INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 5 ISSUE 2 – MAY 2015 - ISSN: 2349 - 9303

Improve a Network Life Time by Least Troublesome Topology Repair Algorithm in WSAN

R. Ananthakumar[1]
[1]PG scholar,[1]Kalasalingam Institute of Technology,
Dept of CSE
ananthk860@gmail.com

R.Sankar[2]
[2]Assistant Professor, [2]Kalasalingam Institute of Technology,
Dept of CSE
sankarwt@gmail.com

Abstract— In wireless sensor network, node reform and restoration is an active area for research. Each sensor node fails either when some critical event occurs at the node or when the battery of the node is completely drained. The Wireless sensor actor network (WSAN) is an enhances the wireless sensor networks by providing more power and recovering a energy consumption. In the WSAN, sensors sense the environment and sent back their data to an actor node. Actors have to coordinate their operations, keep up a firmly associated system topology at all times and achieve the predefined application mission. The Defect Node Reformation Algorithm (DNR) reform a defect node using genetic and the grade diffusion algorithms. A category of solutions for this problem is Least Troublesome Topology Repair Algorithm (LTTR). LTTR is a limited and circulated calculation that influences existing course revelation exercises in the system. LTTR algorithm is used to minimize the power consumption and reform a transmission path. Our results show that achieves performance in terms of packet delivery ratio, packet loss ratio and end to end latency suitable for real network deployments.

Index Terms— Defect Node Reformation (DNR), Least Troublesome Topology Repair (LTTR), Wireless Sensor Network (WSN), Wireless Sensor Actor Network (WSAN)

1 INTRODUCTION

Wireless sensor and actor networks (WSAN) consists of a group of sensors and actors linked by a wireless medium to perform distributed sensing and actuation tasks. In this network, sensors gather information about the environment, then actors take decisions and then takes appropriate actions departs the environment. This network is communicating the environment using remote and automated an interaction with the environment.

The Wireless sensor actor network consists of two types of node. The nodes are actor and sensor. An actor node collects the data from the environment and takes the actions according to the environment. An actor node covers the some region of the network. The Sensor node senses the particular environment and sent back to the data to actor. The Sensor node is totally controlled by an actor node.

In WSANs, the roles of sensor and an actor node are to collect data from the environment and perform appropriate actions based on this collected data, respectively. Thus, as shown in Fig. 1.1 these node are scattered in the sensor/actor field. In these figures constructs the parts are sink, task manager, sensor and actor. The sink which monitors the overall network and communicates with the task manager node and sensor/actor node, if necessary, is separated from the sensor/actor field. After sensors in the sensor/actor field, detect the actions, they either sent their readings to an actor nodes, which can process all incoming data and initiate suitable actions, or route data back to the destination which issues action commands to actors.

The Wireless sensor actor network is based on the node coordination. The node coordination Fig 1.2. is classified into the three types. The node coordination types are sensor to sensor, sensor to actor and actor to actor coordination. The Sensor to Sensor coordination is basically an interaction between the two sensor node. The source sensor sent the data to destination sensors and then destination sensors sent back to the data to particular an actor node based on sensor to actor coordination. The Sensor to Actor coordination is basically interaction between the sensor and an actor node,

- Which sensors communicate with which actors,
- How the transmission happens, and
- What are the requirements of this communication.

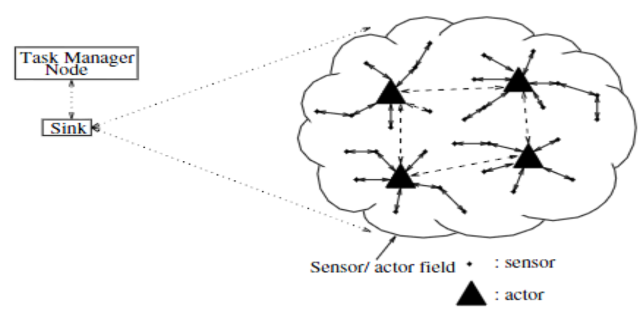

***Fig 1.1** The Physical Architecture of Wireless Sensor Actor Network*

An actor node is controlled the number of sensor node and responding based on the sensor node. An actor node is used to restore the connectivity and response to an actor nodes, which is based on actor to actor coordination. In Actor to actor coordination communicating between the actor to actor node. Actor to Actor coordination based on the communication can occur in the following situation

- The actor recollects the sensing data from the environment due to small range or insufficient energy.

