International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **6**, June **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4045–4058

MAINTENANCE RELIABILITY OF A COMPUTER NETWORK WITH NODES FAILURE IN THE CLOUD COMPUTING ENVIRONMENT

Yi-Kuei Lin and Ping-Chen Chang

Department of Industrial Management
National Taiwan University of Science and Technology
No. 43, Sec. 4, Keelung Rd., Taipei 10607, Taiwan
{ yklin; D9901001 }@mail.ntust.edu.tw

Received March 2011; revised July 2011

Abstract. *With the development of information technology, the cloud computing service has become a new paradigm for the business and industry. In a cloud computing environment, the computer network (CN) can be constructed as a multistate network with several possible states due to failure, partial failure, or maintenance of edges (physical lines) and nodes (switches or routers). In order to guarantee the CN retains a good quality of service, the maintenance action is needed to be taken while the CN falls to a specific state such that it cannot provide sufficient capacity to meet clients' demand. This paper proposes a performance indicator, the maintenance reliability, to evaluate the capability that a CN can send d units of data from the cloud to the client through multiple minimal paths under both the maintenance budget and time constraints. An adjusting procedure based on the branch-and-bound approach is developed to evaluate the performance indicator. According to different maintenance budgets and the corresponding maintenance reliabilities, the system supervisor could determine a reasonable maintenance budget to maintain the CN for retaining a good quality level.*

Keywords: Maintenance reliability, Cloud computing, Computer network (CN), Branch-and-bound approach

1. **Introduction.** In recent years, the applications of computer system and Internet service have grown rapidly and explosively. For a stable usage environment, Internet service providers (ISPs) have to guarantee the computer system retains a good quality of service (QoS) and satisfy their customers/clients all the time. In this paper, we focus on the performance evaluation for a computer network (CN) in the cloud computing environment (CCE) since the cloud computing is broadly applied in our current information society. In a cloud computing paradigm, information is processed or stored by servers on the Internet and cached temporarily on clients [1]. All resources (computing capacity, storage capacity, or network bandwidth) are remotely provided by powerful servers which can be depicted as the "cloud". Thus, clients can submit their requirements from anywhere in the world. From the QoS perspective, a great deal of research [2-4] has been devoted to studying the performance evaluation of an Internet-based network. Chen and Lin [2] adopted the end-to-end delay equalizations to improve the playback speed for virtual path transmissions in Internet environments. Lin [3,4] presented an approach for scheduling the processors of the grid computing network in the Internet environment which assists in improving the computing efficiency. Nevertheless, these works mainly focus on measuring the time attribute rather than the network capability.

In the real world, an ISP company would be interested in evaluating the capability that a CN can send a given units of data from the source (cloud) to the sink (clients/end users). For practical cases, ISP companies can see the physical lines as edges and the switches/routers as nodes. Thus, a CN in the CCE can be illustrated as a network topology. Moreover, the capacity state of each edge and node should be stochastic (i.e., multistate) due to failure, partial failure, or maintenance. A CN characterized by such edges and nodes also has several possible states and it is also a multistate network [5-11]. To guarantee the CN retains a stable QoS, it should be maintained when falling to a specific state such that the cloud cannot provide sufficient capacity to fulfill the client's demand d. Hence, the maintenance budget should be considered. Yeh [11] defined the maintenance cost as the amount needed to restore a network from its failed state back to its original state, where the failed state is that the network sends less than the given d units of data. That is, the edges/nodes in the CN should be restored to their highest capacities when only d units of data can be sent.

However, transmission time is not considered in Yeh's [11] work. When data are transmitted through a CN, it is necessary to select a shortest delayed path to minimize the transmission time [12,13]. Nevertheless, the flow of data transmission is not considered in these works. In order to find a path to deliver the requested number of data from the source to the sink with minimum transmission time, Chen and Chin [14] proposed the quickest path problem. In this problem, both the capacity and the lead time are attributes of each edge and are assumed to be deterministic [14-16]. Variants of quickest path problems such as constrained quickest path problem [17,18], the first k quickest path problem [19,20], and all-pairs quickest path problem [21,22], have been subsequently proposed. Previous works, however, mainly concern the time attribute without considering the maintenance to retain the CN with a sufficient capacity state. Moreover, these literatures assume the nodes are perfect reliable. In a CN, nodes and edges can fail unexpectedly when malfunctions occur. Therefore, all of the failure, maintenance action, and transmission time on nodes are needed to be considered as well. Aggarwal et al. [23] proposed the concept that the failure of a node implies the failure of edges incident from it. Based on this concept, further related works modified the original network with node failure to be a conventional network with perfect nodes [6,24].

This paper addresses the performance evaluation of a CN in the CCE considering demand, maintenance budget, transmission time, and node failure case. The CN must be maintained to deliver a sufficient capacity level so that it can send at least d units of data from the cloud to the client within time T. To shorten the transmission time, the data can be transmitted through k ($k \geq 2$) disjoint minimal paths (MPs) simultaneously, in which an MP is a path whose proper subsets are no longer paths. The performance indicator is defined as the probability that the network can guarantee a sufficient capacity level to meet the demand on time and within budget B. This probability is, henceforth, referred to as maintenance reliability. We first generate all minimal capacity vectors satisfying the time constraint, and subsequently check if they satisfy the maintenance budget B or not. For those unqualified capacity vectors whose total costs exceed B, an adjusting procedure is adopted to elevate them instead of deleting them. A branch-and-bound approach is proposed to generate all (d,B,T)-MPs, the minimal capacity vectors fulfilling d, B, and T. The maintenance reliability is derived in terms of (d,B,T)-MPs by the Recursive Sum of Disjoint Products (RSDP) algorithm afterwards.

Based on the maintenance reliability, the system supervisor could conduct a sensitive analysis to improve or investigate the most important part in a large CN. The remainder of this paper is organized as follows. Section 2 addresses the notations and assumptions. Model formulation and the maintenance reliability are described in Section 3. Algorithms to generate the (d,B,T)-MPs are proposed in Section 4. Examples presented in Section 5 illustrate the algorithm and how the maintenance reliability may be calculated. Discussion on the algorithm and conclusion are summarized in Section 6.

2. **Notations and Assumptions.** Let $G = (\mathbf{E}, \mathbf{N}, \mathbf{W}, \mathbf{C}, \mathbf{L})$ denote a CN with a cloud S_{cloud} and a client S_{client} where $\mathbf{E} = \{e_i | i = 1, 2, \ldots, n\}$ represents the set of edges, $\mathbf{N} = \{e_i | i = n+1, n+2, \ldots, n+r\}$ represents the set of nodes, $\mathbf{W} = \{W_i | i = 1, 2, \ldots, n+r\}$ with the maximal capacity W_i of e_i, $\mathbf{C} = \{c_i | i = 1, 2, \ldots, n+r\}$ with unit maintenance cost c_i of e_i, and $\mathbf{L} = \{l_i | i = 1, 2, \ldots, n+r\}$ with lead time l_i of e_i. Suppose the data are transmitted through $P_1, P_2, \ldots, P_k$ simultaneously, where P_m is the mth MP for $m = 1, 2, \ldots, k$. The capacity vector $X = (x_1, x_2, \ldots, x_{n+r})$ is defined as the system state of G where x_i represents the current capacity of edge/node e_i. Such a G is assumed to further satisfy the following assumptions:

(1) The cloud node S_{cloud} and the client node S_{client} are perfectly reliable.
(2) The capacity of each edge/node is stochastic with a given probability distribution.
(3) The capacities of different edges/nodes are statistically independent.
(4) All data are transmitted through k MPs simultaneously to shorten the transmission time.

