Performance Evaluation 13 (1991) 159–179
North-Holland

Performance analysis of a class of hierarchical hypercube multicomputer networks

Sivarama P. Dandamudi
School of Computer Science, Carleton University, Ottawa, Ontario K1S 5B6, Canada

Received May 1990
Revised February 1991

Abstract

Dandamudi, S.P., Performance analysis of a class of hierarchical hypercube multicomputer networks, Performance Evaluation 13 (1991) 159–179.

Hierarchical interconnection networks (HINs) have been proposed to interconnect large numbers of processors in a multicomputer system. It has been shown that HINs provide better cost–benefit ratios than the corresponding nonhierarchical interconnection networks. This article discusses performance of two schemes that improve fault-tolerance of binary hypercube-based HINs. Both these schemes use hardware redundancy. In one, a standby-spare node is provided to reduce the impact of key node failures on the network reliability; in the other, a part of the network is duplicated. Both these schemes improve the network reliability substantially. The analysis presented here shows that, from a performance point of view, neither of the two schemes dominates the other for all parameter values and system characteristics. If the system supports applications that have varying degrees of communication locality and/or different computation–communication ratios, the duplication scheme is to be recommended. On the other hand, when applications exhibit high degrees of communication locality and high computation–communication ratios, the standby-spare node scheme provides better performance.

The impact of three routing algorithms is also considered. We derive bounds on message delay and saturation message generation rate and compare the performance of these routing algorithms in achieving these bounds. It is shown by means of analytical and simulation models that performance of the replication scheme is less sensitive to the routing algorithm used.

Keywords: fault-tolerance, hypercubes, hierarchical networks, interconnection networks, multicomputers, parallel systems, performance, routing.

1. Introduction

Multicomputer systems communicate by explicitly passing messages. In building large multicomputer systems the underlying interconnection network plays an important role in determining system performance and in allowing system expansion. Several interconnection networks have been proposed in the literature (for details, see [1,14,36]). The hypercube network has been popularly used as an interconnect in multicomputer systems. For the sake of brevity, we will not discuss the advantages and disadvantages of this structure. Rather, the interested reader is referred to the literature, including [1,4,16,36]. The hypercube has been used as the interconnection network in several experimental and commercial multicomputer systems such as the Cosmic Cube [34], the Intel iPSC [31] and the NCUBE/ten system [19].

As the number of nodes increases in the network, the number of links needed for the hypercube network becomes prohibitively large. This, among other factors, imposes a limit on system expansion. Therefore, future systems may be designed to minimize the number of links because most of the system space may be filled with wires [20]. For example, the NCUBE/ten system [19], which organizes 1024 nodes as a binary hypercube, requires as many as 512 wires for inter-PCB (printed circuit board) connections. (The NCUBE/ten actually uses 640 wires to allow connections to I/O devices.) This is one of the motivations for proposing hierarchical interconnection networks (for details, see [10]).

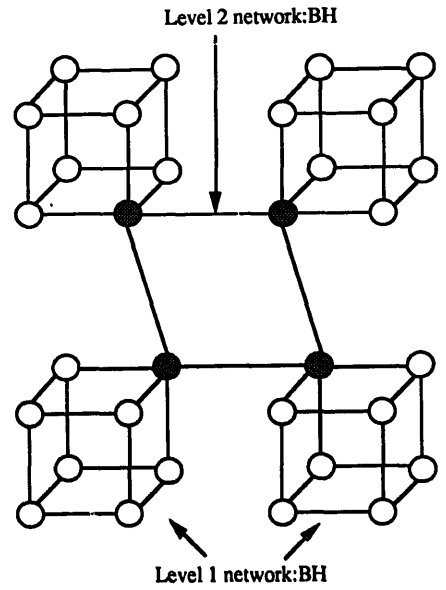

Fig. 1. A BH/BH hierarchical interconnection network.

Hierarchical interconnection networks provide a cost-effective way to interconnect large numbers of processors in a multicomputer system. Several hierarchical interconnection networks have been proposed and studied [6,15,16,21,27,37]. This article considers the hierarchical interconnection networks (HINs) proposed in [10], the structure of which can be informally described as follows. Let N be the total number of nodes in the network. These N nodes are divided into K_1 clusters of $n_1 = N/K_1$ nodes each. Each cluster of n_1 nodes is linked together by a level 1 interconnection network. Then one node from each cluster is selected to act as an interface node and these K_1 interface nodes are again divided into K_2 clusters of $n_2 = K_1/K_2$ nodes each. A level 2 interconnection network is used to link each of these K_2 clusters of n_2 level 1 interface nodes. Then, one node from each level 2 cluster is selected as the level 2 interface node to be linked together by a level 3 network, etc. Fig. 1 shows an example HIN BH/BH with two levels, in which both the levels use the binary hypercube (BH) network. The

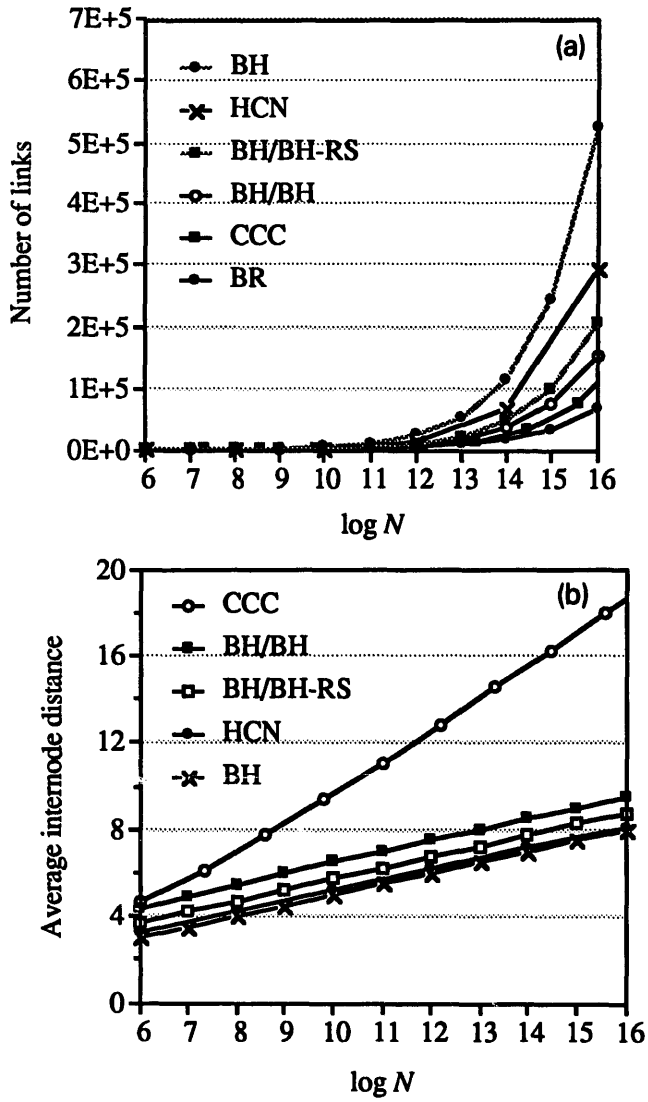

Fig. 2. Link cost and average internode distance of various interconnection networks.

Sivarama P. Dandamudi was born in Andhra Pradesh, India. He received the B.E. degree from the University of Mysore, India, the M. Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, and the M.Sc. and Ph.D. degrees in computer science from the University of Saskatchewan, Saskatoon, Canada in 1984 and 1988, respectively.

Currently, he is an Assistant Professor in the School of Computer Science at Carleton University, Ottawa, Canada. His research interests include parallel systems, database systems, and performance evaluation.

BH/BH HIN can be characterized by parameters $d = \log_2 C$ and $D = \log_2 N$, where C represents the cluster size and N represents the total number of nodes in the network. In the example shown in Fig. 1, $d = 3$ and $D = 5$. Since binary hypercubes are used in both commercial and experimental multicomputers, we consider binary hypercube-based HINs in the remainder of this article.

It has been shown that HINs provide better cost–benefit ratios than the corresponding non-hierarchical interconnection networks. HINs reduce the link cost substantially (see Fig. 2) at the expense of moderately increasing the average internode distance and the average message delay [10]. Thus HINs allow more nodes to be included in the system when constrained by the link cost. For example, in the NCUBE/ten system discussed earlier, the use of the BH/BH HIN (with $d = 3$ and $D = 10$) would greatly reduce the number of wires needed for inter-PCB connections (from 512 wires to 64 wires). This is an important advantage of HINs facilitating the design of large parallel systems.

There are other methods to reduce the link cost. Here we briefly compare the BH/BH HIN to the following two networks [1]: the cube-connected-cycles (CCC) network [30] and the HCN [15,16].

Figure 2a shows the link cost associated with the BH, BH/BH, CCC, and the HCN networks. To represent the lower bound on link cost, the link cost of the bidirectional ring (BR) is also included. The average internode distance under uniform communication, in which each pair of nodes exchange messages at an identical rate, is shown in Fig. 2b (The link cost of a two-level cubelet-based Hypernet [21] is more than that of the hypercube network and its average internode distance is similar to that of the BH/BH network. Therefore, this network is not shown in Fig. 2).

[1] A comparison with other networks such as the chordal ring [2] and the hypertree [17] shows that the link cost of these networks is similar to that of the CCC network. However, the average internode distance of the chordal ring network is much larger and increases rapidly with network size. The hypertree network provides average internode distance that is lower than that of the CCC network (but much higher than that of the BH/BH network). In order not to clutter the graph, these networks are not included in Fig. 2 (for more details, see [8]). Reliability of these networks is briefly discussed in Section 2.1.

Among the four networks of interest – the BH, BH/BH, CCC, and the HCN – the CCC network uses the minimum number of links but its average internode distance is substantially higher than that of the BH network. The BH network, on the other hand, provides the minimum average internode distance but requires a substantially larger number of links to achieve this.

Between the two hierarchical structures, the BH/BH network requires the minimum number of links (and its average internode distance is similar to that of the Hypernet). The HCN provides a better average internode distance that is close to that of the BH network, but this network requires substantially more links to achieve this improvement. However, the HCN organization has the advantage of requiring uniform degree nodes but the degree increases as a function of system size. Of course, the CCC network has the advantage of requiring a constant degree for all nodes independent of the system size. Also, note that all hypercube-based hierarchical networks inherit the inability of the hypercube to grow incrementally. The realizable system sizes are a power of two with the hypercube and BH/BH networks. The gaps between realizable system sizes are even larger with HCN (system sizes should be equal to an even power of two). In addition, in the HCN network, cluster size and the number of clusters are not independent of each other (and in an incomplete HCN, the number of clusters must be less than the number of nodes in cluster).

Disadvantages of BH/BH HINs include the potentially high traffic rates on intercluster links (i.e., the level 2 network links), and thus the potential degradation in performance. (This disadvantage is applicable to other hierarchical networks as well.) However, the performance enhancements suggested in [10] appear to economically alleviate the problem of congestion on intercluster links.

A weakness of the structure, shown in Fig. 1, is that when an interface node fails, the network is partitioned and the set of nodes belonging to the cluster of the failed interface node is isolated. There are several ways in which fault-tolerance of HINs can be improved. Here we consider two schemes. In one, we provide standby-spare interface node to reduce the impact of interface node failures on the network reliability; in the other, we augment the network by duplicating the level 2

network. This article discusses the impact of these two schemes on reliability and performance.

