Optimal Malicious Agreement in a Virtual Subnet-based Cloud Computing Environment

Kuo-Qin Yan, Hsueh-Hsun Huang
Department of Business Administration
Chaoyang University of Technology
Taiwan, R.O.C.
{kqyan; s9937902}@cyut.edu.tw

Shu-Ching Wang*, Shun-Sheng Wang
Department of Information Management
Chaoyang University of Technology
Taiwan, R.O.C.
{scwang; sswang}@cyut.edu.tw)

***Abstract*—Fault-tolerance is an important research topic in the context of distributed systems. In a distributed system, the cooperative tasks must achieve agreement before performing certain tasks. Nowadays, there are a lot of application services on the cloud computing environment. However, mobile cloud computing is widely accepted as a concept which can significantly improve a user's experience when accessing mobile services. The *Byzantine agreement* (BA) problem is a fundamental problem in fault-tolerance with regard to distributed systems. In previous studies, the BA algorithm is designed using traditional network topology. However, these do not perform well in the context of mobile cloud computing. In order to increase the capability of faulty tolerance and ensure network security, it is necessary to provide a stable mobile cloud service environment. To enhance the reliability of a virtual subnet-based cloud computing environment, a new protocol known as an *optimal malicious agreement* (OMA) is proposed to solve the BA problem in this study. OMA uses the minimum number of message exchanges to make all correct nodes agree on a common value and can tolerate the maximum number of faulty components.**

***Keywords*-**Byzantine Agreement; Fault tolerant; Distributed system; Mobile cloud-computing; Virtual subnet

I. INTRODUCTION

Cloud computing has become a significant technology trend as many applications in the context cloud computing increase convenience for users [10]. Furthermore, the concept of mobile cloud computing inherently provides for the advantages of cloud computing available for users but will provide additional functionality to the cloud as well. Mobile cloud computing will help to overcome limitations of mobile devices, particularly with regard to processing power and data storage [3,5]. However, one of the fundamental mobile cloud computing issues is reliability, where the target mobile nodes connected to the mobile cloud service provider must listen to specific tasks from the server and application recovery is needed.

As mobile cloud computing has become increasingly popular, network topology has trended toward wireless connectivity. Thus, providing enhanced support for mobile cloud computing. In short, this technological trend has greatly encouraged distributed system design and support to mobile nodes. Virtual subnets have attracted significant attention recently because they require less infrastructure, can be deployed quickly, and can automatically adapt to changes in topology. Therefore, virtual subnets suit military communication systems, emergency disaster rescue operations and law enforcement [1]. These, in particular, have brought cloud-computing technology to the mobile cloud computing domain [3,5].

The reliability of the mobile node is one of the most important aspects with regard to the virtual subnet. In order to provide a reliable virtual subnet-based cloud computing environment, a mechanism to allow a set of mobile nodes to agree on a value is required. The *Byzantine agreement* (BA) problem is one of the most fundamental problems [2,9] with regard to reaching an agreement value in a distributed system. The original BA problem defined by Lamport *et al.* [4] is as follows:

(1) There are n nodes in a synchronous distributed system; where n is a constant and $n \geq 4$.
(2) All nodes can communicate with each other through a reliable fully connected network.
(3) One or more of the nodes might fail, so a faulty node may transmit incorrect message(s) to other nodes.
(4) After message exchange, all correct nodes should reach a common agreement, if and only if the number of faulty nodes t is less than one-third of the total number of nodes in the network, or $t \leq (n-1)/3$.

The solutions define a protocol which can reach agreement by using the minimal rounds of message exchange to obtain the maximum number of allowable faulty capability. The problem tackled in this paper involves helping the correct nodes to achieve agreement with underlying n-nodes in a virtual subnet-based cloud computing environment. The source node chooses an initial value to start with and communicates with others by exchanging messages. The nodes have reached an agreement if the scenario satisfies the following conditions [4]:

(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 which the source node sent.

