Open access Journal International Journal of Emerging Trends in Science and Technology

An Efficient Fault Tolerance Algorithm for wireless Sensor Networks

Authors

Syed Yaseen Ilahi[1], Santhy Ajish[2], Rakshith K[3]

[1]Mtech Student, Dept of Electronics and Communication, MIT, Mysuru, Karnataka, India

[2]Asst. Professor, Dept of Electronics and Communication, MIT, Mysuru, Karnataka, India

[3]Research Scholar, Dept of Electronics and Communication, MIT, Mysuru, Karnataka, India

Email: *yaseen.ilahi@gmail.com, santhyajish@yahoo.com, raksytk@gmail.com*

Abstract

Wireless sensor network (WSN) is an infrastructure comprising of thousands of wireless sensor nodes that are spread over a geographical area so as to instrument, observe and react to events in that particular environment. Sensor networks are the key for smart environments, analyzing events in buildings, utilities, industrial, home, shipboard, transportation systems automation or elsewhere. The reliability of WSN is affected by faults that may occur due to various reasons such as malfunctioning hardware, software glitches, dislocation, or environmental hazards. In WSN, localization of sensors is the key issue, as collected data from sensors is useful only if the position of a sensor is known. Location information will also help in avoiding faults in the network and improving performance of tasks such as routing, energy conservation, and data aggregation. Knowing the precise location of a sensor is very important. Localization algorithms require techniques for location estimating depending on the beacon nodes location. These are called multi-lateration(ML) techniques. The work makes use of proximity technique and fault information signal to get localization details.

Keywords: *fault information signal, localization, proximity technique, wireless sensor networks.*

1. INTRODUCTION

A wireless sensor network is a network of wireless sensors used to watch out events in the environment and then forwarding data to the destination. Each event will be analyzed and then action will be taken. Each sensor will have a radio transceiver with an antenna for communication, a microcontroller for fusing sensors and an energy source such as a battery.

WSNs are adhoc networks in which there will be no topology, fixed infrastructure and they will be deployed randomly to perform tasks.

Sensors used are inexpensive, power efficient and has the capacity to communicate with other sensors. Every time a node needs to communicate with other node then both nodes should be in communication range of one another, otherwise a node should use intermediate nodes to forward data to the destination node. The first networks were designed with the hardware as the main concern and the software as an afterthought. This strategy no longer works. Network software is now highly structured.

These networks are called broadcast networks in which there will be a shared media by which these will communicate. Each sensor will make use of reliable protocols to make sure that data will reach to the destination. Sensors will use TCP/UDP protocol to have trusty communication, MAC 802.11 for sensing the media and Ad Hoc On Demand Vector (AODV) as routing protocol. The rest of the paper is organized, as follows. Section II, deals with faults that occur in WSNs. Section III will introduce the concept of localization. Section IV would talk about routing protocol used in the work. Section V will introduce implementation

details using ns2, a network simulator. Section VI will talk about related work to show tolerance for faults in WSN. Section VII will brief about the model. Section VIII and section IX will talk about parameters. Section X will brief the results. Section XI will give the conclusion of the work.

2. FAULTS IN WSN

Nodes in WSNs fail to communicate with each other if there are faults in nodes. Fault in node occurs when there is failure in hardware or software part of a node. Nodes can't communicate with each other if there is a fault in link between nodes and any environmental change causes fading.

Fault tolerance in the network is capability of network to perform assigned tasks, even if fault occurs. Fault tolerance can be incorporated at certain levels they are hardware, software, network and application level.

2.1 Hardware level

Faults at this level happens in node if there is any defect in the hardware element of a sensor node such as defects in battery, memory unit, processing unit, sensing unit and network interface unit. Defects in node occur because elements used in nodes are not off good quality, use of battery gives energy constraints on trusty performance of sensor nodes, sensors will be deployed in an uneven environment.

2.2 Software level

The software, part of a node has two elements: middleware in which communication, routing, and aggregation can be done and a system software, such as operating system. Here system software has to provide simultaneous execution of localized algorithms in a distributed manner.

