Brief Contributions

Fault-Tolerant Routing in Distributed Loop Networks

Krishnendu Mukhopadhyaya and Bhabani P. Sinha

Abstract—The ring network is a popular network topology for implementation in local area networks and other configurations. But it has a disadvantage of high diameter and large communication delay. So loop networks were introduced with fixed-jump links added over the ring. In this paper, we characterize some values for the number of nodes for which the lower bound on the diameter of loop networks is achieved. We also give an $O(\delta)$ time algorithm (where δ is the diameter of the graph) for finding a shortest path between any two nodes of a general loop network. We also propose a scheme to find a near optimal path (not more than one over the optimal) in case of a single node or link failure.

Index Terms—Diameter, link-fault, loop networks, network topology, node-fault, routing.

I. Introduction

One very common network topology is the ring. The ring has many attractive properties like simplicity of structure, incremental extensibility, low valency, ease of implementation etc. But it has some drawbacks as well. It is highly vulnerable to faults in the network. Also the diameter of a ring of N processors (nodes), is $\left\lfloor \frac{N}{2} \right\rfloor$ ($\lfloor x \rfloor$ denotes the maximum integer $\leq x$) which leads to large transmission delay. There have been several approaches to bring down the diameter of a ring by adding some more links to it. One such idea, *chordal ring*, was proposed by Arden and Lee [1], where there is one chord from every node of the ring. This is a three-regular graph with diameter $O\left(\sqrt{N}\right)$. Another approach is to use two chords from every node. We define the length of a chord as the distance (along the ring) between the nodes that are joined by the chord. Using this metric, chords are made to be of fixed length. These graphs are four-regular, provided that the chords are not of length $\frac{N}{2}$. These structures are called *double-loop networks* or simply *loop networks*.

Loop networks are special cases of an important class of graphs, called *circulants*. Circulants have been known in graph theory for a long time. According to Davis [2], they were first introduced by Catalan in 1846. A circulant $C_N(s_1, s_2)$ is a graph with N nodes numbered from 0 to $N - 1$ and node i is connected to nodes $(i \pm s_1) \bmod N$ and $(i \pm s_2) \bmod N$. There have been several works on their properties [2], [3], [4].

We consider a set of N nodes labeled $V_0, V_1, \cdots, V_{N-1}$. Each node V_i is adjacent to four other nodes, V_{i+1}, V_{i-1}, V_{i+s}, and V_{i-s}, where s is the length of a chord. Using standard notations [5], let us call this graph $G(N; 1, s)$. Let $d(N; 1, s)$ denote the diameter of $G(N; 1, s)$ and $d(N) = minimum_s \{d(N; 1, s)\}$. Wong and Coppersmith [6] gave a lower bound of $\frac{\sqrt{2N-3}}{2}$ for $d(N)$. Boesch and Wang [4] made the bound tighter to $lb(N) = \left\lceil \frac{\sqrt{2N-1}-1}{2} \right\rceil$, where $\lceil x \rceil$ denotes the minimum integer $\geq x$. However, the lower bound $lb(N)$ may not be achievable for all values of N. For example, Du, Hsu, Li, and Xu [5] showed that for $N = 24$, $d(24) = 4$ with $s = 7$, but $lb(24) = 3$. The graphs whose diameters are equal to $d(N)$ are *optimal* for the given value of N, and those whose diameters are equal to $lb(N)$ are called *tight optimal*. Thus, a graph $G(N; 1, s)$ may be optimal for some s, but may not be tight optimal if $d(N) > lb(N)$. Du et al. gave some classes of values of N, for which the lower bound $lb(N)$ can be achieved. They also gave some other classes, for which the lower bound cannot be achieved; but an optimal choice was given for such graphs. Tzvieli [7] and Bermond and Tzvieli [8] have classified many more values of N for the optimal design of loop networks. A directed variation of the loop network is the FLBH (forward loop backward hop) network. There are arcs from node i to nodes $(i + 1) \bmod N$ and $(i - s) \bmod N$. Some results are available in the literature [9], [10] on the different properties and optimal and quasi-optimal routing in such networks.

About the classification, we suggest some classes of values for N, which achieve the lower bound on the diameter. These classes cover a large class of values of N. They also include many Ns not classified by Du et al. Then, we focus our attention on the problem of routing. Given δ (the diameter of the network), we propose an $O(\delta)$ time algorithm to find a shortest path between any two nodes. We also propose how to find a near optimal path (not more than one over the optimal) in case of a single node or link failure.

