International Journal of
Operations Research

International Journal of Operations Research Vol. 7, No. 1, 53-60 (2010)

Estimation of Maintenance Reliability for a Cloud Computing Network

Yi-Kuei Lin [1,*] **and Ping-Chen Chang** [2]

[1,2]Department of Industrial Management, National Taiwan University of Science & Technology

Taipei 106, Taiwan, R.O.C.

Received April 2010; *Revised May 2010*; *Accepted July 2010*

Abstract— Considering nodes failure cases, this paper mainly proposes an algorithm for the cloud computing network (CCN) to evaluate the capability that the CCN can send d units of data from the cloud to the client through two paths under both the maintenance budget and time constraints. To guarantee a good quality of service (QoS), the CCN should be maintained while falling to a failed state such that it cannot afford enough capacity to satisfy demand. Thus, the maintenance reliability is proposed in this paper. To estimate the maintenance reliability, a bounding approach is utilized to generate two sets of capacity vectors, {UB-MPs} and {LB-MPs}, where a UB-MP is the minimal capacity vector satisfying demand d and time constraint T while a LB-MP is the minimal capacity vector satisfying demand d, maintenance budget B, and time constraint T. Subsequently, the upper and lower bounds of maintenance reliability can be computed in terms of such vectors by applying the recursive sum of disjoint products algorithm.

Keywords — Maintenance reliability, Node failure, Cloud computing network (CCN), Estimation, Minimal paths.

1. INTRODUCTION

In a cloud computing paradigm, information is processed or stored by servers on the internet and cached temporarily on clients (Hewitt, 2008). Moreover, the cloud computing is developed for the enormous requirements in which the "cloud" is structured by powerful servers providing resources (computing, storage, or network bandwidth). For a practical cloud computing network (CCN), the capacity of each edge (physical lines, fiber optics, or coaxial cables) and node (servers or switches) should be stochastic due to failure, partial failure, or maintenance. That is, the CCN with each edge/node having several possible capacities or states is a typical stochastic-flow network (Jane et al., 1993; Lin, 2001, 2004, 2007, 2010; Xue, 1985; Yeh, 2004; Zuo et al., 2007).

To guarantee the CCN keeps a stable quality of service (QoS), it should be maintained when falling to a failed state such that the cloud cannot provide enough capacity to fulfill the client's demand *d*. Yeh (2004) defined the maintenance cost as the amount of restoring a network from its failed state back to its original state, where the original state implies that edges/nodes are with their highest capacities. Hence, the maintenance budget should be considered. The transmission time that data transmitted through the CCN is another important issue to be concerned. When data are transmitted through a CCN, it is desirable to select a shortest delayed path to minimize the transmission time (Bodin et al., 1982; Golden and Magnanti, 1977). However, the flow of data transmission is not considered in these works. In order to find a path which sends the given amount of data from the source (cloud) to the sink (client) with minimum transmission time, Chen and Chin (1990) proposed a version of the shortest path problem called the quickest path problem. In such problem, both the capacity and the lead time are involved in each edge and are assumed to be deterministic (Chen and Chin, 1990; Hung and Chen, 1992; Martins and Santos, 1997). Since then, several related researches of quickest path problems are proposed thereafter (Chen and Hung, 1994; Chen and Tang, 1998, Clímaco et al., 2007; Pascoal et al., 2005, Chen and Hung, 1993; Lee and Papadopoulou, 1993). To shorten the transmission time, the data can be transmitted through several disjoint minimal paths (MPs) simultaneously, in which a MP is a path whose proper subsets are no longer paths. For convenience, we concentrate on two MPs case in this paper. The proposed algorithm can then be easily extended to multiple MPs case. However, these literatures assume the nodes are perfect reliable.

In the CCN, nodes play the role as servers/switches and they would be failure due to unexpectedly malfunction as well as edges. Therefore, all of the failure, maintenance action, and transmission time on nodes are needed to be considered as well. Aggarwal et al. (1975) 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 (Lin, 2001, 2004, 2007). Consider with node failure cases, we propose an algorithm to estimate the probability that the CCN can send *d* units of data from the cloud to the client under both maintenance budget B and time constraint T. Such a probability is named the maintenance reliability herein. That is, *d*, B, and T would be the main factors (or say decision variables)

* Corresponding author's email: yklin@mail.ntust.edu.tw

to be controlled for obtaining the reasonable maintenance reliability and further sensitive analysis. A bounding approach is utilized to generate two sets of capacity vectors; {UB-MPs} and {LB-MPs}, where a UB-MP is the minimal capacity vector satisfying d and T while a LB-MP is the minimal capacity vector satisfying d, B, and T. The estimation of maintenance reliability is derived in terms of UB-MPs and LB-MPs by the Recursive Sum of Disjoint Products (RSDP) algorithm afterwards.

