An Efficient Evaluation for the Reliability Upper Bound of Distributed Systems with Unreliable Nodes and Edges

Mohamed H. S. Mohamed [1], Yang Xiaozong, Liu Hongwei, Wu Zhibo

School of Computer Science and Technology, Harbin Institute of Technology
Harbin 150001, China

***Abstract*—The distributed systems in which nodes and/or edges may fail with certain probabilities have been modelled by a probabilistic network or a graph *G*. Computing the residual connectedness reliability (RCR), denoted by *R*(*G*), of probabilistic networks under the fault model with both node and edge faults is very useful, but is an NP-hard problem. Since it may need exponential time of the network size to compute the exact value of *R*(*G*), it is important to calculate its tight approximate value. In this paper, we present a new approach with an efficient algorithm for evaluating the upper bound of *R*(*G*) of distributed systems with unreliable nodes and edges. We also apply our algorithm to some typical classes of networks to evaluate the upper bounds and show the effectiveness and the efficiency of the new algorithm. Numerical results are presented.**

Keywords- distributed system; residual connectedness reliability; node and edge faults; upper bound

I. INTRODUCTION

The behaviour of a distributed system can be modelled by a probabilistic network or a graph *G* whose nodes and/or edges may fail [1]. The ability of the communication between the residual (remaining working) nodes is measured by the RCR *R*(*G*), which is the probability that the residual nodes can communicate with each other [2]–[4].

Generally, there are three kinds of fault models in a probabilistic network [1]:

- *Node fault model*: The edges of a graph are perfectly reliable, but the nodes fail independently with probability $(1-p_0)$.
- *Edge fault model*: The nodes of a graph are perfectly reliable, but the edges fail independently with probability $(1-p_1)$.
- *Node-and-edge fault model*: Nodes and edges fail independently of each other, with node and edge failure probabilities equal to $(1-p_0)$ and $(1-p_1)$, respectively.

For all these three fault models, it has been shown that the analysis problems are all NP-hard [1], [4]–[6]; that is, there exists no efficient algorithms for computing *R*(*G*).

There are quite a number of papers dealing with approximation algorithms for estimating *R*(*G*) under the edge fault model and the node fault model. To our best knowledge, little work has been done under the fault model with both node and edge faults. Frank and Gaul [2] proposed the bounds to *R*(*G*) for the complete graph. Ball [1] has shown that the point estimate reliability problem is also NP-hard. Chen and He [7] studied the reliability bounds for arbitrary graphs under the node-and-edge fault model. They found good upper bound and lower bound expressions, and efficient algorithms for bounding the reliability *R*(*G*). They also demonstrated that the difference between the upper bound and lower bound gradually tends to zero for large networks, and are very close to zero for small networks. Shpungin [8] suggested a Monte Carlo scheme for evaluating reliability for the networks with unreliable nodes and unreliable edges, for the case of k-terminal connectivity criterion. But for the case of residual connectedness reliability, his scheme evaluates the reliability of the networks with unreliable nodes and with reliable edges.

In this paper, we present a new approach with an efficient algorithm for evaluating the upper bound of *R*(*G*) of distributed systems under the node-and-edge fault model. To show the effectiveness and the efficiency of the new upper bound, we apply the new algorithm to some typical classes of graphs. Numerical results are also derived.

II. NOTATIONS AND ACRONYMS

RCR	Residual Connectedness Reliability
UB	Upper Bound
NEF	Node-and-Edge Fault
G	undirected graph
$R(G)$	residual connectedness reliability
n	number of nodes in G
p_0	node operational probability
p_1	edge operational probability
$\overline{R}(G)$	upper bound of $R(G)$

III. METHOD

Our approach to obtain a tight UB with efficient running time is directly based on the original definition of the RCR:

$$R(G) = \Pr\{\text{the subgraph induced by the surviving nodes and edges of } G \text{ is connected}\}, \quad (1)$$

where $\Pr\{A\}$ stands for the probability of random event A.

