Sensors and Materials, Vol. 29, No. 7 (2017) 1043–1048
MYU Tokyo

S & M 1396

Fault Diagnosis Agreement in Wireless Sensor Network

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

Chaoyang University of Technology, 168 Jifeng E. Rd., Wufeng, Taichung 41349, Taiwan ROC

(Received March 21, 2016; accepted May 15, 2017)

Keywords: fault tolerance, fault detection, fault diagnosis, wireless sensor network

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensor nodes in a wide range of applications. However, the reliability of a WSN is an important issue in the support of the applications. To achieve this goal, the fault tolerance of WSNs must be studied. For underlying malicious failure characteristics on nodes in a WSN, a fault diagnosis agreement (FDA) is proposed to make each fault-free node detect/locate a common set of faulty nodes by collecting the accumulated messages. However, the proposed protocol can cause each fault-free node to obtain a maximum number of a common set of tolerable faulty nodes. Therefore, the proposed protocol can enlarge the fault tolerance capability by allowing malicious faults to exist in a WSN.

1. Introduction

A wireless sensor network (WSN) is a distributed system that comprises thousands of sensor nodes and sinks.[1] A group of sensor nodes in a WSN cooperates to achieve some objectives; each sensor node communicates with other sensor nodes by broadcasting via a WSN, but this leads to severe problems, such as broadcast storms.[2] Many researchers have proposed cluster schemes and limited broadcasting to prevent broadcast storms.[2,3] Researchers have also proposed cluster-based wireless sensor networks (CWSNs) to diminish broadcast storms.[3] In a CWSN, each cluster is composed of many sensor nodes and one cluster head.

In a CWSN, the sensor nodes are interconnected via the wireless; the network is assumed to be reliable and synchronous.[4] If certain sensor nodes in a distributed system were to fail, the faulty nodes in the distributed system must be isolated so that the systems still can operate correctly. The fault diagnosis agreement (FDA) problem[5] is one of the most fundamental problems associated with reaching fault agreements in distributed systems.

The goal of solving the FDA problem is to make each fault-free node able to detect/locate the faulty components in the distributed system.[5] After reaching an FDA, each fault-free node can maintain the performance and integrity of the distributed system to provide a stable environment. Protocols designed to solve the FDA problem should meet the following requirements.[5]

Agreement: All fault-free nodes should be able to identify the common set of faulty nodes.

Fairness: No fault-free node is mistakenly identified as faulty by any other fault-free node.

However, a node is said to be fault-free if it follows protocol specifications during the execution of a protocol; otherwise, the node said to be faulty. In a CWSN, the symptom of a faulty node is usually unrestrained, and such behavior is commonly called a malicious fault.[6] In such a fault,

*Corresponding author: e-mail: kqyan@cyut.edu.tw
**Corresponding author: e-mail: sswang@cyut.edu.tw
http://dx.doi.org/10.18494/SAM.2017.1602

a node can withhold a message to be sent and instead send an irregular message or collide with other faulty nodes. A malicious fault is unpredictable, and the behaviors of other failure types can be treated as special cases of a fault. However, if the issues of the malicious fault, which is the most problematic fault, can be solved, then the other fault types[1] can surely also be managed. Therefore, malicious faulty nodes (MFNs) in CWSN are considered in this study. If a common agreement can be reached in the presence of a malicious fault, then it can also be reached in the presence of other failure modes.

In this study, a new protocol called malicious fault diagnosis agreement (MFDA) is proposed to solve the FDA problem in a CWSN. An MFDA can collect messages and then detect/locate the common set of MFNs by examining the collected messages.

2. Research Method

2.1 Concept of the MFDA protocol

The MFDA protocol is used to solve the FDA problem using evidence gathered from the optimal malicious agreement protocol (OMAP) in a CWSN proposed by Wang *et al.*[4] (OMAP is presented in the Appendix). There are three phases in the MFDA: the message collection phase, the fault diagnosis phase, and the reconfiguration phase. The message collection phase is used to collect all ic-trees (information collect trees) of nodes. The ic-tree is a tree structure used to store a received message without repeated cluster names.[4] The fault diagnosis phase is used to detect/locate the malicious faulty components. The reconfiguration phase is used to reconfigure the network. The protocol for an MFDA is shown in Fig. 1.

