ICIC Express Letters
Part B: Applications
Volume **10**, Number **9**, September **2019**

ICIC International ©2019 ISSN 2185-2766
pp. 773–780

OPTIMAL CONSENSUS ACHIEVEMENT FOR THE INTERNET OF THINGS BASED ON FOG COMPUTING WITHIN DUAL FAULTY TRANSMISSION MEDIA

Shu-Ching Wang[1], Wei-Shu Hsiung[1], Chia-Fen Hsieh[1] and Yao-Te Tsai[2,*]

[1]Department of Information Management
Chaoyang University of Technology
168, Jifeng East Road, Wufeng District, Taichung 41349, Taiwan
{ scwang; s10714902; cfhsieh }@cyut.edu.tw

[2]Department of International Business
Feng Chia University
100, Wenhwa Road, Seatwen, Taichung 40724, Taiwan
*Corresponding author: yaottsai@fcu.edu.tw

Received January 2019; accepted April 2019

Abstract. *The Internet of Things (IoT) paradigm is a dynamic and global network infrastructure. Because Fog computing is an emergency architecture for computing, storage, control, and networking, these services can be distributed to the cloud to the end users of the IoT. Therefore, an IoT platform that integrates Fog and Cloud computing (FC-IoT) is used in this study; it can support the applications of IoT. In order to cope with the impact of a faulty transmission medium, it is important to reach a consensus in the event of a failure before performing certain special tasks. Therefore, the consensus problem is revisited in the FC-IoT within dual faulty transmission media. Then, a highly reliable IoT platform can be supported and the IoT applications can be provided.*

Keywords: IoT, Fog computing, Cloud computing, Consensus, Dual faulty mode

1. **Introduction.** Fog computing extends the Cloud computing paradigm to the edge of the network, thus enabling a new breed of applications and services [1]. And, the characteristics of the Fog computing can make a number of critical IoT services and applications. The IoT has greatly encouraged distributed systems design and practiced to support user-oriented service applications [2]. However, distributed systems have grown rapidly in both size and number. In a distributed computing system, nodes allocated to different places or in separate units are connected together so that they collectively may be used to greater advantage. In many cases, reaching a common agreement in the presence of faulty components is the central issue of fault-tolerant distributed computing, because many applications require such agreement [3]. Furthermore, many applications of IoT provide the convenience of users. For users, the system must provide better reliability and fluency [2]. Therefore, reliability is one of the most important aspects of IoT. To ensure that an IoT environment is reliable, a mechanism to allow a set of nodes to reach an agreed value is necessary.

In order to provide a high flexible and reliable platform of IoT, an IoT platform that integrates Fog computing and Cloud computing (FC-IoT) is used in this study. In an IoT environment, a mechanism to allow a given set of nodes to agree on a common value is necessary for reliable smart application [4]. Such a unanimity problem was called consensus problem [5]. It requires a number of independent nodes to reach consensus in cases where some of those components might be faulty. In our study, the consensus problem of FC-IoT will be explored.

DOI: 10.24507/icicelb.10.09.773

The consensus problem is defined by Meyer and Pradhan [5]. 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. In this study, the solution of consensus problem is concerned in the FC-IoT. The definition of the problem is to make the fault-free nodes in the FC-IoT to reach consensus. Each node chooses an initial value to start with, and communicates to each other by exchanging messages. The nodes are referred to make a consensus if it satisfies the following conditions [5].

Consensus: 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.

In a consensus problem, many cases are based on the assumption of node failure in a fail-safe network [6]. According to the assumption of node failure, a Transmission Medium (TM) fault is unfairly treated as a node fault, regardless the correctness of an innocent node; hence, an innocent node does not involve consensus [6]. This is a contradiction with the definition of consensus problem which requires all fault-free nodes to achieve a consensus. In the FC-IoT, numerous nodes are interconnected. Achieving consensus on a same value in the FC-IoT even if certain TMs are fallible, the protocol is required so that systems can still operate correctly. However, in previous studies, the consensus protocols within faulty TMs were designed in traditional network topology [5]. Wang et al. [7] had solved this problem by protocol FCC on an FC-IoT, but they treated all TM failures as malicious. Actually, the symptom of a faulty TM can be classified into two types: dormant (such as crash, stuck-at, or delay) and malicious. The dormant faults of a TM always can be identified by the receiver if the transmitted message was encoded appropriately (i.e., by NRZ-code, Manchester code [8]) before transmission. On the other hand, the malicious faulty TMs are unpredictable. In this study, the consensus problem to enlarge the fault tolerant capability by allowing both dormant faults and malicious faults exist simultaneously (named dual failure mode) on an FC-IoT is revisited. And, the protocol FC Dual Consensus (FDCC) protocol is proposed in this study to solve the consensus problem.

