SIMULATING THE RELIABILITY OF DISTRIBUTED SYSTEMS WITH UNRELIABLE NODES

ZHONGSHI HE[*], YUFANG TIAN[**] and YINONG CHEN[***]

[*]*College of Computer Science,* [**]*College of Mathematics and Physics, Chongqing University, Chongqing 400044, People's Republic of China, zshe@cqu.edu.cn, yufangtian@yahoo.com*

[***] *Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85287-5406, USA, yinong@asu.edu*

Abstract: This paper discusses the reliability of distributed systems in which nodes may fail with certain probabilities. The distributed systems have been modeled by a probabilistic graph G. We focus on the communication reliability that is characterized by a particular reliability attribute, the residual connectedness reliability, denoted by $R(G)$. The residual connectedness reliability is the probability that all residual nodes are reachable from each other. It has been shown that $R(G)$ is very useful but computing $R(G)$ is #P-Complete, which is at least as hard as an NP-complete problem. Research in this area is focusing on heuristic approaches. In this study, we first propose a deterministic bounding approach to bound $R(G)$. We obtain a deterministic upper bound and a lower bound. To prove that our bounds are tight, we demonstrate 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. In other words, for large distributed systems the upper and lower bounds give us an accurate estimation of $R(G)$. Unfortunately, this approach doesn't work well for small and middle-sized systems with a high failure probability, e.g., greater than 0.1. In the second part of the paper, we present a new approach that combines a Monte Carlo simulation scheme and our deterministic bounding approach to obtain a probabilistic point estimator for $R(G)$. We also determine the confidence interval of the estimator.

Keywords: residual connectedness reliability, Monte Carlo simulation, distributed systems, bounding

1. INTRODUCTION

The research was stimulated by our projects on developing a dependable distributed system, where we explored fault-tolerant protocols supporting dynamic task allocation with possible redundant copies of tasks and system configuration in case of node failures, reliability and performance evaluation of the system [Chen, Galpin , Hazelhurst, Mateer and Mueller, 1999]. A prototype of the system has been constructed over the past few years and we reported the construction of the prototype and measured performance in terms of data throughput on the system in [Chen and Mateer, 2001]. Although the modeling techniques developed in the projects are mainly based on our system, they can be applied to other distributed systems. We reported one of these applications in [Chen and He, 2001], where we analyzed the reliability of homogeneous and heterogeneous distributed systems.

This work investigates our system in a more theoretical way. We consider generally that the system consists of a set of computing nodes and a set of communication links between nodes. We assume that communication links are reliable while nodes may fail with certain probability. A simplified model of the topology of this system is a probability graph $G = (V, L)$ consisting of a set V of n nodes, representing computing nodes, and a set of L of m links, representing communication links. In addition, each node is associated with an independent probability q_v of being failed, or probability $1 - q_v$ of being operational [Boesch, 1986]. Initially, we have a fully connected distributed system. A task may have one, two or three copies (replicas) running on different

nodes of the distributed system, depending on the criticality of the task. The results of replicas will be compared. A disagreement in comparison indicates a possible node failure in one or both of the two nodes compared. When the number of disagreements detected from a certain node exceeds a given threshold, the node is considered to have a permanent fault and a reconfiguration (task reallocation) will be activated. As the result of the reconfiguration, a faulty node is removed (isolated) from the system and replicas of tasks are redistributed among the residual nodes. An optimal task reallocation during the reconfiguration needs information on the communication ability between the residual nodes.

The communication ability between the residual nodes is characterized by the residual connectedness reliability (RCR), $R(G)$, which is the probability that the residual nodes can communicate with each other. The residual connectedness reliability has been studied and applied in different areas. Unfortunately, the existing results in RCR cannot satisfy our needs, that is, highly accurate computing of RCR in short period of time for large distributed systems.

There are two major research questions on RCR that have been studied, the synthesis problem and the analysis problem. Their definitions are given as follows.

Given the number of nodes n, and the number of links m, we call any n-node, m-link graph an (n, m) graph. The **synthesis problem** is to construct an (n, m) graph that maximizes $R(G)$ for the given n, m and a node failure probability q, where $q \in (0, 1)$. Current research made little progress on solving this problem [Smith, 1984; Doty 1989; Smith, 1990; Goldschmidt et al, 1994; Liu et al, 2000]. In our current research we are not interested in constructing a system that maximizes $R(G)$ and thus the synthesis problem will not be discussed further in this paper.

The **analysis problem** is to compute R(G) for a given graph G and a given failure probability of its nodes [Boesch,1986]. This problem has been shown to be #P-Complete, the problem remains #P-Complete for split graphs as well as planar and bipartite graphs [Valiant, 1979; Ball, 1980; Sutner et al, 1991; Colbourn, 1993]. For this reason, significant efforts have been made to obtain the lower and upper bounds on $R(G)$ that can be computed efficiently.

The class of #**P-Complete problems** is a class of computationally equivalent problems that are at least as hard as the NP-complete problems.

We have carefully studied all related papers that we can find. There are quite a number of papers dealing with the synthesis problem [Smith, 1984; Doty 1989; Smith, 1990; Goldschmidt et al, 1994; Liu et al, 2000], but only one dealing with the analysis problem [Colbourn et al, 1993]. Colbourn proposed a polynomial algorithm of certain restricted classes of graphs, including trees, series-parallel graphs, permutation graphs etc [Colbourn et al, 1993].