3. **The CN Model and Maintenance Reliability.** The capacity is the number of data sent through the edge/node/MP per unit of time. For each path, the maximal capacity of P_m is $\min\limits_{i:e_i \in P_m} (W_i)$, where $m = 1, 2, \ldots, k$. Similarly, under the capacity vector X, the capacity of P_m is $\min\limits_{i:e_i \in P_m} (x_i)$. The transmission time to send d units of data through P_m under the capacity vector X, $\Lambda(d, X, P_m)$, is

$$\text{lead time of } P_m + \left\lceil \frac{d}{\text{the capacity of } P_m} \right\rceil = \sum_{i:e_i \in P_m} l_i + \left\lceil \frac{d}{\min\limits_{i:e_i \in P_m} x_i} \right\rceil, \tag{1}$$

where $\lceil x \rceil$ is the smallest integer such that $\lceil x \rceil \geq x$. The transmission time under the capacity X contradicts the time constraint if $\Lambda(d, X, P_m) > \mathrm{T}$. We have the following lemma showing the relationship between capacity vector and transmission time.

Lemma 3.1. *$\Lambda(d, X, P_m) \geq \Lambda(d, Y, P_m)$ for the capacity vector Y if $X < Y$.*

Proof: If $X < Y$, then $x_i \leq y_i$ for each $e_i \in P_m$, and $\min\limits_{i:e_i \in P_m} x_i \leq \min\limits_{i:e_i \in P_m} y_i$. Thus, $\left\lceil \frac{d}{\min\limits_{i:e_i \in P_m} x_i} \right\rceil \geq \left\lceil \frac{d}{\min\limits_{i:e_i \in P_m} y_i} \right\rceil$. So $\Lambda(d, X, P_m) \geq \Lambda(d, Y, P_m)$. The proof is completed.

The data are transmitted through k disjoint MPs, say P_1, P_2, $\ldots$, P_k, simultaneously. The demand d assigned to each P_m is d_m, where $\sum\limits_{m=1}^{k} d_m = d$. For demand vector $\mathbf{d} = (d_1, d_2, \ldots, d_k)$, the minimum transmission time $\Gamma(\mathbf{d}, X)$ under the capacity vector X is

$$\Gamma(\mathbf{d}, X) = \max\{\Lambda(d_1, X, P_1), \Lambda(d_2, X, P_2), \ldots, \Lambda(d_k, X, P_k)\}. \tag{2}$$

Thus, the minimum transmission time to send d units of data under X is $\Delta(d, X) = \min\limits_{\text{all } \mathbf{d}: \sum_{m=1}^{k} d_m = d} \{\Gamma(\mathbf{d}, X)\}$. Any capacity vector X with $\Delta(d, X) \leq \mathrm{T}$ means that it satisfies d and T. For convenience, let Ψ_{T} be the set of the capacity vectors satisfying d and T, and let $\Psi_{\min,\mathrm{T}} = \{X | X \text{ is minimal capacity vector in } \Psi_{\mathrm{T}}\}$. That is, $\Psi_{\min,\mathrm{T}}$ is the set of the minimal capacity vectors satisfying d and T.

We further check the maintenance cost for each X in Ψ_{T}. The total cost to recover the edges/nodes in a CN from the state X is

$$TC(X) = \sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - x_i), \tag{3}$$

where $c_i(W_i - x_i)$ is the maintenance cost for e_i on any MP to restore from the current capacity x_i to its highest capacity W_i. In particular, only the edges/nodes appearing in the MPs are necessary to be restored. The following constraint shows that the total maintenance cost can not exceed the budget B,

$$\sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - x_i) \leq \mathrm{B}. \tag{4}$$

Any X with $TC(X) \leq \mathrm{B}$ and $\Delta(d, X) \leq \mathrm{T}$ means that X can send d units of data from S_{cloud} to S_{client} under time T and maintenance budget B. Let Ψ_{B} store all X fulfilling d, T, and B while $\Psi_{\mathrm{min,B}} = \{X | X \text{ is minimal in } \Psi_{\mathrm{B}}\}$.

Definition 3.1. *$X \in \Psi_{\mathrm{min},B}$ is named a (d,T,B)-MP, equivalently, X is a (d,T,B)-MP if and only if (i) Δ(d,X) $\leq$ T, (ii) TC(X) $\leq$ B, and (iii) Δ(d,Y) > T or TC(Y) > B for any capacity vector Y with Y < X.*

Hence, we have the following property for (d,T,B)-MPs.

Theorem 3.1. *If X is a (d,T,B)-MP, then $Y \in \Psi_B$ for any Y > X.*

Proof: (i) Since X is a (d, T, B)-MP, we obtain $\Delta(d, X) \leq \mathrm{T}$ and $TC(X) \leq \mathrm{B}$. Lemma 3.1 states that $\Lambda(d, Y, P_m) \leq \Lambda(d, X, P_m)$ for any $Y > X$, $m = 1, 2, \ldots, k$. Thus, $\max\{\Lambda(d_1, Y, P_1), \Lambda(d_2, Y, P_2), \ldots, \Lambda(d_k, Y, P_k)\} \leq \max\{\Lambda(d_1, X, P_1), \Lambda(d_2, X, P_2), \ldots, \Lambda(d_k, X, P_k)\}$, equivalently, $\Gamma(\mathbf{d}, Y) \leq \Gamma(\mathbf{d}, X)$. Then $\min_{\text{all } \mathbf{d}: \sum_{m=1}^{k} d_m = d} \{\Gamma(\mathbf{d}, Y)\} \leq \min_{\text{all } \mathbf{d}: \sum_{m=1}^{k} d_m = d} \{\Gamma(\mathbf{d}, X)\}$, and $\Delta(d, Y) \leq \Delta(d, X) \leq \mathrm{T}$.

(ii) Since $Y > X$, it implies that $y_i \geq x_i$ and $(W_i - y_i) \leq (W_i - x_i)$ for each e_i. Then $\sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - y_i) \leq \sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - x_i)$. We thus get the result $Y \in \Psi_{\mathrm{B}}$ by obtaining $TC(Y) \leq TC(X) \leq \mathrm{B}$. The proof is completed.

The maintenance reliability R_M is defined as the probability that a CN can send d units of data from S_{cloud} to S_{client} within time T and under the maintenance budget B; i.e., $R_M = \Pr\{X | \Delta(d,X) \leq \mathrm{T} \text{ and } TC(X) \leq \mathrm{B}\}$. Theorem 3.1 implies that $\Pr\{X | \Delta(d,X) \leq \mathrm{T} \text{ and } TC(X) \leq \mathrm{B}\} = \Pr\{X | X \geq X_v \text{ for a } (d,\mathrm{T},\mathrm{B})\text{-MP } X_v\}$. Suppose $X_1, X_2, \ldots, X_h$ are all (d,T,B)-MPs and thus the maintenance reliability R_M can be represented as $R_M = \Pr\{X | X \in \Psi_{\mathrm{B}}\} = \Pr\{\bigcup_{v=1}^{h} D_v\}$, where $D_v = \{X | X \geq X_v\}$ for $v = 1, 2, \ldots, h$. Several methods such as RSDP algorithm [7-9,25], inclusion-exclusion method [6,10,16,24,26], disjoint-event method [26,27], and state-space decomposition [5,28], may be applied to compute $\Pr\{\bigcup_{v=1}^{h} D_v\}$. The RSDP algorithm has a better computational efficiency than the state-space decomposition for a large network [7-9,25]. Hence, the RSDP algorithm is applied to derive maintenance reliability herein.

