International Journal of Applied Science and Engineering
2019. 16, 1: 57-67

Reaching Fault Diagnosis Consensuson a Multiple Damage Unreliable Wireless Sensor Network

Yao-Te Tsai[a], Shu-Ching Wang[b*] and Mao-Lun Chiang[c]

[a] *International Business, Feng Chia University, Taiwan, R.O.C.*
[b] *Information Management, Chaoyang University of Technology, Taiwan, R.O.C.*
[c] *Information and Communication Technology, Chaoyang University of Technology, Taiwan, R.O.C.*

Abstract: Wireless sensor networks (WSNs) are more and more frequently seen as a solution to large-scale tracking and monitoring applications, because of their low-data-rate, low-energy-consumption, and short-range link network which provides an opportunity to monitor and control the physical world to a previously unprecedented scale and resolution. In order to achieve fault-tolerance of WSN, one must deal with the consensus problem. The consensus problem occurs when the fault-free nodes in a distributed system can reach a common agreement before performing specified in instances where faults may exist. The distributed consensus is discussed in previous works. Most of consensus protocols can reach an agreement by the way of fault masking. However, few of them can detect and locate the faulty components. If the faulty components can be detected and located, then the network can be reconfigured to maintain the performance and integrity of a distributed system. In this study, a new protocol is proposed which can tolerate, detect and locate the maximum number of dual failure transmission media to solve the consensus problem in a WSN.

Keywords: Wireless sensor network, Consensus, Fault tolerant, Fault diagnosis, Dual failure mode.

1. Introduction

Wireless sensor networks (WSNs) are a group of specialized autonomous sensors and actuators with a wireless communications infrastructure, intended to monitor and control physical or environmental conditions at diverse locations and to cooperatively pass their data to a main location and/or pass their control command to a desired actuator through the network [1]. WSNs are the key components of the emerging Internet-of-Things (IoT) paradigm. They are now ubiquitous and used in a plurality of application domains. WSNs are still domain specific and usually deployed to support a specific application [2].

The evolution in WSN, especially on the problems related to energy exhaustion, energy harvesting, mobility and transmission, has open new perspectives for their usage in a smart city application. Since WSNs are deployed in open areas without protection from disaster, they are vulnerable to various types of attacks. Sensor nodes of WSN are essential for detecting various kinds of data of the serious disasters in residential areas [3]. Moreover, the network configuration underlying the emergency must be considered in setting up a new network. Therefore, the stability and reliability of WSNs are important issues to keep environment good for data transmission [4]. In other words, to propose a mechanism to allow all well-perform nodes reach an agreement is necessary to ensure WSN stable and reliable.

Corresponding author; e-mail: scwang@cyut.edu.tw
Received 16 July 2018
doi: 10.6703/IJASE.201906_16(1).057

Accepted 28 June 2019

In a WSN environment, a mechanism to allow a given set of nodes to agree on a common value is necessary for reliable application [5]. Such a unanimity problem was called *agreement problem* [6]. It requires a number of independent nodes to reach agreement in cases where some of components might be faulty. Namely, the goal of agreement is making the fault-free nodes reach a common value. There are three kinds of agreement issues, the *Byzantine agreement* [6], *consensus* [7] and *interaction consistency* (*IC*) [8]. In this study, the *consensus* problem of WSN will be explored.

The *consensus* problem is defined by Meyer & Pradhan [7]. 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 definition of the problem is to make the fault-free nodes 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 [7]:

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.

But the most previous protocols for solving consensus problem are fault masking algorithms to reach agreement [6]. Therefore, in a highly reliable fault tolerant distributed system, the Fault Diagnosis Agreement (FDA) [9,10] is used to detect and locate the faulty components. In this study, the proposed protocol solves the FDA problem if it meets the following constraints:

(Agreement): ***All*** faulty-free nodes identify the common set of faulty components during reaching consensus.

(Fairness): No faulty component is falsely detected as fault-free by any faulty-free node; and no faulty-free node is falsely detected as faulty node by any faulty-free node.

