Bounds on the Reliability of Distributed Systems with Unreliable Nodes & Links*

Yinong Chen, Member IEEE

Computer Science and Engineering Department, Arizona State University, Tempe, AZ 85287-8809. Email: yinong@asu.edu

Zhongshi He

College of Computer Science, Chongqing University, Chongqing 400044, P.R. China. Email: zshe@cqu.edu.cn

Keywords - Distributed system, probabilistic network, residual connectedness reliability, complexity.

Summary & Conclusions - The reliability of distributed systems & computer networks in which computing nodes and/or communication links may fail with certain probabilities have been modeled by a probabilistic network. Computing the residual connectedness reliability (RCR) of probabilistic networks under the fault model with both node & link faults is very useful, but is an NP-hard problem. Up to now, there has been little research done under this fault model. There are neither accurate solutions nor heuristic algorithms for computing the RCR. In our recent research, we challenged the problem, and found efficient algorithms for the upper & lower bounds on RCR. We also demonstrated that the difference between our upper & lower bounds gradually tends to zero for large networks, and are very close to zero for small networks. These results were used in our dependable distributed system project to find a near-optimal subset of nodes to host the replicas of a critical task.

Acronyms†

RCR	Residual Connectedness Reliability
UB	Upper Bound
LB	Lower Bound
NEF	Node-and-Edge Fault

Notations

G	graph
$R(G)$	residual connectedness reliability of graph G
n	number of nodes in graph
q_0	node failure probability
q_1	edge failure probability
$\|A\|$	cardinality of set A

* This work is partly supported by the National Science Foundation of China under the grant number 60173060.

† The singular and plural of an acronym are always spelled the same.

$\lfloor \cdot \rfloor$	greatest integer lower bound (floor)
$\lceil \cdot \rceil$	least integer upper bound (ceiling)
E	event
$\overline{E}$	complement event of E
$N_G(v)$	neighbor node set of node v in graph G
$\overline{R}(G)$	upper bound of $R(G)$
$\underline{R}(G)$	lower bound of $R(G)$
ΔR	difference between $\overline{R}(G)$ and $\underline{R}(G)$, i.e. $\Delta R = \overline{R}(G) - \underline{R}(G)$,
Q_p	p-dimension hypercube
$H_{k,\,n}$	order n Harary graph
S_n	order n Star graph
C_n	order n Circle
K_n	order n Complete graph

I. Introduction

The behavior of a distributed system can be 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 the RCR $R(G)$, which is the probability that the residual nodes can communicate with each other [2], [3], [4], [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 probability q_0.

• **Edge fault model:** The nodes of a graph are perfectly reliable, but the edges fail independently with probability q_1.

• **Node-and-edge fault model:** Edges & nodes fail independently of each other, with node & edge failure probabilities equal to q_0 & q_1, respectively.

The analysis problem is to find $R(G)$ for a given graph G. That is:

Input: Probabilistic graph G, component failure probability q_0 and/or q_1.

Output: RCR $R(G)$.

For all these three fault models, it has been shown that the analysis problems are all NP-hard [1], [5], [6], [7]; that is, there exists no efficient algorithms for computing $R(G)$.

To cope with this problem, various heuristic algorithms have been developed to estimate $R(G)$.

We have carefully studied all the related papers we could find. There are quite a number of papers dealing with approximation algorithms for estimating $R(G)$ under the edge fault model [8], [9], [10],

[11], [12], and under the node fault model [13], [14]. Colbourn [13] proposed a polynomial algorithm of certain restricted classes of graphs, including trees, series-parallel graphs, and permutation graphs. Colbourn & Chen [14] developed efficient algorithms of arbitrary graphs, and bound expressions for estimating $R(G)$. To our best knowledge, little work has been done under the NEF model. The only work known to us is from Frank & Gaul [2], who proposed the bounds to $R(G)$ for the complete graph.

Ball [1] has shown that the following point estimate reliability analysis problem is also NP-hard.

The *point estimate reliability analysis problem* is to find a single estimate value r of reliability $R(G)$, such that the value $r/R(G)$ lies between two given positive numbers.

In this paper, we study the reliability bounds for arbitrary graphs under the NEF model. We find good UB & LB expressions, and efficient algorithms for bounding the reliability $R(G)$. These expressions are simple & easy to understand. The algorithms are highly accurate. In fact, we demonstrate that the difference between the UB & LB gradually tends to zero. As an example application, we apply the bounding algorithms to the dependable distributed system which we developed for secure internet applications. We used these algorithms to find the best-suitable subgraphs to host critical tasks.

The paper is organized as follows. In Section II, we introduce our approach to the UB & LB of reliability $R(G)$, and derive the bound expressions. In Section III, we present the corresponding efficient bounding algorithms. In Section IV, we apply our algorithms to typical classes of graphs, for example, complete graphs, star graphs, hypercube, and Harary graphs; where we analyze the accuracy of the bounding algorithms. Theoretical & numerical results are also derived in this section. The results show that the difference between the UB & LB gradually tends to zero for large networks, and is very close to zero for small networks. As an application, we derive an algorithm for finding a subgraph with near-optimal RCR in Section V. Section VI concludes the paper.

II. Bounding the Reliability

Our approach to obtain bounds with efficient running time & high accuracy is directly based on the original definition of the RCR:

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

where $\Pr\{A\}$ stands for the probability of random event A. Based on the relationship & the operation of the random events, a method for estimating reliability bounds is proposed, and accurate bounds of the reliability are obtained. We have successfully used this approach in bounding $R(G)$ under the node fault model [14]. In the rest of this paper, we will study $R(G)$ under the NEF model.

A. The upper bound

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

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

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

Let E_i be the event that v_i is isolated in the induced graph, $i = 1, 2,..., n$. Then the occurrence of any $E_1, E_2, \ldots, E_n$ implies that E does not occur. Consequently we have

$$\overline{E} \supseteq \bigcup_{i=1}^{n} E_i .$$

Before we give the bounding expressions, we need the following definitions.

A ***k*-independent set** in a graph G, denoted by S, is such a subset of nodes that each pair of nodes has the distance of at least k in graph G, where $k \geq 2$.

A **maximal *k*-independent set**, denoted by S_k, is a k-independent set such that any extra node adding to set S_k will result in that it will no longer be a k-independent set.

Then, the UB of reliability $R(G)$ is given in the following theorem.

Theorem 1: Let S_3 be a maximal 3-independent set, and $r_3 = |S_3|$. If we renumber the nodes in S_3 so that $S_3 = \{u_i: i=1,\ldots, r_3\}$, and use $N_G(u_i)$ to denote the neighboring set of node u_i in G, then

$$R(G) \leq \prod_{i=1}^{r_3} \left(1 - (1 - q_0)\left(1 - (1 - q_0)(1 - q_1)\right)^{f_i}\right)$$

where $|S|$ is the cardinality of set S, $f_i = |N_G(u_i)|$, and q_0 & q_1 are the node & edge failure probabilities, respectively.