II. Optimal Design Criteria

Let $C_N(s, t)$ represent the graph on N nodes labeled $V_0, V_1, V_2, \cdots, V_{N-1}$ such that node V_i is connected to V_{i+s}, V_{i-s}, V_{i+t}, and V_{i-t}. Boesch and Wang [4] have found out that for $N > 6$, $d(N; lb(N), lb(N) + 1) = lb(N)$. We try to use the above result, when it is given that one of the jumps is 1 and we have to minimize the diameter over the other jump of length t.

We shall refer to a link between V_i and V_{i+x} as an x-jump, $x = s$ or t.

LEMMA 1. *If $gcd(N, s) = 1$, then in $C_N(s, t)$ there is a Hamiltonian cycle using only s-jumps.*

PROOF. See [11]. □

LEMMA 2. *If $gcd(N, lb(N)) = 1$ then $C_N(lb(N), lb(N) + 1)$ is equivalent to $G(N; 1, s)$ for some s.*

PROOF. See [11]. □

LEMMA 3. *If $gcd(N, lb(N) + 1) = 1$ then $C_N(lb(N), lb(N) + 1)$ is equivalent to $G(N; 1, s)$ for some s.*

PROOF. See [11]. □

If we combine Lemmas 2 and 3 with Theorem 5 of Boesch and Wang [4], we have the following result.

THEOREM 1. *For $N > 6$, if $gcd(N, lb(N)) = 1$ or $gcd(N, lb(N) + 1) = 1$, then $d(N) = lb(N)$.* □

EXAMPLE 1. Consider the case of $N = 14$ nodes. Here we see that $lb(14) = 3$, i.e., $gcd(N, lb(N)) = 1$. So, by Theorem 1, $G(14; 3, 4)$ has diameter 3 and it can be redrawn as $G(14; 1, s)$ for $s = 6$.

In Fig. 1a, we see $G(14; 3, 4)$ with the 4-jumps shown in broken lines. Here nodes 0, 3, 6, 9, 12, 1, 4, 7, 10, 13, 2, 5, 8, 11, 0 form a Hamiltonian cycle. In Fig. 1b these have been relabeled as 0, 1, 2, ..., 13, 0, respectively. The 4-jumps are converted into 6-jumps.

Using Theorem 1, we get an excellent coverage over the possible values of N, the number of nodes in the network. We have exhaustively searched the optimal designs upto $N = 16{,}000$ and found that the tight optimal designs can be obtained for more than 80% of the values of N by following the scheme of Theorem 1. For $7 \le N \le 5{,}305$, the classes given by Du et al. [5] cover only about 13% of the values of N, whereas Theorem 1 covers about 88.6%; nearly 10% of the values remain unclassified by either scheme. As N increases, the classes defined by Du et al. give lesser coverage.

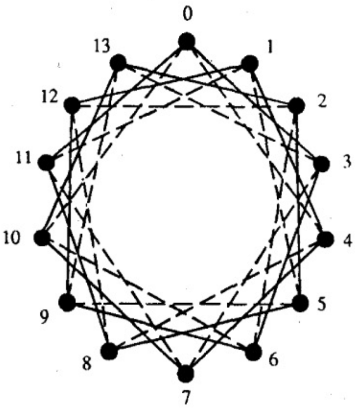

Fig. 1a. The network graph $G(14; 3, 4)$.

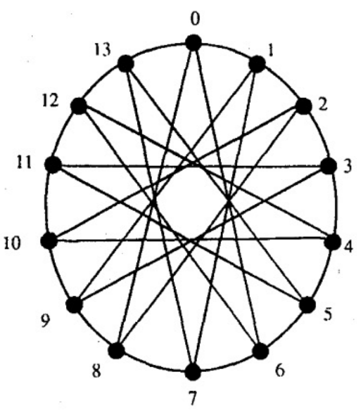

Fig. 1b. The loop network $G(14; 1, 6)$.

III. Shortest-Path Routing

For two nodes with a link connecting them, communication is carried out through that link. In absence of a direct link, the message is transmitted through some intermediate nodes. The number of links traversed in such a path represents the transmission delay. So for any two nodes, it is important to find a path with minimum number of links. Such a path is called a *shortest path*. Note that shortest path between two nodes may not be unique. Here we consider the problem of finding a shortest path from V_i to any arbitrary node V_j. We note that because of the symmetry in the underlying topology it is enough to consider the problem of finding a shortest path from V_0 to an arbitrary node V_u.

