Delivery of Time-Critical Messages Using a Multiple Copy Approach

PARAMESWARAN RAMANATHAN
University of Wisconsin
and
KANG G. SHIN
The University of Michigan

Reliable and timely delivery of messages between processing nodes is essential in distributed real-time systems Failure to deliver a message within its deadline usually forces the system to undertake a recovery action, which introduces some cost (or overhead) to the system. This recovery cost can be very high, especially when the recovery action fails due to lack of time or resources.

Proposed in this paper is a scheme to minimize the expected cost incurred as a result of messages failing to meet their deadlines. The scheme is intended for distributed real-time systems, especially with a point-to-point interconnection topology The goal of minimizing the expected cost is achieved by sending multiple copies of a message through disjoint routes and thus increasing the probability of successful message delivery within the deadline. However, as the number of copies increases, the message traffic on the network increases, thereby increasing the delivery time for each of the copies. There is therefore a tradeoff between the number of copies of each message and the expected cost incurred as a result of messages missing their deadlines. The number of copies of each message to be sent is determined by optimizing this tradeoff Simulation results for a hexagonal mesh and a hypercube topology indicate that the expected cost can be lowered substantially by the proposed scheme.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture—*store and forward networks;* C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications,* C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems—*real-time systems*

General Terms: Design

Additional Key Words and Phrases: Dynamic failure, time-constrained communication

---

A preliminary version of this paper appeared in the *Proceedings of Fault-Tolerant Computing Symposium,* 1991 The work reported here is supported in part by the National Science Foundation under grant MIP-9009154 and the Office of Naval Research under contract N00014-K-85-00122

Authors' addresses: P. Ramanathan, Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706-1691, email: parmesh@engr wisc.edu; K. G Shin, Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48109-2122, email: kgshin@eecs.umich.edu.

1. INTRODUCTION

Due mainly to their potential for high performance and high reliability, distributed computing systems are increasingly being used in the control of critical real-time applications such as avionics, life-support systems, nuclear power plants, drive-by-wire applications and computer-integrated manufacturing systems. A common feature of all these applications is that they can fail not only due to loss of components (*static failure*) but also due to time-critical tasks not completing their execution within the assigned deadline (*dynamic failure*). Since tasks need to exchange information among themselves in order to accomplish a common goal, the probability of dynamic failure can be reduced only if the system supports a mechanism for delivering messages in a reliable and timely fashion, i.e., every message generated by a real-time task must be delivered to the receiving task(s) before the assigned deadline.

Some "cost" will be incurred whenever a message misses its deadline because the system has to undertake a recovery action to retrieve the information contained in the message. The cost depends on the nature of the recovery action. For example, in some applications, the system may be able to recover at a low cost by simply estimating the information contained in the message. On the other hand, in some other applications, the recovery action might involve reexecution of some of the tasks in the system. In this case, the cost will depend on the nature and the number of tasks that need to be reexecuted. The primary objective of the scheme proposed in this paper is to minimize the expected cost incurred as a result of messages failing to meet their deadlines.

Most existing schemes for message passing under deadline constraints are mainly for distributed systems in which the nodes are interconnected through a multiple access network, such as a broadcast bus or a token ring. Zhao and Ramamritham compare several policies that each node can use to select the next message for transmission on a Carrier Sense Multiple Access/Collision Detect (CSMA/CD) network [14, 15]. They compare these policies in terms of four performance metrics: ratio of message loss, effective channel utilization, collision channel utilization and normalized average transmitted message length. They conclude that neither of these policies is better than the others in terms of all four of these performance metrics. Kurose et al. [7] propose a window-based protocol for time-constrained communication for CSMA/CD networks. They develop a policy that each node can use in selecting the next message for transmission so as to maximize the percentage of messages transmitted with a delay below a fixed bound K. A similar message selection problem was addressed by Strosnider et al. [13] for a token ring instead of a CSMA/CD network.

The problem with multiple access networks is that they are susceptible to single point failures. They are also not easily scalable to large applications due to bandwidth limitations of the multiple access medium. As a result, distributed systems with point-to-point interconnection networks such as meshes and hypercubes have recently gained considerable attention and hence the focus of attention in this paper.

The basic idea of the scheme proposed in this paper is as follows. Smaller message delivery times are achieved by making use of the multiple disjoint routes that exist in a point-to-point interconnection topology. Depending on the deadline and the criticality of the message, the orginating/source node chooses to send one or more copies of the message through disjoint routes. A message that is sent through more than one disjoint route has a higher probability of meeting the deadline because at least one of the copies is likely to traverse a less congested route and hence get delivered before the deadline. However, by sending more copies, the average message traffic on the network increases, thereby possibly increasing the time required to deliver each copy. In the proposed scheme, the number of copies to be sent is chosen in such a way that the expected cost incurred as a result of messages failing to meet their deadlines is minimized.

The proposed approach was evaluated by determining the number of copies of a message to be sent in two different point-to-point interconnection networks: C-wrapped hexagonal mesh [1] and hypercube [11]. These copies were then used in a simulator to gauge the benefits of the proposed approach. In all the cases considered, the reduction in the expected cost were substantial. In most cases, the reductions were as large as 70%.

An additional advantage of the proposed approach is that it is less likely to be susceptible to node/link failures. The likelihood of losing a critical message due to a faulty node/link is reduced because the proposed approach will send more copies of a very critical message through disjoint routes. The use of multiple copies to ensure reliable delivery in the presence of component failures has been dealt with elsewhere [3, 8, 9]. Our work is unique in the sense that it deals with the problem of dynamic failures due to congestion. One could integrate our approach with other schemes for ensuring reliable delivery. However, description of such an integrated scheme is beyond the scope of this paper.

This paper is organized as follows. Section 2 describes the target distributed system. Section 3 describes the proposed scheme. The objective used to determine the optimal number of copies is formalized in Section 4 and an evaluation of the proposed approach is presented in Section 5. The paper concludes with Section 6.

2. SYSTEM MODEL

As mentioned earlier, the target is a distributed system with a point-to-point interconnection topology because of its potential for high performance and high reliability using the multiple processing nodes and multiple paths between any pair of nodes in the system. Each node in the system executes a set of real-time tasks, each with an assigned deadline. Based on the *task deadline*, a deadline is assigned to every message the task sends (i.e., *message deadline*). Some cost will be incurred whenever a message fails to meet its assigned deadline. The goal is to develop a message passing scheme that minimizes the expected cost incurred as a result of messages missing their deadlines.

The messages generated in the system and the message passing scheme are assumed to have the following characteristics:

M1 Messages are generated at each node according to a Poisson process with rate λ_g.

M2 The message lengths are exponentially distributed with mean $\bar{l}$. This in turn implies an exponentially distributed message service time at each node with rate μ, where $\mu = \alpha/\bar{l}$ for some constant α that represents the time required to transmit one byte if $\bar{l}$ is expressed in bytes.

M3 Messages belong to one of n criticality classes. A cost K_i is associated with each class i, where K_i represents the average cost incurred by the system as a result of a message in that class missing its deadline. The probability of a generated message belonging to class i is denoted by c_i.

