Dynamic Real-Time Channel Establishment in Multiple Access Bus Networks

Anita Mittal[+] G. Manimaran[*] C. Siva Ram Murthy[+]

[+] Dept. of Computer Science and Engineering

Indian Institute of Technology, Madras 600 036, INDIA

[*] Dept. of Electrical and Computer Engineering

Iowa State University, Ames, IA 50011, USA

Email: *gmani@iastate.edu* *murthy@iitm.ac.in*

Abstract

Real-time communication with performance guarantees is becoming very important to many applications, like computer integrated manufacturing, multimedia, and many embedded systems. The timing guarantees required in these applications are not possible without a network protocol which supports the timely and predictable delivery of messages. Multiple access bus networks are being increasingly used in real-time applications because they are simple, economical, and have small propagation delays. In such networks, the nodes have to coordinate with one another in order to transmit data and control messages over the shared channel. Though several real-time protocols have been proposed for the multiple access bus networks, there is no guarantee based protocol which addresses the problem of integrated scheduling of dynamically arriving periodic (i.e., time triggered) and aperiodic (i.e., event triggered) messages. In this paper, we propose two guarantee based protocols, EDF and BUS protocols, which address this problem. The proposed protocols dynamically establish real-time channels for dynamically arriving periodic message streams and thus ensure predictable communication. In our protocols, we employ the concept of aperiodic server for servicing aperiodic messages. In the simulation studies, the performance metrics, success ratio (measure of schedulability) and channel utilization are used to study the performance of the two protocols. It is observed that the EDF protocol offers higher schedulability as compared to the BUS protocol for periodic messages, while the BUS protocol offers better channel utilization for aperiodic messages as compared to the EDF protocol.

Keywords: Multiple Access Network, MAC Protocol, Real-Time Channel, Periodic and Aperiodic Messages, Message Scheduling.

1 Introduction

The use of distributed computer systems in real-time applications such as embedded command and control systems, image processing and transmission, automated manufacturing, and multimedia is becoming increasingly common. This is not only due to the nature of the applications, but also due to the potential that distributed systems have for providing reliability, resource sharing, and extensibility [1, 2, 3]. Distributed real-time applications are usually realized by executing a number of communicating tasks, which have deadlines imposed by the corresponding mission/function, on multiple nodes. Predictable inter-task communication is of prime importance in real-time systems, because unpredictable delays in the delivery of the messages can affect the completion time of the tasks participating in message communication. Also in multimedia applications, timely delivery of messages is required to guarantee the bounds on delay and delay jitter in delivering video/audio frames consistent with human perception. The timing guarantees required in these applications are not possible without a network protocol which supports timely and predictable delivery of messages.

In order to support real-time communication with performance guarantees, real-time channels[1] are established with the specified traffic characteristics and Quality of Service (QoS) requirements. There are two phases in handling real-time channels: (i) Channel establishment (ii) Run-time scheduling. During the channel establishment phase call admission control is performed to check whether enough resources are available to support the new channel without jeopardizing the guarantees of the existing real-time channels. Run-time scheduling involves scheduling of messages, at run time, adhering to the guarantees of the admitted real-time channels.

Distributed real-time applications are implemented using either real-time multi-hop networks or real-time multiple access networks. The main difference between the two networks is that in multi-hop networks each node has to decide on which of the multiple links a message has to be transmitted while in multiple access networks, multiple nodes contend for a single shared communication channel. While schemes for channel establishment and run-time scheduling algorithms have been proposed for real-time multi-hop networks [4, 5], not much work has been done for real-time multiple access networks.

Multiple access bus networks are being increasingly used, for real-time applications, because they are simple, economical, and their propagation delays are small. In this type of network, the nodes transmit messages via a shared channel. Only one message can be transmitted over the channel at any time. A *collision* occurs if at any time two or more messages are simultaneously transmitted on the channel. In such networks, scheduling is the responsibility of Medium Access Control (MAC) protocol, which arbitrates access to the shared medium and determines which message is to be transmitted at

[1] A real-time channel is a virtual circuit with specified QoS guarantees.

any given time. Collision resolution can be done either in a deterministic manner or in a probabilistic manner. Probabilistic collision resolution protocols are not suitable for hard real-time communication.

Several schemes [6, 7, 8, 9, 10, 11, 12, 13, 14] have been proposed for real-time communication in multiple access networks and are classified into *best effort protocols* or *guarantee-based protocols*. Best effort protocols (e.g., Virtual Time CSMA [7], PB-CSMA [9], and many protocols discussed in [6, 8, 13]) aim to provide real-time guarantee to messages in a best effort manner by employing probabilistic collision resolution. These protocols are suitable for soft real-time communication such as video transmission, but are not suitable for hard real-time communication. Guarantee-based protocols [10, 11, 12, 14] employ admission control and ensure that all the admitted messages meet their deadlines, and hence are suitable for hard real-time communication. Among the guarantee-based protocols, some support periodic messages only (e.g., TDMA [6]; [10]), some support aperiodic messages only (e.g., [11, 14]), and others support both periodic and aperiodic messages (e.g., [12]). Section 2.3 has a detailed discussion on the guarantee-based protocols.

In this paper, we propose two guarantee based protocols for dynamic real-time channel establishment (to support periodic messages) with support for scheduling of aperiodic messages in multiple access bus networks.

The rest of the paper is organized as follows. In section 2, we first present the requirements of a guarantee based protocol and then discuss the prior work in the light of these requirements. In section 3, we present the channel model and the proposed protocols for multiple access bus networks. In section 4, we present the simulation results and finally, in Section 5 we summarize our work.

2 Background

2.1 Message Model

In real-time applications, messages are of two types:

- Periodic messages: A periodic message M_i is characterized by its *service time* (c_i) and *period* (p_i). c_i is the time required for the transmission of one instance of message M_i. Periodic messages require hard real-time guarantees, which are provided by establishing real-time channels for the specified traffic characteristics.

- Aperiodic messages: An aperiodic message M_i is characterized by its *arrival time* (a_i), *service time* (c_i), *and deadline* (d_i). For aperiodic messages soft guarantees suffice but they require fast response time.

2.2 Desirable Properties of Real-time MAC Protocol

A real-time MAC protocol should:

1. support real-time channel establishment,
2. provide predictability,
3. support scheduling of aperiodic messages along with periodic messages, and
4. be distributed in nature.

1. Real-time channel establishment: To provide guaranteed real-time communication, real-time channels need to be established for the periodic messages. The protocol should have an associated admission test which is to be performed while admitting a periodic message stream (schedulability checking). A real-time channel is established by reserving the resources as per the specified traffic characteristics of the periodic message, if the admission of this connection does not jeopardize the guarantees given to the already admitted channels. Since the channel is shared by a number of nodes in the multiple access networks, the nodes need to cooperate and coordinate with one another (i) for transmitting channel establishment requests, (ii) for admission test and resource reservation, and (iii) for transmission of messages on the real-time channels so that all the messages meet their deadlines. Each node needs to have global state information for performing these functions in a distributed manner.

2. Predictability: In many real-time applications such as air-traffic control and multimedia, periodic message requests for channel establishment arrive dynamically. Consequently, one cannot predict whether the messages will meet their deadlines. Therefore, in the context of scheduling dynamically arriving message streams, predictability means that once a periodic message stream is admitted (i.e., a real-time channel is established), all the message deadlines of this real-time channel will be met. The protocol should provide bounded response time to a channel establishment request (i.e, time taken to decide acceptance/rejection of a request based on admission test should be bounded) so that some alternate recovery action can be initiated in case the request is rejected. This is an important and desirable property for hard real-time communication. For example, if a "desirable communication" requirement (which needs more bandwidth) cannot be satisfied by the network, then the request is rejected; and if the rejection decision is made quickly in a bounded time, then the application has the opportunity to initiate a less demanding (i.e., "acceptable") communication requirement as a means to introduce graceful degradation in system performance.