It has been shown in [11] that two is a pragmatic choice for the number of levels in a binary hypercube-based HIN. Furthermore, in such a network, the optimum cluster size (that minimizes the product of the average internode distance and the link cost) is shown to be 8 for large network sizes and for different communication locality characterizations. Therefore, we consider only the two-level binary hypercube-based HINs in which each cluster contains 8 nodes.

The remainder of the article is organized as follows. Section 2 describes the two schemes to improve fault-tolerance of HINs. Section 3 presents the performance analysis and the results are discussed in Section 4. The effect of routing is considered in Section 5 and conclusions are given in Section 6.

2. Fault-tolerant HINs

The physical replication of hardware is perhaps the most common form of redundancy used in digital systems today [23]. The costs of replicating hardware within a system are decreasing simply because hardware costs are decreasing. This section describes two schemes to improve fault-tolerance of HINs. Both these schemes use hardware redundancy.

i) Replication scheme: in this scheme, each cluster uses two nodes as the interface nodes. The level 2 network is duplicated for each interface node. From a performance point of view, it is advantageous to keep the two interface nodes belonging to a level 1 cluster as far apart as possible. For the BH/BH HIN under consideration, the distance between the two interface nodes is d. An example HIN incorporating this fault-tolerance scheme is shown in Fig. 3. This network is referred to as the BH/BH-RS network. Note from Fig. 2a that the link cost of this network is still 1/3 less than the corresponding cost for the HCN network. The average internode distance, shown in Fig. 2b, is marginally higher than that of the BH and HCN networks.

ii) Standby-spare interface node scheme: since the interface nodes are the key components compromising the network reliability, we introduce hardware redundancy by providing standby-spare interface nodes. In this scheme, one interface node is operational and the other serves as a spare. If a fault is detected and located, the faulty node is removed from the operation and replaced with the spare (for details on fault detection and location, see [23]). The reconfiguration operation can be viewed conceptually as a switch whose output is selected from one of the nodes providing the input to the switch [23]. This network is referred to as the BH/BH-SI network. The link cost and the average internode distance of this network are the same as those for the BH/BH network.

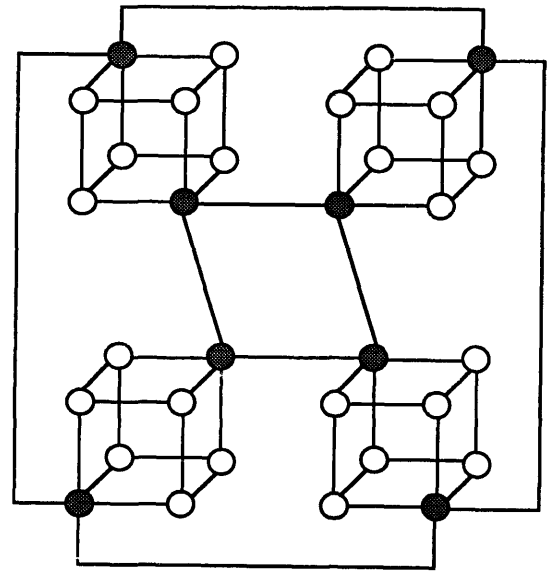

Fig. 3. A BH/BH-RS hierarchical interconnection network.

Note that both these schemes essentially use two interface nodes per cluster. While using more than two interface nodes further improves the network reliability there is an associated cost involved. For example, in the BH/BH-RS network, increasing the number of interface nodes per cluster results in an increase in link cost that may negate the advantage of HINs. (The BH network can be considered as the BH/BH-RS network with 2^d interface nodes.) We show in Section 2.1 that with two interface nodes these schemes would provide substantial improvement in the network reliability (their reliability is comparable to or better than that of the CCC, chordal ring [2], and hypertree [17] networks). Furthermore, a combination of these two schemes (see Section 2.1) provides reliability as good as that provided by the complete connection network. For these reasons, we feel that these two basic schemes provide cost-effective fault-tolerant solutions that are useful in the context of HINs.

2.1. Reliability evaluation

Reliability can be evaluated using either the terminal reliability model or the task-based reliability model [22,12]. In the terminal reliability model, a system is considered to be working as long as any node is connected to any other node in the system. In the other model, reliability $R(t)$ is defined as the probability of at least I communication nodes are operational and connected at time t. In parallel processing systems a job is divided into several tasks, each of which runs on a separate processor in a cooperative fashion. Therefore, task-based reliability is more appropriate for reliability evaluation of parallel processing systems. This reliability model has been used to evaluate the reliability of multicomputer interconnection networks [5,12,24].

We use simulation for reliability evaluation of these networks. The simulation model that we have used is similar to the one reported in [5]. The nodes and links are assumed to have identical and exponential distribution of failure time. Let λ_n and λ_l denote node and link failure rates, respectively. The given interconnection network is represented by its adjacency matrix. The network connectivity is shown by using the reachability matrix. This matrix specifies whether or not there

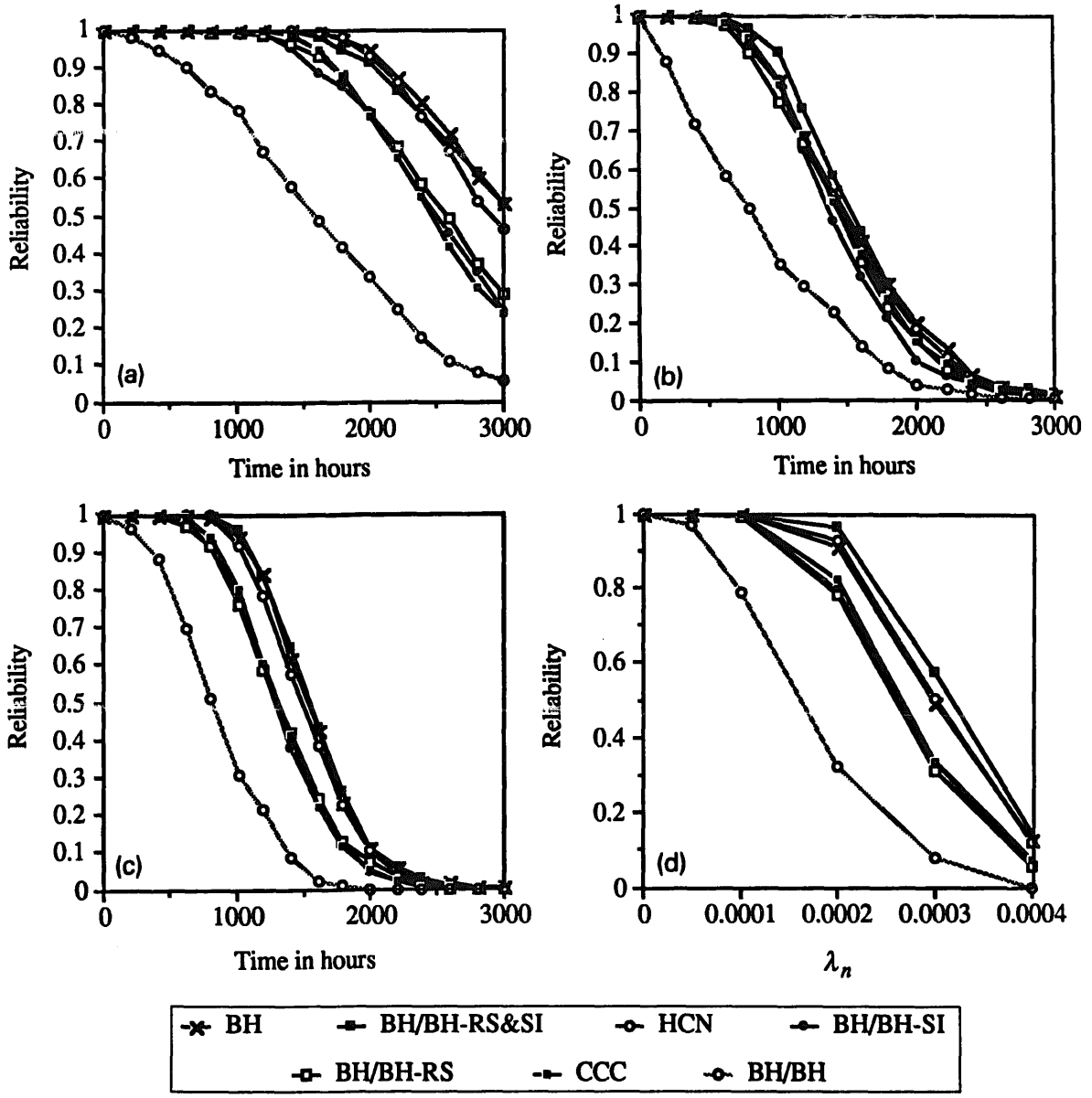

Fig. 4. Reliabilities of various interconnection networks for a network of size 64 (hierarchical networks use a cluster size of 8).

exists a path between two nodes that are not necessarily adjacent. The adjacency matrix is modified depending on the type of fault (i.e., node failure or link failure). The reachability matrix can be obtained from the adjacency matrix using any standard algorithm [13,35].

The results (for a network of size $N = 64$ nodes) are presented in Fig. 4. Most of the results presented here assume a node failure rate of $\lambda_n = 100$ per 10^6 hours and a link failure rate of $\lambda_l = 20$ per 10^6 hours. These values are shown to be representative for a multicomputer system (see [5] for details). The impact of varying these failure rates is also studied. (For the sake of brevity, all results are not presented here. Details on the impact of link failure rates, network sizes, cluster sizes etc. can be found in [9].)

Figure 4a shows reliabilities of various networks with $N = 64$ and $I = 48$. The HINs use a cluster size of 8. It can be seen from this figure that the reliability of the HCN is similar to that of the BH network (which provides reliability similar to that of the complete connection network [5,9]). The reliability of the BH/BH network is substantially worse than that of the BH network. This is because of the special role played by the interface nodes. The CCC network, which uses smaller number of links than does the BH/BH network, provides better reliability [2]. Note, however, the BH/BH network uses substantially fewer (only half as many) number of links compared to the HCN network.

The impact of the two fault-tolerant schemes is also shown in Fig. 4a. Both these schemes provide reliabilities similar to that of the CCC network. Both these schemes provide substantial improvement in reliability compared to the BH/BH network.

A weakness of both the BH/BH-RS and the BH/BH-SI networks is that their reliability is uniformly poorer than the BH network. This drawback can be remedied by combining the two fault-tolerant schemes. This is achieved by adding standby-spare interface nodes to the BH/BH-RS network (we refer to this network as the BH/BH-RS&SI network). This, of course, increases the network cost but this increase (over that of the BH/BH-RS network) is a one-time design cost and can easily be amortized without significantly affecting the overall system cost. It should be noted that, in designing large multiprocessor systems, it is not the VLSI logic cost that is of major concern in system expansion but the link cost. For example, as discussed in Section 1, the NCUBE/ten, which uses 640 wires to provide inter-PCB connections, the link cost creates major problems in system expansion. Thus, from overall system design point of view, the penalty associated with the BH/BH-RS&SI network may not be significant. This, however, improves the network reliability substantially as shown in Fig. 4a. The BH/BH-RS&SI network reliability is as good as that of the BH network. Thus, the increase in network cost is justified from the reliability point of view. Note, however, the performance of this network remains the same as that of the BH/BH-RS network.