2.3 Network level

Shortcomings in this level are because of faults on the wireless links. These flaws happen because of fading and obstructions in the environment. Shortcomings can be avoided by error checking and retransmission of data.

2.4 Application level

Fault tolerance at this level depends on different application. For example, finding multiple paths, provide fault tolerance in routing data to the destination.

3. LOCALIZATION

Localization is finding the location of nodes whose location is unknown in the network. One of the most crucial tasks is to have proper location information of nodes so that nodes can improve the performance of tasks such as energy conservation, routing, data gathering etc. Data from particular sensors is useful if location information of those sensors is known. This has to be done in distributed manner means every node has to calculate its location by itself. As nodes are deployed in random manner use of global positioning system (GPS) system for localization are expensive in terms of energy consumption.

Steps for computing localization in WSN

a) Distance evaluation: - In this distance between nodes are calculated using measurement techniques.

b) Position estimation: - Here the coordinates of unknown node are calculated with respect to beacon nodes. The algorithm which uses beacon nodes to estimate position are called multilateration techniques.

4. ROUTING PROTOCOL WORKING

To determine a suitable path over which data is to be transmitted, a node will make use of routing protocol. The Protocol will also specify how node can share information with each other and address changes in the network. The work makes use of AODV for routing.

The AODV is an active protocol; consequently, routes are decided only when needed.

Hello messages are used to find neighbors which help in forwarding packets to the destination. Each active node periodically broadcasts hello messages that all its neighbors within range can receive and if a node fails to receive a certain number of hello messages, then a link break is detected.

When a source has some data to transmit to an unknown destination, which is not in the range of

source, source broadcasts a Route Request (RREQ) for that destination. If the receiving node receives the RREQ first time and if it is not a destination node, then it will rebroadcast RREQ to its neighbors. If the receiving node is the destination or has an existing route to the destination, it generates a Route Reply (RREP) then RREP will be forwarded in a hop-by-hop manner to the source which is unicast in nature. As the RREP propagates, each interceding node creates a route to the destination. When the source receives the RREP, source archives the route to the destination and can begin sending data. If diverse RREPs are received by the source, the route with the precise hop count is elected. As data moves from the source to the destination, each node along the route refreshes the timers associated with the routes to the source and destination, conserving the entries in the routing table. If a route is silent for some span of time, a node cannot be definite whether the route is still genuine; therefore, the node takes off the entry from its routing table. If data is uninterruptedly flowing and a link break is discovered, a Route Error (RERR) is sent to the source in a hop-by-hop manner. As the RERR moves towards the source, each intermediate node cancels entries to any inaccessible destinations. When the source receives the RERR, it cancels the route and reinitializes route discovery if necessary.

4.1 AODV packet information

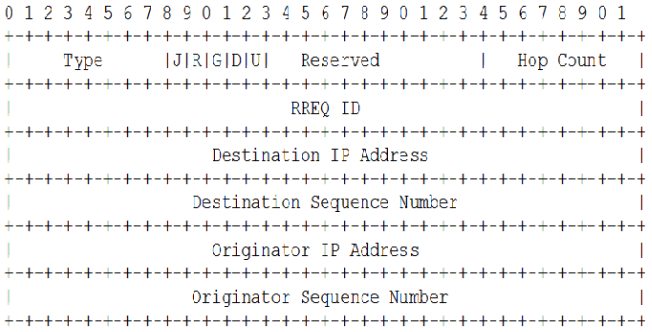

Figure 1: AODV packet

Type: 1

Hop Count: The number of hops from the Originator IP Address to the node handling the request.

RREQ ID: A sequence number uniquely identifying the particular RREQ when taken in conjunction with the originating node's IP address

Destination IP Address: The IP address of the destination for which a route is desired.

Destination Sequence Number: The latest sequence number received in the past by the originator for any route towards the destination.

Originator IP Address: The IP address of the node which originated the Route Request.

Originator Sequence Number: The current sequence number to be used in the route entry pointing towards the originator of the route request.