4. Algorithms.

4.1. The algorithm to generate all $(\boldsymbol{d}, \mathbf{T}, \mathbf{B})$-MPs.

Suppose that all P_m are given. Then all (d,T,B)-MPs can be generated by the following steps.

Step 0. [Initialization] Set $\Psi_{\min,\mathrm{T}} = \emptyset$, $\Psi_{\min,\mathrm{B}} = \emptyset$, and $j = 0$.

Step 1. Find the largest assigned demand $\overline{d_m}$ such that $\sum_{i:e_i \in P_m} l_i + \left\lceil \frac{\overline{d_m}}{\min_{i:e_i \in P_m} W_i} \right\rceil \leq \mathrm{T}$.

Step 2. [Obtain feasible demand vector $\mathbf{d}$] Generate all non-negative integer solutions of $\sum_{m=1}^{k} d_m = d$ where $d_m \leq \overline{d_m}$, $m = 1, 2, \ldots, k$.

Step 3. [Obtain $\Psi_{\min,\mathrm{T}}$] For each demand vector $\mathbf{d}$, do the following steps.

3.1 Find the minimal capacity v_m of P_m such that d_m units of data can be sent through P_m under T, $m = 1, 2, \ldots, k$. That is, find the smallest integers v_m such that

$$\sum_{i:e_i \in P_m} l_i + \left\lceil \frac{d_m}{v_m} \right\rceil \leq \mathrm{T}, \quad m = 1, 2, \ldots, k. \tag{5}$$

3.2 $j = j + 1$. $X_j = (x_1, x_2, \ldots, x_{n+r})$ is obtained according to

$$x_i = \begin{cases} \text{minimal capacity } u_i \text{ of } e_i \text{ such that } u_i \geq v_m & \text{if } e_i \in P_m \text{ for a } m \in \{1, 2, \ldots, k\}, \\ 0 & \text{if others.} \end{cases} \tag{6}$$

3.3 For $w = 1$ to $j - 1$, if $X_j \geq X_w$, then go to Step 3.5; if $X_j < X_w$, then $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \backslash X_w$.

3.4 $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \cup \{X_j\}$.

3.5 Next $\mathbf{d}$.

Step 4. [Generate $\Psi_{\min,\mathrm{B}}$] For each $X_j \in \Psi_{\min,\mathrm{T}}$, do the following steps.

4.1 Find the maintenance cost $TC(X_j) = \sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - x_i)$.

4.2 If $TC(X_j) \leq \mathrm{B}$, then $\Psi_{\min,\mathrm{B}} = \Psi_{\min,\mathrm{B}} \cup \{X_j\}$, $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \backslash X_j$. and go to Step 4.4. //Check the maintenance budget to obtain $\Psi_{\min,\mathrm{B}}$

4.3 If $TC(X_j) > \mathrm{B}$, do the following steps: //Adjusting procedure

4.3.1 $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \backslash X_j$.

4.3.2 For each $e_i \in \bigcup_{m=1}^{k} P_m$, let $X_{j,i} = X_j + \delta_i$. If the capacity of e_i in $X_{j,i}$ is $W_i + 1$, then remove $X_{j,i}$. //Adjust X_j by adding a standard basis vector δ_i, in which $x_i = 1$ and 0 for others.

4.3.3 Compare each $X_{j,i}$ with $X \in \Psi_{\min,\mathrm{B}}$. If $X_{j,i}$ is larger than or equal to any X in $\Psi_{\min,\mathrm{B}}$, then delete $X_{j,i}$; otherwise, $X_{j,i} \in \Psi_{\min,\mathrm{T}}$. If $X_{j,i}$ is less than an X in $\Psi_{\min,\mathrm{B}}$, delete that X from $\Psi_{\min,\mathrm{B}}$. //Check the qualification of the candidate.

4.3.4 Treat each $X_{j,i} \in \Psi_{\min,\mathrm{T}}$ as the role of X_j and go to Step 4.1. //Recursive procedure to repeat Step 4.

4.4 Next $X_j \in \Psi_{\min,\mathrm{T}}$.

Step 5. All $X_j \in \Psi_{\min,\mathrm{B}}$ are the minimal capacity vectors fulfilling d, T, and B.

Step 3 generates all minimal capacity vectors satisfying d and T and thus we obtain the set $\Psi_{\min,\mathrm{T}}$. For those whose maintenance budget exceeds B, Step 4 utilizes an adjusting procedure which applies a branch-and-bound approach (see Figure 1) to generate (d,T,B)-MPs. Each adjusted $X_{j,i}$ is generated from X_j by adding δ_i to reduce the total cost. The following lemma and theorem play critical roles to support the proposed algorithm.

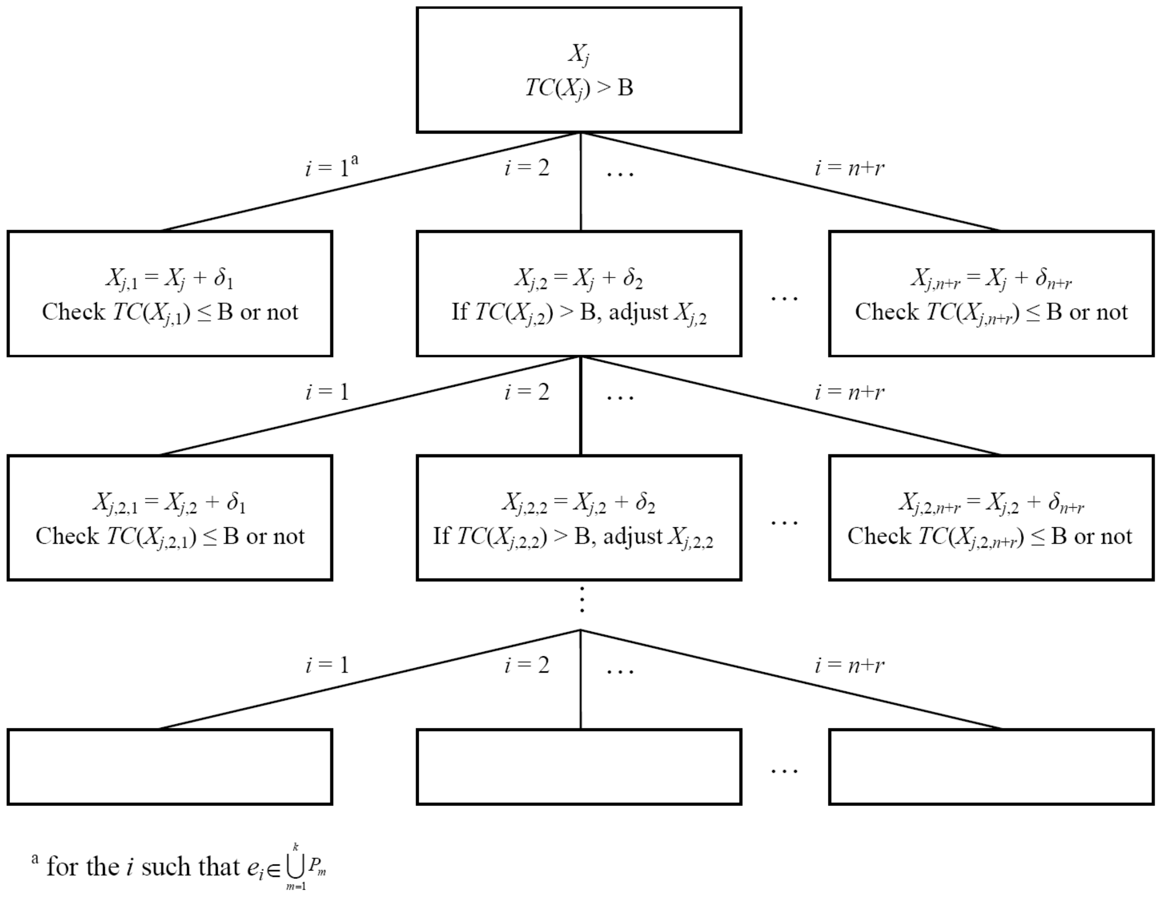

Figure 1. A search tree for the branch-and-bound approach to adjust X_j

Lemma 4.1. *Each $X_{j,i}$ generated from X_j is unnecessary to compare with each other for checking the qualification.*