In previous results [7], the consensus problem is based on the assumption of faulty component is node only and the network is fail-safe. Based on this assumption, a transmission medium (TM) fault is treated as a node fault, this treatment regardless of the validity of an innocent node; hence, an innocent node does not involve a consensus. The assumption of fallible component is node only contradicts the definition of consensus problem, which stipulates that all faulty-free nodes should reach a common value in the consensus. Another not reasonable assumption is that the failure type of faulty components is malicious only; however the symptoms of failure types can be classified into dormant and malicious [6]. A dormant faulty component always can be identified by the receiver if the transmitted message was encoded appropriately (i.e. by NRZ-code, Manchester code) before transmission [11]. The behavior of a malicious faulty component is unpredictable and arbitrary. The message transmitted by a malicious faulty component is random or arbitrary. It is the most damaging failure type and causes the worst problem. In this study, the consensus problem is revisited to enlarge the fault tolerant capability by allowing dual faulty TMs (both dormant fault and malicious fault) exist in the WSN.

On the other hand, in this study, a new method to detect/locate the faulty components is proposed. So the proposed protocol can use the minimum number of rounds of message exchanges to tolerate/detect/locate d dormant faults and m malicious faults which exist simultaneously in a WSN to reach a consensus, where $m \leq \lceil (n-d-3)/2 \rceil$ and n is the number of nodes in a WSN.

The rest of this paper is organized as follows. Section 2 proposes the detail descriptions of the proposed protocol Fault Diagnosis Consensus (FDC). In Section 3, an example of executing the proposed is given. Section 4 provides the correctness of FDC. Section 5 gives the conclusion.

2. The Proposed Protocol

The proposed protocol Fault Diagnosis Consensus (FDC) of WSN which can solve the consensus problem with dual failure TMs in a WSN. The parameters used in this study are set as follows:

- n: the total number of nodes in a WSN.
- d: the number of allowable dormant faulty TMs.
- m: the number of allowable malicious faulty TMs.
- v_i: the initial value of node i.
- λ: the value substituted for the received value from a dormant faulty TM.
- V_i: the vector is stored in node i *and* the elements of vector $V_i = [v_1, v_2,\ldots, v_j,\ldots, v_n]$ are received from node j, for $1\le j\le n$.
- MAT_i: the 2-dimension matrix is stored in node i *and* the column j *is setting* by vector V_j for $1\le j\le n$.
- $FDMAT_i$: the 3-dimension matrix is stored in node i *and* the j-th layer of $FDMAT_i$ *is setting* by matrix MAT_i for $1\le j\le n$.
- $temp\_FDMAT_i$: the 2-dimension matrix is constructed by taking the major value of each j-th layer of $FDMAT_i$, for $1\le j\le n$ and $1\le i\le n$.
- MAJ_i: The majority value of node i.
- M_{ab}: the value in $temp\_FDMAT_i$ represents the message transmitted through the TM between nodes a and b.
- ϕ: The default value, and $\phi\in\{0, 1\}$.

There are three phases in FDC: *message exchange phase*, *decision making phase and fault detection phase*. FDC can tolerate d dormant faults and m malicious faults which exist simultaneously in the network, where $m\le\lceil (n-d-3)/2 \rceil$, as if the nodes always work accurately and TMs are fallible, and costs only two rounds of message exchanges to reach the consensus, and only needs one additional round (the third round of message exchange) to detect and locate the faulty TMs.

In the *message exchange phase*, nodes exchange messages to get enough information. In the first round, each node i transmits its initial value v_i through TMs, $1\le i\le n$, and receives the initial value v_j from node j, for $1\le j\le n$; and then constructs the vector $V_i = [v_1, v_2,\ldots, v_j,\ldots, v_n]$. If a dormant TM, say TM_{ik}, was found, then v_k in the vector V_i is replaced as λ, $1\le k\le n$. In the second round, each node i transmits a vector V_i to other nodes, $1\le i\le n$, and then receives the vectors transmitted by other nodes and constructs MAT_i, (Setting the vector V_j in column j, for $1\le j\le n$). If a dormant TM, say ik, was found, then $V_k = [\lambda, \ldots \lambda,\ldots\lambda\ ]$, $1\le k\le n$. In the third round, each node i transmits MAT_i to other nodes, and then receives the matrices transmitted by other nodes and construct $FDMAT_i$, (Setting the matrix MAT_i in j-th layer of $FDMAT_i$, for $1\le j\le n$. If a dormant TM, say TM_{ik}, was found, then all the values of $FDMAT_i$ is set to λ, $1\le k\le n$.