For our convenience we shall differentiate between two s-links from V_m, depending on whether they are used to go to V_{m+s} or V_{m-s} by using a + or − sign, respectively. Similarly, we define +1 and −1 links. Consider a path involving w, x, y, and z (all non-negative integers) number of $[+s]$, $[-s]$, $[+1]$, and $[-1]$ links, respectively. Let the endpoints of the path be V_i and V_j. Then, the relation $j = (w.s - x.s + y - z) \bmod N$ holds irrespective of the order in which the links appear in the path. Since we are interested only in the lengths of the paths, we shall denote such a path by $(w)[+s] + (x)[-s] + (y)[+1] + (z)[-1]$.

LEMMA 4. *Let $(w)[+s] + (x)[-s] + (y)[+1] + (z)[-1]$ be a shortest path from V_i to V_j. Then at most one of w and x and at most one of y and z is nonzero.*

PROOF. Let both w and x be nonzero. Without loss of generality, let $w \ge x$. Consider the path $(w - x)[+s] + (0)[-s] + (y)[+1] + (z)[-1]$. As $(w)[+s] + (x)[-s] + (y)[+1] + (z)[-1]$ was a path from V_i to V_j, $(i + w.s - x.s + y - z) \bmod N = j$. Hence, $(w - x)[+s] + (0)[-s] + (y)[+1] + (z)[-1]$ is also a path from V_i to V_j and it is shorter than $(w)[+s] + (x)[-s] + (y)[+1] + (z)[-1]$, which contradicts the hypothesis that $(w)[+s] + (x)[-s] + (y)[+1] + (z)[-1]$ is a shortest path. Similarly, at most one of y and z may be nonzero. □

In view of Lemma 4, at most two of w, x, y, and z can be nonzero. From now on, we shall drop the terms with zero coefficient.

As a consequence of Lemma 4, we note that a shortest path from V_0 to V_u would be using either $(+s, +1)$ or $(+s, -1)$ or $(-s, +1)$ or $(-s, -1)$ links. So, if we find the shortest of the paths of each combination of links, that path will be the required shortest path. We shall discuss in details a method for finding a shortest path using $+s$ and $+1$ links. The case of $+s$ and -1 links would be very similar. The other two cases would be the same as finding a shortest path from V_0 to V_{N-u} using $(+s, -1)$ and $(+s, +1)$ links. From now on, by a $(+s, +1)$-shortest path we shall mean a shortest path among the paths using $+s$ and $+1$ links only.

LEMMA 5. *Let $(w)[+s] + (x)[+1]$ be a $(+s, +1)$-shortest path from V_0 to V_u. Then $x < s$.*

PROOF. If $x \ge s$ then $(w + 1)[+s] + (x - s)[+1]$ is a shorter $(+s, +1)$-path from V_0 to V_u. □

LEMMA 6. *A $(+s, +1)$-shortest path from V_0 to V_u has at least $\lfloor \frac{u}{s} \rfloor$ number of +s-links.*

PROOF. If we use less than $\lfloor \frac{u}{s} \rfloor$ number of +s-links, then we have to use more than s number of +1-links. But a group of s number of +1-links can always be replaced by one $+s$-link. □

Let $S_1 = s$ and W_1 be the cost of reaching the node at S_1 from V_0 using +s-links only. That is $W_1 = 1$. For $u > s$, we can use Lemma 6 and reduce it to a problem of reaching V_u from V_0 with $u < S_1$.

LEMMA 7. *For $u < s$, the number of +s-links in a $(+s, +1)$-shortest path is either zero or at least $W_2 = \lfloor \frac{N}{s} \rfloor + 1$.*

PROOF. Let $(w)[+s] + (x)[+1]$ be a shortest path from V_0 to V_u for some w, $0 < w < W_2 = \lfloor \frac{N}{s} \rfloor + 1$. Then $s \le w.s \le N$. The length of this path is $w + (u + N - w.s) > u$. But $(0)[+s] + (u)[+1]$ is a $(+s, +1)$-path between V_0 and V_u and its length is u. *Contradiction!* □

REMARK. W_2 is the cost of reaching the node at $S_2 = (W_2.s) \bmod N = s\left(\lfloor \frac{N}{s} \rfloor + 1\right) - N$, from V_0 by using +s-links only. Clearly, $S_2 = s - N \bmod s \le s = S_1$.