Proof: Assume that the edges e_s, $e_t \in \bigcup_{m=1}^{k} P_m$, $s \neq t$. For e_s, the capacity level for δ_s is 1 and 0 for others; for e_t, the capacity level for δ_t is 1 and 0 for others. Both δ_s and δ_t are standard basis vectors, so we cannot find which one is the minimal vector with each other. Since $X_{j,s} = X_j + \delta_s$ and $X_{j,t} = X_j + \delta_t$, the comparison step between $X_{j,s}$ and $X_{j,t}$ is surplus. The proof is completed.

Theorem 4.1. *The set $\Psi_{\min,\mathrm{B}}$ generated from the algorithm is exactly the set of $(d, \mathrm{T}, \mathrm{B})$-MPs.*

Proof: Suppose X_j is not a (d,T,B)-MP, then there exists a (d,T,B)-MP $Y = (y_1, y_2, \ldots, y_{n+r})$ such that $Y < X_j$ because X_j fulfills d, T, and B (from Steps 3 and 5.2 of the algorithm). Without loss of generality, we assume an edge $e_u \in P_1$ such that $y_u < x_u$. It is known that x_u is the minimal capacity of e_u such that $x_u \geq v_1$. The situation $y_u < x_u$ results in that $y_u < v_1$ and $\sum_{i:e_i \in P_1} l_i + \left\lceil \frac{d_1}{y_u} \right\rceil > \mathrm{T}$. It contradicts that Y is a (d,T,B)-MP. Hence, X_j is a (d,T,B)-MP.

Conversely, we claim that every (d,T,B)-MP is generated from the algorithm. Let $\Psi_{\text{min,B}} = \{X_1, X_2, \ldots, X_w\}$ be generated from the algorithm. Suppose X is a (d,T,B)-MP and $X \notin \Psi_{\text{min,B}}$. Without loss of generality, there exists an edge $e_i \notin \bigcup_{m=1}^{k} P_m$ such that $x_i > 0$. Set $Y = (x_1, x_2, \ldots, x_i - z, \ldots, x_{n+r})$, where $(x_i - z)$ is the maximal capacity of e_i such that $(x_i - z) < x_i$. Then $\Lambda(d,Y,P_m) \leq \text{T}$ due to $Y > X$, where $m = 1, 2, \ldots, k$. That contradicts that X is a (d,T,B)-MP. Hence, any (d,T,B)-MP belongs to $\{X_1, X_2, \ldots, X_w\}$. We conclude that $\{$(d,T,B)-MPs$\}$ is exactly $\Psi_{\text{min,B}}$ generated from the algorithm. The proof is completed.

4.2. **The RSDP algorithm.** In terms of the minimal capacity vectors obtained in Section 4.1, the maintenance reliability R_{M} can be derived by the RSDP algorithm. The RSDP algorithm is a recursive algorithm combined by the sum of disjoint product principle [25]. In this algorithm, a maximum operator, "$\oplus$", is defined as

$$X_{1,2} = X_1 \oplus X_2 \equiv (\max(x_{1i}, x_{2i})) \quad \text{for } i = 1, 2, \ldots, n + r. \tag{7}$$

For example, suppose that two (d,T,B)-MPs, $X_1 = (1,\ 0,\ 1,\ 1,\ 0,\ 0,\ 1,\ 1)$ and $X_2 = (0,\ 0,\ 2,\ 0,\ 0,\ 0,\ 2,\ 2)$. Then, $X_{1,2} = X_1 \oplus X_2 = (\max(1,0), \max(0,0), \ldots, \max(1,2)) = (1, 0, 2, 1, 0, 0, 2, 2)$.

5. Illustrative Examples.

5.1. **Example 1.** To illustrate the solution process, we use a random CN with 12 edges and 6 failure nodes shown in Figure 2. In this example, each edge is combined with several Optical Carrier 18 (OC-18) lines and each line provides two possible capacities, 1 giga bits per second (Gbps) and 0 bits per second (bps). Since the lines are provided by different suppliers, the capacity of each edge follows a distinct probability distribution. Table 1 provides the capacity, lead time, and per unit maintenance cost of each edge/node.

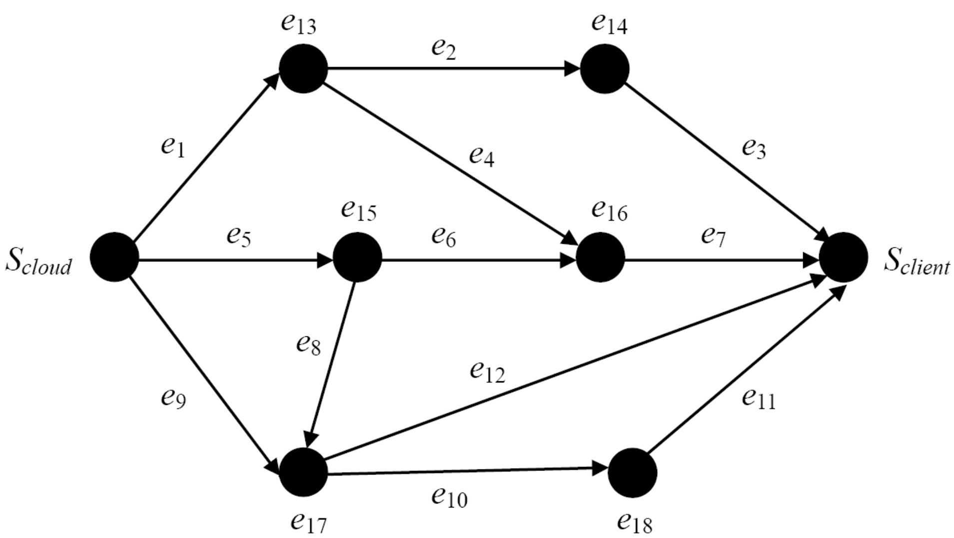

FIGURE 2. A random CN

Assume that the cloud have to send $d = 6$ Gbps data to the client through $P_1 = \{e_1, e_{13}, e_2, e_{14}, e_3\}$, $P_2 = \{e_5, e_{15}, e_6, e_{16}, e_7\}$, and $P_3 = \{e_9, e_{17}, e_{10}, e_{18}, e_{11}\}$ simultaneously within T = 10 seconds and B = 7500. It implies that the CN is falling to the failed state when the capacity level is less than 6 Gbps.

Step 0. Set $\Psi_{\text{min,T}} = \emptyset$, $\Psi_{\text{min,B}} = \emptyset$, and $j = 0$.

