Revisit Consensus with Dual Fallible Communication in Cloud Computing

K.Q. Yan, S.C. Wang, S.S. Wang, C.P. Huang

Abstract—**Today, the network bandwidth increased and hardware devices have continuously to enhanced, resulting the vigorous development of the internet. Nowadays, cloud computing is using the low-power hosts to achieve high reliability that will be to ensure the ability to be better. However, the high capability most with high fault-tolerant by distributed systems, consensus problems will have an impact on the system node to complete the task of it. In previous works, consensus as a variety of connected networks is discussed with fallible components and is proposed a protocol to solve the consensus problem. In this paper, the consensus problem in topology of cloud computing is revisited. The proposed protocol, Dual Consensus Protocol for Cloud Computing (DCPCC), is supported to solve the consensus with malicious and dormant fault with transmission media. The DCPCC protocol can reach consensus with minimal rounds of message exchange and tolerate the maximal number of allowable components.**

Index Terms—**Consensus, Cloud Computing, Distributed System, Fault Tolerant.**

I. Introduction

In a distributed computing system, nodes allocated to different places or in separate units are connected together so that they may collectively be used to greater advantage. Each node in the system exchanges information with one another. Therefore, a task in a distributed system must achieve agreement. Notable examples of such tasks include the two-phase commitment in a distributed database system [7], the whereabouts of a replicated file in a distributed environment [8], and the task of landing a plane controlled by a flight path system [12].

Today, the network bandwidth and hardware devices have continuously to enhancing, then to generate the vigorous and the rapid development of internet applications made under the more diversification. However, the new concept of cloud computing has appeared now [6,8,11]. It has greatly encouraged distributed system design and practice to support user-oriented services [11]. However, there are many applications of cloud computing are to bring for the convenience of users, such as Google G-mail.

Manuscript received December 6, 2008.
K.Q. Yan is with the Chaoyang University of Technology, Taiwan, ROC (e-mail: kqyan@cyut.edu.tw).
S.C. Wang is with the Chaoyang University of Technology, Taiwan, ROC (corresponding author to provide phone: 886-4-2332-3000; fax: 886-2374-2337; e-mail: scwang@cyut.edu.tw).
S.S. Wang is with the Chaoyang University of Technology, Taiwan, ROC (e-mail: sswang@cyut.edu.tw)
C.P. Huang is with the Chaoyang University of Technology, Taiwan, ROC (e-mail: s9714625@cyut.edu.tw).

In cloud computing, the low-power hosts are used to ensure the high reliability [6]. However, the high capability most with high fault-tolerant by distributed systems, consistency problems will have an impact on the system node to complete the task of it. For users, the system needs to provide better reliability and fluency. In this paper, a topology of cloud computing is adapted to use [4]. Moreover, the consensus problem on this topology with failure of component will be discussed.

The symptom of a faulty communication media is classified into two types [3,9,10], dormant fault and malicious fault. The dormant fault of a fallible component is a crash or stuck-at fault. The symptom of a faulty component is usually unrestrained, and is commonly called malicious fault [7]. A malicious fault is unpredictable, and the behaviors of the other failure types can be treated as a special case of malicious fault [9]. In this paper, the consensus with malicious and dormant faulty transmission media in cloud computing will be solved. The proposed protocol DCPCC can make all correct nodes to reach consensus with the minimal number of message exchanges and tolerant the maximal number of allowable fallible transmission media.

The remainder of this paper has organized as follows. Section 2 discusses the topology of cloud computing. The concept of DCPCC is shown in Section 3. An example is given in Section 4. Section 5 concludes this paper.

II. Related Work

In this section, the topology of cloud computing and the related results of consensus are discussed.

A. Topology of Cloud Computing

Fig. 1 is a topology of cloud computing with two-lever groups is using in our research. The characteristics of the network topology are shown in follow:

1) The user request of service needs is received by nodes in the A-level group. Therefore, the capability of A-level group's node is better than the B-level group's node. In addition, the nodes of A-level group can communicate with each other in the same group directly.
2) The application service is provided by nodes in the B-level group's node. Hence, there are many nodes in B-level group. According to the property of nodes, the nodes are clustered in to cluster B_i where $1 \leq i \leq c_n$ and c_n is the total number of clusters in B-level group.
3) For the reliable communication, the redundant transmission media are used to connect between A-level group and B-level group [1].

