www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3, Issue 10 October, 2014 Page No. 8972-8977

Efficient and Secure Node Recovery in Wireless Sensor and Actor Network Using Minimal Topology Changes

A. Kokilapriya[1],G.Sophia Reena[2]

[1],Mphil scholar, P.S.G.R. Krishnammalcolleage for Women,Coimbatore,
Tamil Nadu, India.
priyamahe07@gmail.com

[2]Assistant prof, P.S.G.R Krishnammalcolleage for women, Coimbatore, Tamil Nadu, India.
sophigeo@gmail.com

Abstract: ***In wireless sensor-actor networks, sensors probe their surroundings and send their data to actor nodes. Actors collaboratively react to achieve predefined application mi. Since actors have to synchronize their operation, it is necessary to hold a highly connected network topology all the time. Moreover, the length of the inter-actor connection paths may be constrained to meet latency requirements. Failure of an actor may cause the network to separate into disjoint blocks and break such connectivity. One of the effective restoration methodologies is to autonomously reposition a subset of the actor nodes to restore connectivity. Contemporary restoration schemes either impose high node relocation over-head or extend some of the inter-actor data paths. This paper overcomes these drawbacks and presents a Least-Disruptive topology Repair (LeDiR) algorithm. LeDiR relies upon the local view of a node about the network to devise a recovery plan that relocates minimum number of nodes and ensures that no path between any pair of nodes is expanded. The performance of LeDiR is examined mathematically and validated through extensive simulation experiments.***

Keywords: Wireless Sensor and Actor Network (WSAN), shortest path Routing Table (SRT), Distributed Actor Recovery Algorithm(DARA), Partition Detection and Recovery Algorithm (PADRA)

1. Introduction

Recent years have witnessed a growing interest in the applications of wireless sensor-actor networks (WSANs). Of particular interest are applications in remote and harsh areas in which human intervention is risky or impractical. Examples include space exploration, battle field surveillance, search-and research, and coastal and border protection. A WSAN consists of a set of miniaturized low-cost sensors that are spread in an area of interest to measure ambient conditions in the vicinity. The sensors serve as wireless data acquisition devices for the more powerful actor nodes that process the sensor readings and put forward an appropriate response. For example, sensors may detect a fire and trigger a response from an actor that has an extinguisher. Robots and unmanned vehicles are example actors in practice [8]. Actors work autonomously and collaboratively to achieve the application mission. Given the collaborative actors' operation, a strongly connected inter-actor network topology would be required at all times. Actors usually coordinate their motion so that they stay reachable to each other. Failure of an actor causes the network into disjoint block. The remote setup in which WSANs often serve makes the deployment of additional resources to replace failed actors impractical, and repositioning of nodes becomes the best recovery option [1]. In addition, tolerance of node failure cannot be orchestrated through a centralized scheme given the autonomous operation of the network. On the other hand, distributed recovery will be very challenging since nodes in separate partitions will not be able to reach each other to coordinate the recovery process. Therefore, contemporary schemes found in the literature require every node to maintain partial knowledge of the network state. To avoid the excessive state-update overhead and to expedite the connectivity restoration process, prior work relies on maintaining one- or two-hop neighbor lists and predetermines some criteria for the node's involvement in the recovery Unlike prior work, this paper considers the connectivity restoration problem subject to path length constraints. A novel Least-Disruptive topology Repair (LeDiR) algorithm is proposed.

2. Problem Statement

Two types of nodes involved in WSAN are sensorsandactors.Sensorsareinexpensive andhighly constrained inenergyandprocessingcapacity.Ontheother hand,actorsaremorecapablenodeswithrelatively more onboardenergysupplyandrichercomputation andcommunicationresources thetransmissionrangeofactors isfiniteandsignificantly lessthanthedimensionsofthedeploymentarea.Althoughactorscan theoretically reacheach otherthroughasatellitechannel,thefrequentinter-actor interactionrequiredbyWSANapplications wouldmaketheoften intermittent satellitelinksunsuitable.Itisthusnecessaryfor actorstorelymostlyoncontemporaryterrestrialradiolinks forcoordination amongthemselves.Given theapplication-basedinteraction, anactorisassumedtoknow howmanyactorsarethereinthenetwork. Thefocusofthis paperisonrestoringstrongconnectivity atthelevelofinter-actortopology. Itisassumedthatasensornodecanreachat leastoneactorovermultihop pathsandwillnotbeaffectedif theactorshavetochangetheirpositions.Thus,sensornodesare notpartoftherecovery process.

