International Journal of Computer Applications Technology and Research
Volume 4– Issue 12, 923 - 927, 2015, ISSN: 2319–8656

Redundant Actor Based Multi-Hole Healing System for Mobile Sensor Networks

S. Himabindu
VBIT
Hyderabad

B. Brahma Reddy
VBIT
Hyderabad

Abstract

In recent years, the Mobile Wireless Sensor Network is the emerging solution for monitoring of a specified region of interest. Several anomalies can occur in WSNs that impair their desired functionalities resulting in the formation of different kinds of holes, namely: coverage holes, routing holes. Our ultimate aim is to cover total area without coverage hole in wireless sensor networks. We propose a comprehensive solution, called holes detection and healing. We divided our proposed work into two phases. The first phase consists of three sub- tasks; Hole-identification, Hole-discovery and border detection. The second phase treats the Hole-healing with novel concept, hole healing area. It consists of two sub-tasks; Hole healing area determination and node relocation.

Key words: WSN, coverage holes, routing holes, hole healing.

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.

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. However, a failure of an actor may cause the network to partition into disjoint blocks and would thus violate such a connectivity requirement.

2. RELATED WORK

The [1] wireless communications systems in the mobile computers support a broadcast mechanism, much more flexible and useful ways of sharing information can be imagined. Our proposed routing method allows a collection of mobile computers, which may not be close to any base station and can exchange data along changing and arbitrary paths of interconnection, to afford all computers among their number a (possibly multi-hop) path along which data can be exchanged. In addition, our solution must remain compatible with operation in cases where a base station is available. By the methods outlined in this paper not only will routing be seen to solve the problems associated with ad-hoc networks, but in addition authors will describe ways to perform such routing functions at Layer 2, which traditionally has not been utilized as a protocol level for routing. Packets are transmitted between the stations of the network by using routing tables which are stored at each station of the network. Each routing table, at each of the stations, lists all available destinations, and the number of hops to each. Each route table entry is tagged with a sequence number which is originated by the destination station. To maintain the consistency of routing tables in a dynamically varying topology, each station periodically transmits updates, and transmits updates immediately when significant new information is available, since authors did not assume that the mobile hosts are maintaining any sort of time synchronization, authors also make no assumption about the phase relationship of the update periods between the mobile hosts. These packets indicate which stations are accessible from each station and the number of hops necessary to reach these accessible stations, as is often done in distance-vector routing algorithms.

In [2] report, authors focus on the specific problem of altering the positions of robots in order to achieve a desirable ad hoc network topology starting from an arbitrary initial spatial configuration. Every robot node includes its location information (GPS coordinates or indoor relative location information) whenever it floods an LSU (link state update) to the rest of the network. Additionally neighbour information of a node is also extracted from an LSU in order to construct a view of the current network topology.

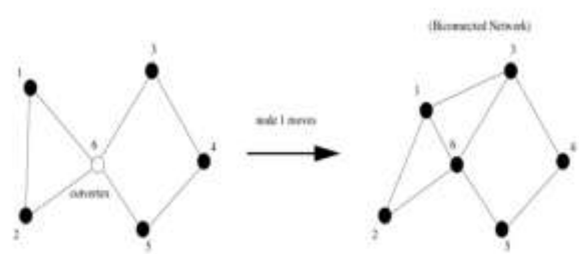

Figure 1: Achieving Biconnectivity by Node Movement

In [2] article authors show that iterative block movement algorithm significantly. Authors have considered a bi-connected network where nodes still can exchange messages with each other to coordinate the recovery process even after failure. And finding an exact polynomial time optimization algorithm for the 2D case is extremely hard. [3] Paper considers the connectivity restoration problem subject to path length constraints. Basically, in some applications, such as combat robotic networks and search-and-rescue operation, timely coordination among the actors is required, and extending the shortest path between two actors as a side effect of the recovery process would not be acceptable.

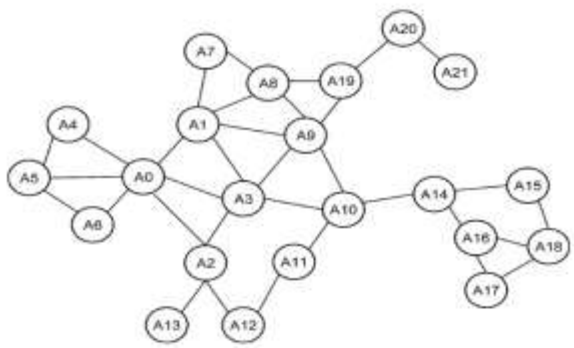

Fig. 1. Example one-connected inter-actor network. Nodes A_0, A_{10}, A_{14}, and A_{19} are cut vertices whose failure leaves the network partitioned into two or multiple disjoint blocks.

