Malaysian Journal of Library & Information Science, Vol.6, no.2, December 2001:101-118

AN EFFECTIVE ALGORITHM FOR OPTIMAL K-TERMINAL RELIABILITY OF DISTRIBUTED SYSTEMS

Chin-Ching Chiu [1]; Yi-Shiung Yeh and **Jue-Sam Chou**
[1]Department of Management Information System
Private Takming College, Taipei, Taiwan, R.O.C.
Institute of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
e-mail: chiu@imd.takming.edu.tw

ABSTRACT

Distributed system provides a cost-effective means of enhancing a computer system's performance in areas such as throughput, fault-tolerance, and reliability optimization. Consequently, the reliability optimization of a distributed system has become a critical issue. A K-terminal reliability is defined as the probability that a specified set, K, of nodes is connected in a distributed system. A K-terminal reliability optimization with an order (the number of nodes in K-terminal) constraint problem is to select a K-terminal of nodes in a distributed system such that the K-terminal reliability is maximal and possesses sufficient order. It is evident that this is an NP-hard problem. This paper presents a heuristic method to reduce the computational time and the absolute error from the exact solution. The proposed algorithm is based on not only a simple method to compute each node's weight and each link's weight, but also an effective objective function to evaluate the weight of node sets. Before appending one node to a current selected set, instead of computing the weight of all links and all nodes of each set, only the weight of a node, which is adjacent to the current selected set, and links between the node and the current selected set are accumulated. Then the proposed algorithm depends on the maximum weight to find an adequate node and assign it to the current selected set in a sequential manner until the order of K-terminal constraint is satisfied. Reliability computation is performed only once, thereby saving much time and the absolute error of the proposed algorithm from exact solution is very small.

Key words: heuristics distributed systems, reliability optimization

INTRODUCTION

The reliability problem of a distributed system (DS) with a general structure is NP-hard (Aggarwal et al., 1982; Irani and Khabbaz, 1982). Efficient algorithms easily implemented on a computer are needed to analyze the reliability of large networks. In addition, such algorithms should yield good approximations of the reliability when the networks are so large that the computational time becomes prohibitive.

The topology of a DS with n processing elements (nodes) and e communication links (links) can be characterized by an undirected graph G = (V, E) where V denotes a set of nodes, and E represents a set of links.

These DS topologies can be characterized by their DS reliability, message-delay, or network capacity. These performance characteristics depend on many properties of G that represent the DS topology (Stankovic, 1984; Kumar and Agrawal, 1993): the number of ports at each node (degree, say $d(v_i)$, of a node v_i), and the number of links. Notably, the number of links directly impacts the system's reliability.

For an undirected graph G the K-terminal reliability of G, $R(G_k)$, is the probability that the set K of nodes of G is connected in G. Two special and widely studied cases are those obtained for K = V and |K| = 2. For example, in Fig. 2, the former K-terminal is $K = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$; the latter K-terminal is one of $K = \{v_1, v_2\}$, $K = \{v_1, v_3\}, \ldots, K = \{v_7, v_8\}$. The first is the all-terminal reliability of G, the second is the 2-terminal reliability (Rai et al., 1987; Satyanarayana and Hagstrom, 1981; Torrieri, 1994). A typical network reliability problem is to compute $R(G_k)$. This is not easy since the reliability problem for a general network has been shown to be inherently difficult and very likely no efficient algorithm can be constructed for its solution. It has been shown that the network reliability with respect to a network with general structure is NP-hard (Aggarwal and Rai, 1981; Lin and Chen, 1992). Very recently it was also established that even for planar networks the computation of the K-terminal reliability, for $K \neq V$, is an NP-hard problem (Politof and Satyanarayana, 1986). The case for which K = V (all-terminal reliability) is at present undetermined for planar networks. The above results mean that it is unlikely that a polynomial algorithm to solve a general network reliability problem will be found.

This work largely focuses on how to compute nearly maximum system reliability subject to the order $K_{request}$ of nodes constraint. For a large DS on various topologies, our results demonstrate that the proposed algorithm is reliable and efficient.

PROBLEM DESCRIPTION

In this section, we describe the problem addressed herein to clarify our research objectives.

Notation and Definitions

Notations

$G=(V,E)$	an undirected DS graph where V denotes a set of processing elements, and E represents a set of communication links.	$w(G_k)$	the weight of G_k obtained by the object function.
		$d(v_i)$	the number of links connected to the node v_i.
n	the number of nodes in G, $n = \|V\|$.	$w(e_{i,j})$	the weight of the link $e_{i,j}$.
v_i	the i^{th} processing element or the i^{th} node.	$V_{adj(Gk)}$	a set of nodes which are adjacent to any node of G_k.
e	the number of links in G, $e = \|E\|$.	V_{Gk}	a set of nodes of G_k.
$e_{i,j}$	an edge represents a communication link between v_i and v_j.	$w(V_{G_k})$	$\sum w(v_j)$, where $v_j \in V_{G_k}$.
		E_{G_k}	a set of direct links between any two nodes in G_k.
$p_{i,j}$	the probability of success of link $e_{i,j}$.		
$q_{i,j}$	the probability of failure of link $e_{i,j}$.	$w(E_{G_k})$	$\sum w(e_{i,j})$, where $e_{i,j} \in E_{G_k}$.
$K_{requestt}$	order of nodes constraint in a DS.	v_s	a starting node for deriving a K-terminal.
G_k	the graph G with the set K of nodes specified, and $\|K\| \geq 2$.	$e_{s,j}$	the direct link $e_{s,j}$ does not exist, but there are at least two paths whose length is two between v_s and v_j.
$R(G_k)$	the reliability of K-terminal solution of a DS graph G.		
$y_{i,j}$	the number of paths whose length is two between v_i and v_j.	$V_{e(G_k)}$	a set of nodes which $e_{s,j}$ exist in G_k.
		$E_{e(G_k)}$	a set of $e_{s,j}$.
$w(v_i)$	the weight of the i^{th} node.	$w(e_{s,j})$	the weight of $e_{s,j}$.