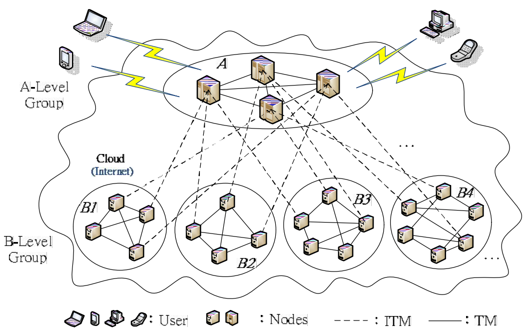

Fig. 1. The network topology of cloud computing

B. *Consensus Problem*

Cloud computing is a new computing concept of distributed system [6,11], that the nodes are interconnected with the Internet; the network is assumed reliable and synchronous. Achieving consensus on a same value in a distributed system, even if certain components in distributed system were failed, the protocols are required so that systems still can be executed correctly [5].

The unanimous problem the first studied by Lamport [2] and that is called the Byzantine Agreement (BA). A closely related sub-problem, the consensus problem, has been extensively studied [3,5,7,9,10] as well. The protocols are defined with solutions, which hope to use the minimum number of rounds of message exchange to achieve the maximum number of allowable faulty capability then to achieve a consensus [12]. In this paper, we have to be concerned with the solution of consensus problem. The definition of the problem is to make the correct nodes in an n nodes distributed system of cloud computing to reach consensus [7]. Every node chooses an initial value to start with, and communicates to each other by exchanging messages. All nodes are referred to make a consensus if they satisfy the following conditions [2]:

(Agreement): All correct nodes agree on a common value.

(Validity): If the source node is correct, then all correct nodes shall agree on the initial value of source node.

In a consensus problem, many cases are based on the assumption of node failure in a fail-safe network [5]. From this viewpoint, a transmission medium fault is treated as a node fault, regardless of the validity of an innocent node; hence, an innocent node does not involve a consensus. This assumption is unreasonable, for it violates the definition of consensus that all correct nodes should agree on a common value [5]. Therefore, it is important to propose a protocol to solve consensus problem in a topology of cloud computing with fallible transmission media. In this study, the consensus problem will be solved with dual transmission media failure mode (both dormant fault and malicious fault).

III. Dual Consensus Protocol for Cloud Computing

In this study, the proposed protocol, Dual Consensus Protocol for Cloud Computing (DCPCC), is invoked to solve the consensus problem due to faulty transmission media in cloud computing including three part, the *group agreement process*, *inter agreement process* and *consensus agreement process*.

The mainly work of the *group agreement process* is collecting the user request from A-level group's nodes to decide an initial value of each node. Subsequently, the A-level group's node forwards its value to B-level group's nodes in *inter agreement process*. In the *inter agreement process*, each node in B-level group's cluster will received many copies of initial value by using redundant transmission media between A-level group and B-level group that is sent from A-level group's node and to determine the initial value of the cluster. The *consensus agreement process* is collecting the service request from the nodes of B-level group's cluster to reach a consensus. The proposed protocol DCPCC is presented in Fig. 2.

DCPCC protocol

Group agreement process

- Each node of A-level group calls procedure *message-gathering*(*A-level group*) to obtain the consensus value DEC_{Ai} of A-level group.

Inter agreement process

- The node in A-level group broadcasts the DEC_{Ai} to B-level group by using redundant transmission media between A-level group and B-level group.
- Node in B-level group's cluster receives DEC_{Ai} and broadcasts to other nodes in the same cluster.
- Each node in the same cluster takes the majority value of the received values as the initial value of each one.

Consensus agreement process

- Each node of B-level group's cluster calls procedure *message-gathering*(*B-level group*) to obtain the consensus value DEC_{Bi} of B-level group's cluster.

Procedure *message-gathering* (i-th node of X-level group with initial value v_i)

Message Exchange Phase:

Round 1:

Node i broadcasts v_i, then receives the initial value from the other nodes in the same cluster, and construct vector V_i. If a dormant fault was found, it will be set to λ by the receiver standing for a dormant fault.

