PERGAMON

Computers & Operations Research 27 (2000) 129–142

computers & operations research

www.elsevier.com/locate/orms

The distributed program reliability analysis on star topologies

Ming-Sang Chang[a], Deng-Jyi Chen[a,*], Min-Sheng Lin[b], Kuo-Lung Ku[c]

[a]*Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsin Chu, Taiwan, ROC*
[b]*Department of Information Management, Tamsui Oxford University College, Tamsui, Taipei, Taiwan, ROC*
[c]*Chung-Shan Institute of Science and Technology, Tao-Yuan, Taiwan, ROC*

Received March 1998; received in revised form November 1998

Abstract

A distributed computing system consists of processing elements, communication links, memory units, data files, and programs. These resources are interconnected via a communication network and controlled by a distributed operating system. The distributed program reliability in a distributed computing system is the probability that a program which runs on multiple processing elements and needs to retrieve data files from other processing elements will be executed successfully. This reliability varies according to (1) the topology of the distributed computing system, (2) the reliability of the communication edges, (3) the data files and programs distribution among processing elements, and (4) the data files required to execute a program. In this paper, we show that computing the distributed program reliability on the star distributed computing systems is NP-hard. We also develop an efficiently solvable case to compute distributed program reliability when some additional file distribution is restricted on the star topology.

Scope and purpose

Recent advances in VLSI circuitry have a tremendous impact on the price-performance revolution in microelectronics. This development has led to an increased use of workstations connected in the form of a powerful distributed computing system. Potential benefits offered by such distributed computing systems include better cost performance, enhanced fault tolerance, increased system throughput, and efficient sharing of resources. Distributed program reliability is an important measure that should be examined for designing a high fault-tolerance distributed computing system. This reliability varies according to (1) the topology of the distributed computing system, (2) the reliability of the communication edges, (3) the data files and programs distribution among processing elements, and (4) the data files required to execute a program. This article is concerned with the analysis of distributed program reliability on star distributed computing systems.

Keywords: Distributed program reliability; Distributed computing system; Algorithms

* Corresponding author. Tel.: + 886 35 712121 x54755; fax: + 886 35 724176.
E-mail address: djchen@csie.nctu.edu.tw (D.-J. Chen)

0305-0548/00/$ - see front matter
PII: S0305-0548(99)00011-8

1. Introduction

A distributed computing system (DCS) consists of processing elements, communication links, memory units, data files, and programs. These resources are interconnected via a communication network and controlled by a distributed operating system. A distributed computing system has become very popular for its high fault tolerance, potential for parallel processing, and better reliability performance. One of the important issues in the design of the DCS is the reliability performance. For traditional networks, many reliability indice have been proposed. They include two-terminal reliability, all-terminal reliability, and K-terminal reliability [1–5]. However, these measures are not applicable to practical DCS's since the reliability measure for DCSs should capture the effects of distribution on redundant data files.

In Prasanna Kumar et al. [6] and Kumar et al. [7] distributed program reliability was introduced to model the reliability of DCS. The distributed program reliability in a distributed computing system is the probability that a program which runs on multiple processing elements and needs to retrieve data files from other processing elements will be executed successfully.

Most of network reliability problems (e.g., K-terminal reliability), in general, are NP-hard. The class of NP-hard problems was introduced by Valiant [8]. Computing distributed program reliability (DPR) for general DCS's is also NP-hard. One possible means of avoiding this complexity is to consider only a restricted class of DCS's.

The star topology is one of the most widely used structures for a communication system. The star network was used because it was easy to control – the software is not complex and the traffic flow is simple. Fault isolation is also relatively simple in a star network because the line can be isolated to identify the problem. In a star topology, each machine on the network has its own dedicated connection to a hub or switch. The star topology is used mostly with twisted pair cabling, usually in an Ethernet environment.

In this paper, we are interested in star topologies. We highlight the problem of computing the distributed program reliability for a star topology because the simplicity of the topology may incorrectly be conceived as trivial. We show that computing the distributed program reliability on the star distributed computing systems is NP-hard. We also develop an efficiently solvable case to compute distributed program reliability when some additional file distribution is restricted on the star topology.

In Section 2, the definitions and notation are given for this paper. In Section 3, we show that computing the distributed program reliability on the star distributed computing systems is NP-hard. In Section 4, we propose an efficiently solvable case of DPR problem for star topologies in which data files are restricted to a certain type of distribution. Finally, summary and concluding remarks are given.

2. Notation and definitions

Notation

$D = (V, E, F)$ an undirected distributed computing system (DCS) graph with vertex set V, edge set E and data file set F

m	the number of distinct files in a DCS
FA_i	the set of files available at node i (note: $F = \cup FA_i$)
e_i	edge (r, v_i)
v_i	the node whose incident edge is e_i
p_i	reliability of edge i
q_i	$1 - p_i$
H	subset of files of F, i.e., $H \subseteq F$, and H contains the programs to be executed and all needed data files for the execution of these programs
$R(D_H)$	the DPR of D with a set H of needed files: Pr{all data files in H can be accessed successfully by the executed programs in H}

Definition. A *file spanning tree* (FST) is a tree whose nodes hold all needed files in H.