Definitions

Definition 1. A K-terminal reliability (KTR) is defined as the probability that a specified set, K, of nodes is connected (where K denotes a subset of the set of processing elements V).

Definition 2. If G_k denotes the G with the set K, $|K| \geq 2$, of nodes specified, a node v_i is directly connected to a set V_{Gk}, a set of nodes of G_k, of nodes if and only if there is a link between v_i and a node in V_{Gk}.

Definition 3. A number of reliability computations (NRC) is the number of computations of a KTR that the order of G_k are equal to $K_{request}$.

Definition 4. Absolute error is defined as the value of subtracting an approximate solution from an exact solution of KTR; i.e. absolute error = $DSR_{opt} – DSR_{app}$, where DSR_{opt} denotes an optimal solution which obtained by running exhaustive search algorithm and DSR_{app} denotes an approximation solution which obtained by running heuristic algorithm.

Definition 5. Relative error is defined as the value of dividing an exact solution into the absolute error; i.e. relative error = $1 – (DSR_{opt} – DSR_{app})$.

Definition 6. The ratio of average relative error is defined as the value of dividing the summation of relative error by the number of the total simulation cases under consideration; i.e. the ratio of average relative error = $(\sum [1 – (DSR_{opt} – DSR_{app})])$ / (total simulation cases).

Definition 7. The random method is defined as the algorithm which select a set, K, with an order $K_{request}$ of nodes using random number generator under consideration.

PROBLEM STATEMENT

A KTR can be obtained using the sum of mutually disjoint events (Satyanarayana and Hagstrom, 1996; Lin et al., 1999). A K-terminal reliability problem can be characterized as follows:

Given

Topology of a DS.

The reliability of each communication link.

Assumption

Each node is perfectly reliable.

Each link is either in the working (ON) state or failed (OFF) state.

Constraint

The total nodes to be required.

Goal

To select a specified set, K, of nodes in a DS, by doing so, K-terminal reliability is adequate under an order $K_{request}$ constraints.

Restated, a set, K, of nodes can be derived from the given set V that constitutes a DS in that K-terminal reliability is adequate. The main problem can be mathematically stated as follows:
Object: Maximize $R(G_k)$
subject to: the order of K-terminal $G_k = K_{request}$
where $R(G_k)$ is defined as the reliability of K-terminal solution of a DS graph G.

Obviously, the problem for a large DS, as in a metropolitan area network, requires a large execution time. Herein, we develop an efficient method that allows the K-terminal reliability optimization in the DS to achieve the desired performance.

HEURISTIC ALGORITHM FOR K-TERMINAL RELIABILITY

In this section, we present a heuristic algorithm to maximize system reliability. The analyses performed herein assume that all of the nodes are perfect and the links are unreliable, i.e. each link is either in the working (ON) state or failed (OFF) state.

The Concept of Proposed Algorithm

As generally known, the exhaustive method spends long execution time in a large DS. The exhaustive method, an optimal solution, cannot effectively reduce the problem space. Occasionally, an application requires an efficient algorithm to compute the reliability due to its resource considerations. Under this circumstance, achieving optimal reliability may not be desirable. Instead, an efficient algorithm with an approximate reliability computation algorithm is highly attractive. The topologies of most DS are large and an increasing number of nodes cause the execution time for a solution to exponentially grow. Therefore, this work presents an algorithm capable of reducing the total execution time to achieve the sub-optimal KTR of a DS.

Consider a DS with n nodes and e links. Restated, the set K of nodes has the maximum reliability and its total nodes is equal to the order $k_{request}$ constraint.

The reliability of a set of selected nodes depends on their links and the link reliability. For any node, the degree of that node affects the number of paths of the information can be transferred from others' nodes. Therefore, in this work, we employed a simple means of computing the node weight, which takes less time and can quickly compute the weight of every node. The weight, say $w(v_i)$, of node v_i is formulate as:

$$w(v_i) = 1 - \prod_{z=1}^{d(v_i)} q_{i,k_z} \, . \qquad (1)$$

where

$q_{i,j}$ denotes the probability of failure of link $e_{i,j}$ that communicate node v_i and v_j

the index $k_z \in 1, ..., n$

$d(v_i)$ denotes the degree of node v_i

$2 \le d(v_i) \le n-1$

n represents the number of nodes in G, $n = |V|$.

The Eq. (1) is easily programmed and reduces many multiplicative operations. If the degree of v_i is $d(v_i)$, the weight of v_i can be computed in one subtraction and $d(v_i)$ multiplication. Thus, we can obtain the weight of every node in n subtractions and $2 \times e$ multiplication.

