Deterministic Best Effort and Planning Based Protocols for Real-time Communication in Multiple Access Networks

Samphel Norden S. Balaji
G. Manimaran C. Siva Ram Murthy*
Department of Computer Science and Engineering
Indian Institute of Technology, Madras 600 036, INDIA

Abstract. In a distributed real-time system, tasks scheduled on different nodes, communicate by message passing mechanisms. In such a system, where messages are characterized by service times and deadlines, the timely delivery of messages before their deadlines, is of prime importance. In this paper, we consider the problem of time-constrained communication over a multiple access network. As a solution to the problem, we propose two classes of protocols, namely, best effort based and dynamic planning based, catering to different application semantics. These protocols are based on CSMA/CD protocol with deterministic collision resolution, and use a concept of *message deferment*, which is based on laxity[2]. The planning based protocol performs an admission test for accepting a message into the system. We also propose a generalised protocol which can realise both the best effort and planning based protocols. We study the effectiveness of the proposed protocols through extensive simulation studies, comparing them with well known protocols. The simulation results indicate the superiority of the proposed protocols.

1 Introduction

In a distributed real-time system, timely execution of tasks and delivery of messages across communication networks, within their associated deadlines, are of prime importance [7]. These time-constrained messages are the basis for applications that operate in a real-time environment. Applications are either of the hard real-time category, where message loss is not tolerated, or of the soft real-time category, where some amount of message loss is allowed (as in multimedia applications). An important example of an environment with hard real-time requirements is an integrated network in a manufacturing shop [8].

Numerous real-time local area network (LAN) communication protocols have been devised to transmit messages in distributed real-time environments. A survey of the most representative ones can be found in [3]. The most common communication network is the multiple access network. In this type of network, nodes transmit messages via a shared channel. Only one message can be successfully transmitted over the channel at any time. A collision occurs if the transmission of two or more messages overlap in time. Based on how collisions are handled, the multiple access protocols can be broadly divided into

* Corresponding Author. Email : murthy@iitm.ernet.in

[2] Laxity is the maximum amount of time that the transmission of a message can be delayed.

three categories: Inference avoiding protocols such as CSMA based protocols, which do not take the past history of the channel into account; Inference seeking protocols, which make inferences based on collision history; and the Deterministic or Collision-free protocols such as Time Division Multiple Access (TDMA), which seek to avoid collisions. This paper proposes protocols belonging to the Inference avoiding category.

Every message in a real-time system is characterized by its service time requirement and its deadline. In a distributed real-time system, the scheduling mechanism should not only ensure guarantees on message delivery, but should also be adaptive to the dynamic nature of message traffic. Thus, in order to ensure timely delivery of messages, the protocol must ensure bounded channel access delay. An adaptive protocol should be able to dynamically prioritize messages, based on the degree of criticality of the messages. The proposed protocols attempt to capture all these features.

The outline of the rest of this paper is as follows. Section 2 provides the background for prior research work in this area, and also highlights the inadequacies of these protocols. Section 3 describes the motivation for a new class of protocols, introduces a protocol of best effort nature, and discusses its performance. Section 4 presents the simulation results for this category. Sections 5 and 6 present the guaranteed delivery and the generalised protocols, respectively. Finally, in Section 7, some concluding remarks are made.

2 Prior Work in Real-time LAN Protocols

Previously proposed real-time LAN protocols attempt to transmit messages before their associated deadlines. Some of the earlier protocols give guarantees on timely message delivery, while others can only provide best effort service. A few of these protocols assume that messages with hard real-time requirements are periodic, and should be provided with guaranteed service strategies, while aperiodic messages (considered non-critical) are provided with best effort service. The key issues to be considered while analyzing these protocols, are whether these protocols can guarantee timely delivery of messages (predictability), and whether they capture the dynamic nature of traffic by assigning priorities in a non-static manner, without violating guarantees given to earlier messages.

