INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DETECTING INFECTED NODES AND BACKTRACKING MESSAGES FROM INFECTED AREA AND REROUTE IN WSN

Mamatha M[1], Mrs. Aruna M G[2]

[1] *M.tech, department of CSE,*
M S Engineering College, Bengaluru,
mamatha0721@gmail.com

[2] *Associate Professor, Department of CSE,*
M S Engineering College, Bengaluru,
aruna_mg@yahoo.co.in

Abstract- Wireless sensor network is widely used technology these days. Fault tolerance is a challenge in wireless sensor network. Failure of sensor nodes takes place for various reasons like, natural hazards, battery failure, software failure, etc. The sensed data can be attacked by malicious attackers; it has to be transferred to the destination without any change in the data. We have to identify the safe path and transfer the data. The infected area is avoided with the help of proposed technique called Bypass routing and twin rolling ball algorithm, which Re-route the data away from infected area. Twin rolling ball uses two balls which rotates in clockwise and anti clockwise direction to fing the next hop safe node. This paper also concentrates on energy efficiency criteria. Back tracking message is sent from infected nodes to the source nodes.

Keywords- Fuzzy C clustering, backtracking message, infected nodes, twin rolling balls, bypassing route.

1. INTRODUCTION

The wireless sensor network has various researches in monitoring application. They are used the sense the data, like weather, temperature, etc, and transfer to the destination. Wireless sensor networks are growing technology which is used worldwide. The sensor nodes may be damaged or infected. Software failure, hardware failure, or battery failure may occur. The energy constraint is mainly concentrated in wireless sensor network. The fault-tolerance is main issue in wireless sensor network, it s difficult to find the node failure and anomalies node.

2. PROBLEM STATEMENT

In Wireless sensor network, large numbers of sensor nodes are deployed in monitoring field. Because of sensor nodes limited capability, they are subjected to various failures (malware attacks, software and hardware failures). When a node gets fail or malfunction, that node can be referred *as infected*; such infected node will fail to do normal sensing and communication tasks. There are real challenges in ensuring timely and maintain accurate data delivery in emergency conditions.
Therefore, the problem needs to be considered in such scenarios
a) How to get the stuck packet out of the stucked region
b) How to bypass infected region and detour the incoming traffic towards uninfected region.

This paper focuses on detecting the infected node/infected region in WSN and avoids such infected nodes in further transmission and communication processes. . We addressed the problem that will be caused if the packets are transmitted into an infected region and we are also concerned with avoiding any infected regions which is crucial to prevent packets from being trapped and lost during transmission.

3. SYSTEM ANALYSIS

After analyzing the system requirement specification of the system, next thing is to analyze the problem and its context. First step towards designing a new system is study the existing system. Proper understanding of the existing system makes easy the task of designing the model of new system.

3.1 EXISTING SYSTEM:

In sensor networks whenever batteries go down it is not possible to replace, due to this reason most of the routing protocols optimizes the communication concentrating on energy efficiency. Most of the routing protocol uses local information from each node for the communication.

3.1.1 BOUNDHOLE:

The idea of by passing such holes is given by the BOUNDHOLE algorithm. This algorithm BOUNDHOLE is use to find the boundary of the hole or infected region and establishes an alternative routes to by-pass the identified infected nodes. This algorithm routes the packets based on GF algorithm. BOUNDHOLE algorithm every node needs remember the shape of the hole and for the further transmission the entire shape needs to be saved which occupies more memory and also another issue of this algorithm is false boundary detection which leads to loop problem.

3.1.2 GAR (Greedy Anti Void Routing)

Greedy Anti-Void Routing (GAR) has been designed to resolve false boundary detection in BOUNDHOLE approach. Similar to BOUNDHOLE, this method will be applied only when the packets fall in to local minima problem.GAR uses a method called Rolling ball technique. The Rolling ball technique is proven to be successful in avoiding infected regions. However, it tends to visit unnecessary nodes causing longer routing delays. The definition of Rolling ball and its Problem is given as follows:

Definition 1 (Rolling Ball) . In a given set of sensor nodes Ni ∈ N, we say a circle is a Rolling Ball (RB) if there exists a circle (RBNi (Si,R/2)) that is attached at a centre point Si∈R2 with a radius of (R/2). No other node such as Nk ∈ N should be located inside the rolling ball.