TABLE 1. The edge/node[a] data of Figure 2

Edge/ Node	Cost	Lead time (sec)	Capacity (Gbps)					
			0	1	2	3	4	5
e_1	250	3	0.001602	0.036262	0.273671	0.688465	0.000000	0.000000[b]
e_2	150	1	0.000008	0.000369	0.007022	0.066895	0.318631	0.607076
e_3	300	2	0.000857	0.024503	0.233422	0.741218	0.000000	0.000000
e_4	200	2	0.000146	0.004738	0.057506	0.310186	0.627422	0.000000
e_5	350	1	0.000187	0.005657	0.064039	0.322202	0.607915	0.000000
e_6	200	1	0.013689	0.206622	0.779689	0.000000	0.000000	0.000000
e_7	400	3	0.001602	0.036262	0.273671	0.688465	0.000000	0.000000
e_8	250	2	0.000016	0.000652	0.010543	0.085301	0.345082	0.558406
e_9	350	3	0.000081	0.003104	0.044350	0.281663	0.670802	0.000000
e_{10}	300	2	0.009025	0.171950	0.819025	0.000000	0.000000	0.000000
e_{11}	250	1	0.001602	0.036262	0.273671	0.688465	0.000000	0.000000
e_{12}	250	2	0.000146	0.004738	0.057506	0.310186	0.627422	0.000000
e_{13}	150	2	0.004096	0.064512	0.338688	0.592704	0.000000	0.000000
e_{14}	250	1	0.000207	0.006083	0.066908	0.327107	0.599695	0.000000
e_{15}	150	2	0.004096	0.064512	0.338688	0.592704	0.000000	0.000000
e_{16}	250	1	0.000207	0.006083	0.066908	0.327107	0.599695	0.000000
e_{17}	150	2	0.004096	0.064512	0.338688	0.592704	0.000000	0.000000
e_{18}	250	1	0.000207	0.006083	0.066908	0.327107	0.599695	0.000000

[a]e_1 to e_{12} for edges; e_{13} to e_{18} for nodes.
[b]The edge does not provide this capacity.

Step 1. The largest demand $\overline{d_1}$ such that $(l_1 + l_{13} + l_2 + l_{14} + l_3) + \left\lceil \frac{\overline{d_1}}{\min\{W_1, W_{13}, W_2, W_{14}, W_3\}} \right\rceil \leq 10$ is $\overline{d_1} = 3$. Similarly, we have the largest demand $\overline{d_2} = 4$ and $\overline{d_3} = 2$.

Step 2. Generate all non-negative integer solutions of $d_1 + d_2 + d_3 = 6$ where $d_1 \leq \overline{d_1}$, $d_2 \leq \overline{d_2}$, and $d_3 \leq \overline{d_3}$. The feasible (d_1, d_2, d_3) are (3,3,0), (3,2,1), (3,1,2), (2,4,0), (2,3,1), (2,2,2), (1,4,1), (1,3,2), and (0,4,2).

Step 3. For $(d_1,d_2,d_3) = (3,3,0)$, do the following steps.

3.1 The lead time of P_1 is $l_1 + l_{13} + l_2 + l_{14} + l_3 = 9$. Then $v_1 = 3$ is the smallest integer such that $\left(9 + \left\lceil \frac{3}{v_1} \right\rceil\right) \leq 10$. Similarly, $v_2 = 2$ and $v_3 = 0$.

3.2 $X_1 = (3, 3, 3, 0, 2, 2, 2, 0, 0, 0, 0, 0, 3, 3, 2, 2, 0, 0)$.

3.3 $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \cup \{X_1\} = \{X_1\}$.

3.4 Next (d_1,d_2,d_3).

$\vdots$

3.5 $\Psi_{\min,\mathrm{T}} = \{X_2, X_4, X_6, X_7, X_9\}$. The results are shown in Table 2.

Step 4. For $\Psi_{\min,\mathrm{T}} = \{X_2, X_4, X_6, X_7, X_9\}$, do the following steps.

4.1 For X_2, $TC(X_2) = 6550$.

4.2 $TC(X_2) = 6550 \leq \mathrm{B} = 7500$, so $\Psi_{\min,\mathrm{B}} = \Psi_{\min,\mathrm{B}} \cup \{X_2\} = \{X_2\}$, $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \backslash X_2 = \{X_4, X_6, X_7, X_9\}$.

4.1a For X_4, $TC(X_4) = 7600$.

4.2a $TC(X_4) = 7600 > \mathrm{B} = 7500$, so X_4 is needed to be adjusted.

4.3a Adjust X_4 by the following steps.

4.3.1a $\Psi_{\min,\mathrm{T}} = \Psi_{\min,\mathrm{T}} \backslash X_4 = \{X_6, X_7, X_9\}$.

4.3.2a For $i = 1, 2, 3, 5, 6, 7, 9, 10, 11$ (edges), and 13-18 (nodes), let $X_{4,i} = X_4 + \delta_i$. We get all $X_{4,i}$ as follows:

TABLE 2. Results of step 3 in Example 1

(d_1, d_2, d_3)	(v_1, v_2, v_3)	X	$X_j \in \Psi_{\min,\mathrm{T}}$ or not	Remark
(3,3,0)	(3,2,0)	$X_1 = (3,3,3,0,2,2,2,0,0,0,0,0,3,3,2,2,0,0)$	No	$X_1 > X_4$
(3,2,1)	(3,1,1)	$X_2 = (3,3,3,0,1,1,1,0,1,1,1,0,3,3,1,1,1,1)$	Yes	-
(3,1,2)	(3,1,2)	$X_3 = (3,3,3,0,1,1,1,0,2,2,2,0,3,3,1,1,2,2)$	No	$X_3 > X_2$
(2,4,0)	(2,2,0)	$X_4 = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$	Yes	-
(2,3,1)	(2,2,1)	$X_5 = (2,2,2,0,2,2,2,0,1,1,1,0,2,2,2,2,1,1)$	No	$X_5 > X_4$
(2,2,2)	(2,1,2)	$X_6 = (2,2,2,0,1,1,1,0,2,2,2,0,2,2,1,1,2,2)$	Yes	-
(1,4,1)	(1,2,1)	$X_7 = (1,1,1,0,2,2,2,0,1,1,1,0,1,1,2,2,1,1)$	Yes	-
(1,3,2)	(1,2,2)	$X_8 = (1,1,1,0,2,2,2,0,2,2,2,0,1,1,2,2,2,2)$	No	$X_8 > X_7$
(0,4,2)	(0,2,2)	$X_9 = (0,0,0,0,2,2,2,0,2,2,2,0,0,0,2,2,2,2)$	Yes	-

$X_{4,1} = X_4 + \delta_1 = (\underline{2},2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0) + (\underline{1},0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) = (3,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$;
$X_{4,2} = X_4 + \delta_2 = (2,\underline{3},2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$;

$\vdots$

$X_{4,18} = X_4 + \delta_{18} = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,\underline{1})$.
The capacity x_6 in $X_{4,6}$ is larger than the maximal capacity $W_6 = 2$, so it is not feasible. Hence, $X_{4,6}$ should be removed.

4.3.3a $\Psi_{\min,\mathrm{B}} = \{X_2\}$. Since no $X_{4,i}$ is larger than or equal to any X in $\Psi_{\min,\mathrm{B}}$, no $X_{4,i}$ is deleted in this step. So, $\Psi_{\min,\mathrm{T}} = \{X_{4,1}, X_{4,2}, X_{4,3}, X_{4,5}, X_{4,7}, X_{4,9}, X_{4,10}, X_{4,11}, X_{4,13}, X_{4,14}, X_{4,15}, X_{4,16}, X_{4,17}, X_{4,18}, X_6, X_7, X_9\}$.

4.3.4 Treat each X in $\Psi_{\min,\mathrm{T}}$ as X_j and go to Step 5.1, respectively.

4.1b For $X_{4,1}$, $TC(X_{4,1}) = 7350$.

$\vdots$

The results are summarized in Table 3.

