ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 4

New Anatomy of Trustworthy Mobile Cloud Computing

Shu-Ching Wang, Shun-Sheng Wang*, Kuo-Qin Yan*

Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C.
e-mail: scwang@cyut.edu.tw, sswang@cyut.edu.tw, kqyan@cyut.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.45.4.12415

Abstract. Nowadays, Mobile Cloud Computing (MCC) is widely accepted as a concept that can significantly improve the user experience when accessing mobile services. For MCCs, a stable and reliable topology is an important research topic. However, the problem of reaching consensus in the distributed system is one of the most important issues to design a fault-tolerance system. The protocols of reaching consensus are required so that the distributed system still can be well performed even if certain components in the system were failed. In this study, the Trusted Timely Computing Base (TTCB) is used when the message is transmitted. However, the consensus problem is revisited with the assumption of transmission medium failure on malicious faults in the Cluster-based MCC in this study. The proposed protocol, Trustworthy MCC (TMCC), can make all fault-free nodes reaching consensus with minimal rounds of message exchanges and tolerate the maximal number of allowable faulty components.

Keywords: distributed consensus problem; mobile cloud computing, fault tolerant, trusted timely computing base; malicious fault.

1. Introduction

Mobile Cloud computing (MCC) at its simplest refers to an infrastructure where both the data storage and the data processing happen outside of the mobile device [1]. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and mobile computing not just to smart phone users but a much broader range of mobile subscribers [2]. There are certain requirements of MCC that need to be met in a cloud such as adaptability, scalability, availability and self-awareness [3]. Therefore, in MCC, a mobile entity can be considered either as a physical mobile device or a mobile computing/storage software agent within a virtualized cloud resource provisioning system [4].

As MCC has become increasingly popular, network topology has trended toward wireless connectivity, thus providing enhanced support for MCC [5]. This technological trend has greatly encouraged distributed system design and support for cloud nodes [6]. The ‘cluster’ has attracted significant attention recently because it requires less infrastructure, it can be deployed quickly, and it can automatically adapt to changes in topology. Therefore, the structure of a cluster can suit military communication, emergency disaster rescue operations, and law enforcement [6], and be used to the cloud-computing technology of MCC [4,6].

In a MCC, the network is assumed reliable and synchronous [7]. The protocols of reaching consensus are required so that the distributed system still can be well performed even if certain components were failed by inner damage or outer intruder. Therefore, the Trusted Timely Computing Base (TTCB) is used in this study when the message is transmitted [8,9]. There are two characteristics of TTCB: security and synchronization. Because the TTCB is security, nodes can receive the same result through the TTCB.

However, the applications of consensus on the cluster-based MCC include two-phase commitment in a cluster-based MCC database system [10], the whereabouts of a replicated file in a cluster-based MCC environment [11], and a landing task controlled by a flight path finding system [12]. Therefore, in this study, the consensus problem is revisited with the assumption of transmission medium failure on malicious faults in the cluster-based MCC. The proposed protocol, *Trustworthy MCC* (TMCC), can make all fault-free nodes reaching consensus with minimal rounds of message exchanges and tolerate the maximal number of allowable faulty components.

[*] Corresponding author

The remainder of this paper is arranged as follows: Section 2 illustrates the topology of cluster-based MCC, consensus problem and the security technology. Section 3 illustrates the concept of the Trustworthy MCC (TMCC). An example is given in Section 4. The correctness and complexity of the proposed protocol is explained in Section 5. Finally, conclusions and future works are presented in Section 6.

2. Related Works

The design and development of the trustworthy consensus protocol has several requirements that must be considered. Therefore, the structure of cluster-based MCC, consensus problem and the security technology will be discussed in this section.

2.1. The Structure of Cluster-based MCC

The MCC would also be based on the basic cloud computing concepts [13]. MCC combined the mobile devices and cloud computing to perform the heavily loaded of computing-intensive tasks and to store massive amounts of data [4]. The topology of MCC is shown in Figure 1 [6].

Currently, the cluster cloud is a more practical kind of cloud computing. A cluster of multiple cloud nodes in a cluster cooperates to achieve some objectives [4]. Cluster-based cloud computing consists of a set of loosely or tightly connected cloud nodes that work together so that, in many respects, they can be viewed as a single system. The components of a cloud cluster are usually connected to each other through fast LANs (local area networks) with each cloud node. All cloud nodes of cluster-based cloud computing are usually deployed to offer improved performance and availability compared to a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

However, in cluster-based MCC, from the aspects of mobile computing and cloud computing, mobile cloud computing is a combination of both technologies, the development of distributed, grid and centralized algorithms, as well as prospects for broad application [14]. The cluster-based MCC is shown in Figure 2.

