World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:3, 2009

# An Efficient Algorithm for Reliability Lower Bound of Distributed Systems

Mohamed H. S. Mohamed, Yang Xiao-zong, Liu Hong-wei, and Wu Zhi-bo



***Abstract*—**The reliability of distributed systems and computer networks have been modeled by a probabilistic network or a graph *G*. Computing the residual connectedness reliability (RCR), denoted by *R*(*G*), under the node fault model 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, especially its lower bound, at a moderate calculation time. In this paper, we propose an efficient algorithm for reliability lower bound of distributed systems with unreliable nodes. We also applied our algorithm to several typical classes of networks to evaluate the lower bounds and show the effectiveness of our algorithm.

***Keywords*—**Distributed systems, probabilistic network, residual connectedness reliability, lower bound.

## I. Introduction

THE reliability of distributed systems and computer networks have been modeled 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 RCR *R*(*G*), which is the probability that the residual nodes can communicate with each other [2]–[5].

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 certain probabilities.
- **Edge fault model**: The nodes of a graph are perfectly reliable, but the edges fail independently with certain probabilities.
- **Node-and-edge fault model**: Nodes and edges of a graph fail independently of each other, with certain node and edge failure probabilities.

For all these three fault models, it has been shown that the analysis problems are all NP–hard [1], [5]–[7]; 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 node fault model [8]-[10]. Colbourn [8] proposed a polynomial algorithm of certain restricted classes of graphs, including trees, series-parallel graphs, and permutation graphs. He and Chen [9] developed efficient algorithms of arbitrary graphs, and bound expressions for estimating *R*(*G*). They demonstrated theoretically and numerically that the difference between the upper and the lower bounds gradually tends to zero as the number of nodes tends to infinity under the condition that the node failure probability is reasonably low, e.g., less than 0.1. He, Tian and Chen [10] presented a new approach that combines a Monte Carlo simulation scheme and the deterministic bounding approach in [9] to obtain a probabilistic point estimator for *R*(*G*). Unfortunately, the complexity of the algorithms in [9] is highly polynomial since the entire operational time complexity for the lower bound is $O(n^5)$.

In this paper, we present a new approach with efficient algorithm for evaluating the reliability lower bound of distributed systems with unreliable nodes. Our algorithm for the lower bound is faster than the proposed algorithm by He and Chen [9], and calculated in $O\left(\min\left\{\kappa^3+n, \kappa n\right\}\kappa n\right)$ time, where $n$ and $\kappa$ are the order and the vertex connectivity of $G$, respectively. The new lower bound is also tighter than the lower bound in [9].

## II. Acronyms and Notations

| | |
|---|---|
| DS | Distributed System |
| RCR | Residual Connectedness Reliability |
| LB | Lower Bound |
| NF | Node Fault |
| $G$ | graph |
| $R(G)$ | residual connectedness reliability of $G$ |
| $n$ | number of nodes in $G$ |
| $q$ | node failure probability |
| $\underline{R}(G)$ | lower bound of $G$ |

## III. Method

Formally, the RCR, *R*(*G*), of a probabilistic graph *G* is the probability that the residual subgraph is connected, or all the operating nodes of the DS can communicate with each other. We thus have the following definition:

Mohamed H. S. Mohamed is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (corresponding author to provide mobile: +86-15124521620; fax: +86-451-86414093; e-mail: mhsm68@ yahoo.com).

Yang Xiao-zong is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: xzyang@ hope.hit.edu.cn).

Liu Hong-wei is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: lhw@ ftcl.hit.edu.cn).

Wu Zhi-bo is with the Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: wzb@ ftcl.hit.edu.cn).

$R(G)$=Pr{the subgraph induced by the residual nodes of $G$ is connected}, (1)

where Pr{$A$} stands for the probability of random event $A$. Suppose that each node fails with the same probability $q$, formula (1) can be rewritten as

$$R(G)=\sum_{i=0}^{n} F_i(G)\,q^{n-i}(1-q)^i, \tag{2}$$

where $F_i(G)$ is the number of connected subgraphs induced by $i$ residual nodes of $G$.

Our approach of obtaining the LB is directly based on formula (1), the original definition of the $R(G)$. Without loss of generality, we assume that graph $G$ is connected initially and its node set is $V = \{v_1, \ldots, v_n\}$.

To derive the LB of $R(G)$, we need the following definitions:

***The vertex connectivity of a graph G***, denoted by $\kappa$, is the smallest number of vertices whose deletion separates the graph or makes it trivial.

