An Optimal Agreement in a Cluster-based Wireless Sensor Network with Malicious Faulty Nodes

S.C. Wang, K.Q. Yan*, C.L. Ho, S.S. Wang*

Chaoyang University of Technology
168, Jifeng E. Rd., Wufeng District, Taichung, 41349 Taiwan, R.O.C.
{scwang; kqyan; s10033902; sswang*}@cyut.edu.tw*
**: Corresponding author*

***Abstract*—A cluster of sensor nodes in Cluster-based Wireless Sensor Network (CWSN) is cooperating to achieve some objectives. For CWSN, the fault-tolerance and reliability has been an important topic. The problem of reaching agreement in the distributed system is one of the most important issues to design a fault-tolerance system. In previous works, reach a common agreement among healthy nodes to cope with the influence from faulty components is significant in a fault-tolerance system. In this study, the Optimal Malicious Agreement Protocol (OMAP) is proposed in a CWSN, which the sensor nodes maybe subject to Byzantine (malicious) failure.**

***Keywords*—Byzantine failure, Distributed system, Fault tolerant, Wireless sensor network**

1. INTRODUCTION

Recently, the Micro-Electro-Mechanical Systems (MEMS) continues to grow at a high rate of speed in Wireless Sensor Networks (WSNs) [3]. However, the sensor node is limited by the energy resource, the memory, the computation, and the communication capability, etc. [6,17]. WSN is a distributed system that comprises thousands sensor nodes and sink. The characteristics of a WSN include small-scale sensor nodes, limited power, mobility, dynamic network topology, wireless communication, etc. [2,5,8,15,16,19-21,26]. However, the sensor nodes will collect the information and deliver it back to the sink node by using multi-hop wireless links from a specific region or nature environment. Moreover, WSN is a non-infrastructure network, there are two states of each sensor: move and fixed [2,3,11,14]. If there are sensor node leave from the original network, then it will communicate with each neighbor sensor node and try to become a new brief network topology. In other word, the network topology will be reconfigured in any necessitous times.

Nowadays, the WSN is practical more and more due to it can provide sensor node joins to the network or leaves away anytime with non-infrastructure. A group of sensor nodes in WSN is cooperating to achieve some objectives; each sensor node communicates with other sensor nodes by using broadcast in WSN, but also leads to a severe problem, such as broadcast storm [4]. Many researchers proposed cluster schemes and broadcast limited to prevent the broadcast storm [22]. However, the clustering topology has been proposed to prolong the lifetime of WSNs by decreasing the energy consumption of sensor nodes [1].

In this study, the Byzantine Agreement problem is revisited with the assumption of sensor node failure on malicious faults in the Cluster-based Wireless Sensor Network (CWSN). The proposed protocol, **O**ptimal **M**alicious **A**greement **P**rotocol (OMAP), can make all healthy sensor nodes reaching agreement with minimal rounds of message exchange and tolerate the maximal number of allowable components.

The rest of this study is organized as follows. Section 2 discusses the CWSN and the related work for Byzantine Agreement problem. The OMAP is illustrated in detail in Section 3. Section 4 gives an example of executing OMAP. Section 5 proves the correctness and complexity of OMAP. Section 6 concludes this study.

2. RELATED WORK

Recent advances in technology have provided portable nodes with wireless interfaces that allow networked 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 network topology of our research and the related results of agreement problem have discussed in this section.

2.1. The Topology of Cluster-based Wireless Sensor

As WSNs need not any infrastructure to provide the multi-hop wireless links for the mobile user, the network will offer the mechanism for the simultaneous uses of many users in order to apply widely for the field of actual practice. However, the method of search-address and ringing is more difficult than the common the network, for this reason, the hierarchical routing approach of WSN is able to solve efficiently the problems of complex routing, while the clustering is used for setting up and keeping the hierarchical routing.

The communicative behaviors in WSNs can be characterized by two different types: routing (*node-to-sink*) and broadcasting (*sink-to-node* or *node-to-node*). Broadcasting is an essential communication requirement for sink and sensor nodes. The sensor node can sense environment information and forward information to next sensor node until sink node that is named routing [5,10-12]. Therefore, how to increase stable, establish a secure network and decrease the consumption of power is an important issue.