The rest of this paper is organized as follows. Section 2 will serve to introduce the FC-IoT used in this study. The proposed FDCC of FC-IoT will be brought up and illustrated in detail in Section 3. An example of executing the proposed protocol is given in Section 4. Section 5 is responsible for proving the complexity of our new protocol. Finally, Section 6 gives conclusions of this research.

2. The Network Structure.

In the IoT environment, various types of sensor data in real life can be collected through a combination of a large number of sensors. Using these huge sensory data from all of them, a wide range of application services can be provided. For example, FC-IoT can be used as a prevent disaster monitoring system. In the FC-IoT, there are three layers: the *IoT sensors layer*, the *Fog computing layer* and the *Cloud computing layer*. The IoT sensors layer consists of sensor nodes that are responsible for sensing the data required for IoT applications. The Fog computing layer is constructed by a set of Fog groups; each Fog group consists of a large number of Fog nodes, which are responsible for processing specific information and judgments. The Cloud computing layer consists of many Cloud nodes that provide services for Cloud users.

At FC-IoT, the sensed data of the sensors in different regions are sent to the corresponding Fog group in the Fog computing layer, and the data is processed by the Fog nodes in the particular Fog group. Each Fog group collects relevant monitoring information for different regions and then analyzes the collected information in each Fog group. Finally, the status of the monitored area is then transmitted to the disaster prevention center of the Cloud computing layer so that government decisions can be made.

In short, the FC-IoT is proposed by the integration of Fog computing and Cloud computing, where data can be analyzed and processed by devices in the network rather than being centralized in the Cloud computing. By coordinating and managing the computing and storage resources at the edge of the network, more and more connected devices and the emerging needs of IoT can be processed by the Fog computing. Therefore, the FC-IoT can be made as an appropriate platform for providing the critical services and applications of IoT, including connected vehicle, smart city and so on.

3. **The Proposed Protocol.** In an FC-IoT, a sender's message is always identifiable by a receiver; and the protocol's processing time can be negligible. If each node always works well during the execution of consensus protocol, but TMs may be damaged due to break, some noise or intruder, thus a TM may be in faulty when its transferred message is changed or delayed. Conversely, a TM is fault-free when the transferred message is always received correctly and on time. The symptom of faulty TMs can be divided into two kinds: dormant and malicious. The dormant faults can be identified by the receiver but the malicious faults cannot. Usually a node's computation time is faster than the message transmission time through a TM; hence, a node's computation time for protocol is ignored. Under such an assumption, the protocol can make the fault-free node in an FC-IoT to reach a predefined common value with the minimal number of rounds. The variables used in our study are listed in Table 1.

Table 1. The definition of variables used in FDCC

	Definition
R_j	Sensing region in the IoT sensors layer
s_{ij}	Sensor node in the sensing region R_j of IoT sensors layer, $1 \leq i \leq n_{R_j}$ where n_{R_j} is the number of sensing nodes in sensing region R_j of IoT sensors layer
F_j	Fog group in the Fog computing layer
f_{ij}	Fog node in the Fog group F_j of Fog computing layer, $1 \leq i \leq n_{F_j}$ where n_{F_j} is the number of Fog nodes in Fog group F_j of Fog computing layer
c_j	Cloud node in the Cloud computing layer, $1 \leq j \leq n_C$, where n_C is the number of nodes in Cloud computing layer
R	The total number of sensing regions in IoT sensors layer
F	The total number of Fog groups in Fog computing layer
TM_{RF_j}	The number of TMs between sensing region R_j and Fog group F_j
TM_{F_j}	The number of TMs in Fog group F_j
TM_{FC_j}	The number of TMs between Fog group F_j of Fog computing layer and Cloud computing layer
$f^m_{RF_j}$	The total number of allowable malicious faulty TMs between sensing region R_j and Fog group F_j
$f^d_{RF_j}$	The total number of allowable dormant faulty TMs between sensing region R_j and Fog group F_j
$f^m_{F_j}$	The total number of allowable malicious faulty TMs in Fog group F_j
$f^d_{F_j}$	The total number of allowable dormant faulty TMs in Fog group F_j
$f^m_{FC_j}$	The total number of allowable malicious faulty TMs between Fog group F_j and Cloud computing layer
$f^d_{FC_j}$	The total number of allowable dormant faulty TMs between Fog group F_j and Cloud computing layer