2. MODEL FORMULATION AND MAINTENANCE RELIABILITY

Notations

S_{cloud}	cloud node
S_{client}	client node
n	number of edges
r	number of nodes except for S_{cloud} and S_{client}
$\mathbf{E}$	$\{e_i \mid i = 1, 2, \ldots, n\}$: the set of edges
$\mathbf{N}$	$\{e_i \mid i = n + 1, n + 2, \ldots, n + r\}$: the set of nodes except for S_{cloud} and S_{client}
e_i	ith edge/node, $i = 1, 2, \ldots, n + r$
l_i	lead time of e_i, $i = 1, 2, \ldots, n + r$: transmission time required to pass through e_i
$\mathbf{L}$	$\{l_i \mid i = 1, 2, \ldots, n + r\}$
c_i	per unit maintenance cost of e_i, $i = 1, 2, \ldots, n + r$
$\mathbf{C}$	$\{c_i \mid i = 1, 2, \ldots, n + r\}$
W_i	maximal capacity of e_i, $i = 1, 2, \ldots, n + r$
x_i	current capacity of e_i where x_i is a non-negative integer, $i = 1, 2, \ldots, n + r$
X	$(x_1, x_2, \ldots, x_{n+r})$: the capacity vector
$\mathbf{W}$	$(W_1, W_2, \ldots, W_{n+r})$: the maximal capacity vector
G	$(\mathbf{N}, \mathbf{E}, \mathbf{C}, \mathbf{L}, \mathbf{W})$: a CCN
k	number of MPs
P_m	mth MP, $m = 1, 2, \ldots, k$
d	demand at S_{client}
$\zeta(d,X,P_m)$	transmission time to send d units of data through P_m under the capacity vector X, $m = 1, 2$
$\lceil x \rceil$	smallest integer that is not less than x
d_1	assigned demand to first MP
d_2	assigned demand to second MP
B	maintenance budget
$TC(X)$	total cost to maintain the edges from the state X
$\Gamma(d_1,d_2,X)$	the minimum transmission time to send d_1 and d_2 through P_1 and P_2, respectively, under the capacity vector X
$\Lambda(d,X)$	minimum transmission time to send d units of data under X
T	time constraint
MR_{UB}	upper bound of maintenance reliability
MR_{LB}	lower bound of maintenance reliability
MR_{EX}	(exact) maintenance reliability
Φ_{T}	set of X satisfying d and T
Φ_{UB}	set of the minimal X satisfying d and T. $\Phi_{\mathrm{UB}} = \{X \mid X \text{ is minimal in } \Phi_{\mathrm{T}}\}$
Φ_{B}	set of X satisfying d, B, and T
Φ_{EX}	set of the minimal X fulfilling d, B, and T. $\Phi_{\mathrm{EX}} = \{X \mid X \text{ is minimal in } \Phi_{\mathrm{B}}\}$
Φ_{LB}	set of $X \in \Phi_{\mathrm{UB}}$ satisfying B. $\Phi_{\mathrm{LB}} = \Phi_{\mathrm{UB}} \backslash \{X_j \mid TC(X_j) > \mathrm{B}, X_j \in \Phi_{\mathrm{UB}}\}$
X_j	$(x_{j1}, x_{j2}, \ldots, x_{j(n+r)})$: the jth capacity vector
h	number of UB-MPs
D_v	$\{X \mid X \geq X_v\}$: a subset of X, $v = 1, 2, \ldots, h$

Nomenclatures

CCN	cloud computing network
MP	minimal path
QoS	quality of service
RSDP	recursive sum of disjoint products

Vector operations are done according to the following rules:

$Y \geq X$ $(y_1, y_2, \ldots, y_{n+r}) \geq (x_1, x_2, \ldots, x_{n+r})$: $y_i \geq x_i$ for each $i = 1, 2, \ldots, n + r$;

$Y > X$ $(y_1, y_2, \ldots, y_{n+r}) > (x_1, x_2, \ldots, x_{n+r})$: $Y \geq X$ and $y_i > x_i$ for at least one i;

Lin and Chang: *Estimation of Maintenance Reliability for a Cloud Computing Network*
IJOR Vol. 7, No. 1, 53−60 (2010)

$Y + X \quad (y_1, y_2, \ldots, y_{n+r}) + (x_1, x_2, \ldots, x_{n+r}) = (y_1 + x_1, y_2 + x_2, \ldots, y_{n+r} + x_{n+r})$.

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 two disjoint MPs simultaneously.

The maintenance cost is calculated in terms of the amount of capacity that each edge/node needs to be restored. The total cost to restore the edges/nodes in a CCN from the state X is

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

where $c_i(W_i - x_i)$ is the maintenance cost for e_i on any MP to recover from the current capacity x_i to its highest capacity W_i. For instance, given the current capacity vector $X = (1,0,1,1,0,0,1,1)$, the maximal capacity vector $\mathbf{W} = (3,3,3,1,2,4,5,4)$, and the unit maintenance cost $\mathbf{C} = (30,15,25,40,20,15,35,30)$. If x_1, x_3, x_4, x_7, and x_8 are on the MPs, the total maintenance cost to restore from state X is $TC(X) = c_1(W_1 - x_1) + c_3(W_3 - x_3) + c_4(W_4 - x_4) + c_7(W_7 - x_7) + c_8(W_8 - x_8) = 30(3–1) + 25(3–1) + 40(1–1) + 35(5–1) + 30(4–1) = 340$. 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,

$$\sum_{i:e_i \in \bigcup_{m=1}^{k} P_m} c_i(W_i - x_i) \leq \mathrm{B}. \tag{2}$$

