IEEE/ACM Transactions on Networking, August 1994, Volume 2, Number 4, pp. 363-373.

Analyzing the Fault Tolerance of Double-Loop Networks *

Jon M. Peha and Fouad A. Tobagi
Address: Carnegie Mellon University, Dept of ECE, Pittsburgh, PA 15213-3890
Phone: (412) 268-7126, Fax: : (412) 268-2860, e-mail: peha@ece.cmu.edu

ABSTRACT

This paper analyzes the fault tolerance of a class of double-loop networks referred to as forward-loop backward-hop (FLBH), in which each node is connected via unidirectional links to the node one hop in front of it and to the node S hops in back of it for some S. A new measure of fault tolerance is described, along with techniques based on Markov chains to calculate upper and lower bounds on the fault tolerance of this network topology quickly and efficiently. The results of these calculations provide a more precise description of network fault tolerance than has been achieved with previously published techniques.

1 Introduction

Loop- or ring-based topologies have become increasingly important for local and metropolitan area networks. Loops support simple and efficient protocols. Their symmetry reduces the diversity of hardware components needed, and their use of unidirectional point-to-point links allows efficient fiber implementations. Many network architectures have been designed with loop-based topologies, [1] including the 802.5 token ring, the Fiber Distributed Data Interface (FDDI) network, the Distributed Queue Dual Bus (DQDB) with redundancy, and the Synchronous Optical Network (SONET) ring. One great liability of the simple loop is that any single failure disrupts communication. As a result, there has been considerable research into methods of enhancing the fault tolerance of a loop network, where fault tolerance is some measure of the likelihood that connectivity among nodes is maintained despite failures. Liu et al. [2] proposed the Distributed Double Loop Computer Network (DDLCN) which consists of two counter-rotating loops. (FDDI, DQDB, and SONET are examples of a pair of counter-rotating loops.) This topology can tolerate any single failure and a few multiple failures without any loss of connectivity. Various algorithms and protocols can be used to perform reconfiguration that will make use of redundant paths in a DDLCN. [3, 4, 5]

To further improve fault tolerance, topologies consisting of a loop with some redundant links have also been explored. [6, 7, 8] One of these loop topologies is the forward-loop backward-hop (FLBH) double-loop network, in which a node has outgoing links to the nodes one "forward" and S "backward" for some S. [9, 10, 11, 12, 13, 14, 15] Thus, if the nodes of an N-node network are numbered from 0 to $N-1$, node i has outgoing links to node $(i-1)$ modulus N and node

*The work was supported, in part, by the National Science Foundation under grant NCR-9210626

$(i + S)$ modulus N. Figure 1 shows an FLBH network with N=15 and S=3. The links from i to $(i - 1)$ modulus N are referred to as short links, while the others are called long links. (To be consistent with the analysis portion of this paper, a "forward" or short link goes in the negative direction from node i to node $i-1$.) In the special case where $S = 1$, an FLBH network reduces to a DDLCN. An FLBH network with $S = 2$ is referred to as a daisy chain. [16] An FLBH network with $S > 1$ has many alternate paths for each source-destination pair, whereas the DDLCN has only two. This makes some aspects of operation more complicated, especially routing, which is receiving considerable attention. [20, 21, 22, 23] However, these additional paths offer many advantages. For the same hardware as the DDLCN, an FLBH network supports significantly greater throughput and smaller mean path distance across all source-destination pairs. [9, 10, 11, 12, 13, 14, 24] In addition to performance, fault tolerance is an important advantage of FLBH networks. As a result, FLBH networks have been considered for applications where reliability is essential, such as in space systems [14] and military systems. (A further generalization would give node i outgoing links to node $i + S$ modulus N and node $i - T$ modulus N for some S, T. However, allowing $T : T \neq 1$ only complicates the topology, and does not reduce the average path length from source to destination. [24])

Figure 1: FLBH Double-loop network. N=15, S=3.

In order to design a network that meets the fault tolerance requirements of its intended application, it is necessary to define a measure of fault tolerance appropriate for that particular application, and to assess the measure for various network configurations. Considering that failures may be described by some probabilistic model, in order to calculate the most appropriate measure for any application, it would be necessary to determine the joint probability distributions that node i is accessible to node j for all combinations of i and j. The only method known to determine these joint distributions is complete enumeration, which is only tractable in very small networks. It is therefore necessary to select a measure which corresponds to a network's ability to maintain connectivity for a wide range of applications, and which can be calculated for large networks. The measure of network fault tolerance chosen in this paper is f, the mean fraction of nodes accessible to a node, given that the latter is up.

$$f = E[\text{number of nodes accessible to a functioning source}]/(N - 1)$$

For applications in which it is important to evaluate the probability of maintaining connectivity between particular nodes, the probability $P(k)$ that node k is accessible from a given functioning node is also considered. Without loss of generality, this given node (the source) is defined to be node 0 throughout the paper. The measure f can be expressed in terms of $P(k)$ as follows.

$$f = \frac{1}{N - 1} \cdot \sum_{k=1}^{N-1} P(k) \tag{1}$$

Fault tolerance of FLBH networks must be measured across the expected range of N for each value of S under consideration. A network designer can then determine whether any FLBH network can meet the intended application's requirements, and if so, a value of S can be selected that yields the desired tradeoff between fault tolerance and other design objectives, such as throughput and ease of reconfiguration. [25]

Section 2 presents the fault tolerance measures and calculation techniques used in previous studies of FLBH networks. In the following section, a new approach to calculating the fault tolerance measure f is presented. Finally, results for several sample networks are presented in Section 4, as is the computer time required to achieve these results.

2 Related Work

There have been several previous efforts to determine the fault tolerance of FLBH networks. Raghavendra et al. [8, 9, 12] use two measures which reflect aspects of fault tolerance, but which do not correspond directly to the probability of maintaining connectivity. They define distance from a source to a destination as the minimum number of hops that must be traversed to get from the source to the destination. The first of their measures is the diameter of the network, i.e., the largest distance across all possible source-destination pairs. This gives some indication of how reliable paths are, since longer paths generally tend to decrease fault tolerance. The second measure is the number of minimum-length paths for a source-destination pair separated by a distance equal to the diameter of the network. This measure gives some indication of the number of alternate paths in the network, where a large number of alternate paths generally tends to increase fault tolerance. They optimize both their measures of fault tolerance with respect to S, and they conclude that the FLBH network where S is approximately $\sqrt{N}$ is optimum for both measures, as well as for delay and throughput characteristics [8, 9, 12], although it has since been shown that the network diameter is not always minimized when $S = \sqrt{N}$. [26, 27] These measures are easy to calculate and provide some insight into the fault tolerance of FLBH networks, but they have obvious limitations. The minimum-length paths are not disjoint, so the measure of alternate paths does not reflect the fact that a single failure could block many paths. There are also many alternate paths that are not minimum length, and the impact of these is not considered in either measure. Also, the only paths considered are those between a source and destination that are separated by a distance equal to the diameter of the network, which may indicate little about the fault tolerance across other source-destination pairs.