Data aggregation is an important work for saving energy consumption whether static or dynamic WSNs. For data aggregation, certain amount of sensors in the vicinity forms a team to aggregate data [9,10]. However, WSN is made up of several clusters of sensors, and several clusters may make up of more large clusters [22]. Therefore, the topology of Cluster-based Wireless Sensor Network (CWSN) has been proposed to prolong the lifetime of WSNs by decreasing the energy consumption of SNs.

In CWSN, the topology is composed of several clusters. Each cluster is composed of many sensor nodes and one cluster manager. The sink controls the state and communication data of all cluster managers. The cluster manager controls the sate and communication data of all sensor nodes. And, the sensor nodes answer to sense data. Fig. 1 is a topology of CWSN.

2.2 Byzantine Agreement Problem

In the CWSN, the sensor nodes interconnected with the wireless; the network is assumed reliable and synchronous [18]. If certain components in distributed system were failed, to achieve agreement in a distributed system the protocols are required so that systems still can be executed correctly.

The Byzantine Agreement (BA) problem [13] is one of the most fundamental problems concerning reaching agreement in distributed systems. First studied by Lamport *et al.* [13], it is a well-known paradigm for achieving reliability in a distributed network of nodes. According to the definition of BA problem by Lamport *et al*.:

1) There are n nodes, of which at most $\lfloor (n-1)/3 \rfloor$ nodes could fail without breaking down a workable network.
2) The nodes communicate with each other through message exchange in a fully connected network.
3) The message sender is always identifiable by the receiver.
4) A node is chosen as a source, and its initial value v_s is broadcasted to other nodes and itself to execute the protocol.

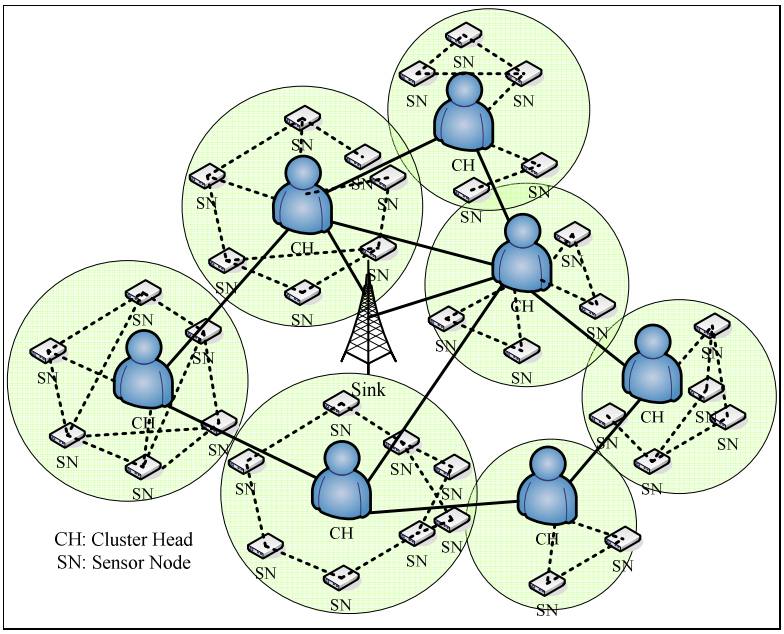

Fig. 1. The topology of Cluster-based Wireless Sensor Network (CWSN)

The solutions have defined as protocols, which achieve agreement and hope to use the minimal rounds of message exchange to obtain the maximum number of allowable faulty capability. We concern the solution of BA problem in this study. The definition of BA problem is to make the healthy nodes in *n*-sensor nodes CWSN to achieve agreement. The source sensor node chooses an initial value to start with, and communicates to each other by exchanging messages. The nodes of a cluster have referred to make an agreement if it satisfies the following conditions [13]:

(Agreement): All healthy sensor nodes agree on a common value.