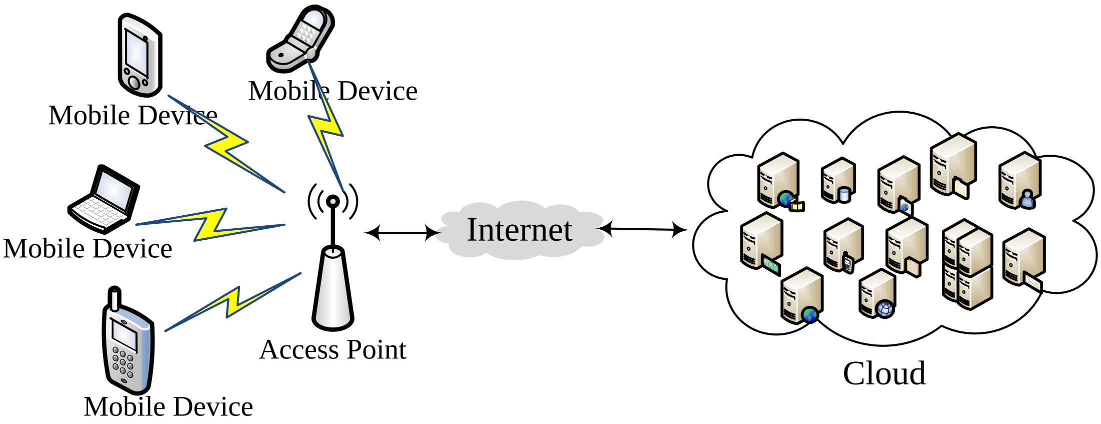


Figure 1. Mobile cloud architecture [6]

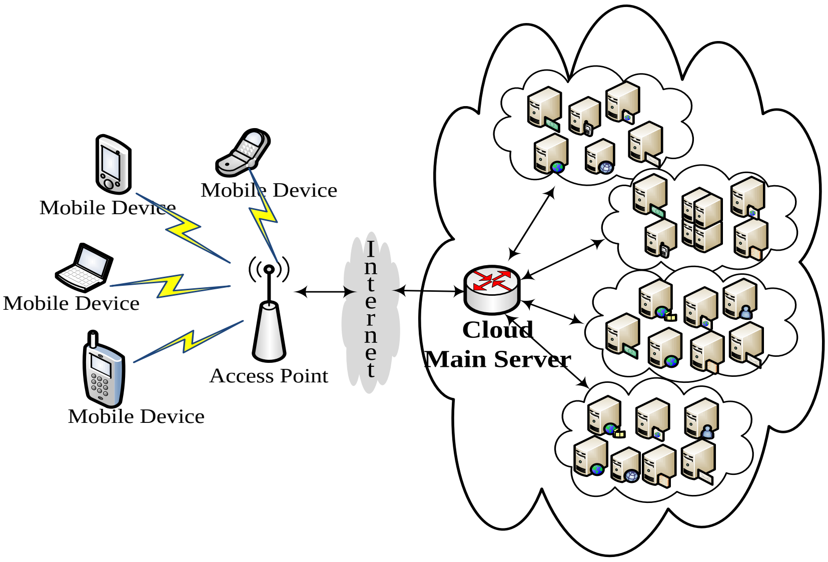


Figure 2. Cluster-based MCC [14]

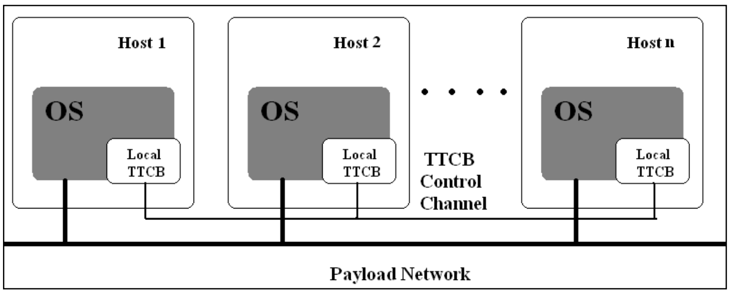


Figure 3. TTCB system structure[8,9]

2.2. Consensus Problem

In the cluster-based MCC, numbers of nodes are interconnected. 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 consensus problem [6] is extended from Byzantine Agreement (BA) problem [15]. The solutions of consensus problem are defined as protocols, which achieve a consensus and hope to use the minimum number of rounds of message exchanges to achieve the maximum number of allowable faulty capability. The solution of consensus problem is concerned in this study. The definition of the problem is to make the fault-free nodes in an n-node cluster-based MCC to reach consensus. Every node chooses an initial value to start with, and communicates to each other by exchanging messages. A group of nodes is referred to make a consensus if it satisfies the following conditions [15]:

(Agreement): All fault-free nodes agree on a common value.

(Validity): If the initial value of each fault-free node n_i is v_i then all fault-free nodes shall agree on the value v_i.

Traditional consensus problem treated transmission medium fault as a node fault in a fail-safe network [16,17]. Base on this assumption, the innocent node connected with a failed transmission medium does not involve consensus [18]. The definition of a consensus problem requires all fault-free nodes to reach a consensus. Therefore, the consensus problem is revisited with the assumption of transmission medium failure on malicious faults in the Cluster-based MCC in this study.