3. Related work:

K. Akkaya et al.[1] had discussed that there are number of schemes have recently been proposed for Restoring network connectivity in partitioned WSANs

Recovery through Node Reposition:

A.Abbasi et al. [2] had discussed that the main idea of recovery schemes is to Reposition some of the healthy nodes in the network to reinstate strong connectivity. LeDiR fits in this category. Published approaches differ in the level of involvement expected from the healthy nodes, in the required network state that needs to be maintained, and in the goal of the recovery process. Distributed Actor Recovery Algorithm (DARA) DARA pursues a probabilistic scheme to identify cut vertices. A best candidate is selected from the one-hop neighbors of the dead actor as a recovery initiator and to replace the faulty node. The best candidate selection criterion is based on the least node degree and physical proximity to the faulty node. The relocation procedure is recursively applied to handle any disconnected children. In other words, cascaded movement is used to sustain network connectivity. K.Akkaya et al. [3] had discussed about thePArtition Detection and Recovery Algorithm PADRA identifies a connected dominating set to determine a dominatee node. The dominatee does not directly move to the location of the failed node; instead, a cascaded motion is pursued to share the burden.

Recovery by Placement of Relay Nodes

F. Senel et al. [4] had discussed that the foregoing algorithms aim to restore the network connectivity by efficiently relocating some of the existing nodes. However, in some setups, it is not feasible to move the neighbors of the failed node due to physical, logistical, and coverage constraints. Therefore, some schemes establish connectivity among the disjoint network segments by placing new nodes. The published schemes generally differ in the requirements of the newly formed topology. For example, SpiderWeb. S.Lee et al. [5] describes about the Distributed algorithm for Optimized Relay node placement using Minimum Steiner tree (DORMS). This opt to not only reestablish the network connectivity but also achieve a certain quality in the formed topology. Basically, both schemes try to avoid the introduction of cut vertices so that some level of robustness, i.e., load balancing and high node degree, is introduced in the repaired network topology. SpiderWeb and DORMS also strive to minimize the required number of relays. Both

SpiderWeb and DORMS 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, DORMS 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. M.Younis, et al. [6] had proposed the Inte-rsegment connectivity ought to maintain some level of quality of service (QoS) while placing the least number of relay nodes. 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, to meet the QoS requirement, optimization is done by finding the cell-based least cost paths and populating nodes along these paths. Al-Turjman*et al.* [7] model the connectivity restoration as a node placement problem on a grid and reposition the deployed nodes to meet varying requirements on the intersegment traffic. As mentioned earlier, LeDiR is a reactive scheme that opts to restore connectivity while imposing the least travel overhead and in a distributed manner.

4. LeDIR Implementation:

Create network model

An undirected graph G (V, E) where the set of vertices V represent the wireless nodes in the network and E represents set of edges in the graph which represents the external or realistic links between the mobile nodes. Detector nodes are placed at a same level. Two nodes that can directly communicate with one another are connected by an edge in the graph. Figure.1 explains the network model.

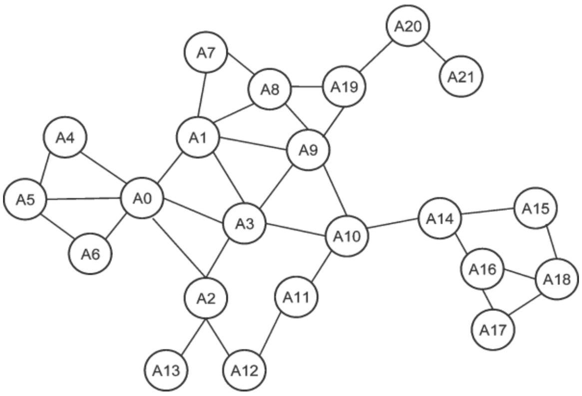

Figure.1

Detection of failure node

In this module, actors will periodically send heartbeat messages to their neighbors to ensure that they are functional, and also report changes to the one-hop neighbors. Missing heartbeat messages can be used to detect the failure of actors. Once a failure is identified in the neighborhood, the one-hop nearer of the failed actor would determine the impact, i.e., whether the failed node is critical to network connectivity. This can be done using the shortest path routing table by executing the well-known depth-first search algorithm. Basically, a cut vertex F has to be on the shortest path between at least two neighbors of F. The Shortest path Routing Table can make the same conclusion for a node that is not a cut vertex but serves on the shortest path of all nodes.