In the case where $S = 2$ (the daisy chain), Grnarov et al. [16] found a technique to evaluate a more direct measure of fault tolerance. They determined the probability that one node is accessible to another in the presence of failures given both are up (the terminal reliability) for each possible pair of nodes. They assumed failures were independent. The algorithm used [28] requires the enumeration of all possible paths from source to destination. The cut-sets are determined using Boolean algebra manipulations of variables representing the links and nodes in each path. Unfortunately, this technique does not extend well to FLBH networks with $S > 2$ because the number of alternate paths is far greater.

Ibe [17] calculates a comparable measure of performance for some special cases of FLBH networks. He compares the reliability of single-loop networks with station bypass, double-loop networks with $S = 1$ (the simple DDLCN), and double-loop networks with $S = N - 2$. (The latter is a "braided network" [18], as used in [19].) Links are assumed to be reliable, but nodes can fail. When $S = 1$ or $S = N - 2$, node i is inaccessible to node $i + j$ if and only if one or two consecutive nodes between i and $i + j$ have failed, respectively. This greatly simplifies analysis, but the approach cannot be applied to other values of S.

In papers by Masuyama [29] and by Masuyama and Ichimori [30], the tolerance to node failures (but not link failures) of FLBH networks with arbitrary S was evaluated by determining, for each possible set of node failures, the exact number of node pairs (i, j) such that i can send a message to j. This is done by enumerating all possible patterns of node failures, where a pattern is defined by the distance between failures, and then counting the communicating source-destination pairs for each pattern. Assuming failures are independent and identically distributed throughout the network, it is then possible to determine the probability that a randomly chosen source can send a message to a randomly chosen destination, given the number of failures that have occurred. They dealt with double and triple node failures explicitly and conjectured results for larger numbers of failures. The primary problem with this approach is that it is extremely difficult to enumerate all possible sets of failures and determine the resulting number of communicating source-destinations pairs. In these two papers, the factor used to differentiate failure patterns is whether failures occur $S+1$ nodes apart. Except when $S = N/2$, they claim that all functioning nodes can communicate unless there is at least one pair of failures that occur $S+1$ nodes apart. While this is the case for double failures (if $S \neq 1$), it is incorrect for three or more failures. There are many failure patterns other than failures occurring $S+1$ apart that can reduce the connectivity of a network, such as the triple-failure case shown in Figure 2. Since their conjecture about networks with more than three failures is based on the same premise, it is also incorrect. See [31] for a more detailed critique of this approach.

Figure 2: An FLBH network in which not all functioning nodes can communicate, distance between failures $\neq S+1$. N=16, S=2.

Smith and Trivedi [32] also evaluate the fault tolerance of FLBH networks using an approach based on enumeration of fault sets. However, their measure of fault tolerance is different: the expected length of time before it is no longer possible for all N nodes to communicate with each other, given that all nodes and links were initially up. The time before link failure is assumed to be independent and exponentially distributed with identical mean for all links. The value of this measure is easier to calculate by enumeration than is the measure used by Masuyama and Ichimori, because it is sufficient to characterize failure sets that reduce the number of communicating nodes without determining the extent of this reduction. Since any node failure makes it impossible for all N nodes to communicate, this is only a useful measure by which to compare network topologies if nodes are considered to be reliable. For the case Smith and Trivedi consider in which $N = S^2$ (and thus N must be a perfect square), they are able to prove some properties about the link failure sets that disrupt communication. They then enumerate the failure sets with these properties. They were able to do this exactly for $N = 4$ and $N = 9$, but for $N > 9$ the large number of failure sets forced them to settle for upper and lower bounds. If they were to broaden their inquiry to include values of $S \neq \sqrt{N}$ using the same approach, then, for a given N, they would get the same upper and lower bounds on fault tolerance for all S. Thus, even when using a measure of fault tolerance that is relatively simple to calculate by enumeration, and only considering link failures, the utility of enumerative techniques is limited.

The primary difficulty with methods based on the enumeration of failure sets or the enumeration of paths is the cardinality of the sets to be enumerated. In a network of N nodes, there are a total of $3N$ elements (nodes and links) that can fail, so there are 8^N failure sets. The number of loop-free paths also goes up rapidly with N for $S > 2$. Since enumeration is only tractable for very small networks, a technique is needed to calculate some useful measure of fault tolerance that is not based on enumeration.

Hu and Hwang [33] successfully avoided enumeration through their choice of a measure for fault tolerance: the probability that there exists some nodes i and j such that node i is functioning, node j is functioning, and node i cannot transmit to node j. Although they did not calculate fault tolerance with this measure, they sought the value of S that would minimize it. Nodes can fail with equal and independent probabilities, and links are assumed to be reliable. They were able to solve this problem if and only if N is even, in which case they conclude that S should equal $N/2 - 1$. This result is apparent from the following observations. First, since we are only concerned with the probability that two functioning nodes will not be able to communicate and not the extent to which communications will be disrupted, double node failures will dominate cases of more than two failures. Second, as Masuyama observed, such a communications failure can only occur with two node failures if those failures are $S+1$ apart. [29, 30] There are N node pairs that are $S+1$ apart, unless N is even and $S = N/2 - 1$, in which case there are $N/2$ such node pairs. Consequently, this measure of fault tolerance is optimized when $S = N/2 - 1$. The problem is that the impact of a two-node disruption is far more severe in the special case where $S = N/2 - 1$. In this case, a functioning node will, on average, find less than half of the other functioning nodes accessible. With any other choice of S between 1 and $N-1$ (except N/2), one functioning node would lose the ability to transmit, one would lose the ability to receive, and all other communications would be unaffected. With this measure of fault tolerance, we would consider such a scenario a network failure, whereas the case of 4 node failures would not be a network failure unless it happened to interfere with communications. As an example, let a disruptive double-node failure occur when $N = 100$. With $S \neq N/2 - 1$, a functioning node would find 95.3 of the other 97 functioning nodes accessible on average. With $S = N/2 - 1$, it would find 48 accessible on average. The poor fault tolerance achieved with $S = N/2 - 1$ when the extent of failure is considered will be demonstrated in the figures in Section 4. Moreover, $S = N/2 - 1$ has adverse effects on throughput, delay [9, 10, 11, 12, 13, 14, 24], and reconfigurability [25], so there is incentive to find an approach that estimates fault tolerance with smaller values of S.

3 Analysis

In this section, we calculate upper and lower bounds on the fault tolerance of FLBH networks given the following.

R_N = the probability a node is up.

R_S = the probability a short link is up.

R_L = the probability a long link is up.

N = the number of nodes in the network.

S = the skip distance.

As described in Section 1, fault tolerance is measured as follows.

$P(k)$ = the probability that node k is accessible to node 0.

f = the expected fraction of nodes that are accessible to a given functioning node.