In the *decision making phase*, each node i eliminate all λ in the MAT_i to lessen the influence of faulty behavior, $1\le i\le n$, and takes the majority value of each row k of MAT_i to be MAJ_k, for $1\le k\le n$, If MAJ_k does not exist, then $MAJ_k=\neg v_{ki}$. Finally the common value will be the majority value of MAJ_k, for $1\le k\le n$, if the majority value does not exist then the common value will be the default value ϕ.

In the *fault detection phase*, each node i takes major value of each j-th layer of $FDMAT_i$ to construct $temp\_FDMAT_i$, for $1\le j\le n$ and $1\le i\le n$, then searches for each value in the $temp\_DFMAT_i$, if the value M_{ab} in the $temp\_DFMAT_i$ is λ then the TM between node a and node b is in dormant fault, $1\le a,b\le n$. In FDC, the malicious faulty TMs are detected by two cases.

- Case 1: If the total number of columns in y-th layer of $FDMAT_i$ which have different values from the same column from $temp\_FDMAT_i$ is more than $\lceil (n\text{-}d\text{-}3)/2 \rceil$, where $1 \le y \le n$. Then the TM between node j and node y is in malicious fault. The reason is that if the TM between node j and node y is in malicious fault, then the values in y-th layer of $FDMAT_j$ and values in j-th layer of $FDMAT_y$ are arbitrary.
- Case 2: If the total number of columns in y-th layer of $FDMAT_i$ which have different values from the same column from temp_FDMAT is equal or lessen than $\lceil (n\text{-}d\text{-}3)/2 \rceil$. Then eliminate all λ to compare each y-th layer of $FDMAT_i$ with $temp\_FDMAT_i$ if the values in column x, $1 \le x \le n$, has different value from column x of $temp\_FDMAT_i$; then the TMs between node x and node y are in malicious fault.

The proposed protocol FDC is defined in Figure 1.

Protocol FDC(For node i with initial value v_i,$1 \le i \le n$)

Message Exchange Phase:

Round 1: Transmit v_i to all other nodes and receive the initial value v_j from node j, for $1 \le j \le n$; and then construct the vector $V_i = [v_1, v_2, \ldots, v_j, \ldots, v_n]$, $1 \le j \le n$. If a dormant TM, say ik, was found, then $v_k = \lambda$.

Round 2: Transmit V_i, and then receive the vectors transmitted by other nodes and construct MAT_i, for $1 \le j \le n$. If a dormant TM, say ik, was found, then $V_k = [\lambda, \ldots, \lambda, \ldots \lambda]$.

Round 3: Transmit MAT_i, and then receive the matrices transmitted by other nodes and construct $FDMAT_i$, for $1 \le j \le n$). If a dormant TM, say TM_{ik}, was found, then all the values of $FDMAT_i$ is set to λ.

Decision Making Phase:

Step 1: Eliminate all λ in the MAT_i to lessen the influence of faulty behavior, and take the majority value of each row k of MAT_i to be MAJ_k, for $1 \le k \le n$

Step 2: If MAJ_k does not exist, then $MAJ_k = \neg v_{ki}$,

Step 3: DEC_i =*majority value*; if majority value does not exit then DEC_i = *default valu*e ϕ.

Fault Detection Phase:

Step 1: Taking major value of each j-th layer of $FDMAT_i$ to construct $temp\_FDMAT_i$

Step 2: Search for each value in the $temp\_DFMAT_i$, if the value M_{ab} in the $temp\_DFMAT_i$ is λ then the TMs between node a and node b are in dormant fault.

Step 3: Case 1: Eliminate all λ to compare each y-th layer of $FDMAT_i$ with $temp\_FDMAT_i$,if the total number of columns in y-th layer of $FDMAT_i$ which have different values from the same column from $temp\_FDMAT_i$ is more than $\lceil (n\text{-}d\text{-}3)/2 \rceil$, where $1 \le y \le n$.

➔The TMs between node a and node b are in malicious fault.

Case 2: Eliminate all λ to compare each y-th layer of $FDMAT_i$ with $temp\_FDMAT_i$ if the total number of columns in y-th layer of $FDMAT_i$ which have different values from the same column from temp_FDMAT is $\le \lceil (n\text{-}d\text{-}3)/2 \rceil$.

➔Eliminate all λ to compare each y-th layer of $FDMAT_i$ with $temp\_FDMAT_i$ if the values in column x, $1 \le x \le n$, has different value from column x of $temp\_FDMAT_i$; then the TMs between node x and node y are in malicious fault.