M4 Nodes have no preferential direction for communication. The probability of a node sending a message to a node that is h hops away is q_h. For convenience of presentation, this probability is assumed to be independent of the criticality class of the message.

M5 Each message is assigned a deadline at the time of its generation. The deadlines of messages of class i that have to traverse h hops are assumed to be distributed around a mean η_{ih} with a probability density function $f_D(i, h, t)$.

M6 Messages with higher criticality[1] are selected (for transmission) ahead of messages with lower criticality. Among messages with same criticality, the selection is based on a first-come-first-served basis.

Modeling assumptions M1–M5 define the characteristics of a message in our system while M6 specifies the message scheduling algorithm used by each node in the system. In particular, M6 states that we will focus only on the first-come-first-served scheduling policy among messages of same class as opposed to more deadline-cognizant policies such as the minimum-deadline-first. The reason for this focus is two-fold. First, the first-come-first-served policy is more analytically tractable than minimum-deadline-first policy. Second, as shown by Rupnick and Ramanathan [10], in a point-to-point interconnection, the minimum-deadline-first policy performs only marginally better than the first-come-first-served policy because, in a point-to-point interconnection network, intermediate nodes usually cannot determine whether a given message will miss its deadline unlike in a distributed system with a multiple access interconnection network.

In addition to M1–M5, the following two assumptions are also made only for the purpose of efficiently determining the number of copies of each message to be sent. Even without these two assumptions, the benefits of the multiple copy approach are substantial as shown later in Section 5.

[1] A message is said to have *higher criticality* if it belongs to class with larger average cost.

M7 The length of a message is regenerated at each intermediate node of its route and is independent of its length at other intermediate nodes.

M8 The delivery time of messages through disjoint routes are mutually independent.

M7 is commonly referred to as Kleinrock's independence assumption [5]. Although M7 is unrealistic in practice, several empirical studies have shown that the mean message delivery times computed using this assumption closely matches the actual mean message delivery times observed in practice [5, 6]. However, the knowledge of mean message delivery times is not sufficient to determine the expected cost incurred as a result of messages failing to meet their deadlines. To compute the expected cost, we need to determine the probability distribution function (PDF) of message delivery times.

The PDF of message delivery times cannot be accurately determined without using M7 and M8. However, even a crude approximation of the PDF seems to be sufficient to determine the number of copies of each message that results in a substantial reduction in the expected cost incurred by the system. This result is demonstrated with the help of a simulator that does not make use of these two assumptions. When the number of copies as determined by using the approximate PDF was used in the simulator, the reductions in the expected cost were over 70% in most cases.

3. PROPOSED SCHEME

In the proposed scheme, the expected cost resulting from missed message deadlines is lowered by making use of the multiple disjoint routes between nodes of the distributed system. When a message is selected for transmission at the source node, the message scheduler at that node determines the number of copies of this message to be sent on the network. These copies will be routed to the destination node through a set of disjoint routes. The rationale behind this approach is that the more the number of copies, the higher the probability that at least one of these copies will reach the destination before the deadline. However, as the number of copies increases, the message traffic on the network increases, thereby increasing the delivery time for each copy. There is therefore a tradeoff between the number of copies of each message and the expected cost incurred as a result of messages missing their deadlines.

Intuitively, the number of copies selected by the source node for a given message should depend on three factors: its criticality and deadline, and the number of hops it has to traverse. For example, when the other two factors are identical, it is advantageous to send more copies of a message with higher criticality, or the message with a shorter deadline or the message that has to traverse more hops. However, determining the optimal number of copies to be sent based on all the three factors can be computationally very expensive. Therefore, in this paper, the number of copies for a particular message will be based only on the criticality and the number of hops the message has to traverse. Simulation results indicate that ignoring the deadline factor while

determining the number of copies has little effect, because the reductions in the expected cost while considering only the other two factors are over 70% for most realistic network loads.

3.1 Formal Description

The message handler at each node can be described in terms of two concurrent processes: *S_process* and *D_process*. The *S_process* is responsible for transmitting the messages to the neighboring nodes while *D_process* is responsible for receiving the messages destined for the local node. Formal descriptions of *S_process* and *D_process* are shown in Figures 1 and 2, respectively. In these descriptions each message is represented by a triple (c, d, h) where c is the criticality of the message, d is the deadline, and h is the number of hops the message has to traverse. The descriptions are also made in terms of the following global parameters and functions.

ρ_g	The normalized message generation rate at each node. That is, it is the ratio of the message generation rate at each node (λ_g) to the message service rate at that node (μ).
Copies[ρ_g][c][h]	A precomputed array containing the number of copies of a message to be sent given the normalized message generation rate (ρ_g), the criticality of the message (c), and the number of hops the message has to traverse (h). An algorithm to compute this array is discussed in Section 4.2.
next( )	Returns the next message to be transmitted based on some message scheduling policy. This call blocks if there are no messages for transmission.
send(M, k)	Determines k disjoint routes to the destination of message M and then transmits copies of the message to the neighboring nodes identified by the disjoint routes.
relay(M)	Relays a single copy of the message to the neighbor specified in the route of the message.
receive( )	Returns a message that has not been processed from the set of messages that are destined for the local node. This call blocks if there are no messages for processing.

S_process: This process is responsible for transmitting the messages from a node. If the message being transmitted was generated locally, then *S_process* determines the number of copies of the message to be sent and the route for each of the copies. It then transmits the copies to the neighbors identified by the disjoint routes. On the other hand, if the message originated from some other node, then *S_process* simply relays the message to the neighbor specified in the route of the message.

When *S_process* has to determine the number of copies of a message to be sent, it looks up a table of precomputed values. The main advantage of this table-driven scheme is that the overhead is minimal when the system is in

```
1.  loop forever
2.      (c, d, h) := next( );
3.      if (c,d,h) was generated locally
4.          no_of_copies := Copies[ρ_g][c][h];
5.          send((c, d, h), no_of_copies);
6.      else
7.          relay((c, d, h));
8.      endif
9.  endloop.
```

Fig. 1. *S-process.*

```
1.  loop forever
2.      (c, d, h) := receive( );
3.      if (c, d, h) is the first copy to be received
4.          deliver (c, d, h) to the waiting process;
5.      else
6.          discard (c, d, h);
7.      endif
8.  endloop.
```

Fig 2. *D-process.*

operation. At first glance, this scheme might seem to be memory-intensive. However, as we will show later, the optimal number of copies of each message does not change rapidly with ρ_g. Therefore, one can easily partition the range of ρ_g, namely $[0, \beta)$, for some $\beta < 1$, into a few disjoint intervals and assume that the optimal solution does not change within each interval. It is necessary to restrict ρ_g to the range less than $[0, 1)$ because each node has to service not only all the messages generated at that node, but also all the transit messages that are being relayed through the node. The value of β depends on the topology, the communication pattern between the real-time tasks and the message routing algorithm used. For a given topology, one can easily compute the value of β if the associated queueing network has a product-form solution [4]. The assumptions M1–M8 are sufficient to ensure a product-form solution for the associated queueing network. For a regular homogeneous topology satisfying the assumptions M1–M8, one can show that the range of ρ_g (or equivalently the range of λ_g) should be such that $\rho_t = \lambda_t/\mu < 1$ where λ_t is related to λ_g by the equation $\lambda_t = \lambda_g \sum_{h=1}^{H} h \cdot q_h$, where H is the maximum number of hops a message has to traverse in this topology and q_h is as defined in M4.

