Available online at: http://www.ijmtst.com/vol4issue5.html

International Journal for Modern Trends in Science and Technology

ISSN: 2455-3778 :: Volume: 04, Issue No: 05, May 2018

# In WSN Secure Fault Tolerant Agreement Model (SFTAM) using BFT Algorithm

Venkataramana K[1] | Dr. Manoj Kumar[2]

[1]Research Scholar, Department of CSE, SVU, Gajraula, UP, India.

[2]Associate Professor, Department of CSE, SVU, Gajraula, UP, India.

**To Cite this Article**

Venkataramana K and Dr. Manoj Kumar, "In WSN Secure Fault Tolerant Agreement Model (SFTAM) using BFT Algorithm", *International Journal for Modern Trends in Science and Technology*, Vol. 04, Issue 05, May 2018, pp.:43-48.

## ABSTRACT

*WSNs are demand in most of the applications in which sensor nodes are always work together in a harsh and hostile manner. This behavior of harsh and hostile manner will leads to faulty nodes in WSNs. The faulty nodes may be due to the sensor faults of which it is broadly categorized into two types such as crash faults where a sensor node becomes inactive in the network and soft fault where the sensor node behaves arbitrarily. If either the faulty nodes were not monitored and properly handled in response, these may lead to inaccuracy of the data, cause congestion on the route and reduce the lifespan of a network. So to improve above all parameters we designed a SFTAM model in which agreement for BFT is set up to diagnose the faults and fault tolerant is achieved using BFT algorithm.*

***KEYWORDS:*** *WSNs, Agreement Model, Fault tolerant, BFT algorithm*



## I. INTRODUCTION

WSNs come with small, affordable and smart sensor nodes. It is a less networked system, and runs on each sensor node with resource constraints such as limited battery power, short communication range, low bandwidth, and limited processing and storage. A sensor network consists of thousands of physically installed, unattended, and often unlettered devices. Due to various inevitable circumstances of natural calamities, WSNs are more prone to errors. Efficient fault diagnosis in WSNs is therefore needed to maintain the quality of WSN's service.Since they are deployed in unattended and hostile environments, sensor nodes used in various application domains are expected to operate autonomously. The sensor nodes are prone to having faults because of this. The underlying cause of sensor fault is device disorder that arises due to mechanical or electrical problems in the sensor node's internal circuits, environmental degradation, depletion of batteries, or hostile abuse, etc. [1]

Sensor nodes in WSNs are always expected to work together autonomously in an unattended, harsh, and even aggressive environment according to the demands of most applications. As a consequence, those nodes appear to be corrupted or go dead over time. If either the faulty nodes were not monitored and properly handled in time, they will lead to unreliability of the data, affect the bandwidth of the network, cause sectional route congestion and reduce the life of a network. [2]Consequently, the reasons for diagnosing fault include the following. Raising the reliability of data for various reasons, sensor nodes become unstable and unreliable e.g. hardware and software failure, effects on the environment, malware attempts that force nodes to deliver faulty data, etc. The latter information will be passed on to an SN or BS, and the BS will minimize the accuracy of judgments. Make effective use of bandwidth: Bandwidth refers to the data transfer rate, expressed in bits per

second. Because a wireless network of sensors is a source-constrained network, it really is challenging to expand the bandwidth according to the needs of the nodes. Inevitably faulty data consumes network bandwidth.

Prolongation of network lifespan: The lifetime of the network is directly related to the capacity of nodes [3]. Nodes begin to die as they spend most of their resources conducting different network operations and data transmission. Limited battery power leads to poor communication between nodes, and when this occurs, the nodes will no longer be part of the network and will cause partitioning of the network instead.[4]

Objectives identified and proposed in our paper are.

- The rest of the paper contains section II is the Related work
- Section III is proposed SFTAM Model, we designed an efficient agreement model for identifying the faults and make recover from it in sensor nodes.
- Section IV is all about the **Byzantine Fault Tolerance** is shown with a working and how the agreement is made in it to achieve fault tolerant.
- Section IV is the conclusion.

## II. RELATED WORK

Jabbari et al. [5] proposed a fault diagnostic algorithm based on the Artificial Neural Network (ANN), under which faulty sensor nodes are identified based on analysis of sensed data generated by individual sensor nodes. This method follows two steps, like the generation and verification of residual. The generalized regression neural network architecture is used for residual generation, and the kernel-based learning approach is implemented for residual verification.