TABLE 3. Results of step 4 in Example 1

X	$X_{j,i} \in \Psi_{\min,B}$ or not	Total Cost	Remark
$X_2 = (3,3,3,0,1,1,1,0,1,1,1,0,3,3,1,1,1,1)$	Yes	655	-
$X_{4,1} = (3,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$	Yes	735	-
$X_{4,2} = (2,3,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$	Yes	745	-
$X_{4,3} = (2,2,3,0,2,2,2,0,0,0,0,0,2,2,2,2,0,0)$	Yes	730	-
$X_{4,5} = (2,2,2,0,3,2,2,0,0,0,0,0,2,2,2,2,0,0)$	Yes	725	-
$X_{4,6} = (2,2,2,0,2,3,2,0,0,0,0,0,2,2,2,2,0,0)$	No	-	$x_6 > W_6$
$X_{4,7} = (2,2,2,0,2,2,3,0,0,0,0,0,2,2,2,2,0,0)$	Yes	720	-
$X_{4,9} = (2,2,2,0,2,2,2,0,1,0,0,0,2,2,2,2,0,0)$	Yes	725	-
$X_{4,10} = (2,2,2,0,2,2,2,0,0,1,0,0,2,2,2,2,0,0)$	Yes	730	-
$X_{4,11} = (2,2,2,0,2,2,2,0,0,0,1,0,2,2,2,2,0,0)$	Yes	735	-
$X_{4,13} = (2,2,2,0,2,2,2,0,0,0,0,0,3,2,2,2,0,0)$	Yes	745	-
$X_{4,14} = (2,2,2,0,2,2,2,0,0,0,0,0,2,3,2,2,0,0)$	Yes	735	-
$X_{4,15} = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,3,2,0,0)$	Yes	745	-
$X_{4,16} = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,2,3,0,0)$	Yes	735	-
$X_{4,17} = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,4,0)$	Yes	745	-
$X_{4,18} = (2,2,2,0,2,2,2,0,0,0,0,0,2,2,2,2,0,4)$	Yes	735	-
$X_6 = (2,2,2,0,1,1,1,0,2,2,2,0,2,2,1,1,2,2)$	Yes	635	-
$X_7 = (1,1,1,0,2,2,2,0,1,1,1,0,1,1,2,2,1,1)$	Yes	740	-
$X_9 = (0,0,0,0,2,2,2,0,2,2,2,0,0,0,2,2,2,2)$	Yes	720	-

Step 5. The set $\Psi_{\min,\mathrm{B}} = \{X_2, X_{4,1}, X_{4,2}, X_{4,3}, X_{4,5}, X_{4,7}, X_{4,9}, X_{4,10}, X_{4,11}, X_{4,13}, X_{4,14}, X_{4,15}, X_{4,16}, X_{4,17}, X_{4,18}, X_6, X_7, X_9\}$ is the set of the minimal capacity vectors fulfilling $d = 6$, $\mathrm{B} = 7500$ and $\mathrm{T} = 10$.

We subsequently obtain the maintenance reliability $R_\mathrm{M} = 0.770811759747860$ by the RSDP algorithm [25].

5.2. **Example 2.** We employ the Taiwan Academic network (TANET) with 31 edges and 25 nodes shown in Figure 3 [29] to demonstrate the utility of the approach for assessing practical case. The TANET is the backbone network that connects all educational and academic organizations in Taiwan. Consider the case where NTU and NSYSU are the source and sink, respectively. The capacity, lead time, and per unit maintenance cost of each edge are given in Table 4. For the case where the TANET has to preserve a minimal service level that delivers at least 20 Gbps of data from the source to the sink through $P_1 = \{e_1, e_{32}, e_2, e_{33}, e_3, e_{34}, e_4, e_{35}, e_5, e_{36}, e_6, e_{37}, e_7, e_{38}, e_8, e_{39}, e_9, e_{40}, e_{10}, e_{41}, e_{11}, e_{42}, e_{12}, e_{43}, e_{13}\}$ and $P_2 = \{e_{22}, e_{49}, e_{23}, e_{50}, e_{24}, e_{51}, e_{25}, e_{52}, e_{26}, e_{53}, e_{27}, e_{54}, e_{28}\}$ in no more than 35 seconds. Given a maintenance budget of 30,000 USD, the maintenance reliability is $R_\mathrm{M} = 0.789893606527573$.

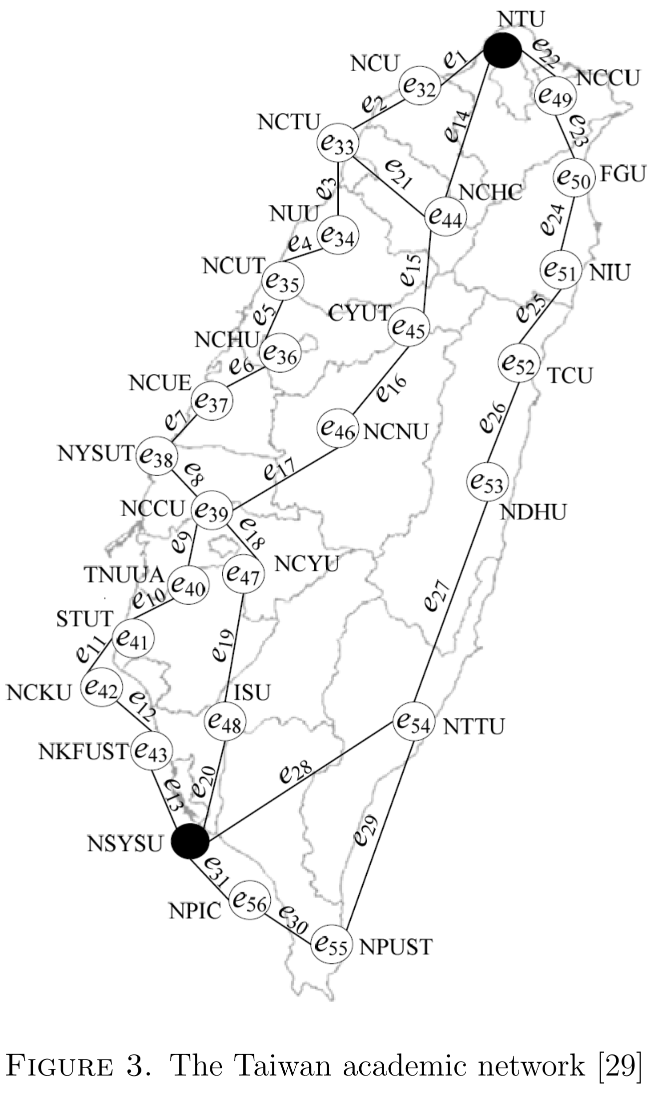

FIGURE 3. The Taiwan academic network [29]