Now, if $S_1 > u \ge S_2$, and $W_2 < S_2$, then we may use W_2 +s-links from V_0 to reach the node number S_2. We may use groups of W_2 +s-links repeatedly, until we would reduce the problem to one of routing to a node within S_2 distance. If, however, $W_2 > S_2$, then $(+s, +1)$-

shortest path will not have any +s-link at all. Because, if a $(+s, +1)$-shortest path has any +s-link, in view of Lemma 7, it will have at least W_2 +s-links. But we can replace a group of W_2 +s-links by a group of S_2 +1-links and get a shorter path. So we may refine our Lemma 7 as follows.

LEMMA 8. *For $u < s$, the number of +s-links in a $(+s, +1)$-shortest path is*

1) *zero,* [*if* $W_2 > S_2$]
or, 2) *at least* $\left\lfloor \frac{u}{S_2} \right\rfloor W_2$, [*if* $W_2 < S_2$] □

REMARK. If $W_2 = S_2$, then $(+s, +1)$-shortest path may be found in the same way as for $W_2 < S_2$, with the only exception that any group of W_2 +s-links may very well be replaced by W_2 $(= S_2)$ +1-links.

EXAMPLE 2. Consider $G(258; 1, 100)$. Suppose we have to find a shortest path from V_0 and V_{70}. So $N = 258$, $S_1 = s = 100$ and $u = 70$. By Lemma 8, we may use $W_2 = 3$ +s-links to reach node number $S_2 = 3 \times 100 - 258 = 42$. If we take W_2 +s-links once more, we shall reach V_{84}, crossing our destination. Since we cannot use −1-links, reaching V_{70} from V_{84} using +1-links would not generate a shortest path. What we can do is, we can use $3 \times 3 = 9$ +s-links from V_{42} to V_{168} and use one −s-link (i.e., use $9 - 1 = 8$ +s-links from V_{42}) to reach V_{68}.

Now we proceed to generalize the results in Lemmas 6-8. In order to do that, we define some terms:

1) $S_0 = N,\ S_1 = s, \cdots S_k = S_{k-1} \cdot \left(\left\lfloor \frac{S_{k-2}}{S_{k-1}} \right\rfloor + 1 \right) - S_{k-2}$

2) $W_0 = 0,\ W_1 = 1, \cdots W_k = W_{k-1} \cdot \left(\left\lfloor \frac{S_{k-2}}{S_{k-1}} \right\rfloor + 1 \right) - W_{k-2}$

We now describe some properties of the sequences $\{S_i\}$ and $\{W_i\}$.

LEMMA 9. *The sequence $\{S_i\}$ satisfies the following properties.*

1) $S_i \geq 0$.
2) $S_0 \geq S_1 \geq S_2 \geq \cdots$
3) *If $S_i = S_{i+1}$, for some $i \geq 0$, then $S_{i+k} = S_i$ for all $k \geq 0$.*

PROOF.

1) As $S_{k-1} \cdot \left(\left\lfloor \frac{S_{k-2}}{S_{k-1}} \right\rfloor + 1 \right) \geq S_{k-2}$ the result follows from the definition of S_k.
2) Again the result follows from the definition and the observation that $S_{k-1} \cdot \left(\left\lfloor \frac{S_{k-2}}{S_{k-1}} \right\rfloor + 1 \right) \leq S_{k-2} - S_{k-1}$.
3) If for some k, $S_k = S_{k-1}$, then $S_{k-1} = S_k \cdot \left(\left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor + 1 \right) - S_{k-1}$ $= 2.S_k - S_{k-1} = S_k$, and so on. □

LEMMA 10. $W_0 \leq W_1 \leq W_2 \leq \cdots$

PROOF. From definition, $W_0 \leq W_1$. Let $W_0 \leq W_1 \leq \cdots \leq W_{k-1}$. As $S_{k-2} \geq S_{k-1}$ (Lemma 9), $W_k \geq 2.W_{k-1} - W_{k-2} \geq W_{k-1}$. □

LEMMA 11. $(W_i.\ s) \bmod N = S_i$ *for $i = 0, 1, 2, \cdots$*

PROOF. The result is easily verified to hold for $i = 0$ and 1. Let the result be true for $i = k$, for some $k > 0$.