Proof: Let F_i be the event that v_i is isolated in the remaining 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_i does not fail, and for each neighbor node w of v_I, either w or edge wv_i fails. That is,

$$\Pr\{F_i\} = (1 - q_0)\left(1 - (1 - q_0)(1 - q_1)\right)^{|N_G(v_i)|} = (1 - q_0)\left(1 - (1 - q_0)(1 - q_1)\right)^{f_i}, \; i = 1, \ldots, r_3 .$$

Furthermore, $F_1, \ldots, F_{r_3}$ are independent of each other, so

$$1 - R(G) = \Pr\{\overline{E}\} \geq \Pr\{\bigcup_{i=1}^{r_3} F_i\} = 1 - \Pr\{\bigcap_{i=1}^{r_3} \overline{F_i}\} = 1 - \prod_{i=1}^{r_3} \Pr\{\overline{F_i}\}$$

$$= 1 - \prod_{i=1}^{r_3} \left(1 - (1-q_0)(1-(1-q_0)(1-q_1))^{f_i}\right)$$

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

$$R(G) \leq \prod_{i=1}^{r_3} \left(1 - (1-q_0)(1-(1-q_0)(1-q_1))^{f_i}\right).$$

If we let δ stand for the maximum degree of nodes in graph G, then

$(1 + \delta + \delta(\delta - 1))r_3 \geq n$, and $r_3 \geq n / (\delta^2 + 1)$.

This expression then induces the following conclusion:

Corollary 1:

$$R(G) \leq \left(1 - (1-q_0)(1-(1-q_0)(1-q_1))^{\delta}\right)^{n/(\delta^2+1)}.$$

B. The lower bound

To derive the lower bound, we need to first define this event: Let $E_d(u, v)$ be the event:

"For a pair of nodes u & v with distance $d(u, v) = d$, there is another path between u & v. This path can be partitioned into a number of subpaths with their length less than or equal to $d - 1$. Each subpath has a fault-free starting node & a fault-free ending node (excluding u & v), where $d \geq 2$".

Figure 1 shows an example with $d = 4$.

Let's examine the simplest case, when $d = 1$. $E_1(u, v)$ is the event: "For the adjacent ($d = 1$) nodes u & v, there is another path between u & v in which all the inner nodes & edges in the path are fault-free".

Based on the event $E_d(u, v)$, we can further define three notations:

$$E_d = \bigcap_{(u,v):d(u,v)=d} E_d(u,v),\ d \geq 1,$$

$$E = \bigcap_{d \geq 1} E_d \text{ , and}$$

X is the number of fault-free nodes in graph G, i.e. X is a random variable.

Subpaths, each of them with length $l \le d-1=3$

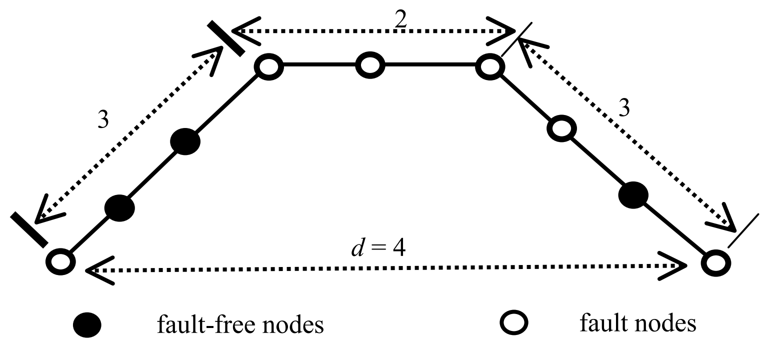

Figure 1. The path joining u & v is partitioned into subpaths.

Based on these definitions & notations, we then have the following lemma:

Lemma 1: If the random event $E \cap \{X \ge 2\}$ occurs, then the subgraph induced by the surviving nodes & edges is connected.

Proof: The occurrence of the random event $E \cap \{X \ge 2\}$ implies that there are at least two surviving nodes. We need to prove that for any pair of fault-free nodes u & v, there is a path joining u & v by fault-free nodes & edges.

We use induction with the distance $d = d(u, v)$ between node pair u & v.

1) If $d(u, v) = 1$, both u & v are fault-free, the occurrence of E implies that E_1 occurs, which means the lemma stands.

2) Assume that for every pair of fault-free nodes u & v with distance $d(u, v) < k$, the lemma holds; we prove that the lemma holds for any pair u & v with $d(u, v) = k$.

Clearly, for the occurrence of E_k where $k \ge 2$, the following fact is implied: There is a path joining u & v. The path can be partitioned into subpaths with distance not more than $k-1$, and each of them starts & ends with fault-free nodes. According to the assumption, each subpath can be replaced by a path consisting of fault-free nodes & edges. Then we obtain a path joining u & v by fault-free nodes & edges, which implies that the lemma holds.

The lemma holds with combinations of the above facts.

From this lemma, we have the following theorem.

Theorem 2: The LB of $R(G)$ is given by

$$R(G) \ge 1 + (n-1)\,q_0^n - n\,q_0^{n-1} - \sum_{d \ge 1} \sum_{d(u,v)=d} p(d, u, v).$$

where $p(d, u, v)$ is given as follows

$$p(1, u, v) = \prod_{j\geq 1}\left(1-(1-q_0)^{j-1}(1-q_1)^{j}\right)^{n_j} ,$$

$$p(2, u, v) = \prod_{j\geq 2}(1-(1-q_0)^{j-1})^{n_j} ,$$

$$p(3,u,v) = (q_0^2)^{n_3}\,(q_0(1-(1-q_0)^2))^{n_4}\,((1-(1-q_0)^2)^2)^{n_5}\,(1-(1-q_0^2)^2(1-q_0))^{n_6};$$

and for $d \geq 4$,

$$p(d,u,v) = (q_0^{d-1})^{n_d}\,(q_0^{d-2}(1-(1-q_0)^2))^{n_{d+1}}\,(q_0^{d-3}(1-(1-q_0)^2)^2)^{n_{d+2}} \times$$

$$(1-(1-q_0^2)^2(1-q_0^{d-2}))^{n_{d+3}}$$

where $n_j = n_j(u, v)$ stands for the number of j-distance paths connecting u with v, in which all paths are disjoined, $j \geq d = d(u, v)$.

Proof: From the original definition of $R(G)$ & Lemma 1, we have

$$R(G) \geq \Pr\{E \cap \{X \geq 2\}\} = \Pr\{E - \{X < 2\}\} \geq \Pr\{E\} - \Pr\{X < 2\}$$

$$= \Pr\{E\} - \Pr\{X = 0\} - \Pr\{X = 1\}.$$

Obviously,

$$\Pr\{X = 0\} = q_0^n,\ \Pr\{X = 1\} = n\,(1 - q_0)\,q_0^{n-1},$$

and

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

Then

$$R(G) \geq 1 + (n-1)\,q_0^n - n\,q_0^{n-1} - \sum_{d\geq 1}\sum_{d(u,v)=d}\Pr\{\overline{E_d(u,v)}\}.$$

To get the UB of $\Pr\{\overline{E_d(u,v)}\}$ for $d \geq 1$, four separate cases need to be proven.