Based on the relationship and the operation of the random events, a method for estimating reliability UB is presented.

Without loss of generality, we assume that graph G is initially connected, and its vertex set is $V = \{v_1, \ldots, v_n\}$. An edge is represented by two vertices.

Let E denote the random event that the surviving nodes and edges induced in the subgraph are connected. Then we have

$$R(G) = \Pr\{E\} = 1 - \Pr\{\overline{E}\}. \quad (2)$$

Let $E(v_i)$ be the event that v_i is isolated in the induced graph, $i = 1,...,n$. Then the occurrence of any $E(v_1)$, $E(v_2)$, …, $E(v_n)$ implies that E does not occur. Consequently we have

$$\overline{E} \supseteq \bigcup_{i=1}^{n} E(v_i). \quad (3)$$

It is obvious that $E(v_i)$ occurs if and only if v_i does not fail, and for each neighbour node w of v_i, either w or edge wv_i fails. The probability that such an event occurs is

$$\Pr\{E(v_i)\} = p_0(1 - p_1 p_0)^{d(v_i)},\ i = 1,...,n \quad (4)$$

where $d(v_i)$ is the degree of v_i.

Then the UB of $R(G)$ is given in Theorem 1.

Theorem 1: Let $r = |S|$, $S \in V$, and $S = \{S_1 \cup S_2\}$, where any two nodes of S have not any common neighbour node in G, and the nodes of any pair of adjacent nodes in S have the same degree in G. Let S_2 be the set of all pairs of adjacent nodes in S, where $S_2 = \{(y_k, z_k) : k = 1,...,r_2\}$ and r_2 is the number of these pairs. Let $S_1 = S - S_2$, where $S_1 = \{x_j : j = 1,...,r_1\}$ and $r_1 = r - 2r_2$. Then the UB of $R(G)$ is

$$R(G) \le \prod_{j=1}^{r_1}\left(1 - p_0(1 - p_1 p_0)^{d(x_j)}\right) \times \prod_{k=1}^{r_2}\Big(1 - p_0(1 - p_1 p_0)^{d(y_k)} - p_0(1 - p_1 p_0)^{d(z_k)}\left(1 - p_0(1 - p_1 p_0)^{d(y_k)-1}\right)\Big), \quad (5)$$

where p_0, p_1 are the node and edge operational probabilities, respectively.

Proof: Let $F(x_j)$ be the event that node x_j is isolated in the remaining subgraph of G, $j = 1,...,r_1$. Then

$$\Pr\{F(x_j)\} = p_0(1 - p_1 p_0)^{d(x_j)}. \quad (6)$$

Let $F(y_k)$ be the event that node y_k is isolated in the remaining subgraph of G, $k = 1,...,r_2$. Then

$$\Pr\{F(y_k)\} = p_0(1 - p_1 p_0)^{d(y_k)}. \quad (7)$$

Let $F(z_k)$ be the event that node z_k is isolated in the remaining subgraph of G, $k = 1,...,r_2$. Then

$$\Pr\{F(z_k)\} = p_0(1 - p_1 p_0)^{d(z_k)}. \quad (8)$$

From the definitions of S, S_1 and S_2 in Theorem 1, we have the following results:

- The distance between any two nodes in S is either one or at least three.
- The distance between any node in S_1 and any other node in S is at least three.
- The distance between any two different pairs of adjacent nodes in S_2 is at least three.

Therefore, it is obvious that

$$\Pr\left\{\bigcup_{j=1}^{r_1} F(x_j)\right\} = 1 - \bigcap_{j=1}^{r_1}\Pr\{\overline{F}(x_j)\} = 1 - \prod_{j=1}^{r_1}\Pr\{\overline{F}(x_j)\} = 1 - \prod_{j=1}^{r_1}\left(1 - p_0(1 - p_1 p_0)^{d(x_j)}\right), \quad (9)$$