In previous studies, the BA algorithm was designed for use in a traditional network topology [2]. However, these do not perform well in a virtual subnet-based cloud computing

 ISBN: 978-1-61208-280-6

environment. In order to increase the capability of faulty tolerance and ensure network security, it is necessary to provide a stable mobile cloud service environment. To enhance fault-tolerance, a new protocol known as *optimal malicious agreement* (OMA) in a virtual subnet-based cloud computing environment is proposed to solve the BA problem in this study. OMA uses the minimum number of message exchanges to allow all correct nodes to agree on a common value and can tolerate the maximum number of faulty components.

The rest of this paper is organized as follows. Section 2 discusses the topology of a virtual subnet-based cloud computing environment. Section 3 illustrates the concept of OMA. An example of the execution of the proposed protocol is given in section 4. Section 5 proves the correctness and complexity of our new protocols. Section 6 concludes this paper and offers direction for future research.

II. RELATEED WORKS

Nowadays, the virtual subnet is more practical as it provides the ability for nodes to join the network and leave anytime with no impact on the infrastructure. A group of multiple nodes in a virtual subnet is cooperating to achieve specific objectives; each node communicates with other nodes by broadcasting in the virtual subnet, but also leads to severe problems, such as broadcast storm [1]. Many researchers have proposed cluster schemes where broadcasting is limited and use a virtual subnet to improve upon problems related to broadcast storm and to reduce conflicts. However, recently, virtual subnets have been a more important topic than others [1]. The virtual subnet is composed of several groups through an overlapping network approach [1]. Figure 1 shows a topology of a virtual subnet-based cloud computing environment. There are three situations where the nodes communicate with the underlying topology.

Situation1. Nodes in the same group communicate with each other directly through a virtual backbone.

Situation2. Nodes in different groups exchange messages with each other via a virtual subnet or physical communication media (Internet IP based), e.g., host/agent communication.

Situation3. Host/agent node can communicate with cloud main service via physical communication media in the network topology.

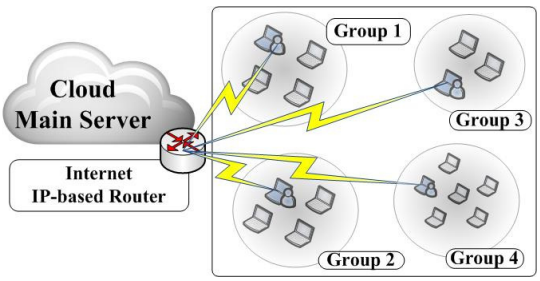

Figure 1. The topology of virtual subnet-based cloud computing environment

In addition, the virtual backbone can be used to: 1) collect topology information for routing; 2) provide a backup route; and 3) multicast or broadcast messages [7]. Hence, a virtual subnet can improve the broadcast storm scenario. In a BA problem, many cases were solved on the assumption of node failure in a fail-safe network [4]. The optimal algorithm for solving the BA problem requires the use of a minimal number of rounds to achieve agreement.

In this study, a new protocol is proposed to solve the BA problem where the communication media in a virtual subnet-based cloud computing environment are reliable but where the node may be faulty through interference from hijackers resulting in the exchanged message exhibiting arbitrary behavior. A protocol reaching agreement in a reliable communication environment involving a traditional fully-connected network was first proposed by Lamport *et al.* [4]. The typical protocol by Fischer [2] can tolerate $f \leq \lfloor (n-1)/3 \rfloor$ faulty nodes in malicious situations and requires σ ($\sigma = f+1$) round(s) to receive enough messages in order to achieve agreement.

However, most of the distributed computing systems may not be fully connected. The network topology has the feature of cluster or group similar to the topology of a virtual subnet. The proposed protocol OMA is used to solve the BA problem underlying a virtual subnet-based cloud computing environment in which the node may fail in a malicious way. When all nodes achieve agreement, the fault tolerance capability has been enhanced even if the communication media has failed between sensor nodes; here, the backbone of the system can be used to provide a backup route [1].