The results presented in Fig. 4a assumed an I value that is equal to 75% of the total nodes N. The effect of changing the I value is shown in Fig. 4b. This figure gives reliabilities of various networks when a task requires $I = 56$ nodes for its execution. Interestingly, the reliabilities of both the BH/BH-RS and the BH/BH-Si networks are very close to the reliabilities of the BH network. The reason for this is that, as the I value approaches N, the node failures basically determine the network reliability (for the failure rates considered here); the link failures play a relatively minor role.

Figure 4c shows reliabilities of various networks when the node failure rate is increased from 0.0001 to 0.0002. The effect of doubling the node failure rate can be seen by comparing Figs. 4a and 4c. Even with the increased node failure rate, the conclusions drawn from Fig. 4a are still valid in that the BH/BH-RS, BH/BH-SI and the CCC networks provide similar reliabilities, and the remaining networks (excluding the BH/BH network) provide similar reliabilities. This is further demonstrated in Fig. 4d, which gives network reliabilities at $t = 1000$ for various node failure rates. Although not presented here, the link failure rate has relatively minor effect on network reliabilities compared to that of the node failure rate.

The results presented in this section indicate that the reliability of the BH/BH network can be substantially improved by using the two fault-

[2] Although not shown here, the CCC network provides better reliability than the chordal ring network, which in turn provides better reliability than the hypertree network [8].

tolerant schemes discussed in Section 2. The BH/BH-RS network, however, increases the link cost when compared to that of the BH/BH-SI network. From a reliability point of view, this increase in link cost associated with the BH/BH-RS network does not result in commensurate improvement in network reliability. However, we will show in the following sections that adding these additional links improves performance (by reducing the average message delay substantially) of the network (in most cases) compensating for the increase in link cost. Therefore, from a reliability point of view, the BH/BH-SI network may be preferred but when both reliability and performance are considered the BH/BH-RS network may be the network of the choice. The BH/BH-RS&SI network improves reliability further and provides reliabilities similar to the reliabilities of the HCN and BH networks while requiring 1/3 fewer links than the HCN network. The link cost and the performance of the BH/BH-RS&SI network is the same as that of the BH/BH-RS network. Performance issues of these networks are considered in the following sections.

3. Performance analysis

This section discusses the impact of these two fault-tolerance schemes on performance. Their performance is compared against the corresponding nonhierarchical BH network. The link cost L is used to measure the network cost. The number of links is used to represent the link cost. The performance of the network is measured by the average message delay R. Similar performance measures have been used in the literature [10,21,32,33]. Average message delay analysis of the BH/BH-RS is presented in Section 3.2. The delay analysis for the BH/BH-SI and BH networks can be found in [10].

3.1. The workload and system model

Let N denote the total number of nodes in the network, n the number of nodes in a cluster, and K the number of clusters (i.e., K nodes participate in the level 2 network), where $N = Kn$. The locality in communication is characterized by a single parameter α. Let α be the probability of both source and destination nodes of a message being in the same cluster. Therefore, $(1 - \alpha)$ represents the probability of inter-cluster communication. The larger the value of α (for a fixed cluster size), the stronger the locality in communication. Locality in communication has been characterized similarly in the literature [10,16,21,37]. It is further assumed that (similar assumptions are made in [21,32,37]):

- intra-cluster communication is uniformly random (i.e., a source node sends a message to each node within its cluster with equal probability);
- inter-cluster communication is uniformly random (i.e., a source node sends an inter-cluster message to each other cluster with equal probability, and to each node within the destination cluster with equal probability).

A simple model is used for the queueing analysis. Each network link is assumed to be full-duplex. In our analysis, we conceptually replace each full-duplex link by two simplex links, each of which is modelled as a queueing center. Expressions are derived for the average message delay R. The following assumptions are made about the network and its workload:

i) each node generates messages at rate λ and the inter-message times are exponentially distributed;
ii) the node message generation processes are independent of each other;
iii) message service times are exponentially distributed; each link processes these messages at rate μ;
iv) each node has unbounded buffering capacity;
v) message-switching is assumed for message transmission;
vi) all messages are routed over the shortest path between the source and destination nodes. If more than one shortest path exists between a pair of nodes, it is assumed that random routing is used (i.e., each shortest path is selected with equal probability). Thus the routing scheme uses no information regarding the current state of the network. Section 5 studies the impact of various routing algorithms on network performance.

To facilitate the analysis, Kleinrock's independence assumption [26] is used. This assumption states that each time a message arrives at a link, a new service time is generated for this message from the exponential service time distribution (i.e.,

there is no "memory" of message lengths from hop to hop). This assumption, though not at all realistic, is often used in the delay analysis of communication networks [18,26]. The results of the simulation experiments in Section 3.3 indicate that the error introduced by this assumption is negligible. These assumptions allow each (simplex) link to be modelled as an M/M/1 queueing center [25].

The "Markovian" and other similar assumptions that we use in this analysis are not at all "realistic", in that they do not reflect the behaviour of some given, particular application. Note, however, that it is not the goal here to accurately predict the performance of a particular system with some particular workload; rather, all that is desired is an evaluation of the relative performance of different networks. Similar assumptions, including the independence assumption discussed above, are made in [4]. A large body of literature and experience with models of this type supports their usefulness [28].

3.2. Delay analysis

This section presents the delay analysis of the BH/BH-RS network. This network has two interface nodes per cluster. When an inter-cluster message is generated, the routing algorithm passes the message to the closest interface node in the source cluster. If more than one interface node is at the same distance, one of them is randomly selected. Thus, the analysis assumes that both the interface nodes are equally utilized.

The links are divided into two groups: cluster (CL) links and noncluster (NCL) links. Cluster links are those that connect only the nodes within a cluster and noncluster links are those that connect two nodes that are in two different clusters. Let μ_{CL} and μ_{NCL} be the message processing rates of cluster and noncluster links, respectively. Further, let $\lambda_{link,CL}$ and $\lambda_{link,NCL}$ be the effective message arrival rates at cluster and noncluster links, respectively. The effective arrival rate for each group of links is derived.

The message load of inter-cluster messages is uniformly distributed over the non-cluster links. In deriving $\lambda_{link,NCL}$ it should be noted that each inter-cluster message uses $[(D-d)\,2^{D-d-1}/(2^{D-d}-1)]$ non-cluster links, on average, and each cluster generates these messages at rate $(1-\alpha)\lambda 2^d$. Therefore

$$\lambda_{link,NCL} = \frac{2^{D-d}2^d(1-\alpha)\lambda\left[\dfrac{(D-d)2^{D-d-1}}{2^{D-d}-1}\right]}{4(D-d)2^{D-d-1}} = \frac{2^{D-2}(1-\alpha)\lambda}{2^{D-d}-1}.$$

The message load of intra-cluster messages is uniformly distributed over the cluster links. However, the message load due to inter-cluster messages is not distributed equally among the links in a cluster. The cluster links closer to the interface nodes receive more inter-cluster messages than those that are farther away. Thus $\lambda_{link,CL}$ is dependent on how close a cluster link is to the interface nodes. To indicate this distance, we label the cluster links by their distance to the interface node under consideration. A j-level link is defined as a cluster link that connects two nodes that are at distance j and $j-1$ form the interface node. Then the arrival rate of messages at a j-level link in the source cluster $\lambda_{link,CLs}$ is given by

$$\lambda_{link,CLs}(j) = \lambda'_s(j) + \lambda''_s(j),$$

where $\lambda'_s(j)$ = arrival rate (at a j-level link in the source cluster) due to intra-cluster messages, and $\lambda''_s(j)$ = arrival rate (at a j-level link in the source cluster) due to inter-cluster messages. Since intra-cluster message load is uniformly distributed over the links within a cluster, $\lambda'_s(j)$ is given by

$$\lambda'_s(j) = \frac{\alpha\lambda 2^{d-1}}{2^d-1} \quad \text{for all } j.$$

However, $\lambda''_s(j)$ is not the same for all j. It is given by

$$\lambda''_s(j) = 2^{d-1}\lambda(1-\alpha)p_1(j) + 2^{d-1}\lambda(1-\alpha) \times p_2(d-j+1),$$

where $p_1(j)$ is the probability that an inter-cluster message generated in the source cluster gets routed through a given j-level link on its way out of the cluster (through one interface node, say, i_0) and $p_2(j)$ is the probability that an inter-cluster message (destined to a node within the source cluster under consideration) passing through the other interface node i_1 that gets routed through a given j-level link on its way to the destination node. We now derive an expression for $p_1(j)$. First consider the case where d is odd. A set of links J, all of whose element links are j-level links (just the

"out" direction), act as intermediate links for all inter-cluster messages generated by any source node within the cluster that is at distance at least j but less than $(d+1)/2$ from the interface node. Therefore, the number of source nodes for which an element of J acts as an intermediate link is $2^{d-1} - \sum_{k=0}^{j-1}\binom{d}{k}$. Since there are $(d-j+1)\binom{d}{j-1}$ such links (i.e., the number of elements in J)

$$p_1(j) = \frac{2^{d-1} - \sum_{k=0}^{j-1}\binom{d}{k}}{(d-j+1)\binom{d}{j-1}2^{d-1}} \quad \text{if } d \text{ is odd.}$$

Similarly, when d is even, $p_1(j)$ is given by

$$p_1(j) = \frac{\left(\sum_{i=0}^{\frac{d}{2}-1}\binom{d}{i} + \frac{\binom{d}{d/2}}{2}\right) - \sum_{k=0}^{j-1}\binom{d}{k}}{(d-j+1)\binom{d}{j-1}\left(\sum_{i=0}^{\frac{d}{2}-1}\binom{d}{i} + \frac{\binom{d}{d/2}}{2}\right)}$$

if d is even.