Then, $(W_{k+1}.s) \bmod N$

$$= \left[\left(W_k \cdot \left(\left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor + 1 \right) - W_{k-1} \right) .s \right] \bmod N \quad \text{[From definition of } W_{k+1}]$$

$$= \left(S_k \cdot \left(\left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor + 1 \right) - S_{k-1} \right) \bmod N \quad \text{[By induction hypothesis]}$$

$$= S_{k+1}$$ □

THEOREM 2. *For $1 \leq p < W_i$, $p.s \bmod N \geq S_{i-1}$, $i = 2, 3, \cdots$*

PROOF. For $i = 2$, $W_2 = \left\lfloor \frac{N}{s} \right\rfloor + 1$. Take p such that, $1 \leq p \leq \left\lfloor \frac{N}{s} \right\rfloor$. So, $s \leq p.s \leq N$, i.e., $p.s \bmod N \geq s = S_1$.

Let the result be true for $2 \leq i \leq k$. We shall show that the result holds for $i = k + 1$.

Let $1 \leq p < W_{k+1} = \left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor W_k + (W_k - W_{k-1})$. From induction hypothesis, the result is true for $1 \leq p < W_k$. For $p = W_k$, $p.s \bmod N = S_k$ (Lemma 11). So we have to consider only $W_k < p < W_{k+1}$. Suppose the result is not true. Then there exists $W_k < p < W_{k+1}$ such that $ps \bmod N = v$ for some $0 \leq v < S_k$.

Case I: $p = x.W_k + y$, where $x = \left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor$ and $y < W_k$. So,

$$[p.s + (W_k - y)\ s] \bmod N$$
$$= [(x.W_k + y + (W_k - y))\ s] \bmod N$$
$$= (x + 1)\ S_k$$

$$[(W_k - y)\ s] \bmod N$$
$$= [(x + 1)S_k - ps] \bmod N$$
$$= (x + 1)\ S_k - v$$
$$\leq x.S_k < S_{k-1} \left[\text{as } x < \left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor \right]$$

But this contradicts the induction hypothesis, as $W_k - y \leq W_k$.

Case II: $p = xW_k + y$, where $x = \left\lfloor \frac{S_{k-1}}{S_k} \right\rfloor$ and $y < W_k - W_{k-1}$.

$$[(xW_k + y + W_{k-1})\ s] \bmod N$$
$$= [v + S_{k-1}] \bmod N$$
$$= [v + x.S_k + (S_k - S_{k+1})] \bmod N \quad \text{[From definition of } S_{k+1}]$$

$$[(y + W_{k-1})]s \bmod N$$
$$= v + (S_k - S_{k-1})$$

As $y + W_{k-1} < W_k$, by induction hypothesis, $v + (S_k - S_{k+1}) \geq S_k$, i.e., $v \geq S_{k+1}$.

Now, $[(p + W_k - y).s] \bmod N = (x + 1)\ S_k$.

$[(W_k - y)s] \bmod N = (x + 1)S_k - v \leq (x + 1)S_k - S_{k+1} = S_{k-1}$ [From definition of S_{k+1}]

But this contradicts the induction hypothesis, as $W_k - y < W_k$. □

THEOREM 3. *For $u < S_i$, the number of +s-links in a $(+s, +1)$-shortest path is*

1) zero [if $W_{i+1} \geq S_{i+1}$]
2) at least $\left\lfloor \frac{u}{S_{i+1}} \right\rfloor W_{i+1}$ [if $W_{i+1} \leq S_{i+1}$]
3) $t.W_{i+1},\ 0 \leq t \leq \left\lfloor \frac{u}{S_{i+1}} \right\rfloor$ [if $W_{i+1} = S_{i+1}$]

PROOF. Let $(p)[+s] + (q)[+1]$ be a $(+s, +1)$-shortest path from V_0 to V_i. Since $(0)[+s] + (u)[+1]$ is a $(+s, +1)$-path from V_0 to V_i, $p + q \leq u$. In particular, $q \leq u < S_i$. So, if $p > 0$, $0 < p.s \bmod N < S_i$. By Theorem 2, we must have $p \geq W_{i+1}$.

If $W_{i+1} > S_{i+1}$, then $(+s, +1)$-shortest path will not have any +s-link at all, because instead of using W_{i+1} +s-links, we can use S_{i+1} +1-links to reach the node at S_{i+1} by a shorter path. If the distance of V_u from V_0 is suntil greater than or equal to S_{i+1}, we can repeat the replacement of W_{i+1} +s-links by S_{i+1} +1-links and, repeating this, we can reach a node within S_{i+1} distance from V_u, by using +1-links only. As $S_{i+1} \leq S_i$, again by Theorem 2, the number of +s-links is either zero or at least W_{i+1}; but a direct +1-path has length $u \leq S_{i+1} < W_{i+1}$.