However, the virtual subnet-based cloud computing environment is different than the traditional network, so the previous protocols used in the context of the BA are not suited for the environment this paper proposes. As a result, the new protocol is proposed such that it can be used to solve the BA problem with a malicious fault type in a virtual subnet-based cloud computing environment.

III. THE PROPOSED PROTOCOL

The purpose of the BA protocol is to allow all correct nodes to reach a common agreement in a virtual subnet-based cloud computing environment. For this reason, nodes should exchange messages with all other nodes. Each correct node receives messages from other nodes during a number of rounds of message exchanges. Afterwards, all correct nodes have enough messages to make a decision value, called an agreement value or common value. Then, all correct nodes agree on the same value.

The assumptions, notations and parameters of the proposed protocol OMA are shown as follows:

- Each node in the network can be identified uniquely.
- A node does not know the fault status of other nodes.
- Let n be the total number of nodes in the network.
- Let g be the number of groups in the network and $g \geq 4$.
- Let x be group identifier where $1 \leq x \leq g$ and $g \geq 4$.
- Let n_x be the number of nodes in group Gp_x, $0 \leq x \leq g$. If there are more than $\lceil n_x/2 \rceil$ malicious faulty mobile nodes in Gp_x, then Gp_x will be named the malicious faulty group.

- Let c be the connectivity of the virtual subnet, where c is g–1.
- Let T_{FG} be the total number of malicious faulty groups.
- Let T_{Fn} be the total number of malicious faulty nodes.

In the BA protocol, the first step is to count the number of required rounds of message exchange, which is determined by the total number of nodes at the beginning of protocol execution. Therefore, if the variety of faulty nodes can be determined, then the number of rounds of message exchange can be reduced and then the fault tolerance capability is increased.

The proposed OMA can solve the BA problem due to faulty node(s), which may send incorrect messages to influence the system to reach agreement in a virtual subnet-based cloud computing environment. By using the proposed OMA protocol, all correct nodes in the environment can reach a common agreement which requires θ rounds of message exchange, where $\theta = \lfloor (g-1)/3 \rfloor + 1$.

The proposed OMA protocol is organized in two phases: 1) the message exchange phase and 2) the decision making phase. In the first round of the message exchange phase, the cloud main server sends its initial value to all groups and the receiver node stores the received value in the root of its mg-tree. The mg-tree is a tree structure which is used to store the received message in the message exchange phase from cloud main server [11]. After the first round of the message exchange phase (σ>1), each node transmits the value at level σ–1 in its own mg-tree to all other nodes. At the end of each round, the receiver node applies the function RMAJ() to the values received from the same group to obtain a single value. Moreover, each receiver node stores the received messages and the value of function RMAJ() in its mg-tree. RMAJ() is defined in Figure 2.

Subsequently, in the decision making phase, each node outside of the cloud main server reorganizes its mg-tree into a corresponding ic-tree. The ic-tree is a tree structure which is used to store a received message without repeated group names [11]. Therefore, the common value VOTE(s) was obtained by using the function VOTE() on the root s of each mobile node's ic-tree. The detailed steps of the proposed OMA protocol is presented in Figure 2.

IV. AN EXAMPLE OF OMA EXECUTED

An example is given to execute OMA and the virtual subnet-based cloud computing environment is described in Figure 3. There are 22 nodes falling into seven groups. Gp_1 includes P_1 and P_2. Gp_2 includes P_3, P_4, P_5 and P_6. Gp_3 includes P_7, P_8, P_9 and P_{10}. Gp_4 includes P_{11} and P_{12}. Gp_5 includes P_{13} and P_{14}. Gp_6 includes P_{15} and P_{16}. P_{17}, P_{18}, P_{19}, P_{20} and P_{21} belong to Gp_7,:

- The messages are sent from the cloud main server; then, execute OMA.
- The source node C_s (cloud main server) is a malicious faulty server.
- C_s sends 1 to all nodes of Gp_2, Gp_4, Gp_5, Gp_6 and Gp_7 and sends 0 to all nodes of Gp_1 and Gp_3.