Not only is the analysis problem #P-Complete, but also, Ball [Ball, 1980] has shown that the following point estimate reliability analysis problem is #P-Complete:

The **point estimate reliability analysis problem** is to find a point estimate value r of reliability $R(G)$ such that the value $r/R(G)$ lies between two given positive numbers $\alpha < 1$ and $\beta > 1$, such that

$$\alpha R(G) \le r \le \beta R(G)$$

There are two cases, one is that each node fails with equal probability q, the other is that node failure probability q_v could be different for each node. Certainly, the former is a special case of the latter. The distributed system we are developing is a homogeneous system. We can assume that node failure probability is the same. We will present an efficient algorithm that finds the deterministic upper and lower bounds on $R(G)$ for the system as well as a simulation scheme that finds the probabilistic interval as the estimates of $R(G)$.

The rest of the paper is organized as follows. Section 2 starts with formulating the reliability model of distributed systems followed by a simple example. Section 3 devotes to efficient deterministic bounding approach to find the deterministic upper and lower bounds on $R(G)$ for large systems. The section also discusses the time complexity of the algorithm and accuracy of the results. In section 4, a combined Monte Carlo simulation scheme is proposed to find the probabilistic point estimator and confidence interval of $R(G)$ for small and middle-sized graphs. Finally, section 5 concludes the paper.

2. RELIABILITY MODEL OF DISTRIBUTED COMPUTING SYSTEMS

A distributed computing system with unreliable nodes can be modeled by probabilistic graph G, in which each node operates independently with the same probability $1 - q$. It is a random event whether a node operates or failed. The node will be removed from the graph if it fails. The residual graph, whose connectedness is a random event, is the subgraph induced by all the remaining (operating) nodes in G.

Formally, the residual connectedness reliability, $R(G)$, of a probabilistic graph G is the probability that the residual subgraph is connected, or all the operating nodes of the distributed computing system can communicate with each other. We thus have the following definition:

$$R(G) = \text{Prob}\{\text{the subgraph induced by the residual nodes of } G \text{ is connected}\}, \quad (1)$$

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

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

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

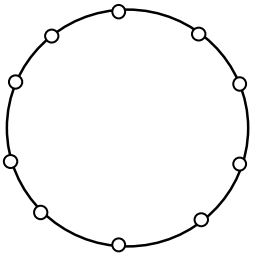
Most distributed systems follow regular topologies to interconnect computing nodes such as circle, rectangular mesh, hypercube and star graph, simply because these structures allow simpler routing algorithms, higher fault-tolerance ability and reliability [Pradhan and Reddy, 1982; Somani and Peleg O, 1996]. For example, in a p-dimension hypercube, or p-hypercube for short, denoted by Q_p, as show in Figure 1, a large number of computing nodes (2^p nodes) are connected using a small number of communication links (p links per node, instead of 2^p-1 links per node as required by a completed 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 to utilize. There are very simple and optimal routing algorithms and a hypercube can be easily decomposed into subcubes of lower dimensions, as can be seen in Figure 1 (b). It has p disjoint paths between each pair of nodes, hence it is fault-tolerant and highly reliable for a system with $p \geq 3$. We will use these classes of systems as example in the rest of this paper. However, the schemes presented in this paper are suitable for other topologies of distributed systems.

According to formula (2) above, the problem of computing the $R(G)$ is in fact finding the combinatorial coefficients F_i. Let's start with a simple example and consider the circle in Figure 1 (a), denoted by C_n. The coefficients in this case are the numbers of induced connected subgraphs. It can be easily seen that

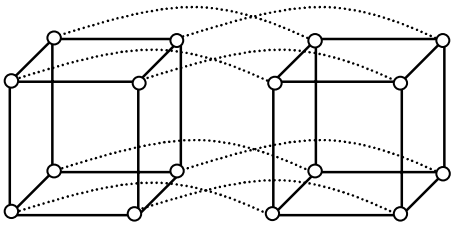
$$F_0 = F_1 = 0, F_2 = F_3 = \ldots = F_{n-1} = n, F_n = 1$$

Put these coefficients in (2) we have

$$R(C_n) = n\left(q^{n-2}(1-q)^2 + q^{n-3}(1-q)^3 + \cdots + q(1-q)^{n-1}\right) + (1-q)^n \quad (3)$$



(a)



(b)

Figure 1. Different topologies of distributed computing systems: Loop and Hypercube

Note that, as we discussed in the last section, it is #P-Complete to compute RCR for general graphs. From the example we can see that the major complexity is caused by computing the coefficients. For this simple example, we can obtain accurate values of $R(C_{10})$ from formula (3) for any given node failure probabilities q. Table 1 and Figure 2 show $R(C_{10})$ with the failure probability q varies from 0 to 0.5.

However, this is an extremely simple case. The time complexity of finding the coefficients for a generic graph is #P-Complete. We cannot obtain the accurate RCR in an acceptable amount of computing time.

The next two sections are devoted to finding efficient deterministic bounding approach and probabilistic point estimator and confidence interval to evaluate $R(G)$ for arbitrary graph G.

Table 1. The accurate $R(C_{10})$ value versus failure probability q

q	0.0500	0.1000	0.1500	0.2000	0.2500
$R(C_{10})$	0.9314	0.7845	0.6187	0.4653	0.3378
q	0.3000	0.3500	0.4000	0.4500	0.5000
$R(C_{10})$	0.2399	0.1694	0.1223	0.0936	0.0791

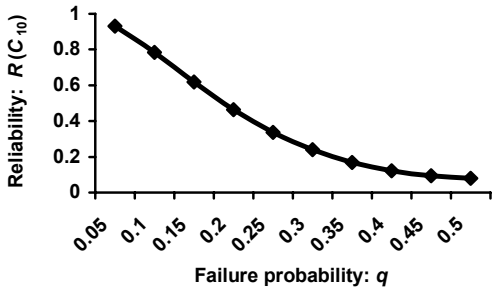


Figure 2. Accurate value of $R(C_{10})$

3. BOUNDING RCR

3.1. The Upper And Lower Bounds

Our scheme of obtaining the upper and lower bounds is directly based 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 $V = \{v_1, ..., v_n\}$. A link in graph G is represented by a pair of nodes (v_i, v_j).