Analogous to the derivation of $p_1(j)$, $p_2(j)$ can be derived as

$$p_2(j) = \frac{\sum_{k=j}^{d}\binom{d}{k}}{(d-j+1)\binom{d}{j-1}2^d}.$$

Therefore, $\lambda_{\text{link,CLs}}$ is given by

$$\lambda_{\text{link,CLs}}(j) = \frac{\alpha\lambda 2^{d-1}}{2^d-1} + 2^{d-1}\lambda(1-\alpha) \times [p_1(j) + p_2(d-j+1)].$$

The final component of the delay is accounted for by the delay encountered by an inter-cluster message in the destination cluster (i.e., the cluster containing the destination node of the inter-cluster message). The effective message arrival rate in the destination cluster $\lambda_{\text{link,CLd}}$ can be derived as

$$\lambda_{\text{link,CLd}}(j) = \lambda'_d(j) + \lambda''_d(j),$$

where $\lambda'_d(j)$ = arrival rate (at a j-level link in the destination cluster) due to intra-cluster messages;

$$\lambda'_d(j) = \frac{\alpha\lambda 2^{d-1}}{2^d-1} \quad \text{for all } j$$

and $\lambda''_d(j)$ = arrival rate (at a j-level link in the destination cluster) due to inter-cluster messages.

$$\lambda''_d(j) = \begin{cases} 2^{d-1}\lambda(1-\alpha)p_2(j) & \text{if } j \leqslant \left\lceil \frac{d}{2} \right\rceil \\ 2^{d-1}\lambda(1-\alpha)[p_2(j) + p_1(d-j+1)] & \\ \quad \text{if } j > \left\lceil \frac{d}{2} \right\rceil & \end{cases}$$

Then the average message delay is given by

$$R^{\text{RS}}_{\text{BH/BH}} = \alpha\left\lceil \frac{d2^{d-1}}{2^d-1} \right\rceil \Delta_{\text{Avg}} + (1-\alpha)\left\{\Delta_{\text{CLs-Avg}} + \Delta_{\text{CLd-Avg}} + \left(\frac{(D-d)2^{D-d-1}}{2^{D-d}-1}\right)\Delta_{\text{NCL-Avg}}\right\},$$

where $\Delta_{\text{NCL-Avg}}$ = average delay, at a link, of an inter-cluster message in the level 2 network:

$$\Delta_{\text{NCL-Avg}} = \frac{1}{\mu_{\text{NCL}} - \lambda_{\text{link,NCL}}}.$$

$\Delta_{\text{CLs-Avg}}$ = average delay encountered by an inter-cluster message in the source cluster:

$$\Delta_{\text{CLs-Avg}} = \sum_{j=1}^{\left\lfloor \frac{d}{2} \right\rfloor}\left\{(d-j+1)\binom{d}{j-1}p_1(j)\Delta_s(j)\right\}.$$

$\Delta_{\text{CLd-Avg}}$ = average delay encountered by an inter-cluster message in the destination cluster:

$$\Delta_{\text{CLd-Avg}} = \sum_{j=1}^{d}\left\{(d-j+1)\binom{d}{j-1}p_2(j)\Delta_d(j)\right\},$$

and Δ_{Avg} = average delay, at a link, of an intra-cluster message:

$$\Delta_{Avg} = \begin{cases} \dfrac{1}{d2^{d-1}}\left[\displaystyle\sum_{j=1}^{\frac{d}{2}}\left\{(d-j+1)\binom{d}{j-1}\Delta_d(j)\right\} + \sum_{j=1}^{\frac{d}{2}}\left\{(d-j+1)\binom{d}{j-1}\Delta_s(j)\right\}\right] \\ \quad \text{if } d \text{ is even} \\ \dfrac{1}{d2^{d-1}}\left[\displaystyle\sum_{j=1}^{\frac{d+1}{2}}\left\{(d-j+1)\binom{d}{j-1}\Delta_d(j)\right\} + \sum_{j=1}^{\frac{d-1}{2}}\left\{(d-j+1)\binom{d}{j-1}\Delta_s(j)\right\}\right] \\ \quad \text{if } d \text{ is odd.} \end{cases}$$

Cluster link delays in source and destination clusters $\Delta_s(j)$ and $\Delta_d(j)$ are given by

$$\Delta_s(j) = \frac{1}{\mu_{CL} - \lambda_{link,CLs}(j)} \text{ and}$$

$$\Delta_d(j) = \frac{1}{\mu_{CL} - \lambda_{link,CLd}(j)}.$$

3.3. Validation of the analysis

To facilitate the delay analysis in Section 3.2, it was assumed that whenever a message arrives at a link on its path to its destination, a new service time is generated from the exponential service time distribution. In a real system, of course, the messages would retain their service times, which reflect the lengths of the messages, from the times they are generated to the times they arrive at their destinations. A simulation model was constructed that reflected this reality, and, therefore, allowed validation of the independence assumption used in the analysis (and to verify the analysis itself). In all other respects, the simulation model matched the analytic model.

The simulation results for the BH/BH-RS network are presented in this section. We have conducted simulation experiments with different parameters. Selected results are presented in Fig. 5 that illustrate well the results of these experiments. Figure 5a gives the average message delay as a function of communication locality α and Fig. 5b gives the average message delay as a function of the network size N. The value of α is varied from 0.5 to 1.0 in steps of 0.1 and N is varied from 32 to 256 nodes. It can be seen from these plots that the results obtained from the analytical formulas match closely those obtained by simulation, thus justifying the use of Kleinrock's independence assumption in this context.

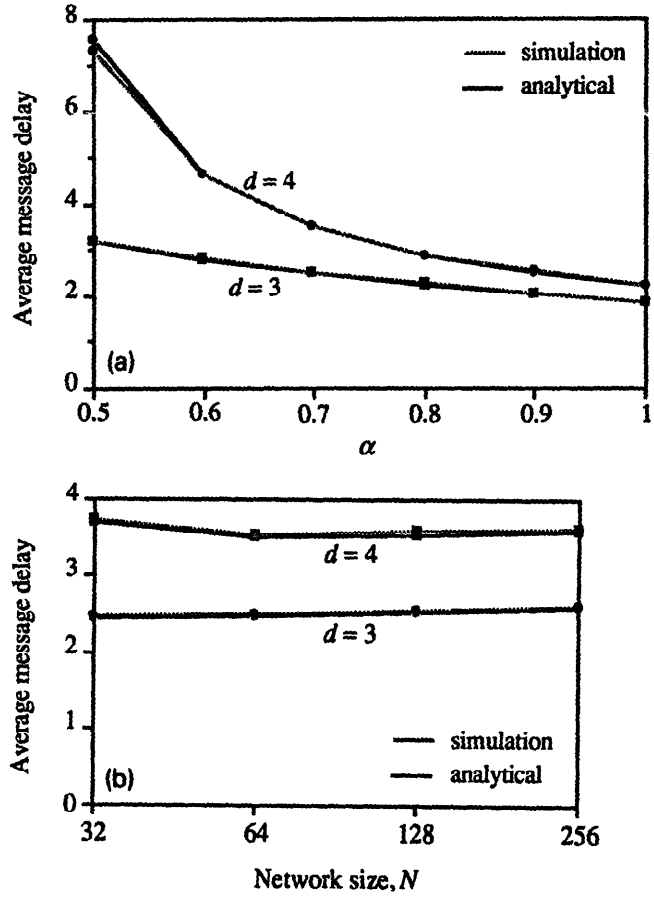

Fig. 5. Comparison of analytical and simulation results for the BH/BH-RS network: (a) $D = 6$, $\lambda = 1$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$; (b) $\lambda = 1$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$, $\alpha = 0.7$.

4. Performance comparison

This section compares the performance of the two fault-tolerant BH/BH networks: the BH/BH-SI network and the BH/BH-RS network. In addition, as stated in Section 3, the standard binary hypercube (BH) network is used as a reference network against which the performance of these two networks is compared. From the results of this comparison we wish to obtain insight into the tradeoffs involved among reliability, cost and

performance. This insight should be useful in designing HINs.

We consider two different scenarios in doing this performance comparison. First, we assume that both the cluster (CL) and non-cluster (NCL) links have the same message processing rates (i.e., $\mu_{CL} = \mu_{NCL}$). For fair performance comparison, these rates are assumed to be equal to that of the reference BH network. Therefore, BH/BH-RS network, which also uses $\mu_{CL} = \mu_{NCL}$ condition, increases the network cost by doubling the number of links in the level 2 network (compared to the BH/BH-SI network). To account for this difference in the cost factor, we use a cost–benefit ratio, called the *LR ratio*, which is defined in [10] as

$$\text{LR ratio} = \frac{L_{HIN} \times R_{HIN}}{L_{REF} \times R_{REF}}$$

where L_{HIN} and L_{REF} are the link costs of the HIN and the reference network, respectively. Similarly, R_{HIN} and R_{REF} refer to the average message delays of the HIN and the reference network, respectively. As stated earlier, we use the BH network as the reference network in our comparison. Smaller LR ratio values (smaller than 1) are preferred. A value of 1 for the LR ratio indicates that the HIN under consideration has the same cost–benefit ratio (as measured by the LR ratio).

In the other scenario, discussed in Section 4.4, we assume that $\mu_{NCL} = 2\mu_{CL}$ for the BH/BH-SI network. This means that each logical link in the level 2 network is mapped to two physical links. It has been shown in [10] that cost-benefit ratio tends to improve with this mapping (when compared to the situation with $\mu_{CL} = \mu_{NCL}$). However, in BH/BH-RS network, we use $\mu_{CL} = \mu_{NCL}$. Thus, in this case, the BH/BH-RS network can be thought of as redistributing the total number of physical links in the level 2 network of the BH/BH-SI network. Thus both the BH/BH-SI and the BH/BH-RS networks use the same number of physical links. Hence the link cost remains the same for both these fault-tolerant HINs. Thus any increase in performance should reflect in a similar improvement in cost–benefit ratio.

4.1. Impact of communication locality

Figure 6a depicts the average message delays involved in the BH, BH/BH-SI, and BH/BH-RS networks for the following parameters: $\lambda = 1$, $\mu_{CL} = \mu_{NCL} = 3$, $d = 3$, and $D = 13$ (i.e., network size $N = 8192$). The value of α is increased from 0.5 to 1 in steps of 0.1. Since HINs are cost-effective when there is communication locality [10], we do not consider α values less than 0.5. (It has been shown in [10] that if a mesh-structured computation, for example, is mapped onto an HIN that uses a cluster size 4 yields a value of 0.5 for α. Larger cluster sizes result in increased α values.) Figure 6b shows the corresponding LR ratio values for the two fault-tolerant BH/BH networks.

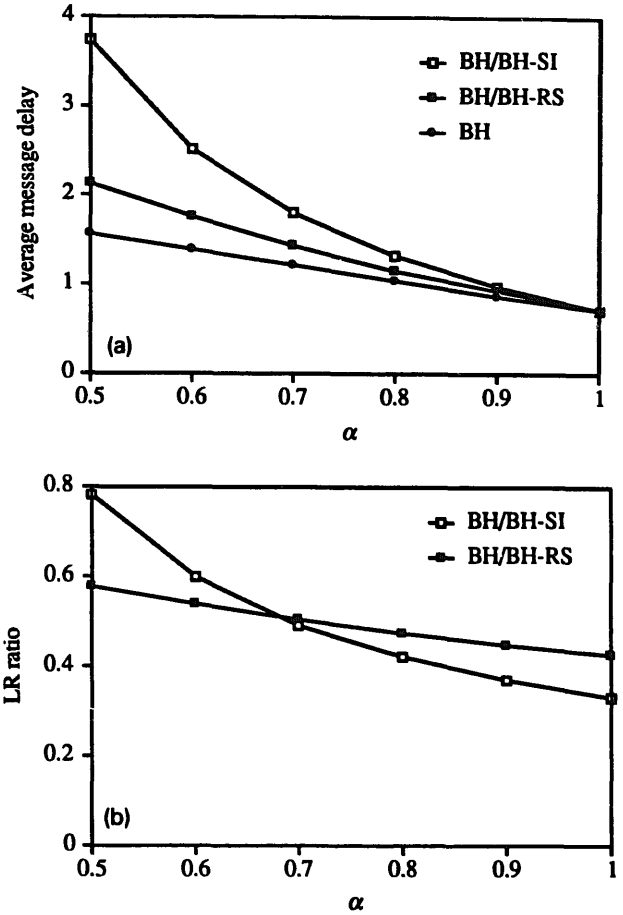

Fig. 6. Average message delay and the corresponding LR ratio ($N = 8192$ and cluster size $= 8$).