Liu et al. [6] suggested a self-fault diagnostic algorithm based on a finite state machine to diagnose the hard-defective sensor nodes present in the network. The hard-defective sensor nodes occur in their approach only due to low battery power or device reset, a neighboring node detects that a sensor node is dead or of low quality due to interference, and high retransmission ratio. The method places overhead signaling on the network. It also requires the high costs of storage and computation.Banerjee et al. [7] explored the process of cellular automaton induced fault diagnosis. They develop a spatial and temporal correlation of sensing information-based approach to fault diagnosis which effectively diagnoses the faulty sensor nodes.

Nandi et al. [8] suggested a classical theory test algorithm based on the diagnosis of fault dependent topology. Entire regions of interest (ROI) have been divided into a variety of sub-squares where the ROI center positions the sensor node si. The author seems to think that all the sensor nodes were used to identify an event that occurred in the surrounding regions. Faulty nodes are identified based on the likelihood of error based on the Neyman Pearson Most Efficient (MP) check determined by the base station. Since each sensor node si sends its sensed data xi directly to the base station, it needs to exchange O (N) messages over the network. This method brings lesser inefficiencies over the channel because multi-hop communication is not needed in this approach. However, because of the direct communication between base station and sensor node, it depletes the battery very rapidly, as energy is proportional to size.

Lau et al. [9] suggested a probabilistic algorithm for fault analysis in WSNs that requires additional consumption of resources. End to end transfer time and the Bayes classifier has been used to identify the networks hard and soft defective sensor nodes. The base station detects a list of defective nodes by obtaining the usual information from the sensor nodes. For fault diagnosis this approach does not consume sensor node resources. Because of the mechanism for diagnosing fault is based on an end-to-end transmission time of the individual packet coming from sensor nodes to the base station.The accuracy of the fault diagnosis is more so since it uses the classifier Bayes. It transmits maximum N packets over the network, due to which the difficulty of its time was O (N). There is no extra overhead for diagnosis since normal data are used for diagnosis purposes.

In order to identify sensor failures, Wang et al. [8] discuss a distributed fusion decision-tolerant approach. Thus, each sensor node sends its decisions sequentially to the base station which requires multi-hop communication. A collective sensor fault diagnosis (CSFD) technique is used when conducting distributed decision fusion to remove the unstable local decisions. An upper bound on the likelihood of fusion error is defined on the basis of the pre-designed fusion law, assuming equivalent local decision laws, and fault-free environments. Depending on this error

bound, a search criterion for the defective nodes is suggested.Once the fusion center detects the defective nodes, all subsequent local decisions are excluded from the estimation of the probability ratios adopted for final decision taking.
Andreas et al. [10] suggested a Byzantine approach for the diagnosis of faults where each sensor node si sends a series of messages to a group of sensor nodes and also receives messages from the same group. If the number of messages sent and received is equal, then the node of the sensor will be marked as free of fault otherwise it will be defective. This method involves multi-hop communication, and to locate the defective node requires coordination among the nodes.
Panda et al. [2] suggested a semi-centralized, test-based approach to fault diagnosis in which the base station assigns each sensor node a role if. All sensor nodes assess the mission and submit the responses to the base station. The base station examines the data received to classify possibly defective sensor nodes.

## III. PROPOSED MODEL

Sensor deficiencies can occur due to a component failure such as microprocessor, transceiver, memory subsystems, energy source, sensors, and actuators or environmental noise. Because faults are unavoidable in WSNs, the determination of the collection of fault-free and defective sensor nodes is crucial. In the presence of a number of defective systems, the method of identifying both fault-free and faulty sensor nodes in a wireless sensor network that we suggest a stable distributed model and sensor network diagnosis is used to achieve reliability over all the system. It often needs systems to decide on certain data value required throughout network computation.