Let E denote the random event that the residual nodes induced subgraph is connected, and $\overline{E}$ is the complement event of E. Then, according to formula (1), we have

$$R(G) = 1 - \text{Prob}\{\overline{E}\} \quad \text{and} \quad \overline{E} \supseteq \bigcup_{i=1}^{n} E_i$$

where E_i represents the event that node v_i is isolated in the induced subgraph, $i = 1, 2,..., n$.

In order to find an upper bound of $R(G)$, we first define a 3-independent set in a graph G, denoted by S_3. A **3-independent set** is such a subset of nodes in which any pair of nodes has a distance of at least 3 in graph G.

Then the upper bound of reliability $R(G)$ is given in theorem 1.

Theorem 1 Let $r_3 = |S_3|$ be the cardinality of set S_3 and $\Delta = \Delta(G)$ be the maximum node degree of graph G. Then an upper bound of $R(G)$ is

$$R(G) \le \left(1 - (1-q)q^{\Delta}\right)^{r_3}.$$

Proof Let $S_3 = \{v_{k_i} : i = 1, \ldots, r_3\}$, $N_G(v_{k_i})$ be the neighboring set of node v_{k_i} in G, and F_i be the event that v_{k_i} is isolated in the residual subgraph of G, $i = 1, 2, \ldots, r_3$. Then

$$\bigcup_{i=1}^{r_3} F_i \subseteq \bigcup_{i=1}^{n} E_i \subseteq \overline{E}.$$

It is obvious that F_i occurs if and only if v_{k_i} is operating and all its neighbor nodes fail. The probability that such an event occurs is

$$\text{Prob}\{F_i\} = (1-q)q^{|N_G(v_{k_i})|} \le (1-q)q^{\Delta},$$
$$i = 1, \ldots, r_3.$$

Further more, we assume that $F_1, \ldots, F_{r_3}$ are independent of each other, this implies

$$1 - R(G) = \text{Prob}\{\overline{E}\} \ge \text{Prob}\{\bigcup_{i=1}^{r_3} F_i\}$$

$$= 1 - \prod_{i=1}^{r_3} \text{Prob}\{\overline{F}_i\} = 1 - \left(1 - (1-q)q^{\Delta}\right)^{r_3}$$

that is,

$$R(G) \le \left(1 - (1-q)q^{\Delta}\right)^{r_3}.$$

Now the question is how tight is this bound with respect to the number of nodes in the 3-independent set. From the proof of the theorem we can see that the more nodes are included in the 3-independent set, the tighter the upper bound will be. Thus, in order to

obtain a tighter bound, we must find the 3-independent set with as many nodes as possible. The accuracy of both upper and lower bounds will be further studied in section 3.3.

Now let's move to the lower bound. The key to the lower bound is to find as many short paths between every pair of nodes as possible. Let $d(u,v)$, or d for short, represent the distance between u and v; $n_j(u, v)$, or n_j for short, represent the number of j-distance paths connecting u with v in which all paths are disjointed, where $j \geq d = d(u, v)$. Based on these notations, we have then the following theorem on the lower bound.

Theorem 2 Let q be the node failure probability, and n_j be the number of disjointed paths. A lower bound on $R(G)$ is

$$R(G) \geq 1+ (n-1)\, q^n - n\, q^{n-1} - \sum_{d\geq 2} \sum_{d(u,v)=d} p(d, u, v),$$

where $p(d, u, v)$ is recursively defined by

$$p(2, u, v) = \prod_{j=2}^{4} (1-(1-q)^{j-1})^{n_j} \ ,$$

$$p(3, u, v) = (q^2)^{n_3} (q(1-(1-q)^2))^{n_4} ((1-(1-q)^2)^2)^{n_5} \ ,$$

and

$$p(d, u, v) = (q^{d-1})^{n_d} (q^{d-2}(1-(1-q)^2))^{n_{d+1}} \times$$
$$(q^{d-3}(1-(1-q)^2)^2)^{n_{d+2}} , \text{ for } d > 3$$

Proof. Let $E_d(u, v)$ be the probability that "For a pair of nodes u and v with $d(u, v) = d$, there is a path connecting u and v; The path can be partitioned into a few subpaths with their distances less than or equal to $d - 1$; Each subpath has an operating starting-node and an operating ending-node", where $d \geq 2$.

Furthermore let

$$E_d = \bigcap_{d(u,v)=d} E_d(u,v) \ , \ d \geq 2; \ E = \bigcap_{d\geq 2} E_d$$

and let X be "The number of operating nodes in graph G".

