RESEARCH ARTICLE
doi: 10.2306/scienceasia1513-1874.2014.40S.008

ScienceAsia 40S (2014): **8–15**

Optimal malicious agreement protocol for cluster-based wireless sensor networks

Kuo-Qin Yan[a], **Shu-Ching Wang**[b,*], **Chin-Shan Peng**[b], **Shun-Sheng Wang**[c]

[a] Department of Business Administration, Chaoyang University of Technology, Taiwan
[b] Department of Information Management, Chaoyang University of Technology, Taiwan
[c] Department of Industrial Engineering and Management, Chaoyang University of Technology, Taiwan

*Corresponding author, e-mail: scwang@cyut.edu.tw

Received 13 Feb 2014
Accepted 9 Jun 2014

ABSTRACT: A wireless sensor network (WSN) is a distributed system that comprises thousands of sensor nodes and a sink. The sensor nodes communicate with other sensor nodes by using broadcast in the WSN and this leads to a severe problem. Hence many researchers proposed cluster schemes to prevent the broadcast storm. Cluster-based wireless sensor networks (CWSNs) have been studied more recently. In order to increase the capability of the fault-tolerance and reliability of CWSNs, the Byzantine agreement problem is revisited. A protocol for achieving the task of agreement between nodes in a CWSN is proposed in this paper. The proposed protocol is referred to as the optimal agreement protocol and is demonstrated to make each healthy node reach an agreement value to cope with the influence from faulty components in the CWSN.

KEYWORDS: Byzantine agreement, distributed system, fault tolerance

INTRODUCTION

One new concept associated with the 'future internet' is 'internet of things' (IoT). The IoT describes a vision where objects become part of the internet, where every object is uniquely identified, and accessible to the network, its position and status known, and where services and intelligence are added to this expanded internet, fusing the digital and physical world, ultimately impacting our professional, personal and social environments[1]. Through the study of the IoT technology, the idea of combining IoT with wireless sensor networks (WSNs) is proposed[1]. In order to improve the efficiency of logistics enterprises, all aspects in the transport process should be monitored, which requires applying the IoT technology into the logistics management system.

Recently, the advancement of technology and the rapid development of micro electro mechanical systems have facilitated the rapid growth of WSNs[2,3]. A WSN consists of spatially distributed autonomous devices which use sensor nodes to monitor physical or environmental conditions cooperatively. However, the sensor node is limited by the energy resource, memory, computation, communication capability, etc[2]. Hence the topology of a Cluster-based Wireless Sensor Network (CWSN) has been proposed to prolong the lifetime of WSNs by decreasing the energy consumption of nodes[4,5].

The reliability of the node is one of the most important requirements of a successful CWSN. To ensure that CWSNs exist in a reliable environment, it is necessary to create a mechanism which allows a set of nodes to reach an agreed value. The Byzantine Agreement (BA) problem[6,7] is one of the most fundamental problems in which an agreement value is reached in a distributed system. The traditional BA problem first defined by Lamport et al[7] makes the following assumptions.

(i) There are n nodes in a synchronous distributed system where n is a constant and $n \geqslant 4$.
(ii) Each node can communicate with the others through a reliable fully connected network.
(iii) One or more of the nodes might fail, so the faulty nodes may transmit unhealthy message(s) to other nodes.
(iv) After message exchanges, all healthy nodes should reach a common agreement, if and only if the number of faulty nodes f_n is less than one-third of the total number of nodes in the network ($f_n \leqslant \lfloor (n-1)/3 \rfloor$).
(v) Only the faulty nodes are considered.

Based on these assumptions, the BA requirement can be satisfied when the following constraints are met:

Agreement: All healthy nodes agree on a common decision value.

Validity: If the source node is healthy, then all healthy nodes agree on the initial value sent by the

source node.

Previous research on the BA problem was solved in a well-defined network environment, such as fully connected network, broadcast network and so on[6–8]. In other words, all nodes reside in a wired network environment. However, the technology of networks continues to grow at a high speed and the applications in wireless networks have reached an astonishing achievement level in the last year, so it is important to solve the BA problem in the wireless networks. Thus this study will focus on the wireless networks and propose a protocol to make all healthy nodes reach an agreement in the CWSN. The proposed protocol is referred to as the Optimal Agreement Protocol (OAP), which can lead to an agreement between each healthy node in the CWSN. However, the proposed protocol OAP is the only protocol to make all healthy nodes reach BA in the case of CWSN with both node and transmission medium (TM) fallible.