2.3. Security Technology

In a cluster-based MCC, the nodes are interconnected. The protocols of reaching consensus are required so that the distributed system still can be well performed even if certain components were failed by inner damage or outer intruder. In this study, the Trusted Timely Computing Base (TTCB) is used when the message is transmitted [8,9]. There are two characteristics of TTCB: security and synchronization. The structure of the TTCB system is shown in Figure 3. The TTCB system is composed of useful firmware, namely Local TTCB, in the Host and Payload Network which offers the way of connecting with each Host. Trust Block Agreement Service (TBA Service) [8,9] is one of the TTCB provided services and major service in consensus protocol. Because the TTCB is security, nodes can receive the same result through the TTCB.

In this study, a distributed system whose nodes are always reliable during the consensus execution in cluster-based MCC is considered. Underlying cluster-based MCC, the transmission media may be malfunctioning by the interference of intruders, and the exchanged messages may be exhibited in arbitrary behavior. A protocol to achieve consensus in an unreliable communication environment has been proposed before [19]. The proposed protocol can tolerate $\lceil c/2 \rceil$-1 faulty transmission media where c is the connectivity of network [17]. When all nodes reach consensus in cluster-based MCC, the fault-tolerance capacity of the system is enhanced due to each node can transmit its messages to others directly without the influence of any transmission medium fault.

3. The Proposed Protocol

This study proposes a new protocol, called *Trustworthy MCC* (TMCC), to solve the consensus problem even if the faulty transmission media change the transmitted messages to influence the system to achieve consensus in a cluster-based MCC. The proposed protocol TMCC consists of two phases, the message exchange phase and decision making phase. Moreover, TMCC only needs two rounds of message exchanges to solve the consensus problem. In the first round of the message exchange phase, each node n_i multicasts its initial value v_i through transmission media by TTCB and then receives the initial value of other nodes by TTCB as well. In the second round, each node n_i acts as the sender, sends the vector V_i received in the first round by TTCB, and constructs a matrix $[V_1, V_2, ..., Vi]$, denoted by MAT_i, $1 \leq i \leq n$. Finally, the decision making phase will reach consensus among the nodes. The assumptions and parameters of this network topology are shown below.

- Each node in the network can be identified uniquely.
- Let n be the total number of nodes in the cluster-based MCC.
- Let C be the total number of clusters in the cluster-based MCC, and $C \geq 4$. However, C can be determined by the specific applications [20].
- Let j be the cluster identifier, where $1 \leq j \leq C$ and $C \geq 4$.
- TM_{ij} is the transmission medium between clusters C_i and C_j.
- IT_{ij} is the set of transmission media between clusters C_i and C_j. If the number of faulty transmission media in IT_{ij} is greater than or equal to a half of the set, then the IT_{ij} is a faulty IT; otherwise, it is a fault-free IT.
- Let f_{IT} be the number of faulty ITs in all clusters.
- Let c be the connectivity of a cluster-based MCC, and $c \geq 2f_{IT}+1$.
- Let TF_{TM} be the total number of allowable faulty TMs.
- Let v_{ki} denoted as the value stored in the k-th row and i-th column of a matrix.

The TMCC protocol is shown in Figure 4. In the TMCC protocol, MAT_i is the matrix set up at node n_i for i = 1 to n. However, the functions MAJ_k and DEC_i are used in TMCC to determine the agreement value. MAJ_k is a majority function that takes the majority value of the k-th row of MAT_i for $1 \leq k \leq n$. DEC_i is a decision function and is defined below:

```
if ($\exists MAJ_k = \neg v_i$) then
    $DEC_i = \phi$;
if ($\exists MAJ_k = ?$) and ($v_{ki} = v_i$) then
    $DEC_i = \phi$;
else
    $DEC_i = v_i$;
```

The concept of TMCC execution is shown in Figure 5. Figure 5(a) is an example of 4-cluster MCC. Figure 5(b) is an example of MAT_i constructed by TMCC.

TMCC protocol (for node n_i with initial value v_i)		
Message Exchange Phase:		
Round 1:	Step 1	Broadcasts v_i by TTCB to all nodes, and receives the initial value of other nodes in the cluster-based MCC by TTCB.
	Step 2	Constructs temporary column vector $TV_i=[v_1,v_2,\ldots v_n]$, where n is the total number of nodes in the cluster-based MCC.
	Step 3	Reconstructs column vector $V_i=[v_1,v_2,\ldots v_C]$ after taking a local majority on the messages received from each cluster, where C is the number of clusters.
Round 2:	Step 1	Broadcasts V_i to all nodes by TTCB, and receives the column vector V_j from node n_j by TTCB, for $1 \leq j \leq n$.
	Step 2	Constructs a temporary matrix $TMAT_i = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \cdots \\ V_n \end{bmatrix}$, where n is the number of nodes in the cluster-based MCC.
	Step 3	Reconstructs matrix $MAT_i = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \cdots \\ V_C \end{bmatrix}$ after taking a local majority on the messages received from each cluster, where C is the number of clusters.
Decision Making Phase:		
Step 1:	Take the majority value of the k-th row of MAT_i to MAJ_k for $1 \leq k \leq n$.	
Step 2:	Search for any MAJ_k. If ($\exists MAJ_k = \neg v_i$), then $DEC_i := \phi$;	
Step 3:	else if ($\exists MAJ_k = ?$) AND ($v_{ki} = v_i$), then $DEC_i := \phi$; else $DEC_i := v_i$, and terminate.	