Round 2:

Node i broadcasts V_i, then receives column vectors broadcasted by other nodes, and construct MAT_{Xi}.

Decision Making Phase:

Step 1: Each λ value is eliminated and does not join to majority.

Step 2: Take the majority value of each column k of MAT_{Xi} to MAJ_k.

Step 3: Search for row k of MAJ_k. If ($\exists MAJ_k = \neg v_i$), then $DEC_{Xi} := \phi$; else if ($\exists MAJ_k = \Omega$) AND ($v_{ki} = v_i$), then $DEC_{Xi} := \phi$; else $DEC_{Xi} := v_i$, and terminate.

Procedure *MAT* (i-th node of X-level group with initial value v_i)

Step 1: Receive the initial value v_j from node j, for $1 \le j \le n$ and $j \ne i$.

Step 2: Construct the vector $V_i = [v_1, v_2, \ldots, v_n]$, $1 \le j \le n$ and $j \ne$ i.

Step 3: Broadcast V_i to all nodes, and receive column vector V_j from node j, $1 \le j \le n$.

Step 4: Construct a MAT_{Xi} (Setting the vector v_j in column j, for $1 \le j \le n$).

Fig. 2. The DCPCC protocol to reach consensus

The *group agreement process* has message exchange phase and decision making phase. The message exchange phase needs to collect enough messages from A-level group's nodes. In second phase of *group agreement process*, the decision making phase, each correct A-level group's node i computes a common value DEC_{Ai} by applying the majority voting function to messages, collected by message exchange phase to reach an agreement.

In the *inter agreement process*, the node in A-level group broadcasts the DEC_{Ai} to B-level group by using redundant transmission media between A-level group and B-level group. The node in B-level group's cluster receives DEC_{Ai} in the first round; and broadcasts the vector to other nodes in the same cluster with second round of message exchange phase. Each node in the same cluster takes the majority value of the received values in the decision making phase then the initial value of each one is obtained.

There are message exchange phase and decision making phase in the *consensus agreement process* too. In the first round of message exchange phase, each node in the same cluster of B-level group broadcasts the initial value obtained from *inter agreement process* to other nodes and receives the other node's initial values in the same cluster. And, in the second round of message exchange phase, node *i* broadcasts the received values in the first round to other nodes and receives the other node's values in the same cluster to construct a MAT_{Bi}. In the decision making phase, a majority value DEC_{Bi} of MAT_{Bi} is taken. Moreover, for simplicity, if a dormant transmission medium was found in message exchange phase, it will be set to λ by the receiver standing for a dormant fault. Afterward, in the decision making phase, eachλ will be eliminated first since it represents a dormant faulty. Finally, the consensus of each correct node is reached.

IV. Example of Executing DCPCC

Subsequently, an example of executing the DCPCC protocol based on the cloud computing is shown in Fig. 3 and illustrated as follows.

The example with cluster of A as A-level group is illustrated in Fig.4. In the first round of message exchange in *group agreement process*, each node *i* multicasts its initial value v_i to all other nodes in the A-level group, and receives the initial value of other nodes as well, as shown in Fig. 5(a). Then, each node uses the received message to construct vector V_i as shown in Fig. 5(b). In the second round of message exchange in *group agreement process*, each node multicasts its vector V_i and receives the vectors from other nodes to construct the matrix MAT_{Ai}. Finally, the decision making phase takes the majority value of MAT_{Ai} to construct the matrix MAJ_i, as shown in Fig. 5(c), and achieves the common value DEC_{Ai} (= 1) of A-level group.

In the *inter agreement process,* an example of cluster B_2 is shown in Fig. 6. The node in A-level group received the DEC_{Ai} (= 1) then broadcasts this value to B-level group by using redundant transmission media between A-level group. The values received of cluster B_2 are sent from nodes of A-level group are shown in Fig. 7(a). Then, the nodes have the DEC_{Ai}, broadcasts the received value DEC_{Ai} to each node in the same cluster as shown in Fig. 7(b). Finally, each node takes the majority value of the received values as the initial value of each one, as shown in Fig. 7(c).