Case 1: $d = 1$. Namely u & v are adjacent; all paths between u & v are inner node-disjoined; there are n_j paths with distance $j \geq 1$and $n_j \geq 0$. With the occurrence of event $\overline{E_1(u,v)}$, the following fact is implied: for each path joining u & v, if either at least one of the $j - 1$ inner nodes fails or at least one of the j edges fails, the probability is $1-(1-q_0)^{j-1}(1-q_1)^j$. Then

$$\Pr\{\overline{E_1(u,v)}\} \le \prod_{j\ge 1}\left(1-(1-q_0)^{j-1}(1-q_1)^j\right)^{n_j}$$

Case 2: $d = 2$. Namely $d(u, v) = 2$. With the occurrence of event $\overline{E_d(u,v)}$, i.e. $\overline{E_2(u,v)}$, each path connecting u, v must contain at least one failed inner node, and the probability is $1-(1-q_0)^{j-1}$. Then

$$\Pr\{\overline{E_d(u,v)}\} \le \prod_{j\ge 2}(1-(1-q_0)^{j-1})^{n_j} = p(2, u, v).$$

Case 3: $d = 3$. According to the paths shown in Figure 2, we can obtain

$$\Pr\{\overline{E_d(u,v)}\} \le (q_0^2)^{n_3}\,(q_0(1-(1-q_0)^2))^{n_4}\,((1-(1-q_0)^2)^2)^{n_5}\,(1-(1-q_0^2)^2(1-q_0))^{n_6},$$

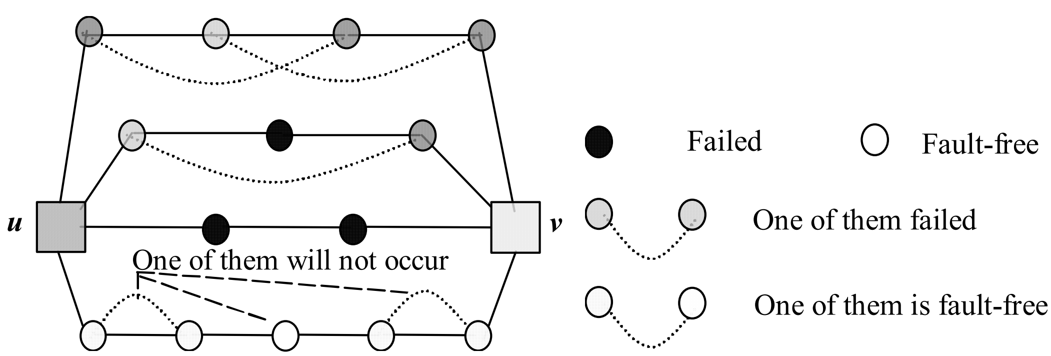

Figure 2: The state of the inner nodes in the paths.

Because $d = 3$, the event that $\overline{E_d(u,v)}$ will occur, is the event that $\overline{E_3(u,v)}$ will occur, as it is shown in Figure 2. This implies that all the paths between u & v must satisfy the following facts respectively:

1. For every length-3 path, its two inner nodes have failed, and the probability is q_0^2.
2. For every length-4 path, its 2nd inner node has failed, while one of the 1st & the 3rd inner nodes has failed, so the probability is $q_0(1-(1-q_0)^2)$.
3. For every length-5 path, either its 1st or 3rd inner node has failed, while one of the 2nd & 4th

inner node has failed, thus the probability will be $(1-(1-q_0)^2)^2$.

4. For every length-6 path, the five inner nodes must satisfy that the following three events don't occur simultaneously: (i) the 3rd node is fault-free, (ii) either the 1st or 2nd node is fault-free, and (iii) either the 4th or 5th node is fault-free. Then the probability is $1-(1-q_0^2)(1-q_0)(1-q_0^2)$.

Case 4: $d \geq 4$. This is similar to case 3. We can obtain

$$\Pr\{\overline{E_d(u,v)}\} \leq (q_0^{d-1})^{n_d} (q_0^{d-2}(1-(1-q_0)^2))^{n_{d+1}} (q_0^{d-3}(1-(1-q_0)^2)^2)^{n_{d+2}}$$

$$(1-(1-q_0^2)^2(1-q_0^{d-2}))^{n_{d+3}}$$

According to the proof above, Theorem 2 holds.

III. Algorithms for Reliability Bounds

Based on the constructive method for estimating the bounds above, we can obtain algorithms which estimate the reliability bounds. According to Theorems 1 & 2, the bigger the extreme 3-independent set, the better the UB is; and the higher the number of short paths between any pair of nodes, the better the LB is.

A. Preparation

Before we derive the algorithms, we need the following procedures to find the maximal 3-independent set, and as many short paths as possible.

Finding a maximal 3-independent set

```
Procedure Independent-Set3(G)
Input: graph G
Output: a maximal 3-independent set of graph G
begin
   S ← Φ;
   V ← V(G);
   while (H ≠ Φ) do
   begin
u ← the node of V with minimum degree in G;
S ← S ∪ {u};
for every v of  N_G(u), H ← H – (N_G(v) ∪ {v});
   end
   return (S);
end
```

Finding short paths

The purpose of this procedure is to find as many disjoined paths as possible with length not more than $d(u, v) + k$, where k is a given positive integer.

Procedure Short-Paths(G, u, v, k)

Input: graph G, a pair of vertices u & v of G, and a given positive integer k

Output: $d(u, v)$, j, n_j

/ * paths between u & v and with length not exceeding $d(u, v)+k$. n_j paths with length j*/

```
begin
   P ← the shortest path between u & v in graph G
   d ← | P | – 1;    /* | P | is the cardinality of P */
   n_d ← 1;
   d(u, v) ← d;
   H ← G;
   for (i = 1 to k) n_{d+i} = 0;
   while (d(u, v) ≤ d + k) do
   begin  /* delete inner nodes of P, then find the shortest path */
H ← (H – P) ∪ {u, v};
P ← the shortest path between u & v of H;
j ← | P | – 1;
n_j ← n_j + 1;
d(u, v) ← j;
      end
   return (d, for (j =0 to k ) n_{d+j});
end
```

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

B. Algorithm for the upper bound

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

Algorithm Upper-Bound

Input: graph G, the node & edge failure probabilities q_0 & q_1

Output: $\overline{R}(G)$ / * $\overline{R}(G)$ according to Theorem 1 * /

begin

   $\overline{R}(G) \leftarrow 1$;

   $S_3 \leftarrow$ Independent-Set3(G);

   for (every node v_j of S_3)

$\overline{R}(G) \leftarrow \overline{R}(G) \times \left(1-(1-q_0)\left(1-(1-q_0)(1-q_1)\right)^{f_j}\right)$;

end

C. Algorithm for the lower bound

The following algorithm finds the LB $\underline{R}(G)$ of $R(G)$.

Algorithm Lower-Bound

Input: graph G, the node failure probability q_0

Output: $\underline{R}(G)$ /* $\underline{R}(G)$ according to Theorem 2 */

```
begin
  n ← order of graph G;
  k ← 3 ;
  R(G) ← 1 + (n – 1) q0^n – n q0^(n–1);
  for (every node pair u & v in G)
  do
  begin
       (d, for (j =0 to k) n_{d+j} ) ← Short-Path(G, u, v, k);
       R(G) ← max{0, R(G)}
       R(G) ← R(G) – p(d, u, v);
  end
end
```

D. The efficiency of the algorithms

For the UB, the main computational process is to find an extreme 3-independent node set in the procedure *Independent-Set3*. 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 complete algorithm for UB takes $O(n^2)$ operations.

For the LB, we need to process the procedure *Short-Paths* for every pair of nodes to find node-disjoined short paths. Every short path needs $O(n^2)$ operations, and there are at most n paths, then this procedure takes $O(n^3)$ operations for every pair of nodes, and we have $n(n-1)/2 < n^2$ pair of nodes. This results in a total of $O(n^5)$ operations. It takes $O(n^2)$ operations to calculate LB directly from the formula in Theorem 2, thus the entire operational time complexity for the LB is then $O(n^5)$.