THE TOPOLOGY OF CWSN

In previous literature, most protocols of the BA problem[6,7] perform well in wired networks. Recent advances in technology have provided portable nodes with wireless interfaces that allow network communication among mobile users. The computing environment, which refers to as mobile computing, no longer requires users to maintain a fixed and universally known position in the network and enables almost unrestricted mobility. The topology of WSN is a type of wireless network topology, and so previous protocols may not be well suited to it. WSN is made up of several clusters of sensors[4]. Each cluster is composed of many sensor nodes and one cluster head. The sink controls the state and communication data of all cluster heads. Fig. 1 is a topology of CWSN.

In the CWSN, messages are always received by receiving nodes within a fixed time period; otherwise, the message's sender is treated as a failure[7]. If certain components in a distributed system fail, a protocol is required to ensure that the system still functions correctly. However, network components may not always work well.

In a BA problem, many cases are based on the assumption of node failure in a fail-safe network[7,9]. Based on this assumption, a TM fault is treated as a node fault, whatever the correctness of an innocent node, so that an innocent node does not involve agreement[10]. Nevertheless, the definition of a BA problem requires all healthy nodes to reach an agreement.

A component is said to be healthy if it follows protocol specifications during the execution of a protocol; otherwise, the component is said to be faulty. However, the behaviour of a maliciously faulty node/TM is unpredictable and arbitrary[7]. Maliciously faulty nodes/TM may fake, lose, or mangle messages. Thus it is the most extreme failure type and causes the serious problems. Herein, a solution to the BA problem in the nodes/TM fallible CWSN with malicious is presented.

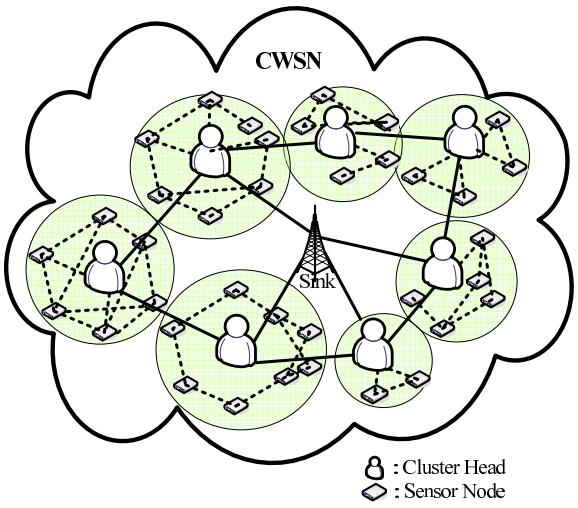

Fig. 1 The topology of CWSN.

THE AGREEMENT PROTOCOL

In this study, the BA problem is discussed in relation to the CWSN with malicious. As such, nodes executing the new protocol should receive messages from other nodes within a predictable period of time. If the messages are not received on time, then they must have been influenced by faulty components. In this study, a new protocol, the Optimal Agreement Protocol (OAP), is proposed to solve the BA problem when resulting from faulty component(s) which may send incorrect messages to prevent the system from reaching agreement in the CWSN. The notation and assumptions of the OAP for the CWSN are shown below:

(i) Each node in the network can be identified uniquely.
(ii) A node does not know the fault status of other components.
(iii) Let n be the number of nodes in the underlying network.
(iv) Let C be the number of clusters in the underlying network and $C \geqslant 4$.
(v) Let C_i be the cluster identifier where $1 \leqslant i \leqslant C$ and $C \geqslant 4$.
(vi) Let n_i be the number of nodes in cluster C_i, $0 \leqslant i \leqslant C$. If there are at least $\lceil n_i/2 \rceil$ malicious

faulty nodes in C_i, then C_i is the malicious faulty cluster.

(vii) Let T_{FT} be the total number of malicious faulty TMs in the CWSN.

(viii) Let T_{FC} be the total number of malicious faulty clusters in the CWSN.

(ix) Let T_{FN} be the total number of malicious faulty nodes in the CWSN.

(x) Let Con be the connectivity of the CWSN, where $\mathrm{Con} > 2(T_{\mathrm{FC}} + T_{\mathrm{FT}})$.

The proposed OAP is used to make each healthy node in the network reach common agreement. In order to reach a common agreement, each node should exchange messages with all other nodes. Then, each healthy node collects enough messages to determine the decision value, i.e., the agreement value; each healthy node collects agreement value should be identical.

The first step of OAP is to determine the number of rounds of messages initially exchanged so that the number of exchanges required can be minimized. After determining the required rounds of message exchange, OAP must execute two phases of work: the message exchange phase and the decision making phase. The task of the message exchange phase is to collect messages from other nodes. Furthermore, the influence of a faulty TM is removed during the message exchange phase by applying a majority function, called the MAJ. Afterwards, in the decision making phase, each healthy node uses the messages received during the message exchange phase to determine the common agreement value.

