アルゴリズム 46−12
（1995. 7. 21）

ローテータグラフにおけるノンアダプティブな耐故障ファイル転送

濱田幸弘　鮑 豊　梅 傲寒　五十嵐善英

群馬大学 工学部 情報工学科
〒376 桐生市天神町 1-5-1

有向グラフ $G=(V,E)$ は，$V=\{a_1a_2\cdots a_n \mid a_1a_2\cdots a_n$ は $1,2,\cdots,n$ の置換$\}$ かつ $E=\{(a_1a_2\cdots a_n, b_1b_2\cdots b_n) \mid$ ある $2\le i\le n$ に対して，$b_1b_2\cdots b_n=a_2\cdots a_ia_1a_{i+1}\cdots a_n\}$ のとき，n 次ローテータグラフと呼ばれる．n 次ローテータグラフの，相異なるノードのどの組に対しても，$n-1$ 本の互いに素なパスが構成でき，それぞれの長さが $2n$ より短いことを示す．これらの互いに素なパスを用いたノンアダプティブな耐故障ファイル転送アルゴリズムを提案する．

Nonadaptive Fault-Tolerant File Transmission in Rotator Graphs

Yukihiro Hamada, Feng Bao, Aohan Mei and Yoshihide Igarashi

Department of Computer Science
Gunma University, Kiryu, 376 Japan

A directed graph $G=(V,E)$ is called the n-rotator graph if $V=\{a_1a_2\cdots a_n \mid a_1a_2\cdots a_n$ is a permutation of $1,2,\cdots,n\}$ and $E=\{(a_1a_2\cdots a_n, b_1b_2\cdots b_n) \mid$ for some $2\le i\le n$, $b_1b_2\cdots b_n=a_2\cdots a_ia_1a_{i+1}\cdots a_n\}$. We show that for any pair of distinct nodes in the n-rotator graph, we can construct $n-1$ disjoint paths, each length $<2n$, connecting the two nodes. We propose a nonadaptive fault-tolerant file transmission algorithm which uses these disjoint paths.

1 Introduction

Graphs can be conveniently used to represent complex networks consisting of processors and communication links. In the graph models of processor networks, the nodes represent the processors and the edges represent the communication links. The topology of a processor network plays an important role in the overall efficiency and reliability of communication in the system. We assume that all the nodes in the graph are synchronized with a global clock.

The rotator graphs were proposed by Corbett[1] as interconnection network models. The n-rotator graph is a directed regular graph with $N=n!$ nodes and $(n-1)n!$ links. Both the indegree and the outdegree of each node are $n-1=O(\log N/\log\log N)$. The rotator graphs are not bounded degree graphs, but the n-rotator graph has relatively small diameter $n-1$ compared with other popular networks, e.g. hypercubes, butterfly networks and shuffle-exchange networks. Corbett [1] showed the uniqueness of the shortest path between any two nodes and the existence of a Hamiltonian cycle in the n-rotator graph. He also showed that the n-rotator graph remains strongly connected even if $n-2$ faulty nodes exist [1].

In the case of file transmission, we can send a file along a single path, and retransmit along a different path in case of failure. Such a transmission scheme is said to be adaptive. In general, an adaptive file transmission entails a loss in time whenever failures are encountered. A transmission method propose by Yamakawa *et al.*[5] for the n-rotator graph is adaptive, and both the time and communication complexities of the method are $O(n^2)$ when the number of faulty links in the network is less than n. An alternative method is nonadaptive file transmission. That is, we can create multiple copies of the file, and send a copy along each of disjoint paths connecting the source node and the destination node. This will result in increase of communication load, but it will be good at transmission time and reliability.

In this paper we propose a nonadaptive fault-tolerant algorithm for file transmission in rotator graphs, and analyze its time and communication complexities. For any pair of distinct nodes, we can construct $n-1$ disjoint paths, each length $< 2n$, connecting the two nodes. Our file transmission algorithm uses these disjoint paths. The size of each message through any link in a single step is assumed to be $O(n \log n)$. When the size of the file is $O(n \log n)$, the time and communication complexities of the file transmission algorithm are $O(n)$ and $O(n^2)$, respectively. For the case where the size of the file is larger than $O(n$ log $n)$, the file is transformed into small pieces and then the Rabin's Information Dispersal Algorithm (IDA) is used[3, 4]. The time and communication complexities of the file transmission algorithm for large files are also analyzed. Furthermore, We estimate the reliability of the file transmission algorithm when failed links are randomly distributed in the n-rotator graph.