In the message collection phase, each node collects all ic-trees of nodes in the OMAP as evidence and then the IC-tree, a set of ic-trees of each node, is formed. Hence, in the MFDA, each node distributes its ic-tree to all nodes by executing the OMAP with its ic-tree as the initial value.

In the fault diagnosis phase, the collected IC-trees are examined to detect/locate the MFNs. The sets of malicious faulty cluster (MFC) and MFN are used to record the MFCs and MFNs, and the examination sequence by each fault-free node is top down and level by level.

Protocol MFDA (for each node in a CWSN)

Message Collection Phase:

Step1: Each node distributes its ic-tree to all the nodes by executing OMAP with its ic-tree as the initial value.

Step2: Then each node stores the other nodes' ic-trees to construct the set of IC-tree =[ic-tree$_1$, ic-tree$_2$, …, ic-tree$_c$]. Then, each fault-free node constructs the same set of IC-tree.

Fault Diagnosis Phase:

Set Malicious Faulty Clusters MFC=Null;
Set Malicious Faulty Nodes MFN=Null;
MFC = MFC ∪ {malicious faulty clusters}
MFN = MFN ∪ {malicious faulty nodes}

Step 1: **Detect/locate the malicious faulty cluster.**
Examine each MAJ value at the same labeled vertex of C_i of the common set of IC-tree. If the number of the most common value is not greater then c-⌊(c–1)/3⌋, then C_i is in malicious faulty cluster. Set MFC = MFC ∪ {C_i}.

Step 2: **Fault diagnosis with source node n_s.**
Examine the values at the roots of the IC-tree. If the number of most common root value is not greater then c-⌊(c–1)/3⌋, then n_s is in malicious fault, set MFN = MFN ∪ {n_s}.

Step 3: **Detect/locate the malicious faulty nodes.**
Examine each n_j value at the same labeled vertex of the common set of IC-tree. If the number of the most common value is not greater then c-⌊(c–1)/3⌋, then the node n_j is in malicious faulty node. Set MFN = MFN ∪ {n_j}.

Reconfiguration Phase:

According to MFC and MFN, the system can isolate the faulty components logically.

Fig. 1. The proposed protocol for MFDA.

(1) Detect/Locate the MFCs
The system examines each majority value (MAJ) at the same labeled vertex of C_i of the common set of IC-trees. Because of the constraint of $f_{mc} \leq \lfloor (c-1)/3 \rfloor$, where f_{mc} is the total number of allowable MFCs and c is the total number of clusters in the CWSN, if the most common value does not appear more than $c - \lfloor (c-1)/3 \rfloor$ times, then cluster C_i is a MFC, and the system sets MFC = MFC $\cup$ $\{C_i\}$.

(2) Fault diagnosis with source node n_s
The system examines all the values at the roots of the IC-tree. If the most common root value does not show up more than $c - \lfloor (c-1)/3 \rfloor$ times, then n_s is a MFN, and the system sets MFN = MFN $\cup$ $\{n_s\}$.

(3) Detect/Locate the MFNs
The system examines each n_j value at the same labeled vertex of the common set of the IC-tree. If the most common value appears more than $c - \lfloor (c-1)/3 \rfloor$ times, then node n_j is a MFN, and the system sets MFN = MFN $\cup$ $\{n_j\}$. The results of MFC and MFN from the fault diagnosis phase are used to reconfigure the network by isolating the faulty nodes logically in the reconfiguration phase. After reconfiguration, the performance and integrity of the network can be guaranteed.

