The Study of Consensus with Fallible Communication of Cloud Computing

K.Q. Yan[1], S.S. Wang[2], S.C. Wang[*3], C.P. Huang[4]

Chaoyang University of Technology
168, Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan (R.O.C.)

[1]kqyan@cyut.edu.tw

[2]sswang@cyut.edu.tw

[*3]scwang@cyut.edu.tw (*: Corresponding Author)

[4]s9714625@cyut.edu.tw

***Abstract*— The reliability of the distributed system has been an important topic of research. Consensus protocols, which allow the correct nodes to agree on a common value, have been brought up to aid the reliable execution of tasks. In previous works, fully connected networks, generalized connected networks or multicasting network with fallible components were proposed to solve the consensus problem. However, the topology of cloud computing is not a certainty structure. In this study, consensus problem is reexamined in a topology of cloud computing. The purposed protocol FLINK (Fallible LINK) can make all correct nodes reaching consensus with minimal number of message exchanges and tolerant the maximal number of allowable fallible components.**

***Keywords*—Consensus, Cloud Computing, Distributed System, Fault Tolerant**

1. INTRODUCTION

Today, the network bandwidth increased and hardware devices have continuously to enhanced, resulting the vigorous development of the internet. However, the rapid development of internet applications made under the more diversification, and the cloud computing of new concept has appeared now [5,10,8]. It has greatly encouraged distributed system design and practice to support user-oriented services. Many of today the internet applications are to bring for the convenience of users, such as Google G-mail [4].

Nowadays, cloud computing is using the low-power hosts to achieve high reliability that will be to ensure the ability to be better. However, it is now well recognized that the consensus problem is a fundamental problem when implementing fault-tolerant distributed services. In many applications, a correct node in a distributed system should be able to reach a consensus even if certain components in the distributed system fail [2,4]. The previous protocols of consensus [2,4] allow nodes to reach a consensus and work correctly in fully connected network, broadcast network, generalized connected network and multicast network [6]. However, since all the networks show a regularized network structure, these protocols may not work for the network of different structure and connections. Nowadays, the application service system needs to provide better reliability and fluency. The previous application is not with a specific by cloud computing to order the operation of internet of topology [3], therefore, in this paper a topology of cloud computing is adapted to use.

The symptom of a faulty component is usually unrestrained, and is commonly called malicious fault [6]. A malicious fault is unpredictable, and the behaviors of the other failure types can be treated as special cases of a malicious fault [9]. However, if the malicious fault, which is the thorniest fault, can be solved, then the other fault types can surely be solved [6]. Therefore, in this paper, the consensus problem on the topology of cloud computing is revisited. The purposed protocol FLINK can make all correct nodes reaching consensus with minimal number of message exchanges and tolerant the maximal number of allowable fallible communication links.

The remainder of this paper is organizes as follows. Section 2 describes the topology of cloud computing. The concept of FLINK is shown in Section 3. An example is given in Section 4. In Section 5 the fault tolerant capability analysis of FLINK is given. Finally, the conclusion is given in the last section.