Definition. A *minimal file spanning tree* (MFST) is an FST such that there exists no other FST that is a subset of it.

Definition. *Distributed program reliability* (DPR) is defined as the probability that a distributed program that runs on multiple processing elements (PEs) and needs to communicate with other PEs for remote files will be executed successfully.

By the definition of MFST, the DPR can be written as

$$R(D_H) = \text{Prob(at least one MFST is operational), or}$$

$$R(D_H) = \text{Prob}\left(\bigcup_{j=1}^{\#\text{mfst}} \text{MFST}_j\right),$$

where $\#$mfst is the number of MFSTs for a given needed file set H.

Definition. A *star* DCS is a topology with $n + 1$ nodes $\{r, v_1, v_2, \ldots, v_n\}$ and n edges $\{(r, v_1), (r, v_2), \ldots, (r, v_n)\}$, where r is a root node of a *star* DCS.

Definition. A set s_i of edges of DCS with a star topology is called a *file cutset* for file f_i if it consists of all edges (r, v_i) such that node v_i contains file f_i, i.e., $s_j = \{(r, v_j) | f_i \in FA_j\}$.

Definition. A star DCS D has the *consecutive file distribution property* iff for each node v, there exists, $f_i \in FA_v$ and $f_j \in FA_v$ then $f_k \in FA_v$ for all k, $i < k < j$.

3. The computational complexity of DPR for a star topology

In this section we show that computing the distributed program reliability on the star distributed computing systems is NP-hard. Complexity results are obtained by transforming a known

NP-hard problem into our reliability problem [9]. For this reason, we first state some known NP-hard problems.

(i) *K-terminal reliability* (KTR) [5]
Input: An undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of edges that fail s-independently of each other with known probabilities. A set $K \subseteq V$ is distinguished with $|K| \geqslant 2$.
Output: $R(G_K)$, the probability that the set K of nodes of G is connected in G.

(ii) *Number of edge covers* ($\#EC$) [10]
Input: an undirected graph $G = (V, E)$.
Output: the number of edge covers for G
$\equiv |\{C \subseteq E$: each node of G is an end of some edge in $C\}|$.

(iii) *Number of vertex covers* ($\#VC$) [11]
Input: an undirected graph $G = (V, E)$.
Output: the number of vertex covers for G
$\equiv |\{K \subseteq V$: every edge of G has at least one end in $K\}|$.

Theorem 1. *Computing DPR for a general DCS is NP-hard.*

Proof. We reduce the KTR problem to our DPR problem. For a given network $G = (V, E)$ and a specified set $K \subseteq V$, we can define an instance of the DPR problem. Construct a DCS graph $D = (V, E, F)$ in which the topology and the reliability of each edge are the same as G. Let $F = \cup_{\text{node } i \in K}\{f_i\}$ and $FA_i = \{f_i\}$ if node $i \in K$ else $FA_i = \emptyset$ for each node $i \in V$. If we set $H = F = \cup_{\text{node } i \in K}\{f_i\}$ then we have $R(D_H) = R(G_K)$. However, Rosenthal [5] and Valiant [8] show that the problem of computing KTR, in general, is NP-hard, so computing DPR, in general, is NP-hard. □

The result of Theorem 1 implies that it is unlikely that polynomial time algorithms exist for solving the DPR problem. One possible means of avoiding this complexity is to consider only a restricted class of structures. The class of interest here is a star topology that is widely used in one-node circuit switched networks.

Theorem 2. *Computing DPR for a DCS with a star topology even with each* $|FA_i| = 2$ *is NP-hard.*

Proof. We reduce the $\#EC$ problem to our problem. For a given network $G = (V_1, E_1)$ where $E_1 = \{e_1, e_2, \ldots, e_n\}$, we construct a DCS $D = (V_2, E_2, F)$ with a star topology where $V_2 = \{r, v_1, v_2, \ldots, v_n\}$, $E_2 = \{(r, v_i) | 1 \leqslant i \leqslant n\}$, and $F = \{f_i |$for each node $i \in G\}$. Let $FA_{vi} = \{f_u, f_v |$ if $e_i = (u, v) \in G\}$ for $1 \leqslant i \leqslant n$, $FA_r = \emptyset$ and $H = F$. From the construction of D, it is easy to show that there is one-to-one correspondence between one of the sets of edge covers and one FST. The DPR of D, $R(D_H)$, can be expressed as

$$R(D_H) = \sum_{\substack{\text{for all FST} \\ t \in D}} \left\{ \prod_{\substack{\text{for each} \\ \text{edge } i \in t}} p_i \prod_{\substack{\text{for each} \\ \text{edge } i \notin t}} (1 - p_i) \right\}.$$