The delays associated with the BH/BH-RS networks are substantially smaller compared to those of the BH/BH-SI network for smaller values of α. This improvement in performance is, however, associated with an increase in network cost. From Fig. 6b it can be concluded that, for smaller values of α, the reduction in the average message delay more than compensates for the increase in link

cost of the BH/BH-RS network. The situation is different for α values greater than about 0.7. For these higher α values, since there are fewer inter-cluster messages, the replication of the level 2 network does not justify the increase in cost. It should, however, be noted that the BH/BH-RS network degrades gracefuliy when α values decrease (as is clear from the less steep LR ratio and message delay curves). This is important in a general-purpose multicomputer system that supports a wide range of workloads. Behavior under low communication locality is important even if typical applications have higher α values (say, $\alpha \geq 0.8$). As illustrated in Fig. 6, the BH/BH-RS network handles low communication locality applications better than the BH/BH-SI network. These observations are valid for other network sizes as well.

4.2. *Impact of network size*

Figures 7a and 7b show the average message delay and the corresponding LR ratios, respectively, versus network size N. The network size is increased from 2^7 through 2^{15}. Each graph is plotted for two α values (0.5 and 0.8). The remaining parameters are fixed as in Fig. 6.

From Fig. 7a, it can be seen that, when $\alpha = 0.5$, the BH/BH-RS network provides average message delays that are closer to the BH network than does the BH/BH-SI network. The difference in the average message delays between the two fault-tolerant BH/BH networks increases as the network size increases. However, at $\alpha = 0.8$, the corresponding differences among the three networks is marginal.

When $\alpha = 0.5$, the BH/BH-RS network is uni-

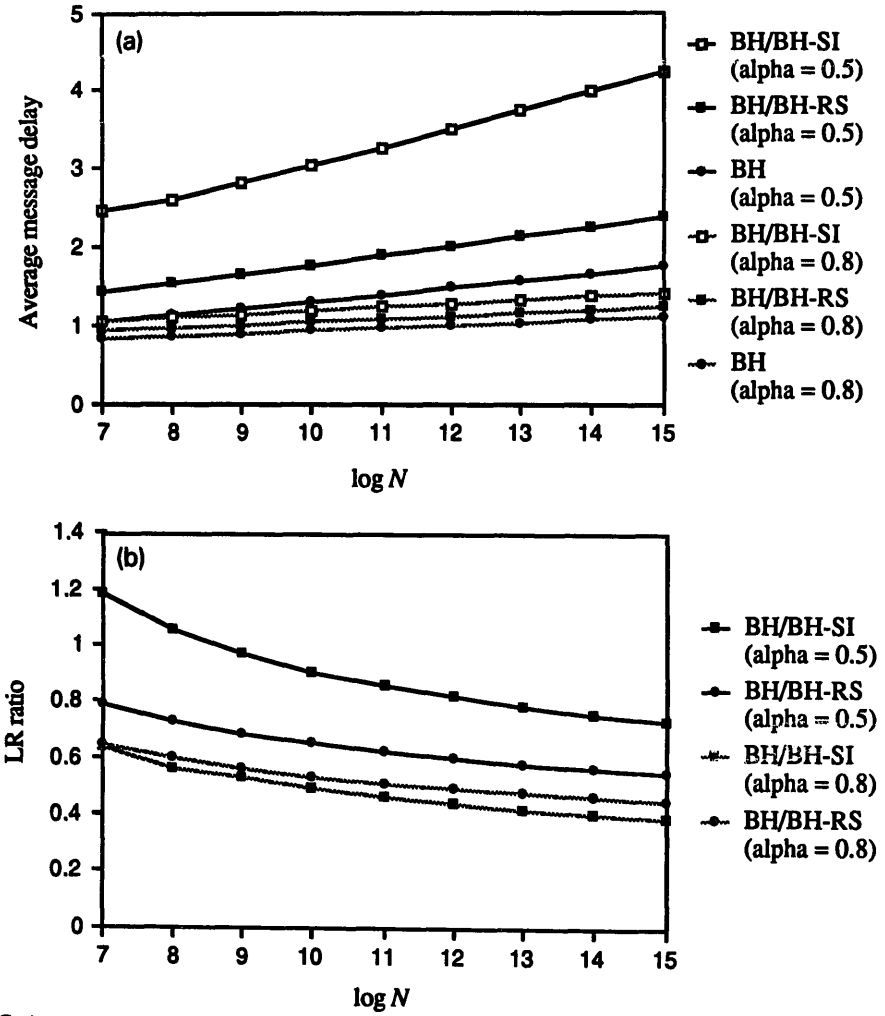

Fig. 7. Average message delay and the corresponding LR ratio as a function of network size N.

formly better than the BH/BH-SI network over the range of network sizes considered. Because the BH/BH-RS network link cost increases with increasing network size, the difference in LR ratios decreases as the network size increases. When $\alpha = 0.8$, the BH/BH-SI network provides marginally better cost–benefit ratio. This is mainly due to the fact that there are fewer inter-cluster messages and the benefit of obtaining decreased message delays is small (see Fig. 7a). This difference in LR ratios increases with increasing network size because of the penalty associated with the BH/BH-RS network in terms of increased link cost without substantially decreasing the average message delay.

4.3. Sensitivity to message generation rate

In message-passing systems, the computation–communication ratio is an important characteristic of an application [29]. If an application is computationally intensive, it tends to generate fewer messages (and hence lower network message load). Here we can model this fact by smaller λ values. On the other hand, a communication-intensive application should be modelled with higher λ values. Furthermore, applications may also vary in their degree of communication locality (in our model, varying values of α). If the overall system workload is of mixed type, the interconnection network should gracefully handle different types of workload.

To assess the capability of these three networks to handle various workloads, we use the saturation message generation rate λ_{sat} supported by each network. Figure 8 shows λ_{sat} values for the three

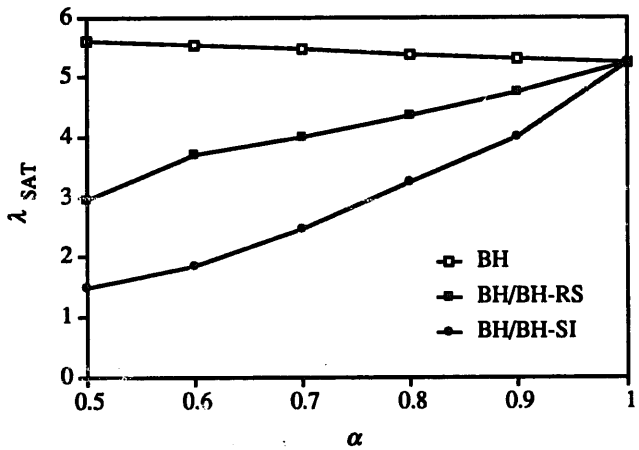

Fig. 8. Saturation message generation rate supported by the three networks as a function of α.

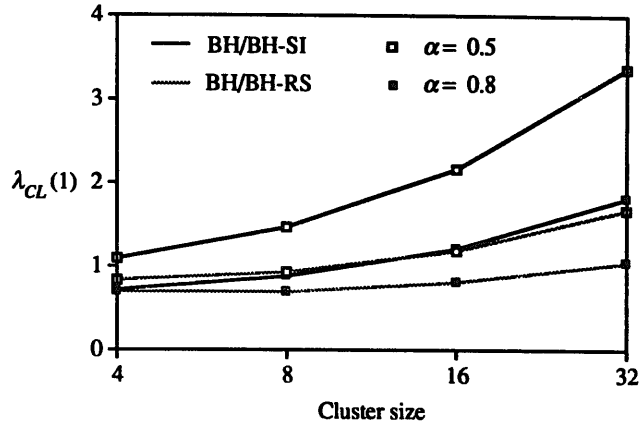

Fig. 9. Effective message arrival rate seen by the cluster links connected to the interface nodes when $\lambda = 1$.

networks with $N = 512$ and cluster size = 8. All other parameters are the same as those used in Fig. 6. At $\alpha = 1$, all three networks are equivalent, and therefore support the same saturation message generation rate. For other α values, the BH/BH-RS network supports λ_{sat} values that are intermediate between those supported by the BH network and the BH/BH-SI network. In particular, for $0.5 \leqslant \alpha \leqslant 0.6$, the BH/BH-RS network supports a message generation rate that is twice that supported by the BH/BH-SI network. This is because at these α values and for the parameters considered here, noncluster links saturate before the cluster links do (this can be verified from $\lambda_{link,CL}$ and $\lambda_{link,NCL}$ equations in Section 3). For higher values of α, it is the cluster links that determine the value of λ_{sat}. (This is the reason why the curves associated with the BH/BH-RS and BH/BH-SI networks show two distinct slopes.)

With increasing cluster size, the BH/BH-SI network supports comparatively smaller λ_{sat} values than does the BH/BH-RS network. This is because, as the cluster size increases, there are more and more inter-cluster messages (for a fixed α value) flooding the cluster links closer to the interface node (Fig. 9). The situation is comparatively better in the BH/BH-RS network because this message load is distributed between the two interface nodes as shown in Fig. 9.

4.4. Performance comparison under constant network cost

This section discusses the relative performance of the two fault-tolerant BH/BH networks under

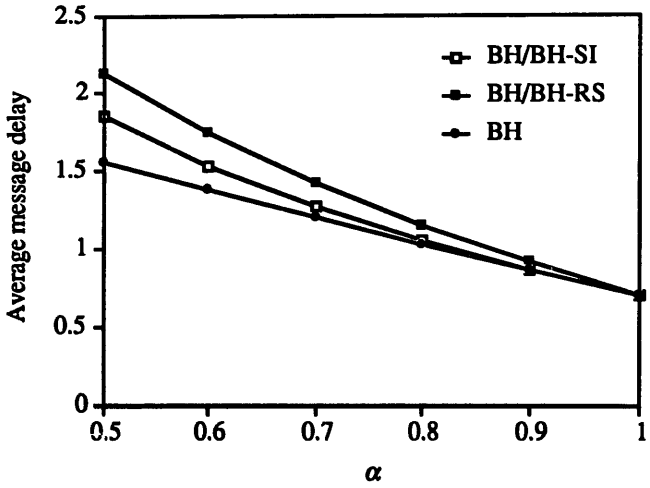

Fig. 10. Average message delay as a function of α under constant network cost.

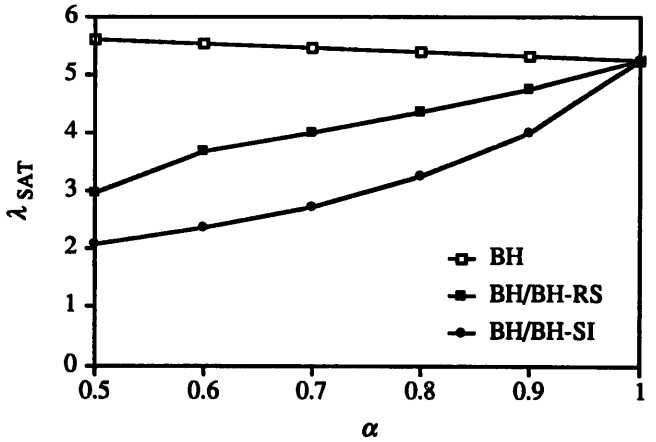

Fig. 12. Saturation message generation rate supported by the three networks under constant network cost.

the constraint that they both use the same number of physical links. As stated earlier, in this case, the BH/BH-SI network maps each logical link in the level 2 network to two physical links. We model this by making $\mu_{NCL} = 2\mu_{CL}$. The BH/BH-RS network essentially redistributes these level 2 network (physical) links of the BH/BH-SI network (therefore we use $\mu_{CL} = \mu_{NCL}$ for the BH/BH-RS network).