The FDCC is used to solve the consensus problem in the FC-IoT with dual fallible TMs. With consideration for efficient consensus, the sensor nodes of IoT sensors layer are used to sense the required data of a specific application, the Fog nodes of Fog computing layer are used to get a common request of the specific application, and the Cloud node in Cloud computing layer is used to serve the cloud services. In the proposed protocol FDCC, the requests of the application services are obtained from sensor nodes. And, the *common value* of the required data for a specific application is determined by Fog nodes of Fog computing layer. There are two phases that Fog nodes need to execute: the *MEC* (*Messages Exchanged and Collected*) *phase* and *DM* (*Decision Making*) *phase*. The MEC phase is used to collect messages from other nodes. Furthermore, the influence of a faulty TM can be removed. Afterward, in the DM phase, each fault-free node uses the messages received during the MEC phase to determine the common value.

In the MEC phase, each node communicates with other nodes and itself via TMs to get the messages. Finally, the DM phase will get the common value among the nodes. In the first round of the MEC phase, each Fog node f_{ij} multicasts its initial value v_i through TMs, and then receives the initial value of other nodes. The receiver can always detect the message(s) through dormant faulty components if the protocol FDCC encodes a transmitted message by using Manchester code [8]. Hence, if the messages pass through any dormant faulty TMs, then the received message will be replaced by λ. In the second round, each node f_{ij} acts as the sender, sending the vector received in the first round, and constructs a matrix, called the MAT_i, $1 \leq i \leq n_{F_j}$. Finally, the DM phase will get the common value among the nodes. In the FDCC, MAT_i is the matrix set up at node f_{ij} for $1 \leq i \leq n_{F_j}$. However, the MAJ_k and DEC_i are used in FDCC to determine the consensus value. MAJ_k is a majority function that takes the majority value of the k-th row of MAT_i for $1 \leq k \leq n_{F_j}$. The common value DEC_i obtained by Fog nodes will be transferred to Cloud computing layer. The majority of the received common values are taken by Cloud node, and then the consensus value can be obtained. The pseudo code of the FDCC is shown in Figure 1.

4. An Example of Executing FDCC.

Taking the disaster prevention monitoring system constructed by FC-IoT as an example to execute FDCC is discussed in Figure 2. Firstly, each sensor node of region R_1 senses the environment status. The TM between s_{11} and Fog group F_1 is assumed in malicious fault, and the TM between s_{15} and Fog group F_1 is assumed in dormant fault. Then, the sensing statuses of the specific application are transferred to Fog group F_1. Because the TM between s_{11} and Fog group F_1 is malicious fault, the message transmitted by the sensor node s_{11} through the malicious faulty TM will be maliciously changed. And, the TM between s_{15} and Fog group F_1 is in dormant fault; hence, the message transmitted by s_{15} is set to λ. In this example, the message is represented by **bold** and *italics* that indicate the message had been modified.

The Fog node receives the requests sent from sensor nodes, and the received requests are taken as the majority. The majority value is used as the initial value (v_i) of Fog node. In the first round of the *MEC phase*, each Fog node f_{ij} broadcasts v_i, and then receives the initial value from the other Fog nodes in the same group, and constructs vector V_i. In this case, the TM between f_{11} and f_{16} is assumed in malicious fault, and the TM between f_{12} and f_{13} is assumed in dormant fault. Then, the vector received in first round of Fog group F_1 of Fog computing layer is obtained. In the second round of *MEC phase*, Fog node f_{ij} broadcasts V_i, and then receives the vectors broadcast by other Fog nodes, and construct MAT_i. After that the *DM phase* takes the majority value of MAT_1 to construct the matrix MAJ_1, and achieves the common value DEC_1 $(= 1)$ can obtain group F_1's Fog nodes. The MAT_1 is constructed in second round and MAJ_1 of MAT_1 as majority value.

