GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

MULTIPLE NODE FAILURE RECOVERY LEDIR ALGORITHM IN WIRELESS SENSOR-ACTOR NETWORKS

Megha Sharma[*1] & Er Gagan Dhawan[2]
[*1]Department of Computer Science Engineering Miet Mohri, M.Tech Scholar
[2]Department of Computer Science Engineering Miet Mohri, Assistant Professor

ABSTRACT

Due to wide area of innovation in wireless technology, Wireless Sensor-Actor Networks (WSANs) have been the interesting area of application for last two or three decades in various fields. In WSANs, sensor node will explore the environment and transmit the gathered information to actor nodes. Actor node gathers or aggregates that information and performs specific operations in response to various events. Actors have to work together so it is essential to retain a strongly connected network topology at all the time. Since, actors perform in hostile environment, so they are prone to failures. Moreover a failure of an actor node leads to partition of the network into disjoint blocks and would thus violate the connectivity goal. For the recovery of faulty nodes, there is a requirement of location table for initiating the recovery process.

LeDiR is a localized and distributed algorithm for single node failure that depends on local view of the node about the network. But in some real time situations, like bombardment and other natural calamity, where large number of node get disrupted or ruptured. There is a requirement to reconsider the above said schemes for multiple node failure recovery. In this paper, a new phase of LeDiR is introduced that may recover simultaneous node failure. Simulation result shows that LeDiR for multiple node failure (MLeDiR) outperforms conventional LeDiR methods in terms of total number of nodes moved and average distance moved during the recovery process.

Keywords: *WSANs; network recovery; fault tolerance; restoring connectivity, LeDiR*

I. INTRODUCTION

Wireless Sensor Network provide huge amount of data from the environment through sensor nodes that are placed in an area of interest. The data collected is executed at the sink nodes for the discovery of occurrence of an event that took place in the environment. WSNs results in simplex method of data delivery, where information from the environment is supplied to the users. Due the development of sensing devices like actuators, two way exchange of information is possible. The data that is collected from a particular area of interest is distributed among all the nodes to perform particular action. This gave rise to the development to above technology, which are efficient to examining the environment, processing the information, providing decision on the basis of results and taking necessary actions[1]. WSANs can be implemented over vast areas such as military application like battlefield surveillance, climatic conditions in buildings, chemical and biological attack detection, smart home, forest fire detection, environmental monitoring. Fire detection is one of the main examples of WSANs. Sensors distributed over an area are used for detecting the source and magnitude of fire. This information is transmitted to water sprinklers which perform the role of actors and restrict the fire. Appropriately, the fire can be easily controlled as soon as possible before it becomes dangerous. The mechanism can be executed without any involvement of humans through an autonomous architecture of WSANs to avoid delays and errors. Likewise light and motion sensors can detect the presence of humans in a particular room in home automation systems. Sensors can direct the apt actor nodes to perform on the basis of pre-specified user priority, e.g. managing the electricity of the house, or trigger or unrigged the various appliances.

The ability to utilize node mobility for various applications is one of the vital merits of WSANs. WSANs performance parameters include accuracy, coverage, connectivity, deployment ease, energy and dependability which can be improved by moving and relocating various nodes in these networks [2]. Moving and relocating of nodes advance dependability of the networks in various ways. For particular situation, dead nodes can be replaced by neighbour nodes by moving them to the dead node location. If the network is subdivided, mobility of nodes can be exploited to recover connectivity by moving one or multiple nodes to desired locations. In previous work, the dependability issues have been examined thoroughly in framework of WSANs [3][4]. In WSANs every node required to maintain partial knowledge of the network. In order to bypass enormous load of state update overhead and to accelerate the connectivity recovery process, prior work of node is to retaining the one or two hop neighbour list and predetermines some norms for the node's association in the recovery process [5]. Overview of a typical sensor-actor network is depicted in Fig. 1.