Thus, a polynomial-time algorithm for computing $R(D_H)$ over a DCS with a star topology and each $|FA_i| = 2$ would imply an efficient algorithm for $\#EC$ *problem*. Since $\#EC$ problem is NP-hard, Theorem 2 follows. □

Theorem 3. *Computing DPR for a DCS with a star topology even when there are only two copies of each file is NP-hard.*

Proof. We reduce the $\#VC$ problem to our problem. For a given $G = (V_1, E_1)$ where $|E_1| = n$ and $V_1 = \{v_1, v_2, \ldots, v_m\}$, we construct a DCS $D = (V_2, E_2, F)$ with a star topology where $V_2 = V_1 \cup \{r\}$, $E_2 = \{e_i = (r, v_i) | 1 \leqslant i \leqslant m\}$, and $F = \{f_i |$ for all edge $i \in G\}$. Let $FA_i = \{f_j |$ for all edge j that are incident on $v_i \in G\}$ and $H = F$. From the construction of D, it is easy to show that there are only two copies of each file in D and one-to-one correspondence between one of the sets of vertex covers and one FST of D. The DPR of D, $R(D_H)$, can be expressed as

$$R(D_H) = \sum_{\substack{\text{for all FST} \\ t \in D}} \left\{ \prod_{\substack{\text{for each} \\ \text{edge } i \in t}} p_i \prod_{\substack{\text{for each} \\ \text{edge } i \notin t}} (1 - p_i) \right\}.$$

Since $\#VC$ problem is NP-hard, Theorem 3 follows. □

By Theorems 2 and 3, we show that computing the DPR for a DCS with a star topology in general is NP-hard.

4. An efficiently solvable case for DPR analysis on a star topology

The results of the previous section indicate that computing DPR over a star DCS is NP-hard. These results imply that it is unlikely that polynomial algorithms exist for solving them. It is, however, possible that an efficient algorithm exists for computing DPR over a star DCS with a certain restricted class of file distribution.

The following sections will use the concept of cutset to compute DPR for DCS with a star topology. We first introduce the concept of cutset for calculating the measure of network reliability. Then, we get the file cutsets from a star DCS. We do not make a reduction of file cutsets before a semilattice is considered because it cannot reduce the time complexity of algorithm REL in Section 4.2. We construct the file cutsets to be a semilattice structure and test whether or not this semilattice structure is also satisfied with some additional properties. If these additional properties are satisfied, then an efficient algorithm exists for computing DPR over a star DCS.

4.1. Basic concept

It is important to be able to assess the reliability of a complex system, based on knowledge concerning the reliabilities of its individual components. In a typical situation, edges of the network are assumed to fail in a statistically independent fashion with known probabilities. For such networks, a variety of probabilistic measures of system performance have been considered.

Numerous algorithms have been proposed for calculating these measures of network reliability. One class of methods is based on the idea of a path, a minimal set of edges whose operation ensures that the system functions. In this approach, paths must first be enumerates and then combine either by applying the inclusion–exclusion principle or by effecting a partition into mutually disjoint events [12–14]. An alternative approach uses instead the enumeration and combination of cutsets, minimal sets of edges whose failure ensures that the system cannot function [15, 16].

The following section uses the concept of cutset to compute DPR for DCS with a star topology. The emphasis is on identifying an underlying semilattice structure that captures certain algorithmically desirable features of such networks. In the following section, we apply the results of Provan and Ball [17] and Shier [18] to obtain a pseudo-polynomial-time algorithm for computing the DPR with a star topology.

4.2. A pseudo-polynomial-time algorithm for computing DPR of a DCS with a star topology

In this section, we will propose a pseudo-polynomial-time algorithm for computing the DPR of a DCS with a star topology. This algorithm is polynomially bounded in the number of file cutsets. However, there exists a method that allows the file cutsets to be generated efficiently in terms of the number of data files. Therefore, processing of these file cutsets by a pseudo-polynomial algorithm might be effective in case the number of distinct files, m, is not too large. On the other hand we believe that our algorithm alone is a polynomial solution if the solution of file cutsets are not considered. Our algorithm REL assumes that file cutsets $S = \{s_1, s_2, \ldots, s_m\}$ are obtained by some other file cutset algorithms as the input of our proposed algorithm. The file cutset problem, in fact, is another important separated issue.

Suppose that $(\mathbf{C}, \mathbf{K})$ is a coherent system with components $\mathbf{C} = \{c_1, c_2, \ldots, c_n\}$ and minimal cutsets $\mathbf{K} = \{k_1, k_2, \ldots, k_m\}$. The collection of minimal cutsets $\mathbf{K}$ is assumed to be endowed with a partial ordering, $\leqslant$, that forms a meet semilattice. In other words, any two k_i and k_j have a greatest lower bound $k_i \wedge k_j$. If this semilattice also satisfies the following two additional properties, then the $O(nm^2)$ algorithm can be applied to calculate the reliability of the system $(\mathbf{C}, \mathbf{K})$ [18].

Property I. If $k_i \leqslant k_r \leqslant k_j$, $c \in k_i$, $c \in k_j$, then $c \in k_r$.