The maximal capacity of P_m is $\min_{i:e_i \in P_m}(W_i)$, where $m = 1, 2, \ldots, k$. Similarly, under the capacity vector X, the capacity of P_m is $\min_{i:e_i \in P_m}(x_i)$. That is, $\min_{i:e_i \in P_m}(x_i)$ is the maximum number of data units which can be transmitted through P_m per unit of time. The transmission time to send d units of data through P_m under the capacity vector X, $\zeta(d,X,P_m)$, is

$$\text{lead time of } P_m + \left\lceil \frac{d}{\text{the capacity of } P_m} \right\rceil = \sum_{i:e_i \in P_m} l_i + \left\lceil \frac{d}{\min_{i:e_i \in P_m} x_i} \right\rceil. \tag{3}$$

If $\zeta(d,X,P_m) > \mathrm{T}$, it will contradict the time constraint. We have the following lemma showing the relationship between capacity vector and transmission time.

Lemma 1. *$\zeta(d,X,P_m) \geq \zeta(d,Y,P_m)$ for $m = 1, 2, \ldots, k$ if $X < Y$.*

Proof. If $X < Y$, then $x_i \leq y_i$ for each $e_i \in P_m$, and $\min_{i:e_i \in P_m} x_i \leq \min_{i:e_i \in P_m} y_i$. Thus, $\left\lceil \frac{d}{\min_{i:e_i \in P_m} x_i} \right\rceil \geq \left\lceil \frac{d}{\min_{i:e_i \in P_m} y_i} \right\rceil$. So $\zeta(d,X,P_m) \geq \zeta(d,Y,P_m)$.

£

For the demand pair (d_1,d_2) assigned to two MPs P_1 and P_2, the minimum transmission time $\Gamma(d_1,d_2,X)$ under X is

$$\Gamma(d_1,d_2,X) = \max\{\zeta(d_1,X,P_1), \zeta(d_2,X,P_2)\}. \tag{4}$$

That is, it spends at least $\Gamma(d_1,d_2,X)$ unit of time to transmit d_1 and d_2 under the condition without delay or stagnation. Thus, the minimum transmission time to send d units of data under X is $\Lambda(d,X) = \min_{\text{all } (d_1,d_2):d_1+d_2=d} \{\Gamma(d_1,d_2,X)\}$.

To estimate maintenance reliability, we calculate the interval estimation in terms of union of subsets. Different from the statistical inference, this interval certainly contains the maintenance reliability. Let Φ_T be the set of the capacity vectors X satisfying d and T, and let $\Phi_{\mathrm{UB}} = \{X \mid X \text{ is minimal in } \Phi_\mathrm{T}\}$. That is, Φ_{UB} is the set of the minimal capacity vectors satisfying d and T. Hence, we can obtain the following definition.

Definition 1: $X \in \Phi_{\mathrm{UB}}$ is called an UB-MP, equivalently, X is an UB-MP if and only if (i) $\Lambda(d,X) \leq \mathrm{T}$, and (ii) $\Lambda(d,X) > \mathrm{T}$ for any capacity vector Y with $Y < X$.

Lemma 2. If X is a UB-MP, then $Y \in \Phi_\mathrm{T}$ for any $Y > X$.

Proof. Since X is a UB-MP, we obtain $\Lambda(d,X) \le$ T. Lemma 1 says that $\zeta(d,Y,P_j) \le \zeta(d,X,P_j)$ for any $Y > X$. Hence, $\max\{\zeta(d_1,Y,P_1), \zeta(d_2,Y,P_2)\} \le \max\{\zeta(d_1,X,P_1), \zeta(d_2,X,P_2)\}$, equivalently, $\Gamma(d_1,d_2,Y) \le \Gamma(d_1,d_2,X)$. Then $\min_{\text{all }(d_1,d_2):d_1+d_2=d}\{\Gamma(d_1,d_2,Y)\} \le \min_{\text{all }(d_1,d_2):d_1+d_2=d}\{\Gamma(d_1,d_2,X)\}$. We conclude that $Y \in \Phi_{\text{T}}$ by obtaining $\Lambda(d,Y) \le \Lambda(d,X) \le$ T. £