Figure 4. The TMCC protocol

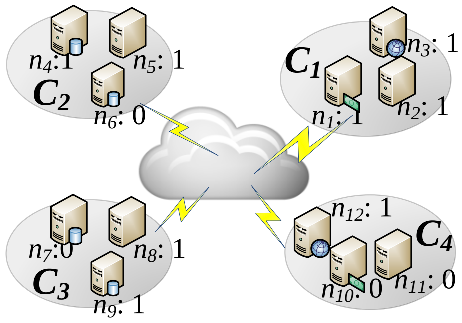
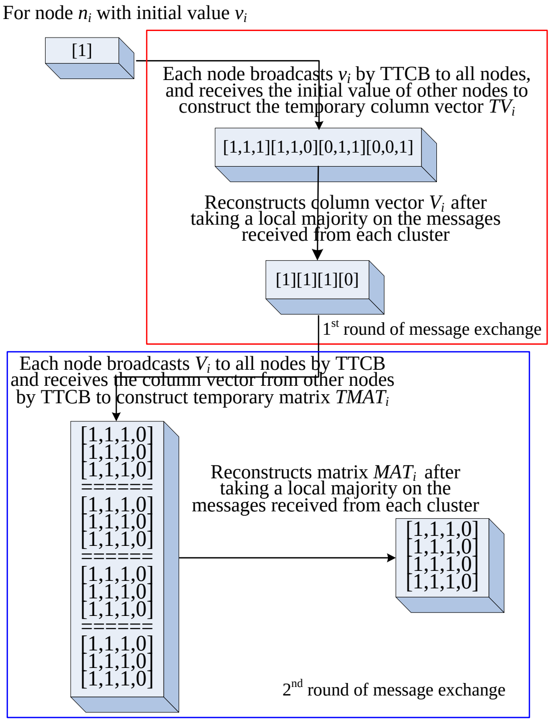


Figure 5(a). An example of 4-cluster MCC


For node n_i with initial value v_i

[1]

Each node broadcasts v_i by TTCB to all nodes, and receives the initial value of other nodes to construct the temporary column vector TV_i

[1,1,1][1,1,0][0,1,1][0,0,1]

Reconstructs column vector V_i after taking a local majority on the messages received from each cluster

[1][1][1][0]

1st round of message exchange

Each node broadcasts V_i to all nodes by TTCB and receives the column vector from other nodes by TTCB to construct temporary matrix $TMAT_i$

[1,1,1,0]
[1,1,1,0]
[1,1,1,0]
======
[1,1,1,0]
[1,1,1,0]
[1,1,1,0]
======
[1,1,1,0]
[1,1,1,0]
[1,1,1,0]
======
[1,1,1,0]
[1,1,1,0]
[1,1,1,0]

Reconstructs matrix MAT_i after taking a local majority on the messages received from each cluster

[1,1,1,0]
[1,1,1,0]
[1,1,1,0]
[1,1,1,0]

2nd round of message exchange

Figure 5(b). An example of MAT_i constructed by TMCC

4. An Example of TMCC Executed

Subsequently, a detailed example of executing the TMCC protocol based on the cluster-based MCC is shown in Figure 6. Figure 6(a) is a 4-cluster MCC in which $IT_{1,10}$, $IT_{2,11}$, $IT_{3,6}$, $IT_{3,12}$, $IT_{4,8}$, and $IT_{9,12}$ are assumed malfunctioning.

In the first round of message exchange, each node n_i multicasts its initial value v_i through transmission media by TTCB to all other nodes, where $1 \le i \le n$, and receives the initial value of other nodes by TTCB. Each node uses the received message to construct vector TV_i, as shown in Figure 6(b). Then, each node reconstructs column vector V_i after taking a local majority on the messages received from each cluster, as shown in Figure 6(c).

In the second round of message exchange, each node multicasts its vector V_i and receives the column vectors from other nodes by TTCB. Each node constructs MAT_i after taking a local majority on the messages received from each cluster as shown in Figure 6(d). Finally, the decision making phase takes the majority value of MAT_i to construct the matrix MAJ_i, as shown in Figure 6(e), and achieves the common value by DEC_i.