This analysis shows that the complexity of the algorithms is highly polynomial but still efficient. There are no better algorithms yet known.

IV. Numerical Accuracy of the Bounds

In this section, we demonstrate theoretically & numerically that the proposed bounding algorithms are highly accurate.

Let $\Delta R(G)$ be the difference between the UB & LB of reliability $R(G)$ obtained from our algorithms. We apply our bounds to the following seven classes of graphs widely studied in research:

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

Both UB & LB of $R(G)$ will be studied. It will be shown that the difference between the bounds tends to zero when the size of the network tends to infinity. In other words, the difference can be smaller than any positive number ε for any failure probability q as long as the size of the graph is sufficiently large.

Notice that the key of the approach is to obtain the maximal 3-independent set for the UB, and to find as many short paths as possible for LB.

A. Bounds on some typical graphs

The following theorems are derived based on the characteristics of the seven classes of graphs mentioned above.

The reliability bounds of the first five classes of graphs are given in Theorem 3, and those of the last two are given in Theorems 4 & 5, respectively.

Theorem 3:

P: $R(P) \ge 1 + 9q_0^{10} - 10q_0^9 - 30q_0(2q_0 - q_0^2)^2 - 15q_1\left(1-(1-q_0)^3(1-q_1)^4\right)^2$,

$$R(P) \le 1-(1-q_0)\left(1-(1-q_0)(1-q_1)\right)^3;$$

P$_n$: $R(P_n) \ge 1 + (n-1)q_0^n - nq_0^{n-1} - \dfrac{n(q_0 - q_0^{n-1})}{1-q_0} + \dfrac{(2-q_0-nq_0^{n-2}+(n-1)q_0^{n-1})q_0}{(1-q_0)^2} - (n-1)q_1$,

$$R(P_n) \le \left(1-(1-q_0)\left(1-(1-q_0)(1-q_1)\right)\right)^2\left(1-(1-q_0)\left(1-(1-q_0)(1-q_1)\right)^2\right)^{n/3-2};$$

S$_n$: $R(S_n) \ge 1 + (n-1)q_0^n - nq_0^{n-1} - \dbinom{n-1}{2}q_0 - (n-1)q_1$,

$$R(S_n) \le 1-(1-q_0)\big(1-(1-q_0)(1-q_1)\big);$$

$$\boldsymbol{C_n}: R(C_n) \ge 1+(n-1)q_0^n - nq_0^{n-1} - \frac{n(1-q_0^{n/2})q_0}{1-q_0} - nq_1\big(1-(1-q_0)^{n-2}(1-q_1)^{n-1}\big),$$

$$R(C_n) \le \Big(1-(1-q_0)\big(1-(1-q_0)(1-q_1)\big)^2\Big)^{(n-2)/3};$$

$$\boldsymbol{K_n}: R(K_n) \ge 1+(n-1)q_0^n - nq_0^{n-1} - \binom{n}{2} q_1\big(1-(1-q_0)(1-q_1)^2\big)^{n-2},$$

$$R(K_n) \le 1-(1-q_0)\big(1-(1-q_0)(1-q_1)\big)^{n-1}.$$

The proof can be given by applying Theorems 1 & 2 to the seven classes of graphs.

The p-dimension hypercube Q_p is a well-known regular graph with degree p & order $n = 2^p$. The following lemma holds for this graph.

Lemma 2: For Q_p, the following relationships hold

1) Diameter $D(Q_p) = p$.

2) The number of node-pairs in Q_p with distance d is $n_d = 2^{p-1}\binom{p}{d}$.

3) For every node-pair u & v in Q_p with distance $d(u, v) = d \ge 1$, there are exactly p paths connecting u & v with disjointed inner nodes, in which d paths are of length d, and $p - d$ paths are of length $p + 2$.

According to Lemma 2, Corollary 1, & Theorem 2 derived before, the bounds of reliability for the hypercube are given below.

Theorem 4: For Q_p with $n = 2^p$ nodes, the following relationship holds

$$R(Q_p) \ge 1+(n-1)q_0^n - nq_0^{n-1} - \frac{1}{2}pnq_1\big(1-(1-q_0)^2(1-q_1)^3\big)^{p-1} - \frac{1}{2}\binom{p}{2} nq_0^2(1-(1-q_0)^3)^{p-2} -$$

$$\frac{1}{2}n(2-q_0)^{2p}q_0^{-p}\sum_{d=3}^{p}\binom{p}{d}(q_0^p(2-q_0)^{-2})^d \text{, and}$$

$$R(Q_p) \le \Big(1-(1-q_0)\big(1-(1-q_0)(1-q_1)\big)^p\Big)^{n/(p^2+1)}$$

For the elementary Harary graph $H_{k,n}$, $H_{k,n} = C_n(1, 2, \ldots, k / 2)$ is a connected circulate graph if k is even. Otherwise, $H_{k,n}$ consists of $C_n(1, 2, \ldots, \lfloor k/2 \rfloor)$ by adding $\lceil n/2 \rceil$ far diagonal edges.

To find the bounds of the Harary graph, we first need to find as many short paths joining node u & v as possible. The following lemma helps to find the number of short paths for the Harary graph.

Lemma 3: For the $H_{k,\,n}$ graph, if k is even & denoted by $k = 2r$, then:

1). Diameter $D(H_{k,n}) = \lceil \lfloor n/2 \rfloor / r \rceil$.

2). If the number of node-pairs with distance d is denoted by n_d, $d \geq 1$, then $n_d \leq rn$.

3). There are at least r paths with length less than or equal to $d + 1$ for every d-distance node-pair u & v.

According to Lemma 3, and Theorems 1 & 2, the following theorem holds, which gives us the reliability bounds of the Harary graph:

Theorem 5: For an $H_{k,n}$ graph, let $r = \lfloor k/2 \rfloor$; $k < n$, then

$$R(H_{k,n}) \geq 1 + (n-1)q_0^n - nq_0^{n-1} - rn\left(1-(1-q_0)(1-q_1)^2\right)^r - \frac{rn(1-(1-q_0)^2)^r}{1-q_0^r},$$

$$R(H_{k,n}) \leq \left(1-(1-q_0)\left(1-(1-q_0)(1-q_1)\right)^k\right)^{n/(k^2+1)}.$$

B. Numerical difference for small networks

In Section IV.A, we derived the bound expressions of different classes of graphs. In this section, we demonstrate the numerical accuracy of these bounds by studying their numerical differences; that is, the difference $\Delta R(G)$, or ΔR for short, between the UB & LB:

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

In the following subsections, we show curves which represent ΔR. The failure probabilities q_0 & q_1 are assumed to vary from 10^{-1} to 10^{-6}, and the number of nodes n from 2 to 64, respectively.

1) Hypercube Q_p

The differences ΔR for hypercube Q_p are shown in Figures 3, 4, & 5. It is evident that, for realistic failure probability values smaller than 10^{-2}, ΔR decreases sharply close to zero with the decrease of q_0 or q_1.

2) Harary graph $H_{k,n}$

The differences ΔR for $H_{k,n}$ are shown in Figures 6, 7, & 8.