Finally, the common value of each Fog node in Fog group F_1 is transferred to Cloud computing layer. In this example, the TM between f_{12} and Cloud computing layer is

FDCC

(1) The requests for the application services are sent to the corresponding Fog group of Fog computing layer by IoT sensor nodes.
(2) The Fog node f_{ij} receives the requests sent from sensor nodes, and the received requests are taken as the majority.
(3) The majority value is used as the initial value (v_i) of Fog node f_{ij}.
(4) The Fog nodes of Fog computing layer execute the following steps to get the common value.

Phase 1: MEC phase

Round 1: Node f_{ij} broadcasts v_i, and then receives the initial value from the other nodes in the same group, and constructs vector V_i. If a dormant fault is found, it will be set to λ by the receiver standing for a dormant fault.

Round 2: Node f_{ij} broadcasts V_i, and then receives the vectors broadcast by other nodes, and MAT_i is constructed by the following steps. If the TM between two nodes is dormant fault then λ is stored.

Step 1: Receive the initial value v_i from node f_{ij}, $1 \leq i \leq n_{F_j}$.
Step 2: Construct the vector $V_i = [v_1, v_2, \ldots, v_n]$, $1 \leq i \leq n_{F_j}$.
Step 3: Broadcast V_i to all nodes, and receive column vector V_k from node f_{kj}, $1\leq k \leq n_{F_j}$.
Step 4: Construct an MAT_i (Setting the vector v_k in column k, $1 \leq k \leq n_{F_j}$).

Phase 2: DM phase:

Step 1: Each λ value is ignored and does not join to majority.
Step 2: Take the majority value of the k-th row of MAT_i to MAJ_k, $1\leq k \leq n_{F_j}$.
Step 3: Search for any MAJ_k. If ($\exists MAJ_k = \neg v_i$), then DEC_i: $= \phi$;
Step 4: Else if ($\exists MAJ_k =?$) AND ($v_{ki} = v_i$), then DEC_i: $= \phi$; else $DEC_i :=v_i$.

(5) The common value DEC_i is obtained and transferred to Cloud computing layer.
(6) The Cloud nodes of Cloud computing layer get the common values DEC_i received from the Fog nodes of Fog computing layer and take the majority value as the consensus value.

Figure 1. The proposed FDCC

assumed in malicious fault, the TM between f_{14} and Cloud computing layer is assumed in dormant fault. The Cloud nodes in Cloud computing layer receive the common value of each Fog node in Fog group F_1, and the received common values are taken as the majority. The majority value is the environmental status of region R_1 in IoT sensors layer determined by the disaster prevention monitoring system.

5. **The Complexity of the FDCC Protocol.** The following theorems are used to prove the complexity of FDCC.

Theorem 5.1. *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. Hence, the consensus problem cannot be implemented. In addition, one round of message exchange is not enough to solve the consensus problem. If node n_i is connected with node n_m by faulty TM, node n_i may not know the initial value in 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.

s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
1	1	1	1	1

(a) The sensing data of each sensor node in the IoT sensors layer

	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	*Majority*
f_{11}	***1***	1	1	1	**λ**	1
f_{12}	***0***	1	1	1	**λ**	1
f_{13}	***1***	1	1	1	**λ**	1
f_{14}	***0***	1	1	1	**λ**	1
f_{15}	***1***	1	1	1	**λ**	1
f_{16}	***0***	1	1	1	**λ**	1

The received requests sent from sensor nodes and take the majority

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	1

The initial value of each Fog node

(b) The initial value of each Fog node in Fog group F_1

	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
f_{11}	1	1	1	1	1	***0***
f_{12}	1	1	**λ**	1	1	1
f_{13}	1	**λ**	1	1	1	1
f_{14}	1	1	1	1	1	1
f_{15}	1	1	1	1	1	1
f_{16}	***0***	1	1	1	1	1

(c) The vector received in first round of Fog group F_1

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	0
1	1	λ	1	1	1
1	λ	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	*0*	*0*	*0*	*0*	*0*
1	1	1	1	1	1
MAJ_{11} of MAT_{11}					

$DEC_{11} = 1$

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	0
1	1	λ	1	1	1
λ	λ	λ	λ	λ	λ
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
MAJ_{12} of MAT_{12}					

$DEC_{12} = 1$

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	0
λ	λ	λ	λ	λ	λ
1	λ	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
MAJ_{13} of MAT_{13}					