$$\Pr\{F(y_k) \cup F(z_k)\} = \Pr\{F(y_k)\} + \Pr\{F(z_k)\}\Pr\{\overline{F}(y_k) | F(z_k)\} \ge p_0(1 - p_1 p_0)^{d(y_k)} + p_0(1 - p_1 p_0)^{d(z_k)}\left(1 - p_0(1 - p_1 p_0)^{d(y_k)-1}\right) \quad (10)$$

and

$$\Pr\left\{\bigcup_{k=1}^{r_2}(F(y_k) \cup F(z_k))\right\} = 1 - \Pr\left\{\bigcap_{k=1}^{r_2}\left(\overline{F(y_k) \cup F(z_k)}\right)\right\} = 1 - \prod_{k=1}^{r_2}\Pr\left\{\overline{F(y_k) \cup F(z_k)}\right\} \ge 1 - \prod_{k=1}^{r_2}\Big(1 - p_0(1 - p_1 p_0)^{d(y_k)} - p_0(1 - p_1 p_0)^{d(z_k)}\left(1 - p_0(1 - p_1 p_0)^{d(y_k)-1}\right)\Big). \quad (11)$$

Consequently

$$\Pr\left\{\left(\bigcup_{j=1}^{r1} F(x_j)\right) \cup \left(\bigcup_{k=1}^{r_2}(F(y_k) \cup F(z_k))\right)\right\} \ge 1 - \left(\prod_{j=1}^{r_1}\Pr\{\overline{F}(x_j)\} \times \prod_{k=1}^{r_2}\Pr\left\{\overline{F(y_k) \cup F(z_k)}\right\}\right). \quad (12)$$

Hence, since

$$\overline{E} \supseteq \bigcup_{i=1}^{n} E(v_i) \supseteq \left(\left(\bigcup_{j=1}^{r_1} F(x_j)\right) \cup \left(\bigcup_{k=1}^{r_2}(F(y_k) \cup F(z_k))\right)\right), \quad (13)$$

we then get

$$1 - R(G) \ge \Pr\left\{\left(\bigcup_{j=1}^{r1} F(x_j)\right) \cup \left(\bigcup_{k=1}^{r_2}(F(y_k) \cup F(z_k))\right)\right\}, \quad (14)$$

and

$$R(G) \le 1 - \Pr\left\{\left(\bigcup_{j=1}^{r1} F(x_j)\right) \cup \left(\bigcup_{k=1}^{r_2}(F(y_k) \cup F(z_k))\right)\right\}. \quad (15)$$

Thus we have proved the Theorem 1; that is

$$R(G) \le \prod_{j=1}^{r_1}\left(1 - p_0(1 - p_1 p_0)^{d(x_j)}\right) \times \prod_{k=1}^{r_2}\Big(1 - p_0(1 - p_1 p_0)^{d(y_k)} - p_0(1 - p_1 p_0)^{d(z_k)}\left(1 - p_0(1 - p_1 p_0)^{d(y_k)-1}\right)\Big).$$

IV. Algorithm for the Upper Bound

Based on the constructive method for estimating the UB above, we can obtain an algorithm which estimates the reliability UB of RCR $R(G)$. According to Theorem 1, the bigger the set S, the better the UB is. Thus, in order to obtain a

tighter UB, we must find the set S with as many nodes as possible.

A. Preparation

Before we derive the algorithm, we need the following procedure to find the set S.

```
Procedure set S(G)
Input: graph G
Output: a set S of graph G
1  H ← V , S1 ← ∅ , S2 ← ∅ , j ← 0 , k ← 0
2  while H ≠ ∅
3      do u ← the node of H with the min degree in G
4         adj ← 0
5         for every neighbour node v in G of node u
6             do if adj ← 0
7                 then if v and u have the same degree in
                       G, and do not have any common
                       neighbour node in G
8                      then k ← k +1 , yk ← u , zk ← v
9                           S2 ← S2 ⋃ {yk , zk }
10                          adj ← 1
11                          remove u and v and their
                            neighbours and neighbours
                            of the neighbours from H
12        if adj ← 0
13           then j ← j +1 , xj ← u
14                S1 ← S1 ⋃ {xj }
15                remove u and its neighbours and neigh-
                  bours of the neighbours from H
16 S ← {S1} ⋃ {S2 }
17 return S
```

Now we are in the position to give the UB $\overline{R}(G)$ of $R(G)$.