The above mentioned is the basic concept underlying the proposed protocol. In a generalized case, both nodes and TMs may become faulty simultaneously. For all healthy nodes to reach a common agreement value, the faults caused by the nodes and the TMs must be removed. The protocol proposed by Yan et al[11] noted that the influence of the malicious faulty TMs must be eliminated first, and only then can the influence of the malicious faulty nodes be removed. Hence the basic strategy of the proposed OAP in a malicious faulty CWSN to solve the BA problem is to remove the influence of the malicious faulty TMs first, and then remove the influence of the malicious faulty nodes. In addition, the number of required rounds is only $T_{\mathrm{FC}} + 2$ rounds of message exchange, where $T_{\mathrm{FC}} = \lfloor (C-1)/3 \rfloor$, C is the total number of clusters, and $T_{\mathrm{FC}} + 2$ is the minimum number of rounds. In other words, if the faulty components are nodes and TMs, then OAP can make all healthy nodes in the CWSN reach a common agreement while requiring minimum rounds of message exchange, and can tolerate the maximum number of faulty components. The OAP is explained in the following subsections.

There are two phases in the OAP: the message exchange phase and the decision making phase. The influence of malicious faulty TMs can be mitigated during the message exchange phase. The influence of malicious nodes can be mitigated during the decision making phase.

By using OAP, any pairs of nodes can remove the influences of malicious faulty TMs in each round of message exchange, if $C > \lfloor (C-1)/3 \rfloor + 2(T_{\mathrm{FC}} + T_{\mathrm{FT}})$. This is because the healthy sending node can send Con copies of a message to healthy receiving nodes. In the worst case, a healthy receiver node can receive $\mathrm{Con} > 2(T_{\mathrm{FC}}+T_{\mathrm{FT}})$ messages transmitted by the healthy sender node. In this case, a healthy destination node can decide which the healthy messages by taking the majority value.

In order to remove the influence of the malicious faulty nodes, Bar-Noy et al[8] proved that each node must execute $\lfloor (n-1)/3 \rfloor + 1$ rounds of message exchange where n is the number of nodes in the underlying network, but the result cannot be directly applied to BA in a CWSN. This is because the exchanged messages of the generalized case include the influence of malicious faulty TMs and nodes. However, when the proposed protocol attempts to remove the influence of the malicious faulty TMs, the MAJ function must be applied to the received messages after two rounds of the message exchange phase. Hence each healthy node must execute $\lfloor (C-1)/3 \rfloor + 2$ rounds of message exchange in a C-cluster CWSN. The progression of the degree of influence of the faulty TMs/nodes removed is discussed as follows.

REMOVING THE INFLUENCE OF MALICIOUS FAULTY TMs

During the message exchange phase, each node collects sufficient messages from other nodes. Thus these received messages can be used to mitigate the influence of malicious faulty TMs. In this phase, a data structure called a ms-tree is used during the message exchange. The ms-tree is a tree structure that is used to store the received messages. First, requiring only two rounds of message exchange, the OAP protocol can be used to remove the influence of malicious TMs. In the first round, the source node multi-casts its initial value versus through TMs. When a healthy node receives the message, it stores the received value, denoted as $\mathrm{val}(s)$, in the root of its ms-tree. After the first round, each node multi-casts the root value of its ms-tree to each node. After two rounds, a

healthy value is obtained which is free of malicious faulty TMs. However, the computed value may still be influenced by a malicious faulty node. Hence OAP requires $T_{\mathrm{FC}}+2$ rounds of message exchange, where T_{FC} denotes the maximum number of allowable faulty clusters and cannot exceed $\lfloor (C-1)/3 \rfloor$, where C is the total number of clusters. However, in order to mitigate the influence of the faulty TMs, the message stored in the ms-tree and the function $\mathrm{MAJ}(a)$ must be applied during each round of message exchange, where a represents a vertex in a ms-tree, and the data reduction function $\mathrm{MAJ}(a)$ is the majority value in the set of $\{\mathrm{val}(aj) \mid 1 \leqslant j \leqslant n\}$, if it exists. Otherwise, the complement of $\mathrm{val}(a)$, denoted as $\neg\mathrm{val}(a)$ is chosen. There are at most $\lfloor C/2 \rfloor - 1$ faulty clusters. After the first round, the number $\neg v_s$ is no greater than $\lfloor C/2 \rfloor - 1$. If the majority value cannot be determined, the message sent during the last round is incorrect. Hence it will use the complement of $\mathrm{val}(a)$ as the majority value.

MITIGATING THE INFLUENCE OF MALICIOUS FAULTY NODES

After finishing $T_{\mathrm{FC}}+2$ rounds of the message exchange phase, each node will execute the decision making phase. In order to mitigate the influence of malicious faulty nodes and avoid the repetition of faulty nodes, no cluster name is repeated in any vertex and each healthy node must reorganize the ms-tree into a corresponding ic-tree. This is performed by using the reorganization rules.

(1) The leaves in level $T_{\mathrm{FC}}+2$ of the ms-tree are deleted.