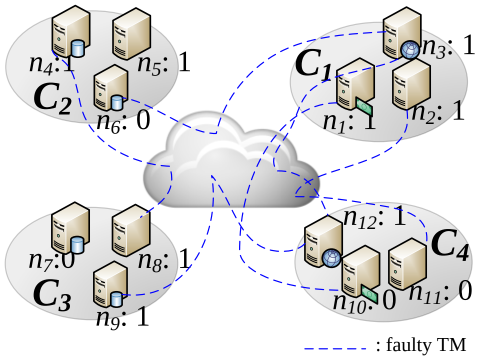


Figure 6(a). A 4-cluster MCC with malicious faulty TMs

	C_1			C_2			C_3			C_4		
	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	n_9	n_{10}	n_{11}	n_{12}
TV_1	1	1	1	1	1	0	0	1	1	***1***	0	1
TV_2	1	1	1	1	1	0	0	1	1	0	***1***	1
TV_3	1	1	1	1	1	***1***	0	1	1	0	0	***0***
TV_4	1	1	1	1	1	0	0	***0***	1	0	0	1
TV_5	1	1	1	1	1	0	0	1	1	0	0	1
TV_6	1	1	***0***	1	1	0	0	1	1	0	0	1
TV_7	1	1	1	1	1	0	0	1	1	0	0	1
TV_8	1	1	1	***0***	1	0	0	1	1	0	0	1
TV_9	1	1	1	1	1	0	0	1	1	0	0	***0***
TV_{10}	***0***	1	1	1	1	0	0	1	1	0	0	1
TV_{11}	1	***0***	1	1	1	0	0	1	1	0	0	1
TV_{12}	1	1	***0***	1	1	0	0	1	***0***	0	0	1

Figure 6(b). The temporary column vector TV_i of each node n_i

	C1	C2	C3	C4
V_1	1	1	1	1
V_2	1	1	1	1
V_3	1	1	1	0
V_4	1	1	0	0
V_5	1	1	1	0
V_6	1	1	1	0
V_7	1	1	1	0
V_8	1	1	1	0
V_9	1	1	1	0
V_{10}	1	1	1	0
V_{11}	1	1	1	0
V_{12}	1	1	0	0

Figure 6(c). The column vector V_i of each node n_i by taking the majority of the value corresponding to each cluster from Figure 6(b)