3.1 Approach

Without loss of generality, we can assume that a node is accessible to the source only if there is a loop-free path from the source to that node, because the existence of a path that is not loop-free implies the existence of one that is. Consequently, one type of path that need not be considered in this analysis is one in which a long hop is followed or preceded by S or more short hops. All remaining paths take one of two forms: paths consisting entirely of short links and paths with at least one long link and no more than $S-1$ consecutive short links. Since it is simple to determine the probability that the one path to a given node containing only short links is up, we will initially concentrate on the paths containing at least one long link.

Loop-free paths containing long links have two characteristics that can be used to simplify analysis. First, a path from node 0 to node k must include at least one node from any block of S consecutive nodes along the way. This means the path includes at least one node in the range from node $(i-S+1)$ modulus N to node i for all $i : 0 \le i \le k$. Second, the final hops before reaching the destination node consist of a long hop followed by 0 to $S-1$ short hops. Thus, a path cannot extend more than $S-1$ nodes beyond the destination unless it traverses the entire loop and approaches the destination once again from the direction of the source. This only occurs in multiple cycle paths, where the number of cycles is defined as follows. Letting distance $d = S\cdot$(number of long hops)$-$(number of short hops), there are $\lceil d/N \rceil$ cycles in a path. See Figure 3 for an example of a two cycle path. When this does occur, by the first property of these paths discussed above, a path that traverses the entire network must include at least one of any S consecutive nodes in the network. Assume that one could somehow determine whether or not some set of S consecutive nodes are accessible to the source. If one also knew whether all of the nodes from these S consecutive nodes to the node $S-1$ past the destination were up, as well as the status of their associated links, it would be possible to determine whether or not the destination was accessible to the source. In particular, since the source is always accessible to itself, if one knew the accessibility of the nodes $\in \{N-S+1, N-S+2, \ldots, N-1\}$, and whether the nodes $\in \{1, 2, \ldots, k+S-1\}$ and their associated links were up, it would be possible to determine whether k was accessible to the source. The status of the nodes $\in \{k+S, \ldots, N-S\}$ would be irrelevant. Given these properties, it is possible to define a node-indexed Markov process, the state of which contains information pertaining to the accessibility of S consecutive nodes.

Figure 3: 2 cycle path in an FLBH network. N=10, S=3.

Let $Z^{(0)}$ be a vector of S elements which denotes the accessibility from the source of the nodes $\in \{N-S+1, N-S+2, \ldots, N-1, 0\}$. If a node is accessible to the source, then the corresponding element is assigned an O; otherwise it is assigned an X. For $i : 0 < i < N$, let $Z^{(i)}$ also be a vector of S elements which indicates the accessibility of the nodes $\in \{(i-S+1)\text{mod}N, \ldots, i\}$, using paths with some restrictions; $Z^{(i)}$ is defined from $Z^{(0)}$ and the state of the nodes $\in \{1, 2, \ldots, i\}$ and their associated links as follows. Let k be a node $\in \{(i-S+1)\text{mod}N, \ldots, i\}$. If node k is up, and there exists a node $j \in \{N-S, \ldots, N-1, 0\}$ such that j is accessible to the source, and there exists a path from j to k that includes no nodes $\in \{i+1, i+2, \ldots, N-S\}$, then the element in $Z^{(i)}$ corresponding to node k is assigned an O. If node k is up, and it is accessible to node i via short links, but there does not exist a node $j \in \{N-S, \ldots, N-1, 0\}$ such that j is accessible to the source, and there exists a path from j to k that includes no nodes $\in \{i+1, i+2, \ldots, N-S\}$, then the element corresponding to node k is assigned an &. Otherwise, an X is assigned. An O indicates that node k is definitely accessible to the source, and an X indicates that node k is definitely not accessible to the source. An & indicates that node k is accessible if and only if node i is accessible, a fact that cannot be ascertained based only on $Z^{(0)}$ and the status of the nodes $\in \{1, 2, \ldots, i\}$, because the only possible paths to node i go through node $i+1$. Note that in the special case where $i = 0$, because i is by definition accessible to the source, the definition of $Z^{(i)}$ reduces to the definition given for $Z^{(0)}$.

It is possible to determine $Z^{(i)}$ from $Z^{(i-1)}$ and the status of node i, the link from node i to node $i-1$, and the link from node $(i-S)$ modulus N to node i. Thus, assuming the status of this node and links is known, the process $Z^{(i)}$ is Markovian and deterministic. If node i is down, it is marked with an X. If it is up, node $i-S$ is accessible to the source, and the link from node $i-S$ is up, then node i is marked with an O. Otherwise, it is marked with an &. The accessibility of a node $\in \{i-S+1, \ldots, i-1\}$ must be the same in $Z^{(i)}$ as in $Z^{(i-1)}$ unless it was marked with an &. In this case, if node i is accessible from the source and the link from node i to node $i-1$ is up, it is marked with an O. On the other hand, if node i is inaccessible (marked X), or the link from node i to node $i-1$ is down, then it should be marked with an X. Otherwise, the label remains &. Note that in $Z^{(i)}$, no node marked with either an X or an O can be preceded by a node marked with an &. As a result, the cardinality of the state space is $2^{S+1}-1$ rather than 3^S.

For a given failure set, the process Z can be used to determine which nodes are accessible even if the accessibility of the nodes $\in \{N-S+1,\ldots,N-1\}$ are not initially known. In this deterministic case, $P(k)$ is 0 or 1 for all k. The procedure will now be described, followed by an example. Let $Z_0^{(0)}$ be the state where only the source is accessible. Proceed to $Z_0^{(N-1)}$ using the step-by-step mechanism described above. $Z_1^{(0)}$ is then calculated from $Z_0^{(N-1)}$ in a manner similar to the typical single-step transition, except that the new node (the source) is automatically considered accessible and marked with an O. The nodes marked as accessible in $Z_1^{(0)}$ are those that can be reached via single cycle paths that begin with a long link or via the path containing only short links. With another iteration of this process, the nodes marked as accessible in $Z_2^{(0)}$ are those marked as accessible in $Z_1^{(0)}$, as well as the rest of the nodes that can be reached with single cycle paths and those that can be reached with double cycle paths beginning with a long link. A path can have no more than S cycles, since a loop-free path must begin each cycle with a different node $\in \{N-S+1,\ldots,N-1,0\}$. Thus, after S iterations, $Z_S^{(0)}$ must equal the desired $Z^{(0)}$, and consequently all $Z^{(i)}$ can be found. Since a path containing long links can extend no more than $S-1$ nodes beyond a destination without traversing the entire network again, node k is only accessible via paths containing long links if the first element of $Z^{((i+S-1) mod N)}$ is an O.