Property II. $k_i \wedge k_j \subseteq k_i \cup k_j$.

We transfer these restrictions into DCS with a star topology for computing DPR. In a DCS $D = (V, E, F)$ with a star topology, V is a node set, $V = \{r, v_1, v_2, \ldots, v_n\}$. E is an edge set, $E = \{e_1, e_2, \ldots, e_n\} = \{(r, v_i) | 1 \leqslant i \leqslant n\}$, and data set F. By identifying edge $e_k \in E$ of a star topology with component c_k of a semilattice framework and identifying the file cutset $s_j \in S$ of a star topology with cutset k_j of a semilattice framework. This means that the file cutsets S of a star topology should be partially ordered by $\leqslant$, forming a semilattice. Moreover, the above properties can be restated in a fairly appealing manner:

Property I′. If $s_i \leqslant s_r \leqslant s_j$, $e \in s_i$, $e \in s_j$, then $e \in s_r$.

Property II′. $s_i \wedge s_j \subseteq s_i \cup s_j$.

Property I′ simply states that each file cutset s_i is a convex set with respect to the ordering $\leqslant$. Property II′ states a closure condition. We are interested in examining the implication of these properties with file distribution of a star topology.

Example 1. Consider a star DCS $D = (V, E, F)$ with edge $E = \{e_1, e_2, e_3, e_4, e_5\}$ shown in Fig. 1. Here the s_i will be the file cutset of each needed file of the network. If we need files $f_1, f_2, f_3, f_4, f_5, f_6$ to complete one program's execution, then the file cutsets of each file can be obtained.

$$s_1 = \{e_1, e_4\}, \quad s_2 = \{e_1, e_4\}, \quad s_3 = \{e_2, e_4, e_5\}, \quad s_4 = \{e_2, e_5\}, \quad s_5 = \{e_3, e_5\}, \quad s_6 = \{e_3\}.$$

The partial order $\leqslant$ is defined by

$$s_i \leqslant s_j, \quad j \leqslant i.$$

Then Properties I′ and II′ will hold. In this case, the Hasse diagram for the partial order is simply a chain shown in Fig. 2.

If the file cutsets of a star topology with partial ordering, $(s_i, \leqslant)$, have the semilattice property and satisfy Properties I′ and II′, then a polynomial time algorithm to compute DPR of a star topology can be fashioned by adapting the algorithm given in [18].

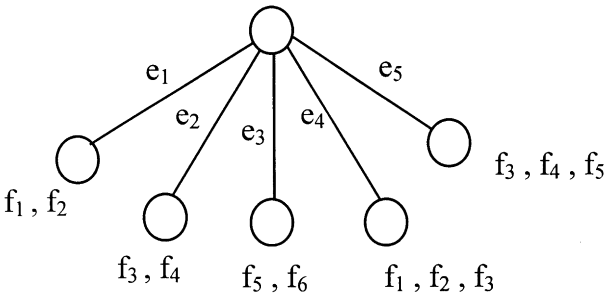


Fig. 1. A star DCS with six files.

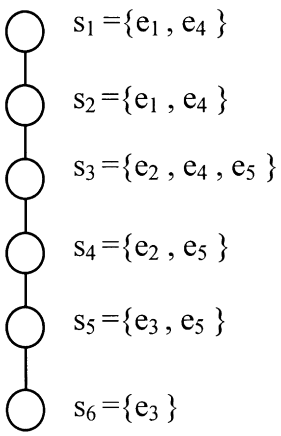


Fig. 2. A semilattice of Fig. 1.

For purposes of exposition, suppose that the edges of a star topology are labelled 1, 2, … , n with q_k being the probability that the edge k fails. Let X_j denote all edges involved in file cutset s_j, $j = 1, 2, \ldots, m$. Fig. 2 displays an illustrative semilattice having six file cutsets and five edges. To present the computing of DPR of a star topology, we define, for any set s_j of file cutsets S,

$$\Omega(X_j) = \prod_{k \in X_j} q_k, \quad \Omega(\phi) = 1.$$

Then the following algorithm correctly determines the DPR of a star topology.

Algorithm REL. This algorithm calculates DPR for a star topology with file cutsets $S = \{s_1, s_2, \ldots, s_m\}$ and edges $E = \{e_1, e_2, \ldots, e_n\}$ that satisfy both the Properties I′ and II′. The edge k fails randomly and independently with probabilities q_k.

1. Place the file cutsets $s_1, s_2, \ldots, s_m$ in topological order so that if $s_i < s_j$, then $j < i$. Let X_j denote the set of edges contained in file cutset s_j.
2. For $j = 1, 2, \ldots, m$,

$$g_j = \Omega(X_j) - \sum_{s_i < s_j} g_i \, \Omega(X_j - X_i),$$

   where g_j is the probability that file cutset s_j is not covered.
3. Output $DPR = 1 - \sum_{j=1}^{m} g_j$.