The Virtual Time CSMA (VTCSMA) [2] and Preemption Based CSMA (PBCSMA) protocols [4] cannot bound the worst case channel access time, as they use a probabilistic approach for collision resolution. The Waiting Room protocol [1], offers a prohibitively large worst case channel access delay, and has high run-time overhead. The protocol proposed by Oh, et.al [6], attempts to address the issue of ensuring predictability for message transmission. Here the system knows the future messages i.e. *a priori* knowledge is required about the messages to be scheduled. This protocol assumes pre-run-time scheduling for messages that are critical, which implies that the channel reservation information is provided before run-time. Hence, the dynamic nature of real-time traffic cannot be captured by this approach. Further, the authors point out that the overhead involved for building such a predictive system may well offset the performance improvements.

2.1 CSMA/CD - DCR Protocol : Description

From the above discussion, it can be observed that there is a need to develop protocols that can ensure a guaranteed worst case channel access delay which is not prohibitive. This would allow schedulability analysis to be performed while admitting messages into the system. In order to give a bounded worst case channel access delay and to ensure predictability, the Deterministic Collision Resolution Protocol (CSMA/CD-DCR) [10] was proposed with an approach to resolve collision in a deterministic way by traversing all the nodes in some implicit order. This protocol assumes messages to be single packets.

In the DCR protocol, when a node has a packet to transmit, it does the following :
1. The node senses the channel.
2. If no collision occurs, the message is transmitted successfully.
3. If a collision occurs, all nodes enter the DCR collision resolution mode where transmission rights are given to nodes based on an implicit order.

One order of traversal is the pre order traversal of a binary tree, as in [5]. Here, the nodes of the LAN are statically mapped to the vertices of a binary tree. If there is a collision, all the nodes enter the collision resolution mode, where each node will get its turn to transmit based on its position in the tree. This methodical approach to collision resolution bounds the worst case channel access time. We adopt this method for the subsequent protocols. The binary tree is called the Collision Resolution (CR) Tree for our discussion.

3 Development of the New Class of Protocols

3.1 Motivation for Our Work

Several applications demand transmission of time-constrained messages which are composed of multiple packets, where a packet is basic unit of data to be transmitted within a deadline. If the DCR protocol is extended to transmit multi packet messages, the performance (in terms of number of messages guaranteed) would be poor. This is a consequence of allowing only one packet of a message to be transmitted in an epoch[3]. A large message would require several epochs for transmission. Hence the deadline would have to be extremely large for successful transmission of the message. This would lead to a majority of the messages being dropped and would reduce the performance of the system. These limitations were the motivations for developing reservation based protocols for messages with multiple packets. In the rest of this section, we propose reservation based protocols which use *notifiers* to reserve the channel for message transmissions.

3.2 System and Channel Model

1. A population of N nodes share the channel.
2. There is a finite channel propagation delay.
3. Channel access is slotted. Transmission can start only at the beginning of a slot.
4. The transmission policy is modified, by splitting channel access into periods of *notification* and *transmission*. Before transmitting a message, node i transmits a notifier

[3] The time elapsed from the time of collision detection to the time for completion of CR, including the time spent in resolving collision and transmitting messages.

indicating the number of service slots required by its message (C_i) and its deadline (D_i).
5. Carrier sensing is instantaneous. Each node is capable of detecting an idle slot, a successful transmission and a collision.
6. Every node participates in collision resolution, irrespective of whether it has a message to send.

3.3 MDCR : Modified DCR Protocol

The first protocol is an extended version of the DCR protocol (known as MDCR) based on notifiers supporting transmission of multi packet messages. A static mapping of nodes in the LAN to vertices of a binary tree is done. The MDCR algorithm at a node proceeds as follows :
1. The node i senses the channel, and attempts to transmit the notifier (C_i, D_i) of a message.
2. If no collision occurs and the message notifier is transmitted successfully, the node i begins transmission of the message provided $C_i \leq D_i$, else message is dropped. During transmission, no other node tries to transmit over the channel.
3. If notifier transmission fails, all nodes enter the collision resolution mode as in DCR.
4. In the collision resolution mode, the tree is traversed in pre-order where the current node transmits a notifier for its message(if any) and then actually transmits the message if $C_i \leq D_i$, else the message is dropped.

3.4 Development of Laxity Based DCR Protocol

Though MDCR allows transmission of multi packet messages, it has several limitations. The main drawback of this approach is that the protocol transmits messages strictly based on the initial mapping of nodes of the LAN to the vertices of the collision resolution tree, thus enforcing a static priority order among nodes. Hence, MDCR would allow messages with larger laxity (smaller priorities) to get transmission rights over more critical messages with smaller laxity (higher priorities), due to the static order of tree traversal. This could lead to priority inversion.