- If multiple actors receive the same information, these actors should ''talk'' to each other so as to decide which one of them performs the action.
- In case of multiple events occurring simultaneously, task allocation can be done via actor to actor coordination.
- After an actor node receive event information, if the event is spreading to other actors' acting areas, an actor node can transmit the sensor data or action command to the particular actors. By this way, there will be no need for sensor in those areas to send information to their nearby actors.
- Which will considerably increase the network lifetime? Moreover, this situation provides the action to be initiated much earlier.

The property of the Wireless sensor Actor systems is,

- The sensor hubs are little, modest gadgets with restricted sensing, processing and remote correspondence abilities, performers are normally asset rich gadgets furnished with better transforming capacities, stronger transmission forces and more battery life.
- In WSANs, contingent upon the application, there may be a need to quickly react to sensor data. Also, to give right activities, sensor information must even now be substantial at the time of acting. Subsequently, the issue of ongoing correspondence is vital in WSANs since activities are performed on nature in the wake of sensing happens.
- The number of sensor hubs conveyed in considering a marvel may be in the request of hundreds or thousands. In any case, such a thick organization is redundant for performer hubs because of the distinctive scope prerequisites and physical communication routines for acting assignment. Thus, WSANs the quantity of on-screen characters is much lower than the quantity of sensor.
- In request to give powerful sensing and acting, a conveyed nearby coordination instrument is important between sensor and actors.

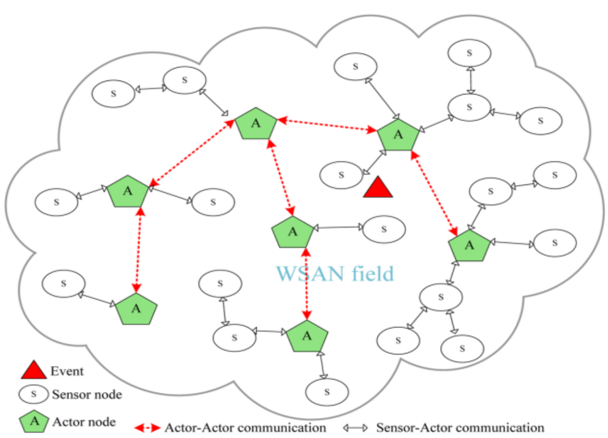

Fig 1.2 *Wireless Sensor Actor Network Coordination*

2 RELATED WORKS

In the Wireless Sensor Actor Network, the number of strategies has recently restored network connectivity. All of the strategies have focused on reconstructing the networks without extending the network topology. The several strategies are described in the below techniques.

A. Recovery Through Node Repositioning

The main idea of this category of recovery schemes is to reposition some of the strength node in the network to reestablish strong connectivity. LTTR fits in this category. Published approaches differ in the level of involvement expected from the strength node, in the required network state that needs to be constructed, and in the goal of the reforming purpose. For example, both Distributed Actor Recovery Algorithm (DARA) [5] and PArtition Detection and Recovery Algorithm (PADRA) [8] require every node to maintain a list of their two-hop neighbor and determine the scope of the recovery by checking whether the failed node is a cut vertex. When a node fails, the best node used for replacement is selected from its one hop neighbor. The node having lowest degree and least distant from the failed node is selected. Any child node disconnected during this process is recovered by the recursive relocation procedure. Thus the shortest path between the node is not maintained. DARA pursues a probabilistic scheme to identify cut vertices.