The quantity g_j needed in step 2 can be computed using the value g_i for file cutset s_i. The validity of this algorithm follows from the development in Shier [18]. As also established there, its worst case complexity is $O(nm^2)$, which is polynomial in the number of edges n and the number of file cutset m.

To illustrate this approach, we apply this algorithm to Example 1. We get

$$g_6 = \Omega(X_6) - \sum_{s_i < s_6} g_i \, \Omega(X_6 - X_i) = \Omega(X_6) = \Omega(e_3) = q,$$

$$g_5 = \Omega(X_5) - \sum_{s_i < s_5} g_i \Omega(X_5 - X_i) = \Omega(e_3, e_5) - \sum g_6 \Omega(e_5) = 0,$$

$$g_4 = \Omega(X_4) - \sum_{s_i < s_4} g_i \, \Omega(X_4 - X_i) = \Omega(e_2, e_5) - [g_6 \Omega(e_2, e_5) + g_5 \Omega(e_2)] = q^2 - q^3,$$

$$g_3 = \Omega(X_3) - \sum_{s_i < s_3} g_i \Omega(X_3 - X_i) = \Omega(e_2, e_4, e_5) - [g_6 \Omega(e_2, e_4, e_5) + g_5 \Omega(e_2, e_4) + g_4 \Omega(e_4)] = 0,$$

$$g_2 = \Omega(X_2) - \sum_{s_i < s_2} g_i \Omega(X_2 - X_i) = \Omega(e_1, e_4) - [g_6 \Omega(e_1, e_4) + g_5 \Omega(e_1, e_4) + g_4 \Omega(e_1, e_4) + g_3 \Omega(e_1)] = q^2 - q^3 - q^4 + q^5,$$

$$g_1 = \Omega(X_1) - \sum_{s_i < s_1} g_i \Omega(X_1 - X_i) = \Omega(e_1, e_4) - [g_6\Omega(e_1, e_4) + g_5\Omega(e_1, e_4) + g_4\Omega(e_1, e_4) + g_3\Omega(e_1) + g_2\Omega(\phi) = 0,$$

$$DPR = 1 - \sum_{j=1}^{m} g_j = 1 - (q + q^2 - q^3 + q^2 - q^3 - q^4 + q^5) = 1 - q - 2q^2 + 2q^3 + q^4 - q^5.$$

For instance, if all edges in the problem of Fig. 2 were randomly available with probability $q = 0.1$, then the DPR of a star topology is computed to be 0.88209.

By Theorem 2, we have shown computing DPR on a star topology with each $|FA_i| = 2$ is a NP-hard problem. In Properties III and IV, we will show classes of semilattice that satisfies Properties I′ and II′ for computing DPR over a star topology with each $|FA_i| = 2$.

Property III. *If a semilattice, for computing DPR over a star topology with each $|FA_i| = 2$, has at least two branches on the same node with length greater than or equal to 2, then the semilattice does not satisfy Properties I′ and II′.*

Proof. We first draw a semilattice with file cutsets that has two branches on the same node with length greater than or equal to 2 as in Fig. 3 and assume that the semilattice satisfies Properties I′ and II′.

From Fig. 3, assume edge $e_x \in s_i \wedge s_j = s_h$. This means the file f_h is in a node, v_x, whose incident edge is e_x. By Property II′, $e_x \subseteq s_i \cup s_j$, then $e_x \subseteq s_i$, or $e_x \subseteq s_j$. Without loss of generality, assume $e_x \subseteq s_i$. Then, the node v_x contains a file f_i. By Property I′, if $s_i \leqslant s_k \leqslant s_h$, $e_x \in s_i$, $e_x \in s_h$ then $e_x \in s_k$. This means the file f_k is in the node v_x.

From the above discussion, the files, f_i, f_h, f_k, are in the same node v_x, which is a contradiction to $|FA_x| = 2$. □

Property IV. *If a semilattice, for computing DPR over a star topology with each $|FA_i| = 2$, has at least three branches with length equal to one, then the semilattice does not satisfy properties I′ and II′.*

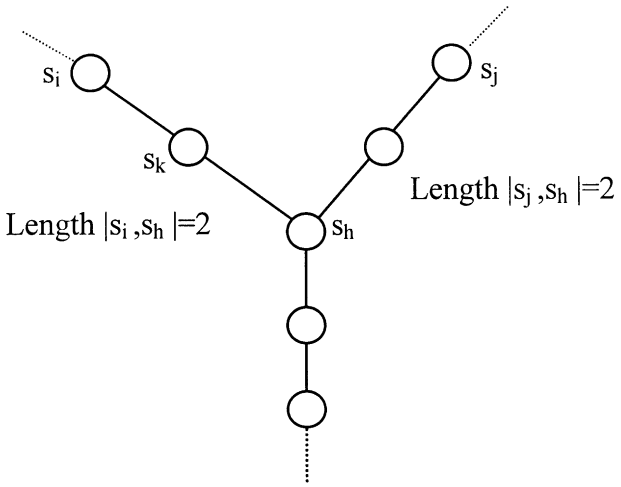


Fig. 3. A semilattice for Property III.