3. Integrated scheduling of periodic and aperiodic messages: Besides these periodic messages which require hard real-time guarantees, the protocol should also support scheduling of aperiodic messages which may have soft deadlines (a deadline is soft if when missed does not compromise the

security of the system) or no deadlines at all (for example, conventional messages required to keep the system running).

4. Distributed in nature: Centralized scheduling systems are prone to single point failure where the failure of the central scheduler paralyzes the entire system. Also, the centralized scheduler incurs substantial overhead for giving channel access rights to the nodes. (The centralized scheduler needs to send some *token message* which specifies the duration for which the selected node can transmit on the channel).

2.3 Related Work – Guarantee Based Protocols

Under guarantee-based communication, the following protocols have been proposed for real-time communication in multiple bus access networks. For each protocol, its strengths and weaknesses are discussed.

- Guarantee-based protocols for scheduling of aperiodic messages in LAN and Switched LAN environments have been proposed in [11, 14]. These protocols do not support scheduling of periodic messages.

- A guarantee based protocol for scheduling of periodic messages in multiple access bus networks has been proposed in [10]. It is a centralized protocol and is suitable for networks which have a control unit for controlling access to the shared link (e.g. Field Bus). In this protocol, the control unit uses tokens to grant channel access rights to the admitted real-time channels. This protocol does not address the issue of predictability. Moreover, it does not support integrated scheduling of periodic and aperiodic messages. Also it suffers from the disadvantages of a centralized protocol, namely, token overhead for access control of the link by the nodes and vulnerability of the centralized control unit.

- A centralized scheduling algorithm for scheduling of periodic and aperiodic messages has also been proposed for the backplane bus of a workstation [17]. In the workstation, instant global knowledge of all the messages contending for the backplane bus is available through separate hard-wired control channels. The bus is not used for exchanging control information. Hence this scheduling algorithm cannot be used as such for multiple access networks, in which global state information (control messages) needs to be collected using the same channel that is used for message transmission.

- The protocol proposed in [12] extends TDMA with the concept of shared channels to support both periodic and aperiodic messages. Though this work [12] does not explicitly address the issue of providing bounded response time for channel establishment, it compliments our work.

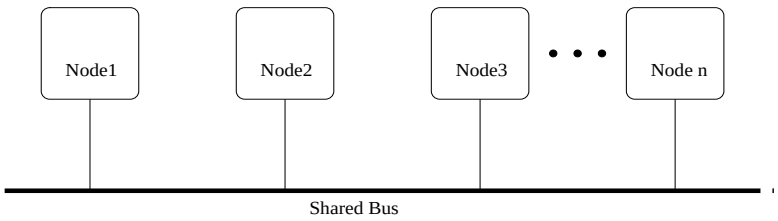

Figure 1: A multiple access bus network

From the above discussion, it can be noted that there is no protocol that addresses all the four requirements of a guarantee based real-time MAC protocol. This motivated us to propose guarantee based protocols for integrated scheduling of periodic and aperiodic messages in multiple access bus networks. In the next section we present the channel model and the proposed protocols.

3 Proposed Protocols

3.1 Channel Model

- A population of N nodes share the channel, as shown in figure 1.
- There is a finite channel propagation delay.
- Channel access is slotted. Transmission can start only at the beginning of a slot.
- The slot time is equal to the maximum end-to-end round trip delay for a bit. In Carrier Sense Multiple Access (CSMA) protocols, this is the time required for a node to be certain that it has the sole access to the channel.
- Carrier sensing is instantaneous. Each node is capable of detecting an idle slot, a successful transmission, and a collision as in CSMA based protocols.
- All the nodes monitor the activity on the shared channel.

3.2 Proposed Protocols

We have proposed two protocols EDF protocol and BUS protocol. These protocols are distributed in nature with each node executing the same algorithm. The protocol alternates between two phases:

- Reservation Phase: During this phase, resources are reserved (i.e., real-time channels are established) for channel establishment requests (periodic messages) that pass the admission test.
- Transmission Phase: During this phase, the nodes transmit messages on the real-time channels so as to guarantee the deadlines of all the admitted messages.

The proposed protocols have the following components:

- Request Server (RS): This is a periodic server used for collecting channel establishment requests and aperiodic message requests and for giving bounded response time to channel establishment requests. RS is activated during the reservation phase of the protocols. RS is responsible for making known the channel establishment requests globally to all the nodes in an ordered fashion so that the admission test can be performed and resources reserved for channels that pass the admission test. The *server* concept has been borrowed from the task scheduling algorithms that use an aperiodic server [15] for executing aperiodic tasks.
- Aperiodic Server (AS): This is also a periodic server used for servicing aperiodic messages.
- Admission Tests: There is an associated admission test with each protocol for call admission control of real-time channels.
- Run-time Scheduling: The run time scheduling algorithms, for the EDF and BUS protocols, are based on the Earliest Deadline First (EDF) algorithm [16] and backplane bus scheduling algorithm [17], respectively. These algorithms are used during the transmission phase of the protocols.

In the proposed protocols, real-time channel establishment and the predictability requirement of real-time MAC protocols are taken care of by RS. Integrated scheduling of periodic and aperiodic messages is supported by the AS. Also the protocols are distributed in nature, thus satisfying all the properties of a real-time MAC protocol.

3.3 Request Server (RS)

The RS is a periodic server responsible for giving (i) bounded response time to real-time channel establishment requests, and (ii) for collecting global state information without incurring much overhead. During the activation of RS, message notifiers are transmitted. The notifier contains information such as, type of request - periodic/aperiodic, computation time, period/deadline, and node number of the message source. One slot is required for transmission of a message notifier. Since all the nodes monitor the channel, and the channel is a broadcast medium, all the nodes get to know about the message traffic at the other nodes through these message notifiers. For collecting channel establishment requests, each

node should get to broadcast at least one message notifier (if any) per server period (i.e., service time should be at least N slots where N is the number of nodes in the network). The period of the server determines the response time to channel establishment requests. The channel bandwidth alloted to RS (which is proportional to $service\_time/period$), represents the overhead incurred for obtaining global state information. Since the message arrivals at the nodes is random and bursty, static reservation of one slot for each node per server period would result in large overhead for collecting message transmission requests. Hence we have used a limited contention protocol, a CSMA/CD based protocol with Deterministic Collision Resolution (DCR), the preorder DCR protocol [18] for transmitting message notifiers. Preorder DCR protocol is an efficient protocol wherein when the load is light, say, only one message is ready in the entire network, then only one slot of RS is used and the remaining slots can be used for transmission of messages. When the load is heavy, collision is resolved deterministically to bound the worst case channel access delay, giving every node a chance to broadcast at least one message notifier in one RS period.

3.3.1 Preorder DCR Protocol

In this protocol, the nodes of the multiple access network are statically mapped to the vertices of a binary tree. Initially, the root node and all its child nodes in the tree which have a message are eligible to transmit. The nodes execute the preorder tree traversal based DCR protocol. The recursive preorder DCR protocol is given below:

PreorderDCR(root)

1. In the current slot, root node and all its children are eligible to transmit and nodes with messages contend for the channel.
2. If there is no collision, transmission is successful, return.
3. If there is collision,
   (a) the next slot is reserved for transmission of message notifier by the root node.
   (b) PreorderDCR(leftsubtree)
   (c) PreorderDCR(rightsubtree)

Service time of RS: The number of slots required so that all nodes get to transmit/broadcast a message notifier (in the worst case when all the nodes have a message request) is $(3 * N/2)$ slots where N is the number of nodes in the network. Given the service time requirement of the RS, its period can be determined in two ways.