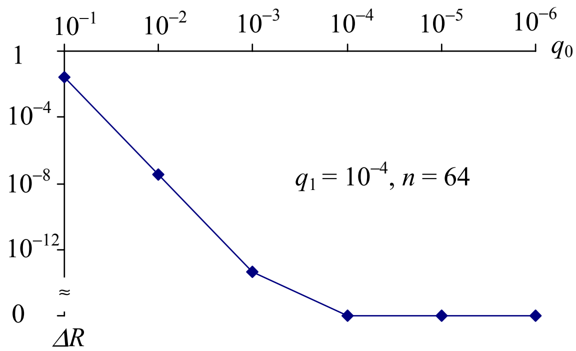

Figure 3: The difference *ΔR* with regard to q_0.

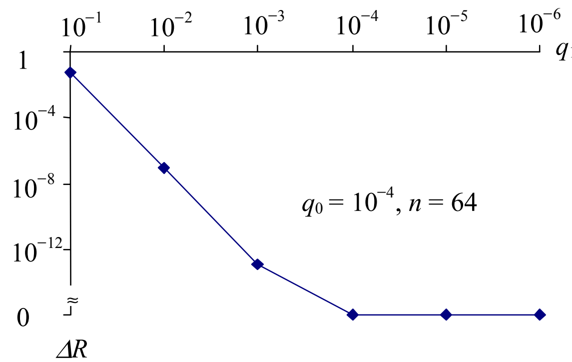

Figure 4: The difference *ΔR* with regard to q_1.

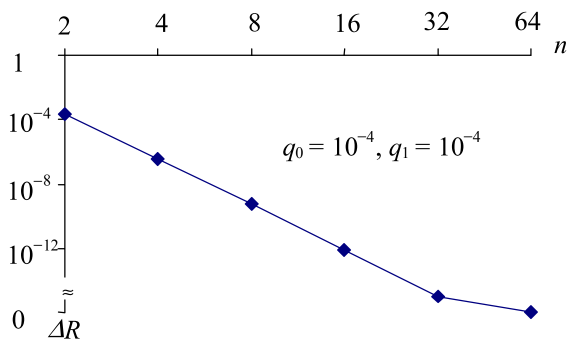

Figure 5: The difference of *ΔR* with regard to *n*.

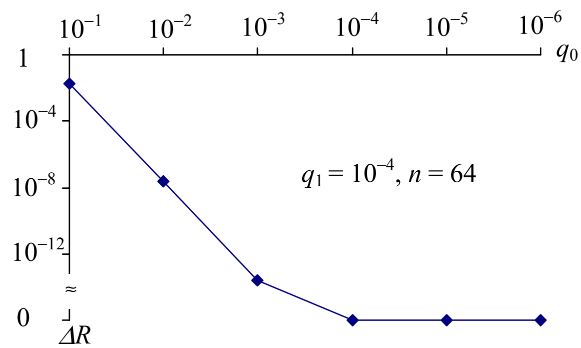

Figure 6: The difference *ΔR* with regard to q_0.

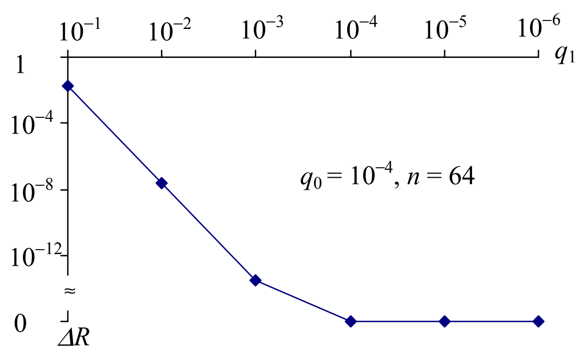

Figure 7: The difference *ΔR* with regard to q_1.

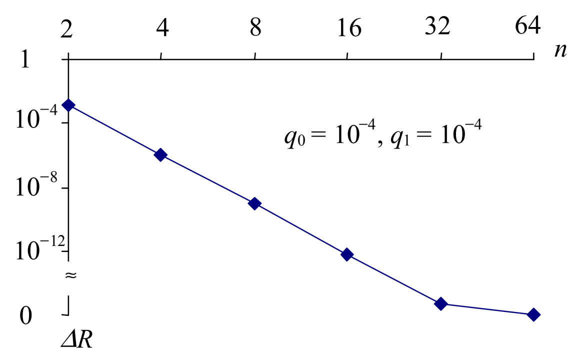

Figure 8: The difference *ΔR* with regard to *n*.

We can also define the average difference between UB & LB, denoted by $\overline{\Delta}R$, where $[q_0, q_1]$ are in the area $[a, b; c, d]$; namely

$$\overline{\Delta}R = \frac{\int_a^b \int_c^d \Delta R \mathrm{d}q_0 \mathrm{d}q_1}{(b-a)(d-c)}.$$

We find that $\overline{\Delta}R$ behaves similarly to ΔR.

According to the numerical results above, the difference between UB & LB is close to zero if the component failure probability is reasonably small, which means that the bounds are highly accurate.

C. Gradual difference for large networks

The discussion in Section IV.B is limited to small networks. In this section, we discuss the gradual difference of bounds for large graphs. We show that the difference ΔR gradually tends to zero.

1) Path, Star, Circle, and Complete graphs

According to the results derived in Section IV.A, and the fact that

$$R(S_n) = (1-q_0)\,(1-(1-q_0)q_1)^{n-1} - (1-q_0)\,q_0^{n-1},$$

we can immediately derive the following results:

Theorem 6: For graphs P_n, S_n, C_n, & K_n, the following results stand. For all $q_0 \in (0, 1)$ & $q_1 \in (0, 1)$;

$\Delta R(P_n) \to 0$,

$\Delta R(S_n) \to 0$,

$\Delta R(C_n) \to 0$,

$\Delta R(K_n) \to 0$ as $n \to \infty$.

2) Hypercube

Theorem 7: For the hypercube Q_p, if q_0 & q_1 satisfy one of the following two conditions:

i) $q_0 + q_1 - q_0q_1 > 0.5$,

ii) $q_0 < 1 - \frac{1}{\sqrt[3]{2}} = 0.2063$, & $(1-q_0)^2(1-q_1)^3 > 0.5$,

then $\Delta R(Q_p) \to 0$ as $n = 2^p \to \infty$.

Proof: As $n = 2^p$, n approaches infinity as p tends to infinity.

i) If $q_0 + q_1 - q_0q_1 > 0.5$, then $2(q_0 + q_1 - q_0q_1) > 1$, which implies that

$$\left\{-(1-q_0)\frac{(2(q_0+q_1-q_0q_1))^p}{p^2+1}\right\} \to -\infty \text{ as } n \to \infty.$$

According to Theorem 4,

$$\overline{R}(Q_p) = \left(1-(1-q_0)(q_0+q_1-q_0q_1)^p\right)^{n/(p^2+1)} \le \exp\left\{-(1-q_0)\frac{(2(q_0+q_1-q_0q_1))^p}{p^2+1}\right\}$$

Consequently, $\overline{R}(Q_p) \to 0$, which implies that $\Delta R(Q_p) \to 0$ as $n \to \infty$.

ii) Now let's prove that $q_0 < 1-\frac{1}{\sqrt[3]{2}}$, and $(1-q_0)^2(1-q_1)^3 > 0.5$.