Figure 1. Protocol FDC to achieve fault detection with consensus.

3. An Example of FDC Executing

In this section, an example for executing FDC is given. A WSN is shown in Figure 2, there are five sensor nodes, the malicious TM is between node *a* and node *d*. The dormant TM is between node *d* and node *e*. And, the initial value of each node is 0, 0, 0, 1, and 1 respectively.

In the *messages exchange phase*, each node *i* transmits its initial value v_i to all other nodes through the TMs in the first round, for $a \le i \le e$. The messages received by nodes *a*, *b*, *c*, *d*, and *e* in the first round are illustrated in Figure 3(a). In the second round, each node does the same step to construct the matrix MAT_i in Figure 3(b). In the third round, each node *i* transmits its MAT_i to all other nodes through the TMs, where $a \le i \le e$ to construct the $FDMAT_i$ as shown in Figure 4(a). In the *decision making phase*, the consensus is reached as 0.

In the *fault detection phase*, each node uses messages from the third round to construct $temp\_FDMAT_i$ as shown in Figure 4(b). By Step 2 and Step 3 in the *fault detection phase*, the malicious TM which is between node *a* and node *d* and dormant TM which is between node *d* and node *e* can be detected and located as shown in Figure 4(c).

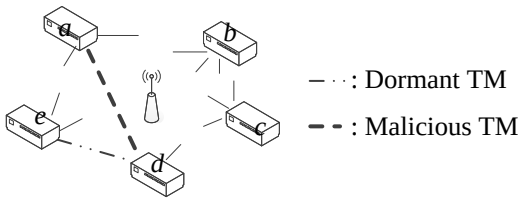

Figure 2. An Example of WSN.

V_a= [0 0 0 ***0*** 1]
V_b= [0 0 0 1 1]
V_c= [0 0 0 1 1]
V_d= [***1*** 0 0 1 λ]
V_e= [0 0 0 λ 1]

Figure 3(a). The received messages of each node in the first round.

$$MAT_a = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & 0 & 1 \end{vmatrix} \quad MAJ_k \text{ of node } a = a \le k \le e \quad \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix} \qquad MAT_i = i = b, c \quad \begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix} \quad MAJ_k \text{ of node } i = a \le k \le e \; i = b, c \quad \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

$$MAT_d = \begin{vmatrix} 0 & 0 & 0 & 1 & \lambda \\ 0 & 0 & 0 & 0 & \lambda \\ 1 & 0 & 0 & 0 & \lambda \\ 1 & 1 & 1 & 1 & \lambda \\ 0 & 1 & 1 & \lambda & \lambda \end{vmatrix} \quad MAJ_k \text{ of node } d = a \le k \le e \quad \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix} \qquad MAT_e = \begin{vmatrix} 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 1 & \lambda & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix} \quad MAJ_k \text{ of node } e = a \le k \le e \quad \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

Figure 3(b). The received messages of each node in the second round
By Step 1,2,3 in the *decision making phase*, DEC_i=0 for all *i*

Figure 3. An example of reaching agreement in WSN.

MAT from node *a* =
$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & 0 & 1 \end{vmatrix}$$
1-st layer of FDMAT$_a$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
2-nd layer of FDMAT$_a$

MAT from node *c* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
3-rd layer of FDMAT$_a$

MAT from node *b* =
$$\begin{vmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{vmatrix}$$
4-th layer of FDMAT$_a$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 1 & \lambda & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
5-th layer of FDMAT$_a$

There are 5 layers in the FDMAT$_a$

MAT from node *a* =
$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & 0 & 1 \end{vmatrix}$$
1-st layer of FDMAT$_b$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
2-nd layer of FDMAT$_b$

MAT from node *c* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
3-rd layer of FDMAT$_b$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & \lambda \\ 0 & 0 & 0 & 0 & \lambda \\ 1 & 0 & 0 & 0 & \lambda \\ 1 & 1 & 1 & 1 & \lambda \\ 1 & 0 & 1 & 1 & 0 \end{vmatrix}$$
4-th layer of FDMAT$_b$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 1 & \lambda & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
5-th layer of FDMAT$_b$

There are 5 layers in the FDMAT$_b$

MAT from node *a* =
$$\begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & 0 & 1 \end{vmatrix}$$
1-st layer of FDMAT$_c$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
2-nd layer of FDMAT$_c$