2 Preliminaries

Let $a_1a_2\cdots a_n$ be a permutation of n symbols $1, 2, \cdots, n$. For an integer $2 \le i \le n$ and a permutation $a_1a_2\cdots a_n$, a rotate operation is defined as $R_i(a_1a_2\cdots a_n) = a_2a_3\cdots a_ia_1a_{i+1}\cdots a_n$. Note that $R_n(a_1a_2\cdots a_n) = a_2a_3\cdots a_na_1$. For any integer $2 \le l \le n$ and any permutation v of $1, 2, \cdots, n$, $R_l^0(v)$ denotes v, and $R_l^{k+1}(v)$ denotes $R_l(R_l^k(v))$ for any integer $k \ge 0$. A directed graph $G = (V, E)$ is called the n-rotator graph if $V = \{a_1a_2\cdots a_n \mid a_1a_2\cdots a_n$ is a permutation of $1, 2, \cdots, n\}$ and $E = \{(u, v) \mid u, v \in V$ and $v = R_i(u)$ for some $i\}$. From the definition of the n-rotator graph $G = (V, E)$, $|V| = n!$ and $|E| = (n-1)n!$. The 3-rotator graph is shown in Figure 1. The n-rotator graph can be inductively

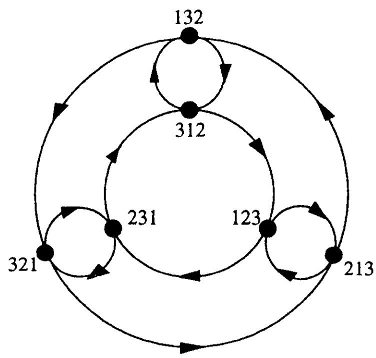

Figure 1: The 3-rotator graph.

constructed in the following way. Suppose that the $(n-1)$-rotator graph has been constructed. Make n

copies G_1, G_2, $\cdots$, G_n of the $(n-1)$-rotator graph. For each $1 \leq i \leq n$, append symbol n to the right end of the label of each node of G_i, and exchange i and n in the modified label of each node. For each node $a_1a_2\cdots a_n$, add a directed link $(a_1a_2\cdots a_n,\ a_2\cdots a_na_1)$. Then the resultant graph is the n-rotator graph.

A transmission from $a_1a_2\cdots a_n$ to $b_1b_2\cdots b_n$ can be executed in the same fashion as a transmission from $f(a_1)f(a_2)\cdots f(a_n)$ to $12\cdots n$, where $f(b_k) = k$ for each $k(1 \leq k \leq n)$. Hence, it is sufficient to consider transmission from each node to the destination $12\cdots n$. Consider a transmission from $s = a_1a_2\cdots a_n$ to $t = 12\cdots n$ $(s \neq t)$. Let i be the largest subscript such that $a_i > a_{i+1}$. Then the subsequence $a_1a_2\cdots a_i$, the subsequence $a_{i+1}a_{i+2}\cdots a_n$ and a_i are called the unsorted prefix, the sorted suffix, and the boundary symbol of s, respectively. The shortest path from s to t is unique. It is obtained by successively inserting the first symbol of a current node in a position so that the length of the sorted suffix increases by one each time. The shortest path from 1643257 to 1234567 in the 7-rotator graph is shown in Figure 2. The distance from s to t equals to the length of the unsorted prefix of s. This implies that the diameter