If more than one copy of the message is to be sent, then *S_process* has to determine a set of disjoint routes (one for each copy) to the destination. This can be easily done for most regular topologies as illustrated in Section 5 for a hypercube [11] and C-wrapped hexagonal mesh [1] topologies.

D_process: This process is responsible for receiving the messages destined for a local node. Since the local node will receive one or more copies of each message, *D_process* has to identify the duplicate copies of each message. This can be done by using message identifiers. Such techniques are commonly used to recover from lost messages. Therefore, there is no real additional penalty at the destination end in using the proposed scheme.

It is clear from the above description that the time overhead of using the proposed scheme is minimal while the system is in operation. However, to demonstrate the viability of our scheme, we need to show that

—the precomputed table of values used by *S_process* can be easily determined off-line; and

—there is significant reduction in the expected cost incurred as a result of using the proposed scheme.

The rest of this paper addresses these two issues.

4. DETERMINATION OF THE OPTIMAL NUMBER OF COPIES

The derivation of the probability distribution function (PDF) of message delivery times in a system satisfying the assumptions M1–M8 is described below. Using this PDF, an analytic estimate of the expected cost incurred by the system due to missed deadlines is derived. This estimate is then used to determine the optimal number of copies for a given message.

4.1 PDF of Message Delivery Times

It is very difficult to compute the exact PDF of message delivery times in a queueing network with multiple classes of messages and prioritized service. Therefore, we approximate the PDFs as follows. Without loss of generality, one can assume $K_1 \geq K_2 \geq \cdots \geq K_n$. Consider the messages in class 1. Since the messages in this class have the highest priority, their delivery times are unaffected by the existence of the other classes of messages. Therefore, for this class, the system behaves as a "product-form network" [4] in which each node acts as an independent M/M/1 queue with arrival rate $\lambda_{t,1}$ and service rate μ, where $\lambda_{t,1}$ is the total throughput at each node due to messages of class 1 generated at that node and in-transit messages of class 1 relayed through that node. For a regular homogeneous topology, $\lambda_{t,1}$ can be derived from λ_g as:

$$\lambda_{t,1} = c_1 \lambda_g \sum_{h=1}^{H} h \cdot q_h$$

where c_1, λ_g, H, and q_h are as introduced in Section 2.

From the above facts, one can derive the PDF of delivery times for a message of class 1 traversing h hops [5] as:

$$F_h(\lambda_g, \mu, t \mid 1) = 1 - e^{-\Lambda_1 t} \sum_{k=0}^{h-1} \frac{(\Lambda_1 t)^k}{k!} \tag{4.1}$$

$$\text{where} \quad \Lambda_1 = \mu\left(1 - \frac{\lambda_{t,1}}{\mu}\right). \tag{4.2}$$

Now consider messages of class 2. The messages in this class have to wait behind messages of class 1. Computation of the PDF of delivery times for this class is not easy. Therefore, we approximate the PDF by using expressions similar to that of class 1 except with higher message generation rates. That is, let

$$\lambda_{t,2} = \lambda_g(c_1 + c_2) \cdot \sum_{h=1}^{H} h \cdot q_h$$

$$F_h(\lambda_g, \mu, t \mid 2) = 1 - e^{-\Lambda_2 t} \sum_{k=0}^{h-1} \frac{(\Lambda_2 t)^k}{k!}$$

$$\text{where} \quad \Lambda_2 = \mu\left(1 - \frac{\lambda_{t,2}}{\mu}\right).$$

Proceeding similarly, we can approximate the PDF of delivery times of any class $i = 1, 2, \ldots, n$ as

$$F_h(\lambda_g, \mu, t \mid i) = 1 - e^{-\Lambda_i t} \sum_{k=0}^{h-1} \frac{(\Lambda_i t)^k}{k!}$$

$$\text{where} \quad \Lambda_i = \mu\left(1 - \frac{\lambda_{t,i}}{\mu}\right) \quad \text{and} \tag{4.3}$$

$$\lambda_{t,i} = \lambda_g \sum_{k=1}^{i} c_k \sum_{h=1}^{H} h \cdot q_h.$$

4.2 Formulation of the Objective Function

This subsection formalizes the objective used to compute the number of copies of a message to be sent given its criticality class and deadline, and the number of hops it has to travel. The objective is to minimize the expected cost incurred as a result of messages missing their deadlines.

The expected cost can be expressed as

$$E[\text{cost}] = \sum_{i=1}^{n} \sum_{h=1}^{H} c_i \cdot q_h \cdot K_i \cdot P[\text{message missing deadline} \mid i, h], \tag{4.4}$$

where H is the maximum number of hops a message has to traverse in the interconnection network, c_i, K_i, and q_h are as defined in modeling assumptions M1–M4, and P[message missing deadline $\mid i$, h] is the conditional

probability of message missing its deadline given that it belongs to class i and that it has to traverse h hops.

The P[message missing deadline $|$ i, h] is a function of the network traffic and the deadline of the message. It can be computed from the PDF of message delivery times as follows. Let $m_i(\rho_g, h, t)$ denote the design parameter that is equal to the number of copies used by the source node for a message of class i, traversing h hops and having a deadline t when the normalized message generation rate at each node is ρ_g. Then, the increased message traffic on the network caused by the additional copies of each message can be thought of as an increased message generation rate. That is, the effective message generation rate at each node, λ_e, is given by

$$\lambda_e = \lambda_g \cdot \sum_{i=1}^{n} \sum_{h=1}^{H} \int_{t=0}^{\infty} c_i \cdot q_h \cdot m_i(\rho_g, h, t) \cdot f_D(i, h, t)\, dt. \tag{4.5}$$

In the above equation, $c_i \cdot q_h$ is the probability of a message being in class i and traversing h hops, $m_i(\rho_g, h, t)$ is the number of copies of that message and $f_D(i, h, t)$ is the probability density function of the deadline of the message. If the multiple copy approach is not being used, then $m_i(\rho_g, h, t) = 1$ for all i, ρ_g, h and t and therefore, $\lambda_e = \lambda_g$. However, if the multiple copy approach is being used, then $m_i(\rho_g, h, t) > 1$ for some i, ρ_g, h and t. In that case, $\lambda_e > \lambda_g$.