The example of cluster B_4 is shown in Fig. 8. In the *inter agreement process*, each node *i* of B-level group's cluster by using the initial value obtained then broadcasts its initial value v_i to all other nodes in the same cluster. And, in the first round of message exchange in *consensus agreement process*, and receives the initial value of other nodes in the same cluster as well, as shown in Fig. 9(a). Then, each node uses the received message to construct vector V_i as shown in Fig. 9(b). In the second round of message exchange in *consensus agreement process*, each node multicasts its vector V_i and receives the vectors from other nodes to construct the matrix MAT_{Bi}. Finally, the decision making phase, each λ value is eliminated and does not join to majority. Then, each node takes the majority value of MAT_{Bi} to construct the matrix MAJ_{Bi}, as shown in Fig. 9(c), and achieves the common value DEC_{Bi} (= 1) of B-level group's node.

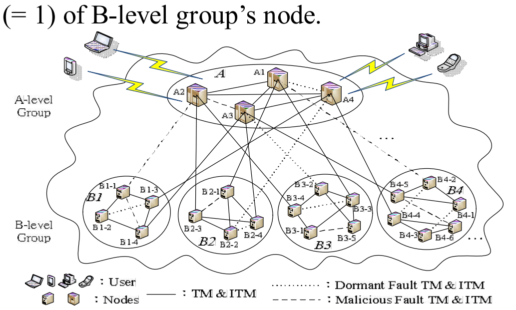

Fig. 3. An example of cloud computing environment

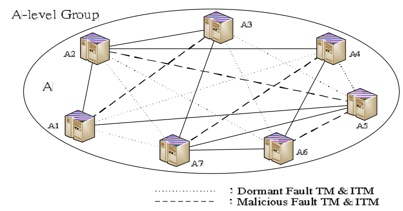

Fig. 4. Example of A-level Group as cluster of A

A_1	A_2	A_3	A_4	A_5	A_6	A_7
1	1	1	1	1	1	1

Fig. 5(a). Initial value of each node

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_1	1	1	***1***	λ	1	1	λ
A_2	1	1	1	1	***0***	λ	λ
A_3	***0***	1	1	1	λ	λ	1
A_4	λ	1	1	1	λ	1	***0***
A_5	1	***0***	λ	λ	1	***1***	1
A_6	1	λ	λ	1	***0***	1	1
A_7	λ	λ	1	***0***	1	1	1

Fig. 5(b). The vector received in first round

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
1	1	***1***	*λ*	1	1	*λ*	
1	1	***0***	*λ*	0	λ	*λ*	
0	1	***0***	*λ*	λ	λ	*λ*	
λ	1	***1***	*λ*	λ	1	*λ*	DEC_{A1}= 1
1	0	***0***	*λ*	1	1	*λ*	
1	λ	*λ*	*λ*	0	1	*λ*	
λ	λ	***1***	*λ*	1	1	*λ*	
1	1	θ	λ	1	1	λ	
MAJ_{A1} of MAT_{A1}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
1	1	1	λ	***0***	*λ*	*λ*	
1	1	1	1	***0***	*λ*	*λ*	
0	1	1	1	***1***	*λ*	*λ*	
λ	1	1	1	*λ*	*λ*	*λ*	DEC_{A2}= 1
1	0	λ	λ	***0***	*λ*	*λ*	
1	λ	λ	1	***0***	*λ*	*λ*	
λ	λ	1	0	***0***	*λ*	*λ*	
1	1	1	1	0	λ	λ	
MAJ_{A2} of MAT_{A2}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
0	1	1	λ	λ	λ	λ	DEC_{A3}= 1
0	1	1	1	λ	λ	λ	
0	1	1	1	λ	λ	1	
λ	1	1	1	λ	λ	0	
0	0	λ	λ	λ	λ	1	
1	λ	λ	1	λ	λ	1	
λ	λ	1	0	λ	λ	1	
0	1	1	1	λ	λ	1	
MAJ_{A3} of MAT_{A3}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
λ	1	1	λ	λ	1	λ	DEC_{A4}= 1
λ	1	1	1	λ	λ	0	
λ	1	1	1	λ	λ	1	
λ	1	1	1	λ	1	0	
λ	0	λ	λ	λ	1	1	
λ	λ	λ	1	λ	1	0	
λ	λ	1	0	λ	1	1	
λ	1	1	1	λ	1	θ	
MAJ_{A4} of MAT_{A4}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
1	1	λ	λ	1	1	λ	DEC_{A5}= 1
1	0	λ	λ	0	λ	λ	
0	1	λ	λ	λ	0	1	
λ	1	λ	λ	λ	0	0	
1	0	λ	λ	1	1	1	
1	λ	λ	λ	0	0	1	
λ	0	λ	λ	1	0	1	
1	θ	λ	λ	1	0	1	
MAJ_{A5} of MAT_{A5}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
1	λ	λ	λ	1	1	λ	DEC_{A6}= 1
1	λ	λ	1	0	λ	λ	
0	λ	λ	1	λ	λ	1	
λ	λ	λ	1	0	1	0	
1	λ	λ	λ	1	1	1	
1	λ	λ	1	1	1	1	
λ	λ	λ	0	1	1	1	
1	λ	λ	1	1	1	1	
MAJ_{A6} of MAT_{A6}							