Figure 10 shows the average message delay as a function of α. The parameters are the same as those used in Fig. 6 (except that the BH/BH-SI network uses $\mu_{NCL} = 6$). By increasing the message processing rate of the level 2 network links, the BH/BH-SI network gives message delays that are intermediate between the corresponding delays of the BH/BH-RS network and the BH network.

The impact of increasing the network size is demonstrated by Fig. 11, which gives the average message delays as a function of network size for $\alpha = 0.5$ and $\alpha = 0.8$. These graphs show that the major component of the delay in message transmission is contributed by the noncluster links in Fig. 6a. However, if the parameters were chosen such that the major component of the delay is contributed by the cluster links, then such improvement in performance could not be associated with the BH/BH-SI network. This is clear from the saturation message generation rate λ_{sat} supported by these networks as shown in Fig. 12 Comparing Figs. 8 and 12, it can be seen that the improvement in λ_{sat} due to increased λ_{NCL} value is marginal. This is mainly due to the fact the λ_{sat} values are now determined, for the whole range of α values, by when the cluster links saturate. (Note

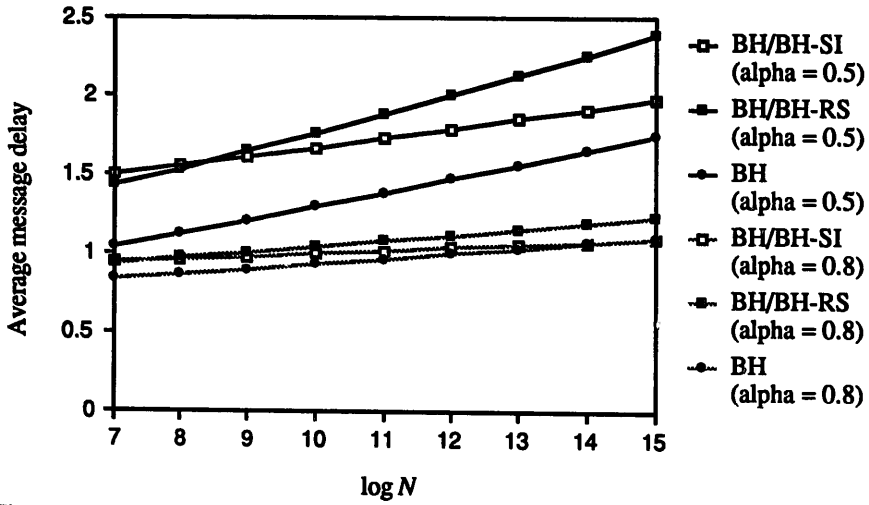

Fig. 11. Average message delay as function of network size under constant network cost.

that, in Fig. 8, for $0.5 \leqslant \alpha \leqslant 0.7$, λ_{sat} is determined by when the noncluster links saturate.)

5. Impact of routing algorithm on network performance

The analysis and the discussion so far have used a random routing algorithm. This section presents the impact of routing algorithm on network performance. The goal is to compare the performance of distributed routing algorithms; in particular our interest is in comparing adaptive routing schemes versus oblivious routing schemes (which disregard the state of the network in making their routing decisions). Note that routing schemes a) and b) belong to the oblivious group; the last one is an adaptive scheme. The adaptive routing scheme requires only the local state information available at a node.

a) Fixed routing: in binary hypercubes routing information can be computed from the source and destination node addresses. The routing code is computed as the exclusive-or of the source and destination addresses. This code indicates the dimensions that the message must traverse to reach its destination. In fixed routing scheme, the routing code is always scanned from lower to higher (we assume here that this corresponds to left to right scan) dimension until a dimension that the message must traverse is found. It should be noted that all commercially available hypercube multicomputer systems use this routing scheme. This scheme has the advantage of avoiding deadlocks when there is finite number of buffers at each node.

b) Random routing: this scheme is similar to the fixed routing scheme described above except that the dimension that the message should follow is selected randomly from the routing code. Random routing, which was used in Section 3, is often assumed for mathematical convenience. Furthermore, it tends to distribute the message load uniformly over all the shortest paths between a pair of nodes in the network.

c) Shortest queue routing: in this routing scheme, at each node, if a message can take one of l links (l is equal to the number of 1's in the routing code computed at that node) the message will be routed to that link which has the shortest queue length. Note that the message still follows the shortest path; since there are several shortest paths in hypercubes, it tends to select a path that minimizes the message delay. Furthermore, all the information it needs is available locally at each node.

5.1. Analysis of bounds on performance

This section derives a lower bound on the average message delay R and also presents network saturation analysis giving an upper bound on the saturation message generation rate (at each node) supported by the network. These values can be used as an yardstick to measure the performance of various routing schemes.

5.1.1. A lower bound on message delay

A characteristic of HINs is that the interface node acts as a "hot-spot" and the cluster links closer to the interface nodes handle higher message traffic than the links that are farther away. While inter-cluster messages will have no choice but to use these links, intra-cluster messages can be routed around these heavily utilized links. In this section, we derive a lower bound on the average message delay over all random routing schemes. This measures the ability of a routing scheme to divert intra-cluster messages as mentioned above.

This bound can be computed by assuming that the message load is uniformly distributed across all links within each cluster and within the level 2 network. We now informally show that this yields a lower bound on the average message delay under the assumptions of Section 3.1. First, note that all messages are routed over a shortest path and, therefore, the total message load is fixed. Thus any deviation from the uniform distribution of this message load means that the message load is transferred from a set of links X to another (disjoint) set of links Y in the network. Since $dR/d\rho$ is an increasing function of load ρ for the M/M/1 queue, the total message delay increase contributed by the set of links Y will be more than the delay reductions achieved by the set of links X. This, therefore, results in a net increase in the average message delay.

Note that if the message load were uniformly distributed, then adaptive routing schemes give average message delays better than this bound.

Acting against this is the imbalance of the message traffic that exists among the cluster links.

BH / BH-RS Network

Consider a source cluster in the BH/BH-RS network. Let λ denote the message generation rate at each node. Following the locality model, the intra-cluster messages are generated at $\alpha\lambda$ rate and inter-cluster messages at $(1-\alpha)\lambda$ rate. Therefore,

message load due to intra-cluster messages

$$= 2^d \alpha\lambda \left(\frac{d2^{d-1}}{2^d - 1} \right).$$

Let P_S and P_d be the average internode distance traversed by intercluster messages within a source cluster and destination cluster, respectively. Then,

$$P_s = \begin{cases} \dfrac{1}{2^d} \left[2 \displaystyle\sum_{i=0}^{\frac{d}{2}-1} i \binom{d}{i} + \frac{d}{2} \binom{d}{d/2} \right] & \text{if } d \text{ is even} \\ \dfrac{1}{2^{d-1}} \left[\displaystyle\sum_{i=0}^{\frac{d-1}{2}} i \binom{d}{i} \right] & \text{if } d \text{ is odd} \end{cases}$$

and

$$P_d = d/2.$$

The message load due to out-going intercluster messages is $2^d(1-\alpha)\lambda P_s$ and that due to incoming intercluster messages (i.e., intercluster messages whose destination nodes are in the cluster under consideration) is $2^d(1-\alpha)\lambda P_d$. Therefore,

$$\lambda_{CL} = \alpha\lambda \left(\frac{2^{d-1}}{2^d - 1} \right) + \frac{(1-\alpha)\lambda}{d} [P_s + P_d].$$

λ_{NCL} is given by $\lambda_{link,NCL}$ in Section 3.2. Then R_B^{RS} is given by

$$R_B^{RS} = \alpha \left(\frac{d2^{d-1}}{2^d - 1} \right) \Delta_{CL} + (1-\alpha) \left[(P_s + P_d) \Delta_{CL} + \left(\frac{(D-d)2^{D-d-1}}{2^{D-d} - 1} \right) \Delta_{NCL} \right],$$

where the average message delay experienced on a cluster link is given by

$$\Delta_{CL} = \frac{1}{\mu_{CL} - \lambda_{CL}}$$

and the average message delay experienced on a non-cluster link is given by

$$\Delta_{NCL} = \frac{1}{\mu_{NCL} - \lambda_{NCL}}.$$

BH / BH-SI Network

For this network, λ_{CL} and λ_{NCL} are given by [7]

$$\lambda_{CL} = \lambda \left[\alpha \left(\frac{2^{d-1}}{2^d - 1} \right) + (1-\alpha) \right],$$

$$\lambda_{NCL} = \frac{2^{D-1}(1-\alpha)\lambda}{2^{D-d} - 1},$$

Then R_B^{SI} is given by

$$R_B^{SI} = \alpha \left[\frac{d2^{d-1}}{2^d - 1} \right] \Delta_{CL} + (1-\alpha) \left[d\Delta_{CL} + \left\{ \frac{(D-d)2^{D-d-1}}{2^{D-d} - 1} \right\} \Delta_{NCL} \right]$$

where

$$\Delta_{CL} = \frac{1}{\mu_{CL} - \lambda_{CL}}$$

and

$$\Delta_{NCL} = \frac{1}{\mu_{NCL} + \lambda_{NCL}}.$$

5.2.2. Saturation analysis

We present an upper bound on message generation rate at individual nodes λ_{sat} that results in network saturation. While the value of λ_{sat} is a function of various network parameters such as the message processing rates of various links and the structure of the network, our interest here is mainly to study the impact of various routing schemes on λ_{sat}. Since there are two types of links in the BH/BH network, we can express λ_{sat} as

$$\lambda_{sat} = \min(\lambda_{sat_{CL}}, \lambda_{sat_{NCL}}).$$

An upper bound on λ_{sat} can be obtained from the analysis of the previous section. It is straightforward to derive the following.

BH/BH-RS Network

$$\lambda_{sat_{CL}} = \frac{\mu_{CL}}{\alpha\left(\frac{2^{d-1}}{2^d-1}\right) + \frac{(1-\alpha)}{d}[P_s + P_d]}$$

and

$$\lambda_{sat_{NCL}} = \frac{\mu_{NCL}(2^{D-d}-1)}{2^{D-2}(1-\alpha)}.$$

BH/BH-SI Network

$$\lambda_{sat_{CL}} = \frac{\mu_{CL}}{\alpha\left(\frac{2^{d-1}}{2^d-1}\right) + 1 - \alpha}$$

and

$$\lambda_{sat_{NCL}} = \frac{\mu_{NCL}(2^{D-d}-1)}{2^{D-1}(1-\alpha)}.$$

5.2. Performance of routing algorithms

The results for the random routing algorithm were obtained analytically (Section 3). We have conducted simulation experiments to obtain the performance of the remaining two routing schemes.

5.2.1. Principal comparison

Figure 13 depicts the average message delay associated with the three routing algorithms for a network with the following parameters: $\lambda = 1$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$, $d = 3$, and $D = 6$ (i.e., network size $N = 64$). The value of α is varied from 0.5 to 1 in steps of 0.1. Figure 14 shows the average message delay as a function of message generation rate at each node λ for $\alpha = 0.7$ (all other parameters remain the same as in Fig. 13).