For this effective message generation rate, the probability of timely delivery of a single copy of a given message of class i with deadline t and traversing h hops is $F_h(\lambda_e, \mu, t \mid i)$ where $F_h(\lambda_e, \mu, t \mid i)$ is given by Eq. (4.3). Therefore, from M8, if $m_i(\rho_g, h, t)$ copies of the message are sent, then

$$P[\text{message missing deadline} \mid i, h] = \int_{t=0}^{\infty} \left(1 - F_h(\lambda_e, \mu, t \mid i)\right)^{m_i(\rho_g, h, t)} \cdot f_D(i, h, t)\, dt. \tag{4.6}$$

Combining Eqs. (4.4) and (4.6), we get

$$E[\text{cost}] = \sum_{i=1}^{n} \sum_{h=1}^{H} \int_{t=0}^{\infty} c_i \cdot q_h \cdot K_i \cdot \left[1 - F_h(\lambda_e, \mu, t \mid i)\right]^{m_i(\rho_g, h, t)} \cdot f_D(i, h, t)\, dt. \tag{4.7}$$

The problem then is to determine a set functions $m_i(\rho_g, h, t)$ that minimizes the expected cost in Eq. (4.7). Clearly, the above objective is difficult to minimize. We therefore impose the following restrictions on the possible solutions for $m_i(\rho_g, h, t)$.

(1) $m_i(\rho_g, h, t_1) = m_i(\rho_g, h, t_2)$ for all $t_1, t_2 \in [0, \infty)$.

(2) $m_i(\rho_g, h, t) \leq M_h$ for some constants $M_h, h = 1, 2, \ldots, H$ and for all $t \in [0, \infty)$.

(3) $K_i > K_j$ implies $m_i(\rho_g, h, t) \geq m_j(\rho_g, h, t)$ for all $t \in [0, \infty)$.

(4) $m_i(\rho_g, h_1, t) \geq m_i(\rho_g, h_2, t)$ if $\dfrac{\text{var}(D_{h_1})}{\eta_{h_1}} \geq \dfrac{\text{var}(D_{h_2})}{\eta_{h_2}}$ for all $h_1, h_2 \in \{1, 2, \ldots, H\}$, and for all $t \in [0, \infty)$, where D_{h_1} and D_{h_2} are the random variables modeling the delivery time of messages traversing h_1 and h_2 hops, respectively, and var$(\cdot)$ denotes the variance.

The first condition means that the deadline of the message is ignored when making decisions on the number of copies to be sent. The second condition arises from the connectivity of the interconnection topology. Basically, M_h denotes the total number of disjoint routes between any two nodes that are h hops away from each other. The third condition implies that more copies of a message are sent if the message has higher criticality. This is intuitively appealing because a higher cost will be incurred if a more critical message misses its deadline. The fourth condition is based on the observation that a larger variance in message delivery times usually implies that more messages will miss their deadlines and hence it is better to send more copies of those messages.

With the above simplifications, it is possible to search the space for an optimal solution. The results obtained by minimizing the simplified objective for a C-wrapped hexagonal mesh and hypercube topologies are given in the following section. Due to the above simplifications and due to the approximations made in estimating the PDF of message delivery times, the results obtained are suboptimal. Further improvements would be achieved if we could efficiently determine the optimal number of copies without any of the above simplifications.

5. EVALUATION OF THE PROPOSED APPROACH

In order to evaluate the proposed approach, simulations were run for two different point-to-point interconnection topologies, C-wrapped hexagonal mesh [1] and the hypercube topology [11]. The goal of these simulations were two-fold: (1) to evaluate the impact of some of the simplifying assumptions made in the heuristic method described in the previous section to determine the optimal number of copies, and (2) to gauge the benefits of the proposed approach under different environments.

To accomplish the first goal, a simulator was written to implement the traditional single copy approach and the proposed multiple copy approach using as few assumptions as possible. In particular, the key differences between the simulator and the proposed heuristic method are

(1) The simulator does not make use of the assumptions M7 and M8. That is, the length of a message is determined at the time of its birth and it remains unchanged while it is being relayed through the network. Furthermore, the actual delivery times of the individual copies of a message are used to determine whether a message missed its deadline;
(2) The simulator does not make any assumptions about the PDF of message delivery times as was necessary in the heuristic method to deal with priority among different classes of message; and

(3) The simulator takes into account many realistic constraints such as length of message headers and the time overhead to make routing decisions at each intermediate node.

As a result of these differences, the reduction in the expected cost observed in the simulator reflects the actual reduction one would observe in practice whereas the heuristic method provides only an estimate of the reduction in the expected cost.

To accomplish the second goal, the simulations were run for a wide range of system parameters. Variations in tightness of deadlines, communication pattern, network loads and ratios of cost of different classes were investigated. Presented below are more details on the range of parameters investigated.

(1) The message generation at each node is assumed to be a Poisson process with rate λ_g. The value of λ_g determines the message traffic on the network and is therefore chosen to correspond to a particular network load, ρ_t. The network load is varied between 0.1–0.9.

(2) Messages are assumed to belong to one of three classes. The probabilities of a message being in class 1, 2, or 3 are assumed to be 0.1, 0.25, 0.65, respectively. Several different cost functions are investigated to evaluate the proposed approach under different environments. The results for the following three cases are shown in this section:

(a) $K_1 = 625.0$, $K_2 = 25.0$, $K_3 = 1.0$
(b) $K_1 = 10.0$, $K_2 = 5.0$, $K_3 = 1.0$
(c) $K_1 = 1.0$, $K_2 = 1.0$, $K_3 = 1.0$.

In the first case, there is a wide disparity in the cost among the three classes. In the second case, there is a moderate disparity while in the third case there is no disparity in the cost among the three classes.

(3) The time for transmitting a message is proportional to its length. Each node is assumed to be capable of transmitting at a rate of 667,000 bytes/sec to a neighboring node. Each node can be transmitting and receiving messages to and from all its neighbors at this rate at the same time. This transmission rate matches the capability of a node in HARTS.

The length of a message is assigned at the time of its generation from an exponential distribution. Messages of class 1, 2, and 3 have mean lengths 64, 128, and 192, respectively in the results presented here.

(4) Two different communication patterns are considered. In the first pattern, the probability of a node communicating with any other node is assumed to be inversely proportional to the length of the shortest route between the two nodes while in the second pattern, each node is equally likely to communicate with any other node.

(5) Each message is assigned a deadline at the time of its generation. The deadline is assigned from a uniform distribution in the interval $[a_1 \cdot M, a_2 \cdot M]$ for some constants a_1, a_2, and M, $a_2 \geq a_1$. The parameter a_1 determines the tightness of the deadline while the difference between a_2

and a_1 determines the variation in the deadline among different messages. The parameter M is an arbitrary scaling constant because given a value for M one can obtain any uniform distribution by choosing appropriate values for a_1 and a_2. In the results presented here, the value of M for a given message is arbitrarily chosen to be the mean delivery time of that message if the network load is 0.3 and a_2 is set to be very large. The parameter of a_1 is varied in the range 2–3.5 where 2 corresponds to a very tight deadline and 3.5 corresponds to a very loose deadline.

5.1 Results for C-Wrapped Hexagonal Mesh Topology

A C-wrapped hexagonal mesh topology [1, 12] is a regular, homogeneous graph in which each node has six neighbors. The graph can be visualized as a simple hexagonal mesh with wrap links added to the nodes on the periphery. A simple hexagonal mesh is composed of a set of concentric hexagons with a central node, where each hexagon has one more node on each edge than the one immediately inside of it. It can be defined succinctly as follows.