Table 1: AODV flags

Flag	Name	Description
J	Join flag	Reserved for multicast.
R	Repair flag	Reserved for multicast.
G	Gratuitous RREP flag	Indicates whether gratuitous RREP should be unicast to the node specified in the destination IP address
D	Destination only flag	Indicates only destination may respond to RREQ
U	Unknown specific number	Indicates the destination sequence number is unknown
Reserved	Reserved	Sent as 0; ignored on reception.

5. DATA FLOW DETAILS IN N S2

WSN analyzed using simulation experiments. Two known simulators are OPNET and NS2. Ns2 is open source software, which gives the flexibility of making changes. It will be easy to understand important details by view of data flow in ns2.

In general, source and sink are the generator and user of data packets. As shown in figure 2, each source or sink associate unique port number. Packets generated from a source agent and is directed to 'entry_'. The intention of doing this is that ns2 wants its 'self-generated' and 'receive data' to go through same standard. Then packets will be passed to address demux which is an illustration of class Dest Hash classifier derived from class classifier. If the destination of packets is a node, the packets should be sent to port demux. In port demux, different packets will be given to the corresponding sink agent; Port number 255 is reserved for routing agent. If no agent matches such a port number, the packets should be discarded. However, if the destination is not the node, the packets should be forwarded by the node, so it is redirected to the routing agent.

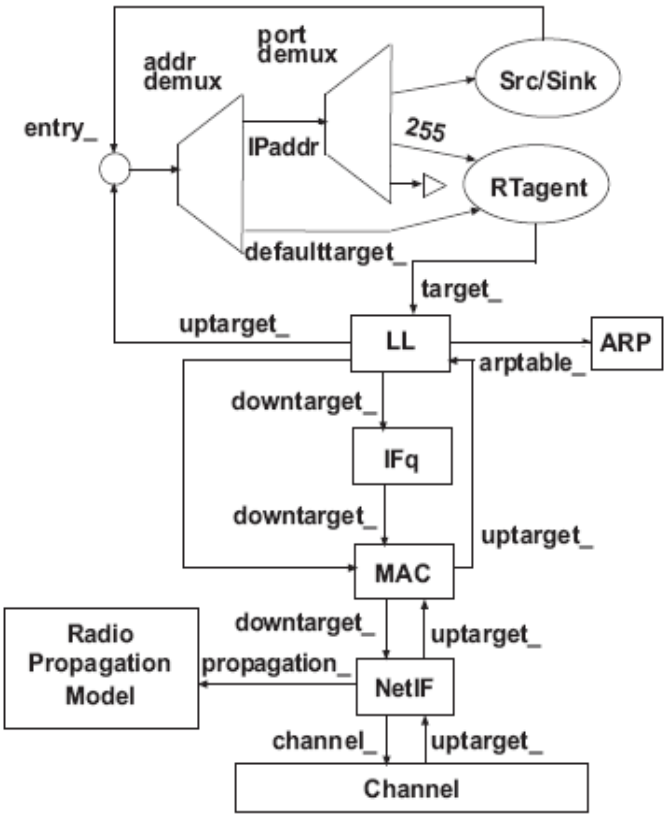

Figure 2: Data flow in ns2

The routing agent accumulates a routing table, and if routing tables contains the needed information, the packet should be put on link layer, if the routing information is not found; the route agent will deal with the packet according to different route protocols. For AODV, the routing agent will accumulate the packet first, and broadcast a request to find a route. Once the packets reach link layer, link layer looks up ARP table to map IP address to MAC address, then forward to network interface. MAC 802.11 will be used to sense medium and can perform collision avoidance. If the packet can transmitted safely, it will be forwarded to network interface. The network interface simulates the wireless card in the real world. Finally the modulated data will be transmitted over wireless channel. For the receiver node network interface, the packet will be delivered to link layer. Link layer convert MAC address to IP address, and direct the packet to 'entry-'.