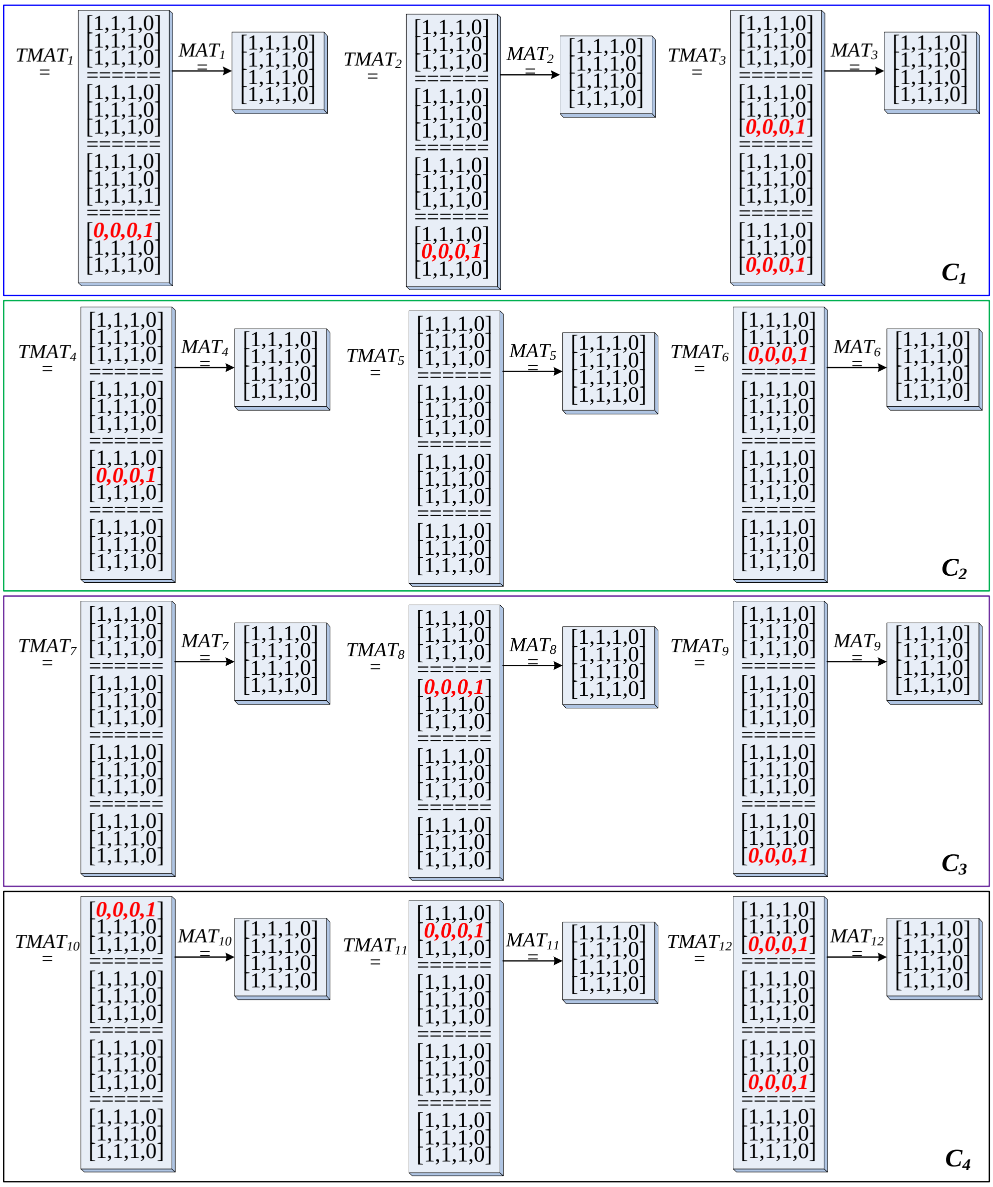


Figure 6(d). The MAT_i after the 2nd round message exchange

$$\begin{array}{l} MAJ \text{ of } MAT_i = \\ \text{for } i = 1 \text{ to } 12 \end{array} \begin{array}{|c|}\hline 1\\\hline 1\\\hline 1\\\hline 1\\\hline\end{array} \quad DEC_i\text{=1 by Step 2 of TMCC}$$

Figure 6(e). The common value DEC_i of node n_i

5. The Correctness and Complexity of TMCC Protocol

The following proofs for the agreement and validity property are given in this section. The lemmas and theorems are used to prove the correctness and complexity of TMCC.

5.1. The Correctness of TMCC Protocol

The lemmas and theorems are used to prove the correctness of TMCC.

Lemma 1. *Let the initial value of sender node n_i is v_i. By using TTCB, the destination cluster's nodes can receive the value v_i from the sender node n_i if $f_{IT} \leq \lceil c/2 \rceil$-1 where f_{IT} is the number of faulty ITs in all clusters and c is the connectivity of a cluster-based MCC.*

Proof. By using TTCB, the sender node can transmit its value to the destination cluster's nodes through TM_{xy} cluster-disjoint paths. According to the assumption of $f_{IT} \leq \lceil c/2 \rceil$-1, the nodes in the destination cluster, in the worst case, can get TM_{xy} values from the sender node. Following the step 3 of round 1 in message exchange phase, a local majority is taken, and then a majority is taken subsequently in the step 1 of decision making phase on these TM_{xy} values. Each of the nodes in the destination cluster gets the value v_s.

Lemma 2. *The decision value DEC_i is equal to majority value.*

Proof. Lemma 2 is proven by the definition of the consensus problem.

Theorem 1. *Protocol TMCC is valid.*

Proof. According to Lemmas 1 and 2, the validity of TMCC is confirmed.

Theorem 2. *Protocol TMCC can make each fault-free node agree on a common consensus.*

Proof. If a node agrees on value Z (where Z = v_i= v_s, and $1 \leq i \leq n$ by Lemma 2) and then all nodes should agree on value Z.

5.2. The Complexity of TMCC Protocol

The complexity of TMCC is evaluated in terms of:
(1) the amount of information exchanges,
(2) the number of rounds of message exchanges, and
(3) the number of allowable faulty components.

Theorems 3, 4 and 5 below will show that the optimal solution is reached.

Theorem 3. *The amount of information exchanges by TMCC is O(n).*

Proof. *In the first round, every node receives one initial value from the source node by using TTCB. In the end round of message exchange phase, n values are received from the other (n–1) nodes in the cluster-based MCC, hence, the total number of message exchanges is 1+(n–1)=n. The result implies that the complexity of information exchanges is O(n).*

Theorem 4: *One round of message exchange cannot solve the consensus problem.*

Proof: *Message exchange is necessary. A node cannot derive whether or not a disagreeable value exists in other nodes without message exchanging. Therefore, consensus problem cannot be implemented. In addition, one round of message exchange is not enough to solve consensus problem. If node n_i of C_x is connected with node n_m of C_y by faulty transmission medium, then node n_i may not know the initial value of node n_m by using only one round of message exchanges. Hence, it is possible to reach a consensus by using one round of message exchanges.*

Theorem 5: *The total number of allowable faulty transmission media by TMCC is optimal.*

Proof. The protocol of Yan et al. [19] can tolerate $\lceil c/2 \rceil$-1 faulty transmission media where c is the connectivity of a fully connected network. However, their results are not appropriate for the cluster-based MCC.