In a BA problem with fallible nodes, the worst situation is such that the source node is no longer honest [2]. Put simply, this is the worst case scenario. Suppose the cloud main server is the source node (let it be C_s), which is a malicious fault; this means C_s may arbitrarily send different message values (e.g., replicate command [9]) to different groups. Therefore, in order to solve the BA problem among correct nodes within this example, OMA requires θ ($\lfloor (g-1)/3 \rfloor + 1$) rounds of message exchange. pre-execute counts of the number of rounds required before the message exchange phase in OMA. Then, three ($\lfloor (g-1)/3 \rfloor + 1 = \lfloor (7-1)/3 \rfloor + 1 = 3$) rounds of message exchange are required.

OMA (Source node with initial value v_s)

Definitions:

1. For the virtual subnet, each mobile node has common knowledge of the entire graphic information $\hat{G}$ = (E, Gp), where Gp is the set of groups in the network and E is a set of group pairs (Gp_x, Gp_y) indicating a physical communication medium (the sensing is covered) between group Gp_x and group Gp_y. [7]
2. Each mobile node communicates with all other mobile nodes via the virtual subnet, virtual backbone or physical communication media [1].
3. The node plays sender, receiver or agent, the behavior dictates which kind of transmission is sent[2].
4. The host agent node communicates with the cloud service via a physical communication media (Internet IP based).
5. The host agent node cannot garble the message between the sender node and receiver node; this has been achieved using encryption technology (such as RSA [6]).

Pre-Execute. Computes the number of rounds required $\theta = \lfloor (g-1)/3 \rfloor + 1$, where g is the total number of groups in the network.

Message Exchange Phase:

Case σ = *1, run*

1. The source node transmits its initial value v_s to each group's nodes.
2. Each receiver node obtains the value and stores it in the root of its mg-tree.

Case 1<σ≤ θ, *run*

1. Each node without the source node transmits the values at level σ–1 in its mg-tree to each group's nodes.
2. Each receiver node applies RMAJ on its received messages and stores RMAJ value in the corresponding vertices at level θ of its mg-tree.

Decision Making Phase:

Step 1. Reorganizing the mg-tree into a corresponding ic-tree. (The vertices with repeated group names are deleted).

Step 2. Using function VOTE on the root s of each node's ic-tree, the common value VOTE(s) will obtain.

Function RMAJ(V)

The majority value of the vector $V_i = [v_1, \ldots, v_{n-1}, v_n]$, if the majority exists.

Otherwise, choose a default value ϕ.

Function VOTE(μ)

If the μ is a leaf, then output the value μ.

If the majority value does not exist, then output the majority value ϕ.

Otherwise, output the majority m, where $m \in \{0, 1\}$

Figure 2. The proposed OMA protocol

The source node C_s transmits replication commands to all other nodes in the first round of the message exchange phase. The replication command obtained from each correct node is listed in Figure 4. In the σ-th (1<σ≤θ) round of message exchange, except for the Cs, each node transmits RMAJ() values at the (σ–1)-th level in its mg-tree to all other nodes and itself. Subsequently, each receiver node applies RMAJ() to its received messages and stores the received messages and RMAJ() values at the corresponding vertices at level σ of its mg-tree. The mg-tree of the correct node P_1 during the second and final round in the message exchange phase are shown in Figures 5 and 6.