3.5 LDCR Protocol Description

To overcome the above limitations, we propose a new protocol, Laxity based DCR (LDCR) protocol, which dynamically assigns priorities to messages based on laxity (higher priorities are assigned to messages having smaller laxity) and also allows multi packet message transmission. Laxity (L_i) of message m_i at time t, is the maximum amount of time by which the transmission of a message can be delayed : $L_i = D_i - C_i - t$ where C_i and D_i are the service time and deadline of message m_i, respectively.

The LDCR protocol works based on a notion of *message deferment*, by which we mean that a message with higher laxity is deferred to allow messages with lower laxity to be transmitted, thus preventing the potential priority inversion. In LDCR, the static mapping of nodes of the LAN to the vertices of collision resolution (CR) tree is similar to DCR-P as discussed in [5]. The manner in which this mapping is done is left to the designer. The traversal of the CR tree is in pre order, that is when a collision occurs, the next three slots are reserved for the root, the left subtree, and the right subtree, respectively. Each node maintains a queue of messages sorted on the increasing order of laxity. In the actual implementation, the queues are maintained as a *delta list* for efficient storage of messages involving minimal overhead to update their laxity. In a delta list based message queue, the laxity of a message is stored relative to the laxity of its predecessor in the queue. Thus updating the delta list, as time progresses, involves minimal overhead, as only the laxity at the head of the queue is to be modified. The updation of the queue due to a newly arrived message involves inserting the message at appropriate position, assigning laxity to it with respect to its predecessor, and modifying the laxity of its successor. Messages with negative laxity are removed from the queue.

The LDCR protocol consists of three phases, namely, contention mode, collision resolution (CR) mode, and least laxity first (LLF) mode, which are described below. This combination of CR mode and LLF mode defines an epoch in the LDCR protocol as contrasted to the DCR pre order and MDCR protocols wherein an epoch is the CR mode alone.

1. **Contention Mode (when channel is idle)**
   (a) Each node examines the message at the head of its queue. If the laxity of the message is negative, the message is dropped.

   (b) Otherwise, it attempts to transmit notifier for the message.

   (c) If the notifier transmission is successful, the message is transmitted.

   (d) If a collision occurs during notifier transmission, all nodes enter into CR mode by traversing the CR tree in pre order.

2. **Collision Resolution Mode**
   (a) When a node n_i gets its turn in the CR mode, it transmits its notifier. Thus every other node is informed of the message requirements of node n_i.

   (b) If the laxity of a message (first message at queue of n_i) is less than a threshold (but non-negative), then the message is transmitted, else the message is deferred and pre order traversal continues. If two or more messages have the laxity less than or equal to the threshold value, the preference is given to message with shorter service time. This minimizes the time that the channel is blocked, thereby improving the chances of successful transmission for subsequent messages.

   (c) Even after the CR tree is completely traversed, if some deferred messages are still to be transmitted, the protocol enters into the LLF mode.

3. **LLF Mode**
   - In the LLF mode, all the deferred messages are transmitted, based on the laxity order of messages among nodes. This phase is required to enable those messages, that arrived in the current epoch, to be transmitted before the next epoch. By doing this, the overhead incurred due to collision between messages of two different epochs is eliminated.

4. If all the messages are either transmitted or dropped, the next epoch begins.

Note that the threshold value mentioned in step 2(b) is fixed, in our protocol, to be one. The importance of this value is brought out in a subsequent section, when comparing this protocol with another representative protocol.

4 Simulation Studies

To study the effectiveness of the proposed protocols, we have conducted extensive simulation studies. For each simulation run, 10000 messages were generated for the entire system. For the performance results 95% confidence intervals were obtained.

4.1 Simulation Model and Performance Metrics

1. N : Number of nodes on the network ranging from 5 to 30.
2. M : Average message length (or service time), 7 to 20 slots.
3. α : laxity parameter (2 to 25), which decides the deadline of messages, directly proportional to deadline. The deadline is uniformly chosen in the interval $[C_i, \alpha * C_i]$.
4. λ : Mean message arrival rate at a node. Message arrival at each node follows Poisson distribution with mean arrival rate λ.
5. L : Load on the network (0.1 to 1) is defined as the product of number of nodes (N), mean message length (M) and the arrival rate, i.e., $L = (\lambda * M * N)$.