The various kinds of faults that occur in a WSN are of two forms.
1. Node fail-stop: The cluster head (CH) nodes fails here in this form of faults and stop the further cycle from submitting no data to the Base Station (BS).
2. Arbitrary node fails: In such a type of faults, the responses may answer to Base Station (BS) with an incorrect result. Sometimes it will return an error result which will intentionally produce misleading results. It will respond to different nodes with incorrect results. The latter types of faults are defined as consensus in our WSNs due to misbehaving processes. There is also another consensus, as we stated below

• What happens when the leader of the cluster (CH) chooses not to follow the rules and mess with the status of its followers?

• What happens if the Cluster Head (CH) is a big part of the network but not the majority?

As these several faults occur, we need our system to be tolerant to fault, so that we can recognize and recover faults. Failure tolerance is a system's ability to sustain a failure within the system and to maintain stable, effective operation. Fault tolerance refers to how much and various types of faults a system can manage. Since these many faults are happening, we need to be aware of the fault with our system.

Some forms of Fault Tolerance are

1. Modular redundancy: Multiple redundant systems feed into a voting network which makes the valid data decision.

2. Byzantine Fault Tolerance: A transient fault handling approach that guarantees correct device performance when certain conditions have been met.

Therefore a secure fault tolerant agreement model (SFTAM) is designed.

### 3.1 SECURE FAULT TOLERANT AGREEMENT MODEL (SFTAM)

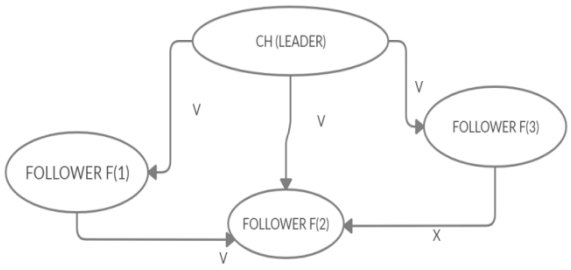
For a stable consensus protocol to be developed, it must be tolerant of fault. So we have developed a stable protocol called Secure Fault Tolerant Agreement Model (SFTAM) that uses algorithm called Byzantine Fault Tolerance that is distributed and decentralized in nature. Here we'll explain how our network module will contribute to it. Yet many problems also impede the widespread adoption of the WSN model. The aim of this Special Secure Fault Tolerant Agreement Model (SFTAM) is to put together revolutionary innovations in WSN-related areas so our system will become tolerant for failure. Therefore, we may consider the following requirements and make an agreement cluster. The rules of agreement cluster are as follows.

### 3.2 AGREEMENT MADE IN SFTAM

- Each Cluster head (CH) node in WSNs must determine if it is a retreat or an attack.
- This cannot be reversed after decision has been made.
- All cluster member (CM) nodes in the WSN must decide on the same Cluster Head (CH) decision and execute it synchronously.
- CH should be structures for the self-discovery of error and self-recovery.

### 3.3 ALGORITHM FOR AGREEMENT MADE IN BFT FOR OM (0)

1. *For any CHs, Algorithm OM (CHs) reaches consensus if there are more than 3CHs and at most m followers.*
2. Let CH be the leader and F[i] be the follower i
3. CH(L) sends valves to all followers
4. F1 sends valve to F2 | F3 sends x to F2
5. F2 ← majority(v,v,x) == v
6. The final decision is the majority vote

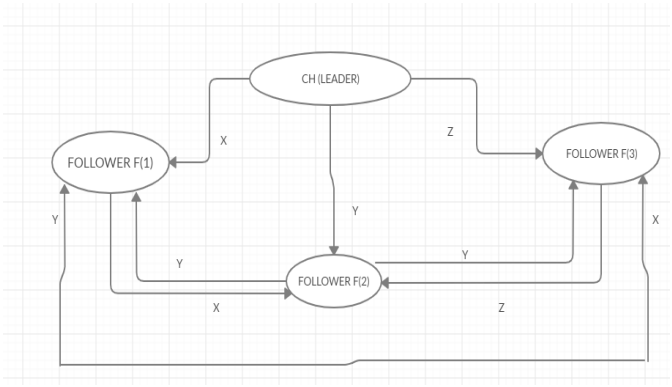



OM (1) FOLLOWERS 3 is a traitor, F2 is a point of view

The key thing to remember is that the aim is to go with the same decision for the majority of followers, not one particular one. The ultimate decision is the majority vote from F1, F2, and F3 and consensus was reached as a result.