 ISBN: 978-1-61208-280-6

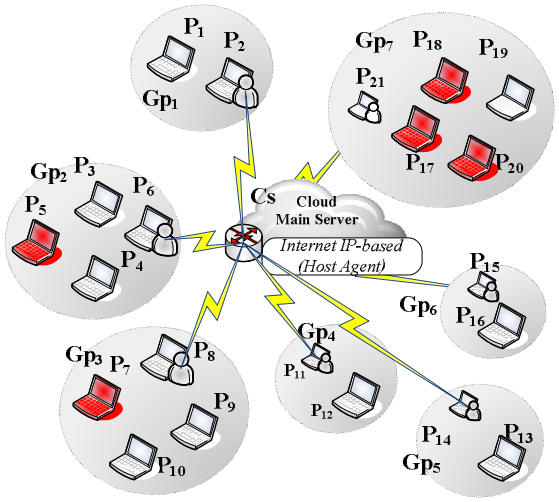

Figure 3. The initial status of executing OMA

After the message exchange phase, the tree structure of each correct node is converted from mg-tree to ic-tree by deleting the vertices with duplicated names. The example ic-tree is shown in Figure 7. Eventually, using the function VOTE to root the value *s* for each correct node's ic-tree {VOTE(*s*) = VOTE(*s1*), …, VOTE(*s7*) = 1}, an agreement value 1 can be obtained, as shown in Figure 8. Here, the decision making phase is complete.

	Level 1 *Root s*
Gp_1's correct nodes	0
Gp_2's correct nodes	1
Gp_3's correct nodes	0
Gp_4's correct nodes	1
Gp_5's correct nodes	1
Gp_6's correct nodes	1
Gp_7's correct nodes	1

Figure 4. The mg-tree of each node at first round

Level 1	*Level 2*	*Take*	*RMAJ*
Val(*s*)=1	*s1*	0	(0,0)
	s2	1	(1,1,***0***,1)
	s3	0	(0,***0***,0,0)
	s4	1	(1,1)
	s5	1	(1,1)
	s6	1	(1,1)
	s7	0	(***0***,***0***,1,***0***,1)

Figure 5. The mg-tree of correct P_1 at second round

V. CORRECTNESS AND COMPLEXITY

The following lemmas and theorems are used to prove the correctness and complexity of OMA.

A. *Correctness of OMA*

To prove our protocol's correctness, a vertex is called common [2] if each correct node has the same value for the vertex. That is, if a vertex is common, then the value stored in the vertex of each correct node's mg-tree or ic-tree is identical. When each correct node has a common initial value received from the source node in the root of an ic-tree, an agreement is reached because the root is common. Thus, the constraints, (Agreement) and (Validity), can be rewritten as:

(Agreement'): Root *s* is common and
(Validity'): VOTE(*s*) = v_s for each correct node, if the source node is correct.

To prove that a vertex is common, the term *common frontier* [11] is defined as: when every root-to-leaf path of the mg-tree contains a common vertex, the collection of the common vertices forms a common frontier. Based on these two terms, *common* and *common frontier*, the correctness of OMA can be examined as follows.

Lemma 1: All correct vertices of an ic-tree are common.
Proof: After reorganization, no repeatable vertices are in an ic-tree. At the level T_{FG} +1 or above, the correct vertex α has at least $2T_{FG}$ + 1 children where at least T_{FG} +1 children are correct. The true values of these T_{FG} +1 correct vertices are common, and the majority value of vertex α is common. The correct vertex α is common in the ic-tree, if the level of α is less than T_{FG} +1. As a result, all correct vertices of the ic-tree are common.

Lemma 2: **The common frontier exists in the ic-tree.**
Proof: There are T_{FG} +1 vertices along each root-to-leaf path of an ic-tree in which the root is labeled by the source name and the others are labeled by a sequence of group names. Since, at most, T_{FG} groups can fail, there is at least one correct vertex along each root-to-leaf path of the ic-tree. Following Lemma 1, the correct vertex is common, and the common frontier exists in each correct node's ic-tree.

Lemma 3: Let α be a vertex; α is common if there is a common frontier in the subtree rooted at α.
Proof: If the height of α is 0 and the common frontier (α itself) exists, then α is common. If the height of α is σ, the children of α are all common following the induction hypothesis with the height of the children being σ-1; then, the vertex α is common.

Corollary 1: The root is common if the common frontier exists in the ic-tree.

Theorem 1: The root of a correct node's ic-tree is common.
Proof: By Lemma 1, Lemma 2, Lemma 3 and Corollary 1, the theorem is proved.