A_1	A_2	A_3	A_4	A_5	A_6	A_7	
λ	λ	1	λ	1	1	λ	DEC_{A7}= 1
λ	λ	1	0	0	λ	λ	
λ	λ	1	0	λ	λ	1	
λ	λ	1	1	λ	1	0	
λ	λ	λ	λ	1	1	1	
λ	λ	λ	1	0	1	1	
λ	λ	1	0	1	1	1	
λ	λ	1	0	1	1	1	
MAJ_{A7} of MAT_{A7}							

Fig. 5(c). Construct MAT in second round and MAJ of MAT as decision value with node in A-level Group

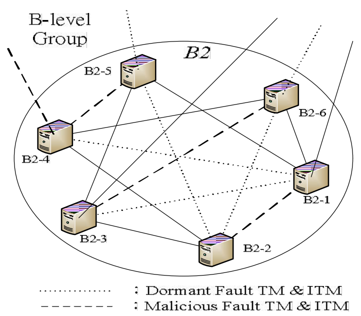

Fig. 6. Example of cluster B_2 in *inter agreement process*

B_{2-1}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}
1	1	0	λ	λ

Fig. 7(a). Received value of B-level group's cluster B_2 from nodes of A-level group

	B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}
B_{2-1}	1	0	λ	λ	1	1
B_{2-2}	1	1	1	1	1	1
B_{2-3}	λ	1	1	1	1	0
B_{2-4}	λ	0	0	0	0	0
B_{2-5}	λ	λ	λ	0	λ	λ
B_{2-6}	λ	λ	0	λ	λ	λ

Fig. 7(b). Received vectors of cluster B_2's nodes in first round

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
1	0	λ	λ	1	1	DEC_{B2-1}= 1
1	1	λ	λ	1	1	
λ	0	λ	λ	1	0	
λ	0	λ	λ	0	0	
λ	λ	λ	λ	λ	λ	
λ	0	λ	λ	λ	λ	
1	0	λ	λ	1	θ	
MAJ_{B2-1} of MAT_{B2-1}						

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
0	0	λ	λ	λ	λ	DEC_{B2-2}= 0
1	1	1	1	λ	λ	
0	1	1	1	λ	λ	
λ	0	0	0	λ	λ	
λ	λ	λ	0	λ	λ	
0	λ	0	λ	λ	λ	
0	θ	θ	θ	λ	λ	
MAJ_{B2-2} of MAT_{B2-2}						

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
λ	0	λ	λ	1	0	DEC_{B2-3}= θ
λ	1	1	1	1	1	
λ	1	1	1	1	0	
λ	0	0	0	0	0	
λ	λ	λ	0	λ	0	
λ	λ	0	λ	λ	λ	
λ	θ	θ	θ	1	0	
MAJ_{B2-3} of MAT_{B2-3}						

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
λ	0	λ	λ	0	1	DEC_{B2-4}= 0
λ	1	1	1	1	1	
λ	1	1	1	0	0	
λ	0	0	0	0	0	
λ	λ	λ	0	λ	λ	
λ	λ	0	λ	0	λ	
λ	θ	θ	θ	0	θ	
MAJ_{B2-4} of MAT_{B2-4}						