First, let $x = 1-(1-q_0)^3$; then $x < 0.5$, and

$$\binom{p}{2} n q_0^2 (1-(1-q_0)^3)^{p-2} \le p^2 q_0^2 x^{-2}(2x)^p \to 0 \text{ as } p \to \infty.$$

Second, let $y = q_0^p(2-q_0)^{-2}$, and

$$y_d = \binom{p}{d}\left(q_0^p(2-q_0)^{-2}\right)^d, \quad d = 3, 4, \ldots, p.$$

For $d = 3$, we have

$$y_3 = \binom{p}{3} q_0^{3p}(2-q_0)^{-6} \to 0 \text{ as } p \to \infty.$$

For $d \ge 3$, we have

$$\frac{y_d}{y_{d-1}} = \frac{p+1-d}{d} y \le py,$$

then

$$y_d \le (py) y_{d-1} \le \cdots \le (py)^{d-3} y_3.$$

Furthermore

$$\sum_{i=4}^{p} y_i \le \sum_{j=1}^{p-3} (py)^j = \frac{1-(py)^{p-3}}{1-py} py \cdot y_3$$

Clearly $py \to 0$ as $p \to \infty$, then

$$\frac{1-(py)^{p-3}}{1-py}\, py \to 0\text{; note that } y_3 \to 0 \text{ as } p \to \infty.$$

Consequently,

$y_4 + y_5 + \ldots + y_p = o(y_3)$

which means that $(y_4 + y_5 + \ldots + y_p) / y_3 \to 0$ as $p \to \infty$. Meanwhile

$$n(2-q_0)^{2p} q_0^{-p} y_3 \le 2^p (2-q_0)^{2p} q_0^{-p} p^3 q_0^{3p} (2-q_0)^{-6}$$

$$= p^3 (2-q_0)^{-6} (2\,(1-(1-q_0)^2)^2)^p \to 0 \text{ as } p \to \infty.$$

Then we have

$$n\,(2-q_0)^{2p} q_0^{-p} \sum_{d=3}^{p} \binom{p}{d} (q_0^p\, (2-q_0)^{-2})^d = n\,(2-q_0)^{2p} q_0^{-p}\, (y_3 + \sum_{d=4}^{p} y_d)$$

$$= n\,(2-q_0)^{2p} q_0^{-p}\, y_3 + o\,(n\,(2-q_0)^{2p} q_0^{-p}) \to 0 \text{ as } p \to \infty .$$

Finally, let $z = 1 - (1-q_0)^2 (1-q_1)^3$. Clearly $2z < 1$ holds when $(1-q_0)^2 (1-q_1)^3 > 0.5$. Then

$$pnq_1\left(1-(1-q_0)^2 (1-q_1)^3\right)^{p-1} = pq_1 z^{-1} (2z)^p \to 0 \text{ as } p \to \infty .$$

In combination with Theorem 4, $\underline{R}\,(Q_p) \to 1$ holds, which implies that $\Delta R(Q_p) \to 0$ as $n \to \infty$.

3) Harary graph

By using the result of Theorem 5, we obtain the following theorem.

Theorem 8: For the Harary graph $H_{k,\,n}$, if q_0 & q_1 satisfy one of the following two cases:

i) $q_0 + q_1 - q_0 q_1 > 0.707$,

ii) $q_0 < 1 - 1/\sqrt{2} = 0.293$ & $(1-q_0)(1-q_1)^2 > 0.5$,

then $\Delta R(H_{k,n}) \to 0$ as $n \to \infty$, where $r = \lfloor k/2 \rfloor = \log n$.

Proof: For $2r+1 \ge k \ge 2r$, $r = \log n$, $n = 2^r$, so both k & r approach infinity when $n \to \infty$.

i) If $q_0 + q_1 - q_0 q_1 > 0.707 = 1/\sqrt{2}$, then

$2(q_0 + q_1 - q_0 q_1)^2 > 1$,

which implies that

$$\left\{-(1-q_0)\frac{\left(2(q_0+q_1-q_0q_1)\right)^r}{p^2+1}\right\} \to -\infty \text{ as } r \to \infty.$$

From Theorem 6,

$$\overline{R}(H_{k,n}) = \left(1-(1-q_0)(q_0+q_1-q_0q_1))^k\right)^{n/(k^2+1)} \le \exp\left\{-(1-q_0)\frac{n(q_0+q_1-q_0q_1)^k}{k^2+1}\right\}$$

$$\le \exp\left\{-(1-q_0)\frac{\left(2(q_0+q_1-q_0q_1)^2\right)^r}{p^2+1}\right\}$$

Consequently, $\overline{R}(H_{k,n}) \to 0$, then $\Delta R(H_{k,n}) \to 0$ as $n \to \infty$.

ii) If $q_0 < 1 - 1/\sqrt{2} = 0.293$, then $(1 - q_0)^2 > 0.5$ holds, and

$$r\,n\,(\,1-(1-q_0)^2)^r/(1-q_0^r) \;= r\,(\,2\,(\,1-(1-q_0)^2)\,)^r/(1-q_0^r\,) \to 0\ (n\to\infty).$$

Furthermore, $(1-q_0)(1-q_1)^2 > 0.5$ implies that $1 - (1-q_0)(1-q_1)^2 < 0.5$; then we have

$$rn\left(1-(1-q_0)(1-q_1)^2\right)^r = r\left(2\left(1-(1-q_0)(1-q_1)^2\right)\right)^r \to 0\ (r\to\infty).$$

From Theorem 6, $\underline{R}(H_{k,n}) \to 1$ holds, then $\Delta R(H_{k,n}) \to 0$ as $n \to \infty$.

4) The relationship between the difference ΔR, and the failure probability.

Figures 9 & 10 show that, as n tends to infinity, the difference ΔR tends to zero, if the failure probability q_0 & q_1 are within the white area. The behavior of ΔR is undecided when the failure probability q_0 & q_1 are within the dotted area. Because q_0 & q_1 are normally less than 10^{-2}, our approach is useful practically.

However, if the dotted area becomes nil, it means that we can get almost accurate values of reliability $R(G)$ for any failure probabilities, which is important not only in practice, but also in a theoretical sense.

For a large network G, the difference between the UB & LB tends to zero for any failure probability values as the size increases. If we take $r = (\overline{R} + \underline{R})/2$ to be an estimate of $R(G)$, we can prove that $|r - R(G)| < \varepsilon$ will hold for any given positive number ε, or ΔR tends to zero.

Summing up the results studied above, we have:

• **Large networks:** the difference between the UB & LB tends to zero when the size of networks tends to infinity, as shown in Theorems 6, 7, & 8, and Figures 9, &10. What size of network is considered to be large? It depends on the requirement of the difference between the UB & LB, and the reliability. For example, for Harary graph, if q_0 and q_1 are less than 0.01, the size n should be less than 64, as shown in Figures 6, 7, and 8.

• **Small networks:** for a practical failure probability, say less than 0.1, the difference between the UB & LB is very small or close to zero (see Figures 3, 4, 5, 6, 7, & 8).

We also conducted the accuracy analysis for other networks, and obtained similar results.

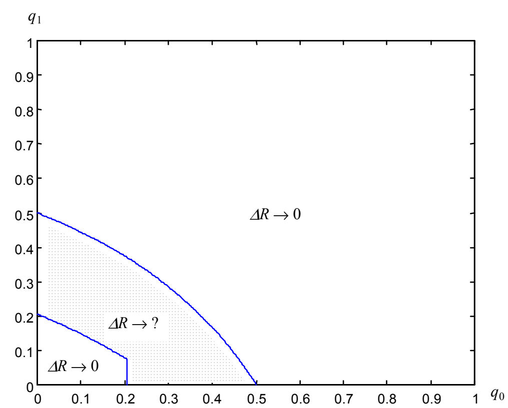

Figure 9: Hypercube: Difference between UB and LB.