(Validity): If the source sensor node is healthy, then all healthy sensor nodes shall agree on the initial value the source sensor node sends.

In a BA problem, many cases had based on the assumption of node failure in a fail-safe network [18]. Base on this assumption, the goal of solving a BA problem is to develop an optimal algorithm can use the minimal number of rounds to achieve an agreement.

Here we consider the network topology in a distributed system whose communication media are reliable during the BA execution in CWSN, while the node may be faulty by interference from hijackers and results in the exchanged message can exhibit arbitrary behavior. A protocol to reach agreement in a reliable communication environment of tradition network topology has proposed first by Lamport *et al,* [13]. The typical protocol by Fischer [7] can tolerate $f \leq \lfloor (n-1)/3 \rfloor$ faulty nodes in malicious and required σ ($\sigma = f+1$) round(s) to get enough messages to achieve agreement.

However, most of distributed computing systems may not be fully connected. The network topology has the feature of cluster or group just like the topology of CWSN. However, OMAP is used to solve the malicious sensor node fault in CWSN. When all sensor nodes achieve agreement, the fault-tolerance capacity has enhanced even if the communication media are fault between sensor nodes and the backbone can be used to provide a backup route [22].

3. THE PROPOSED PROTOCOL

The proposed protocol **O**ptimal **M**alicious **A**greement **P**rotocol (OMAP) can solve the BA problem due to faulty sensor nodes which may send wrong messages to influence the system to reach agreement in a synchronous CWSN. OMAP protocol consists two phases and needs σ rounds of message exchange to solve the BA problem.

3.1. Protocol Notation

The assumptions and parameters of OMAP for the topology of CWSN are shown as follows:

- The underlying network is synchronous.
- Each node in the network can be identified uniquely.
- A node does not know the fault status of other components.
- Let x be the cluster identifier where $1 \le x \le N$ and N is the number of clusters $N \ge 4$.
- Let f_n be a total number of malicious faulty sensor nodes.
- Let F_C be the maximum number of allowable faulty clusters, $F_C \le \lfloor (N-1)/3 \rfloor$.
- Let T_{Fn} is the total number of allowable faulty sensor nodes, $1 \le f_n \le T_{Fn}$.
- Let n_x be the number of sensor nodes in cluster C_x, $0 \le x \le N$.

3.2. Protocol Model

In this section, OMAP is introduced to solve agreement problems with malicious faulty sensor nodes underlying a CWSN. The OMAP is organized as two phases, the *Message Gathering Phase* and *Agreement Making Phase*. In the *Message Gathering Phase*, each node is to collect enough information from other nodes in the CWSN. And, in the *Agreement Making Phase*, the collected information by *Message Gathering Phase* is used to decide the agreement value.

In the first round ($\sigma=1$) of *Message Gathering Phase*, the source node sends its initial value to all sensor nodes, and then receiver node stores the received value in the root of its mg-tree. The mg-tree is a tree structure that is used to store the received message [25]. After the first round of *Message Gathering Phase* ($\sigma>1$), each sensor node without source node transmits the value at level $\sigma-1$ in its mg-tree to all nodes; At the end of each round, the receiver node takes the local majority value on its received values which are from the same cluster, to get a single value. Moreover, each receiver node stores the single value that is majority of the received values in its mg-tree.

Afterward, in the *Agreement Making Phase*, each node without the source node reorganizes its mg-tree into a corresponding ic-tree. The ic-tree is a tree structure that is used to store a received message without repeated cluster names [25]. Therefore, the common value VOTE(s) has obtained by using function VOTE on the root s of each node's ic-tree. The detail steps of our proposed protocol has presented in Fig. 2.

4. AN EXAMPLE BY USING OMAP

In the OMAP protocol, an example is given for executing our protocol OMAP. An example of CWSN topology is shown in Fig. 3(a). There are 22 nodes falling into seven clusters. C_1 includes source node n_s, n_1 and n_2. C_2 includes n_3, n_4, n_5 and n_6. C_3 includes n_7, n_8, n_9 and n_{10}. C_4 includes n_{11} and n_{12}. C_5 includes n_{13} and n_{14}. C_6 includes n_{15} and n_{16}. n_{17}, n_{18}, n_{19}, n_{20} and n_{21} belong to C_7.

