ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

Fault Detection Mechanism for Wireless Sensor Networks

Smita Jangale, Dhanashree Hadsul
Department of Information Technology, V.E.S. Institute of Technology, Mumbai, Maharastra

Abstract— Due to the low cost and the possible harsh or hostile deployment environments, sensors are prone to failure. Faulty sensors are likely to report arbitrary readings that do not reflect the true state of observed physical process. These faulty sensors should be recognized timely, and should be excluded from the data collection process to ensure the overall data quality. In this paper an Ad hoc On Demand Distance Vector (AODV) routing algorithm is used to create wireless sensor network. The proposed fault detection algorithm is based on the spatio-temporal correlations among the sensor measurement series in WSNs. Each sensor node identifies its own state based on similarity test and the dissemination of the node state. The algorithm is computationally affordable and can detect faulty sensor nodes with high detection accuracy and low false alarm rate.

***Index Terms* — AODV, Coefficient Correlation, WSN.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of numerous tiny sensor nodes that are deployed in spatially distributed environment. Each sensor node is having a limited amount of processing, but when coordinated with the information from other nodes, they have the ability to measure the given physical environment in great details or to execute a task with complex functions. Hence, a sensor network can be described as a collection of sensor nodes that coordinate with each other to perform some specific actions. Since each sensor node is fitted with an on-board processor, sensor nodes use their processing abilities to find out simple computations and transmit only the required data. These features allow the sensor networks to use in many applications, like military, security and environment. Wireless sensor networks can also be deployed in the ways that the wired sensor system cannot be deployed such as in the chemical environments that are inaccessible by humans. Wireless sensor networks contain a lot of constrains, such as energy limitation, decentralized collaboration, and fault tolerance. The major aim of this paper is to find the faulty nodes with similarity test. The rest of paper is organized as follows. Section 2 presents the AODV routing protocol. In section 3 the problem is stated and network and faults model with assumptions is discussed. Section 4 explains the suggested fault detection algorithm in detail. Section 5 demonstrates simulation results. Complexity of our algorithm is calculated in section 6. In Section 7 some superiorities of the proposed method are explained and finally in section 8 this paper is concluded.

II. AODV ROUTING PROTOCOL

The AODV Routing protocol uses an on-demand approach for finding routes, that is, a route is established only when it is required by a source node for transmitting data packets. It employs destination sequence numbers to identify the most recent path. AODV offers quick adaptation to dynamic link conditions, low processing and memory overhead, low memory utilization, and determines unicast routes to destinations within adhoc network. The major difference between AODV and Dynamic Source Routing (DSR) is that DSR uses source routing in which a data packet carries the complete path to be traversed. The message types defined by the AODV protocol are Route Requests (RREQs), Route Replies (RREPs) and Route Errors (RERRs). However, in AODV, the source node and the intermediate nodes store the next-hop information corresponding to each flow for data packet transmission. In an on-demand routing protocol, the source node floods the RREQ packet in the network when a route is not available for the desired destination. It may obtain multiple routes to different destinations from a single RouteRequest. The major difference between AODV and other on-demand routing protocols is that it uses a destination sequence number (DestSeqNum) to determine an up-to-date path to the destination. A node updates its path information only if the DestSeqNum of the current packet received is greater than the last DestSeqNum stored at the node.A node offers connectivity information by broadcasting local Hello messages as follows.[6] During every Hello interval milliseconds, the node checks whether it has sent a broadcast within the last Hello_Interval. If

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

it has not sent one, it broadcasts a RREP with TTL = 1, called a Hello message, with the RREP message fields set as follows: The destination IP Address would be the node's IP address, the destination Sequence Number would be the node's latest sequence number. The value of hop count would be the Lifetime Allowed_Hello_Loss * Hello_Interval. AODV makes sure these routes do not contain loops and tries to find the shortest possible route. AODV is also handles changes in routes and can create new routes if there is an error. These message types are received via UDP, and normal IP header processing is applied. So, the requesting node is expected to use its IP address as the Originator IP address for the messages. For broadcast messages, the IP limited broadcast address (255.255.255.255) is used. AODV operation does require certain messages (RREQ) to be disseminated widely. The range of dissemination of such RREQs is indicated by the TTL in the IP header. Nodes which could communicate with directly are considered to be Neighbors. A node keeps track of its Neighbors by listening for a HELLO message that each node broadcast at set intervals. When one node needs to send a message to another node that is not its Neighbor it broadcasts a RREQ message. The RREQ message contains several key bits of information: the source, the destination, the lifespan of the message and a Sequence Number which serves as a unique ID. In Figure 2, [2] Node 1 wishes to send a message to Node 3. Node 1's Neighbors are Nodes 2 + 4. Since Node 1 cannot directly communicate with Node 3, Node 1 sends out a RREQ. The RREQ is heard by Node 4 and Node 2. When Node 1's Neighbors receive the RREQ message the nodes have two choices; if they know a route to the destination or if they are the destination they can send a RREP message back to Node 1, otherwise the nodes will rebroadcast the RREQ to their set of Neighbors. The message keeps getting rebroadcast until its lifespan is up. If Node 1 does not receive a reply in a set amount of time, it will rebroadcast the request except at this time the RREQ message will have a longer lifespan and a new ID number. In AODV, the network is silent until a connection is required. The network node that needs a connection broadcasts a request for connection. Other AODV nodes forward this message, and record the node that they received it from, creating an explosion of temporary routes back to the node which requires the connection. This node then begins using the route that has the least number of hops through other nodes. Unused entries in the routing tables are recycled after a time. The main advantage of this protocol is that routes are established on demand and destination sequence numbers are used to find the latest route to the destination. The connection setup delay is less. AODV provides loop-free routes even while repairing periodic routing advertisements, the demand on the overall bandwidth available to the mobile nodes is substantially less than in the protocols that do necessitate such advertisements.