TABLE 4. The edge and node data of Figure 3

Edge/Node	Cost	Lead time (sec)	Capacity (Gbps)					
			0	1	2	3	4	5
e_1	350	1	0.000064	0.004608	0.110592	0.884736	0.000000	0.000000
e_2	380	1	0.000343	0.013671	0.181629	0.804357	0.000000	0.000000
e_3	360	2	0.000100	0.003600	0.048600	0.291600	0.656100	0.000000
e_4	350	1	0.000041	0.001884	0.032502	0.249180	0.716393	0.000000
e_5	360	1	0.000216	0.010152	0.159048	0.830584	0.000000	0.000000
e_6	380	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_7	350	2	0.001000	0.027000	0.243000	0.729000	0.000000	0.000000
e_8	380	1	0.000008	0.001176	0.057624	0.941192	0.000000	0.000000
e_9	360	1	0.000064	0.004608	0.110592	0.884736	0.000000	0.000000
e_{10}	380	1	0.000027	0.002619	0.084681	0.912673	0.000000	0.000000
e_{11}	350	2	0.000064	0.004608	0.110592	0.884736	0.000000	0.000000
e_{12}	380	1	0.000343	0.013671	0.181629	0.804357	0.000000	0.000000
e_{13}	360	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{14}	300	3	0.000010	0.000450	0.008100	0.072900	0.328050	0.590490
e_{15}	340	2	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{16}	280	1	0.000024	0.001276	0.025428	0.225220	0.748052	0.000000
e_{17}	280	1	0.000008	0.001176	0.057624	0.941192	0.000000	0.000000
e_{18}	220	2	0.000024	0.001276	0.025428	0.225220	0.748052	0.000000
e_{19}	240	2	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{20}	250	1	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{21}	320	3	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{22}	360	1	0.008100	0.163800	0.828100	0.000000	0.000000	0.000000
e_{23}	320	1	0.002500	0.095000	0.902500	0.000000	0.000000	0.000000
e_{24}	380	1	0.001600	0.076800	0.921600	0.000000	0.000000	0.000000
e_{25}	300	2	0.003600	0.112800	0.883600	0.000000	0.000000	0.000000
e_{26}	360	2	0.004900	0.130200	0.864900	0.000000	0.000000	0.000000
e_{27}	380	1	0.004900	0.130200	0.864900	0.000000	0.000000	0.000000
e_{28}	350	3	0.003600	0.112800	0.883600	0.000000	0.000000	0.000000
e_{29}	220	2	0.000006	0.000475	0.013538	0.171475	0.814506	0.000000
e_{30}	240	3	0.000006	0.000475	0.013538	0.171475	0.814506	0.000000
e_{31}	300	1	0.000041	0.001884	0.032502	0.249180	0.716393	0.000000
e_{32}	350	1	0.000343	0.013671	0.181629	0.804357	0.000000	0.000000
e_{33}	360	1	0.000216	0.010152	0.159048	0.830584	0.000000	0.000000
e_{34}	400	1	0.000216	0.010152	0.159048	0.830584	0.000000	0.000000
e_{35}	420	2	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{36}	450	1	0.000013	0.000812	0.019086	0.199340	0.780749	0.000000
e_{37}	380	1	0.000024	0.001276	0.025428	0.225220	0.748052	0.000000
e_{38}	400	2	0.000729	0.022113	0.223587	0.753571	0.000000	0.000000
e_{39}	400	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{40}	360	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{41}	400	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{42}	420	2	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{43}	400	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{44}	380	3	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{45}	420	3	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{46}	440	2	0.000064	0.004608	0.110592	0.884736	0.000000	0.000000
e_{47}	450	1	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{48}	480	3	0.000125	0.007125	0.135375	0.857375	0.000000	0.000000
e_{49}	420	3	0.006400	0.147200	0.846400	0.000000	0.000000	0.000000
e_{50}	450	1	0.006400	0.147200	0.846400	0.000000	0.000000	0.000000
e_{51}	400	1	0.004900	0.130200	0.864900	0.000000	0.000000	0.000000
e_{52}	460	2	0.006400	0.147200	0.846400	0.000000	0.000000	0.000000
e_{53}	400	1	0.004900	0.130200	0.864900	0.000000	0.000000	0.000000
e_{54}	420	1	0.006400	0.147200	0.846400	0.000000	0.000000	0.000000
e_{55}	450	1	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082
e_{56}	450	2	0.000003	0.000188	0.004334	0.049836	0.286557	0.659082

6. **Discussion and Conclusion.** In Example 1, the total maintenance cost $TC(X_4) = 7600$ exceeds the budget B = 7500. Intuitively, we may delete the unqualified capacity vector X_4, which implies that the set $D_4 = \{X|X \geq X_4\}$ is also removed. However, this deleting action unexpectedly removes some X fulfilling d, T, and B. For instance, both $X_{4,1}$ and $X_{4,10}$ adjusted from X_4 are also removed if X_4 is deleted (see Figure 4). In fact, both $X_{4,1}$ and $X_{4,10}$ satisfy not only the maintenance budget but also the demand and time constraints because they are larger than X_4. Besides, neither $X_{4,1}$ nor $X_{4,10}$ are included in any other set D_v. Thus, the maintenance reliability would be underestimated if $X_{4,1}$ and $X_{4,10}$ are neglected.

For other budget cases, Table 5 shows the number of (d,T,B)-MPs and the maintenance reliability by adjusting unqualified capacity vectors and deleting unqualified capacity vectors. In experiment 3, the maintenance reliability is underestimated to be 0.747125520745587 if X_1 is deleted, while the exact maintenance reliability R_{M} is 0.770811759747860. Other experiments show that the number of (d,T,B)-MPs by deleting unqualified capacity vectors would be decreasing with the tighten budgets and thus the maintenance reliability would be underestimated. Therefore, the adjusting procedure is valid and necessary for evaluating maintenance reliability.

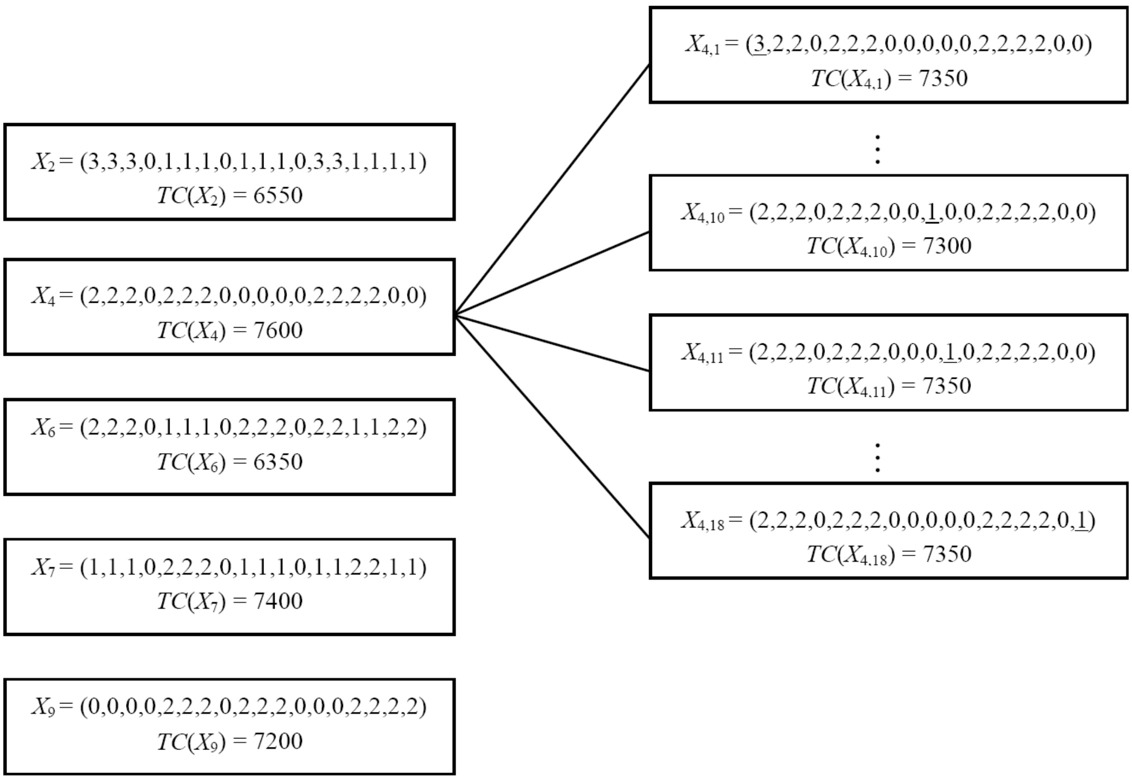

FIGURE 4. Results of X_j and branched X_j for Example 1

In the CN, edges and nodes have various capacities due to failure or maintenance and thus the CN has several possible states. We model the CN as a multistate network with capacity vector X and MPs to describe the flows through the CN, where the maintenance action is taken while the network falls to the failed state such that it cannot provide sufficient capacity to satisfy demand d. Furthermore, the transmission time to send the data from the cloud to the client is also addressed from QoS viewpoint. The maintenance reliability R_{M} is treated as a performance indicator to evaluate the probability that a CN can send d units of data from the cloud to the client through several MPs simultaneously under maintenance budget and time constraints. Under different maintenance budgets and the corresponding maintenance reliabilities, the system supervisor could further determine a reasonable maintenance budget to maintain the CN for keeping a good quality level. According to the maintenance reliability, the system supervisor could conduct a sensitive analysis to identify the most important edge and node improvements, which would have the most significant impact on the maintenance reliability of a large CN.