(2) The vertices with repeated cluster's names are deleted.

Subsequently, all healthy nodes must use function $\mathrm{VOTE}(\alpha)$ to remove the faulty influence of malicious faulty nodes and to obtain a common value.

However, in the first round of the message exchange phase, the source node sends its initial value to all nodes, and then the receiver node stores the received value in the root s of its ms-tree. For $r > 1$ round, each receiver node takes a majority on its received messages from same cluster and stores in the corresponding vertices at level r of its ms-tree. Then each node applies MAJ on the level r of its ms-tree and stores the MAJ values in the corresponding vertices at level $r-1$ of its ms-tree to remove the influence of malicious faulty TM.

Subsequently, in the decision making phase, each node reorganizes its ms-tree into a corresponding ic-tree. Hence the common value $\mathrm{VOTE}(s)$ was obtained by using function VOTE on the root s of each node's ic-tree. The detailed definition of the OAP is shown in Fig. 2.

Protocol OAP (Source node with initial value v_S)

Compute the number of rounds required O =(*C*-1)/3+2

Message Exchange Phase:

r=1 do:

1) The source node transmits its initial value v_s to each cluster's nodes.
2) Each node stores v_s in the root *s* of its ms-tree.

For 1<*r*≤O Do:

1) Each node without the source node transmits the value at level *r*-1 in its ms-tree to each other and itself.
2) Each receiver node takes a majority on the received messages from same cluster and stores in the corresponding vertices at level of its ms-tree.
3) Each node applies MAJ on the level of its ms-tree and stores the MAJ values in the corresponding vertices at level *r*-1 of its ms-tree.

Decision Making Phase:

Step 1: The leaves in level O of each node's ms-tree are deleted.
Step 2: Each node's ms-tree is reorganized into a corresponding ic-tree by deleting the vertices with repeated cluster name.
Step 3: Function VOTE is applied to root *s* of each node's ic-tree. Then the common value VOTE(*s*) is obtained.

Function MAJ

1) The majority in the set of {val(*aj*)|1≤*j*≤*n*}, if it exists.
2) Otherwise, the complement of val(*a*) is chosen.

Function VOTE(*a*)

1) val(*a*), if the *a* is a leaf.
2) The majority value in the set of {VOTE(*ai*)|1≤*i*≤*C* and vertex *ai* is a child of vertex *a*}, if such a majority value exists.
3) A default value is chosen, otherwise.

Fig. 2 The proposed protocol OAP.

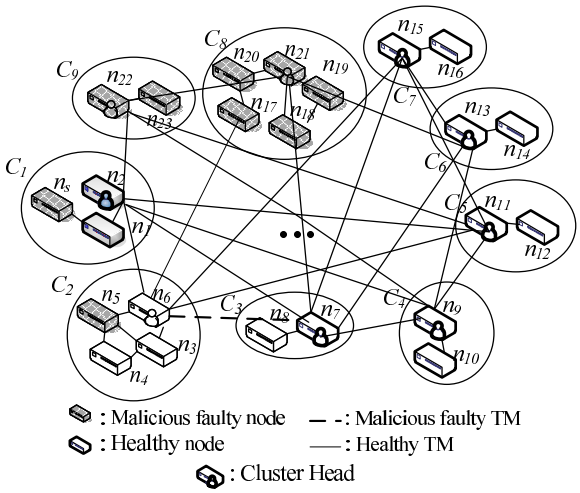

Fig. 3 An example of CWSN.

AN EXAMPLE OF EXECUTING OAP

An example for executing our protocol OAP is given. An example of CWSN is shown in Fig. 3. There are 24 nodes falling into nine clusters. C_1 includes source nodes n_s, n_1, and n_2. C_2 includes n_3, n_4, n_5, and n_6. C_3 includes n_7 and n_8. C_4 includes n_9 and n_{10}. C_5 includes n_{11} and n_{12}. C_6 includes n_{13} and n_{14}. C_7 includes n_{15} and n_{16}. Nodes n_{17}, n_{18}, n_{19}, n_{20}, and n_{21} belong to C_8. Nodes n_{22} and n_{23} belong to C_9.

	Level 1 (Root s)
C_1's healthy nodes	***0***
C_2's healthy nodes	***1***
C_3's healthy nodes	***0***
C_4's healthy nodes	***1***
C_5's healthy nodes	***1***
C_6's healthy nodes	***1***
C_7's healthy nodes	***1***
C_8's healthy nodes	***1***
C_9's healthy nodes	***0***

Fig. 4 The ms-tree of each node at the 1st round.