A. Abbasi et al. proposed a LeDiR algorithm for WSAN recovery assuming non-simultaneous node failure [6]. The issue of simultaneous node failure can happen in the network. So, in this article, a novel LeDiR algorithm is proposed for simultaneous multiple node failure called (MLeDiR) for network recovery with minimal topology changes. Two simultaneous node failure is considered in a single event.MLeDiR depends on the nearby knowledge of nodes in the network to relocate minimum number of nodes and to make sure that no path between any pair of nodes is extended related to its previous location.

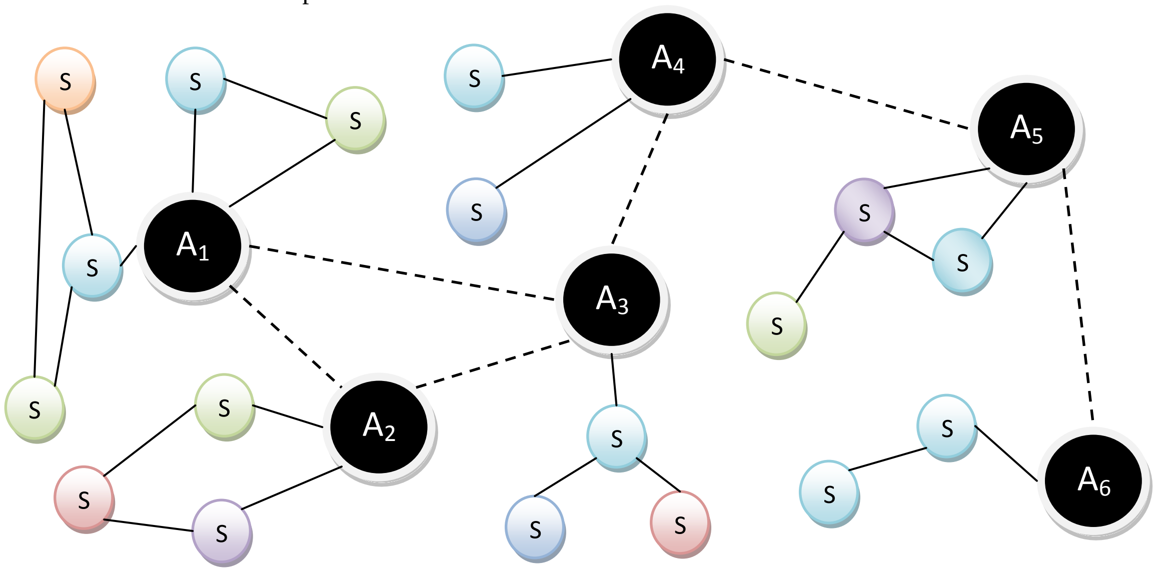

Fig. 1 Overview of WSANs

The rest of the paper is categories as follows. Section 2presents the related work. In section 3, system model will be described. Section 4discusses the results and section 5 concludes the article.

II. RELATED WORK

Many schemes have been proposed for network recovery and connectivity through node repositioning in subcategorised WSANs.The recovery schemes differ on the basis of involvement of the actor nodes in the recovery process.To ensure that recovery schemes lead in an efficient way, schemes need that every node in a network must have knowledge of their two hop neighbours. The knowledge of tow hop neighbour grant the nodes to detect cut vertices.In Recovery through inward motion (RIM), bring about a recovery process by relocating the neighbour nodes of the fail node. The cumulative effect looks like the topology is shrinking inward. In this cascaded movement is used to sustain the network connectivity [5]. Distributed Actor Recovery Algorithm (DARA)opts an efficient way to restore the network connectivity of the inter-actor network that has been affected by the failure of the actor node.In this, they identify best candidate (BC) i.e. is actor node that should be repositioned to re-establish a particular level of connectivity [3]. Partition Detectionand Recovery Algorithm (PADRA) can figure out the portioning in prior and re-establish the connectivity in such failure with minimized node movement i.e. BC will replace the dead node and its children node will move along it to a position where children nodes are in communication range [7]. In DARA & PADRA, network connectivity is sustained by cascaded movement of the nodes.Least Distance Movement Recovery (LDMR) is an appropriate recovery scheme that exploits the non-cut vertices in order to require the least travel distance from the intended nodes. In this recovery process starts with the search phase each node broadcast a message containing the failed node ID, neighbour node ID & TTL. When the neighbour node receives responses, it chooses the BC [8].Least Disruptive Topology Repair Algorithm (LeDiR)utilizes the partial knowledge of the node about the network topology, gained during the route discovery to devise a recovery plan that relocates the least no of nodes and ensures that no path between any pair of node is extended[6].