2.2 Executing an MFDA

The proposed MFDA protocol is based on the OMAP agreement protocol proposed by Wang *et al.*[4] The MFDA collects all the nodes' ic-trees as evidence in the message collection phase. Figure 2 shows an example of a CWSN. Each node distributes its ic-tree to all the nodes in the message collection phase. Then each fault-free node constructs the common set of IC-tree as [ic-tree$_1$, ic-tree$_2$, … , ic-tree$_7$].

Figure 3 shows an example of ic-tree$_1$ and ic-tree$_2$ from the nodes of clusters C_1 and C_2. In the fault diagnosis phase, each fault-free node can detect/locate the common set of faulty components. By Step 1, each fault-free node can detect/locate MFCs. For example, the MAJ values at the vertex $s7$ are (0, 1, 1, 1, 0, 0, 1). The most common value does not appear more than $c - \lfloor (c-1)/3 \rfloor = 7 - 2 = 5$ times. Therefore, C_7 is a MFC. The system sets MFG = MFG $\cup$ $\{C_7\}$. By Step 2, the root values of the IC-tree are (0, 0, 0, 1, 1, 0, 1). The number of the most common root value is not greater then $c - \lfloor (c-1)/3 \rfloor = 7 - 2 = 5$. Therefore, n_s is a MFN. The system sets MFN = MFN $\cup$ $\{n_s\}$. By Step

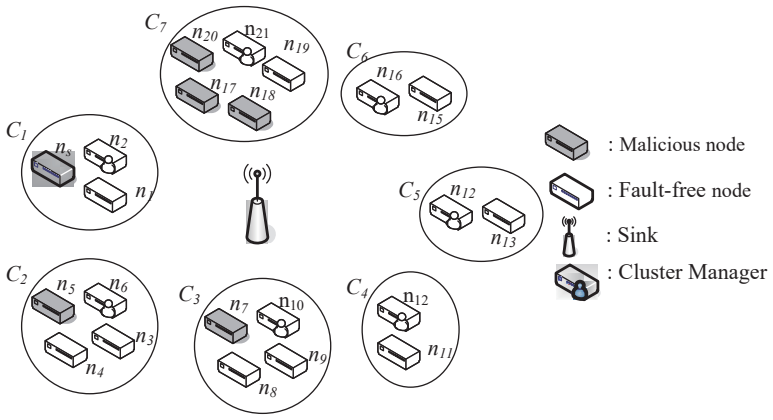

Fig. 2. An example of CWSN.

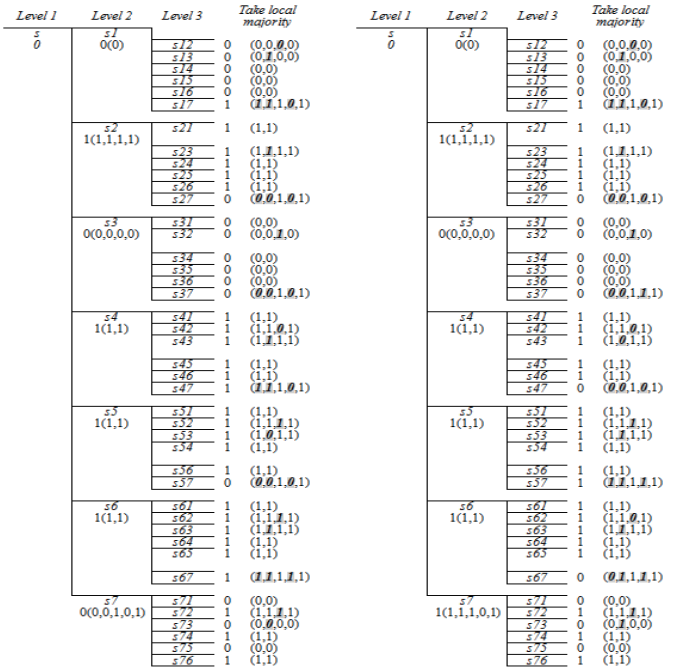

Fig. 3. The common set of IC-tree by each fault-free node. (a) The ic-tree$_1$ from C_1's node. (b) The ic-tree$_2$ from C_2's node.