In [4] paper, authors designed and evaluated distributed self-deployment protocols for mobile sensors. After discovering a coverage hole, the proposed protocols calculate the target positions of the sensors where they should move. Author problem statement is: given the target area, how to maximize the sensor coverage with less time, movement distance and message complexity. In [4] paper authors addressed the problem of placing sensors in target field to maximize the sensing coverage. Although the centralized approach may minimize the sensor movement, central server architecture may not be feasible in some applications.

2.1 Existing system & Disadvantage

In existing research works, the researchers proposed the technique with movement assisted sensor node deployment and then Hole-detection works by a node decides whether it is on the boundary of a hole by comparing its degree with the average degree of its 2-hop neighbors. Not all boundary nodes can be identified correctly by this algorithm.

3. PROPOSED SYSTEM

Our proposed hole and border detection algorithm is distributed and lightweight, and thus more suited to the energy constrained WSNs. we propose collaborative mechanisms to detect and heal holes. Our hole-detection mechanism deals with holes of various forms and sizes. We try to alert a limited number of nodes surrounding the hole, only those nodes have the task of moving and repairing the hole. In this section, we are going to discuss about our enhancement work. Our base method works like reactive mode, if the node is failed then only Hole healing will be start. By our base work wes can cover the holes, but reactive mode will be cause to high level topology changes, and the more number of nodes has to move from own position. Due continuous node failure, the network may not be healed after certain healing process. To avoid this type of problem, we propose the extra temporary SensRob's. in our Enhancement we propose the failure detection based on the energy loss. The node will fail when it loosed remaining energy less than critical level. So in our enhancement, we propose the technique to detect failure of node due to the energy loss.

Each node in the network monitors the own energy loss, if own energy is getting reduce near to critical level then the node will inform to the base station about energy loss. Once base station received error message from the sensor the it will check availability of extra actor. The base node will share the position information of critical node and id details to the extra actor, and then the extra actor has to move to the critical node position.

After reaching the place of critical node, the extra actor has to inform to the critical node to inter change the id in neighbor's neighbor table. The critical node will send the inter change message to neighbor with extra actor id. The neighbors of critical node need to change their neighbor table info such as extra actor id will be placed as the neighbor sensor and neighbor sensor (critical node will be deleted). After id inter changing process, the critical node will be treated as extra actor and extra actor will be treated as normal sensor device, then the extra actor node (inter changed critical node) has to move to base station and its need to repair the energy level. This reclaimed extra actor can be placed in any other critical node in future.

3.1 Algorithm

<u>Hole detection & Healing :</u>

1) **Initialize the Htimer and Neigh_timer**
2) **If $Timer\ expire$**
   - **Generate the hello message**
     - $Pkt \leftarrow nd_{id} \& Pos(x, y)$
   - Broadcast Hello message
   - $Schedule(T_{now} + T_{Rand_{const}})$
3) **If Pkt Recv in node in n_i**
   - If $Pkt.type = Hello$
     - Set $Insert = true$
     - **Foreach $M \in Tbl_N$**

- If $Pkt.src = M.Id$
  - $M.T_{ex} = T_{now} + T_{const}$
  - $M.Pos(Pkt.x, Pkt.y)$
  - Set $Insert = false$
  - $Search_{stop}$
- If $Insert = True$
  - Allocate new Memory $M \rightarrow Tbl_N$
    - $M.Id = Pkt.src$
    - $M.Pos(Pkt.x, Pkt.y)$
    - $M.T_{ex} = T_{now} + T_{const}$