6. RELATED WORK

Localizing and detecting faults in wireless sensor network have been most researched areas and providing tolerance to faults in the sense that even in faults the data has to reach to the destination. [1] Shows that fault tolerance can be done by considering events detected by sensors, here fault considered is that if some event is happening, then sensor has to detect an event, but due to fault in sensor it may not detect an event, then a sensor in a distributed manner provides binary result and faults can be avoided by fusing results of all sensors at fusion sensor. Then fusion sensor will provide a final decision so that, if the decision goes above some K value the event is present and vice versa. [2] Extended work by considering non binary DSCD approach where instead of doing binary level decision at sensor they used multilevel decision at each sensor and then by creating an M by N matrix at fusion sensor which provides final decision. Fault tolerance can also achieve of by having a K connectivity network [3] so that even the failure of K-1 nodes will not disconnect the network and data can still reach the destination. Fault tolerance at the hardware level [4] achieved by providing backup to primary sensors so that if the primary sensor fails, then backup sensor can take place of primary. Backup sensor can be in synchronism with primary or it may be not. [5] Shows that by using position

based protocols a location of each node is maintained at location servers and fault tolerance can be achieved by updating location servers periodically by a technique called synchronized aggregation.

The work in this paper done on fault tolerance, which can be achieved by first localizing all nodes using proximity technique and by sharing fault information signal (FIS)

7. SYSTEM MODEL

The network consists of a set of wireless sensors which have an ability to sense and detect events and even they have the ability to communicate and forward data. Each wireless sensor deployed randomly, so each sensor will hardly know its neighbors.

The AODV protocol is used for routing and MAC for sensing the medium, hence medium can be shared by and among the sensors.

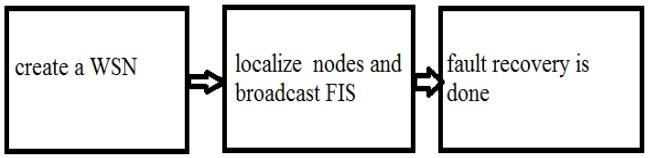

Figure 3: System model

In NS2 first step is to create a network of wireless sensors and each sensor will be equipped with AODV, MAC protocols for sensing medium and routing. In the NS2 simulator, it is possible to detect and correct faults in software, hardware and network communication layer.

If faults in any of layer mentioned above happen to then first know the location of the sensor where fault has happened to be found out, then same node or neighboring node will broadcast fault information signal (FIS). When FIS reaches source node, then source node has to overcome fault by re-initializing the route discovery. In this work the responsibility is to first localize nodes, generate and broadcast FIS if a fault is detected.

7.1 Switching in wireless sensor network

The work can be accomplished by making use of virtual connection between the source node and destination node so that every packet from the source node should have unique identification to use that virtual connection. At the core of network, switching of labels take place to get to destination and end network will be IP router. Virtual circuit path in WSN used is a label switch path (LSP), there is correspondence between LSP and forward equivalence class (FEC) i,e.. There can be different application flows or to simply aggregate all packets and make them to flow in a single LSP destined to same destination IP prefix.

This decision has to be taken at classification rules which should be applied at ingress of WSN. At ingress, packets classified into FEC class and each FEC class are mapped to label. To inform label switch routers (LSR) about label bindings to a particular FEC it makes use of the label distribution protocol.

Protection switching is done when a fault occurs at a primary path and fault information signal (FIS) reaches at ingress node.

7.2 Fault Information Signal (FIS) frame format

Fault information signal (FIS) will be generated if a fault occurs at some node, then same node or neighboring node has to indicate source node by generating FIS. A Source node, then has to switch to an alternate path to reach to a destination.

Broadcast Type	Fault Signal Source	Direction	Stack Field	TTL value

Figure 5: FIS frame

4 bit of identifier.
8 bit of fault Signal Source.
1/0 Upward Direction, Downward Direction
1 bit of bottom stack field.
8 bit of TTL value.

7.3 Label

A short fixed length contiguous identifier which is used to identify a FEC. Label attached between the IP header and link layer header. A basic operation taken by a label switch router is push, pop and swaps a label.

Figure 4: Label

20 bits of identifier.
3 bits of exp field.
1 bit of bottom stack field.
8 bit of ttl value.

7.4 Label allocation and distribution.

Label distribution can be done in two ways.

- Downstream based.
- Upstream based.

1. **Downstream based**: In downstream based the label allocation and distribution done by downstream LSR and packet flow takes from upstream to downstream.
2. **Upstream based:** In upstream based the label allocation and distribution done by upstream LSR and packet flow takes from upstream to downstream.