The performance metrics are as follows :

1. **Success Ratio** (SR) : This is defined as the ratio of the total number of messages transmitted successfully to the total number of messages that arrived in the system. SR is an important metric which reflects the throughput of the network.
2. **Effective Channel Utilization** (ECU) : This is the ratio of the time slots used for successful transmission to the number of time slots that elapsed until all messages were transmitted or dropped. A higher value of ECU would imply that the channel utilization, due to successful message transmission, is large compared to the net overhead due to collision and notifier transmission.
3. **Normalized Transmission Length** (NTL) : This is the ratio of the mean length of the transmitted messages to the mean length of the generated messages. The closer its value to one, the less is the bias towards longer or shorter messages. In other words, when the NTL offered by a protocol is one, the protocol behaves independent of message service time. Owing to space constraints, we do not show any graphs involving NTL.

4.2 Results and Discussion

In this section, we compare the performance of the proposed protocols (MDCR and LDCR) with a recently proposed protocol called PBCSMA. The PBCSMA protocol [4], is chosen for comparison because it can support multi packet messages, and has been reported to perform better than the VTCSMA and CSMA based protocols. Note that the MDCR, LDCR, and PBCSMA are *best effort* protocols, which means that they admit all the messages arriving in the system, without guaranteeing that these messages will be transmitted successfully. In the simulation plots, these protocols are labelled as with an extension BE to indicate "Best Effort".

1. **LDCR is better than PBCSMA :** The reasons, why LDCR is an improvement over PBCSMA, are as follows. In the case of the notifier of a critical message colliding

with the notifier transmission of another critical message, PBCSMA will retransmit the interrupted message with certain probability. This results in many idle slots which in turn leads to missing of message deadlines. Whereas in LDCR-BE, by forcing the threshold for critical messages to be one, the protocol restricts the number of critical messages; in the event of collision, the arbitration is done as per the CR tree allowing one of the nodes to transmit in subsequent slots without introducing idle slots. Further, in LDCR, when two critical messages have laxity less than the threshold value, the smaller message is transmitted which improves the chances of subsequent messages. The superiority of the LDCR protocol over PBCSMA, can be clearly seen, from all the simulation graphs.

Another interesting observation about PBCSMA, also made by the author in [4], is that the SR will degrade when the threshold is increased beyond a certain value, as the number of critical messages (with varying degree of criticality) increases. This results in increased number of collisions between these messages, thereby reducing the SR. In addition, it may also result in priority inversion. The increased number of collisions will also result in a lower ECU value. This situation will not arise in LDCR, as it defers the non-critical (large laxity) messages to a separate mode (LLF), thus avoiding collisions between messages with high and low laxity. This will reduce the number of overall collisions. These deferred messages are transmitted in the LLF mode, using laxity based ordering. This systematic way of collision resolution and message deferment will result in improved SR and ECU.

2. **LDCR is better than MDCR :** The poor performance of MDCR is due to its blind transmission of messages as per the static position of nodes in the CR tree without taking into account the laxity of messages among the nodes. Thus higher priority (lower laxity) messages will be treated in the same way as lower priority (large laxity) messages. This will lead to missing of message deadlines. Further, a large number of messages with tighter laxity, having small service times, could be dropped if a message with larger laxity, and having larger service time, is transmitted.