The process will now be demonstrated on the network of Figure 3 in which $N=10$, $S=3$, and nodes 2 and 9 have failed. Initially, $Z_0^{(0)}$ is chosen so that only the source is accessible, i.e. $Z_0^{(0)}=$ XXO. Because node 1 is up, but is not accessible via the long link from node 8, node 1 is accessible if and only if it can be reached through node 2, which is not yet known. Consequently, $Z_0^{(1)}=$ XO&. Next, when the failed node 2 is considered, it is clear that node 1 and node 2 are both inaccessible, so $Z_0^{(2)}=$ OXX. Note that node 1 is actually accessible, but not from single-cycle paths that begin with a long link, which are the only paths included in this iteration. Node 3 is accessible via the long link from node 0, so $Z_0^{(3)}=$ XXO. Continuing the process, $Z_0^{(4)}=$ XO&, $Z_0^{(5)}=$ O&&, $Z_0^{(6)}=$ OOO, $Z_0^{(7)}=$ OOO, $Z_0^{(8)}=$ OOO, $Z_0^{(9)}=$ OOX. Finally, $Z_1^{(0)}=$ OXO. By repeating the entire process a second time with $Z_1^{(0)}=$ OXO instead of XXO, the two-cycle path to node 1 will be included, i.e. $Z_1^{(3)}=$ OXO, whereas $Z_0^{(3)}=$ XXO.

In the non-deterministic case in which each node and link is considered up with some given a priori probability, one way to find $P(k)$ is to enumerate all possible failure sets and their probability distribution, use the above procedure to determine accessibilities, and then to calculate the average $P(k)$. However, this would constitute an enumerative method, and not the most efficient one. The speed of this calculation could be improved if it were possible to work with all of the failure sets simultaneously. Computational complexity would be reduced from $O(N8^N)$ to $O(N)$.

One might conceive of an approach to achieve this as follows. $Z^{(0)}$ is found in distribution. A method is found of calculating the distribution of $Z^{(i+1)}$ from the distribution of $Z^{(i)}$. Z is then a node-indexed (discrete-time) Markov process. The transition probability from $Z^{(i)}$ to $Z^{(i+1)}$ clearly depends on the probability that node $i+1$ and some attached links are up, i.e. R_N, R_L, and R_S. Section 3.2 and 3.3 will demonstrate how an approach such as this can be used to get upper and lower bounds on fault tolerance. However, this approach cannot be used to derive the desired results exactly, for the following reasons. Since the accessibility of the nodes $\in \{N-S+1,\ldots,N-1\}$ depends on the status of links and nodes throughout the network, there is a correlation between $Z^{(0)}$ and the probability that any link or node is up. Consequently, given $Z^{(0)}$ or the distribution of $Z^{(0)}$, the a posteriori probability that any link or node is up does not equal the known a priori probability. These probabilities are the basis of the transition probabilities of the stochastic process.

Consequently, the stochastic Markov process Z cannot be said to exactly correspond to a physical quantity. Moreover, the distribution of $Z^{(0)}$ is not known in the first place. These complications make an exact calculation of $P(k)$ using the stochastic process Z impractical. Sections 3.2 and 3.3 show that it is possible to make certain simplifying assumptions that allow us to calculate lower and upper bounds for $P(k)$, respectively, in $O(N)$ using the stochastic Markov process Z.

3.2 Lower Bound

A lower bound on $P(k)$ can be found by restricting possible source-destination paths to exclude some possible paths. As previously described, paths can have up to S cycles. Since there are few failure sets that block all single-cycle paths without blocking multiple-cycle paths as well, a good lower bound for $P(k)$ can be achieved by determining the probability that a node is accessible from the source using only single-cycle paths with long links or paths consisting entirely of short links. In single-cycle paths, the only way a path can include a long link out of a node $\in \{N-S+1, \ldots, N-1\}$ is if the path goes from the source to that node using only short links. The probability that such a path can go through any of these $S-1$ nodes does not depend on the status of node i or any of its links for all $i \in \{1, 2, \ldots, N-S\}$. Thus, if only single-cycle paths are considered, a distribution can be found for $Z^{(0)}$ that is independent of the up-down status of nodes $\in \{1, \ldots, N-S\}$ and their associated links. (The special case where the destination is one of the nodes $\in \{N-S+1, \ldots, N-1\}$ requires a minor variation which will be described later in this section.) Thus, Z is a Markovian stochastic process and its transition probabilities are based on the a priori probabilities of link and node failure. Let $\vec{b}$ denote the distribution of $Z^{(0)}$. For this lower bound, the probability that $Z^{(0)}$ is in a given state is simply the joint probability that the nodes $\in \{N-S+1, \ldots, N-1\}$ can be accessed with paths containing only short links, which is easy to calculate. As N grows large with respect to S, the probability that multicycle paths exist decreases, and this lower bound approaches the exact solution.

The single step transition probabilities from $Z^{(i-1)}$ to $Z^{(i)}$ are a function of the probability that node i is up, the probability that the link from node i to node $i-1$ is up, and the probability that the link from node $i-S+1$ modulus N to node i is up. Figure 4 shows the entire state transition probability matrix, $\underline{A}$, for S=2. R_N is the probability that a node is up; R_S is the probability that a short link is up; R_L is the probability that a long link is up. For this matrix it is assumed that failures are independent and identically distributed. If failures are not identically distributed, Z is a non-homogeneous process, and $\underline{A}$ depends on the current node, i. Dependent failures can only be accommodated with this approach if the dependence can be accounted for in the transition matrix. For example, perhaps the fact that node $i-1$ failed makes it more likely that node i also failed. In this case, the probability of node failure would be higher when $Z^{(k)}$ has an X in its last component.

As long as the $S-1$ nodes before the source and the $S-1$ nodes after the destination do not overlap, i.e., $k < N-2S+1$, the distribution of $Z^{(i)}$ is simply $\vec{b}\underline{A}^i$. For k in this range, we now use this fact to derive $P(k)$. We will subsequently return to the case where $k \geq N-2S+1$. To determine whether node k is accessible, it is necessary to examine the state of the nodes $\in \{k, \ldots, k+S-1\}$, since some paths to node k must go from $k-1$ to $k+S-1$ and then back to k through the short links. Thus, node k is accessible via a loop-free path containing at least one long link if and only if $Z^{(k+S-1)}$ has an O in the first component. Let $\vec{V}_o(i)$ be a $1 \times (2^{S+1}-1)$ column vector with one element corresponding to each possible state of Z. The element of $\vec{V}_o(i)$ corresponding to a given state equals 1 if that state has an O in the ith component of the state vector, and otherwise it equals 0. Let $P_L(k)$ be the probability that node k is accessible via a loop-free path that includes

state	OO	OX	XO	XX	O&	X&	&&
OO	$R_N \cdot R_L$	$1-R_N$	0	0	$R_N(1-R_L)$	0	0
OX	0	0	$R_N \cdot R_L$	$1-R_N$	0	$R_N(1-R_L)$	0
XO	0	$1-R_N$	0	0	R_N	0	0
XX	0	0	0	$1-R_N$	0	R_N	0
O&	$R_N R_L R_S$	0	$R_N R_L(1-R_S)$	$1-R_N$	0	$R_N(1-R_L)(1-R_S)$	$R_N(1-R_L)R_S$
X&	0	0	0	$1-R_N$	0	$R_N(1-R_S)$	$R_N \cdot R_S$
&&	0	0	0	$1-R_N$	0	$R_N(1-R_S)$	$R_N \cdot R_S$

Figure 4: State transition probability matrix, $\underline{A}$, for S=2.