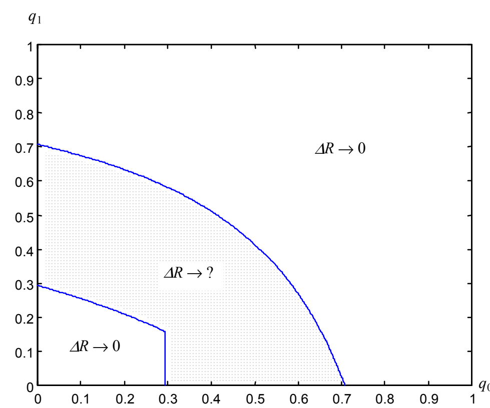

Figure 10: Harary graph: Difference between UB and LB.

V. Application

The results we obtained in previous sections can be directly used in the analysis of the dependability of distributed systems. In this section, we discuss the application of RCR analysis in our dependable distributed system project.

A. System overview

Over the past few years, we have developed a dependable distributed system, and its application to the Internet. The current research related to this project includes design, specification, verification, reliability evaluation, and implementation of a virtual service redirector, and a dependable firewall for critical Internet applications [15], [16].

The distributed system we developed consists of a network of computing nodes. The network can be bus-based, or have dedicated links. Here we consider the case where the computing nodes are connected by dedicated links into an arbitrary topology or a graph. The tasks running on the system are illustrated in Figure 11.

B. Task allocation

The overall system is shown in Figure 1. At the bottom layer, the system consists of a network of off-the-shelf computer nodes. In the prototype we implemented, the nodes used were Intel Pentium computers connected by Ethernet network cards. The operating system used was Linux.

The development station is used to develop programs, and to deploy the initial tasks onto the nodes in the system. Tasks, including independent tasks, parallel, and redundant copies of the identical tasks are allocated to different nodes in the system. In our experiment, we only implemented a firewall application, and thus all tasks are parallel & redundant copies of the firewall application. Nodes running firewall applications are FW nodes in the system. The incoming and outgoing packets are generated by two traffic generation nodes (TG) which simulate the two sides of the firewall, e.g., the Internet, and the Intranet. The results from redundant copies of firewalls will be checked by a fault-tolerant protocol.

Currently, we have re-implemented the prototype by simulation. We used Java threads to simulate modules of the system, and used the synchronized thread communication mechanisms to simulate the interactions among the modules.

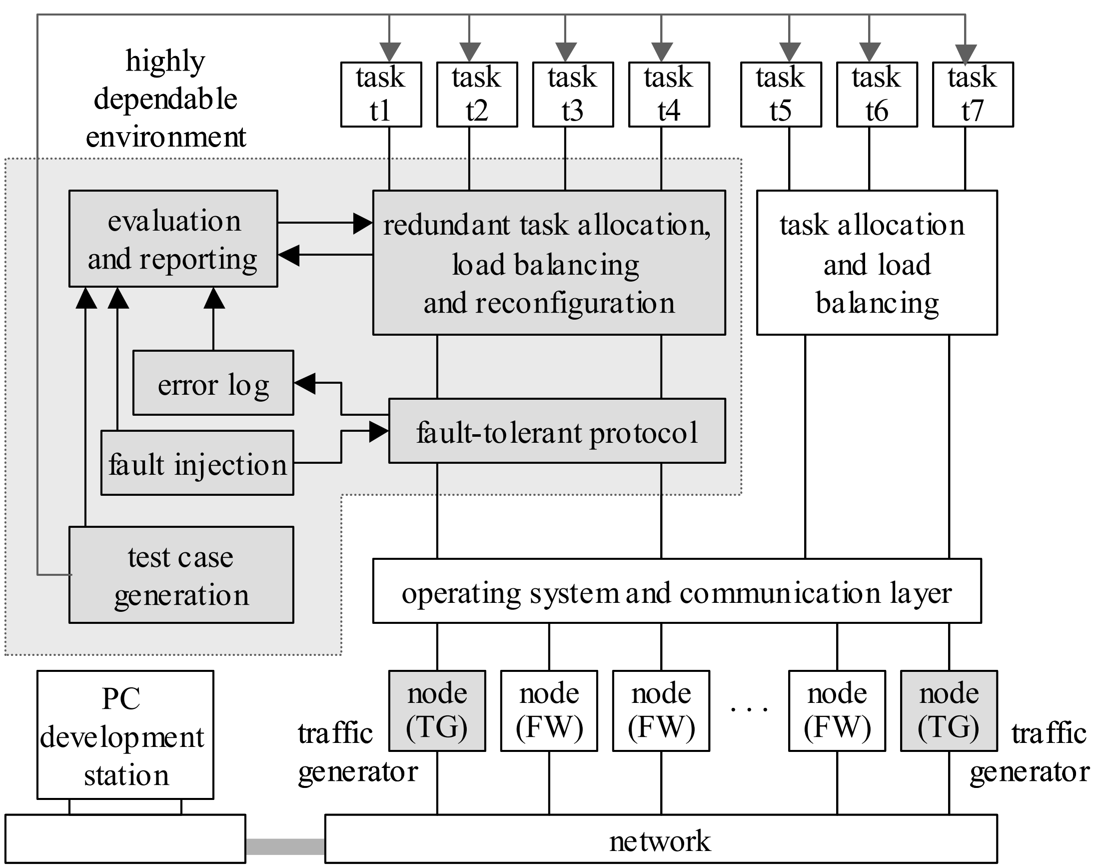

Figure 11: System overview.

In the simulation system, various applications, for example, firewall, mail server, web server etc., with different levels of criticality, can be deployed on the system. The higher the criticality is, the more copies (replicas) of the task will be run. For example, in Figure 12, task t_1 is allocated to three nodes c_1, c_2, & c_5; task t_2 is allocated to nodes c_2 & c_3; task t_3 is allocated to nodes c_3 & c_4; and task t_4 is allocated to node c_4. The fault-tolerant protocols will check the consistency among the replicas, and perform fault detection & isolation. Faulty nodes and links will be removed (isolated) from the system.

The policy of task allocation involves the following:

• If a node becomes faulty, task reallocation will be performed (the replicas of the tasks on the faulty node will be reallocated/recreated onto other working nodes), if the existing replicas of a critical task can no long communicate with each other.

• When tasks are reallocated, it must be ensured that all nodes have relatively balanced loads in terms of the number of tasks.

• replicas of a critical task must be allocated to such a subset of nodes in the system, that the least number of task reallocations are necessary.

• Communication among replicas should use the minimum number of steps if no direct link exists.

• Task reallocation must be completed within the certain deadlines set up by the applications.

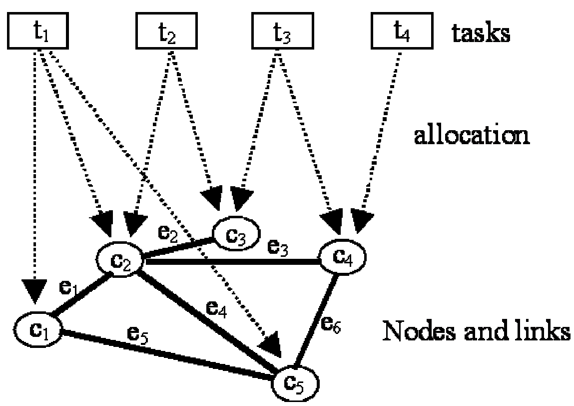

Figure 12: Task allocation.