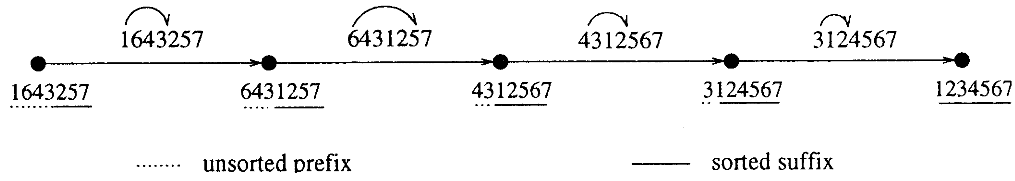

Figure 2: The shortest path from 1643257 to 1234567.

of the n-rotator graph is $n-1$. Corbett[1] proposed an algorithm for transmitting a message from a node to a node in the n-rotator graph with no faults. For any pair of nodes, the algorithm uses the shortest path connecting the two nodes.

A path, $s, R_{i_1}(s), R_{i_2}(R_{i_1}(s)), \cdots, R_{i_m}(\cdots(R_{i_2}(R_{i_1}(s)))\cdots)$, can be denoted by a pair of the node s and the sequence $i_1i_2\cdots i_m$. That is, the sequence $i_1i_2\cdots i_m$ together with s denotes the path specified above. If s is clear from the context, we may denote the path by $i_1i_2\cdots i_m$. Paths connecting a pair of nodes are said to be disjoint if they have no common nodes except for the start node and the destination node. The efficiency of a file transmission algorithm is measured by its time and communication quantities necessary to complete the transmission.

3 Construction of Disjoint Paths

3.1 Paths from $1a_2\cdots a_n$ to $12\cdots n$

In this subsection, we show how to construct $n-1$ disjoint paths, each length $< 2n$, from node $1a_2\cdots a_n$ to node $12\cdots n$. The following two lemmas are immediate.

Lemma 1 *For any node $v \neq 12\cdots n$, the boundary symbols of the nodes except for $12\cdots n$ on the shortest path from v to $12\cdots n$ are identical.*

Lemma 2 *For any nonnegative integers k, k' and $2 \leq i < j \leq n$, $l \geq i$, and any node v of the n-rotator graph, path $R_i(v), R_l(R_i(v)), \cdots, R_l^k(R_i(v))$ and path $R_j(v), R_l(R_j(v)), \cdots, R_l^{k'}(R_j(v))$ are node disjoint.*

Lemma 3 *Let $1a_2 \cdots a_i$ and $a_{i+1} \cdots a_n$ be the unsorted prefix and the sorted suffix of node s, respectively. Then the boundary symbols of the following $n-1$ nodes are distinct, and there are node disjoint paths, excepting s, each length $\leq i-1$, from s to these $n-1$ nodes:*

$$a_2a_3 \cdots a_{i-1}a_i1a_{i+1} \cdots a_n,\ a_2a_3 \cdots a_{i-1}a_ia_{i+1}1a_{i+2} \cdots a_n,\ \cdots,\ a_2a_3 \cdots a_{i-1}a_ia_{i+1} \cdots a_n1,$$
$$a_3a_4 \cdots a_{i-1}a_ia_21a_{i+1} \cdots a_n,\ a_4a_5 \cdots a_{i-1}a_ia_2a_31a_{i+1} \cdots a_n,\ \cdots,\ a_ia_2 \cdots a_{i-1}1a_{i+1} \cdots a_n.$$

Proof: The boundary symbols of the $n-1$ nodes given in the lemma are $a_i, a_{i+1}, \cdots, a_n, a_2, a_3, \cdots, a_{i-1}$. These are obviously distinct.

Consider the $n-1$ adjacent nodes from s. Among these adjacent nodes, $a_2a_3 \cdots a_{i-1}a_i1a_{i+1} \cdots a_n$, $a_2a_3 \cdots a_{i-1}a_ia_{i+1}1a_{i+2} \cdots a_n, \cdots, a_2a_3 \cdots a_{i-1}a_ia_{i+1} \cdots a_n1$ are contained in the set of nodes given in the lemma. We choose the links between s and these nodes as the paths connecting s and these nodes. Then the length of each of these paths is 1. For each $2 \leq k \leq i-1$, choose the shortest path from $a_2 \cdots a_k1a_{k+1} \cdots a_n$ to $a_{k+1} \cdots a_ia_2 \cdots a_k1a_{i+1} \cdots a_n$, and connect the path to the link between s and $a_2 \cdots a_k1a_{k+1} \cdots a_n$. Then the length of th path from s to $a_{k+1} \cdots a_ia_2 \cdots a_k1a_{i+1} \cdots a_n$ is $k+1$. Hence, the length of each path constructed in this way is at most i.