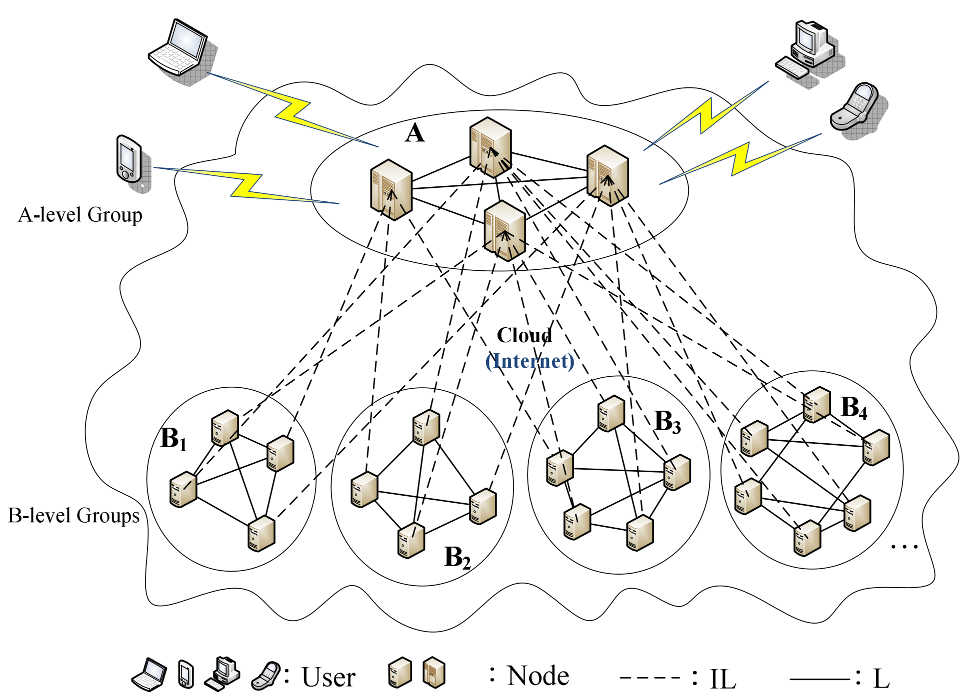
2. RELATED WORK

Recent advances of the internet applications to result distributed systems have to enhance reliability and stability that in order to provide the better quality of services. However, the cloud computing which the fault-tolerant capability of system is the very important topic. Hence, the topology of cloud computing will be introduced in subsection 2.1, the related of consensus problem results are in subsection 2.2.

2.1 Topology of Cloud Computing

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

1) The user request of service needs to fast processing and is received by nodes in the A-level group. Therefore, the capability of A-level group' node is better than the B-level group' node. In addition, the nodes of A-level group can communicate (by link L) with each other in the same group directly.
2) The application service is provided by nodes in the B-level group' 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 $1 \le i \le c_n$ and c_n is the total number of clusters in B-level group.
3) For the reliable communication, the communication links (ILs) between A-level group and B-level group is used to connect between A-level group and B-level group [1].



L: The communication link between nodes of each cluster
IL: The communication link between A-level group and B-level group

Fig. 1. The network topology of cloud computing

2.2 Consensus Problem

In distributed system, cloud computing is a new computing concept, that the nodes are interconnected with the Internet; the network is assumed reliable and synchronous [4]. 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.

The unanimous problem is called the Byzantine Agreement (BA) and is first studied by Lamport [2]. A closely related sub-problem, the consensus problem, has been extensively studied [2,4,6,7,9] as well. The solutions are defined as protocols, which achieve a consensus and hope to use the minimum number of rounds of message exchange to achieve the maximum number of allowable faulty capability. We concern the solution of consensus problem in this paper. The definition of the problem is to make the correct nodes in an n nodes distributed system of cloud computing to reach consensus. 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 the source node sends.

In a consensus problem, many cases are based on the assumption of node failure in a fail-safe network [4]. Base on this assumption, we treat a communication link fault as a node fault, whatever the correctness of an innocent node, so an innocent node does not involve consensus [4]. The definition of a consensus problem requires all correct nodes to reach a consensus [4].

In this paper, we consider a distributed system whose nodes are reliable during the consensus execution in a cloud computing application; while the communication links may be fault by interference from some noise or a hijacker and results in the exchanged message can exhibit arbitrary behavior. Therefore, a protocol to achieve consensus in an unreliable communication environment is proposed in this paper. When all nodes reach consensus in cloud computing topology, the fault-tolerance capacity has been enhanced.

3. CONSENSUS PROTOCOL

In this paper, the proposed protocol, called Fallible LINK (FLINK in short), is invoked to solve the consensus problem due to faulty communication links in cloud computing including three parts, the *group agreement process, inter message exchange process* and *consensus agreement process*.

The mainly work of the *group agreement process* is to collect 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 initial value to B-level group's nodes, and each B-level group's node can obtain the initial value of itself in the *inter message exchange process*. In the *consensus agreement process,* each node in B-level group's cluster collects the service request from the nodes of B-level group's cluster to reach a consensus, in the *consensus agreement process*. The proposed protocol FLINK is presented in Fig. 2.

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 message exchange process*, the node in A-level group broadcasts the DEC_{Ai} to B-level group by using IL. The node in B-level group's cluster receives DEC_{Ai} and the initial of the node 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. Finally, the consensus of each correct node is reached.

FLINK 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 message exchange process

- The nodes in A-level group broadcast the DEC_{Ai} to B-level group's nodes by using the communication links (IL) between A-level group and B-level group.
- Each node of B-level group's cluster receives the DEC_{Ai} as its initial value.