Now consider the case when $S_{i+1} > W_{i+1}$. If we do not use any +s-link, the length of the path is u. By Theorem 2, if we use any

+s-link, we must use at least W_{i+1} +s-links. If $u > S_{i+1}$ then we may use W_{i+1} +s-links and reach S_{i+1}. Even if we take all the remaining links as +1-links, we have path of length $W_{i+1} + (u - S_{i+1}) < u$. Again, if the distance of V_i from S_{i+1} is more than S_{i+1}, we may further reduce the path length by taking groups of W_{i+1} +s-links repeatedly until we reach within a distance of S_{i+1} from V_i. So, in this process, we take $\left\lfloor \frac{u}{S_{i+1}} \right\rfloor W_{i+1} + s-$links. We note that in order to reach V_i we may take some more +*s*-links. But the number of +s-links is at least $\left\lfloor \frac{u}{S_{i+1}} \right\rfloor W_{i+1}$.

If $W_{i+1} = S_{i+1}$ then we can treat the case similar to case 2, but we can replace a group of W_{i+1} +s-links by a group of W_{i+1} (= S_{i+1}) +1-links. We also note that we shall have $W_{i+2} > S_{i+2}$, and hence, after reaching within a distance of W_{i+1} from V_u, we must take only +1-links. □

By repeated application of Theorem 3, we can get a (+*s*, +1)-shortest path as follows.

Algorithm (+*s*, +1)-shortest path :

Step 1: $p := 0; i := 0;$

Step 2: $i := i + 1$; If $W_i > S_i$ then goto step 5;

Step 3: If $u < S_i$ then goto step 2;

Step 4: $p := p + \left\lfloor \frac{u}{S_i} \right\rfloor W_i$; $u := u \bmod S_i$; goto step 2;

Step 5: $q := u$; output $((p)[+s] + (q)[+1])$; stop.

Similarly, we can find the (+*s*, −1)-shortest path from V_0 to V_u. We can also find the (−*s*, +1) and (−*s*, −1)-shortest paths from V_0 to V_u by finding the (+*s*, −1) and (+*s*, +1)-shortest path, respectively, from V_0 to V_{N-u}. So, we can find the shortest paths of all the four types and the shortest of the four will give us a global shortest path between V_0 and V_u. But, such a technique may take $O(N)$ computational steps in the worst case. In order to reduce the time complexity, we change the Step 2 of the above algorithm to

Step 2: $i := i + 1$; if $i > \delta$ then goto step 5; if $W_i > S_i$ then goto step 5;

Let us call this modified algorithm $A_{+,+}$.

THEOREM 4. *If the (+s, +1)-shortest path yields the global shortest path, the algorithm $A_{+,+}$ outputs a global shortest path.*

PROOF. We note that $W_\delta \geq \delta$. But, since the (+*s*, +1)-shortest path is a global shortest path the length of the path is at most δ. So we need not consider including a bunch of more than δ +s-links. □

Since, one of the shortest paths has to be the global shortest path, we can stop each of the algorithms after at most δ executions of the loop and take the shortest of the four paths and yet guarantee that we shall get the global shortest path.

IV. Routing Under Fault

In this section, we consider the problem of routing when one of the nodes (or links) is faulty. We note that the paths as we have defined, specifies only the number of links used for different types of links. It says nothing about the order in which they are traversed. The question is: *Can we always bypass the faulty node (link) by some ordering of the links traversed?* In some cases, we can bypass the faulty node (link). The result is stated in the following theorem.

THEOREM 5. *The links of any shortest path $(p)[+s] + (q)[+1]$ with $p, q > 0$, can always be ordered in a way such that it does not pass through a specified node.*

PROOF. Without loss of generality, let the path be $(p)[+s] + (q)[+1]$. We also assume that one of the endpoints of the path is V_0. First we consider the following realization R of the path, where we traverse all the p +s-links and then the q +1-links. If the faulty node V_f is not on R, then R gives us the path bypassing the faulty node. Suppose V_f is a node on R.