7.5 Algorithm for detecting fault and broadcasting fault information signal

1. Initialize all parameters.
2. Initialize timers.
3. If a node has data to transmit.
4. Then broadcast route request message.
5. A node if it is destination or have route to destination then node will reply using route reply message.
6. Select shortest path.
7. Select nodes which are ingress and egress to WSN network.
8. Select a group of packets belong to same FEC and attach label to it.
9. The downstream node which do all bindings have to inform upstream node by using label distribution protocol
10. The path to destination is called label switched path. All packets with same FEC has to go through same path. Update routing tables.
11. Each node has to perform push, pop, swap for labels.
12. Each node will make use of HELLO protocol to check connectivity among neighbors and wait about ALLOWED HELLO LOSS*HELLO INTERVAL period before deleting neighbor.
13. If a fault occurs on LSP, then same node or neighbor node will transmit FAULT INFORMATION SIGNAL (FIS) to ingress node.
14. After receiving FIS protection switching has to perform and update routing tables.

7.6 Diskjetras algorthim for Shortest path tree (SPT)

To have set of optimal routes from all sources to a given a destination, form a tree rooted to destination. Such a tree called a shortest path tree (SPT).

1. Intilization
2. D[s]=0 distance from s node to itself.
3. For each v€V-{s}
4. Then d[v]<—∞
5. S<—Ø
6. Q<—V it is priority queue, keyed on distance d
7. While Q ≠ Ø
8. Do<—u<— extract-min{Q}
9. If s<—sv{u}
10. For each v€adj[u]
11. Do if d[v]>d[u]+w[u,v]
12. Then d[v]<—d[v]+w(u-v)<—relaxation step

For each vertex and consider last edge (u,v) relaxed then put all together called a shortest path tree.

7.7 Localization of node

Nodes deployed randomly over a region of interest. Proximity between any two nodes i, j can obtained by measuring the Euclidean distance ‘d’.

$$d = \sqrt{(y2 - y1)2 + (x2 - x1)2} \qquad (1)$$

(x2 , x1) and (y1 , y2) are co-ordinates of nodes respectively. This information of the node will be included in the Fault information signal (FIS).

8. PARAMETERS

Table 2: Parameters

Parameter	Value
Topography	1000by1000
Keep alive interval	50
Gateway interval	25
Label switch path interval	10
Border gateway protocol	5
My route timeout	10
Active route timeout	10
Reverse route life	6
Route request retries	3
Route request timeout	10
Label distribution protocol interval	20
TTL start	5
TTl threshold	7
Node traversal time	0.03
Local repair wait time	0.15
Network diameter	3
Hello interval	1
Allowed hello loss	3

9. PERFORMANCE PARAMETERS

Throughput: throughput is the rate at which data go over a link. Throughput measured in bits per second or data packets over a time slot.

Transmission time=size of file/bandwidth

Throughput=size of file/transmission time

Table 3: Throughput table

Nodes	Throughput with fault recovery	Throughput without fault recovery
20	78.5369	36.8967
30	77.4806	33.4589
40	75.433	32.3472
50	74.421	31.8943
60	75.5498	38.5479

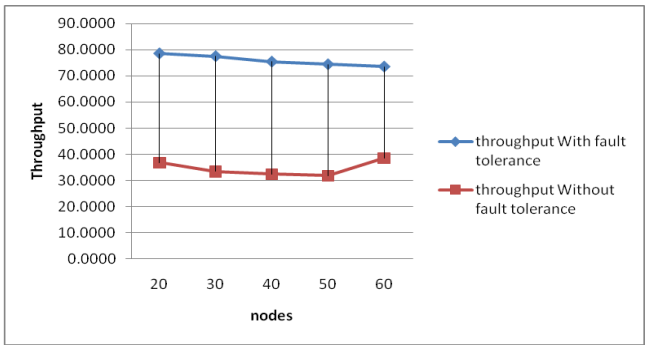

Figure 6: Xgraph of throughput

Jitter: In networking variation in arrival of packets at destination due to congestion in network.