Level 1 Root s	Level 2	To take a majority on the Messages from same cluster received
Val(s)=1	*s1*	0 ←(0,0)
	s2	1 ←(1,1,***0***,1)
	s3	**1** ←(***1***,***1***)
	s4	1 ←(1,1)
	s5	1 ←(1,1)
	s6	1 ←(1,1)
	s7	1 ←(1,1)
	s8	***0*** ←(***0***,***0***,***0***,***0***)
	s9	***1*** ←(***1***,***1***)

Fig. 5 The ms-tree of healthy node n_3 at the 2nd round.

In the BA problem, the worst situation is that the source is no longer honest[9]. In this case, the source node n_s is in a malicious fault state, which means that n_s may arbitrarily send different values to different clusters. In order to solve the BA problem among healthy nodes in this example, OAP requires $O(\lfloor (C-1)/3 \rfloor + 2)$ rounds in the message exchange phase. In OAP, the number of rounds required before the message exchange phase is counted. In this example, four rounds of message exchange are required.

In the first round of the message exchange phase, the source node n_s sends value 0 to C_1, C_3, and C_9, sends value 1 to C_2, C_4, C_5, C_6, C_7, and C_8. The message obtained from each healthy node is listed in Fig. 4. In the r^{th} round, $1 < r \leqslant 0$, of message exchange, each node transmits the values at the $(r-1)^{\text{th}}$ level in its ms-tree to all the others and itself. Subsequently, each receiver node takes a majority on the received messages from same cluster and stores in the corresponding vertices at level r of its ms-tree. Then each node applies MAJ on the level r of its ms-tree and stores the MAJ values in the corresponding vertices at level $r-1$ of its ms-tree. The ms-tree of healthy node n_3 at the second, third, and final round in the message exchange phase is shown in Figs. 5, 6, and 7, and the message exchange

Level 1	Level 2	Level 3	←═══	
s (1)	s1 (0)	s11	0	(0,0)
		s12	0	(0,0,***1***,0)
		s13	***1***	(***1***,***1***)
		s14	0	(0,0)
		s15	0	(0,0)
		s16	0	(0,0)
		s17	0	(0,0)
		s18	***1***	(***0***,***1***,***1***,***1***,***1***)
		s19	***1***	(***1***,***1***)
	s2 (1)	s21	1	(1,1)
		s22	1	(1,1,***0***,1)
		s23	***0***	(***0***,***0***)
		s24	1	(1,1)
		s25	1	(1,1)
		s26	1	(1,1)
		s27	1	(1,1)
		s28	***0***	(***0***,***1***,***0***,***1***,***0***)
		s29	***0***	(***0***,***0***)
		⋮		⋮
	s3 (1)	⋮		⋮
	⋮	⋮		⋮

Fig. 6 The ms-tree of healthy node n_3 at the 3rd round.

Level 1	Level 2	Level 3	Level 4	←═══	
s (1)	s1 (0)	s11 (0)	s111	0	(0,0)
			s112	0	(0,0,***0***,0)
	⋮	⋮	s113	***0***	(***0***,***0***)
	⋮	⋮	s114	0	(0,0)
			s115	0	(0,0)
			s116	0	(0,0)
			s117	0	(0,0)
			s118	***1***	
	(***0***,***1***,***1***,***1***,***1***)				
			s119	***1***	(***1***,***1***)
		s12 (0)	s121	0	(0,0)
			s122	0	(0,0,***1***,0)
		⋮	s123	***1***	(***1***,***1***)
		⋮	s124	0	(0,0)
		⋮	s125	0	(0,0)
		⋮	s126	0	(0,0)
			s127	0	(0,0)
			s128	***0***	
	(***1***,***1***,***0***,***0***,***0***)				
			s129	***1***	(***1***,***1***)
			⋮		⋮
			⋮		⋮

Fig. 7 The final ms-tree of node n_3 after the message exchange phase.

phase is completed.

After the message exchange phase, the leaves in level O of each node's ms-tree are deleted and the tree structure of each healthy node has been converted from ms-tree to ic-tree by deleting the vertices with duplicated names (for example, s_{11} will be deleted) in the decision making phase. The example ic-tree is shown in Fig. 8. Finally, the function VOTE is used to determine the root s value for each healthy node's ic-tree

$$\text{VOTE}(s) = \text{VOTE}(s_1), \ldots, \text{VOTE}(s_9) = 1,$$

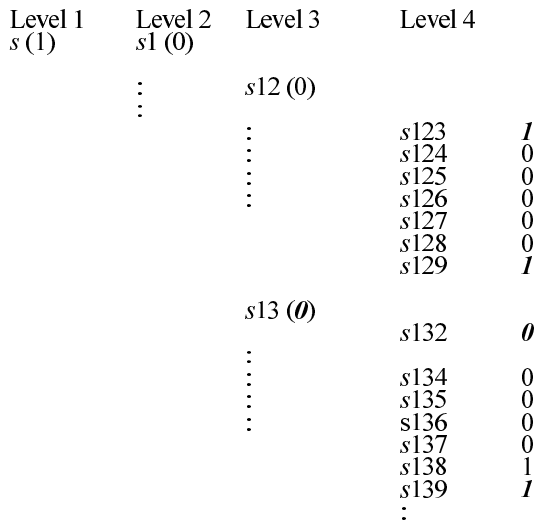

Fig. 8 The tree structure of n_3 is converted from ms-tree to ic-tree.

so that an agreement value 1 can be obtained, and the decision making phase is complete.