In the BA problem, the worst situation is that the source does not honest anymore [13]. Simply, here the worst case of example, suppose the source node n_s is malicious faulty node, which means n_s may send arbitrarily different values to different clusters. Therefore, in order to solve the BA problem among healthy nodes of example, OMAP requires σ ($\lfloor (N-1)/3 \rfloor + 1 = 3$) rounds of *Message Gathering Phase*.

In OMAP, Pre-Execute counts the number of rounds required before *Message Gathering Phase*. In this example, three rounds to message exchange are needed.

The source node n_s transmits messages to all other nodes in the first round of *Message Gathering Phase*. The messages sent by the source node n_s are shown in Fig. 3(a), where n_s sends value 1 to C_2, C_4, C_5, C_6 and C_7; n_s sends value 0 to C_1 and C_3. In addition, the message obtained of each healthy node has listed in Fig. 3(b). In the σ-th ($1<\sigma\le\theta$) round of *Message Gathering Phase*, except for the source node, each node transmits the values at the ($\sigma-1$)-th level in its mg-tree to all the others and itself. Subsequently, each receiver node takes the local majority value on the received values from the same cluster and stores the received messages at the corresponding vertices at level σ of its mg-tree. The mg-tree of healthy node n_1 at the second and final round in the message exchange phase is shown in Fig. 3(c) and (d), and the *Message Gathering Phase* has completed.

After the *Message Gathering Phase*, the tree structure of each healthy node has converted from mg-tree to ic-tree by deleting the vertices with duplicated names (such like *s11* will be deleting) in the *Agreement Making Phase*. The example ic-tree has showed in Fig. 3(e). Eventually, using the function VOTE to root the value s for each healthy node's ic-tree {VOTE(s)

=VOTE(*s1*), …, VOTE(*s7*)=1}, an agreement value 1 can be obtained, as shown in Fig. 3(f), and the *Agreement Making Phase* has completed. In the end, comparing the root *s* value of healthy node in C_1, the root value of all healthy nodes in C_1 and C_3 has altered its different value to 1. In other hand, after executing the OMAP protocol, all healthy nodes agree on a common value 1 for the example (Fig. 3).

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 depends on the behaviours of which kinds of transmission.
4. The sensor node cannot garble the message between the sender node and receiver node; this assumption has 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 σ = 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 σ > 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*) has obtained.

Function VOTE(μ)

If the μ is a leaf, then output the value μ.

Else if the majority value is not existed, then output the majority value ϕ.

Otherwise, output the majority m, where $m \in \{0, 1\}$

Fig. 2. The OMAP protocol

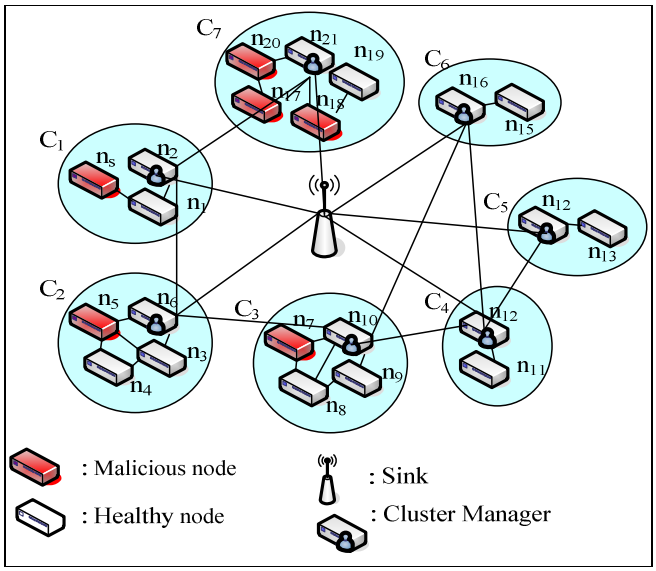

Fig. 3(a). The initial status of executing OMAP

	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

Fig. 3(b). The mg-tree of each node at the 1st round

Level 1	*Level 2*	*Take local majority*	
Val(*s*)=1	*s1*	0	(0,0)
	s2	1	(1,1,***0***,1)
	s3	0	(0,***0***,0,0)
	s4	1	(1,1)
	s5	1	(1,1)
	s6	1	(1,1)
	s7	0	(***0***,***0***,1,***0***,1)