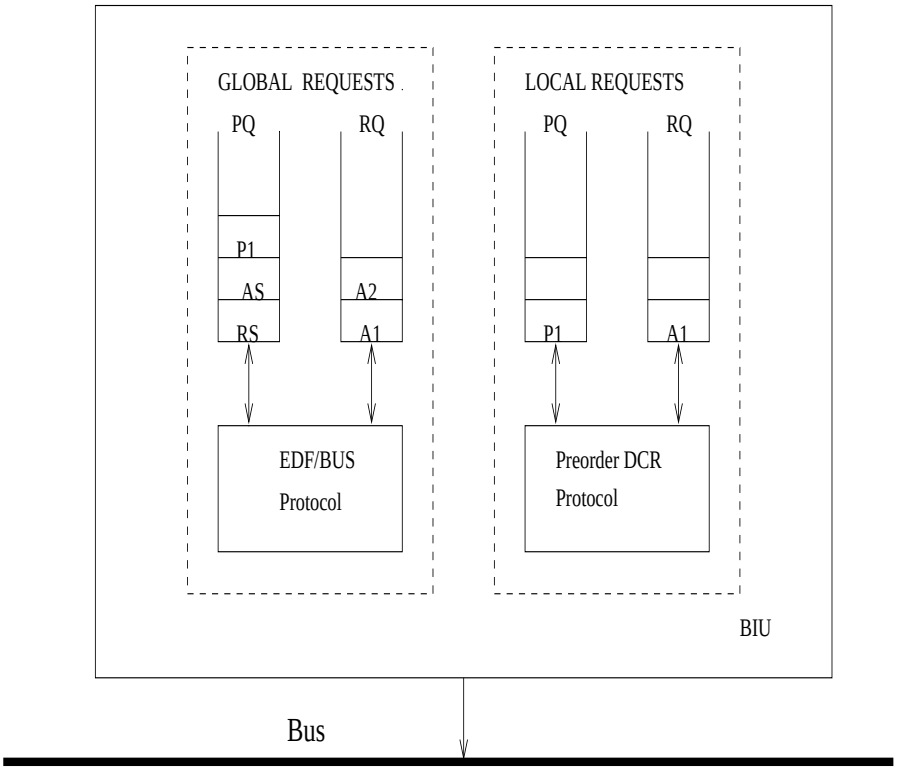

Figure 2: A node in the multiple access network

- Fix the percentage of overhead due to the RS. In this case the period is given by the expression $\frac{\text{service time of server}}{(\text{percent overhead} * \text{channel bandwidth})}$
- Fix the worst case channel access delay for a request notification. This time gives the period of the RS and using the above mentioned expression, percentage of overhead due to RS can be computed.

3.4 Aperiodic Server (AS)

The AS services the notified aperiodic requests. The service could be either guaranteed service for which the guarantee based DCR protocol [14] can be used or could be a best effort service. In the best effort service category, the service policy can be either fairness oriented, where the AS bandwidth is fairly given to all the contending nodes, or aimed at maximizing throughput of aperiodic messages. The percentage of bandwidth reserved for aperiodic traffic is application dependent. The service time and period of the server can be fixed based on the requirements of the application.

3.5 Protocols Description

The aperiodic and periodic messages are generated in each node by the application running on the nodes of the network. Each node maintains two queues (local) for storing the local periodic (PQ) and aperiodic message (RQ) notifiers, as shown in Figure 2. A message notifier has to be broadcast before the transmission of the actual message. Each node also maintains two global queues, aperiodic queue (RQ) for storing the aperiodic message notifiers which have been broadcast successfully on the channel and periodic queue (PQ) for storing the notifiers of admitted real-time channels (periodic messages). The RS and AS notifiers, which are permanent real-time channels, are put in the global queue of each node. The periodic message requests are maintained in arrival time ordered queue. Transmission of one instance of a periodic message should be completed before the arrival of the next instance, since deadline of message equals the period of the message stream. In the EDF/BUS protocol, channel access rights are given to the real-time channels for say, n slots, based on the EDF/BUS run-time scheduling algorithms (presented later in this section). Each node executes the following protocol, once a real-time channel, *RTchannel*, has been chosen by the EDF/BUS scheduling algorithm.

Protocol(*RTchannel***,***n***)** /* *RTchannel* specifies the channel which has access rights to the multiple access bus for n slots */

1. If *RTchannel* = RS, then
   (a) Transmit message notifiers from the local queues with priority to periodic message notifiers over aperiodic message notifiers as per the preorder DCR protocol, until a slot goes idle or RS service time is exhausted.
   (b) If successful(notifier transmission) and notifier is an aperiodic message notifier then insert it in the aperiodic global queue.
   (c) else if successful(notifier transmission) and notifier is a periodic message notifier then
      i. *result* = Admit(notifier) /* Perform admission test using the appropriate EDF/BUS protocol admission test */
      ii. If *result* = *success* then
         A. Insert notifier in global periodic queue.
         B. Reserve resources for this real-time channel and update system utilization information.
      iii. else reject the request.
2. else if *RTchannel* = AS, then

(a) Grant channel access rights to messages in global aperiodic queue based on predetermined aperiodic message service policy/protocol.

3. else give channel access rights to the node which is the source of the real-time channel.

When a node wishes to tear down the real-time channel after completing transmission of all instances of this periodic message stream, the node transmits channel termination request as the last message on this channel. All the nodes then update their global periodic queue and channel utilization information accordingly.

When none of the periodic messages are using the channel (no message is ready in the global periodic queue), then in the idle channel time, aperiodic messages (if any) are serviced. In the event of aperiodic message queue also being empty, the RS is activated which polls for new messages.

We see that, when the system load is light, these protocols behave like a contention based protocol, since RS which follows a CSMA/CD based DCR protocol is active in idle time. When the system load is heavy, the channel is granted to the admitted requests without any collisions based on the EDF/BUS policy. Thus the protocols adapt well to the system load.

When the load is heavy, admission test for a channel establishment request can be performed immediately based on the channel utilization information at the node and the test result communicated to the application. The request can be rejected if already other channel establishment requests are queued in the periodic queue of the node. Similarly if the system load is light, RS would be activated when the channel is idle and hence the request can be broadcast, and channel established if it passes the admission test. Hence on the average the response time to channel establishment requests is much less than the period of the RS.

3.6 EDF Protocol

The nodes follow the EDF algorithm of serving the earliest deadline first message in the global periodic queue. EDF [16] is a dynamic priority scheduling algorithm that always schedules the message with the earliest deadline, preempting any currently scheduled message if required. Preemption of messages takes place at packet boundaries. Since in multiple access networks, there is no counterpart of the interrupt mechanism of processor system, (to signal preemption of the executing task when a new task with earlier deadline arrives) this function is emulated. The earliest deadline message among the periodic messages ready for transmission is the one which gets access rights to the channel. If the message requires the channel access rights for, say, n slots then all message notifiers, in the global periodic message queue, whose ready times are less than the current time plus $n\tau$ are examined, where τ is one slot time. If any of the messages has a deadline earlier than the current message then the

current message is split into two parts: the first part of the message is transmitted till the arrival time of a new message (which needs to preempt the current message) and the second part of the message with the remaining service time is inserted back into the queue as a new message.