Lemma 2 shows that $\Pr\{X \mid \Lambda(d,X) \le \text{T}\} = \Pr\{X \mid X \ge X_j$ for a UB-MP $X_j\}$. Suppose $X_1, X_2, \ldots, X_h$ are all UB-MPs and thus MR_{UB} can be represented as $MR_{\text{UB}} = \Pr\{X \mid X \in \Phi_{\text{B}}\} = \Pr\{\bigcup_{v=1}^{h} D_v\}$ where $D_v = \{X \mid X \ge X_v\}$, $v = 1, 2, \ldots, h$. Several methods such as RSDP algorithm (Zuo et al., 2007), inclusion-exclusion method (Hudson and Kapur, 1985; Lin, 2001, 2004, 2007, 2010; Xue, 1985), disjoint-event method (Hudson and Kapur, 1985; Yarlagadda and Hershey, 1991), and state-space decomposition (Alexopoulos, 1995; Aven, 1985; Jane et al., 1993), may be applied to compute $\Pr\{\bigcup_{v=1}^{h} D_v\}$. The inclusion-exclusion method easily leads to overload in memory when the network size is large. The RSDP algorithm has a better computational efficiency than the state-space decomposition for a large network (Zuo et al., 2007). Hence, the RSDP algorithm is applied to derive maintenance reliability herein.

However, the value MR_{UB} would be an overestimated-solution since we do not check the maintenance budget in this case and UB-MPs may include some capacity vectors that exceed B. Thus, MR_{UB} is an upper bound of maintenance reliability MR_{EX}, where MR_{EX} is defined as $\Pr\{X \mid \Lambda(d,X) \le \text{T}$ and $TC(X) \le \text{B}\}$. Let Φ_{B} be the set of X fulfilling d, B, and T while Φ_{EX} be the set of the minimal capacity vectors fulfilling d, B, and T. Therefore, $\Phi_{\text{EX}} = \{X \mid X$ is minimal in $\Phi_{\text{B}}\}$ and we have the following definition.

Definition 2: $X \in \Phi_{\text{EX}}$ is called an EX-MP, equivalently, X is an EX-MP if and only if (i) $\Lambda(d,X) \le$ T, (ii) $TC(X) \le$ B, and (iii) $\Lambda(d,Y) >$ T or $TC(Y) >$ B for any capacity vector Y with $Y < X$.

Lemma 3. If X is an EX-MP, then $Y \in \Phi_{\text{T}}$ for any $Y > X$.

Proof. (i) Since X is an EX-MP, we obtain $l(d,X) \le$ T and $TC(X) \le$ B. Lemma 1 says that $\zeta(d,Y,P_j) \le \zeta(d,X,P_j)$ for any $Y > X$. $\max\{\zeta(d_1,Y,P_1), \zeta(d_2,Y,P_2)\} \le \max\{\zeta(d_1,X,P_1), \zeta(d_2,X,P_2)\}$, equivalently, $\Gamma(d_1,d_2,Y) \le \Gamma(d_1,d_2,X)$. Then $\min_{\text{all }(d_1,d_2):d_1+d_2=d}\{\Gamma d_1,d_2,Y)\} \le \min_{\text{all }(d_1,d_2):d_1+d_2=d}\{\Gamma(d_1,d_2,X)\}$. We conclude that $Y \in \Phi_{\text{T}}$ by obtaining $\Lambda(d,Y) \le \Lambda(d,X) \le$ T.

(ii) Since $Y > X$, it implies that $y_i \ge x_i$ and $(W_i - y_i) \le (W_i - x_i)$ for each e_i. Then, $\sum_{e_i \in P_j} c_i(W_i - y_i) \le \sum_{e_i \in P_j} c_i(W_i - x_i)$. We thus complete the proof by obtaining $TC(Y) \le TC(X) \le$ B. £

Lemma 3 shows that $\Pr\{X \mid \Lambda(d,X) \le \text{T}$ and $TC(X) \le \text{B}\} = \Pr\{X \mid X \ge X_j$ for an EX-MP $X_j\}$. Intuitively, we may remove the unqualified X_j whose maintenance budged exceeds B from Φ_{UB} to generate EX-MP. Nevertheless, deleting the X_j fulfilling time T but exceeding the maintenance budget B means that we also delete the set $D_v = \{X \mid X \ge X_v\}$ and results in that some $X \ge X_v$ fulfilling T and B are removed. For instance, given two UB-MPs where $X_1 = (1,1,0,0)$ and $X_2 = (0,0,2,1)$ satisfying T. Assume that $\mathbf{W} = (2,3,2,1)$ and $\mathbf{C} = (5,3,2,6)$, we can calculate the total maintenance cost $TC(X_1) = 21$ and $TC(X_2) = 19$ by equation (1). If the maintenance budget is 20, we will delete X_1 whose maintenance cost is over the budget, and also $D_1 = \{X \mid X \ge X_1\}$. In fact, there exists other capacity vectors that are larger than X_1 and fulfill the budget, such as $X_3 = (2,1,0,0)$ with $TC(X_3) = 16$ and $X_4 = (1,2,0,0)$ with $TC(X_4) = 18$, where X_3 and X_4 may be EX-MPs. Besides, neither X_3 nor X_4 are included in the set $D_2 = \{X \mid X \ge X_2\}$. However, it is complicated to list all EX-MPs so that we have the following statement to find a lower bound of maintenance reliability.