Fig. 3(c). The mg-tree of healthy node n_1 at the 2nd round

Level 1	*Level 2*	*Level 3*	*Take local majority*	
s	*s1*	*s11*	0	(0)
0	0(0)	*s12*	0	(0,0,***0***,0)
		s13	0	(0,***1***,0,0)
		s14	0	(0,0)
		s15	0	(0,0)
		s16	0	(0,0)
		s17	1	(***1***,***1***,1,***0***,1)
	s2	*s21*	1	(1)
	1(1,1,1,1)	*s22*	1	(1,1,***1***,1)
		s23	1	(1,***1***,1,1)
		s24	1	(1,1)
		s25	1	(1,1)
		s26	1	(1,1)
		s27	0	(***0***,***0***,1,***0***,1)
	s3	*s31*	0	(0)
	0(0,0,0,0)	*s32*	0	(0,0,***1***,0)
		s33	0	(0,***1***,0,0)
		s34	0	(0,0)
		s35	0	(0,0)
		s36	0	(0,0)
		s37	0	(***0***,***0***,1,***0***,1)
	s4	*s41*	1	(1)
	1(1,1)	*s42*	1	(1,1,***0***,1)
		s43	1	(1,***1***,1,1)
		s44	1	(1,1)
		s45	1	(1,1)
		s46	1	(1,1)
		s47	1	(***1***,***1***,1,***0***,1)
	s5	*s51*	1	(1)
	1(1,1)	*s52*	1	(1,1,***1***,1)
		s53	1	(1,***0***,1,1)
		s54	1	(1,1)
		s55	1	(1,1)
		s56	1	(1,1)
		s57	0	(***0***,***0***,1,***0***,1)
	s6	*s61*	1	(1)
	1(1,1)	*s62*	1	(1,1,***1***,1)
		s63	1	(1,***1***,1,1)
		s64	1	(1,1)
		s65	1	(1,1)
		s66	1	(1,1)
		s67	1	(***1***,***1***,1,***1***,1)
	s7	*s71*	0	(0)
	0(0,0,1,0,1)	*s72*	1	(1,1,***1***,1)
		s73	0	(0,***0***,0,0)
		s74	1	(1,1)
		s75	0	(0,0)
		s76	1	(1,1)
		s77	0	(***0***,***0***,1,***0***,1)

Fig. 3(d). The final mg-tree of node n_1 after the *Message Gathering Phase*

The tree structure has converted from mg-tree to ic-tree by erasing the vertices with repeated names.

Level 1	*Level 2*	*Level 3*	*Take local majority*	
s	*s1*			
0	0	*s12*	0	(0,0,***0***,0)
	(0)	*s13*	0	(0,***1***,0,0)
		s14	0	(0,0)
		s15	0	(0,0)
		s16	0	(0,0)
		s17	1	(***1***,***1***,1,***0***,1)
	s2	*s21*	1	(1)
	1			
	(1,1,1,1)	*s23*	1	(1,***1***,1,1)
		s24	1	(1,1)
		s25	1	(1,1)
		s26	1	(1,1)
		s27	0	(***0***,***0***,1,***0***,1)
	s3	*s31*	0	(0)
	0(0,0,0,0)	*s32*	0	(0,0,***1***,0)
		s34	0	(0,0)
		s35	0	(0,0)
		s36	0	(0,0)
		s37	0	(***0***,***0***,1,***0***,1)
	s4	*s41*	1	(1)
	1	*s42*	1	(1,1,***0***,1)
	(1,1)	*s43*	1	(1,***1***,1,1)
		s45	1	(1,1)
		s46	1	(1,1)
		s47	1	(***1***,***1***,1,***0***,1)
	s5	*s51*	1	(1)
	1	*s52*	1	(1,1,***1***,1)
	(1,1)	*s53*	1	(1,***0***,1,1)
		s54	1	(1,1)
		s56	1	(1,1)
		s57	0	(***0***,***0***,1,***0***,1)
	s6	*s61*	1	(1)
	1	*s62*	1	(1,1,***1***,1)
	(1,1)	*s63*	1	(1,***1***,1,1)
		s64	1	(1,1)
		s65	1	(1,1)
		s67	1	(***1***,***1***,1,***1***,1)
	s7	*s71*	0	(0)
	0	*s72*	1	(1,1,***1***,1)
	(0,0,1,0,1)	*s73*	0	(0,***0***,0,0)
		s74	1	(1,1)
		s75	0	(0,0)
		s76	1	(1,1)