To cope with cluster-based MCC, the total number of faulty ITs in the whole network is $f_{IT}=\lceil c/2 \rceil$ -1. If IT_{ij} is fault, then the total number of faulty TMs between clusters C_i and C_j maybe in the range from $\lceil |TM_{ij}|/2 \rceil$ to $|TM_{ij}|$. Hence, the fault tolerance capability of cluster-based MCC is $\sum_{\min(f_{IT})} \lceil |TM_{ij}|/2 \rceil <= TF_{TM} <= \sum_{\max(f_{IT})} |TM_{ij}| + \sum_{\min(|IT|-f_{IT})} \lceil |TM_{ij}|/2 \rceil$ -1, where TF_{TM} is the total number of allowable faulty TMs, $\max(f_{IT})$ is to get the f_{IT} larger values, $\mathrm{mini}(f_{IT})$ is to get the f_{IT} smaller values, $|TM_{ij}|$ is the number of TM_{ij}s, and |IT| is the number of ITs.

Table 1. The results of previous works dealing with the consensus problem in different network structures

	Network topology			
	FCN	BCN	GCN	cluster-based MCC
Babaoglu and Drummond [25]		◆		
Cheng *et al.* [7]			◆	
Dwork *et al.* [22]	◆			
Wang *et al.*[21]		◆	◆	
Widder *et al.* [23]	◆			
Yan and Chin [24]	◆			
TMCC	◆	◆	◆	◆

6. Conclusion

The consensus problem is a fundamental problem in the distributed environment [15]. The problem has been studied by various kinds of network model in the past [21]. In other words, the consensus problem had been solved in a Fully Connected Network (FCN)[22,23,24], a Broad-Casting Network (BCN)[21,25], or a Generalize Connected Network (GCN)[7,26]. Table 1 summarizes the results of previous works dealing with the consensus problem in different network structures.

According to previous studies, the network topology plays an important role in this problem [7]. However, a FCN-based cloud can be viewed as a special cluster-based MCC with *n* clusters and each cluster contains one node only where *n* is the total number of nodes in the cluster-based MCC. The BCN-based cloud can also be viewed as a special cluster-based MCC with one *n*-node cluster. And, the cluster-based MCC is a kind of GCN. Without losing the generality, the consensus could be reached in FCN, BCN and GCN if the consensus problem can be solved in the case of cluster-based MCC. But, conversely, it is not true.

Therefore, in this study, the consensus problem in cluster-based MCC is revisited. The trust worthy consensus problem is redefined by TMCC protocol within TTCB in a cluster-based MCC and can achieve a common value with two rounds of message exchanges. However, the number of allowable faulty TMs of TMCC is in the range from $\sum_{\min(f_{IT})} \lceil |TM_{ij}|/2 \rceil$ to $\sum_{\max(f_{IT})} |TM_{ij}| + \sum_{\min(|IT|-f_{IT})} \lceil |TM_{ij}|/2 \rceil$-1.

That is, the TMCC has the following features:

- The TMCC can solve the consensus problem in a cluster-based MCC.
- The TMCC allows the design of reliable communication using the Trusted Timely Computing Base (TTCB).
- The TMCC can solve the consensus problem by the minimum number of rounds of message exchanges.
- The TMCC increases the fault tolerance capability by allowing for malicious faulty transmission media.

The symptom of transmission medium faults can be classified as either dormant (e.g., omission, stuck-at, or timing faults) or malicious (also called Byzantine faults). The behavior of malicious faults might be unpredictable and unidentifiable but the receiver can always identify the dormant faults if the protocol appropriately encodes a transmitted message by either the Non-Return-to-Zero code or the Manchester code [19] before transmission. In another word, if the failure types can be classified into malicious fault or dormant fault, then the fault tolerant capability of the proposed protocol can be modified more powerful. Therefore, solving the consensus problem for the highly reliable cluster-basedMCC within malicious and dormant faulty transmission media will be included in our future work.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology MOST 103-2221-E-324-025.

References

[1] **Y. T. Larosa**, **J. L. Chen**, **D. J. Dengy, H. C. Chaoz**. Mobile cloud computing service based on heterogeneous wireless and mobile P2P networks. In: *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey*, 2011, pp. 661-665.

[2] **G. Mukesh, S. Sukhwinder.** Mobile cloud computing. *International Journal of Enhanced Research in Science Technology & Engineering*, 2014, Vol. 3, No. 4, 517-521.

[3] **L. Mei**, **W. Chan, T. Tse.** A tale of clouds: Paradigm comparisons and some thoughts on research issues. In: *Proceedings of the Asia-Pacific Services Computing Conference, Yilan, Taiwan,* 2008, pp. 464-469.

[4] **D. Huang**, **T. Xing**, **H. Wu**. Mobile cloud computing service models: A user-centric approach. *IEEE Network*, 2013, Vol. 27, No. 5, 6-11.