Admission Test

A periodic message is admitted if it satisfies the following condition:

$$util_{curr} + util_{new} \leq 1$$

where $util_{curr}$ is the current utilization of the channel and $util_{new}$ is the channel utilization requirement of the new message, given by the expression:

$$util_{new} = \frac{\text{maximum message size/slot size}}{\text{period of the message/slot time}(\tau)}$$

Algorithm EDF /* Messages are maintained in ascending order of ready time */

1. Find the earliest deadline message, m, among the messages whose ready times are less than time t, in the global periodic queue.
2. If no message is ready for transmission then,
   (a) If global aperiodic queue is not empty then,
      i. Create AS instance with $AS.service\_time = \lceil (first\_message.ready\_time - t)/\tau \rceil$ and $AS.ready\_time = t$.
      ii. m = AS
   (b) else
      i. Create RS instance with $RS.service\_time = \lceil (first\_message.ready\_time - t)/\tau \rceil$ and $RS.ready\_time = t$,
      ii. $m = RS$.
3. $n = m.service\_time$
4. Find the earliest deadline message, $m1$, among the messages whose ready times are less than $(t + n\tau)$.
5. If $m1.deadline < m.deadline$ then,
   (a) Split n into two parts such that first part $n1$ = minimum(n, $\lceil (m1.ready\_time - t)/\tau \rceil$) and second part $n2$ is $(n - n1)$.
   (b) Give channel access to the real-time channel for $n1$ slots

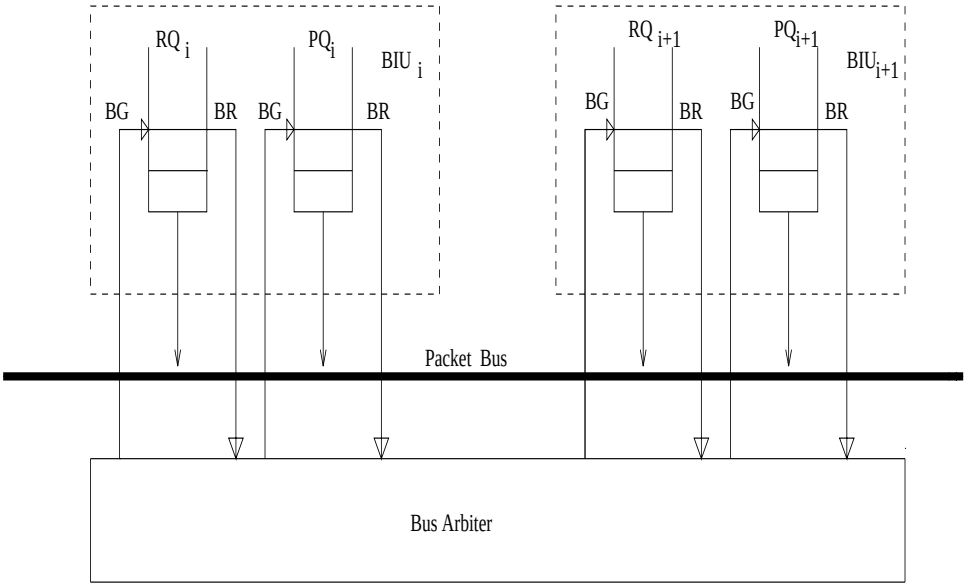

Figure 3: Bus arbitration - backplane bus of workstation

(c) Insert the message with $m.service\_time = n2$ in the global periodic queue in the appropriate position based on ready time order.

6. else give channel access to the real-time channel for n slots.

3.7 BUS Protocol

In this section, we present our BUS protocol which is based on backplane bus scheduling algorithm [17] and uses RS for request collection and AS for servicing aperiodic messages.

3.7.1 Overview of Backplane Bus Protocol [17]

A multimedia workstation consists of a number of hardware units that are connected together by a packet bus as shown in Figure 3. Every unit has a Bus Interface Unit (BIU) that connects it to the packet bus. The output buffer of BIU consists of two parts: periodic queue (PQ) and aperiodic queue (RQ). Each queue has its own request (BR) and grant (BG) signals that are used by the BIU for arbitration. A Bus Arbiter (BA) resolves the contention among the units. The BA executes a service algorithm periodically with a period of N time slots (time slot is one packet transmission time) called the bus service cycle time which is much larger than the period of any periodic message.

Admission Test

The application passes the periodic message parameters, message size M_i and period T_i , to the *bus manager*, an operating system daemon. The bus manager computes c_i, the number of slots to be

reserved for this message in each service cycle.
The stream is admitted if the following condition holds:

$$c_i + Q \leq N - \alpha$$

where c_i is the smallest integer that satisfies

$$M_i \leq c_i(\lceil T_i/(N * \tau) \rceil - 2)$$

Q is the number of time slots in a cycle reserved for periodic messages, and α denotes the number of slots dedicated for random (aperiodic) messages. α is a system parameter that should be changed only when a change in the fraction of allowable periodic traffic is desired.

If the stream is admitted, Q is incremented by c_i. On termination of the stream, signaled by application to the bus manager, Q is decremented. After any change in Q, its value is written in register Q in the BA. PQ's and RQ's contend for the bus when they have messages (packets). The BA is responsible for resolving the contention. The two counters n and q in the BA are updated every service cycle from the N and Q registers, respectively. The value of counter n denotes the remaining number of slots in a service cycle, and q denotes the number of slots allotted to streams that have not yet been used by periodic messages, during this service cycle. The service algorithm executed by BA is as follows:

1. If $n > q$, grant the bus to a RQ if any of them is requesting; otherwise grant the bus to a PQ if any of them is requesting.
2. If $n \leq q$, grant the bus to a PQ if any of them is requesting; otherwise grant the bus to a RQ if any of them is requesting.
3. Decrement q if grant is given to PQ.
4. Decrement n. If $n = 0$, reload counter n from register N and counter q from register Q.

3.7.2 Multiple Access Bus Protocol

In a multiple access bus network, the periodic traffic arrives at each node independently unlike in backplane bus where it is centralized. Moreover, there is no central bus arbiter which performs access arbitration based on channel reservation of the admitted requests. Also there are no control lines to convey bus request and bus grant information. In the BUS protocol for multiple access network, all the nodes follow an agreed upon protocol for servicing the admitted periodic and aperiodic messages. The nodes follow a periodic service policy similar to the one executed by the BA with N slots and

cycle time (period) $N\tau$. The service times and periods of the RS and AS are converted to the number of slots required per service cycle to R_s and A_s, respectively. The algorithm executed by each node is: **Algorithm BUS(R_s,A_s,N,Q)** /* R_s, A_s, and Q are the number of slots alloted for RS, AS, and periodic messages in a bus service cycle of N slots. Initially n and q counters are set equal to N and Q, respectively */

1. If $n = N$, service RS for R_s slots or until a slot goes idle whichever is earlier.
2. If $n > q$ and aperiodic queue is not empty, service AS for $(n - q)$ slots or until aperiodic queue becomes empty whichever is earlier.
3. If $q > 0$ and $n > q$, service periodic requests and decrement q accordingly.
4. If $n > 0$ and $q = 0$, service AS if aperiodic queue is not empty otherwise service RS till a request arrives.
5. if $n = 0$, set n and q to N and Q, respectively.

3.8 Protocols Comparison

Admission Test Complexity: Both EDF and BUS protocols have simple admission test that can be efficiently performed for each newly arriving request as the time taken to perform the admission test is much smaller than the mean inter-arrival time between requests. Among the protocols, EDF has simpler admission test than the BUS protocol.

Schedulability: The EDF protocol offers higher schedulability than the BUS protocol. This is because the admission test of BUS protocol slightly over allocates bandwidth to periodic messages and as a result of this over allocation lesser number of real-time channels can be established. However, at the run time this over allocated channel bandwidth is made available for servicing aperiodic messages. In the EDF protocol, due to the absence of this over allocation, the number of periodic messages admitted is higher than that of the BUS protocol.

Response Time: The BUS scheduling scheme gives fast response time to aperiodic messages without jeopardizing the guarantees given to admitted periodic messages. Aperiodic messages (if any) are serviced when no periodic messages are using the channel, by activating the AS for idle channel duration. The time durations when the channel is idle are uniformly made available with period $N\tau$ for servicing aperiodic messages in the BUS protocol. The service period $N\tau$ is less than the deadline of aperiodic messages, and hence the aperiodic messages can effectively utilize this time. As a result of this, channel utilization by aperiodic messages increases. In the EDF protocol, besides lack of over allocated bandwidth being made available for aperiodic messages, occurrence of idle channel instances when

there are no periodic messages to be transmitted is not uniform (the idle channel bandwidth is made available in bursts) and hence aperiodic messages cannot utilize the idle channel bandwidth efficiently. A number of aperiodic messages might miss their deadline in between successive channel availability for aperiodic message transmission.