Fig. 3(e). The ic-tree of node n_1

- ✓ VOTE(*s1*) = (0, 0, 0, 0, 0, 1) = 0
- ✓ VOTE(*s4*) = (1, 1, 1, 1, 1, 1) = 1
- ✓ VOTE(*s7*) = (0, 1, 0, 1, 0, 1) = ϕ
- ✓ VOTE(*s2*) = (1, 1, 1, 1, 1, 0) = 1
- ✓ VOTE(*s5*) = (1, 1, 1, 1, 1, 0) = 1
- ✓ VOTE(*s3*) = (0, 0, 0, 0, 0, 0) = 0
- ✓ VOTE(*s6*) = (1, 1, 1, 1, 1, 1)) = 1

VOTE(*s*) = (VOTE(*s1*), VOTE(*s2*), VOTE(*s3*), VOTE(*s4*), VOTE(*s5*), VOTE(*s6*), VOTE(*s7*)) = (0, 1, 0, 1, 1, 1, ϕ) = 1

Fig. 3(f). The common value VOTE(*s*) by healthy node n_1

Fig. 3. An example of OMAP execution

5. CORRECTNESS AND COMPLEXITY

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

5.1. Correctness of OMAP

Underlying the proof of our protocol's correctness, a vertex α is called common [24] if the value stored in vertex α of each healthy node's mg-tree or ic-tree is identical. If each healthy node shares a common initial value of source node in the root of an ic-tree, and 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 tree structure, then agreement is reached because the root is common. Thus, the constraints, (Agreement) and (Validity), can be rewritten as:

(**Agreement'**): Root s is common, and
(**Validity'**): VOTE(s)=v_s for each healthy node, if the commander is healthy.

To prove that a vertex is common, the term common frontier [18] is defined as follows: When every root-to-leaf path of a tree (an mg-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 within the common frontier if it exists within a healthy node's tree structure (mg-tree or ic-tree). Subsequently, using the same majority voting function to compute the root value of tree structure, every healthy node can compute the same root value because they all use the same input (the same collected messages within the common frontier). The same computing function results in the same output.

Lemma 1: All correct vertices of an ic-tree are common.
Proof: After reorganization, no repeatable vertices are in an ic-tree. At the level F_C+1 or above, the correct vertex α has at least $2F_C$ +1 child in which at least F_C +1 children are correct. The true value of these F_C +1 correct vertices is in common, and the majority value of vertex α is common. The correct vertex α is common in the ic-tree, if the level of α is less then F_C +1. As a result, all correct vertices of ic-tree are common. ■

Lemma 2: **The common frontier exists in the ic-tree.**
Proof: There are F_C +1 vertices along each root-to-leaf path of an ic-tree in which the root is labeled by the source name, and the others are labeled by a sequence of cluster names. Since at most F_C clusters can be failed, there are at least one vertex is correct along each root-to-leaf path of ic-tree. By Lemma 1, the correct vertex is common, and the common frontier exists in each healthy node's ic-tree. ■