MAT from node *c* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
3-rd layer of FDMAT$_c$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & \lambda \\ 0 & 0 & 0 & 0 & \lambda \\ 1 & 0 & 0 & 0 & \lambda \\ 1 & 1 & 1 & 1 & \lambda \\ 0 & 1 & 1 & \lambda & \lambda \end{vmatrix}$$
4-th layer of FDMAT$_c$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 1 & \lambda & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
5-th layer of FDMAT$_c$

There are 5 layers in the FDMAT$_c$

MAT from node *a* =
$$\begin{vmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$
1-st layer of FDMAT$_d$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
2-nd layer of FDMAT$_d$

MAT from node *c* =
$$\begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
3-rd layer of FDMAT$_d$

MAT from node *b* =
$$\begin{vmatrix} \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda \\ \lambda & \lambda & \lambda & \lambda & \lambda \end{vmatrix}$$
4-th layer of FDMAT$_d$

MAT from node *b* =
$$\begin{vmatrix} 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 1 & 1 & \lambda & \lambda \\ 1 & 1 & 1 & \lambda & 1 \end{vmatrix}$$
5-th layer of FDMAT$_d$

There are 5 layers in the FDMAT$_d$

Figure 4(a). To construct FDMAT$_i$ for $a \le i \le e$.

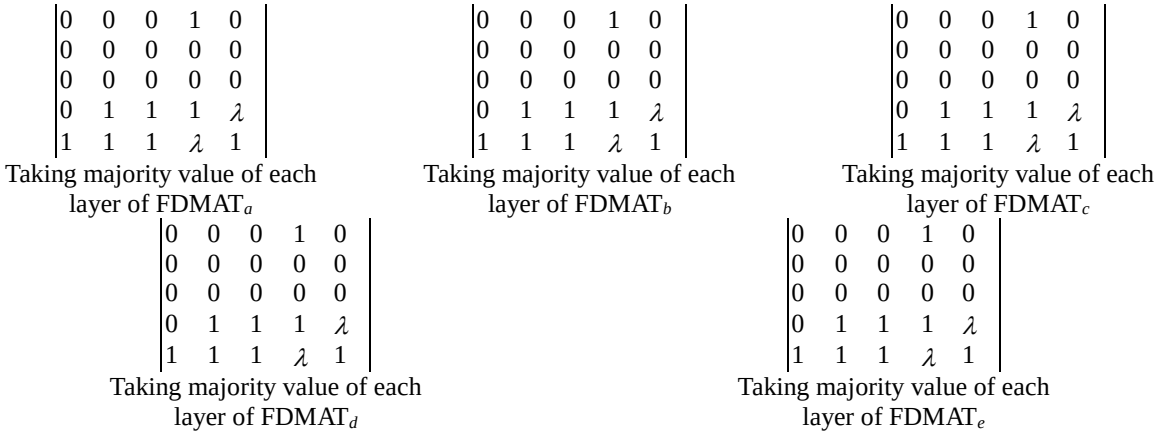

Figure 4(b). To construct temp_FDMATi for a≤i≤e.

By Step 2 in the *fault detection phase*, TM between node *d* and node *e* is in dormant fault.

By Step 3 in the *fault detection phase*, TM between node *a* and node *d* is in malicious fault.

Figure 4(c). The malicious faulty TM and dormant faulty TM are detected and located.

Figure 4. An example of *fault detection phase.*

4. The Correctness of FDC

In this section, the correctness of FDC is proved.

Lemma 1: Let the initial value of node i be v_i and TM_{ij} is fault-free or dormant. Then, the majority value at the i-th row in MAT_j should be v_i.

Proof:

Case 1: TM_{ij} is fault-free, the node j will receive v_i from node i in the first round and $v_{ij} = v_i$ in MAT_j. Meanwhile, the value v_i of node i is broadcasted to the other nodes. There are at most $\lceil (n-d-3)/2 \rceil$ malicious faulty TMs in the WSN. In the second round, node j receives at least $(n-d-1) - \lceil (n-d-3)/2 \rceil = \lceil (n-d+1)/2 \rceil$ v_i's in the i-th row of MAT_j, where d represents the number of λ which is eliminated during the voting for a majority. Hence, at least $\lceil (n-d+1)/2 \rceil$ v_i's are in the i-th row, and the majority value in the i-th row should equal v_i.