a long link. For $k < N-2(S-1)$,

$$P_L(k) = \vec{b}\underline{A}^{k+S-1}\vec{V}_o(1) \tag{2}$$

Node k is accessible by paths using long links and by paths containing only short links. Since these two cases are not disjoint, it is necessary to determine the probability that the node is inaccessible by paths that use long links, given that it can be reached with the path containing only short links. This situation can only occur when there is an & in the element of $Z^{(k+S-1)}$ corresponding to node k, and the $S-1$ nodes before the source are all initially accessible. Analogous to $\vec{V}_o(i)$, let $\vec{V}_\&(i)$ be a state vector such that the component of $\vec{V}_\&(i)$ corresponding to a given state equals 1 if and only if the ith component of that state is a &. $[\overrightarrow{state}]$ is a $1 \times (2^{S+1}-1)$ column vector containing all 0's except in the component corresponding to *state*, which is 1. For $k < N-2(S-1)$,

$$P(k) = \vec{b}\underline{A}^{k+S-1}\vec{V}_o(1) \; + \; [\overrightarrow{\mathrm{OO}\ldots\mathrm{O}}]\underline{A}^{k+S-1}\vec{V}_\&(1)\;(R_S\;R_N)^{N-S-k}R_S \tag{3}$$

When $k \geq N-2(S-1)$, the calculation of $P(k)$ must account for the fact that the $S-1$ nodes before the source overlap the $S-1$ nodes after the destination. A node that is both one of the $S-1$ before the source and one of the $S-1$ after the destination is said to be in the overlap. A path may include a node in the overlap in one of two ways: either near the source, in which case it is accessible from the source using only short links, or near the destination, in which case the destination is accessible to the node using only short links. For a particular failure set, if the only paths from source to destination containing long links include a given node both near the source and near the destination, then the only loop-free path is the one containing only short links. This destination should be considered inaccessible using paths with long links, or else the failure set will be counted twice in $P(k)$. The nodes between the source and the destination in the direction of the short links are divided into two groups: those that are allowed to be near the source in source-destination paths considered in this calculation, and those that can be near the destination. An equal break (to the extent possible) produces the most accurate results. For example, if node 0 is, as usual, the source, and node $N-3$ is the destination, then there are two nodes to divide. Node $N-2$ can only be used near the destination, and $N-1$ only near the source. In this case, node $N-2$ is considered inaccessible near the source, and node $N-1$ is considered inaccessible near the destination.

A node j is made inaccessible before the source by making the probability that $Z^{(0)}$ is in a state in which node j is accessible equal to 0. Let $\vec{b}_{N-j,X}$ be the appropriate vector derived from $\vec{b}$ accordingly. For all the states in which node j is accessible, the corresponding element in $\vec{b}_{N-j,X}$ is 0. For the other states, the element in $\vec{b}_{N-j,X}$ is the same as in $c\vec{b}$, where c is chosen so that the elements of $\vec{b}_{N-j,X}$ sum to 1. Let $\vec{b}_{N-j,XO}$ be the initial state vector derived from $\vec{b}$ by forcing to 0 not only states in which node j is accessible but also the states in which the nodes $\in \{j+1, j+2, \ldots, N-1\}$ are not accessible. This vector is used to calculate the probability that a node is inaccessible from the source with paths that contain long links, but is accessible through the path with only short links. To ignore paths using some of the nodes after the destination, rather than determining whether there is an O or & in the first element of $Z^{(k+S-1)}$, one determines whether there is an O or & in the ith position of $Z^{(k+S-i)}$. For $N - 2(S-1) \geq k$,

$$\begin{aligned} P(k) = \;& \vec{b}_{(N-k+1)div2,X}\; \underline{A}^{(N+k-1)div2}\; \vec{V}_o((k+2S-N+1)div2)\; + \\ & \vec{b}_{(N-k+1)div2,XO}\; \underline{A}^{(N+k-1)div2}\; \vec{V}_{\&}((k+2S-N+1)div2) \cdot \\ & \qquad (R_S\; R_N)^{(N-k)div2}\; R_S \end{aligned} \tag{4}$$

In summary, a lower bound on $P(k)$ is calculated as follows. The transition matrix $\underline{A}$ is determined from R_N, R_L, and R_S. The distribution $\vec{b}$ of $Z^{(0)}$ is found from R_N and R_S based on the assumption that the nodes $\in \{N-S+1, \ldots, N-1\}$ are only accessible if they are accessible via the path of short links. $P(k)$ is calculated for $k < N - 2(S-1)$ using Equation 3. Otherwise, $P(k)$ is calculated using Equation 4.

3.3 Upper Bound

As with the lower bound previously discussed, the correlation between the probability that nodes and links are up and the distribution of $Z^{(0)}$ must be eliminated. This correlation is caused by multicycle paths. To achieve an upper bound, we must calculate $P(k)$ in a way that includes all failure sets in which a path from source to destination exists, and possibly some where no such path exists. A necessary but insufficient condition for a node's accessibility is that the node must be up, and at least one incoming link must be as well. Thus, a simple upper bound can be achieved by assuming that there is a path to a node $\in \{N-S+1, \ldots, N-1\}$ that is up and has at least one functioning incoming node. The $\vec{b}$ corresponding to this assumption is easy to calculate. Moreover, since there is no correlation between the up and down status of any two network elements, there is no correlation between the $Z^{(0)}$ resulting from this assumption concerning accessibility and the transition probabilities of the stochastic process. As with the lower bound, $Z^{(i)}$ is therefore Markovian with transition probabilities that are based on a priori probabilities of node and link failure.

It is possible to improve the upper bound by using a different method for calculating $\vec{b}$. Recall the deterministic case in which we determined the accessibility of all nodes by beginning with a $Z^{(0)}$ in which only the source was up, and using it iteratively to find more accurate $Z^{(0)}$'s. In S iterations, we had exact results. This process can also be done probabilistically rather than deterministically, using the transition matrix $\underline{A}$, to find a distribution for $Z^{(0)}$. After the first iteration, the distribution $\vec{b}$ would indicate the accessibility of the nodes described in $Z^{(0)}$ via single cycle paths that begin with long links and via paths containing only short links. However, in the second cycle, $\vec{b}$ would not describe accessibility via one and two cycle paths in the manner of the deterministic case. $\vec{b}$ at this point is based on some failure sets in which a given node or link is up in the first cycle (or iteration) and down in the second cycle, and vice versa. Consider such a failure set where the up-down status of link or node j changes from the first cycle to the second while we are trying to determine whether or not node $N-1$ is accessible. If element j was down in the first cycle, the resulting accessibility of node $N-1$ can only be improved compared with the case where j's status does not change. If j was up in the first cycle, but was not needed in order to find a path to node $N-1$, then its status in this cycle was irrelevant, and a change in its status from one iteration to the next would not matter. Finally, if j was up in the first iteration and was needed to find a path to $N-1$, then nothing is lost by precluding its use in subsequent iterations, since to do so would cause a loop in the path. Thus, assuming independence of the up-down probabilities in different iterations will result in an upper bound for $P(k)$.