[5] **S. Q. Shahryar, A. Toufeeq, R. Khalid, U. I. Shuja.** Mobile cloud computing as future for mobile applications-implementation methods and challenging issues. In: *Proceedings of the 2011 IEEE International*

Conference on Cloud Computing and Intelligence Systems, Beijing, China, 2011, pp. 467-471.

[6] **A. Khan**, **M. Othman**, **S. Madani**, **S. Khan**. A survey of mobile cloud computing application models. *IEEE Communications Surveys & Tutorials*, 2014, Vol. 16, No. 1, 393-413.

[7] **C.F. Cheng**, **K. T. Tsai**, **H.C. Liao**. A simple and efficient signature-based consensus protocol in the asynchronous distributed system. *Information Technology and Control*, 2012, Vol. 41, No. 2, 183-198.

[8] **C. Miguel, F.N. Nuno, C.L. Lau, V. Paulo.** Low complexity Byzantine-resilient consensus. *Distributed Computing*, 2005, Vol. 17, Issue 3, 237-249.

[9] **P. Veríssimo, A. Casimiro.** The timely computing base model and architecture. *IEEE Transactions on Computers*, August 2002, Vol. 51, No. 8, 916-930.

[10] **B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, R. Sears.** Benchmarking cloud serving systems with YCSB. In: *Proceedings of the 1st ACM Symposium on Cloud Computing, Indianapolis, USA*, 2010, pp. 143-154.

[11] **R. Mukundan, S. Madria**, **M. Linderman**. Efficient integrity verification of replicated data in cloud using homomorphic encryption. *Distributed and Parallel Databases*, 2014, Vol. 32, Issue 4, 507-534.

[12] **V. Lawson, V. Kumar, L. Ramaswamy.** Mobile cloud enabled sensor services: Opportunities, challenges and approaches. In: *Proceedings of 2015 IEEE International Conference on Mobile Services, New York, USA*, 2015, pp. 292-297.

[13] **F. Niroshinie, W.L. Seng, R. Wenny.** Mobile cloud computing: A survey. *Future Generation Computer Systems*, 2013, Vol. 29, No. 1, 84-106.

[14] **W.T. Tsai, P. Zhong, E.J. Elston, X. Bai, Y. Chen.** Service replication with OACM reduce in clouds. In: Proceedings of the International Symposium on Autonomous Decentralized Systems, Tokyo & Hiroshima, Japan, 2011, pp. 381-388.

[15] **L. Lamport**, **R. Shostak**, **M. Pease**. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems,* 1982, Vol. 4 Issue 3, 382-401.

[16] **M. Fischer**. The consensus problem in unreliable distributed systems (A brief survey). *Lecture Notes in Computer Science*, 1983, Vol. 158, 127-140.

[17] **M. Pease, R. Shostak, L. Lamport**. Reaching agreement in the presence of faults. *Journal of the ACM*, 1980, Vol. 27, No. 2, 228-234.

[18] **K.Q. Yan, S.S. Wang, S.C. Wang.** The Agreement problem in unreliable scale-free network. *The Computer Journal*, 2009, Vol. 52, No.4, 499-509.

[19] **K.Q. Yan, S.S. Wang, S.C. Wang.** Reaching an agreement under wormhole networks within dual failure component. *International Journal of Innovative Computing, Information and Control*, 2010, Vol. 6, No. 3, 1151-1164.

[20] **C.A. Leary, R.A. Houze Jr.** The structure and evolution of convection in a tropical cloud cluster. *Journal of the Atmospheric Sciences,* 1979, Vol. 36, 437-457.

[21] **S.C. Wang, C.L. Ho, S.S. Wang, K.Q. Yan**. Reaching consensus underlying fallible cluster-based wireless sensor network. *International Conference on Innovation and Management (IAM 2012), Republic of Palau,* 2012, pp. 101.

[22] **C. Dwork, N. Lynch, L. Stockmeyer**. Consensus in the presence of partial synchrony. *Journal of the ACM*, 1988, Vol. 35, No. 2, 288-323.

[23] **J. Widder, G. Gridling, B. Weiss, J.P. Blanquart.** Synchronous consensus with mortal Byzantines. In: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Edinburgh, *United Kingdom*, 2007, pp. 102-112.

[24] **K.Q. Yan, Y.H. Chin.** An optimal solution for consensus problem in an unreliable communication system. In: *Proceedings of the International Conference on Parallel Processing, Florida, USA*, August 1988, pp. 388-391.

[25] **O. Babaoglu**, **R. Drummond**. Streets of Byzantium: Network architectures for fast reliable broadcasts. *IEEE Transactions on Software Engineering*, June 1985, Vol. SE-11, No. 6, 546-554.

[26] **S.C. Wang, Y.H. Chin, K.Q. Yan**. Byzantine agreement in a generalized connected network. *IEEE Trans. Parallel and Distributed Systems*, 1995, Vol. 6, No. 4, 420-427.

Received October 2015.