As for the path from s to $a_{k+1} \cdots a_ia_2 \cdots a_k1a_{i+1} \cdots a_n$ $(2 \leq k \leq i-1)$, the immediately left symbol of symbol 1 in the label of each node on the path, excepting s, is a_k (when 1 is the leftmost symbol, the immediately left symbol of 1 means the symbol at the ith position). This means that the $n-1$ paths constructed in this way are node disjoint, excepting s. □

Theorem 1 *There are $n-1$ node disjoint paths, each length $\leq 2n-3$, from $1a_2 \cdots a_n$ to $12 \cdots n$.*

Proof: Consider the shortest path from each node given in Lemma 3 to $t = 12 \cdots n$. From Lemma 1 the boundary symbols of the nodes, excepting t, on the shortest path are identical. For each node, excepting t, on the shortest path, the immediate left symbol of 1 is the boundary symbol of the node. We construct a path from $1a_2 \cdots a_n$ to t by connecting the shortest path to the corresponding path given in the proof of Lemma 3. Then from Lemma 3 the $n-1$ paths from $1a_2 \cdots a_n$ to t constructed in this way are node disjoint.

Suppose that the boundary symbol of $1a_2 \cdots a_n$ is a_i. Then the length of the shortest path from any node given in Lemma 3 to t is $i-1$. Hence, from Lemma 3 the length of each path from $1a_2 \cdots a_n$ to t constructed in this way is at most $i-1+i = 2i-1$. Since $i \leq n-1$, the length of each of these paths is at most $2n-3$. □

3.2 Construction for a General Case

For a general case the construction of $n-1$ node disjoint paths from the source node $s = a_1a_2 \cdots a_n$ to the destination node $t = 12 \cdots n$ is more complicated. However, the basic idea of the construction is similar to the case of $s = 1a_2 \cdots a_n$, although for a general case a more careful case analysis is required. As in the previous subsection, for the construction in a general case we also use $n-1$ intermediate nodes such that their boundary symbols are distinct. These nodes are called *the 2nd intermediate destination, the 3rd intermediate destination, $\cdots$, and the nth intermediate destination.* For each $2 \leq r \leq n$, the path from s to the rth intermediate destination is called *the rth route.*

The construction of the rth route is described by the following procedure **Mk_route**$(r, a_1a_2 \cdots a_n)$. If the procedure is called for $r := 2$ to n, then $n-1$ node disjoint routes from s to all the intermediate destinations are constructed. The procedure consists of three parts, Part I, Part II and Part III, and for each r only one part is used. According to the relative position of 1 in $a_1a_2\cdots a_n$ to the position of the boundary symbol of $a_1a_2\cdots a_n$, the procedure decides which part should be used. In the procedure, *Queue* is used to store the rth route where a sequence of integers is stored (note that integer i means rotate operation R_i). Note that Part I is the case of $a_1 = 1$, and that the case was discussed in the previous subsection.

```
Mk_route (r, a1 a2 ··· an)
{calculates the rth route from a1 a2 ··· an
in Queue for a given r(2 ≤ r ≤ n)}
begin
  Let k be the subscript such that ak = 1;
  Let i be the subscript of the boundary symbol
  of a1 a2 ··· an;
  Let j be the subscript such that j ≥ i
  and that aj < a1 < aj+1
  (a1 < aj+1 holds whenever aj+1 exists);
  Queue := nil;
  if k = 1
    then begin {Part I (k = 1)}
           Add r to Queue;
           if r < i
             then for x := 1 to r do
                    Add i to Queue;
    end
    else if k = i + 1
           then begin {Part II (k = i + 1)}
                  Add r to Queue;
                  Case r of
                    r < i : for x := 1 to r - 1 do
                              Add i to Queue;
                    i < r < j : begin
                              for x := 1 to i - 1 do
                                Add r - 1 to Queue;
                              Add r + 1 to Queue;
                    end;
                  end;
           end
           else begin {Part III (1 < k < i)}
                  Add r to Queue;
                  Case r of
                    r < k : begin
                              for x := 1 to r - 1 do
                                Add i to Queue;
                              for x := 1 to k - r do
                                Add i - 1 to Queue;
                              Add i to Queue;
                    end;
                    k < r < i : begin
                              for x := 1 to k - 1 do
                                Add i to Queue;
                              for x := 1 to r - k do
                                Add i - 1 to Queue;
                    end;
                    r = i : begin
                              for x := 1 to k - 2 do
                                Add i - 1 to Queue;
                              Add i to Queue;
                    end;
                    i < r ≤ j : begin
                              for x := 1 to k - 2 do
                                Add r - 1 to Queue;
                              Add r to Queue;
                              for x := 1 to r - k do
                                Add r - 1 to Queue;
                    end;
                  end;
           end;
  Return Queue;
end.
```