### 3.4 ALGORITHM FOR AGREEMENT MADE IN BFT FOR OM (1)

1. *For any CHs, Algorithm OM (CHs) reaches consensus if there are more than 3CHs and at most m followers.*
2. Let CH be the leader and F[i] be the follower i
3. CH(L) sends valves to all followers
4. F1 sends x valve to F2,F3 |F2 sends y to F1,F3 |F3 sends z to F1,F2
5. F1 ← majority(x,y,z) F2 ← majority(x,y,z) F3 ← majority(x,y,z)
6. They all have the same value and thus consensus is reached
7. Default option *retreat.*

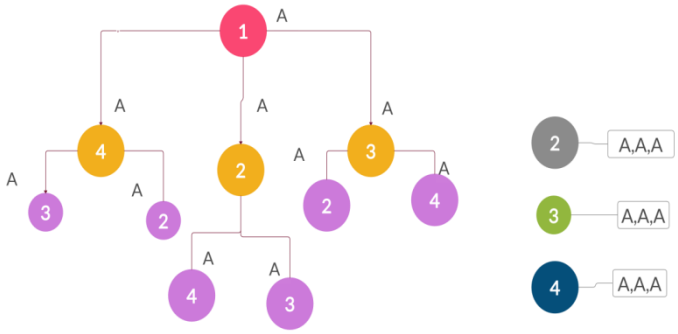



CH (leader) is a traitor

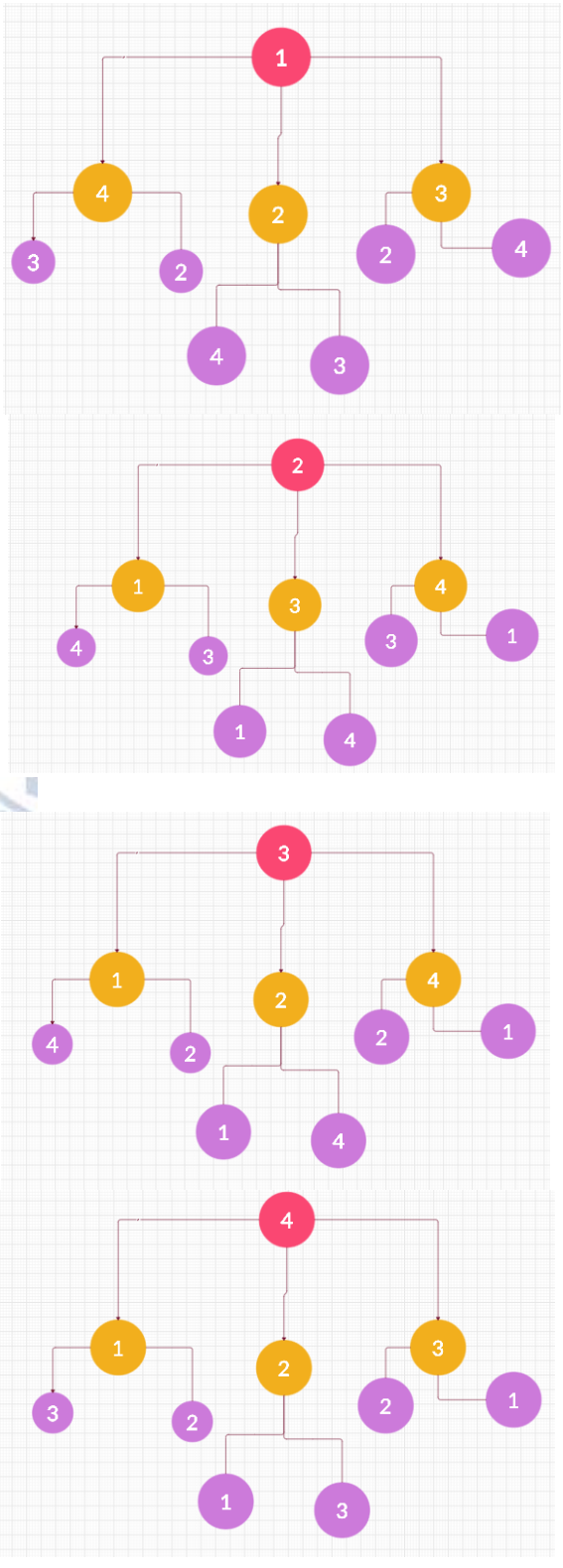
They all have the same interest and thus reach consensus. Pause for a moment here to reflect that the value of majority(x, y, z) is the same for all 3 FOLLOWERS while x, y, z is all different. In the case of x, y, z being completely different commands, we may assume they operate on the default retreat option.