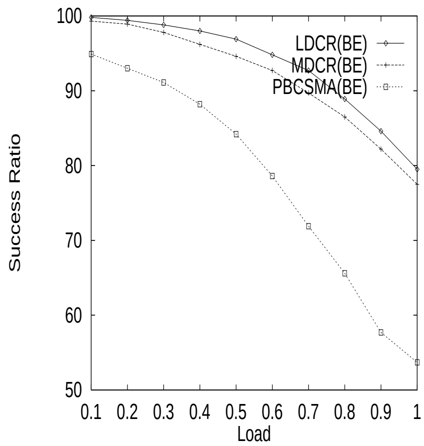

Fig.1a Effect of load on SR

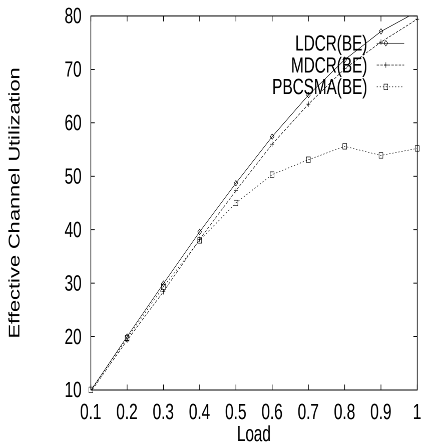

Fig.1b Effect of load on ECU

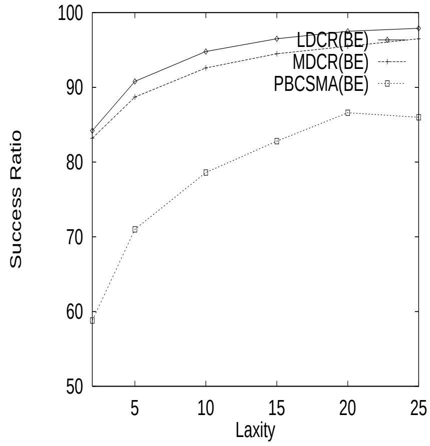

Fig.2a Effect of α on SR

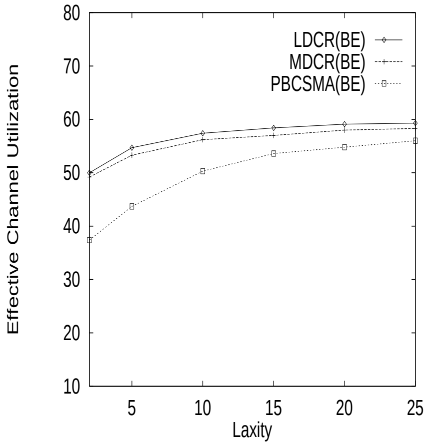

Fig.2b Effect of α on ECU

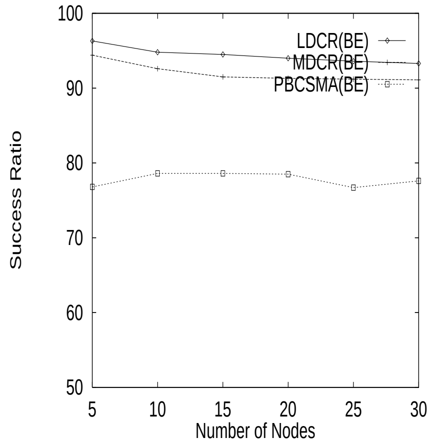

Fig.3a Effect of N on SR

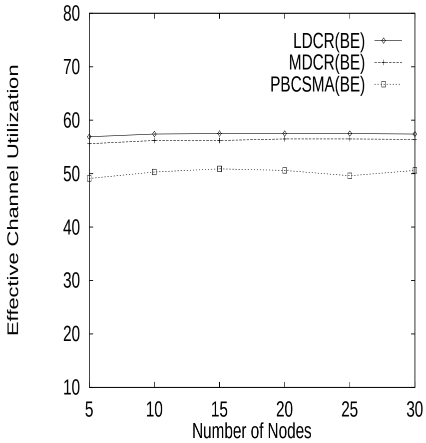

Fig.3b Effect of N on ECU

From the Figs.1a-3a, which study the effect of the different input parameters, on SR, we can conclude the following :

1. SR vs Load : In Fig. 1a, as load increases, there is a predictable fall in SR values for all the three protocols as more number of messages are generated and more number of collisions, thus reducing SR.

2. SR vs α : In Fig. 2a, as the laxity factor increases, the deadline associated with every message increases, resulting in an increase in the number of messages that are transmitted successfully, hence increasing SR.

3. SR vs N : In Fig.3a, as the number of nodes increases, the SR falls. This is due to an increase in the number of messages, as well as a larger epoch time, which leads to more messages missing their deadlines.