Identification of smallest block

LeDiR limits the relocation to nodes in the smallest disjoint block to reduce the recovery process. The smallest block has the minimum number of nodes and would be identified by finding the reachable set of nodes for every direct neighbor of the failed node and then picking the set with the least number of nodes. Since a dangerous node will be on the shortest path of two nodes in separate blocks, the set of reachable nodes can be identified through the use of the shortest path routing table after excluding the failed node. In other words, two nodes will be connected only if they are in the same block.

Replacing faulty node

If node J is the neighbor of the failed node that belongs to the smallest block, J is considered the best candidate to reinstate the faulty node. Since node J is considered the gateway node of the block to the failed critical node (and the rest of the network), refer to it as “parent.” A node is a “child” if it is two hops away from the failed node, “grandchild” if three hops away from the failed node, and so on. The reason for selecting J to replace the faulty node is that the smallest block has the fewest nodes in case all nodes in the block have to move during the recovery. As will be shown later, the overhead and convergence time of LeDiR are linear in the number of nodes, and thus, engaging only the members of the smallest block will expedite the recovery and reduce the overhead. In case more than one actor fits the characteristics of a BC, the closest actor to the faulty node would be picked as a BC. Any further ties will be resolved by selecting the actor with the least node degree. Finally, the node ID would be used to resolve the tie.

Children movement

When node J moves to replace the broken node, probably some of its children will lose direct links to it. In general, do not want this to happen since some data paths may be extended. LeDiR opts to avoid that by satisfying the existing links. Thus, if a child receives a message that the parent P is moving, the child then notifies its neighbors (grandchildren of node P) and travels directly toward the new location of P until it reconnects with its parent again. If a child receives notifications from multiple parents, it would find a location from where it can maintain connectivity to all its parent nodes by applying the procedure.

Implementation of Distributed LeDiR

In this section, an actor may have only partial knowledge about the network with routes to some nodes missing in its shortest path routing table. This can happen because of variations in the topology caused by node mobility or due to the fact that a subset of actors does not need to interact and that a route has yet to be discovered. So, LeDiR may employ probabilistic cut vertex detection systems that use two-hop information to boost the fidelity of the assessment and mitigate the effect of the missing entries in the shortest path routing table. It is important to note that if LeDiR is applied while the failed node F turned out not to be a cut vertex, this is because of the inaccuracy in the probabilistic detection system, the shortest path lengths between nodes will not change since LeDiR sustains the links between nodes in the same block and the network will be in fact connected, i.e., one block. Determining the block size is always based on the entries of the shortest path routing table that neighbors of F have, regardless whether F is a cut vertex or not. Now, if the analysis to determine the block size is based on inaccurate assertion about whether F is a cut vertex, and one of the nearest neighbors F still becomes the best candidate and performs LeDiR successfully, i.e., proceeds to replace the faulty node. Children would follow best candidate to maintain connectivity, and so on. LeDiR imposes a timeoutafter which the neighbor(s) belonging to the second largest block will move.

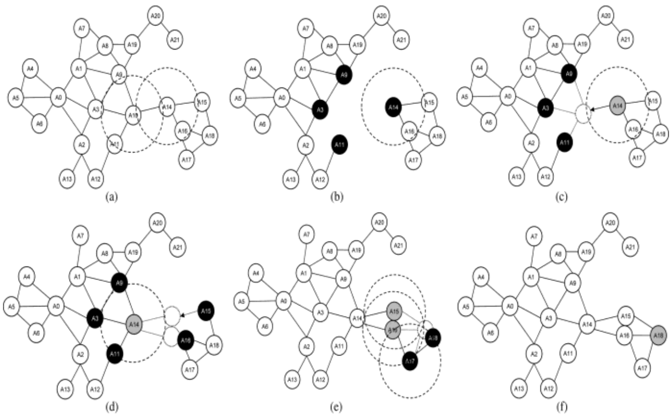

Figure.2

Fig. 2 shows an example for how LeDiR restores connectivity after the failure of *A*10. Obviously, node *A*10 is a cut vertex, and *A*14 becomes the one-hop neighbor that belongs to the smallest block [see Fig. 2(a)–(c)]. In Fig. 4(d), node *A*14 notifies its neighbors and moves to the position of *A*10 to restore connectivity. Disconnected children, i.e., nodes *A*15 and *A*16, follow through to maintain