III. PROPOSED SYSTEM MODEL FOR MLEDIR

WSANs consist of two categories of components: sensors and actors. The phenomenon of sensing and acting are performed by sensors and actor nodes, respectively. Sensors are highly constrained in energy and processing capacity. Whereas, actors are more capable nodes as compared to sensor nodes with more on board power supply, higher computation and communication resources. Like sensor nodes, an actor node might be embedded with different actuators to perform different task using their computation capabilities, multiple actors can perform an action based on information received from multiple sensors.Upon deployment, actor nodes are assumed to discover each other and form one connecting network using location table. In this article, actors are assumed to move on demand to perform tasks to enhance the inter-actor connectivity. The focus of this article is on restoring strong connectivity at the level of inter-actor topology.

Parameters used to vary the characteristics of the topology in the different experiments are

- **Number of deployed actors (N):** This parameter affects the node density and the WSAN connectivity. Increasing N makes the WSAN topology highly connected.
- **Communication range (r):** All actors are assumed to have the same communication range r. The value of r affects the initial WSAN topology. While a small r creates a sparse topology, a large r boosts the overall connectivity.

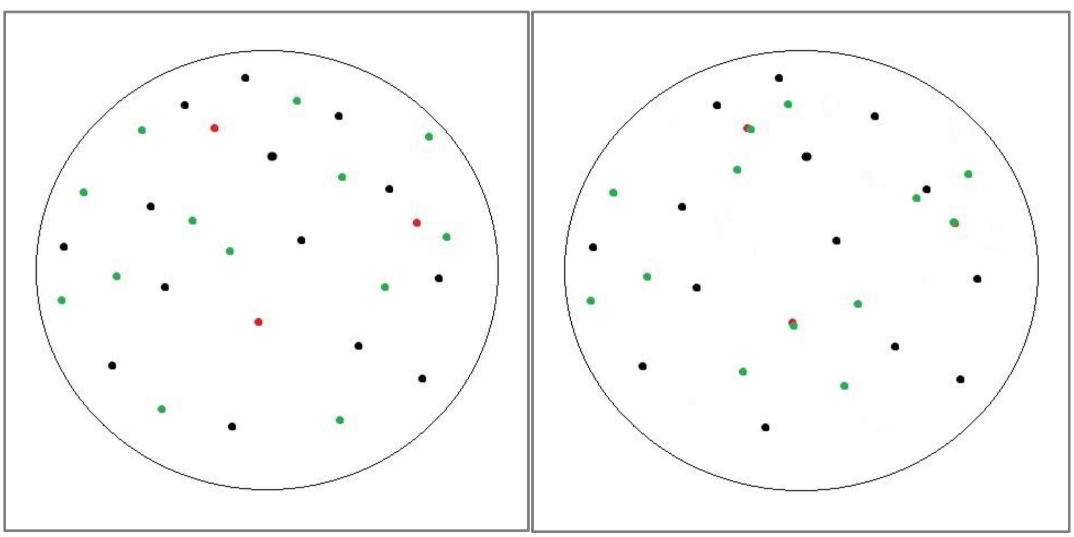

Fig.2 (a) Multiple failure of nodes, (b)Nodes recovered by MLeDiR

Fig.2a shows the multiple failure of nodes in WSANs. The nodes which are red in colour are the faulty nodes, the nodes which are green in colour are the actor nodes and the nodes which are black in colour are the sensor nodes. Fig. 2b shows the recovery of the faulty nodes using MLeDiR technique. In this experiment, 3 nodes failed simultaneously in WSANs. The nodes which are BC, will reposition to the position of the faulty nodes. As we know that actor nodes are the only nodes in the WSANs which can become the BC. After recovery actor nodes takes the position of the faulty node along with their desired position from where they can be in communication range of their parent node and maintain the network connectivity.