Theorem 2: OMA Protocol solves the BA problem in a virtual subnet-based cloud computing environment.
Proof: To prove the theorem, one has to show that OMA meets the constraints (Agreement') and (Validity')
(**Agreement**'): Root *s* is common. By Theorem 1, (Agreement') is satisfied.
(**Validity**'): VOTE(*s*)=*v* for all correct nodes, if the initial value of the source is v_s, say $v = v_s$.

Level 1	*Level 2*	*Level 3*		*Take RMAJ*
s	*s1*	*s11*	0	(0)
0	0(0)	*s12*	0	(0,0,***0***,0)
		s13	0	(0,***1***,0,0)
		s14	0	(0,0)
		s15	0	(0,0)
		s16	0	(0,0)
		s17	1	(***1***,***1***,1,***0***,1)
	s2	*s21*	1	(1)
	1(1,1,1,1)	*s22*	1	(1,1,***1***,1)
		s23	1	(1,***1***,1,1)
		s24	1	(1,1)
		s25	1	(1,1)
		s26	1	(1,1)
		s27	0	(***0***,***0***,1,***0***,1)
	s3	*s31*	0	(0)
	0(0,0,0,0)	*s32*	0	(0,0,***1***,0)
		s33	0	(0,***1***,0,0)
		s34	0	(0,0)
		s35	0	(0,0)
		s36	0	(0,0)
		s37	0	(***0***,***0***,1,***0***,1)
	s4	*s41*	1	(1)
	1(1,1)	*s42*	1	(1,1,***0***,1)
		s43	1	(1,***1***,1,1)
		s44	1	(1,1)
		s45	1	(1,1)
		s46	1	(1,1)
		s47	1	(***1***,***1***,1,***0***,1)
	s5	*s51*	1	(1)
	1(1,1)	*s52*	1	(1,1,***1***,1)
		s53	1	(1,***0***,1,1)
		s54	1	(1,1)
		s55	1	(1,1)
		s56	1	(1,1)
		s57	0	(***0***,***0***,1,***0***,1)
	s6	*s61*	1	(1)
	1(1,1)	*s62*	1	(1,1,***1***,1)
		s63	1	(1,***1***,1,1)
		s64	1	(1,1)
		s65	1	(1,1)
		s66	1	(1,1)
		s67	1	(***1***,***1***,1,***1***,1)
	s7	*s71*	0	(0)
	0(0,0,1,0,1)	*s72*	1	(1,1,***1***,1)
		s73	0	(0,***0***,0,0)
		s74	1	(1,1)
		s75	0	(0,0)
		s76	1	(1,1)
		s77	0	(***0***,***0***,1,***0***,1)

Figure 6. The final mg-tree of node P_1 after the message exchange phase.

Level 1	*Level 2*	*Level 3*		*Take RMAJ*
s	*s1*			
0	0	*s12*	0	(0,0,***0***,0)
	(0)	*s13*	0	(0,***1***,0,0)
		s14	0	(0,0)
		s15	0	(0,0)
		s16	0	(0,0)
		s17	1	(***1***,***1***,1,***0***,1)
	s2	*s21*	1	(1)
	1			
	(1,1,1,1)	*s23*	1	(1,***1***,1,1)
		s24	1	(1,1)
		s25	1	(1,1)
		s26	1	(1,1)
		s27	0	(***0***,***0***,1,***0***,1)
	s3	*s31*	0	(0)
	0(0,0,0,0)	*s32*	0	(0,0,***1***,0)
		s34	0	(0,0)
		s35	0	(0,0)
		s36	0	(0,0)
		s37	0	(***0***,***0***,1,***0***,1)
	s4	*s41*	1	(1)
		s42	1	(1,1,***0***,1)
		s43	1	(1,***1***,1,1)
		s45	1	(1,1)
		s46	1	(1,1)
		s47	1	(***1***,***1***,1,***0***,1)
	s5	*s51*	1	(1)
	1	*s52*	1	(1,1,***1***,1)
	(1,1)	*s53*	1	(1,***0***,1,1)
		s54	1	(1,1)
		s56	1	(1,1)
		s57	0	(***0***,***0***,1,***0***,1)
	s6	*s61*	1	(1)
	1	*s62*	1	(1,1,***1***,1)
	(1,1)	*s63*	1	(1,***1***,1,1)
		s64	1	(1,1)
		s65	1	(1,1)
		s67	1	(***1***,***1***,1,***1***,1)
	s7	*s71*	0	(0)
	0	*s72*	1	(1,1,***1***,1)
	(0,0,1,0,1)	*s73*	0	(0,***0***,0,0)
		s74	1	(1,1)
		s75	0	(0,0)
		s76	1	(1,1)