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
1	λ	λ	0	1	1	DEC_{B2-5}= 1
1	λ	1	1	1	1	
λ	λ	1	0	1	0	
λ	λ	0	0	0	0	
λ	λ	λ	0	λ	λ	
λ	λ	0	0	λ	λ	
1	λ	θ	0	1	θ	
MAJ_{B2-5} of MAT_{B2-5}						

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	B_{2-5}	B_{2-6}	
1	λ	0	λ	1	1	DEC_{B2-6}= 1
1	λ	0	1	1	1	
λ	λ	1	1	1	0	
λ	λ	0	0	0	0	
λ	λ	0	0	λ	λ	
λ	λ	0	λ	λ	λ	
1	λ	0	θ	1	θ	
MAJ_{B2-6} of MAT_{B2-6}						

Fig. 7(c). Construct MAT_{B2} in second round and MAJ_{B2} of MAT_{B2} as decision value

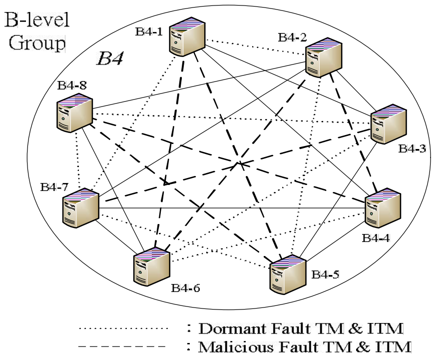

Fig. 8. Example of B-level Group as cluster of B_4

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	1	1	1	1	1	1	1

Fig. 9(a). The initial value obtained in the *inter agreement process* of cluster B_4's node

	B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
B_{4-1}	1	λ	1	1	0	1	λ	1
B_{4-2}	λ	1	1	0	λ	0	1	1
B_{4-3}	1	1	1	1	1	λ	0	λ
B_{4-4}	1	0	1	1	1	λ	1	1
B_{4-5}	1	λ	1	1	1	1	λ	0
B_{4-6}	0	0	λ	λ	1	1	1	1
B_{4-7}	λ	1	0	1	λ	1	1	λ
B_{4-8}	1	1	λ	0	0	1	λ	1

Fig. 9(b). Received vectors of cluster B_4's nodes in first round

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	1	1	0	1	λ	1
λ	λ	1	0	λ	1	λ	1
1	λ	1	1	0	λ	λ	λ
1	λ	1	1	1	λ	λ	1
1	λ	1	1	0	0	λ	0
0	λ	λ	λ	1	1	λ	1
λ	λ	0	1	λ	1	λ	λ
1	λ	λ	0	0	1	λ	1
1	λ	1	1	0	1	λ	1
MAJ_{B4-1} of MAT_{B4-1} => DEC_{B4-1}= 1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
λ	λ	1	1	λ	1	λ	1
λ	1	1	1	λ	0	1	1
λ	1	1	1	λ	0	0	λ
λ	0	1	1	λ	1	1	1
λ	λ	1	1	λ	0	λ	0
λ	0	λ	0	λ	1	1	1
λ	1	0	1	λ	0	1	λ
λ	1	λ	0	λ	1	λ	1
λ	1	1	1	λ	θ	1	1
MAJ_{B4-2} of MAT_{B4-2} => DEC_{B4-2}= 1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	1	1	0	λ	λ	λ
λ	1	1	0	λ	λ	1	λ
1	1	1	1	1	λ	0	λ
1	0	1	1	1	λ	1	λ
1	λ	1	1	1	λ	λ	λ
0	0	λ	λ	1	λ	0	λ
λ	1	0	1	λ	λ	0	λ
1	1	λ	0	0	λ	0	λ
1	1	1	1	1	λ	0	λ
MAJ_{4-3} of MAT_{4-3} => DEC_{4-3}=1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	1	1	0	λ	λ	1
λ	1	1	0	λ	λ	1	0
1	1	1	1	1	λ	0	λ
1	0	1	1	1	λ	1	1
1	λ	1	1	1	λ	λ	0
0	0	λ	λ	1	λ	1	0
λ	0	0	1	λ	λ	1	0
1	1	λ	0	0	λ	λ	1
1	θ	1	1	1	λ	1	0
MAJ_{B4-4} of MAT_{B4-4} => DEC_{B4-4}=1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	1	1	0	1	λ	1
0	λ	1	0	λ	0	λ	1
1	λ	1	1	1	λ	λ	1
0	λ	1	1	1	λ	λ	1
0	λ	1	1	1	1	λ	0
0	λ	λ	λ	1	1	λ	0
0	λ	0	1	λ	1	λ	λ
1	λ	λ	0	0	1	λ	1
0	λ	θ	1	1	1	λ	1
MAJ_{B4-5} of MAT_{B4-5} => DEC_{B4-5}=1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	λ	λ	0	1	λ	1
0	0	λ	λ	λ	0	1	1
0	0	λ	λ	1	λ	0	λ
0	0	λ	λ	1	λ	1	1
1	0	λ	λ	1	1	λ	0
0	0	λ	λ	1	1	1	1
λ	1	λ	λ	λ	1	1	λ
1	1	λ	λ	0	1	λ	1
0	0	λ	λ	1	1	1	1
MAJ_{B4-6} of MAT_{B4-6} => DEC_{B4-6}= 1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
λ	λ	0	1	λ	1	λ	λ
λ	1	1	0	λ	0	1	λ
λ	1	0	1	λ	λ	0	λ
λ	0	1	1	λ	λ	1	λ
λ	λ	1	1	λ	1	λ	λ
λ	0	0	λ	λ	1	1	λ
λ	1	0	1	λ	1	1	λ
λ	1	λ	0	λ	1	λ	λ
λ	1	0	1	λ	1	1	λ
MAJ_{B4-7} of MAT_{B4-7} => DEC_{B4-7}= 1							