Lemma 3: Let α be a vertex, α is common if there is a common frontier in the subtree rooted at α.
Proof: If the height of α is 0, and the common frontier (α itself) exists, then α is common. If the height of α is σ, the children of α are all in common by using induction hypothesis with the height of children at σ-1, then the vertex α is common. ■

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

Theorem 1: 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 2: Protocol OMAP solves the BA problem in a CWSN.
Proof: To prove the theorem, it has to show that OMAP meets the constraints (Agreement') and (Validity')

(**Agreement**'): Root s is common. By Theorem 1, (Agreement') is satisfied.
(**Validity**'): VOTE(s)=v for all healthy nodes, if the initial value of source is v_s say $v = v_s$.

Since most of nodes are healthy, they transmit the message to all others. The value of correct vertices for all healthy nodes' mg-tree is v. When the mg-tree is reorganized to an ic-tree, the correct vertices still exist. As a result, each correct vertices of ic-tree is common (Lemma 1), and its true value is v. By Theorem 1, this root is common. The computed value VOTE(s) = v is stored in the root for all healthy nodes. (Validity') is satisfied. ■

5.2. Complexity of OMAP

The complexity of OMAP is evaluated in terms of 1) the minimum number of rounds; and 2) the maximum number of allowable faulty components. Theorems 3 and 4 below show that the optimal solution is reached.

Theorem 3: OMAP requires F_C +1 rounds to solve the BA problem with malicious fault in a CWSN where $F_C \leq \lfloor (N-1)/3 \rfloor$.

Proof: Due to the message passing is required in the *Message Gathering Phase* only. Thus, the message exchange phase is a time consuming phase. Fischer [7] pointed out that $t+1$ ($t \leq \lfloor (n-1)/3 \rfloor$) rounds are the minimum number of rounds to get enough messages to achieve BA. The unit of Fischer [7] is node, but the unit of CWSN is cluster. So that, the number of required rounds of message exchange in the CWSN is F_C+1($F_C \leq \lfloor (N-1)/3 \rfloor$). Thus OMAP requires F_C +1 rounds and this number is the minimum. ■

Theorem 4: The total number of allowable faulty components by OMAP is F_C malicious faulty clusters, where $F_C \leq \lfloor (N-1)/3 \rfloor$.

Proof: In Siu *et al.* [23] indicates the constraints of BA problem for node faults only is $f \leq \lfloor (n-1)/3 \rfloor$. However, the unit of CWSN is cluster, so we can suppose a node in Siu *et al.* as a cluster in CWSN. Therefore, $f \leq \lfloor (n-1)/3 \rfloor$ in Siu *et al.* imply $F_C \leq \lfloor (N-1)/3 \rfloor$ in CWSN. So the total number of allowable faulty components by OMAP is F_C malicious faulty clusters. ■

6. CONCLUSION

The complex networks had studied in a branch of mathematics known as graph theory in the past. The network topology developed in recent years shows a wireless feature. The previous protocols [13,18,25] cannot adapt to solve BA problem in CWSN, and none of BA protocol is designed for the CWSN. Therefore, we revisit the BA problem in CWSN with malicious faulty nodes. The OMAP can tolerate the most damaging failure type of fallible nodes. However, OMAP can take the minimum number of required rounds to reach an agreement, and tolerate the maximum number of faulty components.

Furthermore, only considering node faults in the BA problem is insufficient for the highly reliable distributed system of CWSN. In the real world, not only might nodes crash, omission or malicious, but also might communication medium crash, omission or malicious. On the other hand, our protocol will be extended to solve when dormant or malicious communication media or nodes exist simultaneously in the CWSN in future work.

ACKNOWLEDGMENT

This work was supported in part by the Taiwan National Science Council under Grants NSC99-2221-E-324-022 and NSC100-2221-E-324-022.

REFERENCES

[1] 802.15.4-2006 IEEE Standard for Information Technology-Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 2006.

[2] K. Akkaya, and M. Youngish, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Network,* Vol. 3, Issue 3, pp. 325-349, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks,* Vol. 38, Issue 4, pp. 393-422, 2002.