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.

Round 2:

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

Decision Making Phase:

Step 1: Take the majority value of each column k of MAT_{Xi} to MAJ_k.

Step 2: Search for any MAJ_k. If ($\exists MAJ_k = \neg v_i$), then $DEC_{Xi}:=\phi$; else if ($\exists MAJ_k.= \lambda$) 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 $1\leq j\leq n$ and $j\neq i$.

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

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

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

Fig. 2. The FLINK protocol to reach consensus

4. EXAMPLE OF EXECUTING FLINK

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

In the first round of message exchange in the *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. 4(a). Then, each node uses the received message to construct vector V_i as shown in Fig. 4(b). In the second round of message exchange in the *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. 4(c), and achieves the common value DEC_{Ai} (= 1) of A-level group.

In the *inter message exchange process*, the node in A-level group broadcasts the DEC_{Ai} (= 1) to B-level group by using ILink. Fig. 5 shows the received values of each B-level group's cluster from nodes of A-level group. Then, the nodes of B-level group's cluster have the DEC_{Ai} as the initial value of each one.

By using the initial value obtained in the *inter message exchange process*, each node i of B-level group's cluster broadcasts its initial value v_i to all other nodes in the same cluster, in the first round of message exchange in the *consensus agreement process*, and receives the initial value of other nodes in the same cluster as well. Then, each node uses the received message to construct vector V_i as shown in Fig. 6(a). In the second round of message exchange in the *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 takes the majority value of MAT_{Bi} to construct the matrix MAJ_{Bi}, as shown in Fig. 6(b)-6(e), and achieves the common value DEC_{Bi} (= 1) of B-level group's node.

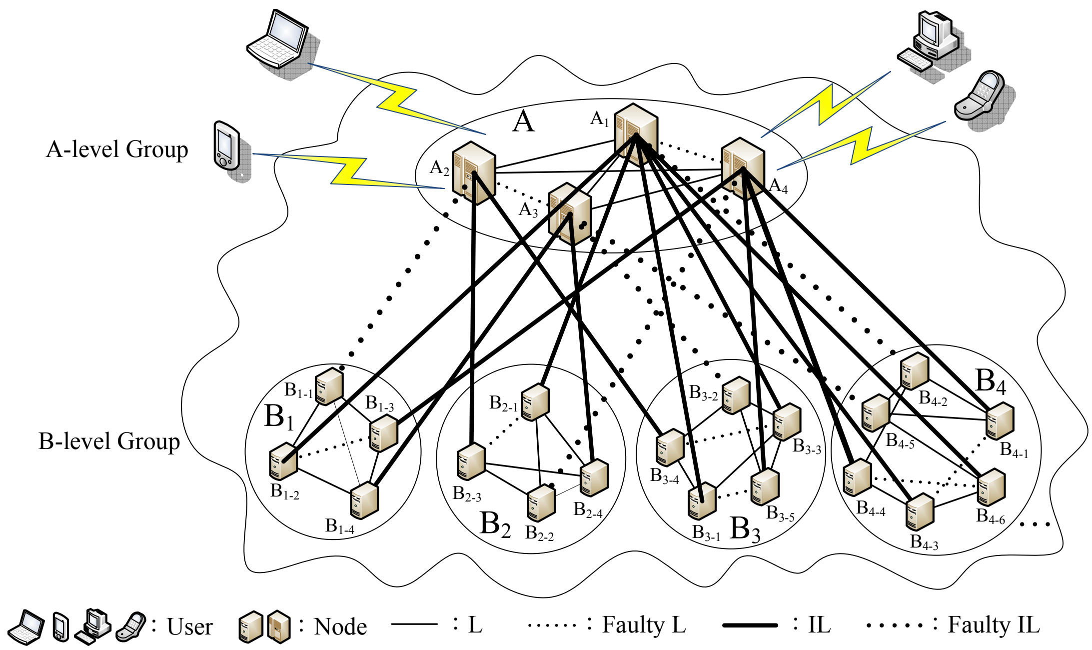


Fig. 3. An example of cloud computing environment

A_1	A_2	A_3	A_4
1	1	1	1

Fig. 4(a). Initial value of each node

	A_1	A_2	A_3	A_4
A_1	1	1	1	***0***
A_2	1	1	***0***	1
A_3	1	***1***	1	1
A_4	***0***	1	1	1

Fig. 4(b). The vector received in first round

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

A_1	A_2	A_3	A_4	
1	1	1	0	DEC_{A2}= 1
1	1	0	1	
0	***0***	***0***	***0***	
0	1	1	1	
1	1	θ	θ	
MAJ_{A2} of MAT_{A2}				