Example: Consider a multiple access bus network with slot size equal to 64 bytes and slot time equal to 5 micro seconds. The bus service cycle is 100 slots i.e., 500 micro seconds. Let 10% of the bandwidth (10 slots) be reserved for AS and 1% (1 slot) for the RS. Let all periodic message requests have a maximum service time (assumed to be same for all requests) of 73 slots (365 micro seconds) with period equal to 4700 micro seconds.

BUS Protocol

The number of slots required per cycle in the BUS protocol for a periodic request is 10 ($73 \leq 10 * (\lceil 4700/500 \rceil - 2)$). Since the number of slots available for periodic requests is 89 $(100 - 10 - 1)$ a maximum of 8 real-time channels can be established simultaneously. If the number of instances of the periodic message stream is 20, then in $(4700 * 20)$ time duration which equals 188 cycles, 1880 slots will be reserved for the periodic message stream of which only 1460 $(20 * 73)$ will be used for periodic message transmission. The remaining 420 $(1880 - 1460)$ slots are made available for aperiodic message transmission.

EDF Protocol

The channel utilization of each periodic request is $73 * 5/4700 = 0.0777$ and hence 11 real-time channels can be simultaneously established with total periodic utilization equal to 0.8547 (less than the maximum of 0.89 permissible). From this example we can see that the number of periodic requests admissible by EDF is larger than that of the BUS protocol. In the BUS protocol, the extra 420 slots will be made available uniformly for servicing aperiodic messages and hence results in higher channel utilization by aperiodic messages. In EDF protocol, though the idle time cannot be given periodically (uniformly), the AS time can be made available uniformly by changing the granularity of the AS, i.e., the period and service time would be divided by the same constant factor. For example, service time of 4 units and period 10 units can be transformed into a server with service time 2 units and period 5 units.

The differences mentioned above between the two protocols can be clearly observed in the simulation results. The effect of granularity of the request and AS, and the effect of system load and message characteristics are also studied in the simulation.

4 Simulation Study

To evaluate the performance of the protocols, we have developed a simulation model.

4.1 Simulation Model

The simulation model is parameterized by the number of nodes, percentage of bandwidth reserved for request and AS, and distributions of message arrivals, transmission times, and laxities (Laxity of a message = deadline - service time - current time). In the simulation studies reported in this paper,

- The periodic and aperiodic messages arrive at each node as a Poisson process. The arrival rate at each node is the same.

- The system load in terms of fraction link bandwidth consists of periodic load (Pload) and aperiodic load (Aload) and are defined as:

  $$Pload = \frac{\text{periodic msg. arrival rate per node*N*avg. periodic msg. length*avg. duration of periodic msg. stream}}{\text{channel bandwidth available for periodic messages}}$$

  (Channel bandwidth available for periodic messages is equal to the total link bandwidth minus the bandwidth reserved for RS and AS.)

  $$Aload = \frac{\text{aperiodic msg. arrival rate per node*N*avg. aperiodic msg. length}}{\text{channel bandwidth reserved for aperiodic messages}}$$

  In our simulations the arrival rate is computed given the other (simulation) parameters.

- The deadline of aperiodic messages is uniformly distributed in the range $(AperMesgSize * AperLaxity/2)$ and $(AperMesgSize * AperLaxity)$.

- The message size and period are uniformly distributed in the range $(4500-6400)$ and $(4000-6000)$, respectively.

- The bandwidth reserved for AS/RS can be generalized as $(C_s/m_{as/rs})/(P_s/m_{as/rs})$, where C_s and P_s denote the service time and period of the AS/RS, respectively and $m_{as/rs}$ parameter (m is an integer, $m \geq 1$) decides the granularity. As $m_{as/rs}$ increases, granularity decreases which means decrease in period and service time of the AS. When $m = 1$, granularity is maximum.

The simulation parameters are given in Table 1. For each simulation run 10,000 periodic messages were generated (aperiodic messages were 10 times the number of periodic messages). The slot time is assumed to be 5 micro seconds and slot size 64 bytes. Since the number of nodes is equal to 16, 24 slots are required for RS and hence the period is 12 ms (using the equations of section 3.3.1). AS gets 240 slots every 12ms. In the BUS protocol, in every bus service cycle of 100 slots, one slot is reserved for RS and 10 slots are reserved for AS.

parameter	explanation	value/range
N	number of nodes	16
RS_BW	bandwidth reserved for RS	1%
AS_BW	bandwidth reserved for AS	10%
PerMesgSize	periodic message size (bytes)	4500-6400
Period	period of messages	4000-6000
AperMesgSize	aperiodic message size (bytes)	1000-5000
AperLaxity	laxity of aperiodic messages	20-100
Instances	number of instances of periodic messages	30-50
PLoad	periodic load	0.1-3
ALoad	aperiodic load	0.5-3

Table 1: Simulation parameters

4.2 Performance Metrics

1. Effective Channel Utilization, ECU, defined as
   $ECU = \frac{\text{total time units channel was used for transmitting messages}}{\text{total time units simulated}}$
   This metric when applied to periodic, aperiodic messages and RS individually gives the metrics PCU, ACU, and RCU respectively. RCU captures the overhead for collecting requests and this should be small.

2. Success Ratio, SR, defined as
   $\text{SR} = \frac{\text{number of messages serviced}}{\text{total number of messages generated}}$
   This metric when applied to periodic and aperiodic messages individually gives the metrics PSR and ASR, respectively.

4.3 Simulation Results

Figures 4-14, represent the performance metrics evaluated by varying: the number of nodes, system load (periodic and aperiodic message load), message characteristics (stochastic nature of periodic message size and laxity of aperiodic messages), and the granularity of the aperiodic and request servers.

1. **Effect of number of nodes**: Figure 4 shows the effect of the number of nodes (N) in the network on PCU and ACU. It can be observed from the figure that, the channel utilization is quite insensitive to the number of nodes in the system. It is to be noted that the system load remains the same, it only gets distributed among the existing nodes. The channel utilization by periodic messages for the BUS protocol is slightly less than that of the EDF protocol because

in BUS protocol the admission test slightly over allocates and therefore the number of requests admitted and hence the channel utilization is lower than that of EDF protocol.

2. **Variation of system load**: Figure 5 shows the channel utilization by periodic and aperiodic messages and RS for varying system load for both the BUS and EDF protocols.

   Effect on RCU: As the load increases, RCU decreases. This is because the RS is active when the channel is idle and hence when load is low RCU is high. When system becomes overloaded, idle time decreases and hence RCU decreases, and approaches the reserved bandwidth utilization bound.

   Effect on PCU: The PCU for the EDF protocol is slightly higher than that of BUS protocol. Reason is the same as mentioned earlier. It may be noted that at low loads PCU is same for both the protocols, since channel bandwidth exceeds the total bandwidth demand of all the message transmission requests.

   Effect on ACU: ACU increases with increasing load up to a point and then starts decreasing. Aperiodic messages use the bandwidth reserved for them and also use the channel when it is idle. When the periodic load increases, PCU increases and hence idle time decreases. Therefore aperiodic message utilization decreases. ACU for the BUS protocol is higher than that of the EDF protocol. Two factors contribute to this: first is that the idle time is uniformly made available to aperiodic messages in BUS protocol. In the EDF protocol, the idle time is allotted in bursts, and hence aperiodic messages cannot utilize this time efficiently. The second factor is that in the BUS protocol, the bandwidth reserved in excess, during admission of periodic messages, is eventually made available for servicing aperiodic traffic, at run time.