Definition 1. A *C-wrapped hexagonal mesh* of dimension e, H_e, is comprised of $3e(e-1)+1$ nodes, labeled from 0 to $3e(e-1)$, such that each node s has six neighbors $[s+1]_{3e^2-3e+1}$, $[s+3e-1]_{3e^2-3e+1}$, $[s+3e-2]_{3e^2-3e+1}$, $[s-1]_{3e^2-3e+1}$, $[s-3e+1]_{3e^2-3e+1}$, and $[s-3e+2]_{3e^2-3e+1}$, where $[a]_b$ denotes a mod b.

Figure 3 shows a C-wrapped hexagonal mesh of dimension 5. This topology is currently being used for the construction of two experimental distributed systems: the HARTS at the Real-Time Computing Laboratory, The University of Michigan [1], and the Mayfly systems at Hewlett Packard Research Laboratories [2].

The key aspect of this topology is that there are pairs of nodes with only one shortest disjoint route between them. For example, in an H_5, nodes 0 and 42 have only one shortest route between them (see Figure 3). However, all pairs of nodes do have six disjoint routes between them, but not necessarily of the same length. For instance, the six disjoint routes between nodes 0 and 42 in an H_5 are of lengths 3, 4, 4, 6, 6, and 6. One of the reasons for studying this topology is to see whether the multiple copy scheme is advantageous even when the multiple routes are not of equal length.

It can be shown that the diameter of H_e is $e-1$ [1]. Furthermore, there are $2e-3$ different categories of nodes in an H_e with respect to the lengths of six disjoint routes. For example, consider all the nodes that are four hops away from node 0 in Figure 3, i.e., nodes on the periphery. Six of these nodes, namely 4, 56, 52, 57, 5, and 9, lie on one of the six *principal axes* of the C-wrapped hexagonal mesh [1]. As a result, the length of the disjoint routes to any of these six nodes differ from the length of the disjoint routes to any of the other nodes. In particular, the disjoint routes from node 0 to any of the peripheral nodes that lie on the principal axes are of lengths 4, 5, 5, 5, 6 and 6, as opposed to 4, 4, 5, 5, 6, and 6 for nodes that do not lie on the principal axes. Consequently, with respect to the lengths of the six disjoint routes, the

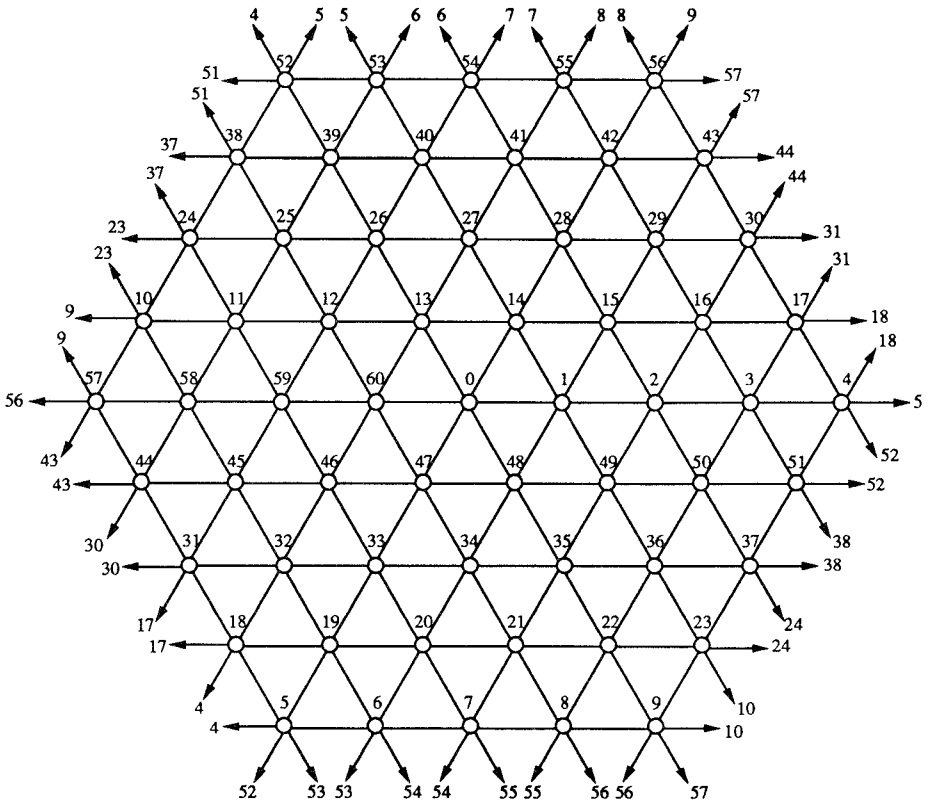


Fig. 3. H_5, a C-wrapped hexagonal mesh of dimension 5.

nodes that are four hops away from node 0 can be partitioned into two categories: nodes that lie on one of the principal axes and nodes that do not lie on any of the principal axes. A similar partitioning can be done for the set of nodes that are h hops away from node 0 for any $h > 1$. For $h = 1$, all nodes lie on the principal axes. Therefore, there are a total of seven categories of nodes with respect to length of the disjoint routes in an H_5.

Table I shows the number of copies as determined by the proposed heuristic method for a few selected network loads in an H_5. Category 1 contains nodes that are one hop away from node 0 while categories h and $h + 1$ for $h > 1$ contain nodes that are h hops away from node 0.

Figure 4 shows the effect of tightness of deadline on the expected cost in the traditional single copy approach and the proposed multiple copy approach. Note that this represents the differences in the application since the deadlines are application-dependent, i.e., this figure compares the benefits of the proposed approach for four different applications when executed on the same distributed system. The deadlines are fairly tight in Figure 4a and fairly loose in Figure 4d. As expected, the cost incurred by the system increases in the traditional and the proposed approach. However, the costs

Table I. Number of Copies as Determined by the Proposed Heuristic Method for a H_5 with Parameters $K_1 = 625$, $K_2 = 25$, $K_1 = 1$, Falloff Communication Pattern and $a_1 = 2.5$

Catgs.	Criticality Class 1	2	3
1	4	3	3
2	3	2	2
3	3	2	2
4	3	2	2
5	3	2	2
6	3	2	2
7	3	2	2

(a) $\rho_t = 0.1$

Catgs.	Criticality Class 1	2	3
1	3	2	2
2	2	2	2
3	2	2	2
4	2	1	1
5	2	1	1
6	2	1	1
7	2	1	1

(b) $\rho_t = 0.3$

Catgs.	Criticality Class 1	2	3
1	3	2	2
2	2	1	1
3	2	1	1
4	2	1	1
5	2	1	1
6	2	1	1
7	2	1	1