B. Algorithm

The following algorithm finds the UB $\overline{R}(G)$ of $R(G)$.

```
Algorithm Upper–Bound
Input: G, S, p0 , p1
Output: R̄(G) /* R̄(G) according to Theorem 1*/
1  R̄(G) ← 1
2  for j ← 1 to j ← r1
3      do R̄(G) ← R̄(G) × (1 − p0 (1 − p1 p0 )^d(xj) )
4  for k ← 1 to k ← r2
5      do R̄(G) ← R̄(G) × (1 − p0 (1 − p1 p0 )^d(yk)
                − p0 (1 − p1 p0 )^d(zk) (1 − p0 (1 − p1 p0 )^(d(yk)−1)))
6  return R̄(G)
```

V. The Complexity of the Algorithm

The main computational process for computing the UB is to find the set S in the procedure *set* $S(G)$. This procedure can be done in time $O(n^2)$, while the calculation of the bound value, taken directly from the Theorem 1, can be done in time $O(n)$. Thus the time complexity for the UB is $O(n^2)$.

VI. Example

To demonstrate the method and the algorithm above, the sample network illustrated by Fig. 1 is considered.

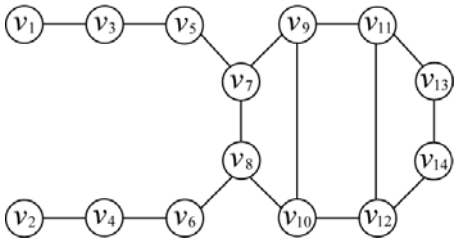

Fig. 1. Sample network

Applying our method and algorithm to sample network in Fig. 1, we get the following results:

$S_1 = \{v_1, v_2\}$, $S_2 = \{(v_{13}, v_{14}), (v_7, v_8)\}$,

$S = \{v_1, v_2, (v_{13}, v_{14}), (v_7, v_8)\}$;

$\Pr\{F(v_1)\} = p_0(1 - p_1 p_0)$, $\Pr\{F(v_2)\} = p_0(1 - p_1 p_0)$,

$\Pr\{F(v_{13})\} = p_0(1 - p_1 p_0)^2$, $\Pr\{F(v_{14})\} = p_0(1 - p_1 p_0)^2$,

$\Pr\{F(v_7)\} = p_0(1 - p_1 p_0)^3$, $\Pr\{F(v_8)\} = p_0(1 - p_1 p_0)^3$.

$$\Pr\{F(v_1) \cup F(v_2)\} = 1 - (1 - p_0(1 - p_1 p_0))^2,$$

$$\Pr\{F(v_{13}) \cup F(v_{14})\} \geq 1 - \left(1 - p_0(1 - p_1 p_0)^2 - p_0(1 - p_1 p_0)^2 (1 - p_0(1 - p_1 p_0))\right),$$

$$\Pr\{F(v_7) \cup F(v_8)\} \geq 1 - \left(1 - p_0(1 - p_1 p_0)^3 - p_0(1 - p_1 p_0)^3 \left(1 - p_0(1 - p_1 p_0)^2\right)\right);$$

$$R(G) \leq (1 - p_0(1 - p_1 p_0))^2 \times \left(1 - p_0(1 - p_1 p_0)^2 - p_0(1 - p_1 p_0)^2 (1 - p_0(1 - p_1 p_0))\right) \times \left(1 - p_0(1 - p_1 p_0)^3 - p_0(1 - p_1 p_0)^3 \left(1 - p_0(1 - p_1 p_0)^2\right)\right).$$

VII. Numerical Results and Discussions

To show the effectiveness and the efficiency of the new UB, we apply our algorithm to some typical classes of graphs such as circle graph, hypercube, and Harary graph, simply because these structures allow simpler routing algorithms, higher fault-tolerance ability and reliability. For example, in a p-dimension hypercube, or p-hypercube for short, denoted by Q_p, a large number of computing nodes (2^p nodes) are connected using a

smaller number of communication edges (p edges per node, instead of $2^p - 1$ edges per node as required by a complete graph) while keeping a minimal communication delay between the nodes. The hypercube has a symmetric and regular topology, which is very easy to understand and utilize.