Table 4: Jitter table

Nodes	Jitter with fault recovery	Jitter without fault recovery
20	1.10426	1.7856
30	1.10404	1.9978
40	1.11488	3.61718
50	1.20651	4.1050
50	1.10187	4.7429

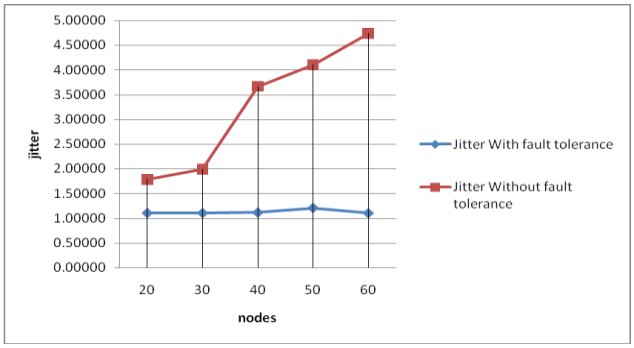

Figure 7: Xgraph of jitter

10. NAM OUTPUT

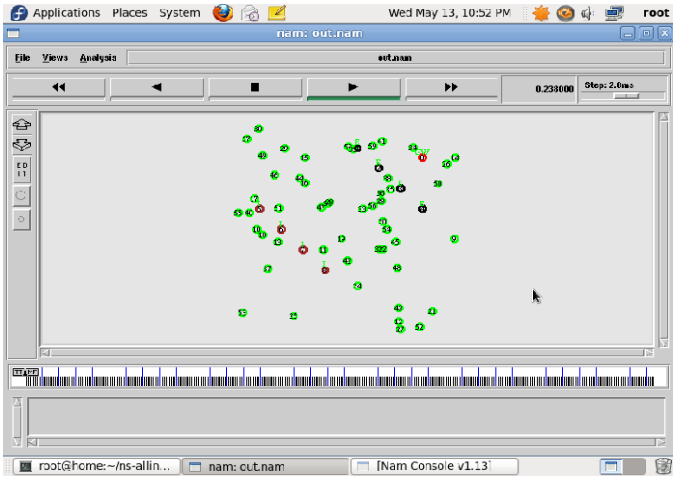

Figure 8: NAM window

11. CONCLUSION

The main motivation behind this work is to give a reliable and efficient fault tolerant algorithm for data protection in WSN when unexpected network failures occur.

The proposed system is able to detect multiple faults in the network and it also recover faults quickly.

REFERNCES

1. Xuanwen Luo, Ming Dong, Yinlun Huang" Optimal Fault-Tolerant Event Detection In Wireless Sensor Networks".
2. Tsang-Yi Wang, Yunghsiang S. Han, Pramod K. Varshney," Fault-Tolerant Distributed Classification Based on Non-Binary Codes in Wireless Sensor Networks" IEEE COMMUNICATIONS LETTERS, VOL. 9, NO. 9, SEPTEMBER 2005.
3. Jonathan L, Bredin Erik D,Mohammed Taghi "Deploying sensor networks with guaranteed fault tolerance"
4. LIUDONG XING1, HAOLI LI2, and HOWARD E. MICHEL1"Fault-Tolerance and Reliability Analysis for Wireless Sensor Networks" International Journal of Performance Engineering, Vol. 5, No. 5, October 2009, pp. 419-431.
5. Roie Melamed, Idit Keidar, Yoav Barel" A Fault-Tolerant and Effcient Ad-hoc Routing Protocol"
6. PritiNarwal, Dr. S.S. Tyagi"Position Estimation Using Localization Technique In Wireless Sensor Networks"International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 6, June 2013 ISSN 2319
7. Zehua Wang, Yuanzhu Peter Chen, Cheng Li" Implementation of the AODV Routing Protocol in ns2 for Multi-hop Wireless Networks".
8. Hai Liu, Amiya Nayak, and Ivan Stojmenovic" fault tolerant algorithm for wireless sensor networks".
9. Luciana Moreira Sa de Souza, Harald Vogt, Michael Beigl "A Survey on Fault Tolerance in Wireless Sensor Networks"