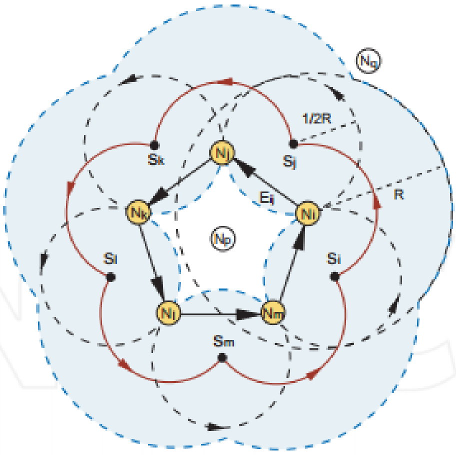


Fig 3.1.3: Rolling Ball Operation

Given Si as the starting point associated with the ball (RB Ni(Si,R/2)) is attached at the centre of the node Ni, it will rolled in a clockwise or counter-clockwise until the next node Nj touches the rolling ball. The same procedures apply to Nj and the same procedure is conducted continuously until the ball enters a termination phase.

3.2 The Proposed Solution:

To resolve the problems mentioned in the existing solutions (GF, BOUNDHOLE, GAR) it has been proposed to new method, By Pass Routing technique.

The proposed routing technique consists of two parts, Identifying Infected Area and by passing the route. Fuzzy data clustering is adopted to detect the infected node and this information used to detour the packets through unaffected nodes using by-passing routing technique. The second part BPR uses information on infected node and diverts the traffic and by pass the routing. Local minima node where the packets are stucked Twin Rolling balls technique is introduced to find the next 1-hop node quickly than the existing approach.

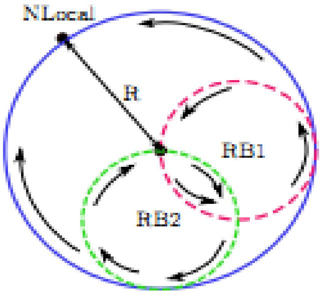


Fig 3.2.1: Twin-Rolling-Ball

4. SYSTEM ARCHITECTURE

The system architecture is as shown below in fig 4.1 it contains all the algorithms used in the paper. There are several algorithms which are implemented based on our requirements specification.

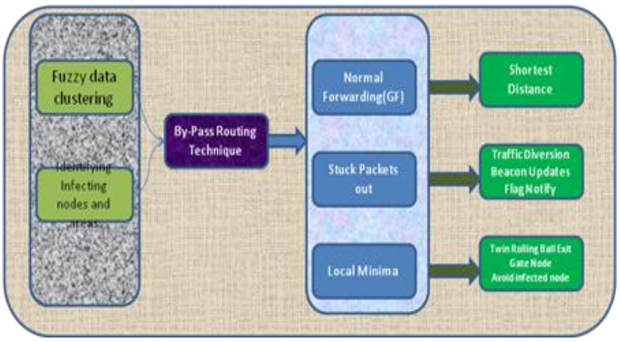

Fuzzy data clustering
Identifying Infecting nodes and areas
By-Pass Routing Technique
Normal Forwarding(GF)
Stuck Packets out
Local Minima
Shortest Distance
Traffic Diversion Beacon Updates Flag Notify
Twin Rolling Ball Exit Gate Node Avoid infected node

fig 4.1 system Architecture

4.1 ALGORITHMS USED

4.1.1 AIA algorithm (Avoiding infected areas):

In this phase, all the network parameters are initialized and divided into 3 steps

- The BS (base station Node) broadcasts the packet to all the nodes at the beginning of the time and computes Energy level

- Receiving the information and record the same and records a table in their memory which contains information of the neighbors.
- Data clustering happens using Fuzz C means of algorithm.

First, to get the stuck or trapped packets out of the corrupted regions in a timely manner by observing real-time applications maximum delays of 150 ms. second, we are also concerned about the incoming traffic that will have to be detoured in order to avoid them from being sent to the infected region.

4.1.2 Twin Rolling Balls Routing (TRBR):

Once the infected packets and the nodes that they are residing have been identified, we need to define the boundary nodes to route the packets away from the infected areas. The detection of the boundary nodes in the proposed method is inspired by the rolling ball [25] technique as described in Section 3.2. However, unlike the previous method [25], the angle of rotations are in both directions; clockwise (dc) and counter-clockwise (dcc) as shown in Figure 7. Rotating in just one direction may take a longer time if the node happened to be located far away from the ball. We counter this problem using two different balls that are attached to the same point (NLocal) and rotate the balls in different directions.

Besides avoiding a close-loop formation and false-boundary detection, this approach automatically avoids packets from being forwarded to the wrong nodes as in BOUNDHOLE. Furthermore, the proposed twin rolling balls ensures faster detection of the next hop since the node can be hit by any of the balls in any directions. For example, if a node is located closer to the ball in clockwise direction, rotating the ball in counter-clockwise may result in longer delay. We define the following properties.