III. SIMILARITY TEST

Similarity can be roughly described as the measure of how much two or more objects are alike. Similarity can also be seen as the numerical distance between multiple data objects that are typically represented as value between the range of 0 (not similar at all) and 1 (completely similar). Depending on the similarity metric used the triangle inequality between objects may hold, but more generally the two properties that must be maintained for similarities is that the measure of similarity must fall within the range of 0 and 1 and symmetry. Symmetry being the property that states that for all x and for all y the similarity of x and y must be the same as the similarity of y and x[1].

Extended Jaccard Coefficient is used to compare documents. It measures the similarity of two sets by comparing the size of the overlap against the size of the two sets. Should the two sets have only binary attributes then it reduces to the Jaccard Coefficient. As with cosine, this is useful under the same data conditions and is well suited for market-basket data .We have selected Jaccard-coefficient measurement because it satisfies many mathematical properties. Which is equivalent to the binary version when the features vector entries are binary [2].

$$j_{kl} = \frac{x_k \bullet x_l}{\left\|x_k\right\|^2 + \left\|x_l\right\|^2 - x_k \bullet x_l}$$

IV. FAULT TOLERANCE IN WIRELESS SENSOR NETWORKS

Fault tolerance is the ability of a system to respond gracefully to an unexpected hardware or software failure. There are three aspects of fault tolerance, namely, fault models, error detection and diagnosis techniques, and resiliency mechanisms. In this paper we are mainly concentrating on one of the major aspect of fault tolerance mechanism i. e. error detection. The algorithm detects faulty nodes in wireless sensor networks. For detecting faulty nodes similarity test is carried out between every node. Extended jaccard coefficient is used to calculate the similarity

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

between two nodes.Before doing similarity computation, we should normalize the original measurement series into the range of [0, 1]. Currently we use

$\bar{\delta} = (d - d_{min}) / (d_{max} - d_{min})$ as the normalization

Method. d stands for an original sensor reading, while d_{max} and d_{min} represent the maximum and minimum of the measurement series respectively.

Threshold value Θ is provided which is defined by the administrator. If $Jkl > \Theta$ than the similarity test succeeds. Every node in the network maintains its own Network Table (NT). This table consists of sensor id, similarity correlation between neighbouring node and the state of the node. Sensor id provides the unique identity of the of the sensor node in wireless sensor networks. Similarity correlation is the similarity test result carried out between the node and its neighbouring node. To calculate similarity test we have used extended jaccard coefficient. While the last column presents the state of the neighbouring node as per the comparison of similarity correlation with the threshold value. The NT table shows three states of nodes unknown, good, faulty [5].