It can be seen from Fig. 13b that fixed routing results in extreme sensitivity to locality in communication α. For smaller α values, fixed routing gives unacceptable performance even when compared to the performance of the random routing algorithm. The reason for this performance difference is that random routing tends to distribute message load uniformly on all the shortest paths between a pair of nodes whereas the fixed routing

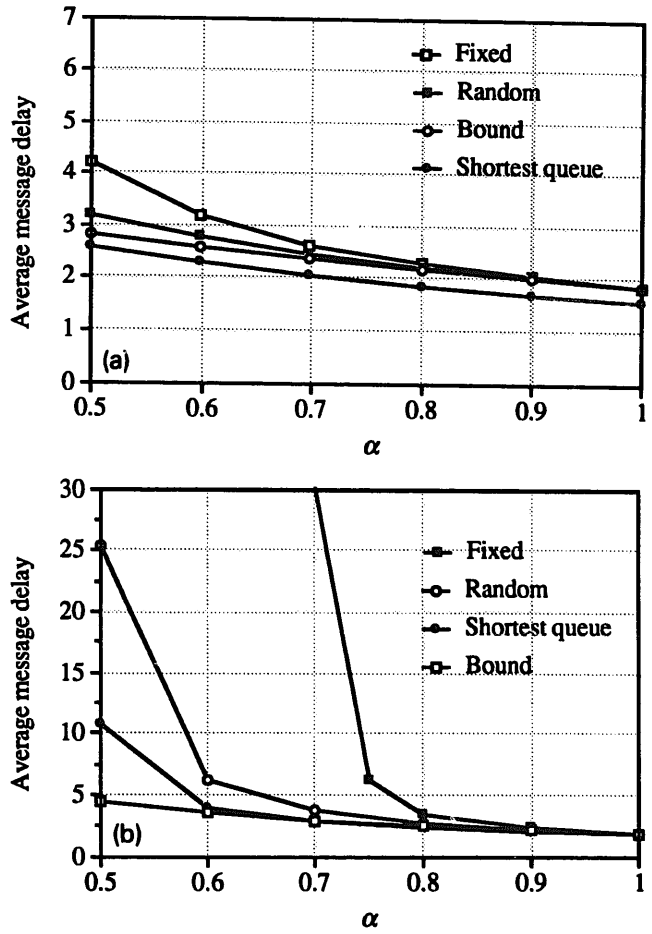

Fig. 13. Average message delay as a function of communication locality α ($D = 6$, $d = 3$, $\lambda = 1$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$): (a) BH/BH-RS network; (b) BG/BH-SI network.

tends to favour one particular shortest path. In a binary hypercube, if communication is uniformly distributed (i.e., every node communicates with every other node with equal probability), both random and fixed routing schemes give similar performance. This is clear from the results for $\alpha = 1$. However, when there is locality in communication, random routing clearly performs better. In HINs, even if there is no locality in communication, because of the structure of the hierarchical network there is "hot-spot" kind of situation where all inter-cluster messages will have to go through their source cluster interface node. With fixed routing, higher dimension *cluster* links attached to the interface node tend to handle larger proportion of intercluster messages. For example, in the BH/BH-SI network, when $d = 3$, the number of intercluster messages processed by the dimension 1 cluster link is twice the corresponding number for the dimension 0 link; the

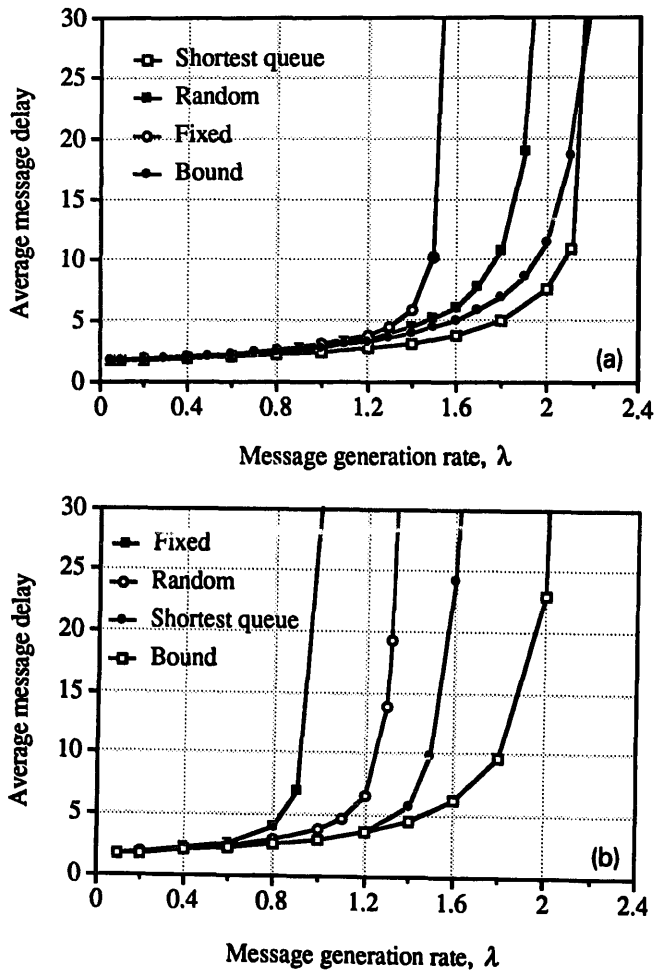

Fig. 14. Average message delay as a function of message generation rate at each node ($D = 6$, $d = 3$, $\alpha = 0.7$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$): (a) BH/BH-RS network; (b) BH/BH-SI network.

dimension 2 cluster link processes four times as many messages. In general, ith dimension cluster link processes 2^i times the number of intercluster messages processed by the 0th dimension cluster link. Since the percentage of intercluster messages increases as α decreases, message delays increase. With random routing, all cluster links attached to the interface node are equally utilized, leading to smaller message delays. For the BH/BH-RS network, the difference in performance between the fixed and random routing is substantially less because the message traffic is better balanced in this network.

The shortest queue routing produces the best performance among the three routing schemes. As α decreases, the performance advantage associated with the shortest queue routing increases. The reason is that, by using the state of certain network links, the shortest queue routing can successfully divert the message traffic to less utilized links. This scheme in particular leads to diverting intra-cluster message load away from the links that are closer to the interface node. For the parameters considered in Fig. 13, the shortest queue routing seems to distribute message load more or less uniformly for $\alpha \geqslant 0.6$. It can be seen that, for the BH/BH-SI network, the shortest queue routing improves performance substantially when compared to random and fixed routing schemes. The network saturation message generation rate increases from 0.9 with fixed routing to 1.3 with random routing to 1.6 with shortest queue routing. For the BH/BH-RS network, the shortest queue routing provides average message delays better than the bound (see Section 5.1.1 for an explanation).

5.2.2. Saturation message generation rate

As in Section 4.3, we use λ_{sat} to assess the capability of these three routing algorithms to handle various workloads. Figure 15 shows the λ_{sat} values for the three routing algorithms when a cluster size of 8 is used. All other parameters are the same as those used in Fig. 13. Note that the upper bound curve for the BH/BH-SI network exhibits two distinct slopes. This is because, for $0.5 \leqslant \alpha \leqslant 0.7$, the noncluster links saturate before the cluster links do. This is not the case for the BH/BH-RS network. For higher values of α, it is the cluster links that determine the value of λ_{sat}. At $\alpha = 1$, all three routing algorithms tend to distribute message load uniformly over the cluster links, and therefore support the same saturation message generation rate. For other α values, the random routing scheme supports λ_{sat} values that are intermediate between those supported by the fixed routing and the shortest queue routing. In particular, for the BH/BH-SI network, for $0.5 \leqslant \alpha \leqslant 0.8$, the fixed routing supports a message generation rate that is approximately half of the corresponding bound value. For the BH/BH-RS network, the shortest queue routing provides λ_{sat} values close to the upper limit. These results suggest that a more sophisticated routing scheme that utilizes global rather than the local state information provides only marginal improvement in performance and may not be worthwhile.

5.2.3. Performance under constant network cost

Figure 16 shows the performance of the three routing schemes under constant network cost (i.e., $\mu_{NCL} = 1.5$ for the BH/BH-RS network and both the networks use the same number of physical links). The average message delay as a function of communication locality is shown in Fig. 16a. Compared to the performance of the BH/BH-SI network (shown in Fig. 13b), the BH/BH-RS network provides substantial performance improvements, particularly at low α values. These results are similar to those shown in Fig. 13a. Similar observations can be made about the results in Fig. 16b.

The saturation message generation rates supported are shown in Fig. 16c as a function of α. The upper bound curve exhibits two distinct slopes compared to that in Fig. 15a. λ_{sat} values are the same for $\alpha \geq 0.8$, whether $\mu_{NCL} = 3$ or $\mu_{NCL} = 1.5$. This is because, at this high communication locality, cluster link message processing capacity determines λ_{sat} values. However, noncluster message processing rates affect λ_{sat} values when $\alpha < 0.8$.

The results presented in this section support, for most part, the conclusions arrived at in the last two sections.

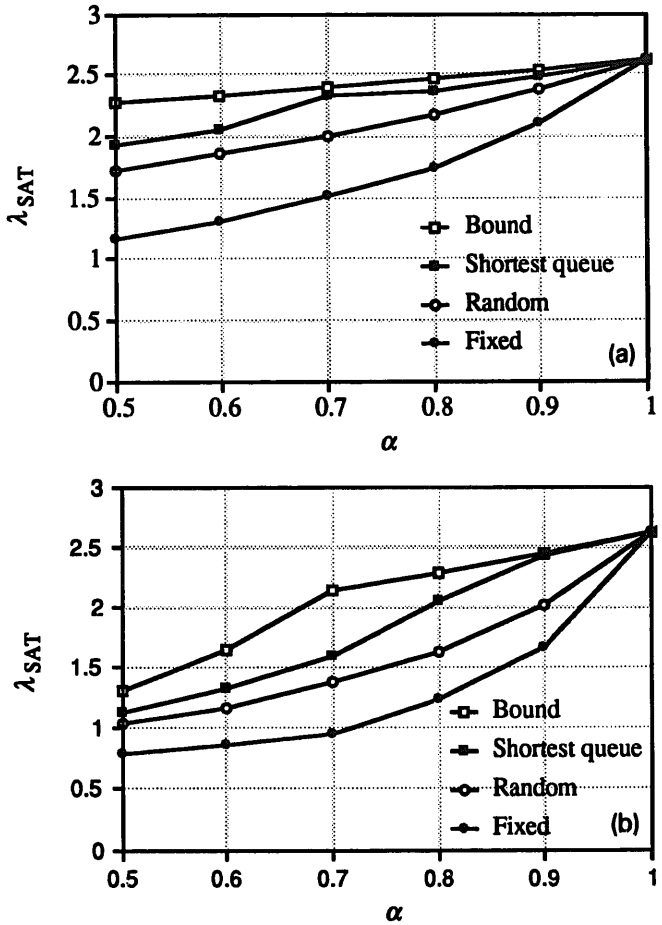

Fig. 15. Saturation message generation rate as a function of communication locality α ($D = 6$, $d = 3$, $\mu_{CL} = 1.5$, $\mu_{NCL} = 3$): (a) BH/BH-RS network; (b) BH/BH-SI network.