In the network, two nodes may contain many paths between them. A path's length is between one and n-1. To reduce the computational time, we consider the path in which the length is not greater than two. The following equation is used to evaluate the weight, say $w(e_{i,j})$, of link $e_{i,j}$.

$$w(e_{i,j}) = 1 - q_{i,j} \prod_{z=1}^{y_{i,j}} (q_{i,k_z} \times p_{k_z,j}) \, , \qquad (2)$$

Where

$p_{i,j}$ ($q_{i,j}$) denotes the probability of success (failure) of link $e_{i,j}$

the index $k_z \in 1, ..., n$

$y_{i,j}$ denotes the number of paths with length two between v_i and v_j

$y_{i,j} \le n - 2$.

The weight of $e_{i,j}$ can be computed in one subtraction and $2 \times (y_{i,j} + 1)$ multiplication. Thus, in the worst case, when the graph is a complete graph, we can obtain all of the weights of each link in $n \times (n-1)/2$ subtractions and $n \times (n-1) \times (n-2)/2$ multiplication.

In the same manner, if $e_{i,j}$ denotes no direct link exits, but $y_{i,j}$ is greater than two between v_i and v_j, the following equation is used to evaluate the weight, say $w(e_{s,j})$, of $e_{i,j}$

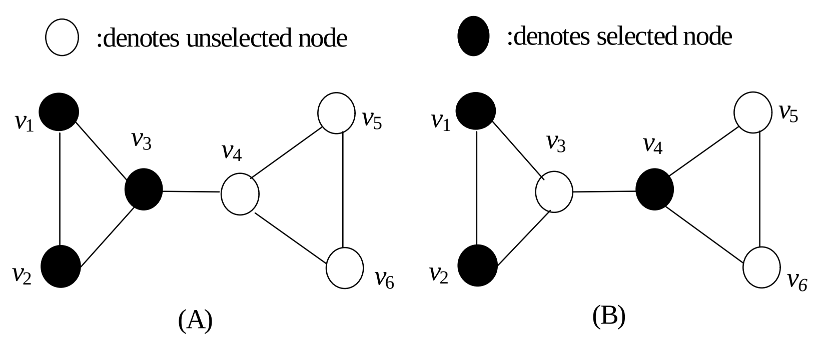
$$w(e_{i,j}) = 1 - \prod_{z=1}^{y_{i,j}} (q_{i,k_z} \times p_{k_z,j}), \qquad (3)$$

Where

$y_{i,j} \leq n-2$.

Assume not only that we have a selected set G_k of nodes with reliability $R(G_k)$, but also that the nodes in G_k are all directly connected. If another set G'_k of nodes exists in which just one node is different from G_k and G'_k has one node which is not directly connected with other nodes in G'_k, then we say that $R(G_k) \geq R(G'_k)$ for $R(G'_k)$ the reliability of set G'_k.

Fig. 1. (A) A selected set G_k ={ v_1, v_2, v_3} under a DS with six nodes and seven links; (B) A selected set G'_k ={ v_1, v_2, v_4} under a DS with six nodes and seven links.



By assuming that the 2-terminal reliability between v_1 and v_2 is R_1, this relation can be represented as $R(\{v_1, v_2\}) = R_1$, and $R(\{v_1, v_3\}) = R_2$, $R(\{v_2, v_3\}) = R_3$, $R(\{v_3, v_4\}) = R_4$. In Fig. 1(A), we select nodes v_1, v_2 and v_3. Therefore, $R(\{v_1, v_2\}) = R_1$, $R(\{v_1, v_3\}) = R_2$, $R(\{v_2, v_3\}) = R_3$. According to Fig. 1(B), we select nodes v_1, v_2 and v_4, Therefore, $R(\{v_1, v_2\}) = R_1$, $R(\{v_1, v_4\}) = R_2 \times R_4$ and $R(\{v_2, v_4\}) = R_3 \times R_4$. Because $R_2 \leq 1$, $R_3 \leq 1$ and $R_4 \leq 1$, $R_2 \times R_4 \leq R_2$ and $R_3 \times R_4 \leq R_3$, the reliability of node v_1, v_2, v_3 $\geq$ the reliability of node v_1, v_2, v_4. Restated, $R(\{v_1, v_2, v_3\}) \geq R(\{v_1, v_2, v_4\})$. However, this assumption is not always true if (a) a path exists between v_4 and v_1 or between v_4 and v_2, and (b) the reliability of the path is larger than the reliability between v_3 and v_1 and between v_3 and v_2. For this reason, in some cases, the maximum reliability cannot be achieved using the proposed method.

This assumption is true if the reliability of any path between X and K is less than that of links between the set K of nodes. Restated, the proposed method can be used to achieve maximum reliability.

In each set of nodes, if the number of members of a set is |K|, the following equation can be used to compute its weight

$$w(G_k) = \{ [w(E_{G_k}) + w(e_{s,j})] / [|K| \times (|K|-1)/2] + w(V_{G_k}) / [(n-1) \times |K|] \}, \quad (4)$$

where

$w(E_{G_k}) = \Sigma w(e_{i,j})$ and $e_{i,j} \in$ a set E_{G_k} of direct links between any two nodes in G_k

$w(V_{G_k}) = \Sigma w(v_j)$ and $v_j \in$ a set V_{G_k} of nodes in G_k.