The tree structure has converted from mg-tree to ic-tree by erasing the vertices with repeated names.

Figure 7. The ic-tree of node P_1.

- ✧ VOTE(*s1*) = (0, 0, 0, 0, 0, 1) = 0
- ✧ VOTE(*s2*) = (1, 1, 1, 1, 1, 0) = 1
- ✧ VOTE(*s3*) = (0, 0, 0, 0, 0, 0) = 0
- ✧ VOTE(*s4*) = (1, 1, 1, 1, 1, 1) = 1
- ✧ VOTE(*s5*) = (1, 1, 1, 1, 1, 0) = 1
- ✧ VOTE(*s6*) = (1, 1, 1, 1, 1, 1)) = 1
- ✧ VOTE(*s7*) = (0, 1, 0, 1, 0, 1) = ϕ

VOTE(*s*) = (VOTE(*s1*), VOTE(*s2*), VOTE(*s3*), VOTE(*s4*), VOTE(*s5*), VOTE(*s6*), VOTE(*s7*)) = (0, 1, 0, 1, 1, 1, ϕ) = 1

Figure 8. The common value VOTE(*s*) by correct node P_1.

 ISBN: 978-1-61208-280-6

Since the most of the nodes are correct, they transmit the messages to all others. The value of correct vertices for all the correct nodes' mg-tree is v. When the mg-tree is reorganized to an ic-tree, the correct vertices still exist. As a result, each correct vertex of the ic-tree is common (Lemma 1) and its true value is v. following Theorem 1, this root is common. The computed value VOTE(s) = v is stored in the root for all correct nodes. Thus, (Validity') is satisfied.

B. Complexity of OMA

The complexity of OMA is evaluated in terms of: 1) the minimal number of rounds; and 2) the maximum number of allowable faulty components. Theorems 3 and 4 below will show that the optimal solution was reached.

Theorem 3: OMA requires T_{FG} +1 rounds to solve the BA problem with malicious faults in a virtual subnet-based cloud computing environment where $T_{FG} \leq \lfloor (g\text{-}1)/3 \rfloor$.

Proof: Message passing is required in the *Message Exchange Phase* only. Thus, the message exchange phase is a time consuming phase. Fischer [2] pointed out that t+1 ($t \leq \lfloor (n\text{-}1)/3 \rfloor$) rounds are the minimum number of rounds to get enough messages to achieve BA. The unit of Fischer [2] is nodes, but the unit of the virtual subnet-based cloud computing environment is groups. Here, the number of required rounds of message exchange in the virtual subnet within the cloud computing environment is T_{FG} +1($T_{FG} \leq \lfloor (g\text{-}1)/3 \rfloor$). Thus, OMA requires T_{FG} +1 rounds and this number is the minimum.

Theorem 4: The total number of allowable faulty components by OMA is T_{FG} malicious faulty groups, where $T_{FG} \leq \lfloor (g\text{-}1)/3 \rfloor$.

Proof: The maximal number of allowable faulty nodes to reach BA underlying a fully connected network is f *and* $f \leq \lfloor (n\text{-}1)/3 \rfloor$ [8]. However, the fully connected nature of the virtual subnet-based cloud computing environment is group related; we can suppose a node in Siu *et al.* acts as a group in a virtual subnet-based cloud computing environment [8]. Therefore, $f \leq \lfloor (n\text{-}1)/3 \rfloor$ in Siu *et al.* implies $T_{FG} \leq \lfloor (g\text{-}1)/3 \rfloor$ in a virtual subnet-based cloud computing environment. Therefore, the total number of allowable faulty components by OMA is T_{FG} malicious faulty groups.