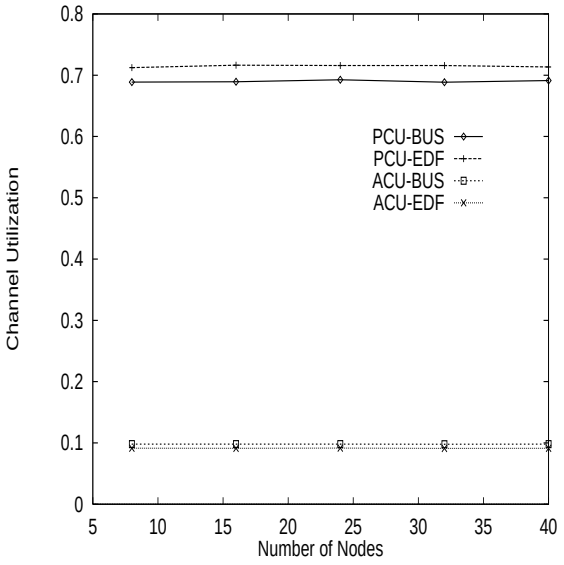

Fig 4. Effect of number of nodes

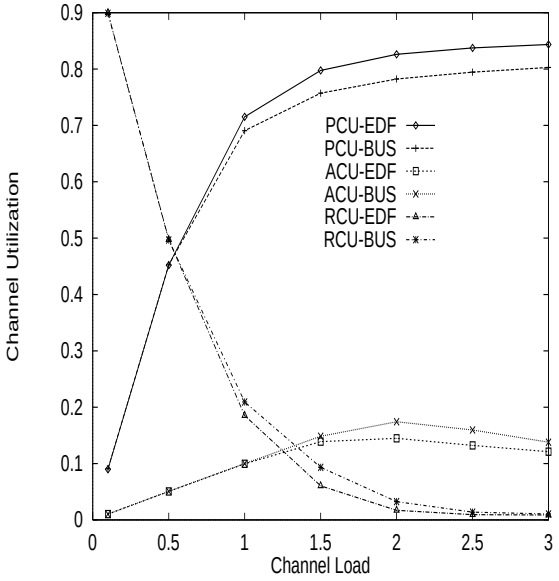

Fig 5. Effect of system load

3. **Variation of periodic load**: Figure 6 shows the channel utilization by aperiodic messages for varying periodic message load. ACU has been plotted for fixed aperiodic loads of 1 and 2. As the periodic load increases, ASR remains same as long as the aperiodic load does not exceed the reserved AS capacity. When aperiodic load is twice the AS capacity, then ACU decreases with increasing periodic load. When periodic load is negligible (0.1) all the aperiodic messages get serviced. This is because the entire channel bandwidth is available for RS and AS. As periodic load increases, the idle time decreases and hence ACU decreases.

4. **Variation of aperiodic load**: Figure 7 shows the PCU for varying aperiodic message load plotted for fixed periodic loads of 0.5, 1 and 2. It can be seen that PCU is independent of aperiodic load. This is expected since the presence or absence of aperiodic requests does not affect the bandwidth reserved for periodic messages and hence insensitivity of the metric PCU to aperiodic load. When the periodic load is 0.5 both the EDF and BUS protocols have the same PCU, but as the load increases, the EDF protocol performs better and the difference in PCU between these two protocols increases. As the load increases, PCU for the BUS protocol decreases due to over allocation of bandwidth for real-time channels by the admission test. It can be seen from the figure that at periodic loads of 0.5, 1, and 2 the difference in PCU are 0, 0.2, and 0.4 respectively.

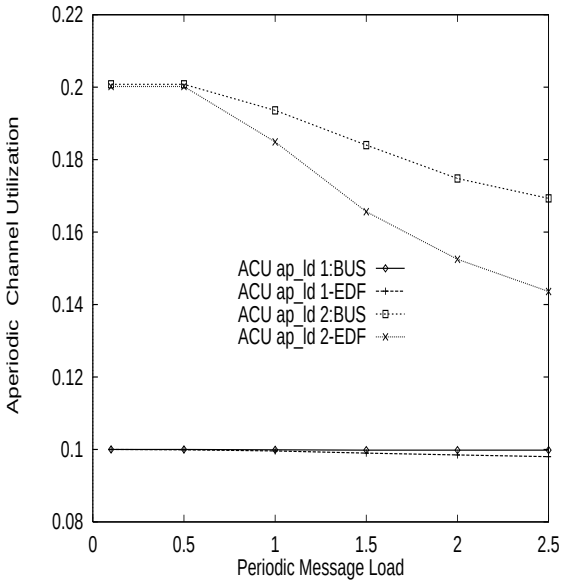

Fig 6. Effect of aperiodic load

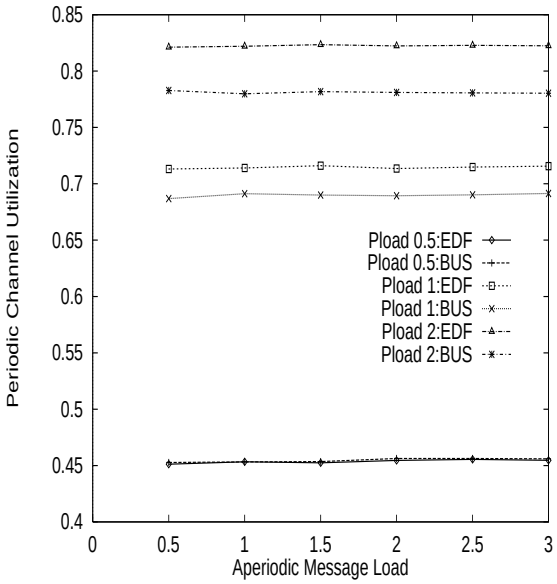

Fig 7. Effect of aperiodic load

5. **Effect of laxity**: Figure 8 shows the effect on ACU by variation in the laxity of aperiodic messages. ACU increases with increasing laxity. As the laxity increases, more number of messages can utilize the channel when it is idle. ACU for the BUS and EDF protocols is same when the load is one and laxity is high. At low laxities, BUS protocol performs much better due to uniform bandwidth allocation and fast response to aperiodic messages. When the laxity is very small, a small increase in laxity improves the ACU.

6. **Effect of stochastic nature of periodic message size**: Figure 9 shows the effect on ACU by variation in the range (difference between the maximum and minimum size of messages of a periodic message stream) of periodic request message size. It is to be noted that the admission test is based on the maximum message size of the periodic message request, and hence some of the reserved bandwidth is not used when instances of that periodic message stream have shorter message sizes. This bandwidth is made available for servicing aperiodic messages and hence the aperiodic channel utilization increases. From the figure it can be seen that, when the range of periodic message size is very large (4500-10000 bytes), aperiodic messages three times the bandwidth reserved for them get serviced (i.e., ACU is 29% though only 10% channel bandwidth is reserved for aperiodic messages). The implication of this result is that the bandwidth reservation for aperiodic messages needs to be less, if the periodic message sizes are stochastic in nature.