An interesting observation from the graphs is that the ECU is not always influenced by the SR. Normally, one would expect that as SR decreases, the ECU will also decrease since less number of slots are used for successful transmission. Sometimes, this relation may not hold. An example scenario in which even if less number of messages are transmitted which have larger service times than the dropped messages (see Fig.1a and 1b), the ECU can still increase.

5 Dynamic Planning Based Protocol

As mentioned earlier, the protocols described in the previous sections are of best effort nature since they do not perform any admission test for guaranteeing the transmission of messages. Many hard real-time applications require *predictability* of message transmission, which means that, once a message is admitted into the system, it will always be transmitted before its deadline. This motivated the development of planning based protocols for guaranteeing the transmission of messages, on-line.

In a dynamic planning based protocol, when a message arrives at a node, the message scheduler at the node, dynamically determines the feasibility of scheduling this new message without jeopardizing the guarantees that have been provided for the previously scheduled messages. Thus, for predictable message transmission, schedulability analysis (also known as *message admission test*) must be done before a transmission is begun. If the message passes the admission test, the channel slots are reserved for the messages. This planning based approach allows admission control and results in reservation-based system. In this section, we propose a dynamic planning based protocol, which is a variant of the LDCR protocol, namely, *guarantee based protocol.* The guarantee based protocol admits a message based on an estimate of worst case message service time, and ensures successful transmission of all the admitted messages.

5.1 Guarantee Based LDCR Protocol

The semantics of the protocol is as follows : *messages are admitted based on an estimate of worst case message service time, and their transmission is guaranteed if their service times do not exceed this estimate.*

The message admission test uses worst case channel access time (WCT) to compute the worst case time before the message can be transmitted. Before giving the admission test, we present the following terms: Queue : Each node maintains a queue of messages in the order of arrival, which are to be transmitted in subsequent epochs.
M_{est} : Worst case service time estimate per message, N : Number of nodes,
Rank : Position of message in the arrival queue, Pos_tree : Position of node in the CR tree,
N_col : Maximum number of collisions in the tree traversal, Time_for_Not : Time for notifier transmission, t : Current time.

For a message (C_j^k, D_j^k) arriving at node j having rank k, the worst case channel access time is computed as follows :

1. First we compute the time slots used for transmission of all messages, which are to be transmitted in epochs, prior to the epoch in which the new message is to be transmitted. The number of such epochs is captured by the rank term, k. This time is $(M_{est} + Time\_for\_Not) * (N * (k-1))$.

2. Next we compute the time elapsed due to transmission of messages, at nodes above the current node in the pre order traversal of the CR tree. This time is : $(M_{est} + Time\_for\_Not) * (Pos\_tree - 1)$.

3. Lastly, we add the overhead due to collision. This time is : N_col * k. Note that the overhead for notifier transmission, is accounted for in the previous two computations.

Combining these times, we get :

$$WCT_j^k = (M_{est} + Time\_for\_Not) * (N * (k-1) + Pos\_tree - 1) + N\_col * k \qquad (1)$$

The admission test **(AT-1)** for message at node j having rank k is given as:

$$WCT_j^k + C_j^k \leq D_j^k - t \quad (2)$$

There is a tradeoff between performance and the strictness of the admission test. A larger value for M_{est}, implies a more pessimistic estimate of WCT, which results in lower performance. Thus, the value of M_{est} should be carefully chosen depending on the application requirements. The proof of correctness of the protocol is given below.

Lemma : If a message passes the admission test (AT-1) and its service time is less than or equal to the worst case service time (M_{est}), then the protocol guarantees the successful transmission of the message.

Proof : The proof is by contradiction. Assume that a message passes the admission test (AT-1), and its service time is less than or equal to M_{est}. Invalidation of this guarantee can occur only if the WCT of the message (which is computed using M_{est}) increases subsequently. The WCT of the message can increase only if the service time of any other admitted message exceeds M_{est}. Clearly, this is a contradiction to the assumption that all the admitted messages have service time less than or equal to M_{est}. Hence proved.