Definition 3: Each capacity vector $X \in \Phi_{\text{LB}}$ is called a LB-MP, where $\Phi_{\text{LB}} = \Phi_{\text{UB}} \setminus \{X_j \mid TC(X_j) > \text{B}, X_j \in \Phi_{\text{UB}}\}$. Equivalently, X is a LB-MP if and only if (i) $X \in \Phi_{\text{UB}}$, and (ii) $TC(X) \le$ B.

Definition 3 implies that Φ_{LB} is a subset of Φ_{UB}, where each X satisfies the maintenance budget B. However, some X satisfying maintenance budget may be neglected if we delete the unqualified X_j (i.e., $TC(X_j) >$ B) and thus $D_v = \{X \mid X \ge X_v\}$ is also removed. Thus, the value $MR_{\text{LB}} \equiv \Pr\{X \mid X \ge X_j$ for a LB-MP $X_j\}$ is a lower bound of maintenance reliability. The interval $(MR_{\text{LB}}, MR_{\text{UB}})$ certainly contain MR_{EX}.

3. THE PROPOSED ALGORITHM

3.1 The algorithm to generate UB-MPs and LB-MP

All UB-MPs and LB-MPs can be generated by the following steps.

Step 0. [Initialization] Set $\Phi_{\text{UB}} = \varnothing$, $\Phi_{\text{LB}} = \varnothing$, and $j = 0$.

Step 1. Find the largest assigned demands $\overline{d_1}$ and $\overline{d_2}$ such that $\sum_{i:e_i\in P_1} l_i + \left\lceil \frac{d}{\min_{i:e_i\in P_1} x_i} \right\rceil \le$ T and $\sum_{i:e_i\in P_2} l_i + \left\lceil \frac{d}{\min_{i:e_i\in P_2} x_i} \right\rceil \le$ T, respectively.

Step 2. [Generation of feasible demand pairs] Generate all non-negative integer solutions of $d_1 + d_2 = d$ where $d_1 \le \overline{d_1}$ and $d_2 \le \overline{d_2}$.

Step 3. [Generation of UB-MPs] For each demand pair (d_1,d_2), do the following steps.

3.1 Find the minimal capacity v_1 (resp. v_2) of P_1 (resp. P_2) such that d_1 (resp. d_2) units of data can be sent through P_1 (resp. P_2) under T. That is, find the smallest integer v_1 and v_2 such that

$$\sum_{i:e_i\in P_1} l_i + \left\lceil \frac{d_1}{v_1} \right\rceil \le \text{T and } \sum_{i:e_i\in P_2} l_i + \left\lceil \frac{d_2}{v_2} \right\rceil. \tag{5}$$

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

$$x_i = \begin{cases} \text{minimal capacity } u \text{ of } e_i \text{ such that either} \\ \qquad u \ge v_1 \text{ if } e_i \in P_1 \text{ or } u \ge v_2 \text{ if } e_i \in P_2 \\ 0 \qquad\qquad\qquad\qquad \text{if others} \end{cases} \tag{6}$$

3.3 For $w = 1$ to $j - 1$, if $X_j \ge X_w$, then go to step 3.5; if $X_j < X_w$, then $\Phi_{UB} = \Phi_{UB} \backslash X_w$.
3.4 $\Phi_{UB} = \Phi_{UB} \cup \{X_j\}$.
3.5 Next (d_1,d_2).

Step 4. [Generation of LB-MPs] Set $\Phi_{LB} = \Phi_{UB}$. For each $X_j \in \Phi_{LB}$, 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 $TC(X_j) >$ B, $\Phi_{LB} = \Phi_{LB} \backslash X_j$.
4.3 Next $X_j \in \Phi_{LB}$.

Step 5. Two sets, Φ_{UB} and Φ_{LB} are generated.

Lemma 4. The set of UB-MPs is the set of X generated from the proposed algorithm.

Proof. We first claim that every obtained X_j from the algorithm is a UB -MP. Suppose X_j is not an UB-MP, then there exists an UB-MP $Y = (y_1, y_2, \ldots, y_n)$ such that $Y < X_j$. Without loss of generality, we assume an arc $e_u \in P_1$ such that $y_u < x_u$. It is known that x_u is the minimal capacity of e_u such that $x_u \ge v_1$. The situation $y_u < x_u$ results in that $y_u < v_1$ and $\sum_{i:e_i\in P_1} l_i + \left\lceil \frac{d_1}{v_1} \right\rceil >$ T. It contradicts that Y is an UB -MP. Hence, X_j is an UB-MP.