3, the values of n_5 at the vertex $s7$ are (0, 1, 0, 1, 0, 1, 1). The most common value does not appear more than $c - \lfloor (c-1)/3 \rfloor = 7 - 2 = 5$ times. Therefore, the node n_5 is a MFN, and MFN = MFN $\cup$ $\{n_5\}$. Following all the steps in the fault diagnosis phase, the MFC C_7 and the MFNs n_s, n_5, n_7, n_{17}, n_{18}, and n_{20} can be detected/located by each fault-free node. Finally, in the reconfiguration phase, each fault-free node isolates n_s, n_5, n_7, n_{17}, n_{18}, and n_{20} logically to reconfigure the network, as shown in Fig. 4.

3. Correctness and Complexity of MFDA

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

Lemma 1. Each fault-free node receives the same common set of IC-tree as evidence in the message collection phase if $f_{mc} \le \lfloor (c-1)/3 \rfloor$.

Proof: The agreement protocol can make each fault-free node agree on a single common value regardless of whether the source node is fault-free or not.[6] Hence, each node can reliably distribute its ic-tree to all the other nodes by executing an OMAP with its ic-tree as the initial value. Hence, each fault-free node can receive the same common set of IC-tree.

Lemma 2. Each fault-free node can detect/locate the same faulty components.

Proof: Each fault-free node receives the same evidence owing to Lemma 1 and uses the same FDA protocol, namely MFDA, so each fault-free node will accurately detect/locate the same faulty components.

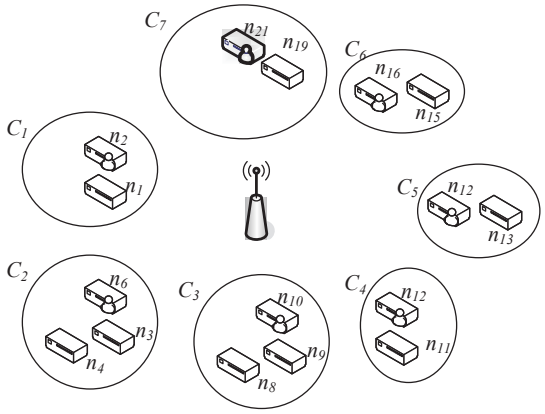

Fig. 4. A CWSN after reconfiguration ($n = 16$, $c = 7$).

Theorem 1. MFDA Protocol satisfies the agreement of FDA.
Proof: By Lemmas 1 and 2, all the fault-free nodes identify the common set of faulty nodes.
Lemma 3. MFNs/MFCs can be detected/located if $f_{mc} \leq \lfloor (c-1)/3 \rfloor$.
Proof: Because of the constraint $f_{mc} \leq \lfloor (c-1)/3 \rfloor$, there are, at most, f_{mc} MFCs, so there are, at most, f_{mc} values at the same labeled vertex in the IC-tree different from the most common value, that is, $f_{mc} \leq \lfloor (c-1)/3 \rfloor$. If the most common value does not appear at the same labeled vertex in the IC-tree more than $c - f_{mc} - 1$ times, then the component is a malicious fault.
Theorem 2. MFDA Protocol satisfies the fairness requirement of FDA.
Proof: By Lemma 3, no fault-free node is falsely detected as faulty by any fault-free nodes if $f_{mc} \leq \lfloor (c-1)/3 \rfloor$.
Theorem 3. MFDA Protocol solves the FDA problem in a CWSN if $f_{mc} \leq \lfloor (c-1)/3 \rfloor$.
Proof: By theorem 1 and theorem 2, this theorem is proved.
Theorem 4. The maximum number of detectable/locatable faulty components by MFDA is f_{mc} MFCs, where $f_{mc} \leq \lfloor (c-1)/3 \rfloor$.
Proof: Fischer *et al.* indicated the constraints of the agreement problem for node faults only is $f \leq \lfloor (c-1)/3 \rfloor$ and the unit is one node, where f is the total number of allowable MFNs and n is the total number of nodes in a fully connected network.[7] However, the unit of CWSN is the cluster, so we can consider a node in Fischer *et al.*'s study as a cluster in CWSN. Therefore, $f \leq \lfloor (c-1)/3 \rfloor$ in Fischer *et al.*'s study[7] can be applied to $f_{mc} \leq \lfloor (c-1)/3 \rfloor$, where f_{mc} is the total number of allowable MFCs and c is the total number of clusters in a CWSN. The total number of detectable/locatable faulty components by MFDA is f_{mc} MFCs, which is the maximum if $f_{mc} \leq \lfloor (c-1)/3 \rfloor$.