According to $w(E_{G_k})$ and $w(V_{G_k})$ in Eq. (4), only the sum of the weight of the links between v_i and G_k and the weight of node v_i should be derived. Therefore, the weight of the K-terminal with another node, say v_i, can be obtained easily and efficiently using the equation as follows:

$$w(G_k \cup \{v_i\}) = \{ [w(E_{G_k}) + w(e_{s,j}) + \sum_{v_j \in V_{G_k}, v_i \notin V_{G_k}, e_{i,j} \notin E_{G_k}} w(e_{i,j})] / [((|K|+1) \times |K|)/2] + [w(V_{G_k}) + w(v_i)] / [(n-1) \times (|K|+1)] \}. \quad (5)$$

Herein, a node of the heaviest weight is selected and serves as the starting node for deriving an adequate K-terminal. Before assigning one node to a selected set, the proposed algorithm only inspects those nodes that are adjacent to any node of the selected node for the reason of reducing computation time. In the first node assigned to a selected set, the proposed algorithm also probes nodes in $V_{e(G_k)}$. The notation $V_{e(G_k)}$ represents a set of nodes which $e_{s,j}$ exist in G_k.

The Proposed Heuristic Algorithm

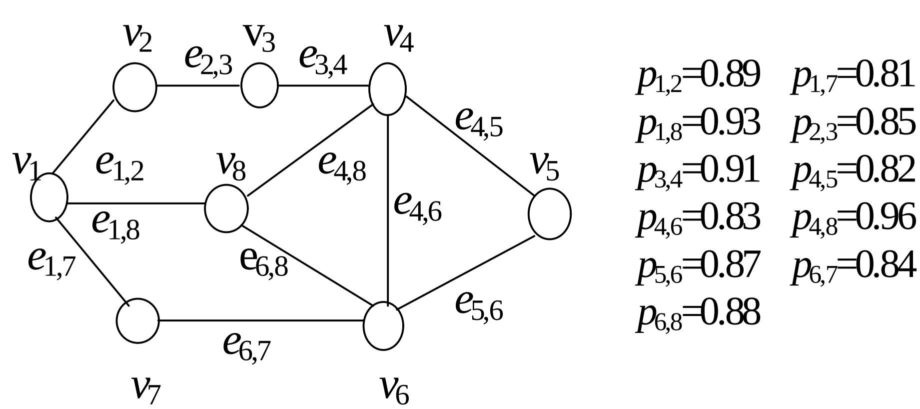
We present a heuristic algorithm by carefully selecting the starting node v_s according to a node's weight. Before assigning a node to the selected set, the proposed algorithm probes those nodes $V_{adj(Gk)}$ that are adjacent to any node of a selected node except for the selected nodes. After obtaining the K-terminal, SYREL (Hariri and Raghavendra, 1987) is applied to compute the reliability. The proposed heuristic algorithm is to maximize K-terminal reliability of a DS. The detailed steps for KTR are described in Appendix 1.

ILLUSTRATIVE EXAMPLES

Example 1

Figure 2 illustrates the topology of a DS with eight nodes and eleven links. The problem involves determining a subset, K-terminal, of the DS which includes some of the nodes v_1, v_2, …, v_8 whose reliability is maximal.

Fig. 2. The DS with eight nodes and eleven links.



In Step 1, each node's weight is evaluated using Eq. (1). The weight of v_1, v_2, …, and v_8 are 0.998537, 0.9835, 0.9865, 0.9998898, 0.9766, 0.9995756, 0.9696 and 0.999664, respectively. Therefore, v_4 is the node with maximal weight and is served as starting node for obtaining an adequate K-terminal. Notably, G_k is $\{v_4\}$.

In Step 2, because $e_{s,j}$ does not exist, set $E_{e(G_k)}$ to empty.

In Step 3, each link's weight is evaluated using Eq. (2).

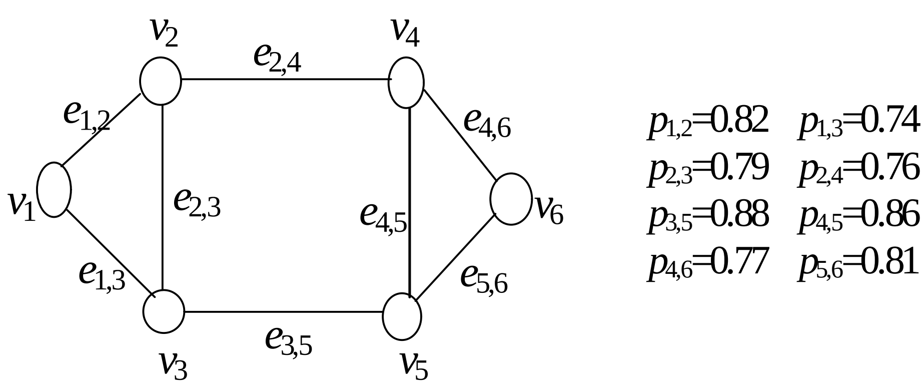
In Step 4, let $V_{tmp} = V_{e(G_k)} = \emptyset$, $w(V_{G_k}) = w(v_s)$, $w(E_{G_k}) = 0$.