Conversely, we claim that every UB-MP is generated from the algorithm. Let X be a UB -MP. Suppose $\{X_1, X_2, \ldots, X_w\}$ is the set of X generated from the algorithm, and $X \notin \{X_1, X_2, \ldots, X_w\}$. Without loss of generality, there exists an arc $e_i \notin P_1 \cup P_2$ such that $x_i > 0$. Set $Y = (x_1, x_2, \ldots, x_i - z, \ldots, x_n)$, where $(x_i - z)$ is the maximal capacity of e_i such that $(x_i - z) < x_i$. Then $\zeta(d,Y,P_1) \le$ T and $\zeta(d,Y,P_2) \le$ T. That contradicts that X is an UB-MP. Hence, any UB-MP belongs to $\{X_1, X_2, \ldots, X_w\}$. We conclude that {UB-MPs} is the set of X generated from the algorithm. £

3.2 The RSDP algorithm

The RSDP algorithm is a recursive algorithm combined by the sum of disjoint product principle (Zuo et al., 2007). In this algorithm, a maximum operator, "$\oplus$", is defined as

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

For example, suppose that two UB-MPs, $X_1 = (1, 0, 1, 1, 0, 0, 1, 1)$ and $X_2 = (0, 0, 3, 0, 0, 0, 3, 3)$. By equation (7), $X_{1,2} = X_1 \oplus X_2 = (\max(1, 0), \max(0, 0), \max(1, 3), \max(1, 0), \max(0, 0), \max(0, 0), \max(1, 3), \max(1, 3)) = (1, 0, 1, 1, 0, 0, 3, 3)$. The RSDP algorithm is presented as the following pseudo codes.

IJOR Vol. 7, No. 1, 53−60 (2010)

RSDP algorithm //Compute maintenance reliability $R = \Pr\{\bigcup_{v=1}^{h} D_v\}$

```
function NR = RSDP(X1, X2, …, Xh) //Input h capacity vectors
for i = 1 : h
    if i == 1
        NR = Pr(X ≥ Xi);
    else
        TempNR_1 = Pr(X ≥ Xi);
        if i == 2
            TempNR_2 = Pr(X ≥ max(X1, Xi)); //max(X1, Xi) = (X1 ⊕ Xi)
        else
            for j = 1 : i – 1
                Xj = max(Xj, Xi); //max(Xj, Xi)= (Xj, Xi)
            end
            h = h – 1
            TempNR_2 = RSDP(X1, X2, …, Xh); //Execute recursive procedure
        end
    end
  NR = NR + TempNR_1 – TempNR_2; //Return NR
```

4. AN ILLUSTRATIVE EXAMPLE

We use a benchmark CCN (Chen and Hung, 1993, 1994; Chen and Chin, 1990) with 8 edges and 3 failure nodes shown in Figure 1 to illustrate the solution process. In this example, each edge is combined with several OC-18 (Optical Carrier 18) lines and each line provides two capacities, 1Gbps (giga bits per second) and 0 bps. Since the lines are provided by different suppliers, the edge's capacity has different probability distribution. The capacity, lead time, and per unit maintenance cost of each edge are shown in Table 1.

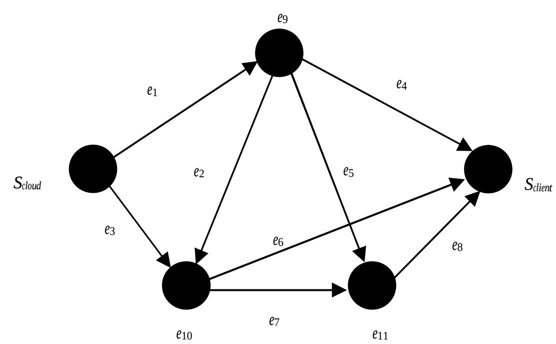

Figure 1 A benchmark CCN.

Table 1. The edge/nodea data of Figure 2.

edge	cost	lead time (sec)	capacity (Gbps) 0	1	2	3	4	5
e_1	30	2	0.102503	0.349562	0.397366	0.150569	0.000000 [b]	0.000000
e_2	15	1	0.002744	0.050568	0.310632	0.636056	0.000000	0.000000
e_3	25	3	0.000001	0.000297	0.029403	0.970299	0.000000	0.000000
e_4	40	3	0.468000	0.532000	0.000000	0.000000	0.000000	0.000000
e_5	20	1	0.014400	0.211200	0.774400	0.000000	0.000000	0.000000
e_6	15	2	0.018340	0.125985	0.324550	0.371586	0.159540	0.000000
e_7	35	2	0.000054	0.001652	0.020295	0.124667	0.382906	0.470427
e_8	30	1	0.000207	0.006083	0.066908	0.327107	0.599695	0.000000
e_9	40	2	0.219024	0.497952	0.283024	0.000000	0.000000	0.000000
e_{10}	25	1	0.000001	0.000297	0.029403	0.970299	0.000000	0.000000
e_{11}	25	1	0.000001	0.000297	0.029403	0.970299	0.000000	0.000000

[a] e_1 to e_8 for edges; e_9 to e_{11} for nodes.
[b] The edge does not provide this capacity.

In this example, the cloud have to send 5 Gbps data to the client through $P_1 = \{e_1, e_9, e_4\}$ and $P_2 = \{e_3, e_{10}, e_7, e_{11}, e_8\}$ simultaneously within 10 seconds and under maintenance budget 450. It implies that the CCN is falling to the failed state when the capacity level is less than 5 Gbps. The estimated and exact maintenance reliabilities are derived as follows.

Step 0. Set $\Phi_{UB} = \varnothing$, $\Phi_{LB} = \varnothing$, and $j = 0$.