***The maximum degree of a graph G***, denoted by $\Delta$, is the maximum degree of its vertices.

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

***Theorem 1***: Let $G$ be a graph with order $n$, node failure probability $q$, maximum degree $\Delta$ and vertex connectivity $\kappa$. Then the lower bound of $R(G)$ is

$$R(G)\ge\sum_{i=2}^{r}(r-i+1)\left(q^{\kappa}\right)^{r-i}\left(1-q^{\kappa}\right)^i, \tag{3}$$

where $$r=\left((n-\Delta)\left(\left\lceil\frac{n(\kappa-1)}{\kappa}+1\right\rceil\right)\right)+2^{\left\lfloor\frac{1}{\kappa}\right\rfloor-\left\lfloor\frac{3}{\Delta+\kappa}\right\rfloor-\left\lfloor\frac{\Delta}{n-1}\right\rfloor}+1 \tag{4}$$

***Proof***: Let $g$ be a path graph with order $r$ and node failure probability $q^{\kappa}$. Then

$$R(g)=\sum_{i=2}^{r}(r-i+1)\left(q^{\kappa}\right)^{r-i}\left(1-q^{\kappa}\right)^i, \tag{5}$$

where $r$ is defined in (4).

Let $\Lambda(G)$ and $\Lambda(g)$ are the max numbers of minimum vertex cuts in $G$ and $g$, respectively.

Then, we can obtain from (4) that:

$$\Lambda(g)=r-\left(2\left\lceil\frac{r-2}{r-1}\right\rceil\right)$$

and

$$\Lambda(G)\le\Lambda(g)$$

Since $G$ and $g$ have the same node connectivity probability $q^{\kappa}$, and all minimum cuts of $g$ are disjointed, then

$$R(G)\ge R(g).$$

This leads to the proof of the theorem; that is:

$$R(G)\ge\sum_{i=2}^{r}(r-i+1)\left(q^{\kappa}\right)^{r-i}\left(1-q^{\kappa}\right)^i.$$

## IV. Algorithm

Before we derive the algorithm, we need to first find the vertex connectivity $\kappa$ of $G$, using the proposed algorithm by Henzinger, Rao and Gabo [11]. Then the LB $\underline{R}(G)$ of $R(G)$ is obtained by the following algorithm.

**Algorithm: Lower–Bound**
**Input** : Probabilistic graph $G, q, \Delta, \kappa$
**Output**: $\underline{R}(G)$ /* $\underline{R}(G)$ according to Theorem 1*/

1 : $r\leftarrow\left((n-\Delta)\left(\left\lceil\frac{n(\kappa-1)}{\kappa}+1\right\rceil\right)\right)+2^{\left\lfloor\frac{1}{\kappa}\right\rfloor-\left\lfloor\frac{3}{\Delta+\kappa}\right\rfloor-\left\lfloor\frac{\Delta}{n-1}\right\rfloor}+1$

2 : $\underline{R}(G)\leftarrow 0$

3 : **for** $i\leftarrow 2$ **to** $r$

4 : **do** $\underline{R}(G)\leftarrow\underline{R}(G)+\left((r-i+1)\left(q^{\kappa}\right)^{r-i}\left(1-q^{\kappa}\right)^i\right)$

5 : **return** $\underline{R}(G)$

## V. Complexity

The main computational process for obtaining the LB of $R(G)$ is to find the vertex connectivity $\kappa$ of $G$. This procedure can be done in time $O\left(\min\left\{\kappa^3+n,\kappa n\right\}\kappa n\right)$, using the proposed algorithm by Henzinger, Rao and Gabo [11], while the calculation of the bound value, taken directly from Theorem 1, can be done in time $O(n^2)$. Thus the complete algorithm for the LB takes $O\left(\min\left\{\kappa^3+n,\kappa n\right\}\kappa n\right)$ operations.

## VI. Computational Results and Discussion

In order to show the efficiency of our algorithm, we apply our method to some typical classes of graphs, for example:

- Order $n$ path $P_n$,
- Star graph $S_n$,
- Circle graph $C_n$,
- $p$–dimension hypercube $Q_p$, and
- Harary graph $H_{k,n}$.

To computationally examine the effectiveness and the efficiency of our LB five sets of typical graphs were used (see Tables I–V). When compared to the proposed LB by He and Chen [9], in each instance our LB was tighter (see Tables I–V).

For computing their LB, He and Chen [9] need to obtain as many short paths between each pair of non-adjacent nodes as possible. If Dijkstra's shortest algorithm is employed, which is $O(n^2)$-complex, to find the shortest path one after another. Since there are at most $n$–1 disjointed paths between each pair of nodes and $n(n–1)/2$ pairs of nonadjacent nodes resulting at most $n^3$ paths. Thus the complexity of their LB is $O(n^5)$. This analysis shows that their algorithm for computing LB is highly polynomial.