A good candidate (GC) is selected from the one-hop neighbor of the dead actor as a recovery initiator and to replace the defect node. This selection criterion is based on the least node degree and physical proximity to the defect node. The rearranging procedure is circular applied to handle any disconnected children. In other words, cascaded movement is used to sustain network connection. The another algorithm of PADRA identifies a connected dominating set to determine a dominate node. The dominate does not directly move to the location of the failed node instead, a cascaded motion is pursued to share the burden. In [6], the focus is also on recovering from the failure of a cut vertex. Only a special case is considered where the failures cause the network to split into two disjoint blocks. To relink these blocks, the closest node is moved toward each other. The other node in the blocks follows in a cascaded manner. None of these approaches care for the path length between node. While LTTR also employs cascaded relocation, the criteria for selecting the leader node and other participants are different.

B. Recovery By Placement of Relay Node

The prior calculations mean to restore the system integration by productively moving a portion of the current hubs. Notwithstanding, at a few setups, it is not doable to move the neighbor of the fizzled hub because of physical, logistical, and scope imperatives. Accordingly, a few plans make integration among the disjoint system fragments by putting new hubs. The distributed plans for the most part contrast in the necessities of the recently structured topology. Case in point, Spiderweb [9] and Dispersed calculation for Improved Hand-off hub situation utilizing Least Steiner tree (Residences) [7] select to restore the system integration as well as accomplish a certain quality in the framed topology. Fundamentally, both plans attempt to keep away from the presentation of cut vertices with the goal that some level of strength, i.e., burden adjusting and high hub degree, is presented in the repaired system topology. Spiderweb also strives to minimize the required number of relays. The Spiderweb deploy relays inwards toward the center of the deployment area. The former considers the segments situated at the perimeter and establishes a topology that resembles a spider web. Meanwhile, it initially forms a star topology with all segments connected through a relay placed at the center of the area. Then, adjacent branches are further optimized by forming a Steiner tree for connecting two segments and the center node to reduce the required relay count.

Meanwhile, in [10], intersegment connectivity ought to maintain some level of quality of service (QoS) while placing the least number of relay node. The proposed approach initially models the deployed area as a grid with equal-sized cells. Each cell is assessed based on the uncommitted capacity of the relay node residing in the cell. Finally, meet the QoS requirement, the optimization is done by finding the cell-based least cost paths and populating node along these paths. On the other hand, form a biconnected intersegment topology [13] by placing redundant node so that the failure of a node can be tolerated and the network operations continues without interruption. The connectivity restoration [14] as a node placement problem on a grid and reposition the deployed node meet varying requirements on the intersegment traffic. As mentioned earlier, LTTR is reactive schemes that restore connectivity while imposing the least travel overhead and in a distributed manner.

C. Topology Management

Topology management [11] issues are very important in the context of wireless networks such as sensor networks. Topology management is to ensure that the various network connectivity parameters are managed so as to ensure that the parameter values are within certain bounds. Each sensor node in a sensor network to have only a pre-defined number of neighbor. This can be a requirement of a higher sensor information processing application in order to make sure that the sensor are distributed evenly over a region. Topology management has been addressed from various perspectives such as energy conservation in a wireless sensor actor network. Topology based protocols use the principle that every node in a network maintains topology information and that the main process of protocol operations was based on the topology of the network.

D. Fault Tolerant Management Routing Protocol

Fault Tolerant Management Routing Protocol [12] propose a solution to fault management for Wireless sensor networks because of their own limitations and the scalability issue. By introducing new network equipments, one can improve the traditional distributed hierarchical management structure, the equipment can quickly locate the failure and analyze the cause of the failure, therefore can greatly improve the efficiency of network maintenance. We also propose a new low-energy fault management protocol, which can quickly respond to failures. The experimental results show that compared with traditional protocols, this protocol can detect failure, responds quickly to failures and recover from failure at minimal costs, therefore reduce the impact of failure on networks. In these techniques, we analyze the cause of failure and take the action based on the particular failure using low energy and minimal costs and does not provide the efficiency of network maintenance.