Pseudo code for MLeDiR
After detection of failure
Step1: get failure function for fn1, fn2 = X1, X2; \\ specify location and no. of node failing
Step2: range of failing nodes=R1, R2; \\Parent node selection starts from here
Step3: for loop 1 -> N1 \\N1 is the no of actor nodes
Step4: if (N location < R) \\R is the range of each node.
Step5: DT=get distance of N1;
Step6: end if condition;
Step7: end for loop;
Step8: get min. of DT=PN1, PN2; \\PN1,PN2 is the parent node 1,2\\ DT is the distance table \\following steps till step 18 is for preference in case of same parent node.
Step9: if (PN1==PN2)
Step10: get no. actor nodes na1, na2 within R1,R2 of fn1,fn2
Step11: if (na1>na2)
Step12: use PN1 for fn1;
Step13: find new PN2 for fn2
Step14: else
Step15: use PN2 for fn2;
Step16: find new PN1 for fn1
Step17: end if condition;
Step18: end if condition;
Step19: location parent node PN1, PN2 =location failure node fn1, fn2\\node replacement by parent \\ Child movement start from here
Step20: for loop 1 -> N2 \\N2 is no. of all nodes within range R1, R2 for child movement
Step21: if (D1N2 > D2N2) \\ D2N2 & D1N2 is the previous & current distance from PN1.
Step22: Location of N2=midpoint of D1N2 \\ child movement
Step23: end if condition
Step24: end for loop\\repeat steps 20 to 24 for PN2 also.

IV. SIMULATION RESULTS

In this paper, nodes are deployed in an area of 600*600 units. There are a total of 50 nodes. Range of actor node is 100 units i.e. these nodes can transmit or can communicate with other nodes in the network which are in its range. Nodes are varied in ratio such that there are 40 sensor nodes and 10 actor nodes in an experiment. These numbers are varied again in next experiment such that number of sensor nodes are 30 and number of actor nodes are 20. These number change to 20 and 30 for sensor nodes and actor nodes respectively. Again the number of node is varied to 10 sensor nodes and 40 actor nodes.

Metrics used to measure the performance of MLeDiR are

1) **Total travelled distance:** reports the distance that the involved nodes collectively travel during the recovery. This can be envisioned as a network-wide assessment of the efficiency of the applied recovery scheme.
2) **Number of nodes moved:** reports the number of nodes that moved during the recovery. This metric assesses the scope of the connectivity restoration within the network

Figure 3 shows the comparison of total distance travelled during the recovery process for LeDiR and MLeDiR. In this we have triggered five events for both of LeDiR and MLeDiR for simultaneous node failure. In every event two nodes will fail at same instant of time and in case of single node only a single node will fail. The output taken is average of ten simulations and the distance travelled by the nodes during the recovery process is calculated.