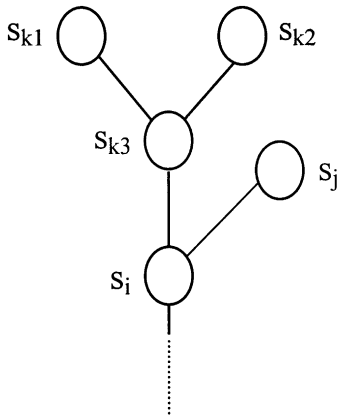


Fig. 4. A semilattice for Property IV.

Proof. We first draw a semilattice of three branches with length equal to 1 as in Fig. 4 and assume that the semilattice satisfies Properties I′ and II′.

From Fig. 4 and Property II′, we have $s_{k1} \wedge s_j = s_i$ and $s_i \subseteq s_{k1} \cup s_j$. Without loss of generality, we assume there is an edge e_x such that $e_x \in s_i$ and $e_x \notin s_j$. This implies $e_x \in s_{k1}$. With the same reason, $e_x \in s_{k2}$ and $e_x \in s_{k3}$. From the above discussion, we conclude that

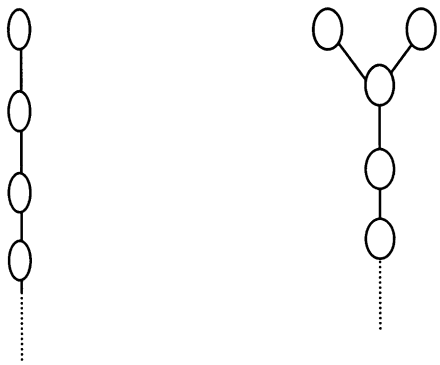
$$e_x \in s_i,\ e_x \in s_{k1},\ e_x \in s_{k2}, \text{ and } e_x \in s_{k3}.$$

This means the files, f_i, f_{k1}, f_{k2}, and f_{k3} are in the same node v_x whose incident edge is e_x. This contradicts $|FA_x| = 2$. □

Theorem 4. There are only two cases of semilattice, for computing DPR over a star topology with $|FA_i| = 2$ that can satisfy Properties I′ and II′.

(1) The semilattice is a linear chain.
(2) The semilattice has only two branches with length one on the top node of the linear chain.

Proof. We first draw these two cases of semilattice as follows:



From the proof of Properties III and IV, it is easy to prove Theorem 4. □

Example 2. Consider a DCS $D = (V, E, F)$ with a star topology and $|FA_i| = 2$ shown in Fig. 5. The file cutset s_i of each file f_i are

$$s_1 = \{e_1, e_2\}, \quad s_2 = \{e_2, e_3\}, \quad s_3 = \{e_1, e_3\}, \quad s_4 = \{e_4\}, \quad s_5 = \{e_4, e_5\}, \quad s_6 = \{e_5\}.$$

The partial order $\leqslant$ is defined by

$$s_1 > s_3, \quad s_2 > s_3, \quad \text{and} \quad s_3 > s_4 > s_5 > s_6.$$

Then, the semilattice is as shown in Fig. 6. From case (2) of Theorem 4, the semilattice satisfies Properties I′ and II′.

In the following, we propose a method to distribute the files without $|FA_i| = 2$ restriction on a star topology. The semilattice can be easily obtained by this method and the DPR of a star topology can be computed in polynomial time. We use Theorem 5 to show this method.

Theorem 5. *A file cutset, $S = \{s_1, s_2, \ldots, s_m\}$ is a semilattice with topological order $s_m < s_{m-1} < \cdots < s_2 < s_1$,* satisfying Properties I′ and II′ iff all files in a star DCS have the consecutive file distribution property.

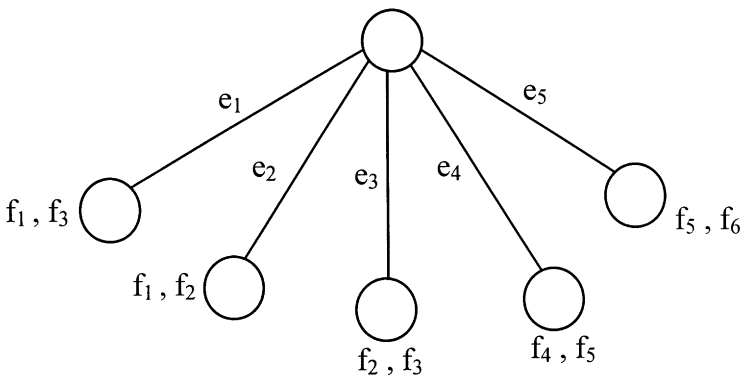


Fig. 5. An example for a star topology with each $|FA_i| = 2$.