(c) $\rho_t = 0.5$

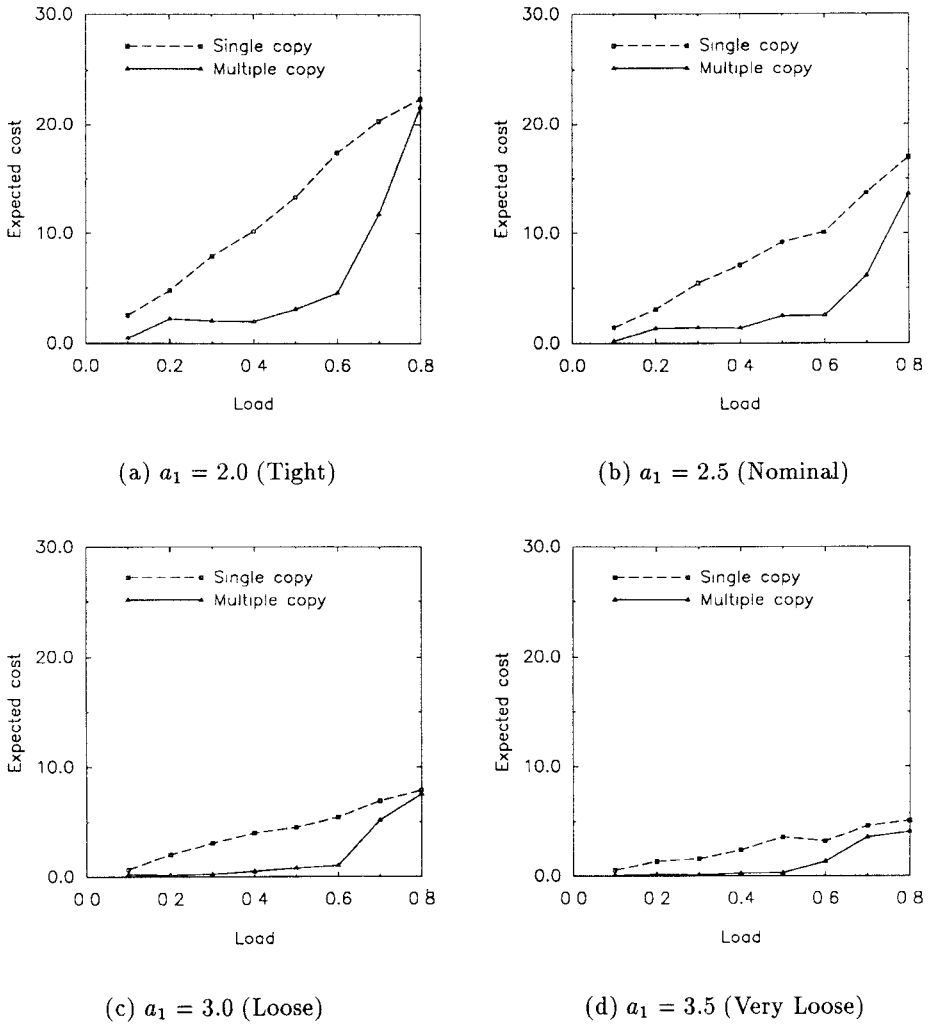
Catgs.	Criticality Class 1	2	3
1	2	1	1
2	2	1	1
3	2	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1

(c) $\rho_t = 0.7$

incurred in the proposed approach are substantially lower than in the traditional approach.

Table II shows the percentage reduction in the expected cost due to the multiple copy approach for the four cases shown in Figure 4. Note that the reductions in most cases are over 70%. One can also observe that the benefits of the proposed approach drop significantly for loads greater than 0.7. This is, however, not a serious problem because distributed real-time systems usually cannot operate at such high loads. This is because at high loads the percentage of missed deadlines become unacceptable. For instance, at a load of 0.8, approximately 33% of class 2 messages missed their deadlines in the single copy approach when the deadlines were tight (i.e., $a_1 = 2$).

The % reduction in the cost shown in Table II shows the actual reduction observed in the simulator using the number of copies as determined by the



(a) $a_1 = 2.0$ (Tight)

(b) $a_1 = 2.5$ (Nominal)

(c) $a_1 = 3.0$ (Loose)

(d) $a_1 = 3.5$ (Very Loose)

Fig. 4. Variation in the tightness of deadlines.

heuristic method in the previous section. However, as stated earlier, further improvements can be achieved if we could efficiently determine the optimal number of copies without any of the simplifications needed in the heuristic method. To get a feel for the potential improvement that can be achieved, a better solution was obtained using only the simulator, i.e., without the simplifying assumptions. It required a couple of days of CPU time on a SUN 4/60 to determine this better solution as opposed to few minutes of CPU time in the heuristic method. Figure 5 compares the resulting cost of the multiple copy approach using copies as determined by the heuristic method and copies as determined using only the simulator for parameters corresponding to Figure 4b. From this figure, one can conclude the proposed heuristic method

Table II. % Reduction in Expected Cost for Four Different Deadline Environment

Load	% reduction in cost			
	$a_1 = 2.0$	$a_1 = 2.5$	$a_1 = 3.0$	$a_1 = 3.5$
0.1	81.6	85.9	67.4	88.4
0.2	54.3	55.9	91.1	88.9
0.3	74.4	74.3	91.1	91.1
0.4	80.6	80.6	87.4	88.8
0.5	76.8	73.3	82.7	91.4
0.6	73.8	75.2	81.2	58.3
0.7	42.2	55.3	25.4	22.8
0.8	3.4	19.6	4.3	20.2

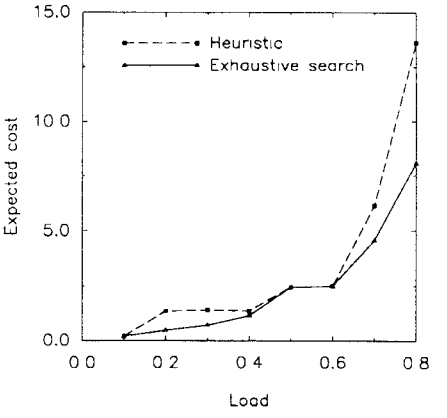
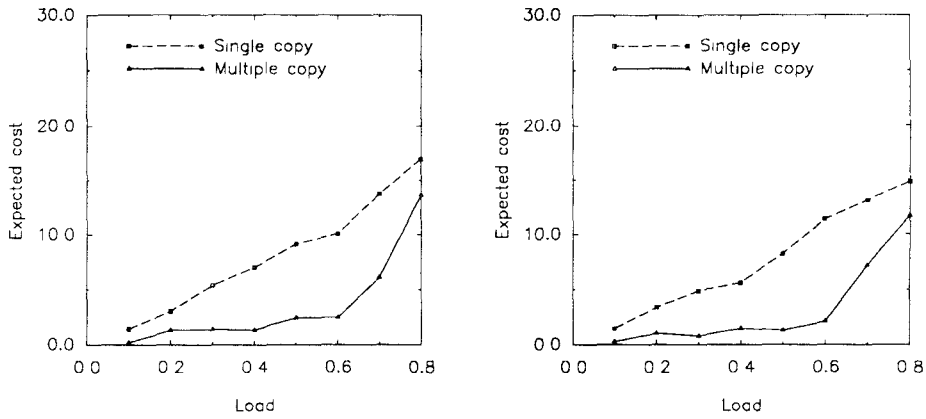


Fig. 5. Comparison of proposed method with the solution obtained from an exhaustive search

does fairly well at low and moderate loads. At high loads, the solution obtained by the heuristic method can be improved substantially. However, this is not a major concern since most real-time applications cannot be operated at these high loads because the number of messages that miss their deadlines at these loads is very high.