3 EXISTING SYSTEM

The Defect Node Reformation (DNR) algorithm [1] for WSNs based on the grade diffusion (GD) algorithm and the genetic algorithm (GA). The Grade Diffusion algorithms create a routing for each sensor node. It is also identifies a set of neighbor node to reduce the transmission loading. It can record the some information regarding data relay and updates the routing path. This algorithm creates the node information, node identifier, neighbor node for each sensor node using the grade diffusion algorithms. The Grade Diffusion algorithms are used to create the routing table and reform the defect node using genetic algorithm. The Genetic algorithm is based on the reform the defect node. The genetic algorithm is based on the gene values. The genetic algorithm is classified into several types.

Initialization has generated the chromosomes. The gene value is based on zero or one. Each chromosome is a combination of the two solutions. The chromosome length is based on the number of depletion or nonfunctioning node. Evaluation is based on the fitness values. The fitness value is calculated based on the fitness function. The node is reformed using the fitness function. The fitness value is used to reuse the routing paths. The Selection is based on the fitness function. The Selection step removes a chromosome, which chromosome has a low fitness value. Select two chromosomes from a node according to their fitness .Keep the half of the chromosomes with better fitness values. The worse chromosomes will be deleted. Crossover is the similar to natural reproduction. Crossover combines genetic material from two folks, with a specific end goal to deliver predominant posterity. The Crossover is based on the reform the single node at a time. Mutation introduces traits not found in the original individuals. Flip a gene randomly in the chromosome. Mutation is performed on a small part of the population, keeping in mind the end goal to abstain from entering the unsteady state.

E. Drawbacks

The drawbacks of the Defect Node Reformation algorithm is,

- It does not focus of multiple node failure.
- Node reformation time is high.
- Sensing area is small.
- It does not concentrate the topology management.

4 PROPOSED SYSTEM

Least Troublesome Topology Repair algorithm is localized and distributed algorithm which restore network connectivity with minimum number of node movements. This algorithm is a decentralized algorithm. Each node coordinates with all the node. Each node and routing information will store the routing table. Each node aware the information of neighboring node with the help of routing tables. The node recovery is based on the actor node and sensor node. If the Sensor node failed in the network, the particular sensor node reformed using another sensor or actor node. The above operations is totally controlled by an actor node; otherwise the actor node will fails, then reform an actor node using the reconstruct the topology and restore the connectivity.

A. Modules Description

The Least Troublesome Topology Repair algorithm is classified into several modules. The modules are briefly explained about the proposed concepts. The modules are based on the creation of sensor and an actor node. The sensor node has a low cost compared with an actor node. An Actor node has a high power compared with the sensor. The name of the modules is,

a. Topology Formation
b. Failure Detection
c. Identifying the smallest block
d. Replacing the failure node
e. Children Movement

a. Topology Formation

The topology formed between the sensor and actors node. The Sensor nodes, which is used to screen the environment and send the data to actors. Actors are more power devices which are capable of responding, moving the actions. The network constructs the routing table. The routing table carries the information such as node identifier, neighbor node and node position. The node position identified from the routing table.

b. Failure Detection

The node screens the environment and when the node captures any information from the environment then the particular node store the information. Before the node transmits information to their one hop distant neighbor periodically we should identify a set of paths to reach the sink node. The source node selects one of the best paths among the selecting paths. Then the source node sent the heart beat messages to intermediate and sink node. The purposes of heartbeat message checks the node energy of the particular intermediate and sink node. The source node receive the response from the node and then split the node based on the energy. If the node has high energy then check the node availability; otherwise the node has low energy then check the particular node failed or not.