4.1.3 Forwarding the Stuck Packets

The path selection using our approach results in shorter path diversion. This is because we avoid visiting the unnecessary nodes that will lead to an undesirable longer routing path. Unlike the existing rolling ball technique, once the local minima occurs, there will be two balls attached at the local minima node that rotate in two directions simultaneously; clockwise and counter clockwise. Our method suggests that the first node that hits the ball in either direction and is not infected will be chosen as the next hop (N6 using counter-clockwise rotation). This node also determines the direction for the rest of the rotation. The ball continues moving counter-clockwise and the same process continues until all the nodes within the communication range (R) of NLocal are used.

4.1.4 Back Tracking

This Phase is used to back Track the message from infected area to the source nodes. Then the source node will send the date in alternative path which is shortest and safe path. The back track messages are sent from all the nodes which are infected in the route path.

4.1.5 By-Passing Infected Areas

The alternative route to detour the affected packets. The initial phase of the proposed BPR technique is based on the GF algorithm. Neighbours' location and distance to other neighbours are obtained through frequent beacon updates and kept in each node's routing table. There are three processes in this method. First is flag notification of the infected nodes. This is followed by traffic diversion and finally the beacon updates.

4.1.5 Flag Notification of the Infected Nodes

Through the Fuzzy data clustering technique, each node is aware of their infection status. Once infected, the corresponding node will quickly notify the source node so that it will no longer receive any incoming packets. Here, the flag is set to 1 if there is any infection, and stays 0 in normal mode. The back-pressure is a message sent backwards to notify the senders of any events. This will also involve intermediate nodes that reside within the same route with the affected nodes. Upon receiving this notification, source node will stop sending through the infected node

4.1.7 Traffic Diversion

Each intermediate node knows the position and the shortest distance to their 1-hop neighbour. This information is obtained through periodic beacon updates between nodes. At this point, a node's routing table will only contain a fresh list of its 1-hop unaffected neighbors. This will be the nodes located outside the boundaries of the affected regions and thus will be able to forward the packets to the correct destinations. In order to send a packet, a 1-hop neighbour with the closest geographical distance to destination will be chosen. The identification of infected nodes prior to transmission is crucial to discover the way to divert the incoming traffic away from the infected areas. This method avoids the packets from being trapped in there and lost. This saves considerable time and resources for retransmissions and ensures that packets are sent with the least possible delay.