$DEC_{13} = 1$

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	0
1	1	λ	1	1	1
1	λ	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
MAJ_{14} of MAT_{14}					

$DEC_{14} = 1$

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
1	1	1	1	1	0
1	1	λ	1	1	1
1	λ	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
MAJ_{15} of MAT_{15}					

$DEC_{15} = 1$

f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	*0*	*1*	*0*	*0*	*0*
1	1	λ	1	1	1
1	λ	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
MAJ_{16} of MAT_{16}					

$DEC_{16} = 1$

(d) Construct MAT_1 in second round and MAJ_1 of MAT_1 as majority value

	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	*Majority*
c_1	1	*0*	1	λ	1	1	1
c_2	1	*0*	1	λ	1	1	1
c_3	1	*1*	1	λ	1	1	1
c_4	1	*0*	1	λ	1	1	1
c_5	1	*1*	1	λ	1	1	1

(e) The consensus value of each node in Cloud computing layer

FIGURE 2. An example of executing FDCC

Theorem 5.2. *The constraint of $t > 2TM_m + TM_d$ can be applied to DFCC where t is the total number of TMs in a distributed system, TM_d is the number of allowable dormant faulty TMs, and TM_m is the number of allowable malicious faulty TMs.*

Proof: According to the assumption, the sender node can transmit t copies of the same value to the destination nodes through t TMs. Due to the constraint of $t > 2TM_m + TM_d$, in the worst case, the destination node can get $t - TM_d$ copies of the value from the sender node. Since $t - TM_d > 2TM_m$, the majority value can be taken on these $t - TM_d$ values and let each destination node get the value v_i. Therefore, the constraint of $TM_{RF_j} > 2f^m_{RF_j} + f^d_{RF_j}$ between sensing region R_j of IoT sensors layer and the Fog group F_j of Fog computing layer, the constraint of $TM_{F_j} > 2f^m_{F_j} + f^d_{F_j}$ in Fog group F_j of Fog computing layer, and the constraint of $TM_{FC_j} > 2f^m_{FC_j} + f^d_{FC_j}$ between Fog group F_j of Fog computing layer and the Cloud nodes in Cloud computing layer can be applied.

Theorem 5.3. *The total number of allowable faulty TMs by FDCC is optimal.*

Proof: The total number of allowable faulty TMs by FDCC can be discussed by three parts.

1) **TMs between IoT sensors layer and Fog computing layer:** By using FDCC, the sensor nodes in sensing region R_j can transmit the sensing data to the Fog group F_j through TM_{RF_j} paths. According to the assumption of $f^m_{RF_j} \leq \left\lceil \left(TM_{RF_j} - f^d_{RF_j}\right)/2 \right\rceil - 1$ and $f^d_{RF_j} \leq TM_{RF_j} - 1$, the nodes in the Fog group F_j, in the worst case, can get $TM_{RF_j} - f^d_{RF_j}$ values from the sensor nodes. Since $f^m_{RF_j} \leq \left\lceil \left(TM_{RF_j} - f^d_{RF_j}\right)/2 \right\rceil - 1$ and $f^d_{RF_j} \leq TM_{RF_j} - 1$, the majority can be taken on these $TM_{RF_j} - f^d_{RF_j}$ values and let each of the nodes in the Fog group F_j get the value v_i.
2) **TMs in Fog computing layer:** For the same reason, each Fog node f_{ij} in Fog group F_j can transmit its v_i to other nodes in the same Fog group through TM_{F_j} paths. According to the assumption of $f^m_{FC_j} \leq \left\lceil \left(TM_{F_j} - f^d_{F_j}\right)/2 \right\rceil - 1$ and $f^d_{F_j} \leq TM_{F_j} - 1$, the nodes in the Fog group F_j, in the worst case, can get $TM_{Fj} - f^d_{F_j}$ values from other nodes in the same Fog group. Since $f^m_{FC_j} \leq \left\lceil \left(TM_{F_j} - f^d_{F_j}\right)/2 \right\rceil - 1$ and $f^d_{F_j} \leq TM_{F_j} - 1$, the majority can be taken on these $TM_{F_j} - f^d_{F_j}$ values and let each of the Fog nodes in the Fog group F_j get the common value.
3) **TMs between Fog computing layer and Cloud computing layer:** In this case, the Fog nodes in Fog group F_j can transmit the common value to the Cloud computing layer through TM_{FC_j} paths. According to the assumption of $f^m_{FC_j} \leq \left\lceil \left(TM_{FC_j} - f^d_{FC_j}\right)/2 \right\rceil - 1$ and $f^d_{FC_j} \leq TM_{FC_j} - 1$, the Cloud nodes in the Cloud computing layer, in the worst case, can get $TM_{FC_j} - f^d_{FC_j}$ values from the Fog nodes in the Fog group F_j. Since $f^m_{FC_j} \leq \left\lceil \left(TM_{FC_j} - f^d_{FC_j}\right)/2 \right\rceil - 1$ and $f^d_{FC_j} \leq TM_{FC_j} - 1$, the majority can be taken on these $TM_{FC_j} - f^d_{FC_j}$ values and let each of the Cloud nodes in the Cloud computing layer get the consensus value.