In Step 5, for the set of nodes, ($V_{adj(Gk)}$ Y V_{tmp})={v_3, v_5, v_6, v_8}, find v_i, in ($V_{adj(Gk)}$ Y V_{tmp}). Using Eq. (5) to evaluate weight, we have w({v_4, v_3}) = 1.051885, w({v_4, v_5}) = 1.091156, w({v_4, v_6}) = 1.135257 and w({v_4, v_8}) = 1.132041, respectively. Because the weight of {v_4, v_6} is maximum, v_6 is appended to G_k. Notably, G_k is {v_4, v_6}, V_{tmp} = {}. The algorithm goes back step 5 and continues.
In Step 5, $V_{adj(Gk)}$ = {v_3, v_5, v_7, v_8} and V_{tmp} = {}. Therefore, ($V_{adj(Gk)}$ Y V_{tmp}) = {v_3, v_5, v_7, v_8}. Using Eq. (5) to evaluate weight, we have w({v_4, v_6, v_3}) = 0.776335, w({v_4, v_6, v_5}) = 1.08682, w({v_4, v_6, v_7}) = 0.752197 and w({v_4, v_6, v_8}) = 1.128572, respectively. Because the weight of {v_4, v_6, v_8} is maximum, v_8 is appended to G_k. Notably, G_k is {v_4, v_6, v_8}.
In Step 6, the reliability of the K-terminal {v_4, v_6, v_8} is computed using SYREL. We have R({v_4, v_6, v_8}) = 0.9966876 which has the maximum reliability under $K_{request}$. The number of reliability computation is exactly one.
The result is the same as in the K-terminal, which is derived by an exhaustive method.

Example 2

Figure 3 illustrates the topology of a DS with six nodes and eight links. The problem involves determining a subset, K-terminal, of the DS which includes some of the nodes v_1, v_2, …, v_6 whose reliability is maximal.

Fig. 3. The DS with six nodes and eight links.



In Step 1, after evaluating each node's weight using Eq. (1), v_5 is the heaviest node and is served as starting node for obtaining an adequate K-terminal. Note that G_k is $\{v_5\}$.
In Step 2, after finding $e_{s,j}$, we have $E_{e(G_k)}$={ $e_{5,2}$ }.
In Step 3, each link's weight is evaluated using Eq. (2) and each $e_{s,j}$ in $E_{e(G_k)}$ is evaluated using Eq. (3)
In Step 4, let V_{tmp} = $V_{e(G_k)}$= $\{v_2\}$, $w(V_{G_k})$ = $w(v_s)$, $w(E_{G_k})$ = 0.
In Step 5, for sets of nodes, $V_{adj(Gk)} = \{v_3, v_4, v_6\}$ and $V_{tmp} = \{v_2\}$, find v_i, in ($V_{adj(Gk)}$ Y V_{tmp}). Using Eq. (5) to compute the weight, we have $w(\{v_5, v_3\})$ = 1.079026, $w(\{v_5, v_4\})$ = 0.1.146226, $w(\{v_5, v_6\})$ = 1.131129 and $w(\{v_5, v_2\})$ = 1.093191. Because the weight of $\{v_5, v_4\}$ is maximum, v_4 is appended to G_k. Notably, G_k is $\{v_5, v_4\}$.
In Step 6, the reliability of the K-terminal $\{v_5, v_4\}$ is computed using SYREL. We have $R(\{v_5, v_4\})$ = 0.9452258 which has the maximum reliability under $K_{request}$. The number of reliability computation is exactly one.
The result is the same as in the K-terminal, which is derived by an exhaustive method.

COMPARISON AND DISCUSSION

Results obtained from our algorithm were compared with those of exhaustive method and random method. Although capable of yielding the optimal solution, conventional techniques such as exhaustive method cannot effectively reduce the reliability count. An application occasionally requires an efficient algorithm to compute reliability owing to resource considerations. Under this circumstance, deriving the optimal reliability may not be feasible. Instead, an efficient algorithm yielding approximate reliability is preferred.

In contrast to the computer reliability problem, which is static-oriented, the KTR problems in the DS are dynamic-oriented since many factors, e.g. DS topology, link reliability, and the number of paths between each node, can significantly affect the efficiency of the algorithm(Aziz, 1997; Nakazawa, 1981; Makri and Psillakis, 1997). Next, the accuracy and efficiency of the proposed algorithm are verified by implementing simulation programs C language that are executed on a Pentium 133 with 16M-DRAM on MS-Windows 95. We use many network topologies and generated several hundreds of data for simulation. The reliability of each link was generated using a random number generator. For verifying the sensitivity of our proposed algorithm, two data categories were given in different ranges. For the link reliability, we considered the following range: 0.0~1.0, 0.5~1.0 and 0.8~1.0. For the number of nodes in K-terminal, we consider that the $k_{request}$ is equal to 2, 3 and 4, respectively. Table 1 (Appendix) presents the data on the results obtained using different methods for various DS topologies. In contrast to the exhaustive method, the number of reliability computations grew rapidly when the DS topology size is increased.

Table 1. Comparison with Other Methods Under the Range of Link Reliability in 0.8 ~ 1.0.

size		$K_{request}$	exhaustive method			RM & PM		RM	PM
N	e		Max_Rel	NRC	time(sec)	NRC	time(sec)	absolute err	absolute err
12	21	4	0.9993535	1365	39.285	1	0.0288	0.2569579	0.0002387
12	21	3	0.9996854	220	8.462	1	0.0385	0.1513619	0
12	21	2	0.9999972	66	1.318	1	0.0199	0.2487065	0
19	29	4	0.9997547	3876	397.573	1	0.1026	0.1054234	0
19	29	3	0.9999088	912	49.689	1	0.0545	0.0791060	0
19	29	2	0.9999860	171	4.395	1	0.0257	0.0574796	0
25	31	4	0.9997928	12650	517.747	1	0.0409	0.4111645	0
25	31	3	0.9997537	2300	64.615	1	0.0281	0.5188945	0.0000882
25	31	2	0.9999160	300	5.714	1	0.0190	0.5145090	0
30	30	4	0.8558104	27405	224.175	1	0.0081	0.5393191	0
30	30	3	0.9207547	4060	29.780	1	0.0073	0.5715915	0
30	30	2	0.9984403	435	2.967	1	0.0068	0.6360298	0
30	31	4	0.9462816	27405	235.549	1	0.0086	0.6404305	0
30	31	3	0.9769668	4060	30.824	1	0.0076	0.5076126	0
30	31	2	0.9955244	435	3.022	1	0.0069	0.5205490	0
Average				5710.7	107.7	1	0.02688	0.3839420	0.0000218