As a result, OMA requires a minimal number of rounds and tolerates a maximal number of faulty components to reach a common agreement with correct nodes. Thus, the optimality of the protocol is proven

VI. CONCLUSION

Mobile cloud computing can provide advantages creating better mobile services for users. However, due to the mobility of the network, the nodes of mobile cloud computing may immigrate or emigrate from the network at any time. Furthermore, some of the nodes in the network may be fallible, so the network would not be stable. Notably, the network topology developed in recent years shows a mobile feature [1]. The previous protocols [2,10,11] cannot adapt to solve the BA problem in a virtual subnet of the mobile cloud computing environment. To enhance fault-tolerance, a new OMA protocol is proposed to solve the BA problem herein. OMA uses the minimum number of message exchange rounds to allow all correct nodes to agree on a common value and can tolerate the maximum number of allowable faulty components.

Furthermore, in a generalized case, the fallible components are not only nodes, but also communication media. The OMA protocol may be extended to reach BA in a generalized case underlying the topology of a virtual subnet-based cloud computing environment in the future.

REFERENCES

[1] T.C. Chiang, H.M. Tsai, and Y.M. Huang, "A partition network model for ad hoc networks," Proc. IEEE International Conf. on Wireless and Mobile Computing, Networking and Communications, vol. 3, 2005, pp. 467-472.

[2] M. Fischer and N. Lynch, "A lower bound for the assure interactive consistency," Information Processing Letters, vol. 14, no. 4, 1982, pp. 183-186.

[3] A. Klein, C. Mannweiler, J. Schneider, and H.D. Schotten, "Access schemes for mobile cloud computing," Proc. 2010 Eleventh International Conf. on Mobile Data Management, 2010, pp. 387-39.

[4] L. Lamport, R. Shostak, and M. Pease, "The byzantine general problem," ACM Trans. on Programming Language and Systems, vol. 4, no. 3, 1982, pp. 382-401.

[5] Y.T. Larosa, J.L. Chen, D.J. Dengy, and H.C. Chaoz, "Mobile cloud computing service based on heterogeneous wireless and mobile p2p networks," Proc. IEEE 7th International Wireless Communications and Mobile Computing Conference, 2011, pp. 661-665.

[6] B. Lehane and L. Doyle, "Shared RSA key generation in a mobile ad hoc network," Proc. IEEE Conf. of the Military Communications, vol. 2, 2003, pp. 814-819.

[7] M. Min, F. Wang, D.Z. Du, and P.M Pardalos, "A reliable virtual backbone scheme in mobile ad-hoc networks," Proc. IEEE International Conf. on Mobile Ad-hoc and Sensor Systems, 2004, pp. 60-69.

[8] H.S. Siu, Y.H. Chin, and W.P. Yang, "A note on consensus on dual failure modes," IEEE Trans. on Parallel and Distributed Systems, vol. 7, no. 3, 1996, pp. 225-230.

[9] W.T. Tsai, P. Zhong, E. J. Elston, X. Bai, and Y. Chen, "Service replication with mapreduce in clouds," Proc. IEEE International Sym. on Autonomous Decentralized Systems, 2011, pp. 381-388.

[10] S.S. Wang, K.Q. Yan, and S.C. Wang, "Achieving efficient agreement within a dual-failure cloud-computing environment," Expert Systems with Applications, vol. 38, 2011, pp. 906-915.

[11] K.Q. Yan, S.S. Wang, and S.C. Wang "Reaching an agreement under wormhole networks within dual failure component," International Journal of Innovative Computing, Information and Control, vol.6, no.3, 2010, pp. 1151-1164.

 ISBN: 978-1-61208-280-6