Lemma 4 *Any two boundary symbols of the $n-1$ intermediate destinations determined by the $n-1$ calculated routes are distinct. Furthermore, any two of the $n-1$ paths along the calculated routes are disjoint.*

The proof of this lemma is omitted in this version due to the space limitation. We only give Table 1 and Table 2 that show the correspondence between value r and the rth intermediate destination for Part II and Part III, respectively. In both the tables, i, j and k denote the subscripts defined in the procedure Mk_route.

Since any two boundary symbols of the $n-1$ intermediate destinations are distinct, any two shortest paths from the intermediate destinations to t are disjoint. For each $r(2 \le r \le n)$, we define an *rth path* from s to t as the path from s to the rth intermediate destination along the rth route followed

Table 1: The rth intermediate destination and its boundary symbol for **Part II**($k = i + 1$).

r	the rth intermediate destination	the boundary symbol
$r < i$	$a_1 a_{r+1} \cdots a_i a_2 \cdots a_r a_{i+1} \cdots a_n$	a_r
$r = i, r \geq j$	$a_2 \cdots a_r a_1 a_{r+1} \cdots a_n$	$a_1(r = i), a_i(r = j), a_r(r > j)$
$i < r < j$	$a_{i+2} \cdots a_r a_2 \cdots a_i a_1 a_{r+1} a_{i+1} a_{r+2} \cdots a_n$	a_{r+1}

Table 2: The rth intermediate destination and its boundary symbol for **Part III**($1 < k < i$).

r	the rth intermediate destination	the boundary symbol
$r < k$	$a_{k+1} \cdots a_i a_2 \cdots a_{r-1} a_1 a_{r+1} \cdots a_{k-1} a_r a_k a_{i+1} \cdots a_n$	a_r
$k < r < i$	$a_1 a_{r+1} \cdots a_i a_2 \cdots a_{k-1} a_{k+1} \cdots a_r a_k a_{i+1} \cdots a_n$	a_r
$r = i$	$a_{k+1} \cdots a_i a_2 \cdots a_{k-1} a_1 a_k a_{i+1} \cdots a_n$	a_1
$i < r \leq j^{\dagger}$	$a_2 \cdots a_{k-1} a_1 a_{k+1} \cdots a_r a_k a_{r+1} \cdots a_n$	a_r
$r = k, r > j$	$a_2 \cdots a_r a_1 a_{r+1} \cdots a_n$	$a_i(r = k), a_r(r > j)$

† If $i = j$ then this case is empty.

by the shortest path from the rth intermediate destination to t. We call the path from s to the rth intermediate destination along the rth route, *the rth preceding path.* We call the shortest path from the rth intermediate destination to t, *the rth subsequent path.* So as to establish $n-1$ disjoint paths from s to t, we need to show the following lemma.

Lemma 5 *For distinct r and r', the rth preceding path and the r'th subsequent path are disjoint.*

Lemma 6 *Let L be the maximum length of the $n-1$ disjoint paths from s to t. Then,*

$$L = \begin{cases} 2n-3 & \text{if } k = 1 \\ 2n-1 & \text{if } k = i+1 \text{ or } 1 < k < i, \end{cases}$$

where i and k are the subscripts defined in the procedure Mk_route.