B_{4-1}	B_{4-2}	B_{4-3}	B_{4-4}	B_{4-5}	B_{4-6}	B_{4-7}	B_{4-8}
1	λ	λ	1	0	1	λ	1
λ	1	λ	0	0	0	λ	1
1	1	λ	1	1	λ	λ	λ
1	0	λ	0	0	λ	λ	1
1	λ	λ	0	1	1	λ	0
0	0	λ	0	0	1	λ	1
λ	1	λ	1	λ	1	λ	λ
1	1	λ	0	0	1	λ	1
1	1	λ	0	0	1	λ	1
MAJ_{B4-8} of MAT_{B4-8} => DEC_{B4-8}= 1							

Fig. 9(c). Construct MAT_{B4} in second round and MAJ_{B4} of MAT_{B4} as decision value

V. Fault Tolerance Capability Analysis

According to literatures [12], we may obtain a protocol which can tolerate the transmission media faults in a system provided that $\lceil c/2 \rceil$-1 faulty transmission media where c is the connectivity of network [12]. However, the results are not appropriate for the cloud computing environment. We can drop a fault tolerance capability for cloud computing environment of topology as follows.

To cope with the network topology of cloud computing, the notations and parameters of this network topology are shown as follows:

Notation	Discussion
TM_{ij}:	The transmission media between node *i* and node *j*.
ITM:	The transmission media between A-level and B-level group.
n:	The number of nodes in topology with cluster.
c:	The connectivity of network topology.
fd_{TM}:	The number of faulty transmission media with the dormant fault in each cluster.
fm_{TM}:	The number of faulty transmission media with the malicious fault in each cluster.
fd_{ITM}:	The number of faulty transmission media with the dormant fault between A-level and B-level group.
fm_{ITM}:	The number of faulty transmission media with the malicious fault between A-level and B-level group.
tf_{TM}:	The number as tolerate faults of transmission media.