To computationally examine the effectiveness of the new UB, three sets of graphs were used (see Tables I–III). When compared to the UB by Chen and He in [7], in each instance the new UB of this paper was tighter.

TABLE I
COMPARISONS OF UB VALUES FOR CIRCLE GRAPH C_n

n	q_0	q_1	UB by [7]	New UB
64	0.9	0.95	0.669529	0.582604
64	0.95	0.9	0.654624	0.566327
64	0.99	0.99	0.991799	0.988417
256	0.99	0.995	0.981366	0.9727
256	0.995	0.99	0.981273	0.972565
256	0.999	0.999	0.999661	0.999498
1024	0.999	0.9995	0.999234	0.998856
1024	0.9995	0.999	0.999234	0.998855
1024	0.9999	0.9999	0.999986	0.99998

TABLE II
COMPARISONS OF UB VALUES FOR HYPERCUBE Q_p WITH $p = 4$

q_0	q_1	UB by [7]	New UB
0.9	0.95	0.999602	0.996825
0.95	0.9	0.99958	0.996649
0.95	0.95	0.999914	0.999314

TABLE III
COMPARISONS OF UB VALUES FOR HARARY GRAPH $H_{k,n}$ WITH $k = 3$ AND $n = 18$

q_0	q_1	UB by [7]	New UB
0.9	0.95	0.99452	0.967826
0.95	0.9	0.994216	0.966082
0.95	0.95	0.99824	0.989528

As a result, the new UB is tighter than the UB by Chen and He in [7], as shown in Tables I–III, and is also calculated in time $O(n^2)$. The new algorithm can easily be implemented for evaluating reliability UB of distributed systems with unreliable nodes and edges, and produces a good approximation for RCR that can be used in general study in graphs and computer networks

VIII. CONCLUSIONS

In this paper, we presented a new method and efficient algorithm for evaluating reliability upper bound of distributed systems with unreliable nodes and edges. To numerically show the efficiency and the effectiveness of our algorithm, we have applied it to some typical classes of graphs. The new algorithm produces a good approximation for RCR that can be used in general study in graphs and computer networks.

ACKNOWLEDGMENT

The authors would like to thank the Editor/anonymous reviewers for their helpful comments on our submitted manuscript.

REFERENCES

[1] M. O. Ball, "Complexity of network reliability computations," *Networks*, vol. 10, pp. 153–165, 1980.
[2] O. Frank and W. Gaul, "On reliability in stochastic graphs," *Networks*, vol. 12, pp. 119–126, 1982.
[3] F. T. Boesch, "Synthesis of reliable networks–A survey," *IEEE Transactions on Reliability*, vol. 35, pp. 240–246, 1986.
[4] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal of Computing*, vol. 8, pp. 410–421, 1979.
[5] L. Fortnow and S. Homer, "A short history of computational complexity," Bulletin of the EATCS, vol. 80, pp. 95-133, 2003.
[6] H. L. Bodlaender and T. Wolle, "A Note on the Complexity of Network Reliability Problems," Technical Report: UC-CS-2004-001, Utrecht University, 2004.
[7] Y. Chen and Z. He, "Bounds on the reliability of distributed systems with unreliable nodes and links," *IEEE Transactions on Reliability*, vol. 53, pp. 205-215, 2004.
[8] Y. Shpungin, "Combinatorial approach to reliability evaluation of network with unreliable nodes and unreliable edges," *International Journal of Computer Science*, vol. 1, no. 3, pp. 177–182, 2006.