6. **Conclusion.** As the same with Wang et al. [7], we consider an FC-IoT whose nodes are reliable during the consensus execution; while the TMs may be disturbed by some faults, break down, stuck-at, noise or an intruder. A new efficient and reliable protocol to achieve consensus in an unreliable transmission FC-IoT is proposed first; then its efficiency and reliability are proved later. The proposed protocol FDCC can tolerate d dormant faulty TMs and m malicious faulty TMs simultaneously exist to reach consensus, where $t > 2m + d$ and t is the total number of TMs in each layer of FC-IoT. In addition,

Table 2. Comparison of the fault tolerant capability between FDCC and FCC

The connectivity of one layer in FC-IoT	FDCC		FCC	
	m	d	m	d
3	0	≤ 2	0	0
	1	0	1	0
4	0	≤ 3	0	0
	1	≤ 1	1	0
5	0	≤ 4	0	0
	1	≤ 2	1	0
	2	0	2	0
6	0	≤ 5	0	0
	1	≤ 4	1	0
	2	≤ 1	2	0
7	0	≤ 6	0	0
	1	≤ 4	1	0
	2	≤ 2	2	0
	3	0	3	0
8	0	≤ 7	0	0
	1	≤ 5	1	0
	2	≤ 3	2	0
	3	≤ 1	3	0

the proposed protocol requires only two rounds of message exchanges. The fault tolerant capability is much better than the results of FCC proposed by Wang et al. [7] whose protocol can tolerate $t > 2m$ faulty TMs only. Table 2 shows the comparison between these two protocols.

Acknowledgment. This work was supported in part by the Ministry of Science and Technology MOST 107-2221-E-324-005-MY3.

REFERENCES

[1] A. Munir, P. Kansakar and S. U. Khan, IFCIoT: Integrated fog cloud IoT: A novel architectural paradigm for the future Internet of Things, *IEEE Consum. Electron. Mag.*, vol.6, no.3, pp.74-82, 2017.

[2] D. Puthal, B. P. S. Sahoo, S. Mishra and S. Swain, Cloud computing features, issues, and challenges: A big picture, *Proc. of 2015 International Conference on Computational Intelligence and Networks*, Bhubaneshwar, India, pp.116-123, 2015.

[3] P. Kumar and S. K. Gupta, Abstract model of fault tolerance algorithm in cloud computing communication networks, *International Journal on Computer Science and Engineering*, vol.3, no.9, pp.3283-3290, 2011.

[4] A. Whitmore, A. Anurag and D. X. Li, The Internet of Things – A survey of topics and trends, *Information Systems Frontiers*, vol.17, no.2, pp.261-274, 2015.

[5] F. J. Meyer and D. K. Pradhan, Consensus with dual failure modes, *IEEE Trans. Parallel & Distributed Systems*, vol.2, no.2, pp.214-222, 1991.

[6] S.-S. Wang and S.-C. Wang, The consensus problem with dual failure nodes in a cloud computing environment, *Information Sciences*, vol.279, pp.213-228, 2014.

[7] S.-C. Wang, Y.-J. Lin and K.-Q. Yan, The optimal solution of consensus in a fog computing based IoT within unreliable communication, *ICIC Express Letters, Part B: Applications*, vol.9, no.12, pp.1209-1216, 2018.

[8] A. M. Călean, B. Cagneau, L. Chassagne, M. Dimian and V. Popa, Novel receiver sensor for visible light communications in automotive applications, *IEEE Sensors Journal*, vol.15, no.8, pp.4632-4639, 2015.