n: the number of nodes in G, n = |V| e: the number of links in G, e = |E|
NRC: the number of reliability computation $K_{request}$: the order of K-terminal
Max_Rel: maximum reliability satisfies our constraints RM: random method
PM: the proposed method

Tables 2 and 3 (Appendix) list the results obtained using the random method and our proposed method for three different topologies (ring, bridge, hyper-cube) with eight nodes, respectively. These data show that the proposed method is more effective than the random method. When the DS topology and the link reliability are fixed, the order of K-terminal affects the average exact KTR solution. For example, the average exact KTR solution when $K_{request}$ is set to four is worse than when $K_{request}$ is set to three or two. Without a loss of generality, the ratio of the average relative error was negatively correlated with the link reliability range and the number of links. The complexity of exhaustive method is $O(n^{\min(k,n-k)} \times m^2)$, where e denotes the number of edges, n represents the number of nodes, k represents the order of

K-terminal and m represents the number of paths of a selected K-terminal (Hariri and Raghavendra, 1987). In the proposed algorithm, in the worst case, the complexity of evaluating the weight of each node is O(e) and each link is O(e× n), selecting an adequate K-terminal is $O(n^3)$, and computing the reliability of the K-terminal using SYREL is $O(m^2)$ (Hariri and Raghavendra, 1987). Therefore, the complexity of the proposed algorithm is $\max(O(n^3), O(m^2))$.

Table 2. Random Method for Three DS Topologies with Eight Nodes.

T(s)	Lr	$K_{request}$	AES	HitR	ARErrR	UpErrBnd	UpErrBndR	ARErrRlnk	ARErrRT
Ring (n8e8)	0.0~1.0	4	0.326084	0	0.763126	0.417357	0.957736		
		3	0.503812	0	0.829212	0.733091	0.964845		
		2	0.886631	20	0.644896	0.842723	0.949601	0.745745	
	0.5~1.0	4	0.638387	0	0.327150	0.320436	0.479467		
		3	0.732953	0	0.355200	0.381432	0.555065		
		2	0.963593	10	0.339181	0.457403	0.568255	0.340510	
	0.8~1.0	4	0.897291	0	0.062130	0.081107	0.091602		
		3	0.951859	0	0.088845	0.128657	0.140203		
		2	0.987630	0	0.065677	0.138905	0.141486	0.072217	0.386157
Bridge (n8e11)(Fig. 3.)	0.0~1.0	4	0.667060	0	0.532687	0.607786	0.807201		
		3	0.716601	0	0.713274	0.890054	0.924501		
		2	0.936734	0	0.369311	0.692768	0.734436	0.538424	
	0.5~1.0	4	0.937299	0	0.195759	0.316703	0.392177		
		3	0.965475	0	0.149418	0.318579	0.319403		
		2	0.984210	0	0.094193	0.208457	0.214969	0.146457	
	0.8~1.0	4	0.991387	0	0.042027	0.065841	0.065841		
		3	0.997596	0	0.027991	0.056993	0.057117		
		2	0.998858	0	0.018207	0.047118	0.04732	0.029408	0.23096
Hypercube (n8e12)	0.0~1.0	4	0.558898	0	0.561959	0.483877	0.951321		
		3	0.759641	10	0.338054	0.395725	0.748639		
		2	0.952303	0	0.388869	0.029266	0.648013	0.429627	
	0.5~1.0	4	0.956241	0	0.048489	0.094668	0.096577		
		3	0.972747	10	0.048654	0.103147	0.104987		
		2	0.996003	20	0.041780	0.148503	0.151313	0.046308	
	0.8~1.0	4	0.997204	10	0.003449	0.008865	0.008914		
		3	0.998569	0	0.004190	0.009181	0.009193		
		2	0.999521	0	0.002618	0.005022	0.005032	0.003419	0.159785
Average				2.9	0.261346	(0.295691)	(0.412415)	0.261346	0.261346

T(s): represents the topology (size) of a DS.
Lr : the range of link's reliability, the reliability is obtained by random generator.
AES : the value of average exact solution whose value is
$[\Sigma(DSR_{opt}]/(\text{total simulation cases})$.
HitR : the ratio of obtaining exact solution.
ARErrR : the value of $(\Sigma[1-(DSR_{app}/DSR_{opt})])/(\text{total simulation cases})$.

UpErrBnd : the upper error bound whose value is
the max(DSR_{opt}-DSR_{app}) in total simulation cases.
UpErrBndR : the value of the max[(DSR_{opt}-DSR_{app})/ DSR_{opt}] in total simulation cases.
ARErrRlnk : the value of average of ARErrR which are in same Lr and T(s).
ARErrRT : the value of average of ARErrR which are in same T(s).
(.): denotes the value is just for reference
DSR_{app} : approximation solution which obtained by running heuristic algorithm
DSR_{opt} : optimal solution which obtained by running exhaustive search algorithm.