The node availability checks the node status, such as busy or available. If the node status is available then forward the data; otherwise wait for some time and then forward the data when the node status is available. If the particular node failed is conformed then the data forward through the other alternative paths. The source node finds if the failed node is critical or not in the network. If the failed node is not critical, the actor can reform the node. Otherwise, if the node is critical, it can reform the node based on the inter node and actor connectivity. The node is critical in the sense it is a parent node of the some sensor node.

c. Identifying the Smallest Block

After identifying the failed node, the smallest block must be identified. The smallest block is the block which selects the number of neighbor node of failed nodes. This block also selects the one hop distant neighbor when the neighbor node has high children. The failed node should replace among the selected node using the best candidate.

d. Replacing the Failure Node

The selection of the best candidate is based on the least number of node. The Selection of the best candidate is a one of the neighbor node and one hop neighbor of failed nodes. Then, the failed node location move into the best candidate location.

e. Children Movement

The node that are at two hops distance from the faulty node are called children and those at a distance of three hops are called grandchildren. Once the child knows that its parent is moving to a new location, it tells its one hop neighbor that is grandchildren and follows the path of its parent. If a child has two parent node that are transposed, it rearranges to a position to maintain its connection with both the parent node.

B. Flow Diagram

The flow diagram of Least Troublesome Topology Repair algorithm generates originating information from the starting node of the network. When the node gathers some information from the environment and stores the information itself. The node should construct the routing table. The node keeps the information such as node identifier, neighbor node and node position. The node selects the shortest path among the selected path in the network.

The particular node transmits the data before check the whether the intermediate and sink node are good or bad determined using heartbeat messages. If the source node receives the response then forwards the data; otherwise the node would not receive the response then assumed that the particular node has a low energy. If the node has high energy, then check the system availability. If the node status has available then forward the data; otherwise the node has any busy status, then waits some time and then starts the operations when the node is available status.

After finding the node failed, check whether the node is critical is not. If the node is not critical then the failed node reformed using an actor node and move to the neighbor node using least hop distance with the help of routing tables. The node is critical whether the node is an actor node or sensor node. If the sensor node is failed select the best candidate among the smallest block. The smallest block is the block finds the neighbor node of the failed node.

The best candidate is the one of the neighbor node. The best candidate selection based on the least hops itself. The failed node moves to the best candidate position which node has a least hop and finds the new position. The failed node moved to the new position and then sent the messages to children node itself. The children node are move to the new position then broadcast the message to the network. If the failed node is an actor node is based on the reestablish the network connection. Then follows the above procedure to recovers the particular an actor node and broadcast the messages to the network.

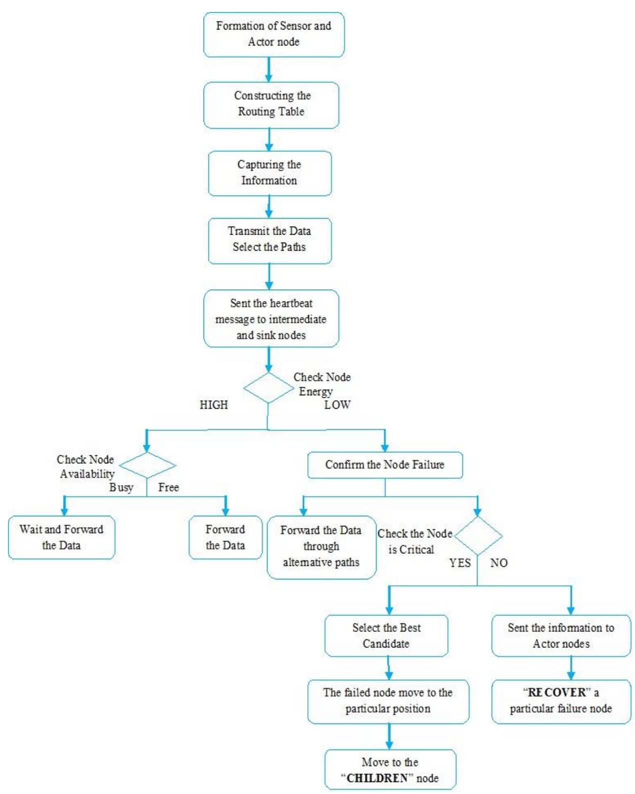

Fig 4.1 *Flow Diagram of Least Troublesome Topology Repair Algorithm*

C. Pseudocode

The pseudocode Fig 4.2 of this algorithm when the node is starting the network operations and constructs the routing table. If the node J detects a failed of the node F then check whether the node is critical or not. If the node F is critical not, then moving to the neighbor node position which node has the least distance and move to the particular node position; otherwise the node F is first critical selects the smallest block. This block is selects the best candidate of node F among the one of the neighbor node of the failed node F. The node J has least hops compared with the neighbor node of F, then node F move to the new position of node J. If the node F recovered by node J, then the node J broadcast the message to the network.