Case I: $f = z.s \bmod N$, $0 < z \leq p$. We consider another realization R' of the path $(p)[+s] + (q)[+1]$, where we first traverse a +1-link, then p +s-links and lastly the remaining $(q - 1)$ +1-links. We claim that V_f is not a node on R'. Suppose V_f is a node on R'. The segments of R and R' from V_0 to V_f must have equal length. Otherwise, we can replace the longer segment by the shorter one to get a shorter path. Note that a typical node on R' is V_i, where $i = (t.s + 1) \bmod N$, $0 \leq t \leq p$, or $i = (p.s + j) \bmod N$, $2 \leq j \leq q$.

Subcase Ia: $f = (t.s + 1) \bmod N$, $0 \leq t \leq p$.
From the equality of path lengths from V_0 to V_f, we have, $z = t + 1$. So, $f = (t + 1)s \bmod N$.
$(t.s + 1) \bmod N = (t + 1)\, s \bmod N = (t.s + s) \bmod N$
$(s - 1) \bmod N = 0$, *Contradiction!*

Subcase Ib : $f = (p.s + j) \bmod N$, $2 \leq j \leq q$.
Again, from the equality of path lengths from V_0 to V_f, we have $z = p + j$. But we know that $z \leq p$ and $j \geq 2$. *Contradiction!*

Case II: $f = (p.s + j) \bmod N$, $1 \leq j \leq q - 1$. Here we take R' to be the realization of the path where first we traverse $(j + 1)$ +1-links, then the p +s-links and then the rest of the $(q - j + 1)$ +1-links. A typical node on R' is V_i, where $i = t$, $0 \leq t \leq j + 1$, or $i = (t.s + j + 1) \bmod N$, $1 \leq t < p$ or $i = (p.s + t) \bmod N$, $j + 2 \leq t \leq q$.

Subcase IIa: $f = t$, $0 \leq t \leq j + 1$.
Again, from the equality of path lengths from V_0 to V_f, we have $p + j = t$. So, $p + j = (p.s + j) \bmod N$ $(s - 1) \bmod N = 0$, *Contradiction!*

Subcase IIb: $f = (t.s + j + 1) \bmod N$, $1 \leq t < p$.
From the equality of path lengths from V_0 to V_f, we have, $p + j = t + j + 1$. So, $((p - 1)\, s + j + 1) \bmod N = (p.s + j) \bmod N$. $(s - 1) \bmod N = 0$. *Contradiction!*

Subcase IIc: $f = (p.s + t) \bmod N$, $j + 2 \leq t \leq q$.
From the equality of path lengths from V_0 to V_f, we have, $p + j = p + t$ or $j = t$. *Contradiction!* □

For nodes which do not have a mixed (using both types of links) shortest path, length of a shortest path in faulty situation will be at least one more than that in the fault-free case. We may, however, add one each of + and − links of the type of link not used in the fault-free shortest path and this path may be at most one link longer than a shortest path in the faulty case.

V. Conclusion

In this paper, we have classified many values of N for which tight loop networks exist. Though this gives a much wider coverage than the classes defined by Du et al., some values of N remain yet to be classified. Some further works on the classification have also been reported in the literature [7], [8]. We also give an algorithm to find a shortest path between any pair of nodes and a near optimal routing in the presence of single node or link failure.

For improvement over the ring we have considered the addition two chords from every node. One may consider a further generalization where there are $2k$ chords from every node. Let $G(N;\, 1, s_1, s_2, \ldots, s_k)$ denote the supergraph of ring where from each node V_i there are links to the nodes $V_{i\pm1}, V_{i\pm s_1}, V_{i\pm s_2}, \cdots, V_{i\pm s_k}$. Below, we list some of the problems which remain to be solved.

1) Deriving an analytical formula for the diameter of $G(N; 1, s)$,
2) Design of optimal loop networks for all values of N,
3) Optimal routing under single as well as multiple faults,
4) Analysis of generalized loop networks $G(N; 1, s_1, s_2, ..., s_k)$, etc.

REFERENCES

[1] B.W. Arden and H. Lee, "Analysis of chordal ring," *IEEE Trans. Computers*, vol. 30, pp. 291-292, 1981.

[2] P.J. Davis, *Circulant Matrices*. New York: John Wiley and Son, 1979.

[3] F.T. Boesch and R. Tindell, "Circulants and their connectivities," *J. Graph Theory*, vol. 8, pp. 487-499, 1984.

[4] F.T. Boesch and J.-F. Wang, "Reliable circulant networks with minimum transmission delay," *IEEE Trans. Circuits and Systems*, vol. 32, pp. 1,286-1,291, 1985.