communication link with *A*14 [see Fig. 4(e)]. Note that the objective of the children movement is to avoid any changes to the current routing table. Nodes *A*15 and *A*16 would notify their children *A*17 and *A*18 before they move. Since *A*18 had communication links with nodes *A*15, *A*16, and *A*17, it moves to a new location where it can stay directly connected to these nodes [see Fig. 4(f)]. The links between *A*17 and nodes *A*16 and *A*18 are not affected by the relocation process, and thus, *A*17 would not need to reposition. Fig. 4(f) shows the repaired network topology where the paths from nodes *A*14, *A*15, *A*16, *A*17, and *A*18 to the other nodes in the network are not extended.

5. Performance Analysis:

In this the performance of Least Disruptive Topology Repair (LeDiR) algorithm will be analyzed. Network lifetime and Tour length of the nodes will be calculated. Network lifetime will be calculated on the basis of the time taken to the first sensor node failure. Length of the path travelled by the node will be calculated as the Tour length. Figure.2 and Figure.3 explains about the Network Lifetime. Figure.4 and Figure.5 explains about the length of the path travelled by the node.

Life Time:

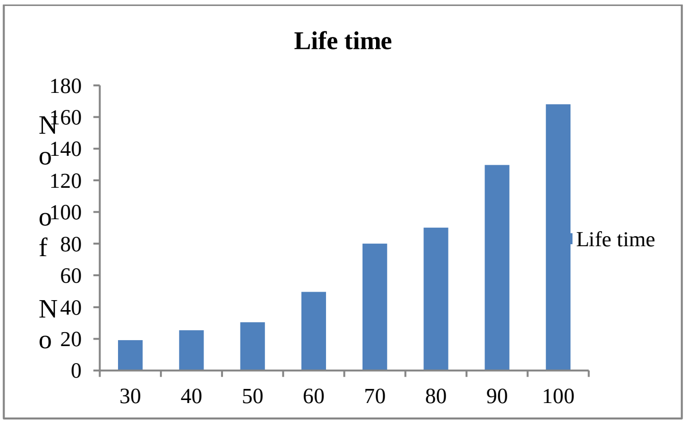

Figure.3

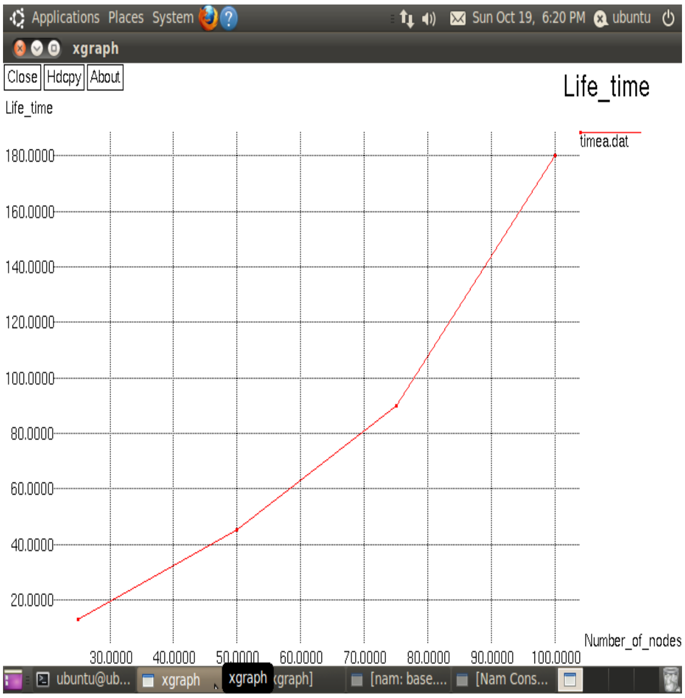

Figure.4

Tour Length:

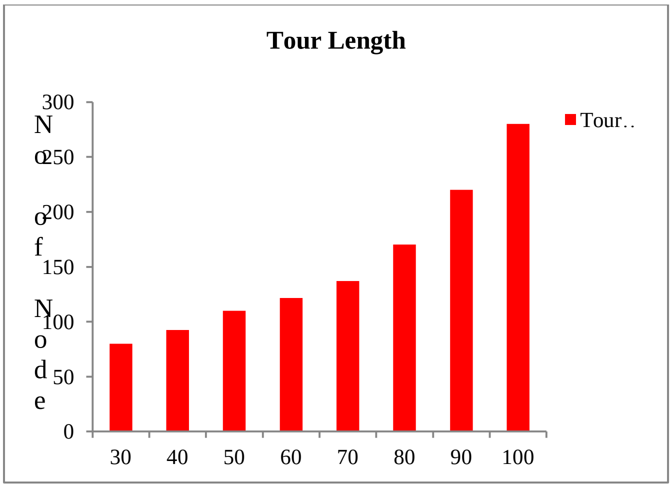

Figure.5

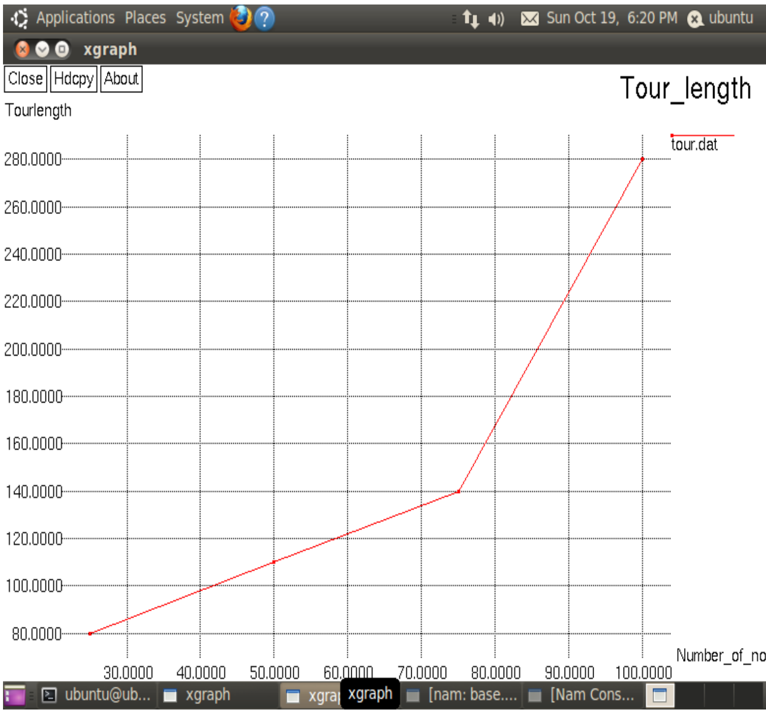

Figure.6

6. Conclusion:

In recent years, wireless sensor and actor (actuator) networks (WSANs) have started to receive growing attention due to their potential in many real-life applications. This paper has tackled an important problem in mission critical WSANs, that is, reestablishing network connectivity after node failure without extending the length of data paths. We have proposed a new distributed LeDiR algorithm that restores connectivity by careful repositioning of nodes. LeDiR relies only on the local view of the network and does not impose prefailure overhead. The performance of LeDiR has been validated through rigorous analysis and extensive simulation experiments. LeDiR also works very well in dense networks and yields close to optimal performance even when nodes are partially aware of the network topology. LeDiR can recover from a single node failure at a time.

7. References:

[1] M. Younis and K. Akkaya, “Strategies and techniques for node placement in wireless sensor networks: A survey,” *J. Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, Jun. 2008.

[2] Abbasi, M. Younis, and K. Akkaya, “Movement-assisted connectivity restoration in wireless sensor and actor networks,” *IEEE Trans. ParallelDistrib. Syst.*, vol. 20, no. 9, pp. 1366–1379, Sep. 2009.

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

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

[5] 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.

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

[7] 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.

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

[9] M. Younis, S. Lee, S. Gupta, and K. Fisher, “A localized self-healing algorithm for networks of moveable sensor nodes,” in *Proc. IEEE GLOBECOM*, New Orleans, LA, Nov. 2008, pp. 1–5.

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