### 3.5 FAULT IDENTIFICATION IN SFTAM

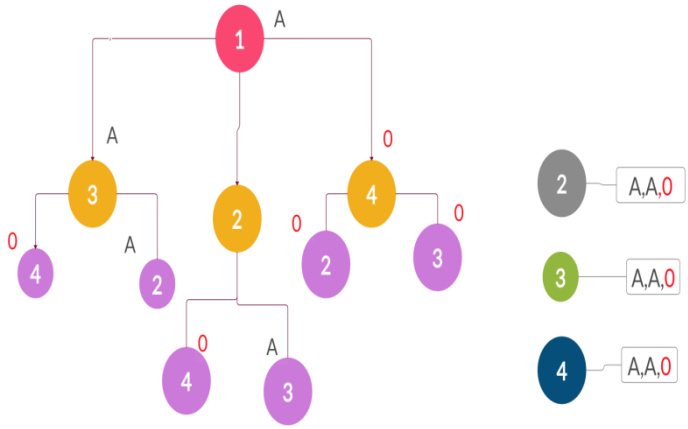
The diagram shows a network with M number of nodes, with some sensors connected for each redundancy system. The LEADER then transmits what it has read from its sensor to all other nodes in the system known as "FOLLOWERS." The FOLLOWERS then transfer the value they have obtained from the LEADER to all the other FOLLOWERS and then vote on the results. A vote must be based on a majority as all nodes should have received the same values. The cycle repeats until all nodes have become LEADER, then all modules wait for an interrupt to start the process again at this stage. The flow of data can be viewed by the following image:




Each numbered bubble in the above image represents one node which acts as a LEADER in a system with 4 followers. The diagram illustrates the flow of information while node 1 is the LEADER. The flow of data for the entire process with all nodes can be seen in the following diagram:



As already mentioned, a system with 4 nodes can tolerate 1 network fault. We simulate device flaws for our design by inserting an error in the communication channel. When we model our method with simple binary data, a bit of a flip would result in the error. We could see how a mistake propagates through network while in the diagram below, node 1 is general:




**Link error between node 1 and node 4**

From the diagram above, we can see a data transfer error occurring between nodes 1 and 4. Node 1 reads its sensor value 1, and node 4 receives a 0. Because node 4 is not defective, the value it got from the general will be sent to other followers in the network assuming it is the right value to send. That we see as a result is that the output of each node being voted is still right. When they do a majority vote based on the data they will got the output matches what the leader reads from their sensor. We can also explain a case in which our system will crash, which is when there are two faults.We can also explain a case in which our system will fail, which is when there are two faults. Looking at the data transfer with two inserted faults, we can see that each of the followers has provided a plurality of faulty data resulting in incorrect data output.

## IV. BYZANTINE FAULT TOLERANCE (BFT) AND ITS WORKING

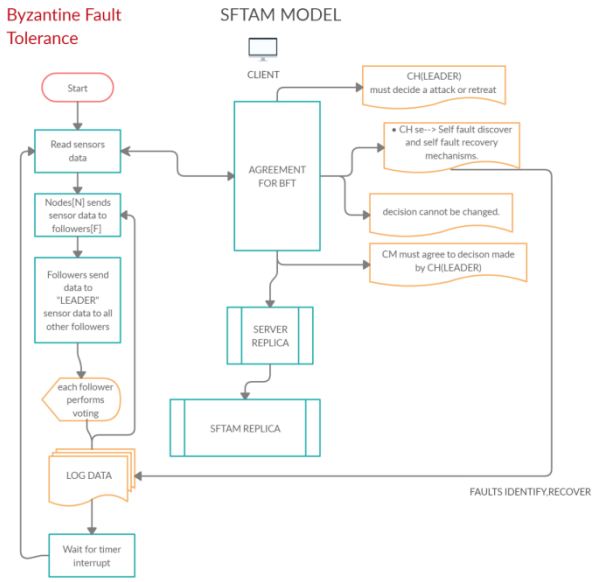
Byzantine fault tolerance is a method of managing faults which can ensure correct system performance under the following circumstances: the device must have 3*J+1 fault handling node; if the sender is not defective, the recipients must obtain the correct values; all recipients agree on the same value as the SFTAM agreement model. Nodes in a distributed system allowed by BFT are ordered sequentially with one node being the leader node, and others being named the backup nodes. The purpose is to help all honest nodes reach consensus about the condition of the system using voting system made according to our SFTAM model.