Then we have

$$R(G) \geq \text{Prob}\{E \cap \{X \geq 2\}\} = \text{Prob}\{E - \{X < 2\}\}$$
$$\geq \text{Prob}\{E\} - \text{Prob}\{X < 2\}\}$$
$$= \text{Prob}\{E\} - \text{Prob}\{X = 0\} - \text{Prob}\{X = 1\}$$

It can be easily shown that

$$\text{Prob}\{X = 0\} = q^n, \ \text{Prob}\{X = 1\} = n\,(1-q)\, q^{n-1},$$

On the other hand

$$\text{Prob}\{E\} = 1 - \text{Prob}\{\overline{E}\}$$
$$= 1 - \text{Prob}\{\bigcup_{d\geq 2} (\bigcup_{d(u,v)=d} \overline{E_d(u,v)})\}$$
$$\geq 1 - \sum_{d\geq 2} \sum_{d(u,v)=d} \text{Prob}\{\overline{E_d(u,v)}\} \ .$$

Then

$$R(G) \geq 1 + (n-1)\, q^n - nq^{n-1} -$$
$$\sum_{d\geq 2} \sum_{d(u,v)=d} \text{Prob}\{\overline{E_d(u,v)}\}$$

In order to obtain the upper bounds of $\text{Prob}\{\overline{E_d(u,v)}\}$ for $d \geq 2$, three cases need to be discussed.

Case 1 $d = 2$, or $d(u, v) = 2$. All paths between u and v are inner disjointed nodes. There are n_j paths with distance j, j=1,2,…, and $n_j \geq 0$. $\overline{E_d(u,v)}$ or $\overline{E_2(u,v)}$ occurs implies that each path connecting u, v must contain at least one failed inner node. Then

$$\text{Prob}\{\overline{E_d(u,v)}\} \leq \prod_{j\geq 2} (1-(1-q)^{j-1})^{n_j} = p(2, u, v).$$

Case 2 $d = 3$, or $d(u, v) = 3$. Use the same method we used in case 1, we can obtain

$$\text{Prob}\{\overline{E_d(u,v)}\} \leq (q^2)^{n_3} (q(1-(1-q)^2))^{n_4} \times$$
$$((1-(1-q)^2)^2)^{n_5} \ .$$

Because $d = 3$, the event that $\overline{E_d(u,v)}$ occurs, e.g., $\overline{E_3(u,v)}$ occurs, implies that all the $(n_3 + n_4 + \ldots)$ paths must be satisfied by the following facts, respectively:

For every 3-distance path: both of the two inner nodes must have failed, thus the probability is q^2;

For every 4-distance path: the 2nd inner node must have failed, while at least one of the 1st and the 3rd inner nodes must have failed. Thus, the probability is $q(1-(1-q)^2)$;

For every 5-distance path: either the 1st or the 3rd inner node must have failed, while either the 2nd or the 4th inner node must have failed. Thus, the probability is $(1-(1-q)^2)^2$.

Case 3 $d \geq 4$, or $d(u, v) \geq 4$. For the same reason as in case 2, we can obtain

$$\text{Prob}\{\overline{E_d(u,v)}\} \leq (q^{d-1})^{n_d} (q^{d-2}(1-(1-q)^2))^{n_{d+1}} \times (q^{d-3}(1-(1-q)^2)^2)^{n_{d+2}}$$

Thus we have the theorem proved.

Please note that the coefficients of the bounding polynomial on $R(G)$ in theorems 1 and 2 are invariant with respect to failure probability q. Therefore, we need only reevaluate the polynomial instead of calculating the coefficients for every different q. The evaluation of the coefficients takes most of computing time.

3.2. Complexity

Now let's discuss the complexity of computing the upper and lower bounds.

To obtain the upper bound in theorem 1, we need to find a 3-independent set S_3 by recursively adding a node with the minimal degree and removing its neighbors and neighbors of the neighbors. The time complexity of this procedure is $O(n^2)$, where n is the number of nodes in the graph.

For example, from the 3-independent set S_3 we can obtain all the coefficients $f_1, f_2, \ldots, f_{r_3}$ and r_3 in order n time. Therefore, the total time complexity for computing the upper bound is $O(n^2)$.

For computing the lower bound, we must 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 $\Delta(G) < n$ disjointed paths between each pair of nodes and $n(n-1)/2 - m$ pairs of nonadjacent nodes resulting at most n^3 paths. Thus the time complexity for the lower bound is $O(n^5)$.

Therefore, both upper and lower bounds can be computed in polynomial time.

3.3. Accuracy of the Bounds

The accuracy of the bounds depends on the topologies of graphs and the failure probability of their nodes. There is no general solution for all graphs.

We will use the regular graphs as examples to study the accuracy of our bounds. We first apply the bounding formulas to the hypercube that is one of the most popular distributed system topologies as we mentioned before.

A p-dimension hypercube Q_p is a regular graph with degree p and the number of nodes is $n = 2^p$. The upper and lower bounds can be derived from theorems 1 and 2, respectively. The difference between the upper and lower bounds is employed to measure the accuracy of the bounding scheme. That is:

$$difference = \overline{R}(G) - \underline{R}(G)$$

where $\overline{R}(G)$ and $\underline{R}(G)$ are the upper bound and lower bound on $R(G)$, respectively.

Table 2 presents the numerical difference between the upper and the lower bounds for 6-hypercube Q_6 with the node failure probability q varies from 0.02 to 0.2. From this table, the difference increases as q increases. We have calculated the differences for other hypercubes and obtained similar results.

Now let look at the impact of the size of the distributed system. Table 3 shows the numerical results on the difference between the upper and the lower bounds for $q = 0.16$ with different hypercubes Q_p where p varies from 6 to 24. Figure 3 illustrated the differences graphically. It can be seen that the difference decreasing as p increasing.