CORRECTNESS AND COMPLEXITY OF PROTOCOL

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

Correctness of OAP

In order to prove the correctness of the OAP, a vertex μ is called common[8] if each healthy node has the same value for μ. That is, if vertex μ is common, then the value stored in vertex μ of each healthy node's ms-tree or ic-tree is identical. When each healthy node has a common initial value of the source node in the root of an ic-tree, if the root s of an ic-tree in a healthy node is common and the initial value received from the source node is stored in the root of the tree structure, then an agreement is reached because the root is common. Thus the constraints $(\text{Agreement}')$ and $(\text{Validity}')$ can be rewritten as

$(\text{Agreement}')$: when root s is common;

$(\text{Validity}')$: when $\text{VOTE}(s) = v_s$ for each healthy node, if the source node is healthy.

To prove that a vertex is common, the term common frontier[7] is defined as follows: When every root-to-leaf path of the tree (a ms-tree or an ic-tree) contains a common vertex, the collection of common vertices forms a common frontier. In other words, every healthy node has the same messages collected in the common frontier if a common frontier exists in a healthy node's tree structure (ms-tree or ic-tree). Subsequently, using the same majority voting function to compute the root value of the tree structure, every healthy node can compute the same root value because they used the same input (the same collected messages in the common frontier). The same computing function will cause the same output (the root value).

Since OAP can solve the BA problem, the correctness of OAP should be examined in the following two ways:

(i) Healthy vertex: Vertex μ_i of a tree is a healthy vertex if cluster C_i (the last cluster name in vertex μ_i's cluster name list) is healthy. In other words, a healthy vertex is a place to store the value received from a healthy node.

(ii) True value: For a healthy vertex μ_i in the tree of a healthy node in the healthy cluster C_j, $\text{val}(\mu_i)$ is the true value of vertex μ_i. In other words, the stored value is called the true value.

By definition, a healthy vertex is one that contains a stored value that is received from the nodes in a healthy cluster, and a healthy cluster always transmits the same value to all nodes. Hence the healthy vertices of such a ms-tree are common. After reorganizing the ms-tree into its corresponding ic-tree by deleting the vertices with repeated cluster names, the values stored on the healthy vertices of an ic-tree shall be the same. As a result, all the healthy vertices of an ic-tree are also in common. Again, by the definition of a healthy vertex, a common frontier does exist in the ic-tree. Hence the $(\text{Agreement}')$ and $(\text{Validity}')$ are true no matter whether the source node is healthy or has failed if the BA problem has been solved.

Theorem 1 *The healthy destination node can receive the message(s) from a sender node without influence from any faulty components between the sender node and destination node if* $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$.

Proof: The influences of malicious faulty components between any pairs of nodes in each round of message exchange can be ruled out if $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$. The reason is that the healthy sender node sends Con copies of a message to healthy destination nodes. In the worst case, a healthy destination node can receive $\text{Con} - (T_{\text{FC}} + T_{\text{FT}})$ messages transmitted by the healthy sender node. Due to the fact that $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$, hence a healthy destination node can decide which the healthy messages are by taking the majority value. □

Lemma 1 *All healthy vertices of an ic-tree are common.*

Proof: After reorganization, no repeatable vertices exist in an ic-tree. At the level $\theta - 2$ or above, where

$\theta = \lfloor (C-1)/3 \rfloor + 2$, the healthy vertex μ has at least $2\theta - 3$ children in which at least $\theta - 1$ children are correct. The true value of these $\theta - 1$ healthy vertices is in common, and the majority value of vertex μ is common. The healthy vertex μ is common in the ic-tree, if the level of μ is less then $\theta - 1$. As a result, all healthy vertices of the ic-tree are common. □

Lemma 2 *A common frontier exists in the ic-tree.*

Proof: There are $\theta - 1$ vertices along each root-to-leaf path of an ic-tree in which the root is labelled by the source name, and the others are labelled by a sequence of cluster names. Since at most $\theta - 2$ clusters can fail, there is at least one vertex that is healthy along each root-to-leaf path of the ic-tree. Using Lemma 1, the healthy vertex is common, and the common frontier exists in each healthy node's ic-tree. □

Lemma 3 *Let μ be a vertex, and μ is common if there is a common frontier in the subtree rooted at μ.*