Table 1: NT table

Node state with number	Node state
0	Faulty
1	Good
2	Unknown

V. SIMULATION TOOLS

Simulation is defined as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for the operation of the system. Ns-2 is a packet-level simulator and essentially a centric discrete event scheduler to schedule the events such as packet and timer expiration. Centric event scheduler cannot accurately emulate "events handled at the same time" in real world, that is, events are handled one by one. The C++ classes of ns-2 network components or protocols are implemented in the subdirectory "ns-2", and the TCL library in the subdirectory of "tcl". NS2 is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours [3]. NS-2 provides the Network components like Node, Link, Queue, etc. they are created from the corresponding C++ classes; The other are compound components, that is, they are composed multiple simple C++ classes like Link are composed of Delay (emulating propagation delay) and Queue. We can say that in ns-2, all network components are created, plugged and configured from TCL. NS-2 provides the Event Scheduling that is associated with time. class Event is defined by {time, uid,next, handler}, where time is the scheduling time of the event, uid is the unique id of the event, next is the next scheduling event in the event queue that is a linklist, and handler points to The function to handle the event when the event is scheduled. Events are put into the event queue sorted by their time, and scheduled one by one by the event scheduler [4].The version in use for this dissertation is version 2.34, with installation of the all-in-one package that operates on Linux. The routing protocol is implemented using C++ and the scenarios are implemented with scripts written in TCL that comprise commands and parameters for simulator initialization, node creation and configuration. The basic parameters needed for simulation are the movement pattern file, the communication pattern file and the configuration file. The movement pattern file describes all node movements while the communication pattern file describes the packet workload presented at the network layer during simulation. These two files essentially constitute the description of the simulation scenario. The final input is the configuration file that defines the ad hoc network routing protocol which is often the main file where the scenario files are called. The procedure for running the scenarios is shown in Figure 1.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

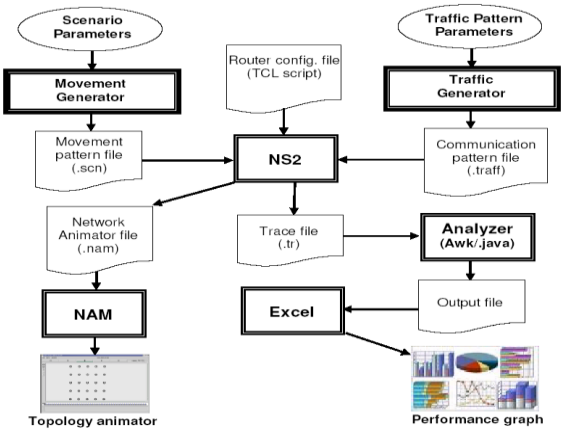

Fig 1: Flow Diagram for Running Scenerio in NS-2

As described above there are two types of simulation results a text based output file known as trace file and graphical based file. The major component of the Ns-2 simulator is the event scheduler. Each packet in Ns-2 is unique and has its own Packet ID. The event scheduler recognizes packet by its Packet ID and fire all the events in the event scheduler queue for the current time invoking the appropriate network components.

VI. PROJECT METHODOLOGY

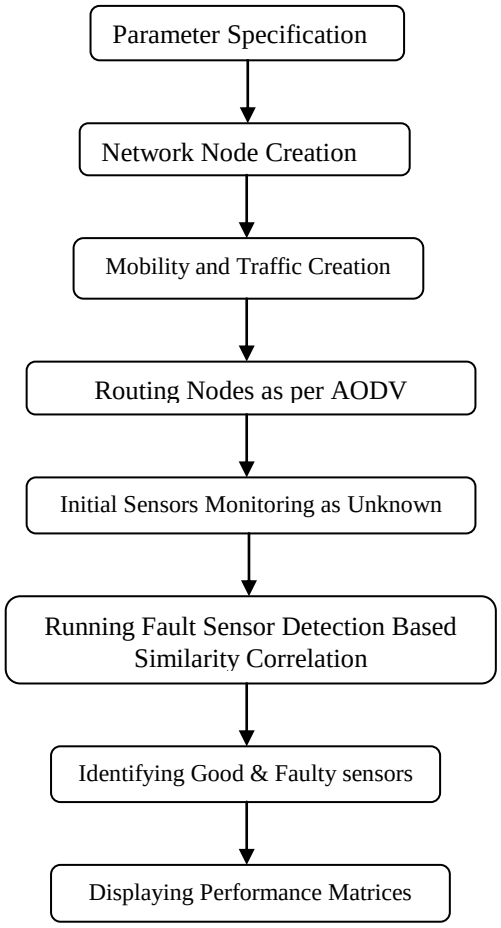

Fig 2: Project Methodology

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

We have used ns 2.34 network simulator for simulation. We simulate a sensor network deployed to monitor the temperature of an area of 1500m * 1500m. For simplicity, we assume sensor nodes are randomly distributed in the area. In this area 50 nodes are distributed randomly. Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed. The main focus of this paper is to focus on the detection of faulty wireless sensors in particular network. In mobility and traffic generation we have define the type of antenna, the radio-propagation model, the type of ad-hoc routing protocol used by mobile nodes. In traffic model Continuous bit rate (CBR) traffic sources are used. The source-destination pairs are spread randomly over the network.