Step 1. The largest demand $\bar{d}_1$ such that $(l_1 + l_9 + l_4) + \left\lceil \dfrac{\bar{d}_1}{\min\{W_1, W_9, W_4\}} \right\rceil \le 10$ is $\bar{d}_1 = 3$. The largest demand $\bar{d}_2$ such that $(l_3 + l_{10} + l_7 + l_{11} + l_8) + \left\lceil \dfrac{\bar{d}_2}{\min\{W_3, W_{10}, W_7, W_{11}, W_8\}} \right\rceil \le 10$ is $\bar{d}_2 = 6$.

Step 2. Generate all non-negative integer solutions of $d_1 + d_2 = 5$ where $d_1 \le \bar{d}_1$ and $d_2 \le \bar{d}_2$. The feasible (d_1,d_2) are (3,2), (2,3), (1,4), and (0,5).

Step 3. For $(d_1,d_2) = (3,2)$, do the following steps.

3.1 The lead time of P_1 is $l_1 + l_9 + l_4 = 7$. Then $v_1 = 1$ is the smallest integer such that $(7+\left\lceil \dfrac{3}{v_1} \right\rceil) \le 10$. Similarly, the lead time of P_2 is $l_3 + l_{10} + l_7 + l_{11} + l_8 = 8$. Then $v_2 = 1$ is the smallest integer such that $(8+\left\lceil \dfrac{2}{v_2} \right\rceil) \le 10$.

3.2 $X_1 = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_9, x_{10}, x_{11}) = (1,0,1,1,0,0,1,1,1,1,1)$.

3.4 $\Phi_{UB} = \Phi_{UB} \cup \{X_1\} = \{X_1\}$.

3.5 Next (d_1,d_2).

$\vdots$

3.4 $\Phi_{UB} = \{X_1, X_4\}$.The results are shown in Table 2.

Step 4. Set $\Phi_{LB} = \Phi_{UB} = \{X_1, X_4\}$ and do the following steps.

4.1 For X_1, $TC(X_1) = 30(3–1) + 25(3–1) + 40(1–1) + 35(5–1) + 30(4–1) + 40(2–1) + 25(3–1) + 25(3–1) = 480$.

4.2 Since $TC(X_1) = 480 > B = 450$, $\Phi_{LB} = \Phi_{LB} \backslash X_1 = \{X_4\}$.

4.1a For X_4, $TC(X_4) = 30(3–0) + 25(3–3) + 40(1–0) + 35(5–3) + 30(4–3) + 40(2–0) + 25(3–3) + 25(3–3) = 310$.

4.2a $TC(X_4) = 310 \le B = 450$, we do not remove X_4 from Φ_{LB}.

$\vdots$

The results concluded in Table 2 show that $\Phi_{LB} = \{X_4\}$.

Step 5. $\Phi_{UB} = \{X_1, X_4\}$ and $\Phi_{LB} = \{X_4\}$.

After executing the proposed algorithm, the results summarized in Table 2 show that $\Phi_{UB} = \{X_1, X_4\}$, and $\Phi_{LB} = \{X_4\}$. By the RSDP algorithm, we subsequently obtain the interval (MR_{UB}, MR_{LB}) = (0.891942791659912, 0.828023320538282) which contains MR_{EX}.

Table 2. Results of steps 3 and 4 in example.

(d_1,d_2)	(v_1,v_2)	X	$X_j \in \Phi_{UB}$ or not	Total Cost	$X_j \in \Phi_{LB}$ or not	Remark
(3,2)	(1,1)	X_1 = (1,0,1,1,0,0,1,1,1,1,1)	Yes	480	No	exceed budget
(2,3)	(1,2)	X_2 = (1,0,2,1,0,0,2,2,1,2,2)	No	-	-	$X_2 > X_1$
(1,4)	(1,2)	X_3 = (1,0,2,1,0,0,2,2,1,2,2)	No	-	-	$X_3 > X_1$
(0,5)	(0,3)	X_4 = (0,0,3,0,0,0,3,3,0,3,3)	Yes	310	Yes	-

5. COMPUTATIONAL COMPLEXITY

Computational complexity of the proposed algorithm in section 3.1 is analyzed as follows. In step 1, it takes at most $O(n + r)$ time to find the largest assigned demands $\bar{d}_1$ and $\bar{d}_2$. In step 2, there are at most $(d + 1)$ solution of $d_1 + d_2 = d$. For each (d_1,d_2), it takes at most $O(n + r)$ time to test time constraint (steps 3.1) and transform to X (step 3.2). The set Φ_{UB} contains at most $(d + 1)$ elements. Hence, each X_j needs $O(d(n + r))$ time to compare with other X in the worst case, and step 3 needs $O(d^2(n + r))$ time to all UB-MPs. Step 4 subsequently spends $O(n + r)$ time to check the budget constraint and to obtain LB-MPs in the worst case. In sum, it takes at most $O(d^2(n + r))$ time to execute the proposed algorithm. Hence, the computational time is linear with the number of edges and nodes, and is linear with the square of number of demand.