 ISSN 2348 – 8034
Impact Factor- 4.022

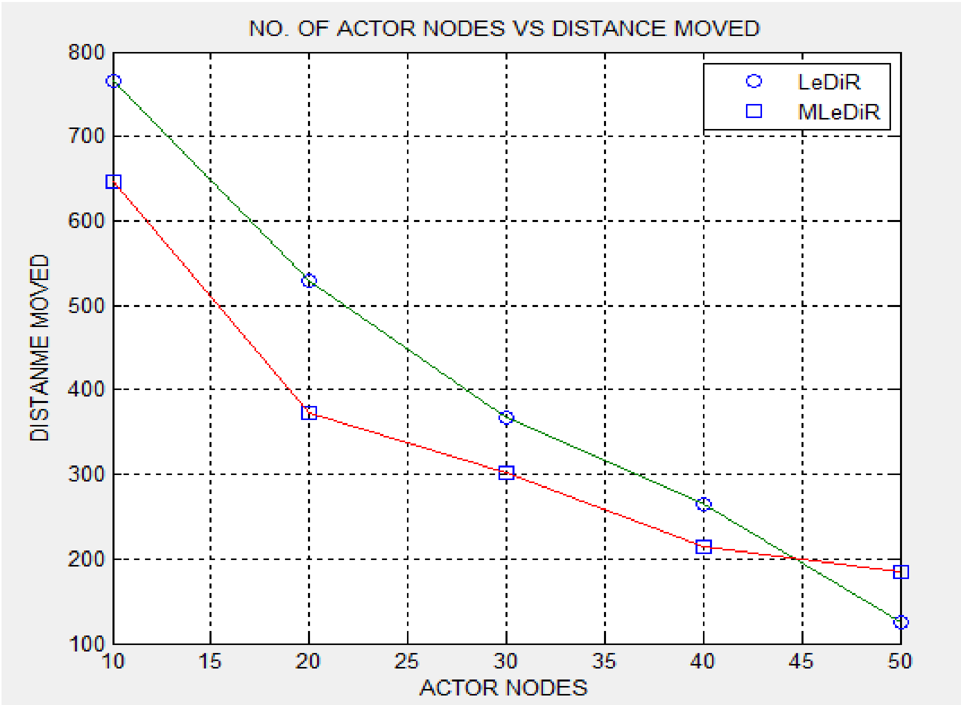

Fig.3 Comparison of Distance travelled in LeDiR and MLeDiR.

Figure 4 shows the comparison of LeDiR and MLeDiR for numbers of nodes moved in the recovery process. The output taken is average of ten simulations and the number of nodes moved during the recovery process is calculated.

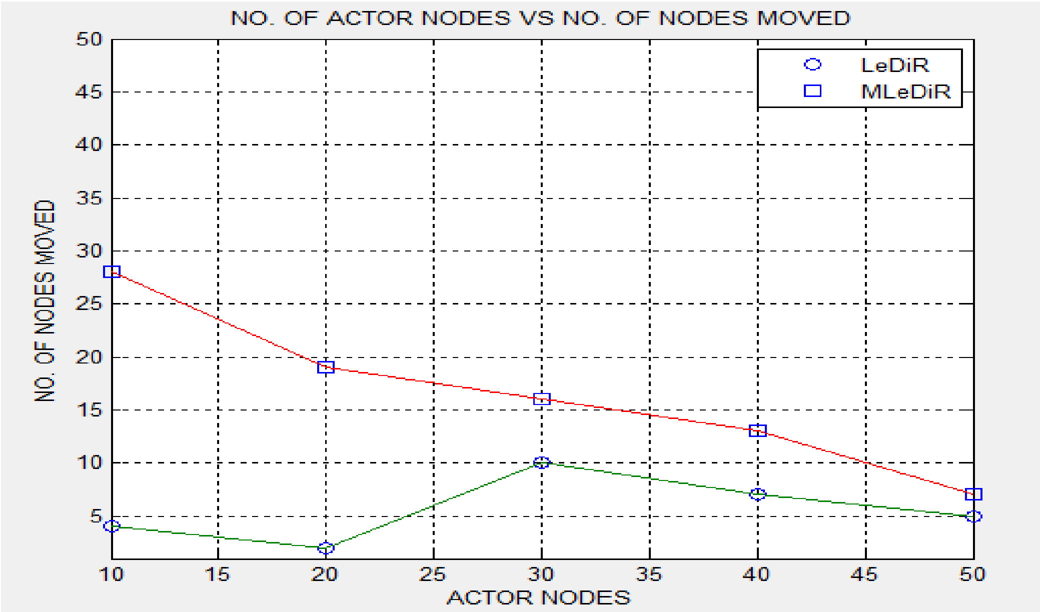

Fig.4 Comparison of nodes moved in LeDiR and MLeDiR.

Figure 5 shows the comparison of LeDiR and MLeDiR for path extending distance after recovery process.