1) A correct receiving node can detect the dormant faulty sending node by DCPCC as *group agreement process* and *consensus agreement process*. In protocol with *group agreement process* and *consensus agreement process*, value of λ is to declare the dormant fault of topology.

$$c \le n-1,\ fd_{TM} \le \lfloor c/2 \rfloor \quad \therefore\ c \ge 2fd_{TM}$$

2) In *group agreement process* and *consensus agreement process*, each node can through decision making phase of DCPCC to remove the influence of malicious faulty transmission media. Each node collects enough value then after to execute decision making phase, the influence of malicious faulty transmission media can be removed.

$$c \le n-1,\ fm_{TM} \le \lfloor c/3 \rfloor \quad \therefore\ c \ge 3fm_{TM}+1$$

3) In *inter agreement process*, the correct node in B-level group's cluster collects the value sent from A-level nodes and takes the majority value then the initial value of each node can be obtained.

$$fd_{ITM} \le \lfloor ITM/2 \rfloor,\ fm_{ITM} \le \lfloor ITM/3 \rfloor \quad \therefore\ ITM \ge 2fd_{ITM}+fm_{ITM}$$

4) The maximum number of allowable dual faulty transmission media by DCPCC is:

$$tf_{TM} \ge (2fd_{TM}+fm_{TM})+2fd_{ITM}+fm_{ITM}$$

$$\Rightarrow fm_{TM} \le \lfloor c/3 \rfloor,\ fm_{ITM} \le \lfloor c/3 \rfloor$$

In *group agreement process* and *consensus agreement process*, $c \ge 3TM_{ij}+1$ is gotten; and in *inter agreement process*, $c \ge 3ITM+1$ is obtained, hence, $c \ge 3tf_{TM}+1$ is gained.

VI. Conclusion

According to previous studies [3,5,7,9,10], the network topology plays an important role in the consensus problem. The consensus problem with the fault-tolerance capacity is a fundamental problem in the distributed environment [5,7]. The problem has been studied by various kinds of network model in the past [7]. In this paper, DCPCC is proposed to make all correct nodes reaching consensus. However, the consensus problem on dual transmission media failures in topology of cloud computing is revisited. The fault-tolerance capacity is enhanced by DCPCC.

Acknowledgment

This work was supported in part by the Taiwan National Science Council under Grants NSC96-2221-E-324-021 and NSC97-2221-E-324-007-MY3.

Reference

[1] F. Halsall, Data Links, Computer Networks and Open Systems. 4th ed., Addison-Wesley Publishers, 1995, pp. 112-125.
[2] L. Lamport et al., "The Byzantine General Problem," ACM Transactions on Programming Language and Systems, Vol. 4, No. 3, Jul. 1982, pp. 382-401.
[3] F.J. Meyer and D.K. Pardhan, "Consensus with Dual Failure Modes," IEEE Transactions on Parallel and Distributed System, Vol. 2, No. 2, 1991, pp. 214-222.
[4] G. Pei, M. Gerla, X. Hong, and C.C. Chiang, "A Wireless Hierarchical Routing Protocol with Group Mobility," IEEE Wireless Communications and Networking Conference, Sep. 1999, pp. 1538-1524.
[5] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in Presence of Faults," Journal of ACM, Vol. 27, No. 2, Apr. 1980, pp. 228-234.
[6] M.A. Vouk, "Cloud Computing- Issues, Research and Implementations," Information Technology Interfaces, Jun. 2008, pp. 31-40.
[7] S.C. Wang Y.H. Chin, and K.Q. Yan, "Achieving Byzantine Agreement in a Generalized Network Model," IEEE Computer Society Press, Proceedings of CompEuro 89, Vol. 4, 1989, pp. 139-145.
[8] L.H. Wang, J. Tao, and M. Kunze, "Scientific Cloud Computing: Early Definition and Experience," the 10th IEEE International Conference on High Performance Computing and Communications, 2008, pp. 825-830.
[9] S.C. Wang and K.Q Yan, "Revisit Consensus Problem on Dual Link Failure Modes," Proceedings of the International Computer Software & Applications Conference, Aug. 1921, 1998, Vienna, Austria, pp. 8489.
[10] S.C. Wang, K.Q. Yan, S.S. Wang, and G.Y. Zheng, "Reaching Agreement Among Virtual Subnets in Hybrid Failure Mode," IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 9, Sep. 2008, pp. 1252-1262.
[11] A. Weiss, "Computing in The Clouds," netWorker, Vol. 11, No. 4, 2007, pp. 16-25.
[12] K.Q. Yan and Y.H. Chin, "An Optimal Solution for Consensus Problem in an Unreliable Communication System," International Conference on Parallel Processing, 1998, pp. 388-391.