After detecting the failed node F, then the node F reaches the sink via the number of reachable node is zero. Then the node F compared with the number of reachable node and the smallest block size. If the smallest block size is greater than compared with the number of reachable nodes then smallest block is assigned to the number of reachable node. The smallest block size consists of number of neighbor node of node F. The node F can recover by anyone neighbor node of the network and then node F move to the computing any new neighboring position.

The node J receive some notification message from node F, and then forwards the data via node F. After that the node F is one of the children of node J. Before the node F reformed, transmits the message to the node F children. The node F has anyone children the node F finds the best candidate using the smallest block size. The smallest block size is found the number of neighbor node in the network. If the node J is least hopped, then node F move to the new position of node F; otherwise the node F finds another best candidate of node F neighbor. Then the node F moves to the particular recovering node position.

```
//EVERY NODE CONSTRUCT A ROUTING TABLE
//BASED ON THE ROUTE DISCOVERY ACTIVITIES

LTTR(J)
IF node J detects a failure of its neighbor F
  IF neighbor F is a critical node
    IF IsBestCandidate(J)
       Notify_Children(J);
       J moves to the position of neighbor F;
       Moved_Once ←— TRUE;
       Broadcast(Msg('RECOVERED'));
       Exit;
    END IF
  END IF
ELSE IF J receives notification messages from F
   IF Moved_Once || Received Msg('Recovered')
      Exit;
   END IF
   NewPosition ←— Compute_newPosition(J);
   IF NewPosition != CurrentPosition(J)
     Notify_Children(J);
     J moves to NewPosition;
     Moved_Once ←— TRUE;
    END IF
END IF

IsBestCandidate(J)
// Check whether J is the best candidate for tolerating the failure
NeighborList[] GetNeighbors (F) by accessing column F in RT;
 SmallestBlockSize ←— Number of neighbor nodes of failed node F;
 BestCandidate ←— J;
 FOR each node i in the NeighborList[]
       // Use the RT after excluding the failed node to
       // find the set of reachable nodes;
     Number of reachable nodes ←— 0;
     FOR each node k in RT excluding i and F
          Retrieve shortest path from i to k by using RT;
          IF the retrieved shortest path does not include node F
             No. of reachable nodes ←— No. of reachable nodes + 1;
          END IF
       END FOR
       IF Number of reachable nodes < SmallestBlockSize
            SmallestBlockSize ←— Number of reachable nodes;
            BestCandidate ←— i;
        END IF
END FOR
IF BestCandidate == J
     Return TRUE;
ELSE
     Return FALSE;
END IF
```

Fig 4.2 *Pseudocode for LTTR algorithm*

5 Simulation Results

The simulation result consists of several results based on the modules description. In Fig 5.1, the node 7 capturing the information from the environment. Then the source node 7 selects the best path to reach the destination. The node 7 check the node energy by sent the heartbeat messages to intermediate and sink node. In Fig 5.2, the node 7 checking the node energy. The failed node is based on which node has low energy among the intermediate and sink node. The node 15 has low energy and then node 15 checking the node is critical or not. In Fig 5.3, after detecting the failed node, the failed node select the best candidate. The best candidate in the sense which node has a low hops compared with other neighbors. The node 15 is critical in the sense recovering a node 15 by using a node 16. In Fig 5.4, check the node is available or not. The source, intermediate and sink nodes are available and then forward the data. In these results, recovering the multiple node failures and reduces the data losses.