From Lemma 4, Lemma 5 and Lemma 6, we have the following theorem.

Theorem 2 *There exist $n-1$ node disjoint paths with length shorter than $2n$ between any two nodes in the n-rotator graph.*

4 Reliability of Nonadaptive Routings

In the previous section, we have constructed $n-1$ disjoint paths with length shorter than $2n$ between any two nodes in the n-rotator graph. In this section, we consider point-to-point routings in a faulty n-rotator graph. We assume that all nodes are healthy, and that each node knows failures of its incident links, but does not know the conditions of the other links.

Firstly, we assume that the size of a message is $O(n \log n)$. A naive nonadaptive routing in the faulty n-rotator graph exploits the $n-1$ disjoint paths to transmit each of $n-1$ copied messages concurrently. Thus the communication complexity of the naive nonadaptive routing algorithm is $(n-1)(2n-1) = O(n^2)$ while the time complexity is $2n-1 = O(n)$.

Since the connectivity of the n-rotator graph is $n-1$, no routing algorithm succeeds in the worst case if more than $n-2$ links fail. However, if we assume that links fail randomly, the nonadaptive routing can tolerate much more faulty links with high probability. Next we show the relation between the probability with which the routing succeeds and the probability with which the links fail randomly.

Suppose that links fail independently with a fixed probability $p = c/2n$, where $c(0 < c < 1)$ is a constant. Since the length of each path used by the naive nonadaptive routing is less than $2n$,

$$Prob\{\text{the path is healthy}\} > (1 - \frac{c}{2n})^{2n} > 1 - c.$$

Hence, the probability of a path being faulty is less than c. Since the links fail independently, the disjoint paths also fail independently. The probability that all the $n-1$ paths used by the naive nonadaptive routing fail simultaneously is less than c^{n-1}. Thus we have the following theorem.

Theorem 3 *The naive nonadaptive routing succeeds with probability higher than* $1 - c^{n-1}$ *if the links fail independently with probability* $c/2n$.

It means that the naive nonadaptive routing succeeds with high probability even when the links fail with probability $O(1/n)$ (In this case, the expected number of faulty links is $O(n!)$).

We next consider the case that the size of a message is not $O(n \log n)$. In this case, the message is split so that the size of each piece of the message is $O(n \log n)$, and each piece is transmitted one by one. Let m denote the size of the message. Then, the communication complexity of the naive nonadaptive routing is $O(n^2(m/n \log n)) = O(nm/\log n)$. Observe that the communication complexity of any routing algorithm is $\Omega(n(m/n \log n)) = \Omega(m/\log n)$. If m is $\Omega(n^2 \log n)$, we can reduce the communication complexity of the naive nonadaptive routing algorithm to $O(m/\log n)$, by using Rabin's Information Dispersal Algorithm (IDA)[4].

In Rabin's IDA, a message M of size m is transformed into $n-1$ messages $M_1, M_2, \cdots, M_{n-1}$. The size of each M_i $(i = 1, 2, \cdots, n-1)$ is $m/d(n-1)$, where $d(0 < d < 1)$ is supposed to be a constant such that $d(n-1)$ is an integer. Any $d(n-1)$ messages of $M_1, M_2, \cdots, M_{n-1}$ suffice to reconstruct the original message M. We transmit $M_1, M_2, \cdots, M_{n-1}$ through the $n-1$ disjoint paths respectively. If $m/d(n-1)$, the size of M_i, is not $O(n \log n)$, then the transformed message M_i is split so that the size of each piece of the message is $O(n \log n)$. Each piece of M_i is transmitted one by one through the same path, but no piece of $M_j (j \neq i)$ can use the same path as M_i's. If no more than $(1-d)(n-1)$ paths are faulty, the message M can be reconstructed at the destination. In this case, the communication complexity is $1/d(n-1)$ of that of the naive nonadaptive routing.