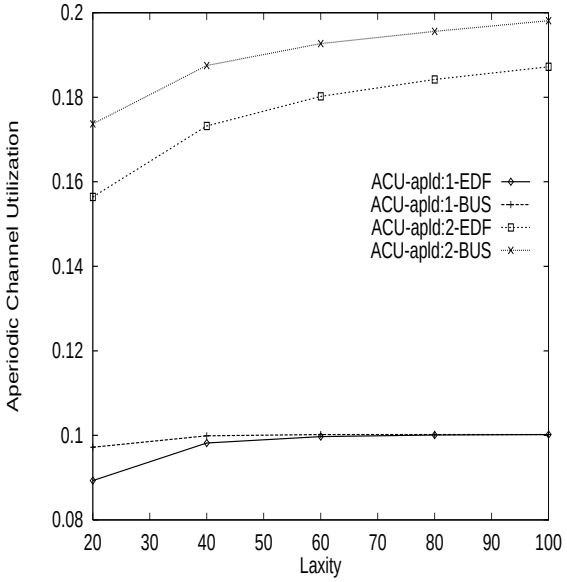

Fig 8. Effect of laxity of aperiodic messages

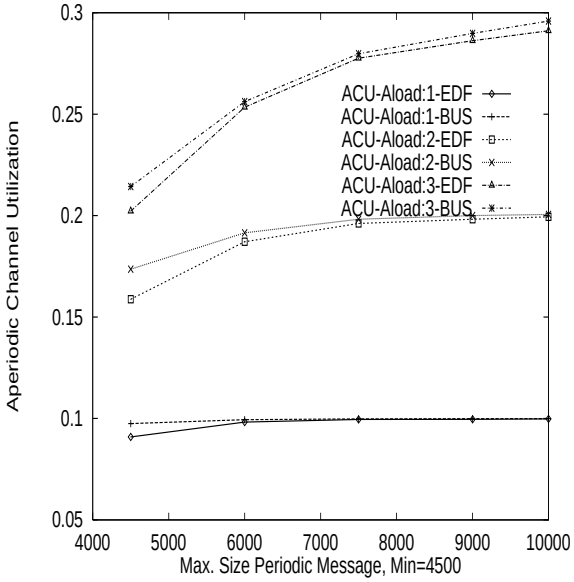

Fig 9. Effect of stochastic periodic message size

7. **Effect of server granularity**: Here, we examine the effect on ASR, by varying the AS granularity for the EDF protocol. The RS granularity can also be varied to improve response time and ASR for aperiodic messages. In Figures 10-14, the influence of RS granularity, laxity, periodic message load, and aperiodic message load on aperiodic message service performance (for varying AS granularity), is studied.

   (a) **Effect of RS granularity**: Figure 10 studies the influence on ASR by variation in AS granularity for RS granularity m_{rs} values equal to 1, 12, and 24. The curves are plotted for periodic and aperiodic message load of 1, aperiodic message size of 1000 bytes, and laxity value of 20. It can be observed that RS granularity significantly effects ASR. This is because as the granularity of RS decreases (m_{rs} increases), responsiveness to newly arrived requests increases. Since the period of RS becomes small, the number of new requests in that period decreases, decreasing the probability of collisions and as a result the efficiency of RS increases. ASR is maximum when the RS granularity is least i.e, $m_{rs} = 24$. The ASR increases with decrease in granularity, till $m_{as} = 12$ and then decreases for further decrease in granularity ($m_{as} = 24$). This is because the number of slots required for transmission of an aperiodic message equals the slots serviced per AS period when $m_{as} = 12$, and on further increase in m_{as} no aperiodic message is serviced in entirety in one period of AS and hence the decrease in ASR. It is to be observed that ASR is maximum for $m_{rs} = 24$ and $m_{as} = 3$. Appropriate choice of these two parameters needs to be made depending on the characteristics of aperiodic messages.

(b) **Effect of laxity**: Figure 11 shows the effect on ASR for variation in AS granularity for different laxities of aperiodic messages. It can be seen that the granularity effect is almost negligible for higher laxities. This is because with increase in laxity of aperiodic messages, the deadline of messages becomes larger than the highest granularity of AS with $m_{as} = 1$. The RS granularity effect can still be observed though it is moderate for higher laxities.

(c) **Effect of message size**: Figure 12 shows the effect on ASR for varying AS granularity for message size of 1000 and 5000 bytes. The laxity of message is based on its computation time and hence when message size increases laxity increases. Because of this, we see an increase in ASR for large message size. The peak shifts to $m_{as} = 3$, when message size is 5000 bytes. This is because of the reason mentioned earlier that for this value of m_{as}, message size matches the service time of the AS. It is also to be observed that when laxities are much larger than server period, then granularity of RS and AS does not influence ASR.

(d) **Effect of periodic load**: Figure 13 shows the effect on ASR for varying AS granularity for periodic load of 1 and 2. For this experiment the values of aperiodic load, laxity, and message size are fixed at 1, 20, and 5000, respectively. The same pattern is observed for both the loads. The vertical shift in the curves, for periodic load of 1 and 2, is due to increase in periodic message utilization which decreases idle time and hence decreases ASR.

(e) **Effect of aperiodic load**: Figure 14 shows the effect on ASR for varying AS granularity for aperiodic load of 1 and 2. For this experiment the values of periodic load, laxity, and message size are fixed at 1, 20, and 5000, respectively. It can be observed that ASR is maximum when AS service time matches the aperiodic message size. On further decrease in AS granularity ASR decreases and RS granularity does not have any impact for aperiodic load equal to 1. When the load is high, decrease in RS granularity has a negative effect. This is because at low granularities (large m_{rs}), since the load is high most of the request server slots are lost in collisions whereas when $m_{rs} = 1$, some of the aperiodic messages miss their deadlines hence contention is less, thereby improving RS efficiency.