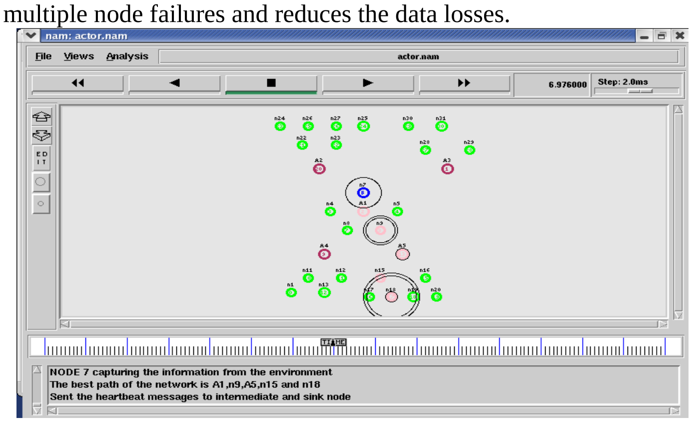

Fig 5.1 *Checking the Node Energy*

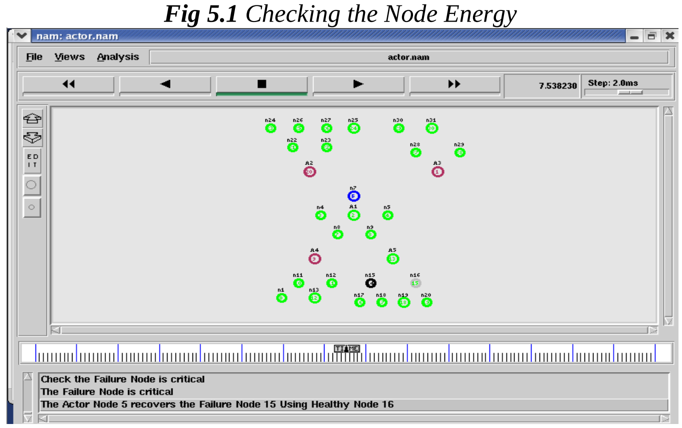

Fig 5.2 *Detection of Failure Node*

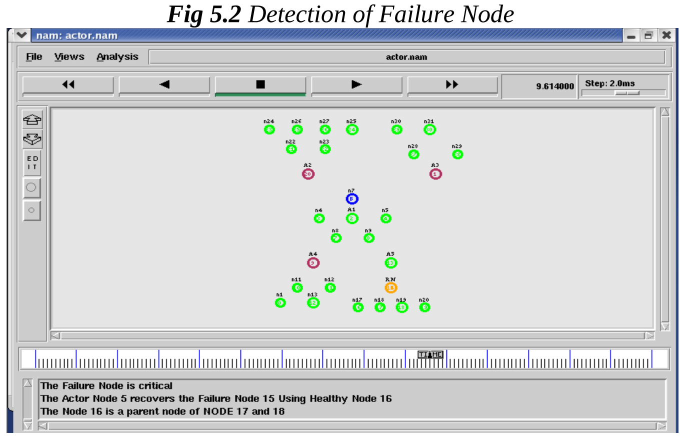

Fig 5.3 *Recovering a Failed Node*

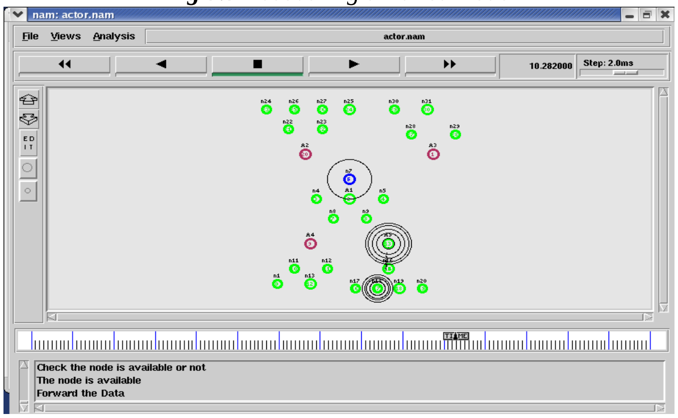

Fig 5.4 *Checking the Node Status*

6 Conclusion And Future Enhancement

Least Troublesome Topology Repair algorithm is used to reconstructing the network connectivity without extending the number of routing paths. LTTR also works very well in dense networks and yields close to optimal performance even when node is partially aware of the network topology. By using this algorithm recover from multi node failures at a time; recover the transmission and routing path at a time. Considering such a problem with gather node failure is more complex and challenging with nature. In the future enhancement, we plan to scrutinize this issue. Our future plan also includes factoring in coverage and ongoing application tasks with the recovery process, focusing on the security and developing a test bed for evaluating the various failure recovery schemes.

References

[1] Hong-Chi Shih, Jiun-Huei Ho, Bin-Yih Liao, and Jeng-Shyang Pan, "Fault Node Recovery Algorithm in WSN" IEEE SENSORS JOURNAL, VOL. 13, NO. 7, JULY 2013.

[2] T. H. Liu, S. C. Yi, and X. W. Wang, "A Fault Management Protocol for Low-Energy and Efficient Wireless Sensor Networks," Journal of Information Hiding and Multimedia Signal Processing, vol. 4, No. 1, pp. 34-45, 2013

[3] F. Akyildiz and I. H. Kasimoglu, Oct. 2004, "Wireless sensor and actor networks: Research challenges," Ad Hoc Netw. J.,vol. 2, no. 4, pp. 351–367.

[4] A.Abbasi, M. Younis, and K. Akkaya, Sep. 2009, Movement-assisted connectivity restoration in wireless sensor and actor networks," IEEE Trans. Parallel Distrib. Syst., vol. 20, no. 9, pp. 1366–1379.