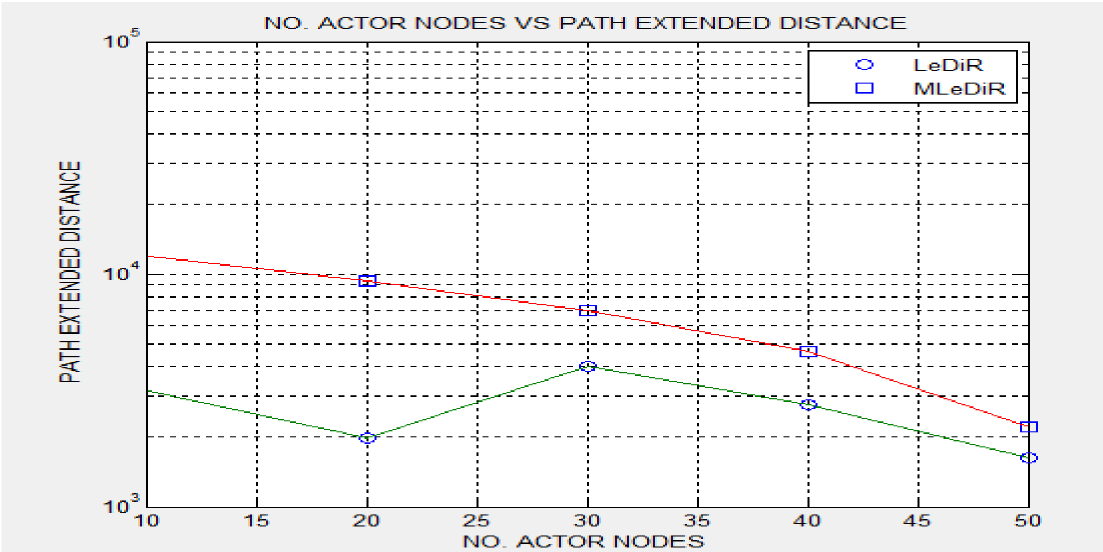

Fig.5 Comparison path extended distance in LeDiR and MLeDiR

V. CONCLUSION

In current years, WSANs based applications growing day by day due to their immense capabilities. This article has attempted to solve the complex and challenging issue of simultaneous node failure in WSANs. A new LeDiR algorithm called MLeDiRis proposed to restore connectivity with minimal change in network topology. The recovery is carried out through relocation of actor nodes towards the faulty nodes position. MLeDiR proves to be a better selection for maintain the network topology. The parent node election is based on the distance from the faulty node position. However traffic load on nodes should also be considered for parent node selection. This will remain the area of future scope.

REFERENCES

1. *I. F. Akyildiz, I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges", Journal of Ad Hoc Networks, Volume 2, Issue 4, 2004, pp. 351-367.*
2. *M. Younis, K. Akkaya, "Strategies and Techniques for node placement in wireless sensor networks: A survey", Ad-Hoc Networks, Elsevier, Volume 6, Issue 4, 2008, pp. 621-655*
3. *A. Abbasi, M. Younis, K. Akkaya, "Movement assisted connectivity restoration in wireless sensor and actor networks", IEEE transaction on Parallel and Distributed Systems, Volume 20, Issue 9, 2009, pp. 1366-1379.*
4. *K. Akkaya, F. Senel, A. Thimapuram, S. Uludag,"Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility", IEEE transaction on Computers, Volume 59, Issue 2, 2010, pp. 258-271.*
5. *M. Younis, S. Lee, S. Gupta, k. Fisher, "A localized self healing algorithms for networks of moveable sensor nodes", Proc. IEEE Globe-COM, New Orleans, LA, Nov. 2008, pp. 1-5.*
6. *A. Abbasi, M. Younis, U. Baroudi, "Recovering from a node failure in wireless sensor-actor networks with minimal topology changes", IEEE Transactions on Vehicular Technology, Volume 62, Issue 1, 2013, pp. 256-271.*
7. *K. Akkaya, A. Thimapuram, F. Senel, S. Uludag,"Distributed recovery of actor failure in wireless sensor and actor networks", Proc. IEEE WCNC, Las Vegas, NV, 2008, pp.2480-2485.*
8. *A. Alfadhly, U. Baroudi, M. Younis, "Least distance movement recovery approach for large scale wireless sensor –actor networks", Proc. Workshop FedSenS, Istanbul, Turkey, 2011.*