Table 5. Comparison of adjusting unqualified capacity vectors and deleting them for Example 1

Experiment[a]	Budget	Adjusting unqualified capacity vectors		Deleting unqualified capacity vectors	
		Number of (d,T,B)-MPs	Maintenance reliability	Number of (d,T,B)-MPs	Maintenance reliability
1	8000	5	0.770811759747862	5	0.770811759747862
2	7750	5	0.770811759747862	5	0.770811759747862
3	7500	18	0.770811759747860	4	0.747125520745587
4	7250	86	0.770811759747841	3	0.583432444739906
5	7000	408	0.770811759747822	2	0.525156484084328
6	6750	1566	0.770811759747743	2	0.525156484084328
7	6500	5018	0.770811759747663	1	0.441757117934860

[a]Fix $d = 6$ and $\mathrm{T} = 10$ for all experiments.

Acknowledgment. This work was supported in part by the National Science Council, Taiwan, under Grant No. NSC 99-2221-E-011-066-MY3.

REFERENCES

[1] C. Hewitt, ORGs for scalable, robust, privacy-friendly client cloud computing, *IEEE Internet Computing*, vol.12, no.5, pp.96-99, 2008.

[2] R. C. Chen and C. M. Lin, Adaptive end-to-end delay equalizations for TCP virtual path transmissions in Internet environments, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.1069-1078, 2010.

[3] J. F. Lin, Performance analysis and discussion on a heuristic approach for scheduling multiprocessor tasks in a grid computing environment, *International Journal of Innovative Computing, Information and Control*, vol.6, no.12, pp.5451-5462, 2010.

[4] J. F. Lin, Scheduling parallel tasks with intra-communication overhead in a grid computing environment, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.881-896, 2011.

[5] C. C. Jane, J. S. Lin and J. Yuan, Reliability evaluation of a limited-flow network in terms of minimal cutsets, *IEEE Transactions on Reliability*, vol.42, no.3, pp.354-361, 1993.

[6] Y. K. Lin, Reliability evaluation for an information network with node failure under cost constraint, *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, vol.37, no.2, pp.180-188, 2007.

[7] Y. K. Lin and C. T. Yeh, Evaluation of optimal network reliability under components assignments subject to a transmission budget, *IEEE Transactions on Reliability*, vol.59, no.3, pp.539-550, 2010.

[8] Y. K. Lin and C. T. Yeh, Optimal resources assignment to maximize multistate network reliability, *Computers and Operations Research*, vol.37, no.12, pp.2229-2238, 2010.

[9] Y. K. Lin and C. T. Yeh, Optimal carrier selection based on network reliability criterion for stochastic logistics networks, *International Journal of Production Economics*, vol.128, no.2, pp.510-517, 2010.

[10] J. Xue, On multistate system analysis, *IEEE Transactions on Reliability*, vol.34, no.4, pp.329-337, 1985.

[11] W. C. Yeh, Multistate network reliability evaluation under the maintenance cost constraint, *International Journal of Production Economics*, vol.88, no.1, pp.73-83, 2004.

[12] L. D. Bodin, B. L. Golden, A. A. Assad and M. O. Ball, Routing and scheduling of vehicles and crews: The state of the art, *Computers and Operations Research*, vol.10, no.2, pp.63-212, 1983.

[13] B. L. Goldena and T. L. Magnanti, Deterministic network optimization: A bibliography, *Networks*, vol.7, no.2, pp.149-183, 1977.
[14] Y. L. Chen and Y. H. Chin, The quickest path problem, *Computers and Operations Research*, vol.17, no.2, pp.153-161, 1990.
[15] Y. C. Hung and G. H. Chen, Distributed algorithms for the quickest path problem, *Parallel Computing*, vol.18, no.7, pp.823-834, 1992.
[16] E. D. Q. V. Martins and J. L. E. D. Santos, An algorithm for the quickest path problem, *Operations Research Letters*, vol.20, no.4, pp.195-198, 1997.
[17] G. H. Chen and Y. C. Hung, Algorithms for the constrained quickest path problem and the enumeration of quickest paths, *Computers and Operations Research*, vol.21, no.2, pp.113-118, 1994.
[18] Y. L. Chen and K. Tang, Minimum time paths in a network with mixed time constraints, *Computers and Operations Research*, vol.25, no.10, pp.793-805, 1998.
[19] J. C. N. Clímaco, M. M. B. Pascoal, J. M. F. Craveirinha and M. E. V. Captivo, Internet packet routing: Application of a k-quickest path algorithm, *European Journal of Operational Research*, vol.181, no.3, pp.1045-1054, 2007.
[20] M. M. B. Pascoal, M. E. V. Captivo and J. C. N. Clímaco, An algorithm for ranking quickest simple paths, *Computers and Operations Research*, vol.32, no.3, pp.509-520, 2005.
[21] G. H. Chen and Y. C. Hung, On the quickest path problem, *Information Processing Letters*, vol.46, no.3, pp.125-128, 1993.
[22] D. T. Lee and E. Papadopoulou, The all-pairs quickest path problem, *Information Processing Letters*, vol.45, no.5, pp.261-267, 1993.
[23] K. K. Aggarwal, J. S. Gupta and K. B. Misra, A simple method for reliability evaluation of a communication system, *IEEE Transactions on Communications*, vol.23, no.5, pp.563-566, 1975.
[24] Y. K. Lin, Reliability of a stochastic-flow network with unreliable branches & nodes under budget constraints, *IEEE Transactions on Reliability*, vol.53, no.3, pp.381-387, 2004.
[25] M. J. Zuo, Z. Tian and H. Z. Huang, An efficient method for reliability evaluation of multistate networks given all minimal path vectors, *IIE Transactions*, vol.39, no.8, pp.811-817, 2007.
[26] J. C. Hudson and K. C. Kapur, Reliability bounds for multistate systems with multistate components, *Operations Research*, vol.33, no.1, pp.153-160, 1985.
[27] R. Yarlagadda and J. Hershey, Fast algorithm for computing the reliability of communication network, *International Journal of Electronics*, vol.70, no.3, pp.549-564, 1991.
[28] C. Alexopoulos, A note on state-space decomposition methods for analyzing stochastic flow networks, *IEEE Transactions on Reliability*, vol.44, no.2, pp.354-357, 1995.
[29] Y. K. Lin and C. T. Yeh, Using minimal cuts to optimize network reliability for a stochastic computer subject to assignment budget, *Computers and Operations Research*, vol.38, no.8, pp.1175-1187, 2011.