[4] S. Bandyopadhyay and E.J. Coyle, "Minimizing Communication Costs in Hierarchically-Clustered Networks of Wireless Sensors," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 44, Issue 1, pp. 1-16, 2004.

[5] S. Bandyopadhyay and E.J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proceeding of INFOCOM 2003*, Vol. 3, pp. 1713-1723, 2003.

[6] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 263-270, 1999.

[7]]M. Fischer, M. Paterson and N. Lynch, Impossibility of distributed consensus with one faulty process, *Journal of ACM*, Vol. 32, pp. 374-382, 1985.

[8] W. R. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless

Microsensor Networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS-33),* Vol. 2, pp. 1-10, 2000.

[9] W. B. Heinzelman, A. P. Chandrakasan and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications,* Vol. 1, Issue 4, pp. 660-670, 2002.

[10] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking,* pp. 174-185, 1999.

[11] G. Indranil, D. Riordan, and S. Srinivas, "Cluster-head Election using Fuzzy Logic for Wireless Sensor Networks," *Proceedings of the 3rd Annual Conference on Communication Networks and Services Research,* pp. 255-260, 2005.

[12] R. Krishnan and D. Starobinski, "Efficient Clustering Algorithms for Self-Organizing Wireless Sensor Networks," *Ad Hoc Networks,* Vol. 4, Issue 1, pp. 36-59, 2006.

[13] L. Lamport, R. Shostak and M. Pease, The Byzantine General Problem, *ACM Transactions on Programming Language and Systems*, Vol. 4, No. 3, pp. 382-401, 1982.

[14] B. Lehane and L. Doyle, Shared RSA Key Generation In A Mobile Ad Hoc Network, *in the Military Communications of IEEE Conference*, Vol. 2, pp 814-819, 2003.

[15] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems," *Proceedings of the IEEE Aerospace Conference,* pp. 3-1125 - 3-1130, 2002.

[16] J. S. Liu and C. H. Richard Lin, "Energy-efficiency Clustering Protocol in Wireless Sensor Networks," *Ad Hoc Networks,* Vol. 3, Issue 3, pp. 371-388, 2005.

[17] V Mhatre, C Rosenberg, D Kofman, R. Mazumdar and N. Shroff, "Design of Surveillance Sensor Grids with a Lifetime Constraint," *The First European Workshop on Wireless Sensor Networks (EWSN),* pp. 263-275, 2004.

[18] M. Pease, R. Shostak and L. Lamport, Reaching Agreement in Presence of Faults, *Journal of ACM*, Vol. 27, No. 2, pp. 228-234, 1980.

[19] L. Qing, Q.X. Zhu and M. W. Wang, "Design of a Distributed Energy-Efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks," *Computer Communications,* Vol. 29, Issue 12, pp. 2230-2237, 2006.

[20] C. Schurgers and M. B. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks," *Proceedings on Communications for Network-Centric Operations: Creating the Information Force (MILCOM),* pp. 357-361, 2001.

[21] R. C. Shah and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC2002)*, Vol. 1, pp. 350-355, 2002.

[22] B. Shen, S.Y. Zhang, and Y.P Zhong, "Cluster-Based Routing Protocols for Wireless Sensor Networks," *Journal of Software,* Vol.17, No.7, pp. 1588−1600, 2006.

[23] H.S. Siu, Y.H. Chin, W.P. Yang (1996), "A note on consensus on dual failure modes", *IEEE Transactions on Parallel and Distributed Systems 7 (3)*, pp. 225-230.

[24] S.C. Wang, K.Q. Yan and G.Y. Zheng, Reaching Consensus Underlying Fallible Virtual Subnet of Mobile Ad-Hoc Network, *Twelfth Mobile Computing Workshop*, pp. 257-263, 2006.

[25] K.Q. Yan, S.C Wang, Group Byzantine Agreement, *in Computer Standards & Interfaces*, pp. 75-92, 2005.

[26] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks," *IEEE Transactions on Mobile Computing,* Vol. 3, Issue 4, pp. 660-669, 2004.