TABLE I
LOWER BOUNDS FOR $P_n$

| $q$ | $n$ | New LB | LB by [9] |
|---|---|---|---|
| $2^5$ | $1\times10^1$ | 0.043458 | 0 |
| | $1\times10^2$ | 0.739851 | 0.697072 |
| | $1\times10^3$ | 0.970433 | 0.969971 |
| $2^8$ | $1\times10^2$ | 0.07788 | 0 |
| | $1\times10^3$ | 0.775595 | 0.745747 |
| | $1\times10^4$ | 0.974919 | 0.974597 |
| $2^{10}$ | $1\times10^3$ | 0.359691 | 0 |
| | $1\times10^4$ | 0.902844 | 0.89779 |
| | $1\times10^5$ | 0.989832 | 0.98978 |

TABLE II
LOWER BOUNDS FOR $S_n$

| $q$ | $n$ | New LB | LB by [9] |
|---|---|---|---|
| $2^5$ | $1\times10^1$ | 0.891 | 0 |
| | $1\times10^2$ | 0.989901 | 0 |
| | $1\times10^3$ | 0.998999 | 0.535 |
| $2^8$ | $1\times10^2$ | 0.989901 | 0 |
| | $1\times10^3$ | 0.998999 | 0 |
| | $1\times10^4$ | 0.9999 | 0 |
| $2^{10}$ | $1\times10^3$ | 0.998999 | 0 |
| | $1\times10^4$ | 0.9999 | 0 |
| | $1\times10^5$ | 0.99999 | 0 |

TABLE III
LOWER BOUNDS FOR $C_n$

| $q$ | $n$ | New LB | LB by [9] |
|---|---|---|---|
| $2^5$ | $1\times10^1$ | 0.005943 | 0 |
| | $1\times10^2$ | 0.950276 | 0.676768 |
| | $1\times10^3$ | 0.99949 | 0.967968 |
| $2^8$ | $1\times10^2$ | 0.03775 | 0 |
| | $1\times10^3$ | 0.967765 | 0.743744 |
| | $1\times10^4$ | 0.999672 | 0.974397 |
| $2^{10}$ | $1\times10^3$ | 0.591978 | 0 |
| | $1\times10^4$ | 0.994771 | 0.89759 |
| | $1\times10^5$ | 0.999948 | 0.98976 |

TABLE IV
LOWER BOUNDS FOR $Q_5$

| $q$ | New LB | LB by [9] |
|---|---|---|
| 0.1 | 0.992736 | 0.967696 |
| 0.08 | 0.997614 | 0.988708 |
| 0.06 | 0.999433 | 0.997137 |
| 0.04 | 0.999925 | 0.999596 |
| 0.02 | 0.999998 | 0.999986 |

TABLE V
LOWER BOUNDS FOR $H_{4,16}$

| $q$ | New LB | LB by [9] |
|---|---|---|
| 0.1 | 0.98452 | 0 |
| 0.08 | 0.99363 | 0.240162 |
| 0.06 | 0.99798 | 0.564867 |
| 0.04 | 0.999601 | 0.802995 |
| 0.02 | 0.999975 | 0.949799 |

As a result, our algorithm for computing the LB for reliability $R(G)$ of distributed systems with unreliable nodes is faster than the proposed algorithm by He and Chen [9], and the new LB is also tighter.

## VII. CONCLUSION

In this paper, we presented a new approach with efficient algorithm for evaluating the reliability lower bound of distributed systems with unreliable nodes. We have applied our algorithm to several typical classes of graphs (networks) to show the efficiency of the algorithm. Our approach produces a good evaluation for RCR that can be used in general study in graphs and computer networks.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and editors for their future comments.

## 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] T. Evans, and D. Smith,"Optimally reliable graphs for both edge and vertex failures," *Networks*, vol. 16, pp. 199–204, 1986.
[4] F. T. Boesch, "Synthesis of reliable networks–A survey," *IEEE Transactions on Reliability*, vol. R–35, pp. 240–246, 1986.
[5] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal of Computing*, vol. 8, pp. 410–421, 1979.
[6] J. S. Provan, and M. O. Ball, "The complexity of counting cuts and computing the probability that a graph is connected," *SIAM Journal of Computing*, vol. 12, pp. 777–788, 1983.
[7] K. Sutner, A. Satyanarayana, and C. Suffel, "The complexity of the residual node connectedness reliability problem," *SIAM Journal of Computing*, vol. 20, pp. 149–155, 1991.
[8] C. J. Colbourn, A. Satyanarayana, C. Suffel, and K. Sutner, "Computing residual connectedness reliability for restricted networks," *Discrete Applied Mathematics*, vol. 44, pp. 221–232, 1993.
[9] Z. He, and Y. Chen, "Efficient algorithms for residual connectedness reliability of distributed systems," in *Proceedings of the First International Workshop on Distributed Computing, Communication and Applications*, Islamabad, Pakistan, May 2000, pp. 151–159.
[10] Z. He, Y. Tian, and Y. Chen, "Simulating the reliability of distributed systems with unreliable nodes," *I. J. of Simulation*, vol. 3 No. 1–2, pp. 68–79, 2002.
[11] M. R. Henzinger, S. Rao, and H. N. Gabow, "Computing vertex connectivity: new bounds from old techniques," *J. of Algorithms*, vol. 34 No. 2, pp. 222–250, 2000.