Table 3. The Proposed Method for Three DS Topologies with Eight Nodes.

T(s)	Lr	$K_{request}$	AES	HitR	ARErrR	UpErrBnd	UpErrBndR	ARErrRlnk	ARErrRT
Ring (n8e8)	0.0~1.0	4	0.326084	90	0.021321	0.020473	0.106605		
		3	0.503812	80	0.023556	0.049896	0.099507		
		2	0.886631	90	0.000429	0.003579	0.004288	0.015102	
	0.5~1.0	4	0.638387	80	0.012382	0.048862	0.101626		
		3	0.732953	60	0.030009	0.086259	0.127607		
		2	0.963593	90	0.002623	0.025769	0.026230	0.015005	
	0.8~1.0	4	0.897291	50	0.007564	0.030080	0.033039		
		3	0.951859	90	0.000397	0.003841	0.003970		
		2	0.987630	90	0.000158	0.001548	0.001577	0.002706	0.010938
Bridge (n8e11)(Fig. 3.)	0.0~1.0	4	0.667060	70	0.021885	0.056531	0.151792		
		3	0.716601	90	0.007242	0.016407	0.072416		
		2	0.936734	90	0.001230	0.012162	0.012595	0.010129	
	0.5~1.0	4	0.937299	60	0.011850	0.060240	0.063696		
		3	0.965475	50	0.011232	0.041352	0.042474		
		2	0.984210	70	0.001616	0.009778	0.010454	0.008233	
	0.8~1.0	4	0.991387	30	0.007660	0.018952	0.018994		
		3	0.997596	80	0.001346	0.008294	0.008323		
		2	0.998858	100	0.0	0.0	0.0	0.003002	0.007121
Hypercube (n8e12)	0.0~1.0	4	0.558898	80	0.011515	0.070260	0.092964		
		3	0.759641	70	0.016015	0.070496	0.092135		
		2	0.952303	70	0.006012	0.035770	0.037123	0.011181	
	0.5~1.0	4	0.956241	40	0.008914	0.027166	0.029424		
		3	0.972747	90	0.000588	0.005626	0.005883		
		2	0.996003	70	0.000746	0.005544	0.005570	0.003416	
	0.8~1.0	4	0.997204	40	0.001738	0.004195	0.004207		
		3	0.998569	70	0.000139	0.000810	0.000810		
		2	0.999521	90	0.000011	0.000117	0.000117	0.000629	0.005075
Average				73.3	0.007711	(0.026445)	(0.042719)	0.007711	0.007711

The mean of notations is described in footnote of Table 2.
The value of AES is same as Table 2.

Table 4 lists the results obtained using an exhaustive method and our proposed method for three different topologies (ring, bridge, hyper-cube) with eight nodes, respectively.

Table 4. The Average Exhaustive Method Execution Time and the Proposed Method for Three DS Topologies with Eight Nodes.

			exhaustive metohd			the proposed method		
T(s)	Lr	$K_{request}$	AT(sec)	ATlnk(sec)	ATT(sec)	AT(sec)	ATlnk(sec)	ATT(sec)
Ring (n8e8)	0.0~1.0	4	0.544			0.008		
		3	0.313			0.006		
		2	0.121	0.3260		0.004	0.0059	
	0.5~1.0	4	0.544			0.008		
		3	0.297			0.005		
		2	0.121	0.3205		0.004	0.0058	
	0.8~1.0	4	0.549			0.008		
		3	0.319			0.006		
		2	0.132	0.3333	0.3266	0.005	0.0061	0.0059
Bridge (n8e11)(Fig. 3.)	0.0~1.0	4	0.791			0.011		
		3	0.396			0.007		
		2	0.148	0.4451		0.005	0.0079	
	0.5~1.0	4	0.791			0.011		
		3	0.429			0.008		
		2	0.143	0.4542		0.005	0.0080	
	0.8~1.0	4	0.775			0.011		
		3	0.423			0.008		
		2	0.137	0.4451	0.4481	0.005	0.0078	0.0079
Hypercube (n8e12)	0.0~1.0	4	1.610			0.023		
		3	0.868			0.016		
		2	0.280	0.9194		0.010	0.0162	
	0.5~1.0	4	1.621			0.023		
		3	0.879			0.016		
		2	0.286	0.9286		0.010	0.0164	
	0.8~1.0	4	1.615			0.023		
		3	0.863			0.015		
		2	0.286	0.9212	0.9231	0.010	0.0162	0.0163
Average			0.5659	0.5659	0.5659	0.0100	0.0100	0.0100

AT : the seconds of average execution time,

AT = $(\Sigma\,(\text{execution time}))\,/\,(\text{total simulation case})$.

ATlnk : the seconds of average execution time of same Lr and T(s).

ATT : the seconds of average execution time of same T(s).

The mean of other notations is described in footnote of Table 2.

These data show that the proposed method is more efficient than the exhaustive method. In the random method, which obtains the exact solution below 3%, the average error from exact solution surpasses 0.26. In our simulation case, the reliability count for the proposed algorithm is exactly one. The exact solution can be obtained above 73.3%, in which the average error from exact solution is under 0.008. In a few cases, an adequate node which has arrived for selected node set through many paths and the length of a great number of those paths exceeds two, the node may be lost when using our equation for computing link's weight. Notably, the proposed algorithm cannot obtain the exact solution.