Proof: If the height of μ is 0, and the common frontier (μ itself) exists, then μ is common. If the height of μ is σ, the children of μ are all consistent using the induction hypothesis with the height of the children at $\sigma - 1$, the vertex μ is then common. □

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

Theorem 2 *The root of a healthy node's ic-tree is common.*

Proof: By Lemma 1, Lemma 2, Lemma 3 and Corollary 1, the theorem is proved. □

Theorem 3 *Protocol OAP solves the BA problem in a CWSN.*

Proof: To prove the theorem, it must be shown that OAP meets the agreements $(\text{Agreement}')$ and $(\text{Validity}')$;

$(\text{Agreement}')$: Root s is common by Theorem 2, therefore, $(\text{Agreement}')$ is satisfied;

$(\text{Validity}')$: $\text{VOTE}(s) = v$ for all healthy nodes, if the initial value of the source is v_s, say $v = v_s$.

Since most of the nodes are healthy, the value of healthy vertices for all healthy nodes' ms-tree is v. When the ms-tree is reorganized to an ic-tree, the healthy vertices still exist. As a result, each healthy vertex of the ic-tree is common by Lemma 1, and its true value is v. Using Theorem 2, this root is common. The computed value $\text{VOTE}(s) = v$ is stored in the root for all healthy nodes. $(\text{Validity}')$ is satisfied. □

COMPLEXITY OF OAP

The complexity of OAP is evaluated in terms of the minimal number of rounds and the maximum number of allowable faulty components. Theorem 4 and Theorem 5 below will show that the optimal solution is reached.

Theorem 4 *OAP requires O rounds to solve the generalized BA with malicious faulty CWSN if $C > \lfloor (C-1)/3 \rfloor + 2(T_{\text{FC}} + T_{\text{FT}})$ and $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$, where $O = \lfloor (C-1)/3 \rfloor + 2$, and O are the minimum number of rounds of exchanged messages.*

Proof: Because message passing is required only in the message exchange phase, the message exchange phase is a time consuming phase. Yan et al showed that $f_n + 1$ rounds, where $f_n \leqslant \lfloor (n-1)/3 \rfloor$, are the minimum number of rounds to get enough messages to reach BA in a node fault only environment where n is the total number of nodes in a network[11]. The network topology of Yan et al is traditional network architecture (such as fully connected network), and the unit of Yan et al is a node[11].

In case of faulty nodes only, it has been proven that $f_n + 1$ rounds of exchanged messages are required to demonstrate optimality[11]. Furthermore, Wang et al showed that two rounds are the minimum number of rounds to solve the TMs fault[10]. Hence the required number of rounds for solving the generalized BA problem in CWSN should not be less than $T_{\text{FC}} + 1$. As a result, the number of required rounds of message exchange in a CWSN is $O = \lfloor (C-1)/3 \rfloor + 2$. Thus OAP requires a minimum of O rounds of message exchange. □

Theorem 5 *The maximum number of allowable faulty components by OAP is T_{FC} malicious faulty clusters and T_{FT} malicious faulty transmission media where $C > \lfloor (C-1)/3 \rfloor + 2(T_{\text{FC}} + T_{\text{FT}})$ and $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$.*

Proof: In the past, Yan et al[11] showed that the constraints of the BA problem for node faults is only $n > \lfloor (n-1)/3 \rfloor + 2(f_n)$ and $\text{Con} > 2(f_n)$. The unit of Yan et al[11]. is node, and the unit of CWSN is composed of several clusters[5]. In this paper, the fault status of a CWSN with faulty nodes and faulty transmission media are discussed. Hence the constraints are rewritten as $\text{Con} > \lfloor (C-1)/3 \rfloor + 2(T_{\text{FC}} + T_{\text{FT}})$ and $\text{Con} > 2(T_{\text{FC}} + T_{\text{FT}})$. In other words, the total number of allowable faulty components of OAP is T_{FC} malicious faulty clusters and T_{FT} malicious faulty TMs. □

Theorem 6 *The number of allowable faulty nodes T_{FN} is the maximum.*

Proof: Every healthy node agrees on a value, which is dominated by most of the nodes in a cluster. When the number of faulty nodes is greater than a half of all the nodes in a cluster, the cluster is a faulty cluster. For this reason, two cases of fault tolerance are discussed, the best case and the worst case. There is the maximum number of faulty nodes in a CWSN, and no more faulty node can be increased, named best case; if a faulty node is increased in any non-faulty, and let the non-faulty cluster be a faulty cluster, named worst case.

