol performs better than the virtual time protocol VTCSMA/CD-L (a popular real-time network access protocol proposed in [11]) under most of the conditions tested. The primary factor leading to that result is that VTCSMA/CD-L has no mechanism to favor critical messages. Another advantage of the new protocol over VTCSMA/CD-L is that it is easier to implement, because the overhead of maintaining a virtual clock at each station is eliminated by this protocol.

As a future work, we are planning to extend our work in several ways:

- The proposed protocol can be adopted to ‘soft’ real-time communication systems where all messages are sent regardless of whether they have missed their deadlines or not. The late messages in such an environment can be considered to belong to the class of noncritical messages.
- The priority granularity in our protocol is limited to two; i.e., only two priority levels are considered. The protocol can be extended to consider multiple priority levels.⁵ In that extension, the notifier preceding each message can contain the priority of the message, so that each station becomes able to preempt the transmitting message if a message with higher priority is waiting to be transmitted from that station.

⁵Deadlines or laxities of messages can be mapped into priorities.

References

- [1] Bux, W. 'Local-Area Subnetworks: A Performance Comparison', *IEEE Transactions on Communications*, Vol 29 No 10 (October 1981) pp 1465-1473
- [2] Ciminiera, L., Montuschi, P., Valenzano, A. 'Some Properties of Double-Ring Networks with Real-Time Constraints', *Proc. Real-Time Systems Symposium*, (1989) pp 360-368
- [3] Lee, Y., Shen L. 'Real-Time Communication in Multiple Token Ring Networks', *Proc. Real-Time Systems Symposium*, (1990) pp 146-154
- [4] Malcolm, N., Zhao, W. 'The Timed-Token Protocol for Real-Time Communications', *IEEE Computer*, Vol 27 No 1 (January 1994) pp 35-41
- [5] Ng, J.K.Y., Liu, J.W.S. 'Performance of Local Area Network Protocols for Hard Real-Time Applications', *Proc. Distributed Computing Systems*, (1991) pp 318-326
- [6] Schulzrinne, H.G. *Reducing and Characterizing Packet Loss for High-Speed Computer Networks with Real-Time Services*, Computer Science Technical Report 93-54, University of Massachusetts, (May 1993)
- [7] Schwetman, H. 'CSIM: A C-Based, Process-Oriented Simulation Language', *Proc. Winter Simulation Conference*, (1986) pp 387-396
- [8] Shin, K.G., Hou, C. 'Analysis of Three Contention Protocols in Distributed Real-Time Systems', *Proc. Real-Time Systems Symposium*, (1990) pp 136-145
- [9] Strosnider, J.K., Marchok, T., Lehoczky, J. 'Advanced Real-Time Scheduling Using the IEEE 802.5 Token Ring', *Proc. Real-Time Systems Symposium*, (1988) pp 42-52
- [10] Yao, L., Zhao, W. 'Performance of an Extended IEEE 802.5 Protocol in Hard Real-Time Systems', *Proc. IEEE INFOCOM*, (1991) pp 469-478
- [11] Zhao, W., Ramamritham, K. 'Virtual Time CSMA Protocols for Hard Real-Time Communication', *IEEE Transactions on Software Engineering*, Vol 13 No 8 (August 1987) pp 938-952
- [12] Zhao, W., Stankovic, J.A., Ramamritham, K. 'A Window Protocol for Transmission of Time-Constrained Messages', *IEEE Transactions on Computers*, Vol 39 No 9 (September 1990) pp 1186-1203