- Else-If $Pkt.type = Hole_{disc}$
  - If $Pkt.fid \in Nb_{list} \in Nb_{table}$ & $n_i \notin Pkt.path$ & $Pkt.src \neq n_i$
    - Set $dst = Dist(Pkt.SPos_{x,y}$ & $n_{i_{(x,y)}})$
    - If $dst > Pkt.dst_{mx}$
      - $Pkt.dst_{mx} = dst$
      - $Pkt.Cpos \leftarrow (x_{n_i}, y_{n_i})$
    - $n_i \cup pkt.path$
    - $Reset(Timer_{Hole_{disc}})$
    - Set $Pkt.hop = 0$
    - $Rebroadcast(Pkt)$
  - Else-If $Pkt.fid \neq Id_{fail}$ & $n_i \notin Pkt.path$ & $Pkt.src \neq n_i$ & $Pkt.hop < 1$
    - $Pkt.hop++$
    - $Rebroadcast(Pkt)$
  - Else-If $Pkt.src = n_i$
    - If $MDist_{fail} < Pkt.dst_{mx}$
      - Set $N.type = HM$
      - $MDist_{fail} = Pkt.dst_{mx}$
      - $x_c = (x_{n_i} + Pkt.Cpos_x)/2$
      - $y_c = (y_{n_i} + Pkt.Cpos_y)/2$
    - $Free(pkt)$
    - $Timer_{Hole_{Heal}}.Schedule(T_{now} + T_{Rand_{const}})$
  - Else
    - $Free(pkt)$
- Else-if $Pkt.type = Hole_{Heal}$
  - If $Near_{hole}$ & $n_i \notin Pkt.path$ & $Pkt.src \neq n_i$ & $(\nexists Lnk_{n_i, Pkt.Path_{i-1}} \,||\, \nexists Lnk\_(H_c, Pkt.Path_{i-1}))$
    - If $\left(Pkt.\frac{dst}{2}\right) > Dist\ (Cpos_{x,y}$ & $(x_{n_i}, y_{n_i}))$
      - Set $Future_{pos} \leftarrow \frac{Cpos_{x,y} + \left(x_{n_i}, y_{n_i}\right)}{2}$
      - $Mov(Future_{pos})$
    - $Rebroadcast(Pkt)$
  - Else if
    - If $!Near_{hole}$ & $n_i \notin Pkt.path$ & $Pkt.src \neq n_i$ & $\nexists Lnk\_(n_i, Pkt.Path_{i-1}.Ftr_{x,y})$
      - $Move\_(100 \rightarrow Pkt.Path_{i-1})$
  - Else
    - $Free(pkt)$

4) **If Node $n_i's$ $Timer_{Neig}.expired$**
   - **For each $M \in Tbl_N$**
     - If $M.T_{ex} < T_{now}$
       - Set $Id_{fail} = M.Id$
       - $Delete(M)$
     - $Timer_{Hole_{disc}}.Schedule(T_{now} + T_{Rand})$
   - $Schedule(T_{now} + T_{Rand})$
5) **Node $n_i's$** If $Timer_{Hole_{disc}}$**.expired**
   - Generate pkt
     - $MDist_{fail} = 0$
     - $n_i \rightarrow pkt.src$
     - $pkt.type = Hole_{disc}$
     - $(x_{n_i}, y_{n_i}) \rightarrow pkt.Spos$
     - $n_i \rightarrow pkt.path$
     - $Init\ (pkt.hop$ & $pkt.Cpos)$
   - $broadcast(pkt)$
6) If $Timer_{Hole_{Heal}}.expired$ in **Node n_i**
   - Generate Pkt
     - $Pkt.src = n_i$
     - $(x_c, y_c) \rightarrow Pkt.Cpos$
     - $MDist_{fail} \rightarrow Pkt.dst$
     - $Pkt.type = Hole_{Heal}$
   - $broadcast(Pkt)$

3.2 Pre-Failure Controller

Let, E_c for Current energy level, E_{Th} for threshold energy level, L_{critic} for critical node list, L_{Exact} for Available Extra actor list, Id_{Ex} for Extra actor Id, Pos for postion,

1) If $E_c < E_{Th}$
   a. Generate $Pkt.critical$
   b. $Pkt.Nd = N_{id}$
   c. Broadcast Pkt
2) If Pkt recv in N
   a. If pkt is Duplicate
      i. Free Pkt
      ii. $Return$
   b. If $Pkt.critical$
      i. If $N \neq BS$
         1. Rebroadcast Pkt
      ii. If $N = BS$
         1. $Pkt.Nd \cup L_{critic}$
         2. If $L_{Exact} \neq Null$
            a. $Id_{Ex} = L_{Exact}(1)$
            b. $Rearrange(L_{Exact})$
            c. $Move(Id_{Ex} \rightarrow L_{Critic}(1).Pos)$
            d. $Rearrange(L_{Critic})$
   c. If $Pkt.Exact_{arrive}$
      i. If $N \neq Nd_{Crical}$
         1. $Ignore(Pkt)$
         2. $return$

ii. $Switch_{neigh}(N \rightarrow Id_{Ex})$

iii. $Move(N \rightarrow BS.pos)$

4. REQUIREMENTS

Hardware: Single PC with **20** Gb Hard disc space 1Gb RAM

Software: Linux OS (Ubuntu 10.04)**,** NS2.34

Languages: TCL (Front end type project only)**, C++ (Optional)**

- **Result**

We have tested our output with ns2 simulator and we got a two results, one is NAM, Xgraph.