A_1	A_2	A_3	A_4	
1	1	1	0	DEC_{A3}=1
1	***0***	***0***	***0***	
1	1	1	1	
0	1	1	1	
1	1	1	θ	
MAJ_{A3} of MAT_{A3}				

A_1	A_2	A_3	A_4	
0	***1***	***0***	***0***	DEC_{A4}=1
1	1	0	1	
1	1	1	1	
0	1	1	1	
θ	1	θ	1	
MAJ_{A4} of MAT_{A4}				

Fig. 4(c) Construct MAT in second round and MAJ of MAT as decision value

B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}
0	1	1	1

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}
1	***0***	1	1

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}
1	***0***	1	1	1

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

Fig. 5. Received values of each B-level group's cluster from nodes of A-level group

	B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}
B_{1-1}	0	1	1	1
B_{1-2}	0	1	***0***	1
B_{1-3}	0	***0***	1	1
B_{1-4}	0	1	1	1

	B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}
B_{2-1}	1	0	***0***	1
B_{2-2}	1	0	1	1
B_{2-3}	***0***	0	1	1
B_{2-4}	1	0	1	1

	B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}
B_{3-1}	1	0	1	1	***0***
B_{3-2}	1	0	1	1	1
B_{3-3}	1	0	1	***0***	1
B_{3-4}	1	0	***0***	1	1
B_{3-5}	***0***	0	1	1	1

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

Fig. 6(a). The vector received in first round of each cluster's node

B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}	
0	1	1	1	DEC_{B1-1}=1
0	1	0	1	
0	0	1	1	
0	1	1	1	
0	1	1	1	
MAJ_{B1-1} of MAT_{B1-1}				

B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}	
0	1	1	1	DEC_{B1-2}=1
0	1	0	1	
0	***0***	***0***	***1***	
0	1	1	1	
0	1	θ	1	
MAJ_{B1-2} of MAT_{B1-2}				

B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}	
0	1	1	1	DEC $_{B1-3}$=1
0	***0***	***0***	***0***	
0	1	1	1	
0	1	1	1	
0	0	1	1	
MAJ_{B1-3} of MAT_{B1-3}				

B_{1-1}	B_{1-2}	B_{1-3}	B_{1-4}	
0	1	1	1	DEC $_{B1-4}$=1
0	1	0	1	
0	0	1	1	
0	1	1	1	
0	1	1	1	
MAJ_{B1-4} of MAT_{B1-4}				

Fig. 6(b) Construct MAT_{B1} in second round and MAJ_{B1} of MAT_{B1} as decision value

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

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	
1	0	0	1	DEC_{B2-2}=1
1	0	1	1	
0	0	1	1	
1	0	1	0	
1	0	1	1	
MAJ_{B2-2} of MAT_{B2-2}				

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

B_{2-1}	B_{2-2}	B_{2-3}	B_{2-4}	
1	0	0	1	DEC_{B2-4}=1
1	0	1	1	
0	0	1	1	
1	0	1	1	
1	0	1	1	
MAJ_{B2-4} of MAT_{B2-4}				

Fig. 6(c) Construct MAT_{B2} in second round and MAJ_{B2} of MAT_{B2} as decision value

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}	
1	0	1	1	0	DEC_{3-1}=1
1	0	1	1	1	
1	0	1	0	1	
1	0	0	1	1	
0	***1***	***0***	***0***	***0***	
1	0	1	1	1	
MAJ_{3-1} of MAT_{3-1}					

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}	
1	0	1	1	0	DEC_{3-2}=1
1	0	1	1	1	
1	0	1	0	1	
1	0	0	1	1	
0	0	1	1	1	
1	0	1	1	1	
MAJ_{3-2} of MAT_{3-2}					