Table 2. The difference of the bounds for Q_6: where q takes values from 0.02 to 0.2.

q	0.0200	0.0400	0.0600	0.0800	0.1000
difference	0.0000	0.0001	0.0014	0.0074	0.0259
q	0.1200	0.1400	0.1600	0.1800	0.2000
difference	0.0711	0.1650	0.3381	0.6300	0.9999

Table 3. The difference of the bounds for Q_p: where p takes values from 6 to 24.

p	6	8	10	12	14
difference	0.1650	0.1632	0.1390	0.1080	0.0789
p	16	18	20	22	24
difference	0.0551	0.0372	0.0245	0.0158	0.0100

Based on the above observation and our further analysis, we have the following more general results.

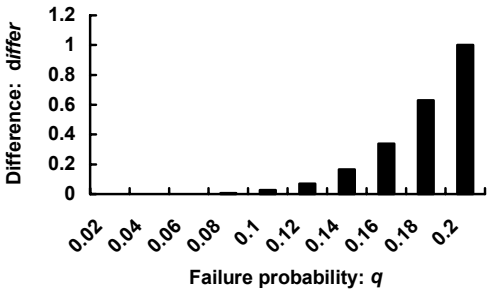
For hypercube Q_p, we couldn't prove that the difference between the upper and lower bounds tends to zero unconditional, but we can prove the following propositions:

Proposition 1: For the hypercube Q_p, if

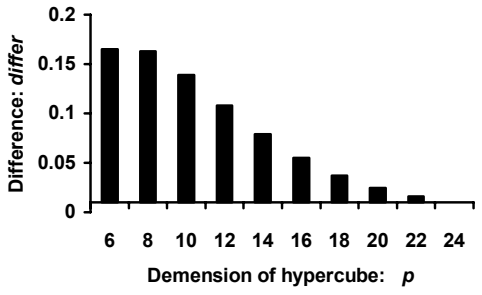
$q < 1 - \frac{1}{\sqrt[3]{2}} \approx 0.2063$, then

$difference(Q_p) \to 0$ as $n = 2^p \to \infty$.

Paying further attention to the fact that Q_p is regular with maximum degree $\Delta(Q_p) = p = \log n$ and the logarithmic base is 2, we can obtain the more general results for general graph G_n and they are presented in the following two propositions.



(a)



(b)

Figure 3. The difference between the upper and lower bounds on $R(Q_p)$

Proposition 2: Let G_n be any graph with n nodes and $\Delta(G) = O(\log n)$. If

$q \geq 0.2063$ and $\Delta(G_n) < 0.4391 \log n$, then

$difference(G_n) \to 0$ as $n \to \infty$.

Note, $-1/\log\left(1 - \frac{1}{\sqrt[3]{2}}\right) \approx 0.4391$.

Proposition 3: Let G_n be any graph with n nodes, and $\Delta(G_n) \leq c \log n$. For any given $q \in (0, 1)$, if

$c < -1/\log q$, then

$difference(G_n) \to 0$ as $n \to \infty$.

These propositions indicate that the bounding scheme have following strength and weakness.

The strength of the bounding scheme is that the larger the distributed system is, the higher the accuracy of the bounding scheme will be. Thus, for very large distributed systems, these bound schemes are very accurate.

However, for small and middle-sized distributed systems the bounds may not be accuracy. Further more, the bounding scheme cannot be used in the system where nodes have different failure probabilities, or have an extremely high failure probability, e.g., $q > 0.2063$. To overcame these problems, we have developed a Combing Monte Carlo simulation scheme to investigate the probabilistic point estimator and confidence interval on $R(G)$ in the next section.

4. COMBINED MONTE CARLO SIMULATION

4.1. Simulation Design

The Monte Carlo method for estimating residual connectivity reliability $R(G)$ consists of sampling random system states and counting the number of residual subsystem (subgraph) that are connected.

Given a graph G with node set $V = \{v_1, v_2, \ldots, v_n\}$ in which each node v_i fails independently with a probability q_i. Let X_i be the random state of v_i, $i = 1, 2, \ldots, n$, where $X_i = 1$ if v_i is operating, and $X_i = 0$ otherwise. In fact, X_i is a Bernoulli random variable and its probability density function (*pdf*) is as following

X_i	1	0
Probability	$1 - q_i$	q_i

Let Y represent the connectedness status of the residual subgraph, where $Y = 1$ if the residual sub-graph is connected and $Y = 0$ otherwise. This gives $Y = f(X_1, X_2, \ldots, X_n)$.

As an example, let's consider two special cases: $Y = 1$ if $X_1 = X_2 = \ldots = X_n = 1$, and $Y = 0$ if $X_1 = X_2 = \ldots = X_n$

= 0. The *pdf* of Y is then:

Y	1	0
Probability	$R(G)$	$1 - R(G)$

Notice that $R(G)$ is the expected value of the random variable Y, i.e.

$$R(G) = \mathrm{E}(Y) = \mathrm{E}(f(X_1, X_2, \ldots, X_n)) .$$

Obviously, $R(G)$ would be evaluated if $f(X_1, X_2, \ldots, X_n)$ were known. The problem is that $f(X_1, X_2, \ldots, X_n)$ is very difficult to be evaluated in general cases.

Monte Carlo simulation scheme is usually used to obtain a probabilistic point estimator or confidence interval on $E(Y)$. The process of the simulation generally consists of following tasks.

Sampling random states $(X_1, X_2, \ldots, X_n)$ over the sampling distribution: Let $(X_1, X_2, \ldots, X_n)_k$ be the samples, where $k = 1, 2, \ldots, N$, and N is the number of samples. Please note that this capital N is different from the little n which is the number of nodes in the system.

Counting $Y_k = f((X_1, X_2, \ldots, X_n)_k)$, $k = 1, 2, \ldots, N$.