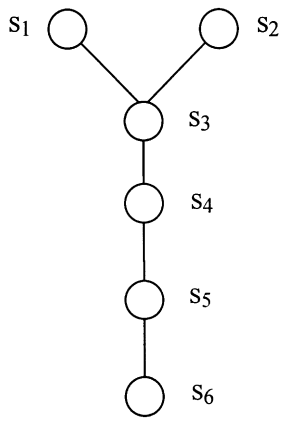


Fig. 6. A semilattice of Fig. 5.

Proof. We first draw an example to show that all files in a star DCS have the consecutive file distribution property as Fig. 7.

If: For edge e_x, assume its adjacent node, v_x, contains consecutive files by its index, f_i, f_{i+1}, f_{i+2}, … ,f_j. Then, the file cutsets s_i, s_{i+1}, s_{i+2}, … , s_j contain the edge e_x and the semilattice of file cutsets with total order relation, $s_j < \cdots < s_{i+2} < s_{i+1} < s_i$, satisfies Properties I′ and II′.

Only if: We assume there exists a semilattice with a total order relation, $s_j < \cdots s_{i+2} < s_{i+1} < s_i$, and satisfies Properties I′ and II′, then the file cutsets s_i, s_{i+1}, s_{i+2}, … s_j contain the same edge e_x. This means the files, f_i, f_{i+1}, f_{i+2}, … ,f_j, are in the same node of incident edge e_x. □

4.3. The experiment result for algorithm REL

In this paper, we want to see the real affection of different number of edges and data files of star topologies. We use an ALR PC with 386/33 CPU to run the algorithm REL. We have run 72 sets of

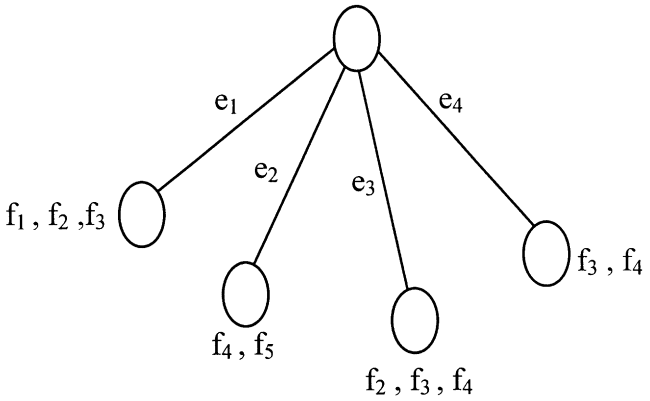


Fig. 7. An example for Theorem 5.

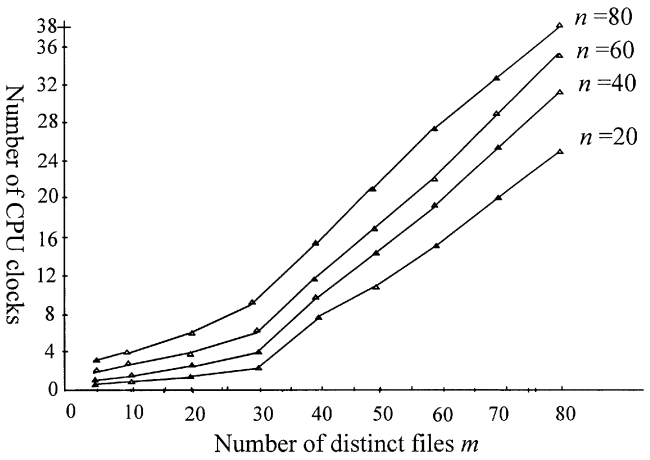


Fig. 8. The curve of CPU time of fixing number of edge n.

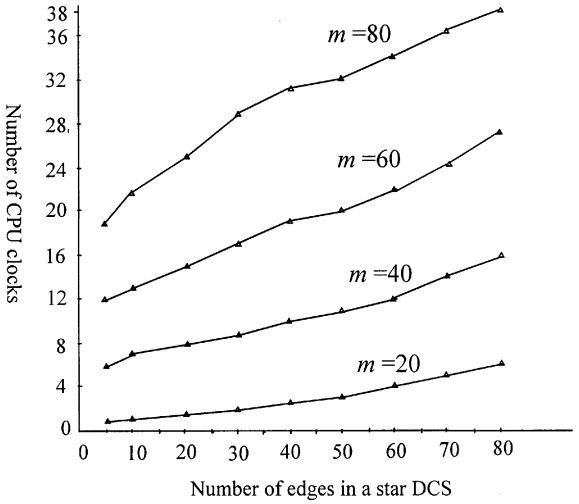


Fig. 9. The curve of CPU time of fixing number of distinction file m.

data files, generated by the consecutive file distribution property, on star topologies to show how the execution time varies according to the changes in the number of edges n and the number of distinct files m. One CPU clock time is 0.054945 s in Figs. 8 and 9. In Fig. 8, we fix the number of edges and vary the number of distinct files. In Fig. 9, we fix the number of distinct files and vary the number of edges.