6 Generalisation of LDCR and LDCR-G Protocols

In many real-time applications where guarantees are given to messages, applications must be given sufficient time to recover, if messages generated by them fail the admission test i.e. *the admission decision for a message is taken on or before a certain time (recovery period) from its deadline, and the protocol guarantees that all the admitted messages will be transmitted successfully.* Admission test for giving guarantees should be performed when $L_i \geq RP$ for some message i, where RP is called **recovery period** i.e. a message is guaranteed to be sent, or it is rejected before its laxity becomes less than the recovery period. This will allow the application to take recovery action. From the semantics of LDCR and LDCR-G protocols, we find that they are different instances of a general protocol as described below.

- In the general protocol, when RP = 0, the guarantee needs to be given only when laxity is equal to zero. This means that the message will not be rejected till its $L_i = 0$. This reduces to a best effort protocol.

- In the general protocol, when RP of a message is set to be the initial laxity of the message (laxity on arrival), the protocol should perform the admission test on arrival of the message itself. If the message passes the admission test, it is guaranteed to be sent, else it is rejected. This is the semantics of the LDCR-G protocol.

Thus, we find that on changing the value of RP, different semantics can be attained. LDCR and LDCR-G are two extremes in this spectrum of protocols. The particular value of RP can be chosen based on the application requirements. Owing to space constraints, we do not show the graphs to support the above observations.

7 Conclusions

In this paper, we have identified the need for real-time protocols to support multi packet real-time message transmission which is the form of communication in a typical real-time system. We have highlighted the inadequacy of the existing protocols to support this requirement. As a solution to the problem, we have proposed two classes of protocols, one with best effort service, and the other with dynamic planning based service, catering to different application semantics. In the best effort category, two protocols, namely, MDCR and LDCR are proposed. The MDCR is a straightforward extension of DCR and the LDCR is based on the concept of message deferment. The dynamic planning based protocols perform message admission tests, for guaranteeing timely message transmission. In the dynamic planning based category, guarantee based LDCR protocol is proposed.

We have demonstrated the effectiveness of our best effort protocols through simulation, for a wide range of parameters for multiple performance metrics, by comparing them with a protocol (PBCSMA) recently proposed for the same problem. Our simulation results reveal that the LDCR protocol performs better than the PBCSMA and MDCR. We have also proposed a generalisation of the real-time protocols, based on recovery period. This leads to a generic protocol, of which LDCR (best effort) and LDCR-G are instances. This generalisation allows the applications to flexibly choose the type of guarantees required by them. The performance of the generalised protocol is also studied.

References

1. K.Ramamritham. "Channel characteristics in local-area hard real-time systems," *Computer Networks and ISDN Systems,* vol.13, pp.3-13, 1987.
2. W. Zhao and K. Ramamritham, "Virtual time CSMA protocols for hard real-time communication," *IEEE Trans. on Software Engineering,* vol.13, no.8, pp.938-52, Aug. 1987.
3. James F. Kurose, M. Schwartz, and Y. Yemini, "Multiple-access protocols and time-constrained communication," *ACM Computing Surveys,* vol.16, no.1, pp.43-70, Mar. 1984.
4. Ozgur Ulusoy, "Network access protocol for hard real-time communication systems," *Computer Communications,* vol.18, no.12, Dec. 1995.
5. V. Upadhya, "Design and analysis of real-time LAN protocols," *M.S. Thesis,* Department of Computer Science and Engineering, Indian Institute of Technology, Madras, Jan. 1996.
6. S.K. Oh and G.H. Macewen, "Task behaviour monitoring for adaptive real-time communication," *Real-time Systems,* vol.11, pp.173-175, 1996.
7. J.A. Stankovic and K. Ramamritham, "Hard Real-time Systems," *Computer Society Press of IEEE,* 1730, Massachussets Avenue, Washington D.C., 1993.
8. R. Yavatkar, P. Pai and R. Finkel, "A Reservation-based CSMA protocol for integrated manufacturing networks," *IEEE Trans. on Systems, Man and Cybernetics,* vol.24, no.8, Aug. 1994.
9. G. Le Lann and N. Rivierre, "Real-time communications over broadcast networks : The CSMA-DCR and the DOD-CSMA-CD Protocols," *Technical Report,* Institut National De Recherche En Informatique Et En Automatique, France, Mar. 1993.
10. Intel Corp, "CSMA/DCR deterministic access LANs," *Presentation Viewgraphs,* Folsom Microcomputer Group, Feb. 1988.