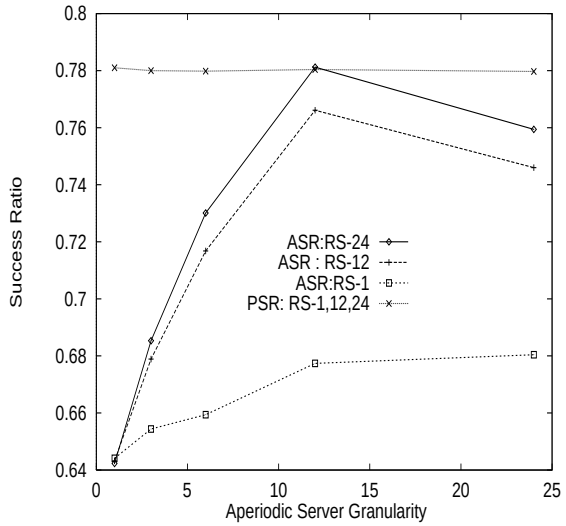

Fig 10. Effect of request server granularity

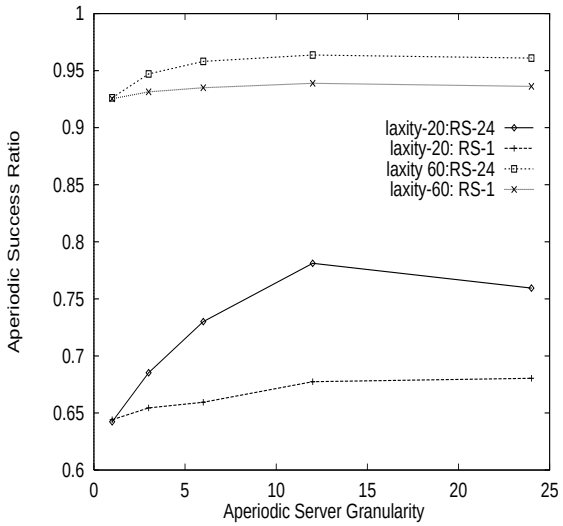

Fig 11. Effect of laxity

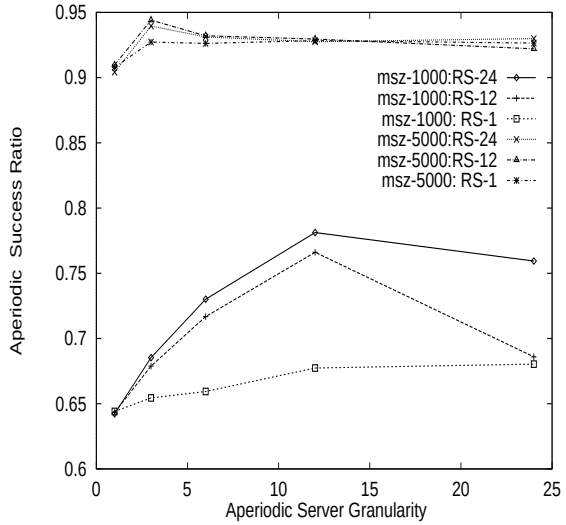

Fig 12.Effect of message size

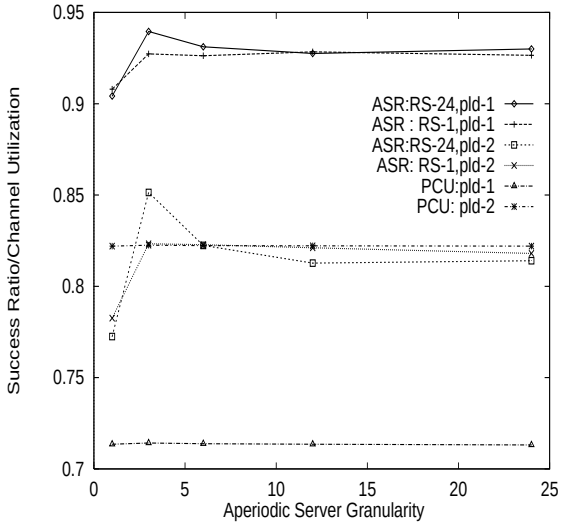

Fig 13. Effect of periodic load

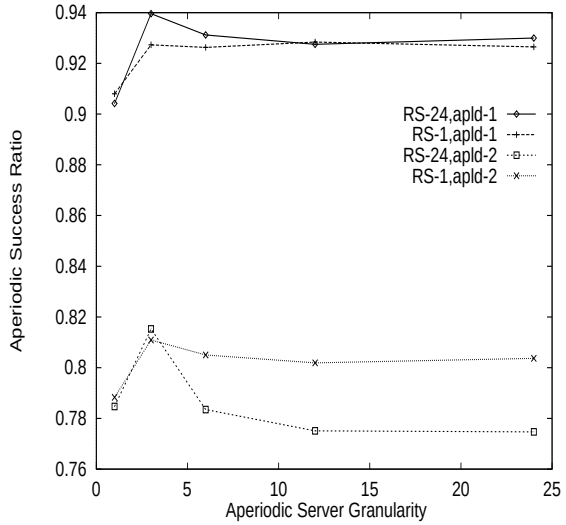

Fig 14. Effect of aperiodic load

5 Conclusion

Multiple access bus networks are being increasingly used in distributed real-time systems because of their simplicity, economy, and low propagation delays. In many real-time applications such as air-traffic control and multimedia, periodic message requests for channel establishment arrive dynamically. Any multiple access guarantee based protocol for such applications should (i) support real-time channel establishment, (ii) provide predictability, (iii) support integrated scheduling of periodic and aperiodic messages, and (iv) be distributed in nature. Earlier proposed protocols do not address all these requirements of real-time MAC protocols. We proposed two guarantee based real-time MAC protocols, EDF and BUS protocols, based on the EDF [16] and backplane bus scheduling [17] algorithms respectively. The protocols have the components -Request Server (RS), Aperiodic Server (AS), and run-time scheduling algorithms with their associated admission tests. RS is responsible for collecting channel establishment requests and for giving bounded response time to channel establishment requests. AS is used for scheduling aperiodic messages. The guaranteed messages (those that pass the admission test) are scheduled by the run-time scheduling algorithms so as to meet the deadlines of all the admitted messages. From the simulation studies it can be observed that:

- The EDF protocol offers better schedulability as compared to the BUS protocol.
- The BUS protocol offers better response time and higher channel utilization for aperiodic messages as compared to the EDF protocol.
- Channel utilization offered by the EDF protocol, for aperiodic messages, approaches that of BUS protocol when the laxity of aperiodic messages is large (does not hold for overloaded systems) while giving better utilization for periodic messages.
- When the periodic message size is stochastic in nature, channel utilization by aperiodic messages is much higher than the bandwidth reserved for them.

Messages in real-time systems have variable QoS requirements - some messages have hard deadlines while for others occasional deadline misses can be tolerated. The proposed protocols can be extended to exploit the variable QoS requirements of messages to improve schedulability and fault tolerance of the system. The (m, k)-firm guarantee model [19] which provides scheduling flexibilty by compromising on quality to meet deadlines, can be used to capture the QoS requirements of the message streams. In this model, for an (m, k)- firm guarantee stream, the result is of acceptable quality as long as at least m messages in a window of k consecutive messages meet their deadlines.

References

[1] L.Kleinrock, "Distributed systems", *Communications of the ACM*, vol.28, no.11, pp.1300-1213, Nov. 1985.

[2] J.A.Stankovic, "A perspective on distributed computer systems", *IEEE Trans. on Computers*, vol.33, no.12, pp.1102-1115, Dec. 1984.

[3] S.Yau, editor, "Special issue on distributed computer systems", *IEEE Trans. on Computers*, vol.38, no.8, Aug. 1989.

[4] D.Ferrari,and D.C.Verma, "A Scheme for real-time channel establishment in wide area networks", *IEEE Journal on Selected Areas in Communications*, vol.8, no.4, pp.368-379, Apr. 1990.

[5] D.D.Kandlur, K.G.Shin, and D.Ferrari, "Real-time communication in multi-hop networks", *IEEE Trans. on Parallel and Distributed Systems*, vol.5, no.10, pp. 1044-1055, Oct. 1994.

[6] J.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.

[7] W. Zhao and K. Ramamritham, "Virtual Time CSMA protocols for hard real-time communication," *IEEE Trans. on Software Engineering,* vol.13, no.8, pp.938-952, 1987.

[8] N.Malcom and W.Zhao, "Hard real-time communication in mutiple-access networks", *Real-Time Systems*, no.8, pp.35-77, 1995.

[9] P. Ulusoy, "Network access protocols for hard real-time communication systems," *Computer Communications,* vol.18, no.12, pp.943-948, 1995.

[10] C.C.Chou, and K.G.Shin, "Statistical real-time channels on multi-access bus networks", *IEEE Trans. on Parallel and Distributed Systems*, vol.8, no.8, pp.769-780, Aug. 1997.

[11] S. Norden, S. Balaji, G. Manimaran, and C. Siva Ram Murthy, "Deterministic protocols for real-time communication in multiple access networks," *Computer Communications,* vol.22, no.2, pp.128-136, Jan. 1999.

[12] M. Mock and E. Nett, "Real-time communication in autonomous robot systems," in Proc. *Intl. Symposium on Autonomous Decentralized Systems,* pp.34-41, 1999.

[13] S.K. Kweon and K.G. Shin, "Achieving real-time communication over Ethernet with adaptive traffic smoothing," in Proc. *Real-Time Technology and Applications Symposium (RTAS),* pp.90-100, 2000.

[14] S.Norden, G.Manimaran, and C. Siva Ram Murthy, "Dynamic planning based protocols for real-time communication in LAN and Switched LAN environments", *Computer Communications,* vol.24, no.13, pp.1256-1271, Aug. 2001.

[15] N.Homayoun, and P.Ramanathan, "Dynamic priority scheduling of periodic and aperiodic tasks in hard real-time systems", *Real-Time Systems*, vol.6, pp.207-232, June. 1994.

[16] C.L.Liu, and J.W.Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment", *Journal of the ACM*, vol.20, no.1, pp.45-61, Jan. 1973.

[17] S.H.Khayat, and B.D.Bovopoulos, "A simple and efficient bus management scheme that supports continuous streams", *ACM Trans. on Computer Systems*, vol.13, no.2, pp.122-140, May. 1995.

[18] V.Upadhya, "Design and analysis of real-time LAN protocols", *M.S.Thesis, Dept. of Computer Science and Engg., Indian Institute of Technology, Madras*, 1996.

[19] P.Ramanathan, "Graceful degradation in real-time control applications using (m,k)-firm guarantee," in Proc. *Fault-Tolerant Computing Symposium (FTCS),* pp.132-141, 1997.