In the best case of malicious faulty nodes, let $n_{\max(i)}$ be the number of nodes in i^{th} maximum cluster. The number of malicious faulty nodes is $\sum_{i=1}^{T_{\mathrm{FC}}} n_{\max(i)}$. An additional number of malicious faulty nodes $\sum_{j=T_{\mathrm{FC}}+1}^{C} \lceil n_{\max(i)}/2 \rceil - 1$ cannot influence the network, and the number of malicious faulty nodes cannot be increased. If the number of malicious faulty nodes is increased, the assumption of $C > \lfloor (C-1)/3 \rfloor + 2(T_{\mathrm{FC}} + T_{\mathrm{FT}})$ is contradicted. Hence the number of allowable malicious faulty nodes can be written as $T_{\mathrm{FN}} = \sum_{i=1}^{T_{\mathrm{FC}}} n_{\max(i)} + \sum_{j=T_{\mathrm{FC}}+1}^{C} \lceil n_{\max(i)}/2 \rceil - 1$.

In the worst case of malicious faulty nodes, let $n_{\min(i)}$ be the number of nodes in the i^{th} minimum cluster. The number of nodes in the malicious faulty cluster is $\sum_{i=1}^{T_{\mathrm{FC}}} \lceil n_{\min(i)}/2 \rceil$. An additional number of malicious faulty nodes $\lceil n_{\min(T_{\mathrm{FC}}+1)}/2 \rceil - 1$ cannot influence the network. Nevertheless, if a malicious faulty node is increased in $\min(T_{\mathrm{FC}} + 1)$th cluster, then a malicious faulty cluster is increased, and the assumption of $C > \lfloor (C-1)/3 \rfloor + 2(T_{\mathrm{FC}} + T_{\mathrm{FT}})$ is contradicted. Thus the number of allowable malicious faulty nodes can be written as $T_{\mathrm{FN}} = \sum_{i=1}^{T_{\mathrm{FC}}} \lceil n_{\min(i)}/2 \rceil + \lceil n_{\min(T_{\mathrm{FC}}+1)}/2 \rceil - 1$.

As the results of the above cases illustrate, the number of allowable faulty components is maximal in OAP. The OAP requires a minimum number of rounds and tolerates a maximum number of faulty components to ensure all healthy nodes reach a common agreement; hence the optimality of OAP has been proven. □

CONCLUSIONS

The changes in network topology developed in recent years[2,3] has demonstrated a trend towards increasingly mobile features. The proposed protocol, OAP, can solve the BA problem with dual failure modes on fallible nodes and transmission media in the CWSN. The proposed protocol, OAP, can taking the minimum number of required rounds to achieve an agreement, and tolerating the maximum number of faulty components.

Moreover, reaching an agreement is insufficient for the highly reliable distributed system in a CWSN. A related closely problem called the Fault Diagnosis Agreement (FDA) problem[12]. The objective of solving the FDA problem is to make each healthy node can detect or locate the common set of faulty components in the distributed system[6]. Hence solving the FDA problem for the highly reliable distributed system underlying CWSN is included in our future work.

REFERENCES

1. Christin D, Reinhardt A, Mogre PS, Steinmetz R (2009) Wireless sensor networks and the internet of things: selected challenges. In: *Proceedings of the 8th GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze"*, pp 31–4.
2. Akyildiz IF, Weilian S, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. *IEEE Comm Mag* **40**, 102–14.
3. Olifer N, Olifer V (2006) *Computer Networks: Principles, Technologies and Protocols for Network Design*, John Wiley & Sons.
4. Gupta I, Riordan D, Sampalli S (2005) Cluster-head election using fuzzy logic for wireless sensor networks. In: *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pp 255–60.
5. Wang SC, Yan KQ, Cheng CF (2004) Efficient multicasting agreement protocol. *Comput Stand Interface* **26**, 93–111.
6. Fischer MJ, Lynch NA (1982) A lower bound for the time to assure interactive consistency. *Inform Process Lett* **14**, 183–6.
7. Lamport L, Shostak R, Pease M (1982) The Byzantine generals problem. *ACM Trans Program Lang Syst* **4**, 382–401.
8. Bar-Noy A, Dolev D, Dwork C, Strong HR (1987) Shifting gears: changing algorithms on the fly to expedite Byzantine agreement. In: *Proceedings of the Sixth ACM Symposium on Principles of Distributed Computing*, pp 42–51.
9. Wang SC, Chiang ML, Yan KQ, Jea KF (2005) Streets of consensus under unknown unreliable network. *ACM SIGOPS Oper Syst Rev* **39**, 80–96.
10. Wang SS, Yan KQ, Wang SC (2011) Achieving efficient agreement within a dual-failure cloud-computing environment. *Expert Syst Appl* **38**, 906–15.
11. Yan KQ, Wang SC (2005) Grouping Byzantine Agreement. *Comput Stand Interface* **28**, 75–92.
12. Chiang ML, Wang SC, Tseng LY (2009) An early fault diagnosis agreement under hybrid fault model. *Expert Syst Appl* **36**, 5039–50.