[5] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, Mar. 2008 "Distributed recovery of actor failures in wireless sensor and actor networks," in Proc.IEEE WCNC, Las Vegas, NV, pp. 2480–2485.

[6] M. F. Younis, Sookyoung Lee, and Ameer Ahmed Abbasi, Dec. 2010, "A Localized algorithms for restoring inter node connectivity in networks of movable sensor," IEEE Trans. On computers, vol. 59, no. 12.

[7] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, Feb. 2010, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," IEEE Trans. Comput., vol. 59, no. 2, pp. 258–271.

[8] F. Senel, M. Younis, and K. Akkaya, May 2011, "Bio-inspired relay node placement heuristics for repairing damaged wireless sensor networks," IEEE Trans. Veh. Technol., vol. 60, no. 4, pp. 1835–1848.

[9] S. Lee and M. Younis, Jun. 2009, "QoS-aware relay node placement in a segmented wireless sensor network," in Proc. IEEE ICC, Dresden, Germany, pp. 1–5.

[10] Nikolaos A. Pantaziz, Stefanos A. Nikolodakis and Dimitrios D. Vergados, Jun.2010, "Topology Management wireless sensor network, "IEEE Communications survey Vol. 15 no. 2.

[11] T. H. Liu, S. C. Yi, and X. W. Wang, "A fault management protocol for low-energy and efficient wireless sensor networks," J. Inf. Hiding Multimedia Signal Process., vol. 4, no. 1, pp. 34–45, 2013

[12] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in Proc. 26th Annu.Joint Conf. INFOCOM, Anchorage, AK, May 2007, pp. 1649–s1657.

[13] F. Al-Turjman, H. Hassanein, and M. Ibnkahla, "Optimized node repositioning to federate wireless sensor networks in environmental applications,"in Proc. IEEE Int. GLOBECOM, Houston, TX, Dec. 2011, pp. 1–5.

[14] S. Lee and M. Younis, "Recovery from multiple simultaneous failures in wireless sensor networks using minimum Steiner tree," J. Parallel Distrib.Comput., vol. 70, no. 5, pp. 525–536, May 2010.