Let $\underline{B}$ be the matrix associated with the transition from $Z^{(N-1)}$ to $Z^{(0)}$, such that node 0 is automatically accessible to the source. The only random element is whether the attached short link is up, so that any nodes marked with an & are also accessible.

$$\vec{b} = \overrightarrow{[\mathrm{X}\ldots\mathrm{XO}]}(\underline{A}^{N-1}\underline{B})^S \tag{5}$$

Unfortunately this requires a great deal of computation. A related upper bound which is somewhat higher but requires significantly less computation can be achieved by allowing an infinite number of iterations. This $\vec{b}$ is calculated by finding an eigenvector.

$$\vec{b}\underline{A}^{N-1}\underline{B} = \vec{b} \tag{6}$$

If there is no overlap between the $S-1$ nodes before the source and the $S-1$ nodes after the destination, calculation of the upper bound is identical to calculation of the lower bound, except that a different $\vec{b}$ is used. Thus for $k < N-2(S-1)$, Equation 3 produces an upper bound.

To complete the analysis, $P(k)$ must be determined for $k \geq N-2(S-1)$. There are two cases: $k > N-S$ and $k \leq N-S$. For $k \leq N-S$, the only problem with the above equation is that the $S-1$ nodes before the source and the $S-1$ nodes after the destination overlap. Consider the case where the overlap contains exactly one node. The paths from source to destination that begin with $S-1$ consecutive short links will go through this node, as will the paths that end with $S-1$ consecutive short links. To consider this node as two different nodes, one before the source and the other after the destination, produces an upper bound. It is never true that a path must go through the node in both places, since such a path would include a loop. If the only loop-free path to the destination for a particular failure pattern is the path containing only short links, and a path does exist that traverses the entire network and includes a node in the overlap twice, then this failure pattern would be counted twice in our basic equation for $P(k)$: once for the path with the loop and once for the path without it.

For the upper bound, it is therefore acceptable to assume that link and node status near the source is independent of that near the destination when calculating the probability that a node is accessible by a path using long links. However, independence cannot be assumed in the second half of the equation which calculates the probability that a node is accessible using only short links but inaccessible using paths containing long links. Thus, for this part of the equation, the state of Z should not be considered at $Z^{(k+S-1)}$. Instead, the $S-1$ nodes before the source are simply excluded from the final state of Z in the same manner we excluded some nodes in the overlap when calculating the lower bound. If there are $i-1$ nodes in the overlap, it must be determined whether there is an & in the ith element of $Z^{(k+S-i)}$ rather than the first element of $Z^{(k+S-1)}$. For $N-2(S-1) \leq k \leq N-S$,

$$P(k) = \vec{b}\underline{A}^{k+S-1}\vec{V}_o(1) \; + \; \overrightarrow{[\mathrm{OO}\ldots\mathrm{O}]}\underline{A}^{N-S}\vec{V}_{\&}(k-N+2S)(R_S\;R_N)^{S-1}R_S \tag{7}$$

If $k > N-S$, then the $S-1$ nodes before the source include the destination, and the $S-1$ nodes after the destination include the source. Because of the latter condition, paths that include the source twice must be explicitly removed from consideration. This is done by using $Z^{(N-1)}$ rather than $Z^{(k+S-1)}$ for the nodes in this range. Because $k+S-1 \geq N$, $Z^{(k+S-1)}$ would inappropriately include information pertaining to the accessibility of the source. The initial state vector, $\vec{b}$, must also be changed from the one used for $k \leq N-S$, because a path to destination node k should not pass through node k along the way; this would form a loop. Thus the probability that $Z^{(0)}$ is in a state where node k is already accessible should be zero. $\vec{b}_{N-k,X}$ and $\vec{b}_{N-k,XO}$ are defined as they were for the lower bound. For $N-S<k$,

$$\begin{aligned} P(k) = \quad & \vec{b}_{N-k,X}\underline{A}^{N-1}\vec{V}_o(k+S+1-N) + \\ & \vec{b}_{N-k,XO}\underline{A}^{k}\vec{V}_{\&}(S)(R_S\ R_N)^{N-k-1}R_S \end{aligned} \tag{8}$$

In summary, an upper bound on $P(k)$ is calculated as follows. First, $\vec{b}$ must be determined. For a simple upper bound, $\vec{b}$ is calculated from R_N, R_S, and R_L assuming that the node is accessible if a node and at least one incoming link are up. For the Eigenvector bound, $\vec{b}$ is determined from Equation 6. An even more accurate upper bound can be found by calculating $\vec{b}$ using Equation 5. Once $\vec{b}$ and $\underline{A}$ are determined, $P(k)$ is calculated using Equation 3, Equation 7, and Equation 8, for $k<N-2(S-1)$, $N-2(S-1) \leq k \leq N-S$, and $N-S<k$, respectively.

4 Results

In order to evaluate our calculated bounds on fault tolerance, we wish to compare them with exact values for f. These values are found by enumerating all possible failure sets. For a given failure set, we use a recursive flooding routine which works as follows. When the routine is called on node i, it declares i accessible. It then examines the two outgoing links from node i and the two associated nodes. If an outgoing link is up, the associated node j is up, and node j is not presently declared accessible, the routine is called on node j. The program begins with no nodes marked accessible by calling the routine on the source. By determining which nodes are accessible for each failure set, as well as the probability that each failure set occurs, the average number of accessible nodes f can be found.

Figure 5 shows the fault tolerance f of an 8-node network with R_N=.9, R_S=.97, and R_L=.93, as a function of S. All failures are assumed to be independent. Both upper bounds, the lower bound, and the exact value of f are shown. For $S \leq 4$, the lower bound is close to the exact solution, more so than the upper bound. The simpler upper bound seems to perform almost as well as the eigenvector-based method, so the improved accuracy of the eigenvector-based method may often not be worth the computation time.

Figure 5: Fault tolerance, f, vs. skip length, S. 8 nodes, R_N=.9, R_S=.97, R_L=.93.

The lower and upper bounds are always exactly correct for $S=1$. This is because the initial vector $Z^{(0)}$ contains only one element which indicates the accessibility of the source node. This node is always accessible by definition, which means that the distribution $\vec{b}$ is known with certainty and the a priori and a posteriori probabilities of failure are equal for all network elements. As S increases, the accuracy of both bounds tends to decrease. This is because, for both upper and lower bounds, approximations are used for the accessibility of the $S-1$ nodes before the source, and for the $2(S-1)$ destination nodes for which there are nodes in the overlap. As S increases, so do the effects of these approximations. The lower bound is also generally more accurate than the upper bound; the inaccuracies in the estimated f caused by ignoring multiple cycle paths are not as great as those caused by assuming some paths exist simply because the destination node and one incoming link are up. This is useful since in practice, it is the lower bound that typically must be guaranteed.