Evaluating the mean value $\overline{Y}$ of $Y_1, Y_1, \ldots, Y_N$, using the formula $\overline{Y} = \frac{1}{N}(Y_1 + Y_2 + \cdots Y_N)$.

Since $\overline{Y}$ is an unbiased estimator of $\mathrm{E}(Y)$, $\overline{Y}$ is employed to estimate $\mathrm{E}(Y)$. According to the definition of $R(G)$, $\overline{Y}$ is also an unbiased probabilistic point estimator of $R(G)$. If we use $\hat{R}$ to denote the unbiased probabilistic point estimator, we have

$$R(G) \approx \hat{R} = \overline{Y} = \frac{1}{N}(Y_1 + Y_2 + \cdots Y_N).$$

According to this formula, all we need to do to estimate $R(G)$ is to count the number of possible connected residual subgraphs.

The next question is how accurate is the estimator? A meaningful measure of the quality of an estimator is to use the relative error

$$\frac{\left|\hat{R} - G(G)\right|}{R(G)}$$

Then we can further define a (ε, α) Monte Carlo scheme, which estimates the probability that the relative error is greater than ε is at most α, that is

$$\mathrm{Prob}\left\{\frac{\left|\hat{R} - R(G)\right|}{R(G)} > \varepsilon\right\} \le \alpha, \text{ or}$$

$$\mathrm{Prob}\left\{\frac{\left|\hat{R} - R(G)\right|}{R(G)} \le \varepsilon\right\} \ge 1 - \alpha$$

This results in the (ε, α) confidence interval of $R(G)$:

$$R(G) \in \left[\frac{\hat{R}}{1+\varepsilon}, \quad \frac{\hat{R}}{1-\varepsilon}\right] \tag{4}$$

Now we can discuss what the suitable number of samples N should be. An N that is sufficient to guarantee the (ε, α) confidence is given by the Center Limit Theorem as

$$N \ge \frac{1 - R(G)}{R(G)} \times \frac{\left(u_{1-\alpha/2}\right)^2}{\varepsilon^2} \tag{5}$$

This formula actually gives a lower bound on N. Unfortunately, this lower bound depends on $R(G)$ that is the unknown factor we are trying to estimate using N number of samples. However, this formula can be used if we can find a lower bound $\underline{R}$ on $R(G)$. Then we have

$$N \ge \underline{N} = \frac{1 - \underline{R}}{\underline{R}} \times \frac{\left(u_{1-\alpha/2}\right)^2}{\varepsilon^2} \tag{6}$$

After these discussions we now are in the position to propose our combined Monte Carlo simulation scheme as follows:

Step 1. Obtain the deterministic lower bound $\underline{R}$ on $R(G)$ by employing the bounding scheme defined in section 2.

Step 2. Compute the lower bound $\underline{N}$ from formula (6) for given (ε, α) confidence parameters.

Step 3. Sample random states $(X_1, X_2, \ldots, X_n)_k$, $k = 1, 2, \ldots, N$, according to their probability density function, and then evaluate $Y_1, Y_2, \ldots, Y_N$, where the number of samples is $\underline{N}$.

Step 4. Calculate the mean value $\overline{Y}$ and obtain the probabilistic point estimator $\hat{R}$, as well as the confidence interval using formula (4) for $R(G)$.

Note:

(1). This scheme can be used for the case where nodes have different failure probabilities, whereas the deterministic bounding scheme cannot handle this case, as discussed in section 3. This is because we can sample random states $(X_1, X_1, \ldots, X_n)$ for any *pdf*

(Prob$\{X_i = 0\} = q_i$) in Step 3 above.

(2). If we couldn't obtain the deterministic lower bound $\underline{R}$, (for the reason that, say, nodes have different failure probabilities or the deterministic lower bounds $\underline{R}$ is too small to calculate $\underline{N}$), then we have to use trial-and-error method. We can try to use

$$\underline{N} = \frac{(u_{1-\alpha/2})^2}{\varepsilon^2}$$

which is the special case when $R(G) = 0.5$. If the samples result in an estimator $\hat{R} \geq 0.5$, we are luck because the current $\underline{N}$ is sufficient for producing the (ε, α) confidence interval according to formula (4). On the other hand, if the samples result in $\hat{R} < 0.5$, the estimated $\underline{N}$ is not big enough. But all we need to do now is to reevaluate $\underline{N}$ one more time according to formula (6), this time with $\underline{R}$ replaced $\hat{R}$.

4.2. Simulation Results.

Now let's apply the bounding scheme and combined Monte Carlo scheme to the two kinds of distributed systems, the circles and the hypercubes as we have discussed in section 2. All computations take confidence parameter $(\varepsilon, \alpha) = (0.025, 0.05)$.

We use the following relative error to measure the quality of bounding scheme and simulation scheme.

$$E_r = \frac{\left|\hat{R} - G(G)\right|}{R(G)}$$

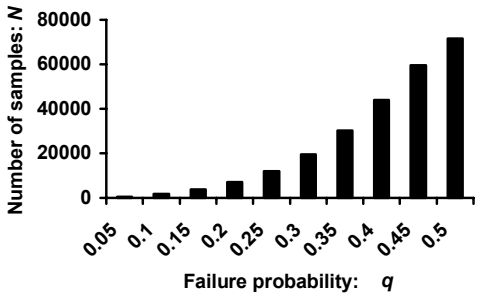
First, we study the circle C_n. Table 4 summarizes all data we evaluated and Figure 4(a) presents the number of samples that we must use to satisfy the (ε, α) confidence interval where $\underline{R}$ takes the accurate value $R(C_{10})$ as shown in Figure 2. The relative errors E_r is also shown in Figure 4(b). It can be seen from Table 4 that 9 out of 10 point estimators satisfy the relative error control, only one doesn't (the number in bold italic). This result indicates that the confidence interval is suitable.