Case 2-1: n is odd and TM_{ij} is dormant, the node j receives λ from node i in the first round and $v_{ij} = \lambda$ in MAT_j. Meanwhile, the value v_i of node i is broadcasted to the other nodes. There are at most $\lceil (n-d-3)/2 \rceil$ malicious faulty TMs and d dormant TMs in the WSN. After the second round, node j receives at least $(d+1)$ λ*'s* and at least $n-(d+1) - \lceil (n-d-3)/2 \rceil = \lfloor (n-d+1)/2 \rfloor$ v_i's in the i-th row of MAT_j, where d denotes the number of λ which is eliminated during the voting for a majority. If n is odd then the majority required must be larger than $\lceil (n-1-(d+1)+2)/2 \rceil = \lceil (n-d)/2 \rceil$. Hence, there are $n-(d+1)$ non-λ*'s* and at least $\lfloor (n-d+1)/2 \rfloor$ v_i's in the i-th row. Therefore, the majority value in the i-th row should equal v_i.

Case 2-2: n is even and TM_{ij} is dormant. The node j receives λ from the node i in the first round and $v_{ij} = \lambda$ in MAT_j. Meanwhile, the value v_i of node i is broadcasted to the other nodes. There are at most $\lceil (n-d-3)/2 \rceil$-1 malicious faulty TMs and d dormant TMs in the WSN as if $d \geq 1$ and n is even. After the second round, node j receives at least $(d+1)$ λ*'s* and at least $n-(d+1)-(\lceil (n-d-3)/2 \rceil-1)=\lfloor (n-d+1)/2 \rfloor+1$ v_i's in the i-th row of MAT_j, where d represents the number of λ which is eliminated during

the voting for a majority. Hence, there are n-(d+1) non-λ's and at least $\lfloor(n$-d+1)/2$\rfloor$+1 (larger than $\lceil(n$-1-(d+1)+2)/2$\rceil$=$\lceil(n$-d)/2$\rceil$ the majority required when n is even) v_i's in the i-th row. Therefore, the majority value in the i-th row should equal v_i.

Lemma 2: If the initial value of node i is v_i, then the majority value at the i-th row of MAT$_j$, $1 \le i,j \le n$, should be either v_i or cannot be determined with $v_{ij} = \phi$ regardless of the fault-free condition of TM$_{ij}$.

Proof: By Lemma 1, when TM$_{ij}$ is fault-free or dormant, the majority value of the i-th row in node j is v_i, for $1 \le i,j \le n$. When TM$_{ij}$ is under the influence of malicious fault, there are two cases after running the first round are considered.

Case 1: $v_{ij} = v_i$

Since there are at most $\lceil(n$-d-3)/2$\rceil$ malicious faulty TMs connected to node j, at most $\lceil(n$-d-3)/2$\rceil$ values that may be $\neg v_i$ in the second round. The number of v_i's is (n-d)-$\lceil(n$-d-3)/2$\rceil$ = $\lceil(n$-d+1)/2$\rceil$ in the i-th row where d denotes the number of λ's which is eliminated during the voting for a majority. Therefore, the majority of the i-th row in MAT$_j$ is v_i.

Case 2: $v_{ij} = \phi$

There are at most $\lceil(n$-d-3)/2$\rceil$ malicious faulty TMs. Therefore, in the second round, the total number of ϕ does not exceed $\lceil(n$-d-3)/2$\rceil$+1=$\lceil(n$-d-1)/2$\rceil$ and the number of v_i's is at least (n-d-1)-$\lceil(n$-d+1)/2$\rceil$= $\lfloor(n$-d-1)/2$\rfloor$. If n-d-1 is an even number, then $\lceil(n$-d-1)/2$\rceil$ = $\lfloor(n$-d-1)/2$\rfloor$, the majority of the i-th row in MAT$_j$ cannot be determined. If c-d is an odd number, then $\lceil(n$-d-1)/2$\rceil$>$\lfloor(n$-d-1)/2$\rfloor$. Hence, the majority of the i-th row in MAT$_j$ is v_i.

Corollary 1: For all fault-free nodes, their value MAJ$_i$ shall equal to the initial value of node i respectively for all i as if $m \le \lceil(n$-d-3)/2$\rceil$.