Our enhancement method provides best results such as no node failure and less movement.

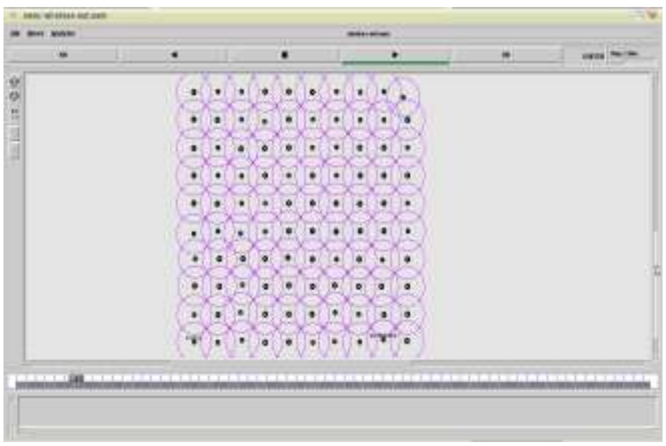

Fig. A1 Network placement and sensing area

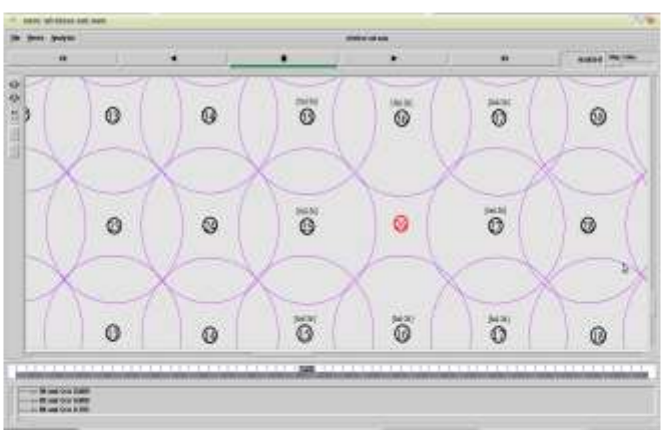

Fig. A2 Hole detection

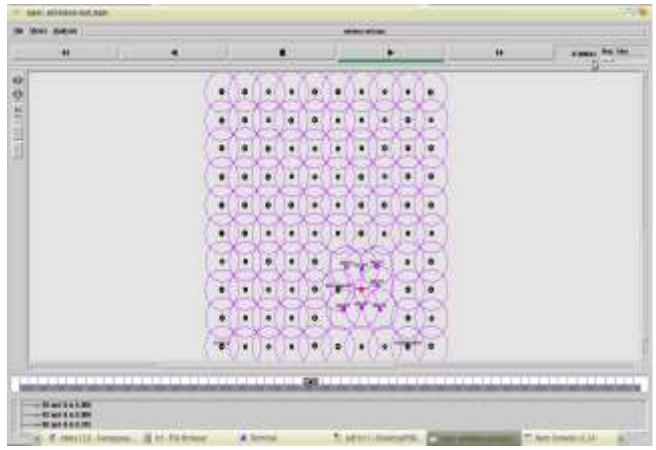

Fig. A3 Hole healing

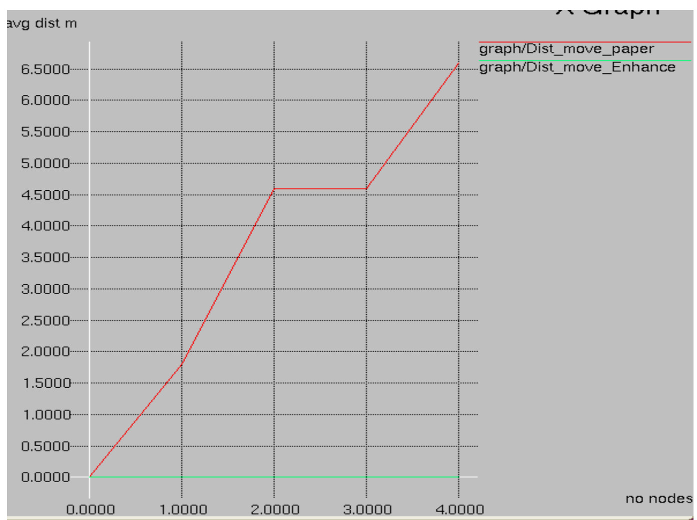

Fig.A4 Graph for avg moving distance

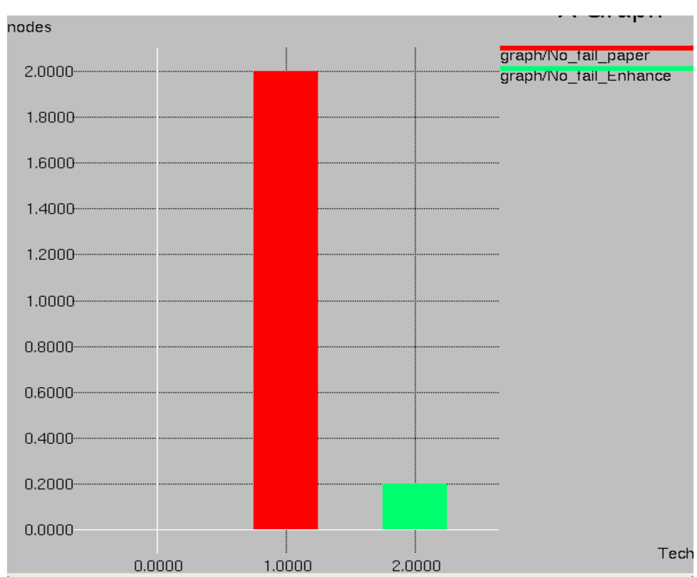

Fig.A5 Node failure

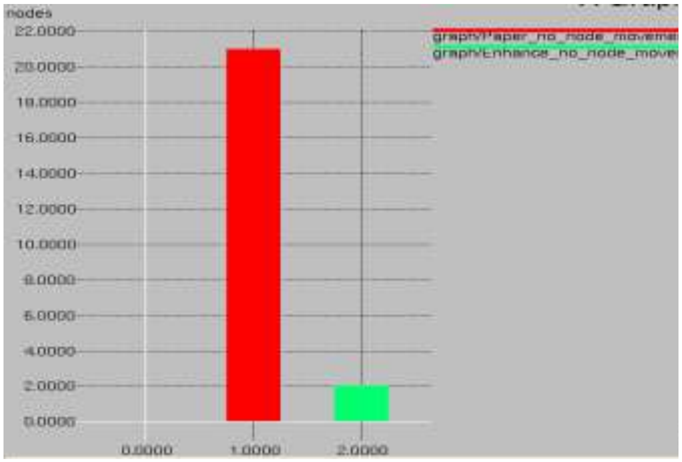

Fig.A6 Node movement

5. CONCLUSION

The Mobile Wireless Sensor Network is the emerging solution for monitoring of a specified region of interest. Several anomalies can occur in WSNs that impair their desired functionalities resulting in the formation of different kinds of holes, namely: coverage holes, routing holes. Our ultimate aim is to cover total area without coverage hole in wireless sensor networks. We propose a comprehensive solution, called holes detection and healing. We divided our proposed work into two phases. The first phase consists of three sub- tasks; Hole-identification, Hole-discovery and border detection. The second phase treats the Hole-healing with novel concept, hole healing area. It consists of two sub-tasks; Hole healing area determination and node relocation.

6. REFERENCES