Table 4: $R(C_{10})$: Accurate value R, # of samples $\underline{N}$, simulation value $\hat{R}$, and relative errors E_r

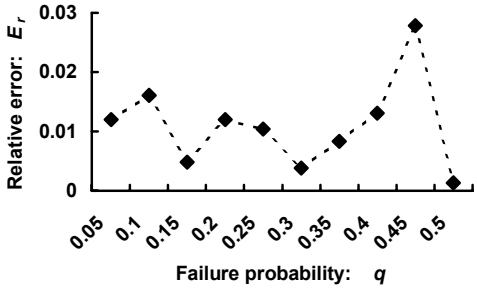
q	0.0500	0.1000	0.1500	0.2000	0.2500
R	0.9314	0.7845	0.6187	0.4653	0.3378
$\underline{N}$	453	1688	3788	7063	12049
$\hat{R}$	0.9426	0.7719	0.6217	0.4597	0.3413
E_r	0.0120	0.0161	0.0048	0.0120	0.0104
q	0.3000	0.3500	0.4000	0.4500	0.5000
R	0.2399	0.1694	0.1223	0.0936	0.0791
$\underline{N}$	19475	30138	44111	59522	71560
$\hat{R}$	0.2408	0.1680	0.1207	0.0910	0.0790
E_r	0.0038	0.0083	0.0131	***0.0278***	0.0013

Now, let's study the hypercube Q_p. In order to get the relative errors, we use $\hat{R}$, the simulation estimator of $R(Q_6)$ for $N = 38146$, to replace $R(Q_6)$, it is shown in the last row of Table 5 (in italic). Figure 5 shows the relative error of the lower bounds $\underline{R}$ for $R(Q_6)$ where node failure probability varies from 0.10 to 0.19.

Figure 6 Shows the number of samples obtained from formula (6) based on $\underline{R}$, as well as the relative error of the simulation point estimator. It can be seen that all the relative error are very small: less than 0.001, much smaller than the required accuracy level $\varepsilon = 0.025$.



(a)



(b)

Figure 4. Relative errors for simulation results for $R(C_{10})$.

The confidence interval for $R(C_{10})$ and $R(Q_6)$ can be

obtained from formula (4) that is omitted here.

Table 5: $R(Q_6)$: lower bound $\underline{R}$, number of samples $\underline{N}$, simulation value $\hat{R}$, and relative errors E_r

q	0.10	0.11	0.12	0.13	0.14
$\underline{R}$	0.9741	0.9560	0.9288	0.8897	0.8349
E_r	0.0258	0.0439	0.0711	0.1100	0.1648
$\underline{N}$	163	283	471	762	1215
$\hat{R}$	1.0000	1.0000	1.0000	1.0000	1.0000
E_r	0.0001	0.0001	0.0001	0.0003	0.0003
$\hat{R}$	*0.9999*	*0.9999*	*0.9999*	*0.9997*	*0.9997*
q	0.15	0.16	0.17	0.18	0.19
$\underline{R}$	0.7605	0.6618	0.5336	0.3700	0.1647
E_r	0.2391	0.3376	0.4658	0.6293	0.8348
$\underline{N}$	1935	3141	5373	10467	31168
$\hat{R}$	1.0000	0.9987	0.9996	0.9989	0.9973
E_r	0.0005	0.0004	0.0008	0.0007	0.0002
$\hat{R}$	*0.9995*	*0.9991*	*0.9988*	*0.9982*	*0.9971*

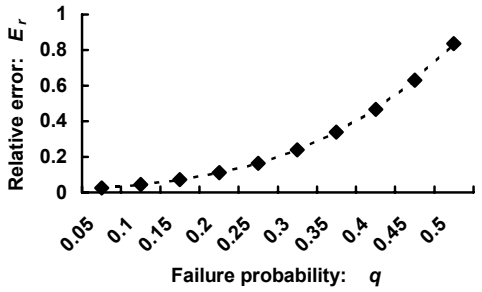
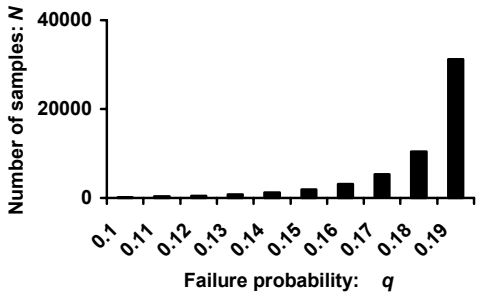
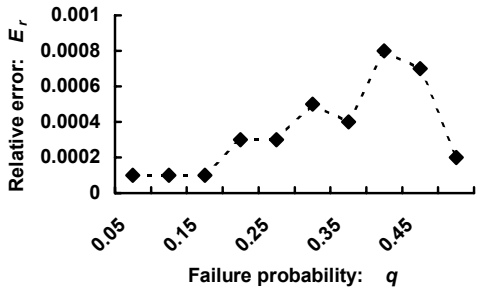


Figure. 5 Relative errors for bounding results on $R(Q_6)$.



(a)



(b)

Figure 6: Relative errors for simulation results on $R(Q_6)$.

5. CONCLUSIONS

As a part of our dependable distributed system project, we studied the residual connectedness reliability (RCR). We developed in this paper a deterministic bounding scheme and a probabilistic simulation scheme to obtain the approximation values of RCR.