Proof: By Lemma 2, if the initial value of node i is v, then the majority value MAJ$_i$ = majority value in (v_{i1}, v_{i2},…, , v_{in}) at the i-th row of MAT$_i$, $1 \le i,j \le$n, always be v regardless the fault-free condition of TM$_{ij}$. The Corollary 1 is proved.

Corollary 2: For all fault-free nodes, the value M_{jk} of temp_FDMAT shall equal to the value MAJ$_j$ which is received from node j after the first round of message exchange by nodes j for $1 \le j,k \le n$ as if $m \le \lceil(n$-d-3)/2$\rceil$.

Proof: By Corollary 1, every node uses a value v to start the protocol and to make the value MAJ$_i$ equal to the initial value v of node i respectively for all i. The initial value is the value before starting the first round. At the second round, each node uses n values v_1, v_2, …, v_n to initiate the protocol simultaneously. That means the second round can be treated as n times of running the first round of message exchange, and the initial value v for each node is replaced by v_1, v_2, …, v_n respectively. By the same reasons of Corollary 1, the value M_{jk} of temp_FDMAT shall equal to the value v_j which is received from all nodes k respectively for $1 \le j,k \le n$ as if $m \le \lceil(n$-d-3)/2$\rceil$.

Theorem 1: At the phase of reaching consensus, the protocol FDC can make all fault-free nodes reach a consensus as if $m \leq \lceil (n-d-3)/2 \rceil$.

Proof:

(1) **Consensus**: If a fault-free node computes a majority value of vector $[v_1, v_2, \ldots, v_n]$, then all fault-free nodes compute the same majority value.
By Corollary 1, all fault-free nodes compute and treat the same value MAJ_i to be the initial value of node i respectively for all i as if $m \leq \lceil (n-d-3)/2 \rceil$ after two rounds of message exchanges. By the protocol FDC, all fault-free nodes agree on the majority value of vector $[v_1, \ldots, v_i, \ldots, v_n]$ = majority value of vector $[MAJ_1, ..., MAJ_i, ..., MAJ_n]$, then all fault-free nodes compute the same majority value due to every MAJ_i computed by every fault-free node is the same.

(2) **Validity**: If the initial value of node i is v_i then the i-th element to be agreed on in the common value should be v_i.
By Corllary 1, for all fault-free nodes, their value MAJ_i shall equal to the initial value v of node i respectively for all i as if $m \leq \lceil (n-d-3)/2 \rceil$. Due to the value MAJ_i is the i-th element of the vector $[MAJ_{1,...,} MAJ_{i,...,}MAJ_n]$ which are computed by all fault-free nodes, and then the common value should be the majority value of vector $[MAJ_{1,...,} MAJ_{i,...,}MAJ_n]$.

Lemma 3: All dormant TMs can be detected and located by all fault-free nodes.

Proof: By Corollary 1, the value MAJ_i of $[MAJ_{1,...,} MAJ_{i,...,}MAJ_n]$ is the initial value of node i respectively for every fault-free node i as if $m \leq \lceil (n-d-3)/2 \rceil$. And by Corollary 2, the values M_{jk} of temp_FDMAT shall equal to the value received from node j after the first round of message exchange by nodes j for $1 \leq j,k \leq n$. Therefore, if the value M_{jk} of temp_FDMAT is λ implies the TM between node j and node k is in dormant fault.

Lemma4: All malicious TMs can be detected and located by all fault-free nodes.

Proof: By Corollary 1, the value MAJ_i of $[MAJ_{1,...,} MAJ_{i,...,}MAJ_n]$ is the initial value of node i respectively for every fault-free node i as if $m \leq \lceil (n-d-3)/2 \rceil$. And by Corollary 2, the values M_{jk} of temp_FDMAT shall equal to the value received from node j after the first round of message exchange by nodes j for $1 \leq j,k \leq n$. Therefore, all λ can be eliminated to compare each j-th layer of FDMAT with temp_FDMAT if the values in column y have different values from column y of $temp\_FDMAT_i$ where $1 \leq y \leq n$ that implies the TMs between the node j and node y is in malicious fault.

Theorem 2: FDC can make all fault-free nodes detect/locate a common set of faulty components if the components failed during the reaching of consensus.

Proof: By Lemma 3 and Lemma 4, Theorem 2 has been proved.

Theorem 3: FDC can solve the FDA problem.

Proof:

(1) **Agreement**: By Theorem 2, the constraint can be met by protocol FDC.