Figure 6 shows the effect of communication pattern on the expected cost in the traditional single copy approach and the proposed multiple copy approach. In Figure 6a, the probability of a node communicating with another node is inversely proportional to the shortest distance between them. This corresponds to a situation in which the designer has carefully mapped the given application in such way that nodes which communicate more often are mapped onto nodes that are closer to each other. In contrast, in Figure 6b, each node is equally likely to communicate with any other node. This corresponds to a situation in which no special attention has been given to the communication pattern when mapping the application onto the nodes of the distributed system. Here again, the proposed approach performs significantly better than the traditional single copy approach.



(a) Falloff (b) Uniform

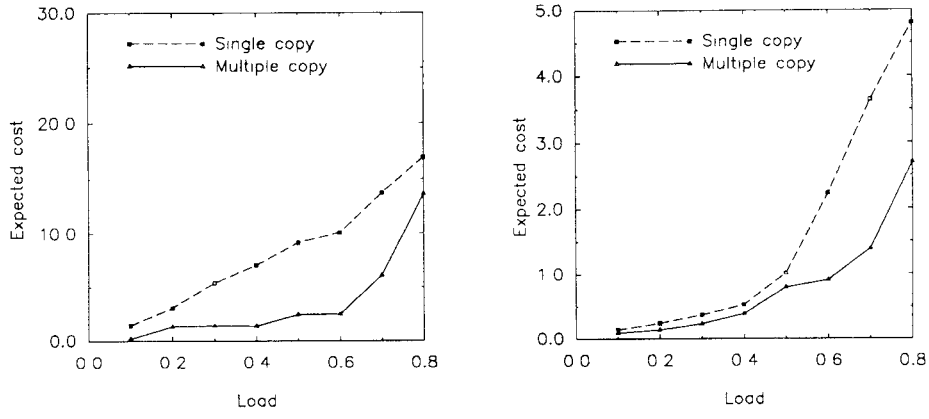
Fig. 6. Variation in the communication pattern.

Figure 7 shows the effect of disparity in the cost of different classes of message. Figure 7a corresponds to the situation where there is a considerable disparity in the costs for different classes, Figure 7b shows the results for a moderate disparity in the costs while Figure 7c shows the results for the case in which all classes have an identical cost. Here, the proposed approach does not always perform significantly better than the traditional approach. When all classes have the same cost, the proposed approach is comparable to the traditional single copy approach. In other words, the multiple copy approach is more useful in applications in which there is a wide disparity in the cost associated with different classes of message.

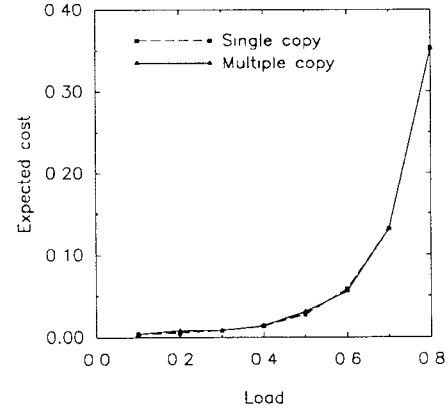
Table III shows the actual count of messages out of a total of 200,000 messages that failed to meet their deadline in the traditional single copy approach and the multiple copy approach when the network load was 0.4 and the other parameters as in Table I. It is clear from this table that there is a considerable reduction in the number of highly critical messages that missed their deadline without much change in the number of noncritical messages missing their deadlines.

5.2 Results for Hypercube Topology

The key difference between a hypercube and a C-wrapped hexagonal mesh topology is that in a hypercube there are multiple shortest routes between most pairs of nodes. A hypercube is also a regular, homogeneous graph. Each node in an n-dimensional hypercube, Q_n, has n (immediate) neighbors. A Q_n is comprised of 2^n nodes, labeled from 0 to $2^n - 1$, such that two nodes are neighbors if and only if the binary representation of their labels differ by one and only one bit.



(a) $K_1 = 625$, $K_2 = 25$, $K_3 = 1$ (b) $K_1 = 10$, $K_2 = 5$, $K_3 = 1$



(c) $K_1 = 1$, $K_2 = 1$, $K_3 = 1$

Fig. 7. Variation in the cost associated with different criticality classes

Table III. Count of Messages out of 200,000 that Failed to Meet their Deadlines

Approach	Criticality class		
	1	2	3
Single copy	179	435	2106
Multiple copy	23	435	2132

The hypercube topology has several nice properties out of which the three relevant properties are listed below:

(1) *Single copy routing*: There is a simple and elegant algorithm for routing messages in a hypercube [11]. Each node (including the source node)

EXCLUSIVE-ORs its own label with the label of the destination and sends the message along the direction corresponding to the rightmost 1 in the result of the EXCLUSIVE-OR.

(2) *Disjoint routes*: There are k disjoint shortest routes between two nodes that are k hops away.

(3) *Multiple copy routing*: The routing algorithm described above can be extended to send multiple copies of a message through disjoint shortest routes. The source node EXCLUSIVE-ORs its label with the label of the destination. The source node then sends one copy of the message along every direction corresponding to the 1's in the result of the EXCLUSIVE-OR. Before sending a copy, the source node marks (in the copy) the direction it chose for that copy. When a node receives a copy of a message from some other node, it EXCLUSIVE-ORs its own label with the label of the destination and sends the copy along the rightmost 1 (with wraparound) starting from the direction marked by the source node. The proof that this results in disjoint routes is given by Ramanathan and Shin [9] and Saad and Schultz [11].

As a result of these three properties, it is easy to implement the proposed scheme on a hypercube topology. Basically, the overhead of routing multiple copies through a hypercube network is minimal. Table IV shows the number of copies as determined by the proposed heuristic method for a few selected loads in a Q_7. Unlike in the C-wrapped hexagonal mesh, the multiple copies were sent only through the shortest routes. We could have easily permitted multiple copies to be sent through all n disjoint routes in a Q_n. This would have slightly increased the search time in the heuristic method but the reductions achieved in the expected cost would have been greater. The other parameters used for determining these copies are the same as the ones used for the C-wrapped hexagonal mesh in Table I except for the deadlines. Given the criticality and the number of hops a message has to traverse, the deadline of that message was taken from a uniform distribution centered around three times the mean delivery time of that message when the network load is 0.3. The deadlines varied from $\frac{2}{3}$rds to $\frac{4}{3}$rds of the mean deadline.