A Byzantine Fault Tolerant framework may operate on the condition that the maximum number of malicious nodes in the network should not be greater than or equal to one-third of all nodes. The

network becomes more stable, as the number of nodes increases.BFT consensus rounds are split into 4 phases

- The client sends an application that serves as a leader to the CH node.
- The leader node transmits the request to all the members of the Cluster (CM) stored in server replica.
- During each view built according to our agreement model the CH leader node is updated.
- If our agreement model so requires, most nodes will vote on the validity of the current lead node and replace it with the next lead node in line.

In the following block diagram one implantation of SFTAM in byzantine fault tolerance is shown below. Here SFTAM Model is correlated to the BFT algorithm, in which SFTAM is setup with the agreement model and there replicas are being stored in the server side. The procedure for the BFT algorithm is as follows with the SFTAM Model, from which faults are identified and diagnoses, thus achieving fault tolerant.




**SFTAM MODELWITH A BFT**

## V. CONCLUSION

The main purpose of this paper is to identify the faults and achieve fault tolerant in WSNs. For this we designed a secure fault tolerant agreement model (SFTAM) in which an agreement is made for identifying and recognizing the faults occurs in WSNs. After this identification of faults, we need our system to be fault tolerant so as to manage faults to get the accurate results of the faulty nodes, our SFTAM model of agreement is given to BFT algorithm. Thus, faults are identified using agreement model of SFTAM and fault tolerant is achieved using BFT algorithm which offers high capability of good network connectivity at all nodes and sink levels in WSNs.

## VI. REFERENCES

[1] Patton R. J. and J. Chen, "Observer-based fault detection and isolation: robustness and applications," Control Eng.Practice, Vol. 5, Issue 5, May 671 (1997)

[2] Iri, M., K. Aoki, E. O'Shima and H. Matsuyama, "An Algorithm for Diagnosis of System Failures in the Chemical Process," Comput. Chem. Eng., 3, 489 (1979)

[3] Clark R. N., "A simplified instrument failure detection scheme," IEEE Trans. Aerosp. Electron. Syst., 14(4), 558 (1978b)

[4] Harte, S., & Rahman, A. (2005). Fault tolerance in sensor networks using self-diagnosing sensor nodes. In The IEEE international workshop on intelligent environment (pp. 7–12).

[5] Singh, A. K., & Purohit, N. (2014). An optimised fuzzy clustering for wireless sensor networks. International Journal of Electronics, 101(8), 1027–1041.

[6] Swain, R. R., & Khilar, P. M. (2017). Composite fault diagnosis in wireless sensor networks using neural networks. Wireless Personal Communications, 95(3), 2507–2548

[7] Ludeña-Choez, J., Choquehuanca-Zevallos, J. J., & Mayhua-López, E. (2018). Sensor nodes fault detection for agricultural wireless sensor networks based on NMF. Computers and Electronics in Agriculture.

[8] Cheraghlou, M. N., Khadem-Zadeh, A., & Haghparast, M. (2017). Increasing lifetime and fault tolerance capability in wireless sensor networks by providing a novel management framework. Wireless Personal Communications, 92(2), 603–622.

[9] Karim, L., Nasser, N., & Sheltami, T. (2009). A fault tolerant dynamic clustering protocol of wireless sensor networks. In IEEE communications.

[10] Gupta, G., & Younis, M. (2003). Fault-tolerant clustering of wireless sensor networks. Wireless Communications and Networking, 3, 1579–1584.

[11] G. Venkataraman, S Emmanuel and S.Thambipillai, "Energyefficient cluster-based scheme for failure management in sensor networks" lET Commun, Volume 2, Issue 4, April 2008 Page(s) 528 – 537