(2) **Fairness**: A faulty component will falsify a message at least at the first round or at second round, which will emerge as results of M_{jk} for $1 \leq j,k \leq n$. That means the faulty components shall be detected as fault-free by any fault-free node through comparing M_{jk}. On the other hand, if a fault-free component is falsely detected as faulty by any fault-free node that means the related M_{jk} is different with the original value of the fault-free component. It is contradicted with the definition of a fault-free component.

5. Conclusion

WSN is crucial for the future of IoT since they cover a wide application range essential for the IoT. They are a network of small, wireless, ad hoc sensor nodes also called motes, which are interconnected and deployed in an area of interest (e.g. home, forest, battlefield, etc.) [12]. Therefore, WSN can be used in a wide range of application scenarios, like military, healthcare, environment, home, etc.

To achieve high reliability in a WSN of IoT, a mechanism that allows a set of nodes to reach a common agreement, even in the presence of faulty components, is needed. Traditionally, the consensus problem was visited in a distributed system with the assumption of the fallible component is node only and failure type of faulty components is malicious only. Hence, some of the innocent nodes are treated as the faulty components [13] and the failure type of faulty components is malicious only is also not reasonable. And most of previous results cannot detect and locate the faulty components.

In this study, the consensus problem is revised on an unreliable network with dual failure mode, the proposed protocol can tolerate/detect/locate the faulty components in a WSN. That is, the protocol FDC can solve the consensus problem with dual failure mode in a fallible WSN. Using FDC, it only needs two rounds of message exchanges to reach the consensus and only needs one additional round (the third-round of messages exchange) to detect and locate maximum number of faulty TMs, $m \leq \lceil (n-d-3)/2 \rceil$, even the TMs with dual failure mode. Actually, the proposed protocol only costs the minimum rounds of message exchanges and tolerates/detects/locates maximum number of TMs by allowing dual failure mode in a WSN.

Acknowledgment

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

References

[1] Yang, K. 2014. Wireless sensor networks. *Principles, Design and Applications*, Springer-Verlag, London.

[2] Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M., and Polakos, P. 2016. Wireless sensor network virtualization: A survey. *IEEE Communications Surveys & Tutorials*, 18, 1: 553-576.

[3] Kouche, A. E. 2012. Towards a wireless sensor network platform for the Internet of Things: Sprouts WSN platform. In *IEEE international conference on communications*, 10-15 June, 632-636.

[4] Sangdeh, P. K., Mirmohseni, M., and Poursabzi, F. 2015. Applying the Byzantine agreement in wireless sensor networks based on clustering. In *IEEE 23rd Iranian Conference on Electrical Engineering*, 10-14 May, 619-624.

[5] Li, X., Chen, X., and Xie, Y. 2015. Agreement of networks of discrete-time agents with mixed dynamics and time delays. *Mathematical Problems in Engineering*, 2015, http://dx.doi.org/10.1155/2015/957028.

[6] Lamport, L., Shostak, R., and Pease, M. 1982. The Byzantine generals problem. *ACM Trans. Programming Language Systems*, 4, 3: 382-401.

[7] Meyer, F. J. and Pradhan, D. K. 1991. Consensus with dual failure modes. *IEEE Transactions on Parallel and Distributed Systems*, 2, 2: 214-222.

[8] Fischer, M. and Lynch, N. 1982. A lower bound for the assure interactive consistency. *Information Processing Letters*, 14, 4: 183-186.

[9] Hsiao, H. S., Chin, Y. H., and Yang, W. P. 2000. Reaching fault diagnosis agreement under a hybrid fault model. *IEEE Transactions on Computers*, 49, 9: 980-986.

[10] Wang, S .C., Chin, Y. H., and Yan, K. Q. 1990. Reaching a fault detection agreement. *In International Conference on Parallel Processing*, August 13-17, 251-258.

[11] Căilean, A. M., Cagneau, B., Chassagne, L., Dimian, M., and Popa, V. 2015. Novel receiver sensor for visible light communications in automotive applications. *IEEE Sensors Journal*, 15, 8: 4632-4639.

[12] Farash, M. S., Turkanović, M., Kumari, S., and Hölbl, M. 2016. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Networks, 36*: 152-176.

[13] Abraham, I., Devadas, S., Nayak, K., and Ren, L. 2017. Brief announcement: Practical synchronous byzantine consensus. In *31st International Symposium on Distributed Computing*, October 16-20, 41:1-41:4.