1) *Taeyoung Kim,1 Minhan Shon,1 Mihui Kim,2 Dongsoo S. Kim,3 and Hyunseung Choo1 "Anchor-Node-Based Distributed Localization with Error Correction in Wireless Sensor Networks", 2012*

2) *"Accurate Anchor-Free Node Localization in Wireless Sensor Networks"* Adel Youssef, Ashok Agrawala and Mohamed Younis.

3) *"Movement-Assisted Sensor Deployment"* Guiling Wang, Guohong Cao, and Tom La Porta.

4) *"Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks"* Prithwirh Basu and Jason Redi.

5) 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.

6) K. Akkaya and M. Younis, "COLA: A coverage and latency aware actor placement for wireless sensor and actor networks," in Proc. IEEE VTC, Montreal, QC, Canada, Sep. 2006, pp. 1–5.

7) A. Youssef, A. Agrawala, and M. Younis, "Accurate anchor-free localization in wireless sensor networks," in Proc. 1st IEEE Workshop Inf. Assurance Wireless Sensor Netw., Phoenix, AZ, Apr. 2005.

8) S. Vemulapalli and K. Akkaya, "Mobility-based self route recovery from multiple node failures in mobile sensor networks," in Proc. 10th IEEE Int. Workshop WLN, Denver, CO, Oct. 2010.

9) S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks," IEEE Trans. Parallel Distrib. Syst., vol. 18, no. 8, pp. 1108–1121, Aug. 2007.

10) G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment," IEEE Trans. Mobile Comput., vol. 5, no. 6, pp. 640–652, Jun. 2006.