The accuracy of f is especially poor for $S > N/2$, since this overlap problem occurs with every node in the network. If desired, this easily can be remedied by altering the analysis slightly. A backward hop of S is the same as a forward hop of $N - S$. If $S > N/2$, then $Z^{(0)}$ should be an $N - S$ element vector, each element of which indicates the accessibility to the source of the nodes $j \in \{0, 1, \ldots, N - S - 1\}$, i.e., the $N - S$ nodes in the positive direction from the source as opposed to the S nodes in negative direction. Single step state transitions would go in the direction that we have defined as negative, i.e., $Z^{(i)}$ would be a function of the status of node i and its associated links and $Z^{(i+1)}$, as opposed to $Z^{(i-1)}$. The rest of the analysis would proceed analogously. From the figure, it is clear that the shape of the curve for the exact solution is roughly symmetric with respect to $S = N/2$, although fault tolerance is slightly better for $S = N/2 - j$ than $S = N/2 + j$ where $0 < j < N/2$. The evaluation of fault tolerance for networks with $S \geq N/2$ is generally less important, since network performance (throughput and delay) in this region has been shown to be inferior, [8, 9, 10, 11, 12, 13, 14] and fault tolerance is expected to be as well. [8, 9, 12]

An evaluation of the accuracy of the upper and lower bounds would be more meaningful in a network with more than eight nodes. Unfortunately, it is difficult to get an exact solution for larger networks, since an exact calculation of fault tolerance for a network with only nine nodes and potential link and node failures takes about 36 hours of computation on the Sun 3/80 workstations used for this research. To allow the exact calculation of f in a larger network, reliable links are used, thereby reducing the number of elements that can fail from $3N$ to N. Figure 6 shows results for a 20-node network with node failures but no link failures. Even with $S = N/2 - 1$, both the upper and lower bound are accurate to within less than 4%. From the exact curve, it is also clear that $N = S/2 - 1 = 9$ is not optimal with respect to this measure as Hu and Hwang found [33]; fault tolerance is better with any value of S from 3 to 8. In addition, there is little difference between the two upper bounds, indicating that the simpler to calculate would probably suffice in this example. Figure 7 shows the lower and upper bounds on a larger network of 50 nodes with unreliable links as well as nodes. This example shows how fault tolerance increases very quickly with S when S is small, and then plateaus. For the 50-node network, although the lower bound of f peaks at $S = 6$, any value of S such that $4 \leq S \leq 9$ would produce roughly the same results.

Figure 6: Fault tolerance, f, vs. skip length, S. 20 nodes, R_N=.9, R_S=1, R_L=1.

Figure 7: Fault tolerance, f, vs. skip length, S. 50 nodes, R_N=.9, R_S=.97, R_L=.93.

Figure 8 shows the upper and lower bounds of $P(k)$ for the 50-node network with $S = 6$. Except for the nodes $\in \{1, 2, \ldots, S - 1\}$, the variation of $P(k)$ is relatively small, which is a useful characteristic of the network. It means that there are no node pairs for which communication is particularly unreliable relative to the rest of the network, so it is generally not necessary to consider fault tolerance when determining the placement of particular nodes within the network. The primary exception are the nodes $\in \{1, 2, \ldots, S - 1\}$, but the extent of this effect is difficult to evaluate because this is also the region where the lower and upper bounds differ the most. This is because the fact that the lower bound only accounts for single cycle paths has the most impact here. For example, there is only one single cycle path from node 0 to node 1, the one consisting of $S - 1$ short links followed by a long link. However, there are likely to be many multiple cycle paths, since the path ends on the first node of the second cycle. The low fault tolerance of the lower bound in this region also produces damped waves of increasing and decreasing fault tolerance with period S. The other area where the lower and upper bounds differ significantly is near node $N-S$ where approximations were necessary to accommodate the overlap between the $S-1$ nodes before the source and the $S-1$ nodes after the destination.

Figure 8: Probability of reaching node k, $P(k)$. 50 nodes, $S = 6$, R_N=.9, R_S=.97, R_L=.93.

Let us now consider a practical example in which it must be determined in a reasonable length of time which values of S lead to a level of fault tolerance that the network designer considers adequate. Since networks are not typically redesigned every time a node is added or removed, we will assume that some minimum fault tolerance must be guaranteed over a given range. Let N vary from 100 to 125, with R_N=.9, R_S=.97, and R_L=.93. Failures are independent. In Figure 9, the smallest lower bound in this range of N is shown for every S between 1 and 12. It appears that the best selection of S in terms of f would be 7 or 8, but any value of S from 6 to 10 would yield roughly the same fault tolerance. Using the methods of Raghavendra et al., [8, 9, 12] one would predict that the best value of S is 10 or 11. This is close, but judging from Figure 9, setting S to 11 would significantly reduce fault tolerance. Although this is only a lower bound, and thus does not actually prove that f is significantly lower for $S = 11$, the accuracy of the lower bound generally seems to be too good to account for all of the drop in f shown in Figure 9.

Figure 9: Minimum lower bound of fault tolerance within range, f, vs. skip length, S. From 100 to 125 nodes, R_N=.9, R_S=.97, R_L=.93.

Figure 10 shows the time it took to calculate both the upper and lower bounds for all N in this range as a function of S. Computation was performed on a 3 million instructions per second (MIPS) Sun 3/80 workstation without the floating point coprocessor. Even with S=12, the computation for the entire range of N took less than four and a half hours. Larger networks are not a significant problem, since computational complexity of finding lower and upper bounds is linear with respect to N. The complexity of computing bounds is exponential with respect to S, because the cardinality of the matrices is exponential with respect to S. However, in Figure 10, the rate of exponential increase is only about 2.15, presumably because of the inherent sparsity of the matrices which also goes up with S. For comparison, an exact calculation of fault tolerance for a single, 7- node network with given S took over 40 minutes, and computation for an 8- node network took almost four and a half hours. Times for exact computation should go up at a rate of approximately 8^N. Under this assumption, an exact numerical computation of f in a single 125-node network with given S would take about 10^{102} years. Therefore, finding an exact measure of f or $P(k)$ is not feasible in many typical FLBH networks, but bounds can be calculated for most practical networks in a tolerable amount of time.

Figure 10: Time to determine upper and lower bounds vs. skip length, S. From 100 to 125 nodes.

5 Conclusion

In this paper, we have presented an efficient method of calculating upper and lower bounds on fault tolerance in forward-loop backward-hop (FLBH) networks. It enables us to evaluate networks much larger than would be possible through enumeration, and with a direct and meaningful measure of fault tolerance. With the lower bound on fault tolerance, a network designer can determine whether an FLBH network is guaranteed to meet fault tolerance specifications, and if so, with what values of S. In the examples considered here, we found the optimal value of S, based on our lower bound on fault tolerance, to be much smaller than $N/2-1$ as proposed in [33], and slightly smaller than $\sqrt{N}$ as proposed in [8, 9, 12] We also found that, in large networks, fault tolerance is close to optimal for a wide range of values for S.

References

[1] Y. J. Tang and S. G. Zaky, "A Survey of Ring Networks: Topologies and Protocols," in *Proc. IEEE Infocom*, Mar. 1985, pp. 318-25.