5. Conclusions

In this paper, we investigated the problem of distributed program reliability on star distributed computing systems. We have shown that it is computationally intractable for star distributed computing systems. Furthermore, we identify one class of star topology, in which the file distribution is performed with respect to a consecutive property, for computing distributed program reliability can be done in polynomial time. We also propose a pseudo-polynomial time algorithm to compute the distributed program reliability over the class of a star topology. Theorem 4 is proposed to conclude that only two cases of semilattice on star distributed computing systems with $|FA_i| = 2$ can satisfy Properties I′ and II′. Finally, we propose Theorem 5, which is a method to distribute files on each node for computing DPR in polynomial time without $|FA_i| = 2$ restriction on star distributed computing systems.

References

[1] Satyanarayana A, Hagstrom JN. A new algorithm for the reliability analysis of multi-terminal networks. IEEE Transactions on Reliability 1981;R-30:325–34.
[2] Agrawal KK, Rai S. Reliability evaluation in computer-communication networks. IEEE Transactions on Reliability 1981;R-30:32–5.

[3] Grnarov AP, Gerla M. Multi-terminal reliability analysis of distributed processing system. Proceedings of the 1981 International Conference on Parallel Processing, August 1981:79–86.
[4] Kevin Wood R. Factoring algorithms for computing K-terminal network reliability. IEEE Transactions on Reliability 1986;R-35:269–78.
[5] Rosenthal A. A computer scientist looks at reliability computations. In: reliability and fault tree analysis. SLAM 1975;133–52.
[6] Prasanna Kumar VK, Hariri S, Raghavendra CS. Distributed program reliability analysis. IEEE Transactions on Software Engineering 1986;SE-12:42–50.
[7] Kumar A, Rai S, Agrawal DP. On computer communication network reliability under program execution constraints. IEEE JSAC 1988;6:1393–9.
[8] Valiant LG. The complexity of enumeration and reliability problems. SIAM Journal on Computing. 1979;8: 410–21.
[9] Lin MS. Program reliability analysis in distributed computing system. PhD Dissertation, National Chiao Tung University, Taiwan, ROC, 1994.
[10] Ball MO, Provan JS, Shier DR. Reliability covering problems. Networks 1991;21:345–57.
[11] Provan JS, Ball MO. The complexity of counting cuts and of computing the probability that a graph is connected. SIAM Journal on Computing 1983;12(4):777–88.
[12] Agrawal A, Barlow RE. A survey of network reliability and domination theory. Operations Research 1984;32:478–92.
[13] Kim Y, Case K, Ghare P. A method for computing complex system reliability. IEEE Transaction on Reliability 1972;R-21:215–9.
[14] Satyanarayana A, Prabhakar A. New topological formula and rapid algorithm for reliability analysis of complex networks. IEEE Transactions on Reliability 1978;R-27:82–100.
[15] Abel U, Bicker R. Determination of all minimal cut-sets between a vertex pair in an undirection graph. IEEE Transactions on Reliability 1982;R-31:167–71.
[16] Nelson A, Batt J, Beadles R. A computer program for approximating system reliability. IEEE Transactions on Reliability 1970;R-19:61–5.
[17] Provan JS, Ball MO. Computing network reliability in time polynomial in the number of cuts. Operations Research 1984;32:516–26.
[18] Shier DR. Algebraic aspects of computing network reliability. In: Ringeisen RD, Roberts FS, editors. Applications of discrete mathematics. Philadelphia, PA: SIAM, 1988:135–47.

Ming-Sang Chang received the BS degree in Electronic Engineering from National Taiwan University of Science and Technology (Taipei, Taiwan) and the MS degree in Information Engineering from TamKang University (Taipei, Taiwan) and PhD degree in Computer Science and Information Engineering from National Chiao Tung University (Hsin Chu, Taiwan). He is currently working in Chunghwa Telecommunication Training Institute (Taipei, Taiwan). His research interests include computer network, performance evaluation, distributed system, and reliability evaluation.

Deng-Jyi Chen received the BS degree in Computer Science from Missouri State University (Cape Girardeau), and the MS and PhD degrees in Computer Science from the University of Texas (Arlington). He is now a professor at National Chiao Tung University (Hsin Chu, Taiwan). He has published nearly 100 referred journal and conference papers in the area of reliability and performance modeling of distributed systems, computer networks, object-oriented systems, and softwere reuse. Professor Chen is a consultant for many local companies.

Min-Sheng Lin received his MS and PhD in Computer Science & Information Engineering from National Chiao Tung University (Hsin Chu, Taiwan). He is currently an associate professor at Tamsui Oxford University College (Taipei, Taiwan). His research interests include reliability and performance evaluation of distributed computing systems.

Kuo-Lung Ku received the BS degree in Control Engineering from National Chiao Tung University (Hsin Chu, Taiwan), the MS degree in Electronic Engineering from National Chiao Tung University (Hsin Chu, Taiwan), and PhD degree in Electric Engineering from the University of Washington (Seattle). He is currently working in Chung-Shan Institute of Science and Technology. His research interests include parallel computing, image processing, and computing system reliability.