CONCLUSIONS

Computing DS reliability is generally NP-hard. In this work, we presented a heuristic algorithm to obtain a K-terminal with sub-optimal reliability. The reliability computation in our algorithm is only exactly one. Therefore, KTR in the DS can provide the desired performance.

In addition, the algorithm proposed herein is compared with an exhaustive method and a random method for various topologies. According to that comparison, the proposed algorithm is more efficient in terms of execution time for a large DS. When the proposed method fails to provide an exact solution, the error from the exact solution is only slight.

REFERENCES

Aggarwal, K. K. and S. Rai. 1981. Reliability evaluation in computer communication networks. *IEEE Transactions on Reliability*, R-30, no.1: 32-35.

Aggarwal, K. K., Y. C. Chopra, and J. S. Bajwa. 1982. Topological layout of links for optimizing the S-T reliability in a computer communication system. *Microelectronics and Reliability*, vol.22, no.3: 341-345.

Aziz, M. A. 1997. Pathset enumeration of directed graphs by the set theoretic Method. *Microelectronics and Reliability*, Vol. 37, no.5: 809-814.

Hariri, Salim. and C. S. Raghavendra. 1987. SYREL: a symbolic reliability algorithm based on path and cuset methods. *IEEE Transactions on Computers*, Vol.C-36, no.10: 1224-1232.

Irani, K. B. and N. G. Khabbaz. 1982. A methodology for the design of communication networks and the distribution of data in distributed supercomputer systems. *IEEE Transaction on Computers*, Vol. C-31, no.5: 420-434.

Kumar, Anup. and Dharme P. Agrawal. 1993. A generalized algorithm for evaluation distributed program reliability. *IEEE Transactions on Reliability*, Vol.42, no.3: 416-426.

Lin, M. S. and D. J. Chen. 1992. New reliability evaluation algorithms for distributed computing systems. *Journal of Information Science and Engineering*, Vol.8, no.3: 353-391.

Lin M. S., M. S. Chang and D. J. Chen. 1999. Efficient algorithms for reliability analysis of distributed computing systems. Journal of Information Science and Engineering, Vol.117, no.1-2.

Makri, F. S. and Z. M. Psillakis. 1997. Bound for reliability of k-within connected-(v,s) out-of (m,n) failure systems. *Microelectronics and Reliability*, Vol.37, no.8: 1217-1224.

Nakazawa, H. 1981. Decompositon methos for computing the reliability of complex networks. *IEEE Transactions on Reliability*, Vol. R-32, no.2: 289-292.

Politof, T. and A. Satyanarayana. 1986. Efficient algorithms for reliability analysis of planar netwotks—a survey. *IEEE Transactions on Reliability*, Vol.R-35, no.3: 252-259.

Rai, S., A. K. Sarje, E. V. Prasad and A. Kumar. 1987. Two recursive for computing the reliability of k-out-of-n systems. *IEEE Transactions on Reliability*, Vol.R-36, no.2: 261-265.

Satyanarayana, A., and J. N. Hagstrom. 1981. A new algorithm for the reliability analysis of multi-terminal networks. *IEEE Transactions on Reliability*, Vol.R-30, no.4: 325-333.

Stankovic, J. A. 1984. A perspective on distributed computer systems. *IEEE Transaction on Computer*, Vol.c-33, no.12: 1102-1115.

Torrieri, Don. 1994. Calculation of node-pair reliability in large networks with unreliable nodes. *IEEE Transactions on Reliability*, Vol.43, no.3: 375-377.

Appendix 1

This appendix describes the detail steps of KTR.

Algorithm KTR

Step 0 Initialize, read system parameters: n, e; read the order $K_{request}$ of K-terminal.

Random generate probability $p_{i,j}$ of each $e_{i,j}$ in G.

Step 1 Evaluate the weight of each node using Eq. (1) and choose the heaviest one as the starting node, say v_s, for deriving an adequate K-terminal. Notably, G_k is initialized to $\{v_s\}$.

Step 2 Find each $e_{s,j}$ and insert it into a set $E_{e(G_k)}$ of $e_{s,j}$.

Step 3 Evaluate the weight of each link using Eq. (2).

Evaluate the weight of each $e_{s,j}$ in $E_{e(G_k)}$ using Eq. (3).

Step 4 Let $V_{tmp} = V_{e(G_k)}$. /* V_{tmp} denotes a set of nodes */

Let $w(V_{G_k}) = w(v_s)$.

Let $w(E_{G_k}) = 0$.

Step 5 Dowhile ($|G_k| < K_{request}$)

/* find an adequate v_i after evaluating each $w(G_k \cup \{v_i\})$ using Eq. (5)*/

Find v_i, such that $w(G_k \cup \{v_i\}) = \max\{w(G_k \cup \{v_i\}) \mid v_i \in (V_{adj(Gk)} \cup V_{tmp})\}$.

Let $G_k = G_k \cup \{v_i\}$.

Let $w(V_{G_k}) = w(V_{G_k}) + w(v_i)$.

Let $w(E_{G_k}) = w(E_{G_k}) + \sum_{v_j \in V_{G_k}, v_i \notin V_{G_k}, e_{i,j} \notin E_{G_k}} w(e_{i,j})$.

Let $V_{tmp} = \emptyset$.

$\forall\, e_{s,j} \in E_{e(G_k)}$, let $w(e_{s,j}) = 0$.

End_Dowhile

Step 6 Compute $R(G_k)$ using SYREL, output the K-terminal G_k and its reliability.

End KTR