[2] M. T. Liu, D. D. Tsay, C. P. Chou, and C. M. Li, "Design of the Distributed Double Loop Computer Network (DDLCN)," *J. Digital Systems*, vol. 5, no. 1-2, pp. 3-37, Spring 1981.

[3] J. M. Peha, "Station Interconnection on Dual Rings and Other Multichannel Networks," in *Proc. 13th IEEE Conf. on Local Computer Networks*, Oct. 1988, pp. 108-13.

[4] R. Rom and N. Shacham, "A Reconfiguration Algorithm for a Fault Tolerant Token Ring Local Area Network," *IEEE Trans. Comput.*, vol. 37, no. 2, pp. 182-89, Feb. 1988.

[5] J. J. Wolf, M. T. Liu, B. W. Weide, and D. P Tsay, "Design Of A Distributed Fault-Tolerant Loop Network," in *Proc. IEEE Intl. Symp. Fault-Tolerant Computing*, June 1979, pp. 17-23.

[6] B. W. Arden and H. Lee, "Analysis of Chordal Ring Network," *IEEE Trans. Comput.*, vol. 30, no. 4, pp. 291-5, Apr. 1981.

[7] D. K. Pradhan and S. M. Reddy, "A Fault-Tolerant Communication Architecture for Distributed Systems," *IEEE Trans. Comput.*, vol. 31, no. 9, pp. 863-9, Sept. 1982.

[8] C. S. Raghavendra, M. Gerla, and D. S. Parker, "Multi-Connected Loop Topologies For Local Computer Networks," in *Proc. IEEE Infocom*, Mar. 1982, pp. 184-190.

[9] C. S. Raghavendra and M. Gerla, "Optimal Loop Topologies For Distributed Systems," in *Proc. 7th IEEE Data Commun. Symp.*, Oct. 1981, pp. 218-23.

[10] C. S. Raghavendra and J. A. Silvester, "A Comparative Study of a Class of Double Loop Network Architectures," in *Proc. IEEE ICC*, May 1984, pp. 149-52.

[11] J. A. Silvester and C. S. Raghavendra, "Analysis and Simulation of a Class of Double Loop Network Architectures," in *Proc. IEEE Infocom*, Mar. 1984, pp. 30-5.

[12] C. S. Raghavendra, M. Gerla, and A. Avizienis, "Reliable Loop Topologies for Large Local Computer Networks," *IEEE Trans. Comput.*, vol. 34, no. 1, pp. 46-54, Jan. 1985.

[13] C. S. Raghavendra and J. A. Silvester, "Double Loop Network Architectures: A Performance Study," *IEEE Trans. Commun.*, vol. 33, no. 2, pp. 185-7, Feb. 1985.

[14] B. J. Jambor and G. W. Eger, "Fault Tolerant High Speed Network for Space Systems," in *Proc. IEEE Milcom*, Oct. 1988, pp. 221-8.

[15] S. Angelopoulos, G. Papadopoulos, and S. Koubias, "A New Double-Loop Computer Network: Its Simulated Performance," in *Proc. IEEE Intl. Phoenix Conf. on Computers and Commun.*, March 1988, pp. 16-18.

[16] A. Grnarov, L. Kleinrock, and M. Gerla, "A Highly Reliable Distributed Loop Network Architecture," in *Proc. IEEE Intl. Symp. Fault Tolerant Computing*, Oct. 1980, pp. 319-24.

[17] O. C. Ibe, "Reliability Comparison of Token-Ring Network Schemes," *IEEE Trans. Reliability*, vol. 41, no. 2, pp. 288-93, June 1992.

[18] L. M. Casey, R. C. Dittburner, and N. D. Gamage, "FXNET: A Backbone Ring for Voice and Data," *IEEE Commun. Mag.*, vol. 24, no. 12, pp. 23-8, Dec. 1986.

[19] D. E. Huber, W. Steinlin, and P. J. Wild, "SILK: An Implementation of a Buffer Insertion Ring," *IEEE J. Sel. Areas Commun.*, vol. 1, pp. 766-74, Nov. 1983.

[20] Khiem Van Le and C. S. Raghavendra, "Fault Tolerant Routing In A Class of Double Loop Networks," in *Proc. IEEE Infocom*, Mar. 1988, pp. 264-73.

[21] Li Layuan, "A Distributed Optimal Local Double Loop Network," in *Proc. IEEE ICCC*, Nov. 1990, pp. 397-403.

[22] P. Sheu, W. Chen, and J. Chiou, "Efficient Fault-Tolerant Routing Algorithms for Forward Loop Backward Hop Networks," in *Proc. IEEE Conf. Local Computer Networks* , Oct. 1991, pp. 408-17.

[23] Y. Cheng, F. K. Hwang, I. F. Ikyildiz, and D. F. Hsu, "Routing Algorithms for Double Loop Networks," *Intl. J. Foundations of Computer Science*, vol. 3, no. 3, pp. 323-31, Sept. 1992.

[24] M. A. Fiol, J. L. A. Yebra, I. Alegre, and M. Valero, "A Discrete Optimization Problem in Local Networks and Data Alignment," *IEEE Trans. Comput.*, vol. 36, no. 6, pp. 702-13, June 1987.

[25] H. F. Wedde and P. H. L. Tjoie, "Long-Term Reconfiguration of Double-Ring Networks Under Real-Time Constraints," in *Proc. IEEE Real-Time Systems Symp.*, December 1987, pp. 104-111.

[26] F. K. Hwang, "Comments on 'Reliable Loop Topologies for Large Local Computer Networks'," *IEEE Trans. Comput.*, vol. 36, no. 3, pp. 383-4, Mar. 1987.

[27] F. K. Hwang and Y. H. Xu, "Double Loop Networks with Minimum Delay," *Discrete Mathematics*, vol. 66, no. 1-2, pp. 109-18, Aug. 1987.

[28] A. Grnarov, L. Kleinrock, and M. Gerla, "A New Algorithm For Network Reliability Computation," in *Proc. NBS Computer Networking Symp.*, Dec. 1979, pp. 17-20.

[29] H. Masuyama, "Fault Tolerant Ability To Multi-Node Failures In Double-Loop Computer Networks," in *Proc. IEEE Infocom*, Mar. 1987, pp. 615-24.

[30] H. Masuyama and T. Ichimori, "Tolerance of Double-Loop Computer Networks to Multi-Node Failures," *IEEE Trans. Comput.*, vol. 38, no. 5, pp. 738-41, May 1989.

[31] J. M. Peha and F. A. Tobagi, "Comments on 'Tolerance of Double-Loop Computer Networks to Multi-Node Failures'," *IEEE Trans. Comput.*, vol. 41, no. 11, pp. 1488-90, Nov. 1992.

[32] W. E. Smith and K. S. Trivedi, "Dependability Evaluation of a Class of Multi-Loop Topologies for Local Area Networks," *IBM J. Res. and Dev.*, vol. 33, no. 5, pp. 511-23, Sept. 1989.

[33] X. D. Hu and F. K. Hwang, "Survival Reliabilities of Double Loop Networks," in *Proc. IEEE Globecom*, Dec. 1990, pp. 674-7.