4. Conclusions

In this study, the proposed FDA protocol, MFDA, can detect/locate the maximum number of MFNs in a CWSN. The proposed MFDA can not only reach an agreement but also detect and locate the faulty components in an unreliable CWSN. Therefore, the proposed protocol can enlarge the fault tolerance capability by identifying malicious faults that exist in a network. That is, MFDA can tolerate, detect, and locate the maximum number of faulty nodes with a malicious failure mode to solve the fault diagnosis agreement problem in a CWSN by a minimum number of rounds of message exchanges.

References

1 J. Zhang and V. Varadharajan: J. Netw. Comput. Appl. **33** (2010) 63.
2 S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris: Algorithms **6** (2013) 29.
3 M. Sasikumar and R. Anitha: Int. J. Innov. Res. Adv. Eng. **1** (2014) 197.
4 S. C. Wang, K. Q. Yan, C. L. Ho, and S. S. Wang: Int. J. Adv. Inf. Technol. **8** (2014) 1.
5 M. L. Chiang and H. C. Hsieh: Inf. Technol. Control. **41** (2012) 151.
6 L. Lamport, R. Shostak, and M. Pease: ACM Trans. Program. Lang. Syst. **4** (1982) 382.
7 M. Fischer, M. Paterson, and N. Lynch: J. ACM. **32** (1985) 374.

Appendix

The OMAP is organized in two phases, the message gathering phase and agreement making phase. In the message gathering phase, each node collects sufficient information from other nodes in the CWSN. In the agreement making phase, the information collected in the message gathering phase is used to decide the agreement value. The OMAP is presented in Fig. A1.

OMAP (Source node with initial value v_s)

Definitions:

1. For the CWSN, each sensor node has the common knowledge of entire graphic information $G = (E, C)$, where C is the set of clusters in the CWSN and E is a set of cluster pairs (C_x, C_y) indicating a communication medium (the sensing is covered) between cluster C_x and cluster C_y.
2. Each sensor node can communicate with all other sensor nodes.
3. The sensor node plays sender or receiver depending on the behaviour of the transmission.
4. The sensor node cannot garble the message between the sender node and receiver node; this assumption has been achieved by the technology of encryption (such as RSA [14]).

Pre-Execute.

Computes the number of rounds required, $\sigma = \lfloor (N-1)/3 \rfloor + 1$, where N is the total number of clusters in the CWSN.

***Message Gathering Phase*:**

Case* $\sigma = 1$, *run

A) The source node transmits its initial value v_s to each cluster's nodes.

B) Each receiver node obtains the value and stores it in the root of its mg-tree.

Case* $\sigma > 1$, *run

A) Each node without the source node transmits the values at level $\sigma - 1$ in its mg-tree to each cluster's nodes.

B) Each receiver node takes the local majority value on the received values from the same cluster and stores the majority single value in the corresponding vertices at level σ of its mg-tree.

***Agreement Making Phase*:**

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

Step 2: Using function VOTE on the root s of each node's ic-tree, then the common value VOTE(s) is obtained.

Function VOTE(μ)

If the μ is a leaf, then output the value μ.
Else if the majority value does not exist, then output the majority value ϕ.
Otherwise, output the majority m, where $m \in \{0, 1\}$

Fig. A1. The OMAP protocol.[4]