Figure 8 shows the percentage reduction in the expected cost for different input parameters. However, since we did not develop a simulator for a hypercube, these percentage reductions are as estimated by the heuristic method making use of assumptions M7 and M8. Curve A shows the percentage reduction in the expected cost when multiple copies of a message were sent using Table IV. Curve B has the same parameters as Curve A except that the costs when a message of class 1, 2, or 3 misses its deadline were assumed to be 100.0, 10.0, 1.0 instead of 625.0, 25.0 and 1.0. Curve C has the same parameters as Curve A except that each node was equally likely to communicate with every other node. Curve D shows the percentage reduction in the expected cost when the deadline requirements are less stringent than in Curve A. Except in the case of less stringent deadlines (Curve D), the percentage reductions in the expected cost are considerable even for a hypercube topology.

Table IV. Number of Copies as Determined by the Proposed Heuristic Method for a Q_7

Hops	Criticality Class 1	2	3
1	1	1	1
2	2	2	2
3	3	2	1
4	4	1	1
5	5	1	1
6	2	1	1
7	1	1	1

(a) $\rho_t = 0.1$

Hops	Criticality Class 1	2	3
1	1	1	1
2	2	2	1
3	3	2	1
4	4	1	1
5	2	1	1
6	1	1	1
7	1	1	1

(a) $\rho_t = 0.2$

Hops	Criticality Class 1	2	3
1	1	1	1
2	2	1	1
3	3	1	1
4	2	1	1
5	1	1	1
6	1	1	1
7	1	1	1

(a) $\rho_t = 0.5$

Hops	Criticality Class 1	2	3
1	1	1	1
2	2	1	1
3	2	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1

(a) $\rho_t = 0.8$

The percentage reductions in the expected cost for a C-wrapped hexagonal mesh seem to be better than the estimated reductions for a hypercube. However, since the percentage reductions for a hypercube are still in the range of 40–50% for low and moderate loads, the multiple copy approach offers substantial improvement in both these topologies.

6. CONCLUSION

A scheme was proposed to minimize the expected cost incurred as a result of messages missing their deadlines. The scheme made use of the multiple disjoint routes in a point-to-point interconnection network to send one or more copies of a message depending on the criticality and the number of hops the message has to traverse. Evaluation of the proposed approach was done using a simulator. Simulation results showed that reductions of more than 70% can be achieved at low to moderate loads. Since the proposed scheme

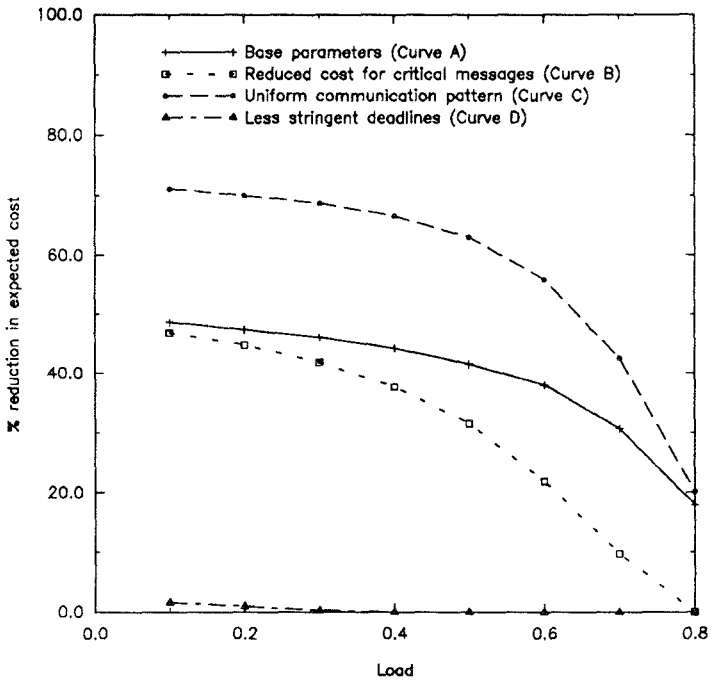


Fig. 8. % reduction in expected cost for different input parameters in a Q_7.

imposes minimal overhead while the system is in operation, it can be easily implemented along with the existing schemes for message passing under deadline constraints to achieve substantial improvements in message delivery times.

REFERENCES

1. Chen, M.-S., Shin, K. G., and Kandlur, D. D. Addressing, routing and broadcasting in hexagonal mesh multicomputers. *IEEE Trans. Comput. 39*, 1 (Jan. 1990), 10–18.
2. Davis, A., Hodgson, R., Schediwy, B., and Stevens, K. Mayfly system hardware Tech. Rep. HPL-SAL-89-23, Hewlett-Packard Company, Apr. 1989.
3. Dolev, D. The Byzantine generals strike again. *J. Algorithms 3*, 1 (Mar. 1982), 14–30.
4. Jackson, J. R. Networks of waiting lines. *Oper. Res. 5*, 4 (Aug. 1957), 518–521.
5. Kleinrock, L. *Communication Nets: Stochastic Message Flow and Delay*. Dover Publications, New York, 1972.
6. Kleinrock, L., and Naylor, W. E. On measures behavior of the ARPA network. In *Spring Joint Computer Conference AFIPS Conference Proceedings* (Chicago, Ill. May 1974), 767–780.
7. Kurose, J. F., Schwartz, M., and Yemini, Y. Controlling window protocols for time-constrained communication in multiple access networks. *IEEE Trans. Commun. 36*, 1 (Jan. 1988), 41–49.
8. Orda, A., and Rom, R. Routing with packet duplication and elimination in computer networks. *IEEE Trans. Commun. 36*, 7 (July 1988), 860–866.
9 Ramanathan, P., and Shin, K. G. Reliable broadcast in hypercube multicomputers. *IEEE Trans. Comput. 37*, 12 (Dec. 1988), 1654–1657.

10. RUPNICK, G. M , AND RAMANATHAN, P Deadline constrained message scheduling in distributed systems with point-to-point interconnection topology. Masters report, Dept. of Electrical and Computer Engineering, Univ. of Wisconsin-Madison, Madison, Wisc , May 1991.
11. SAAD, Y., AND SCHULTZ, M. H. Topological properties of hypercubes *IEEE Trans. Comput. 37*, 7 (July 1988), 867-872.
12. STEVENS, K. S. The communication framework for a distributed ensemble architecture. AI Tech. Rep. 47, Schlumberger Research Laboratory, Feb. 1986.
13. STROSNIDER, J. K., MARCHOK, T., AND LEHOCZKY, J. P. Advanced real-time scheduling using the IEEE 802.5 token ring In *Proceedings of Real-Time Systems Symposium* (Huntsville, Ala., Dec. 1988), pp. 42-52.
14. ZHAO, W., AND RAMAMRITHAM, K. Virtual time CSMA protocols for hard real-time communications *IEEE Trans. Softw. Eng., SE-13*, 8 (Aug. 1987), 938-952.
15. ZHAO, W , STANKOVIC, J. A., AND RAMAMRITHAM, K. A window protocol for transmission of time constrained messages *IEEE Trans. Comput. 39*, 9 (Sept 1990), 1186-1203

Received November 1990; revised June 1991; accepted August 1991