The deterministic bounding scheme computes an upper and a lower bound in polynomial time. The advantages of the bounding scheme are twofold:

(1). The larger the distributed system is, the higher the accuracy of the bounding scheme will be. This is particular useful for very large systems where it would be impossible to obtain the accurate value of RCR.

(2). The time consuming part of computing the bounds is to compute the coefficients in the bounding polynomial on $R(G)$. However, according to Theorems 1 and 2, the coefficients in our bounding polynomial are invariant with respect to failure probability q. Therefore, if we have a different failure probability q, we only need to reevaluate the polynomial without re-computing the coefficients for every different q.

The disadvantages of the bounding scheme also have two sides:

(1). The bounding scheme doesn't suit the distributed systems with different node failure probabilities.

(2). The smaller the distributed system is, the worse the accuracy of the bounding scheme will be.

For the very small system, we may obtain the accurate value of RCR by a brute-force algorithm. However, for the systems between very small and very large, none of those methods work. We thus developed a new scheme that combined our deterministic approach with a Monte Carlo simulation scheme.

The combined scheme can produce the probabilistic point estimator and confidence interval for RCR. The scheme can also handle distributed systems with different node failure probabilities.

The motivation of this research was to support our dependable distributed system project. We have applied this research result to support the design of our dynamic task reallocation, fault-tolerant protocols and system reconfiguration after fault detection. On the other hand, this work also contributes to the well-studied residual connectedness reliability in graph and networking theory. Our deterministic scheme and the combined deterministic and probabilistic approach produce a good approximation for RCR that can be used in general study in graphs and computer networks.

REFERENCES

1. Ball M.O. 1980, "Complexity of network reliability computations", *Networks*. 10. pp.153-165.
2. Boesch F. T. 1986, "Synthesis of reliable networks – A survey", *IEEE Trans. On Reliability*, R-35. pp.240-246.
3. Chen Y., Galpin V., Hazelhurst S., Mateer R. and Mueller C. 1999, "Modeling software development of a decentralized virtual service redirector for Internet applications". *In Proc. the 7th IEEE Workshop on Future Trends of Distributed Computing Systems*, Cape Town. pp.235 - 241.
4. Chen Y. and He Z. 2001, "Dependability Modeling of homogeneous and heterogeneous Distributed systems". *In the Proc. of the Fifth International Symposium on Autonomous Decentralized Systems*, Dallas. pp.176 - 183.
5. Chen Y. and Mateer R. 2001, "Performance Simulation of a Dependable Distributed System, Simulation". *Special Issue on Modeling and Simulation Applications in Scheduling Multiprocessor Systems*, Vol.77, No.5-6. pp.230-237.
6. Colbourn C.J. Satyanarayana A, Suffel C. and Sutner K. 1993, "Computing residual connectedness reliability for restricted networks", *Discrete Applied Mathematics*, 44. pp.221-232.
7. Doty L.L. 1989, "Extremal connectivity and vulnerability in graphs", *Networks*, 19. pp.73-78.
8. Goldschmidt O., Jaillet P. and LaSota R. 1994, "On reliability of graphs with node failures", *Networks*, 24. pp.251-259.
9. Liu S., Cheng K. and Liu X. 2000, "Network reliability with node failures", *Networks*, 35. pp109-117.
10. Pradhan D. K. and Reddy S. M. 1982, "A fault-tolerant communication architecture for distributed systems", *IEEE Trans. On Computer*, 31. pp.863-870
11. Smith D. 1984, "Graphs with the smallest number of minimum cut sets", *Networks*, 14. pp.47-61.
12. Smith D. 1990, "On the construction of optimally reliable graphs", *Networks*, 20. pp723-729.
13. Somani A. K. and Peleg O. 1996, "On diagnosability of large fault sets in regular topology-based computer systems", *IEEE Trans. On Computer*, 45. pp.892-903.
14. Sutner K., Satyanarayana A., and Suffel C. 1991, "The complexity of the residual node connectedness reliability problem", SIAM, J. Comput. 20. pp.149-155.
15. Valiant L.G. 1979, "The complexity of enumeration and reliability problems", *SIAM J.Comput*. 8. pp. 410-421.

BIOGRAPHY

Zhongshi He received his BSc and MSc in applied mathematics from Chongqing University, Chongqing, and his PhD in theoretical computer science from the same university in 1996. From 1999 to 2001, he had been a visiting researcher at the University of Witwatersrand, Johannesburg. Currently he is an associate professor at the College of Computer Science, Chongqing University. His research interests include fault-tolerant computing, system reliability analysis, graph theory and mathematical modeling. He has published 4 books and over 30 technical papers.

Yufang Tian received her BSc and MSc in applied mathematics from Chongqing University, Chongqing in 1987 and 1990, respectively. Currently she is a lecturer at the College of Mathematics and Physics, Chongqing University. Her research interests include operations research, mathematical modeling. She has published 4 books and several technical papers.

Yinong Chen received his doctoral degree from the University of Karlsruhe, Germany, in 1993. He did postdoctoral research in Germany in 1994 and in France in 1995 and 1996. From 1994 to 2001, he was a faculty member in the School of Computer Science and the leader of the Research Programme for Highly Dependable Systems at the University of the Witwatersrand, Johannesburg. Since 2001, he has been a faculty member in the Computer Science and Engineering Department, Arizona State University. Dr. Chen's research interests are in the areas of fault tolerant computing, software testing, distributed systems and computer networks. He has published a number of books and over 50 technical papers in these areas. Dr. Chen was an NRF (South African National Research Foundation) rated researcher from 1996 and was the principal investigator of a number of NRF and university projects.