To implement this task reallocation policy, we need to find a subgraph to host the replicas of a critical task in the give graph so that the subgraph can survive the maximum number of node or edge (link) faults without breaking the connectedness of the subgraph.

Computing $R(G)$ is a crucial part of this task reallocation policy. A heuristic algorithm with an adaptable time requirement, and a polynomial complexity, has been devised to find a near optimal subgraph for task reallocation as follows:

- Input: graph $G = (V, E)$, edge & node failure probabilities, integer m.
- Output: subgraph of G with m nodes, and RCR $R(G)$.

The algorithm is divided into two steps:

i. Constructing a rooted tree recursively:

  a. Root: the root is any one of the nodes with the maximum degree, numbered with 1;

  b. Children: the children are the adjacent nodes to the parent that are not numbered;

  c. Numbering: the children are numbered in decreasing degree. Nodes with equal degree can be numbered in any order.

ii. Choosing the first *m* nodes: The output subgraph is the subgraph induced by the first *m* nodes (numbered from 1 to *m*) in the rooted tree.

iii. Evaluate the deadline imposed by the semantics of the applications.

  a. If the deadline does not permit optimization, use the subgraph for task reallocation; Exit.

  b. Else estimate the upper & lower bounds on RCR and use the average of low and high;

  c. Determine the subgraph with the best RCR among the subgraphs obtained;

  d. If the deadline does not permit optimization, use the subgraph for task allocation; Exit.

  Else, go back to step i.a, and choose another node with the maximum degree.

The idea of this algorithm is to find a subgraph based on the star graph with as many edges as possible. The reason is, as studied in Theorem 3, that the more edges a star graph has, the higher the reliability will be. The star graph S_n is more reliable than other graphs with the same order & number of edges. However, because the algorithm is heuristic, it may not find the subgraph with the best RCR. To improve the result, we take an adaptive approach according to the deadline imposed by the application. If time permits, we will try to find multiple subgraphs, and choose the best one among them according to their RCR.

C. Task reallocation and reliability of critical task

Due to the occurrence of faults, tasks have to be reallocated from time to time. As shown in Figure 12, the most critical task in this example is task t_1, which is allocated to three nodes (c_1, c_2, & c_5). Consider the subgraph hosting task t_1, denoted by G_1, which is induced by the three nodes & related links. In fact, it is a 3-node complete graph. Considering the communication function, we assume that if the residual nodes can communicate with each other when some nodes or links fail, then we don't need a task-reallocation. Otherwise, we must reallocate task t_1.

We have used Markov chains to model the behaviors of distributed systems. The model changes its state when an event occurs. These events include the occurrence of node or link faults, and task reallocation. Currently, we are looking at the following questions:

- Which and how many task-reallocations are required?
- How long can a task survive? What is the reliability of a task?

In all states in which the residual nodes can communicate with each other, there is no need to perform task-reallocation. For example, if the failure of any node or any link does not disconnect the residual nodes, no reallocation is needed.

Apparently, the probability of states in which no task-reallocation is needed is $R(G_1)$, or the RCR of G_1. Thus we can obtain the probability of task t_1 surviving from beginning to the end as:

$$\Pr\{\text{task } t_1 \text{ surviving}\} = R(G_1) + (1 - R(G_1))\, R(G_2) + \ldots + (\prod(1 - R(G_i)))\, R(G_n)$$

where G_i, i = 1, 2, …, n-1, is the graph hosting task t_1 after $(i\text{-}1)^{th}$ task-reallocation. It is a research question to find the maximum number n of reallocatons for a given graph, which depends on the structure of the original graph & the reallocation algorithm used to choose the location of subgraph G_i.

References

[1] M.O.Ball, "Complexity of network reliability computations", *Networks.* 10, 1980, pp.153-165.

[2] O.Frank and W.Gaul, "On reliability in stochastic graphs", *Networks,* 12, 1982, pp.119-126.

[3] T.Evans and D.Smith, "Optimally reliable graphs for both edge and vertex failures", *Networks*, 16, 1986, pp.199-204.

[4] F.T.Boesch, "Synthesis of reliable networks – A survey", *IEEE Transactions on Reliability* R-35, 1986, pp.240-246.

[5] L.G.Valiant, "The complexity of enumeration and reliability problems", *SIAM Journal of Computing* 8, 1979, pp. 410-421.

[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* 12, 1983, pp. 777-788.

[7] K.Sutner, A.Satyanarayana, and C.Suffel, "The complexity of the residual node connectedness reliability problem", *SIAM, Journal of Computing* 20, 1991, pp.149-155.

[8] J.S.Provan, "Bounds on reliability of networks", *IEEE Transactions on Reliability, R*-35, 1986, pp. 260-268.

[9] T.B.Brecht and C.J.Colbourn, "Improving reliability bounds in computer networks", *Networks*, 16,1986, pp.369-380.

[10] C.J.Colbourn and D. D.Harms, "Bounding all-terminal reliability in computer networks", *Networks,* 18, 1988, pp.1-12.

[11] H.M.AboElFotoh and C.J.Colbourn, "Efficient algorithms for computing the reliability of permutation and interval graphs ", *Networks,* 20, 1990, pp.883-898.

[12] L.D.Nel and C.J.Colbourn, "Combing Monte Carlo estimates and bounds for network reliability", *Networks,* 20, 1990, pp.277-298.

[13] C.J.Colbourn, A. Satyanarayana, C.Suffel and K.Sutner, "Computing residual connectedness reliability for restricted networks", *Discrete Applied Mathematics,* 44, 1993, pp.221-232.

[14] Z. He. Y. Chen. Efficient algorithm for residual connectedness reliability of distributed system. *Proceedings of the first international workshop on distributed computing, communication and applications*. IEEE Computer Society, Islamabad, Pakistan, May 2000, pp. 151-159.

[15] Y. Chen, V. Galpin, S. Hazelhurst, R. Mateer, and C. Mueller, "Modelling software development of a decentralised virtual service redirector for Internet applications", *The 7th IEEE Workshop on Future Trends of Distributed Computing Systems*, Cape Town, December 1999, pp. 235 - 241.

[16] Y. Chen, Z. He. Evaluating a dependable distributed system with multiple critical tasks. *Proceedings of the first international workshop on distributed computing, communication and applications*. IEEE Computer Society, Islamabad, Pakistan, May 2000, pp. 116-120.

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 & 1996. From 1994 to 2000, he was a faculty member in the School of Computer Science, and was the founder & the leader of the Research Program for Highly Dependable Systems at the University of the Witwatersrand, Johannesburg. He joined the faculty of Computer Science & Engineering Department at Arizona State University in 2001. Dr. Chen's research interests are in the areas of fault tolerant computing, software testing, computer architecture, distributed systems, and computer networks. He (co-) authored 4 books and, over 50 technical papers in these areas.

Zhongshi He received his BSc & MSc in applied mathematics from Chongqing University, Chongqing; and his Ph.D. in theoretical computer science from the same university in 1996. He is currently a professor in the College of Computer Science at Chongqing University. Dr. He was a visiting researcher in the Program for Highly Dependable Systems at the University of the Witwatersrand, Johannesburg, from October 1999 to September 2001. His research interests include fault-tolerant computing, system reliability analysis, graph theory, and mathematical modelling. He has published 4 books, and over 30 technical papers.