[5] D.-Z. Du, D.-F. Hsu, Q. Li, and J. Xu, "A combinatorial problem related to distributed loop networks," *Networks*, vol. 20, pp. 173-180, 1990.

[6] C.K. Wong and D. Coppersmith, "A combinatorial problem related to multimodule memory organizations," *J. ACM*, vol. 21, pp. 392-402, 1974.

[7] D. Tzvieli, "Minimal diameter double-loop networks. I. Large infinite optimal classes," *Networks*, vol. 21, pp. 387-415, 1991.

[8] J.-C. Bermond and D. Tzvieli, "Minimal diameter double-loop networks. II. Dense optimal families," *Networks*, vol. 21, no. 1, pp. 1-10, Jan. 1991.

[9] C.S. Raghavendra, M. Gerla, and A. Avizienis, "Reliable loop topologies for large local computer networks," *IEEE Trans. Computers*, vol. 34, no. 1, pp. 46-55, Jan. 1985.

[10] C.S. Raghavendra and J.A. Silvester, " A survey of multiconnected loop topologies for local computer networks," *Computer Networks and ISDN Systems*, vol. 11, pp. 29-41, 1986.

[11] K. Mukhopadhyaya and B.P. Sinha, "Optimal design and routing of distributed loop networks," *Proc. IEEE Int'l Symp. Circuits and Systems*, pp, 1,021-1,024, Aug. 11-14, 1991.

Modular Asynchronous Arbiter Insensitive to Metastability

M. Valencia, M.J. Bellido, J.L. Huertas, A.J. Acosta, and S. Sánchez-Solano

***Abstract*—The purpose of this paper is to present a novel modular N-user asynchronous arbiter circuit which is insensitive to metastable operation (i.e., the new arbiter cannot fail because of metastability), operating asynchronously and incorporating a modular architecture. A 1.5μm CMOS prototype arbiter has been designed and tested. Laboratory tests demonstrate the arbiter operates correctly.**

***Index Terms*—Asynchronous circuits, arbitration, metastability, modular design, Q-flop resolver.**

I. INTRODUCTION

An N-user arbiter is a module designed to control access to a single common resource through the arbitration of contending request signals coming from N systems. This is a circuit component which has received a lot of attention for many years. Its key role in the control mechanism of asynchronous computer interactions has justified the proposal of many solutions for the problem of designing efficient arbiter implementations. This problem is still open for new ideas. Many different aspects have been addressed by the reported arbiters: programmability, priority schemes, hardware simplicity, modularity, resource granting speed. They have been some of the topics covered by researches [1], [2], [3]. One of the most relevant problems in arbiters is caused by the indetermination due to metastability [4], [5], [6].

The nondeterministic evolution of metastability may provoke diverse arbitration failures (the resource is not granted to any requesting device, the priority scheme is violated, the resource is granted to more than one user simultaneously, etc). Arbitration failures due to metastability have been reported (VMEbus controller and Multibus II [7]). Arbiters design must include, at least, a previous analysis of failure probability when entering to its metastable state. When the N users operate asynchronously (i.e., without a common clock), the metastable operation brings, as a consequence, indetermination in both output logic levels and time elapsed to reach a stable and correct state. However, there are some circuits (such those presented in [8]) which detect metastable operation and generate well-defined outputs at a logic level, even during metastability. This means that using circuit techniques, indetermination due to metastability is restricted to resolution time to the stable state, and arbitration failures (in a logical sense) can be avoided.

The purpose of this paper is to present a novel arbiter circuit insensitive to metastable operation, in the sense that it does not provoke any arbitration failure (i.e., the arbiter grants the resource according to the priority scheme and it guarantees mutually-exclusive access). Furthermore, the proposed arbiter incorporates a N-user modular architecture, operating asynchronously. The arbiter samples the request lines even if no request is present and grants the resource on a fix priority policy.

The paper is organized into five more sections. Section II deals with the overall structure of the new arbiter formed by interconnecting a few basic modules; Section III discusses the hardware implementation of these modules, while Section IV considers the essential issue of control signals and timing. In Section V, we give practical results validating the approach. Finally, we draw some conclusions in Section VI.

Manuscript received Nov. 23, 1992; revised Dec. 14, 1994.

The authors are with the Analog Design Department, Centro Nacional de Microelectronica, Sevilla, Spain; e-mail: bellido@cnm.us.es.

To order reprints of this article, e-mail: transactions@computer.org, and reference IEEECS Log Number C95135.