We discuss the reliability of the nonadaptive routing using Rabin's scheme. In our probabilistic consideration, we will use the following lemma known as Chernoff bound[2].

Lemma[2] *Let* X *be the number of successes in a series of* n *Bernoulli trials with success probability* p. *For any constant* ϵ *with* $0 < \epsilon < 1$,

$$Prob\{X \leq \epsilon pn\} \leq e^{-\frac{(1-\epsilon)^2 pn}{2}}.$$

We still assume that the links fail independently with probability $c/2n$, where $c(0 < c < 1)$ is a constant. Let d be the constant described above and we choose two constants c and d such that $d < 1 - c$.

Theorem 4 *The nonadaptive routing using Rabin's IDA as described above succeeds with probability higher than* $1 - e^{-\frac{(1-c-d)^2}{2(1-c)}(n-1)}$.

Proof: Let X denote the number of successes in a series of $n-1$ Bernoulli trials with success probability $1-c$. From Chernoff bound, we have

$$Prob\{\ X < d(n-1)\} \quad < \quad Prob\{\ X \leq \frac{d}{1-c}(1-c)(n-1)\}$$

$$\begin{aligned} &\leq e^{-(1-\frac{d}{1-c})^2(1-c)(n-1)/2} \\ &= e^{-\frac{(1-c-d)^2}{2(1-c)}(n-1)}. \end{aligned}$$

Let $P_1, P_2, \cdots, P_{n-1}$ denote the events that the $n-1$ paths from a source to the destination are healthy. Since the $n-1$ paths are node disjoint, events $P_1, P_2, \cdots, P_{n-1}$ are probabilistically independent. We have $Prob\{P_i\} > 1-c$ for $i = 1, 2, \cdots, n-1$ since the length of each path is shorter than $2n$. Thus,

$$Prob\{\text{less than } d(n-1) \text{ events of } P_1, P_2, \cdots, P_{n-1} \text{ take place}\} < Prob\{\ X < d(n-1)\}.$$

Hence, the nonadaptive routing using Rabin's Information Dispersal Algorithm succeeds with probability higher than $1 - e^{-\frac{(1-c-d)^2}{2(1-c)}(n-1)}$. □

5 Concluding Remarks

Corbett showed that the n-rotator graph remains strongly connected in the presence of $n-2$ faulty nodes[1], that is, connectivity of the n-rotator graph turns out to be $n-1$. We described the procedure which explicitly derives how to construct $n-1$ node disjoint paths in the n-rotator graph. All the $n-1$ paths were shown to have length shorter than $2n$. Using the $n-1$ node disjoint paths, we proposed two nonadaptive fault-tolerant routing algorithms and showed the probabilities that the algorithms succeed.

For further investigations, we note permutation routing and sorting in the n-rotator graph. Rabin's IDA[4] is also one of the useful tools for permutation routing since we can ignore the overflow of the buffer of each node to a certain extent. We conjecture that the n-rotator graph has the high bisection width $n!$. Thus, it is interesting to consider the load balancing problem in the n-rotator graph. We lastly mention a disadvantage of the rotator graph family that the numbers of nodes are considerably discrete. However, since rotator graphs are constructed inductively, it is worth studying how to connect several $(n-1)$-rotator graphs and extra nodes so that the diameter of the graph constructed may not exceed $n-1$.

References

[1] P. F. Corbett, "Rotator graphs: an efficient topology for point-to-point multiprocessor networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, pp. 622–626, 1992.

[2] T. Hagerup and C. Rub, "A guided tour of Chernoff bounds", *Information Processing Letters*, vol. 33, pp. 305–308, 1990.

[3] Y-D. Lyuu, "Fast fault-tolerant parallel communication for de Bruijn and digit-exchange networks using information dispersal", *Networks*, vol. 23, pp. 365–378, 1993.

[4] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance", *Journal of the ACM*, vol. 36, pp. 335–348, 1989.

[5] P. M. Yamakawa, H. Ebara and H. Nakano, "A routing algorithm in a faulty n-rotator graph", *Proceeding of Joint Symposium on Parallel Processing*, pp. 65–72, 1994.