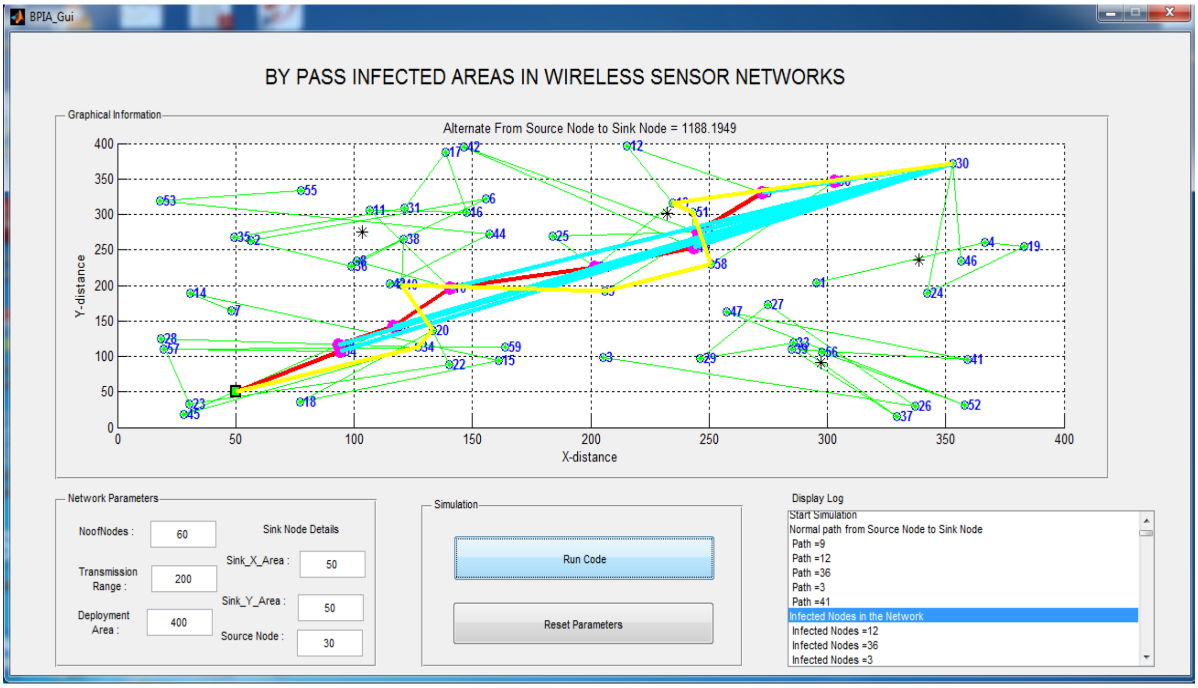
4.1.8 Beacon Updates

The tested and justified impact of choosing a particular amount of updates in the evaluation section is used. If the ACK is not received from the 5th node after a certain threshold, the source node will retransmit the same packet from which the ACK is missing.

The source node will piggyback any received infection notification message to downstream nodes. The first node that receives the notification will change its next hop information and send the packets to the next available hop in its routing table. Nevertheless, considering a larger duration in sending the ACK (e.g. 10th nodes) may lead to considerable delay in detecting errors, thus waste substantial energy. For example, if the first node is infected during transmission, it can only be detected after the 10th node.

5. RESULTS

This paper is implemented using MATLAB, the performance evaluation is done using MATLAB simulation tools the snapshot is as shown below fig 5.1. This figure shows the infected path and bypassing in shortest path and safe path. Backtracking message is sent from infected nodes to source node.



NoofNodes : 60
Transmission Range : 200
Deployment Area : 400
Sink Node Details
Sink_X_Area : 50
Sink_Y_Area : 50
Source Node : 30

```
Start Simulation
Normal path from Source Node to Sink Node
Path =9
Path =12
Path =36
Path =3
Path =41
Infected Nodes in the Network
Infected Nodes =12
Infected Nodes =36
Infected Nodes =3
```

Fig 5.1 simulation scenario

Conclusion

We have studied the effectiveness of our proposed By-Passed Routing (BPR) in avoiding infected areas and its efficacy in improving the overall performance. The infected areas are anomalous nodes detected using a fuzzy data clustering method and the information collected is used in the proposed BPR technique. The backtrack message is sent from infected nodes to source node, which helps in rerouting the message. Performance evaluation is done to show the effectiveness of our proposed technique.

REFERENCES

[1] A. Abdulla, H. Nishiyama, J. Yang, N. Ansari, and N. Kato, “HYMN: A novel hybrid multi-hop routing algorithm to improve the longevity of wsns,” IEEE Transactions on Wireless Communications, vol. 11, no. 7, pp. 2531–2541, July 2012.

[2] M. Ahmadi Livani and M. Abadi, “An energy-efficient anomaly detection approach for wireless sensor networks,” in IST: 5th International Symposium on Telecommunications, 2010,pp. 243–248.

[3] N. Ahmed, S. S. Kanhere, and S. Jha, “The holes problem in wireless sensor networks: a survey,” SIGMOBILE Mob. Comput.Commun. Rev., vol. 9, no. 2, pp. 4–18, Apr. 2005.

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[5] N. Arad and Y. Shavitt, “Minimizing recovery state in geo-graphic ad hoc routing,” IEEE Transactions on Mobile Comput-ing, vol. 8, no. 2, pp. 203–217, 2009.

[6] Ashwini and P. A. S, “Information dissemination between nodes of different intersections intersection in city environ-ment using hop greedy routing protocol (BAHG),” Interna-tional Journal of Ethics in Engineering and Management Eductaion, vol. 1, no. 4, pp. 232–236, April 2014.

[7] D. Chen and P. K. Varshney, “On-demand geographic forward-ing for data delivery in wireless sensor networks,” Computer Communications, vol. 30, no. 1415, pp. 2954 – 2967, 2007.

[8] S. Chen, G. Fan, and J. hong Cui, “Avoid ”void” in geographic routing for data aggregation in sensor networks,” International Journal of Ad Hoc and Ubiquitous Computing, vol. 1, pp. 169–178, 2006.

[9] R. Di Pietro, L. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, “Data security in unattended wireless sensor net-works,” IEEE Transactions on Computers, vol. 58, no. 11, pp.
1500–1511, Nov 2009.

[10] Q. Fang, J. Gao, and L. Guibas, “Locating and bypassing holes in sensor networks,” Mobile Networks and Applications, vol. 11, no. 2, pp. 187–200, 2006.