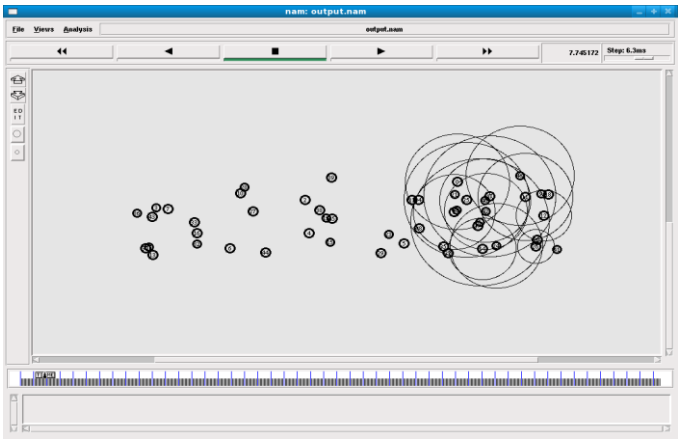

Fig 3 : NS 2 Simulation of WSN

AODV algorithm helps in transferring the packet in the WSN. A WSN is created using AODV and the routing of packets is by the AODV algorithm. Fault detection algorithm uses the distributed algorithm for detection of faulty nodes in wireless sensor networks. The sense value is used for the calculation of coefficient correlation. Initially all sensor nodes are in unknown state. Then every node starts building a network table. The table consists of the identification number of the neighboring nodes. The similarity test results are provided with coefficient correlation. Finally the state of the node is given. It indicates the current faulty state of the node as discussed in section 4. After this entire setup the fault detection algorithm is carried out to find the faulty nodes with the help of jaccard coefficient correlation.

Table 2: Node state table

Sensor ID	SimiCorr	NodeState
1	1.00	1
2	0.34	2
3	0.85	1
4	0.01	2
5	0.46	0
6	0.78	1

VII. EXPERIMENTAL RESULTS

The results that we have found are related to packet delivery ratio, false negative rate and false positive rate.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 2, Issue 2, March 2013

Packet delivery ratio: Packet delivery ratio is defined as the ratio of data packets received by the destinations to those generated by the sources. This performance metric gives us an idea of how well the protocol is performing in terms of packet delivery at different speeds using the above traffic models.

PDR = ∑ Number of packet receive / ∑ Number of packet send.

Results related to false negative rate and false positive rate are also analyzed with the previous results.

The false positive rate is the proportion of absent events that yield positive test outcomes, i.e., the conditional probability of a positive test result given an absent event.

False Negative rate means that how many percentages of the authentic test samples are incorrectly classified as the imposter class.

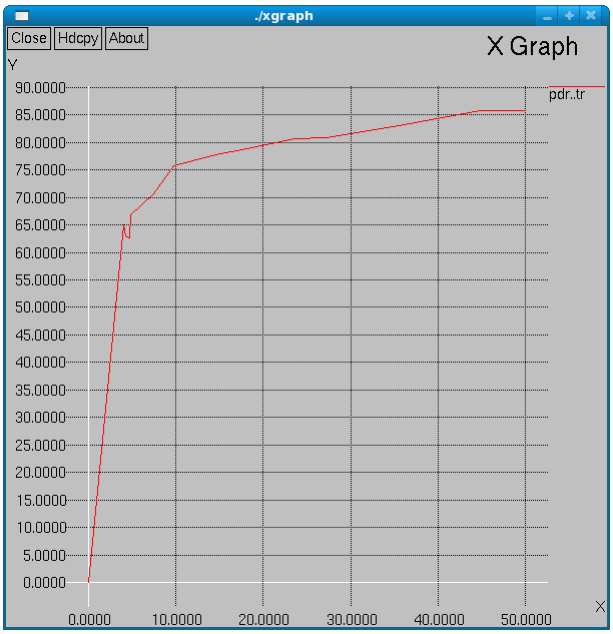

Fig 4 : Packet Delivery Ratio

VIII. CONCLUSION

A distributed detection algorithm is used in the fault tolerance mechanism. Similarity test detects the faulty sensor nodes in wireless sensor networks. While performing the detection AODV algorithm is used to do the packet management activity. In future work, fault detection algorithm to tolerate transient faults in sensor reading and inter-node communication can be extended and modified.

REFERENCES

[1] http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/bfindley/similarity.html

[2] http://inside.mines.edu/~ckarlsso/mining_portfolio/similarity.html

[3] Thammakit Sriporamanont and Gu liming," Wireless Sensor Network Simulator", Technical report, IDE0602, January 2006.

[4] Sukumar Panda, Rahul Mohapatra," implementation and Comparison of Mobility Models In Ns-2", National Institute of Technology, Rourkela 2009.

[5] "Sensor Fault Detection in Wireless Sensor Networks" Http://ieeexplore.ieee.org/ stamp/stamp.jsp ?tp=&arnumber=5522073

[6] http://inside.mines.edu/~ckarlsso/mining_portfolio/similarity.html

[7] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Published by Addison Wesley.