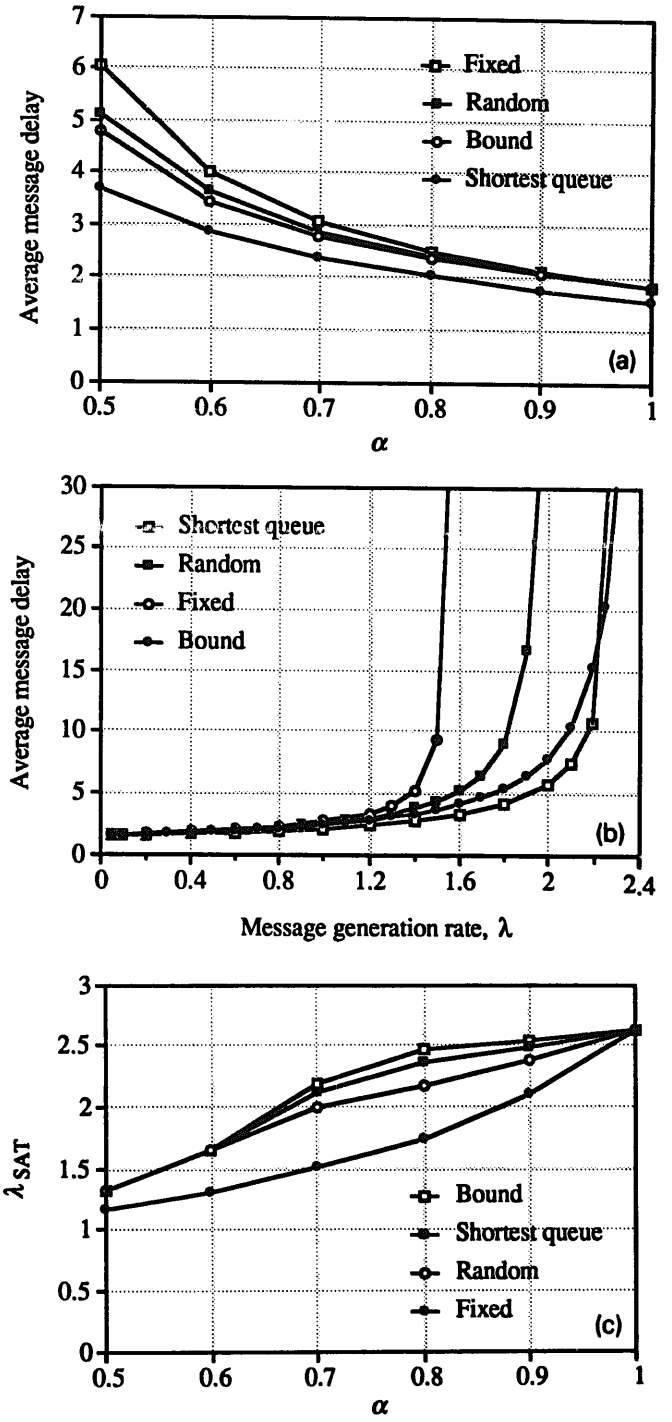

Fig. 16. Performance of the BH/BH-RS network under constant network cost ($D = 6$, $d = 3$, $\mu_{CL} = \mu_{NCL} = 1.5$).

6. Conclusions

As shown in Fig. 4, the BH/BH-SI and the BH/BH-RS networks provide similar reliability improvements. However, from a performance point of view, neither of the two networks dominates the other for all parameter values and system characteristics. If the network supports applications that have varying degrees of communication locality and/or different computation–communication ratios, the BH/BH-RS network is to be recommended. On the other hand, if applications exhibit high degrees of communication locality and high computation–communication ratios, the BH/BH-SI network provides marginally better performance. It should, however, be noted that as the cluster size increases, the BH/BH-RS network tends to improve its performance over that of the BH/BH-SI network.

The fixed routing scheme tends to utilize links in a nonuniform fashion. This leads to increased message delays and decreased saturation throughputs. In contrast, the random routing algorithm tends to balance message traffic better than the fixed routing scheme. Even though the random routing takes no knowledge of the network state in making its routing decisions, it provides considerable performance improvement over the fixed routing scheme. The shortest queue routing scheme provides the best performance among the routing schemes considered in this study. It provides performance close to the bound values when there is moderate to strong locality in communication. The results show that any improvement obtainable by using a more complex routing scheme that uses global rather than local network state information should be marginal and may not be worthwhile. The BH/BH-RS network is less sensitive to the routing algorithm because this network is a better balanced network than the BH/BH-SI network.

A weakness of the BH/BH-RS and BH/BH-SI networks is that their reliability is uniformly poorer than the BH network. This drawback can be remedied by adding standby-spare interface nodes to the BH/BH-RS network (we refer to this network as the BH/BH-RS&SI network). This improves overall network reliability substantially as shown in Fig. 4. Its reliability is comparable to that of the BH network. The performance of this network, however, remains the same as that of the BH/BH-RS network. The BH/BH-RS&SI network, therefore, may be attractive in high reliability applications providing reliability as good as that of the BH and the complete connection networks.

Acknowledgements

I thank the anonymous referees for their constructive comments on previous versions of this article. I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and by Carleton University.

References

[1] D.P. Agrawal, V.K. Janakiram and G.C. Pathak, Evaluating the performance of multicomputer configurations, *IEEE Comput.* **19** (5) (1986) 23–37.

[2] B.W. Arden and H. Lee, Analysis of chordal ring network, *IEEE Trans. Comput.* **30** (4) (1981) 291–295.

[3] L.N. Bhuyan and D.P. Agrawal, Design and performance of generalized interconnection networks, *IEEE Trans. Comput.* **32** (12) (1983) 1081–1090.

[4] L.N. Bhuyan and D.P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, *IEEE Trans. Comput.* **33** (4) (1984) 323–333.

[5] L.N. Bhuyan and C.R. Das, Dependability evaluation of multicomputer networks, in: *Proc. 1986 Int. Conf. Parallel Processing* (1986) 576–583.

[6] D. Carlson, The mesh with a global mesh: a flexible, high-speed organization for parallel computation, in: *Proc. 1st Int. Conf. on Supercomputing Systems* (IEEE Computer Society Press, 1985) 618–627.

[7] S.P. Dandamudi, A performance comparison of routing algorithms for hierarchical hypercube multicomputer networks, in: *Proc. Int. Conf. Parallel Processing* (1990) vol. 1, pp. 281–285.

[8] S.P. Dandamudi, A comparison of link-oriented multicomputer interconnection networks, Research Note, School of Computer Science, Carleton University, Ottawa, Canada (1990).

[9] S.P. Dandamudi, *Hierarchical Hypercube Multicomputer Interconnection Networks* (Ellis Horwood, Market Cross House, Chichester, 1991).

[10] S.P. Dandamudi and D.L. Eager, Hierarchical interconnection networks for multicomputer systems, *IEEE Trans. Comput.* **39** (6) (1990) 786–797.

[11] S.P. Dandamudi and D.L. Eager, On hypercube-based hierarchical interconnection network design, *J. Parallel and Distributed Comput.* **12**(3) (1991) 283–289.

[12] C.R. Das, J.T. Kreulen, M.J. Thazhuthaveetil and L.N. Bhuyan, Dependability modeling for multiprocessors, *IEEE Comput.* **23** (10) (1990) 7–19.

[13] N. Deo, *Graph Theory with Applications to Engineering and Computer Science* (Prentice-Hall, 1974).

[14] T-Y. Feng, A survey of interconnection networks, *IEEE Comput.* **14** (1981) 12–27.

[15] K. Ghose and K.R. Desai, The HCN: a versatile interconnection network based on cubes, in: *Proc. Supercomputing Conf.* (1989) 426–435.

[16] K. Ghose and K.R. Desai, The design and evaluation of the hierarchical cubic network, in: *Proc. 1990 Int. Conf. Parallel Processing* (1990) vol. 1, pp. 355–362.

[17] J.R. Goodman and C.H. Sequin, Hypertree: a multiprocessor interconnection topology, *IEEE Trans. Comput.* **30** (12) (1981) 923–933.

[18] G. Gopal and J.W. Wong, Delay analysis of broadcast routing in packet-switching networks, *IEEE Trans. Comput.* **30** (1981) 915–922.

[19] J.P. Hayes, T.N. Mudge, Q.F. Stout, S. Colley and J. Palmer, Architecture of a hypercube supercomputer, in: *Proc. 1986 Int. Conf. Parallel Processing* (1986) 643–660.

[20] W.D. Hillis, *The Connection Machine* (MIT Press, Cambridge, 1985).

[21] K. Hwang and J. Ghosh, Hypernet: a communication-efficient architecture for constructing massively parallel computers, *IEEE Trans. Comput.* **36** (12) (1987) 1450–1466.

[22] A.D. Ingle and D.P. Seiwiorek, Reliability models for multiprocessor systems with and without periodic maintenance, in: *Proc. 7th Ann. Int. Conf. FTC* (1977) pp. 3–9.

[23] B.W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems* (Addison-Wesley, Reading, 1989).

[24] J. Kim, C.R. Das, W. Lin, and T. Feng, Reliability evaluation of hypercube multicomputers, *IEEE Trans. Reliab.* **38** (1) (1989) 71–73.

[25] L. Kleinrock, *Queueing Systems, vol.* 1 (Wiley, New York,, 1975).

[26] L. Kleinrock, *Queueing Systems, vol.* 2 (Wiley, New York, 1976).

[27] S. Lakshmivarahan and S.K. Dhall, A new hierarchy of hypercube interconnection schemes for parallel computers, *J. Supercomput.* 2 (1988) 81–108.

[28] E.D. Lazowska, J. Zahorjan, G.S. Graham and K.C. Sevcik, *Computer System Performance* (Prentice-Hall, Englewood Cliffs, 1984).

[29] B. Lint and T. Agerwala, Communication issues in the design and analysis of parallel algorithms, *IEEE Trans. Software Eng.* **7** (1981) 174–188.

[30] F.P. Preparata and J. Vuillemin, The cube-connected-cycles: a versatile network for parallel computation, *Comm. ACM* **24** (1981) 300–309.

[31] J. Rattner, Concurrent processing: a new direction in scientific computing, in: *Proc. AFIPS Conf.* **54** (1985) NCC pp. 157–166.

[32] D.A. Reed and D.C. Grunwald, The performance of multicomputer interconnection networks, *IEEE Comput.* **20** (6) (1987) 63–73.

[33] D.A. Reed and R.M. Fujimoto, *Multicomputer Networks: Message-Based Parallel Processing* (MIT Press, Cambridge, 1987).

[34] C.L. Seitz, The cosmic cube, *Comm. ACM* **28** (1985) 22–33.

[35] A.S. Tanenbaum, *Computer Networks* (Prentice-Hall, 1981).

[36] L.D. Wittie, Communication structures for large networks of microcomputers, *IEEE Trans. Comput.*, **30** (4) (1981) 264–273.

[37] S.W. Wu and M.T. Liu, A cluster structure as an interconnection network for large multimicrocomputer systems, *IEEE Trans. Comput.* **30** (4) (1981) 254–264.