6. SUMMARY

When a CCN falls to the failed state where it cannot provide enough capacity to satisfy client's requirements, the maintenance action should be taken on each edge/node for keeping a good QoS. Moreover, the transmission time that data sent from the cloud to the client is also an important issue to be concerned. With nodes failure, we construct a network model to describe the flows and capacities in terms of minimal paths. We treat the maintenance reliability as a performance index and thus a bounding approach is developed to derive the estimated maintenance reliability. In particular, the LB-MPs are obtained from UB-MPs easily and efficiently by checking a maintenance budget constraint. The lower bound MR_{LB} can also be determined from the steps of deriving MR_{UB} since $\Phi_{UB} \supseteq \Phi_{LB}$. Thus, it is unnecessary to take additional steps for computing MR_{LB} but getting it in part of the steps of evaluating MR_{UB}. Based on the maintenance reliability, the system supervisors can conduct the sensitive analysis to improve/investigate the most important part in a large CCN.

REFERENCES

1. Aggarwal, K.K., Gupta, J.S. and Misra, K.B. (1975). A simple method for reliability evaluation of a communication system. *IEEE Transactions on Communications*, 23, 563-566.
2. Alexopoulos, C. (1995). A note on state-space decomposition methods for analyzing stochastic flow networks. *IEEE Transactions on Reliability*, 44, 354-357.
3. Aven, T. (1985). Reliability evaluation of multistate systems with multistate components. *IEEE Transactions on Reliability*, 34, 473-479.
4. Bodin, L.D., Golden, B.L., Assad, A. A. and Ball, M.O. (1982). Routing and scheduling of vehicles and crews: the state of the art. *Computers and Operations Research*, 10, 63-211.
5. Chen, G.H. and Hung, Y.C. (1993). On the quickest path problem. *Information Processing Letters*, 46, 125-128.
6. Chen, G.H. and Hung, Y.C. (1994). Algorithms for the constrained quickest path problem and the enumeration of quickest paths. *Computers and Operations Research*, 21, 113-118.
7. Chen Y.L. and Chin, Y.H. (1990). The quickest path problem. *Computers and Operations Research*, 17, 153-161.
8. Chen, Y.L. and Tang, K. (1998). Minimum time paths in a network with mixed time constraints. *Computers and Operations Research*, 25, 793-805.
9. Clímaco, J.C.N., Pascoal, M.M.B., Craveirinha, J.M.F. and Captivo, M.E.V. (2007). Internet packet routing: application of a k-quickest path algorithm. *European Journal of Operational Research*, 181, 1045-1054.
10. Golden, B.L. and Magnanti, T.L. (1977). Deterministic network optimization: a bibliography. *Networks*, 7, 149-183.
11. Hewitt, C. (2008). ORGs for scalable, robust, privacy-friendly client cloud computing. *IEEE Internet Computing*, 12, 96-99.
12. Hudson, J.C. and Kapur, K.C. (1985). Reliability bounds for multistate systems with multistate components. *Operations Research*, 33, 153-160.
13. Hung, Y.C. and Chen, G.H. (1992). Distributed algorithms for the quickest path problem. *Parallel Computing*, 18, 823-834.
14. Jane, C.C., Lin, J.S. and Yuan, J. (1993). On reliability evaluation of a limited-flow network in terms of minimal cutsets. *IEEE Transactions on Reliability*, 42, 354-361.
15. Lee, D.T. and Papadopoulou, E. (1993). The all-pairs quickest path problem. *Information Processing Letters*, 45, 261-267.
16. Lin, Y.K., (2001). A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers and Operations Research*, 28, 1277-1285.
17. Lin, Y.K. (2004). Reliability of a stochastic-flow network with unreliable branches & nodes under budget constraints. *IEEE Transactions on Reliability*, 53, 381-387.
18. Lin, Y.K. (2007). Reliability evaluation for an information network with node failure under cost constraint. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 37, 180-188.
19. Lin, Y.K. (2010). System reliability of a stochastic-flow network through two minimal paths under time threshold. *International Journal of Production Economics*, 124, 382-387.
20. Martins, E.D.Q.V. and Santos, J.L.E.D. (1997). An algorithm for the quickest path problem. *Operations Research Letters*, 20, 195-198.
21. Pascoal, M.M.B., Captivo, M.E.V. and Cl´ımaco, J.C.N. (2005). An algorithm for ranking quickest simple paths. *Computers and Operations Research*, 32, 509–520.
22. Xue, J. (1985). On multistate system analysis. *IEEE Transactions on Reliability*, 34, 329-337.
23. Yarlagadda, R. and Hershey, J. (1991). Fast algorithm for computing the reliability of communication network. *International Journal of Electronics*, 70, 549-564.
24. Yeh, W.C., (2004). Multistate network reliability evaluation under the maintenance cost constraint. *International Journal of Production Economics*, 88, 73-83.
25. Zuo, M.J., Tian, Z. and Huang, H.Z. (2007). An efficient method for reliability evaluation of multistate networks given all minimal path vector. *IIE Transactions*, 39, 811-817.