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}	
1	0	1	1	0	DEC_{3-3}=1
1	0	1	1	1	
1	0	1	0	1	
0	***0***	***0***	***0***	***0***	
0	0	1	1	1	
1	0	1	1	1	
MAJ_{3-3} of MAT_{3-3}					

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}	
1	0	1	1	0	
1	0	1	1	1	
0	***1***	***0***	***0***	***0***	DEC_{3-4}=1
1	0	0	1	1	
0	0	1	1	1	
1	0	1	1	1	
MAJ_{3-4} of MAT_{3-4}					

B_{3-1}	B_{3-2}	B_{3-3}	B_{3-4}	B_{3-5}	
0	***0***	***0***	***1***	***0***	
1	0	1	1	1	
1	0	1	0	1	DEC_{3-5}=1
1	0	0	1	1	
0	0	1	1	1	
1	0	1	1	1	
MAJ_{3-5} of MAT_{3-5}					

Fig. 6(d) Construct MAT_{B3} in second round and MAJ_{B3} of MAT_{B3} as decision value

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

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

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

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

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

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

Fig. 6(e) Construct MAT_{B4} in second round and MAJ_{B4} of MAT_{B4} as decision value

5. THE FAULT TOLERANCE CAPABILITY

According to literatures [11], we may obtain a protocol which can tolerate the communication links in a system provided that $\lceil c/2 \rceil$-1 faulty communication links where c is the connectivity of network [11]. 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 showed as follows:

Notation	Discussion
L_{ij}:	The communication link among node i and node j of each cluster.
IL	The communication link between A-level group and B-level group.
Tf_L:	The total number of faulty communication links in the network topology of cloud computing.
f_L:	The number of faulty communication links in each cluster.
f_{IL}:	The number of faulty communication links between A-level group and B-level group.
c:	The connectivity of network topology.
tf_L:	The number as tolerate faults of communication links.

Case 1: The communication links among node *i* and node *j* of A-level and B-level group's cluster:

$$L_{ij} \ge \lfloor 2f_L \rfloor + 1 \quad \text{and} \quad c \ge 3f_L$$

Case 2: The communication links between A-level group and B-level group:

$$IL \ge \lfloor 2f_{IL} \rfloor + 1 \quad \text{and} \quad c \ge 3f_{IL}$$

Therefore,

$$Tf_L = \sum_{\min(f_L)} |(|L_{ij}|/2)+(|IL|/2)|$$

$$=(||\min(L_{ij})|/2|+1)+(||\min(IL)|/2|+1)$$

$$\therefore\ tf_L \le \lceil c/2 \rceil - 1$$

The best case of fault tolerant capability:

$$Tf_L = \sum_{\min(|L_{ij}|-f_L)} ||L_{ij}|/2| + 1 + \sum_{\min(|IL-f_{IL})} ||IL|/2| + 1$$

(2) The worst case fault tolerant capability:

$$TfL = \sum_{\max(f_L)} |L_{ij}| + \sum_{\max(f_{IL})} |IL|$$

6. Conclusion

The consensus problem is a fundamental problem in the distributed environment [2,4,7,9]. The problem has been studied by various kinds of network model in the past [6,7,9]. According to previous studies, the network topology plays an important role in this problem [7,11]. However, the cloud computing is a new concept of distributed system but the previous protocols cannot adapt to it. Therefore, in this paper, the consensus problem in cloud computing is revisited. The purposed protocol FLINK makes all correct nodes reaching consensus. FLINK derive its bound of allowable faulty communication links in Section 5 of this paper, with two rounds of message exchange.

Acknowledgment

This work was supported in part by the Taiwan National Science Council under Grants NSC95 -2221-E-324-019-MY3 and NSC96-2221-E-324 -021.

References

[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] 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.

[4] 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.

[5] M.A. Vouk, "Cloud Computing- Issues, Research and Implementations," *Information Technology Interfaces*, Jun. 2008, pp. 31-40.

[6] 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.

[7] 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.

[8] S.C. Wang, Y.H. Chin, K.Q. Yan, "Consensus Problem on a Generalized Network Connected by Unreliable Transmission," *International Conference on Computing and Information*, Canadian Scholars' Press, 1990, pp. 413-417.

[9] 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.

[10] A. Weiss, "Computing in The Clouds," *netWorker*, Vol. 11, No. 4, 2007, pp. 16-25.

[11] 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.