Algorithms for Dynamic Node Events and Fault Tolerance in Wireless Sensor Networks

V.Srikanth, D.Swathi, M.Tabassum* I.Ramesh Babu^
srikanth_vemuru@yahoo.com swathid286@yahoo.com tabassum_mahanaz@yahoo.com*
rinampudi@hotmail.com^
Dept of IST, K.L. college of Engineering, Vaddeswaram, Guntur 522502*
Dept of CSE, Acharya Nagarjuna University, Guntur 522502^

Keywords: fault tolerance, dynamic node events, sensor networks

Abstract

Wireless sensor networks are new type of emerging networks with bunch of applications in all fields due to their low cost and low power scheme. As these networks follow open wireless communication, they undergo dynamic node problems and fault node detection due to malicious node activities. In this paper we overcome these problems with the help of algorithms that will provide uninterrupted communication and also save node's energy and time.

1. Introduction

A typical wireless sensor network with a number of sensor nodes and a base station has a variety of applications due to their low cost and scalability. Each sensor node is a small, cheap device with limited battery power, small memory and low computational ability. The base station is an aggregation point for collected data and is assumed to have greater memory and processing power, and does not have the constraints of the sensor nodes. These networks are vulnerable to routing level attacks[1] which potentially strand well behaved nodes and cause large holes in the sensing environment. This leads to dynamic node events like link failure or node death.

2. Dynamic Node Events

Nodes commonly rely on each other to route messages to a base station in WSNs which conserves power. When a base station ceases to receive response from a region of nodes it can't immediately determine due to node destruction or node failure or node death or link failure between nodes. Failure of nodes in network may be due to depletion of their batteries or security breaches.

The failure of a single node can result in a whole network becoming silent as sensor nodes route measurements in a tree like fashion with base station as its root. This ends up in network remaining unable to complete its intended function. It is therefore critical for the base station to find out the reason for the node failure. To distinguish these cases, the base station needs to trace all dead nodes.

Another interesting challenge for sensor network is reseeding, or addition of new nodes to fill up dead nodes in already framed network In order to implement this, base station need new node's information and initialization phase of protocol used should be friendly for this kind of activity.

2.1 Possible solutions

One solution to this problem can be rerunning the entire protocol, an expensive process in terms of the number of additional messages that must be exchanged. In the later part of the paper we propose a solution for these dynamic node events in the form of algorithms with modification in secure unicast messaging protocol SUMP[7].

3. Fault tolerance

During the communications between the nodes, we come across some challenges like validity of the packets received and realization of fault tolerance. Validity of node must be ensured, because the transmission of invalid data may lead to integrity problems. Fault nodes[4] in a network lead to reduced throughput which decreases efficiency and performance of network. So identification of fault nodes should be done.

3.1 Possible Solutions

In order to detect these fault nodes, we have many systems like SCALE, SYMPATHY which are implemented using existing centralized

architecture. This approach is not scalable, inaccurate and lack of assumption of high data rate back channel.The above limitations can be overcome with usage of de-centralized architecture. We proposed an algorithm for fault tolerance which is available in later part of the paper.

4. Security

While directing communication towards their ultimate destination the sensor nodes may face the security issues. To prevent them, a light weight group authentication method is required.

4.1 Possible solutions

Several group authentication methods are available for membership testing. But due to the constraints of the sensor nodes, many of them are not feasible.Benaloh et al.[6] proposed the use of one way accumulators for authentication, where one key value is stored by a node and messages are prepended with authentication values. This method requires less memory but requires computation to verify membership. Liu et al. [2] propose a key chain commitment that requires the node to store several key values, increasing the memory requirement on the nodes. Merkle hash trees, as discussed in [8], exhibit a low computational cost and storage requirement but still provide secure group membership authentication. This will be best suited for security basis for implementation models like SUMP.

5. Related Work

5.1 Dynamic node events

It is found that dynamic node events for sensor network during unicast of messaging remain open. We overcome them by performing the following modifications during the initialization and messaging phase in SUMP.To identify whether a node is dead or alive and adding up a new node to a network we made the following assumptions:

1. path established for the network should be biconnected i.e. failure of one node should not disconnect entire network
2. Node should not face any failure at articulation points of network.
3. new nodes should be available at articulation point ready for initialization
4. the base station should be able to communicate with any node in network without dependent nodes.
5. In above assumption, if dependents exist, they should not face any failures.

5.1.1 Initialization

The initialization phase of SUMP comprises of path establishment and verification. During path establishment phase we add an extra field response time to each node structure in addition to its ID and path length. When the hello message is sent from the base station to all nodes, the base station records response time simultaneously along with path.Once multiple paths are established with nodes in the network, threshold value is calculated by sending “hello” message to each node using the multiple paths which is recorded. When the “hello” message is send to node it starts the incrementing response time until it receives a message from same node which can be identified by checking the node’s id.

Steps to initialize response time:

1. Send message to node [i]
2. do
   Increment response time
3. While (received mess id= = id of send node[i]);
4. Store response time of node[i] =response time
5.Use multiple paths for node[i] and send message
6. update its threshold node[i] value
7. End

In verification step the base station verifies the hop count of each node based on path length recorded along with response time of each node. If a discrepancy exists in calculating the hop count, the base station rectifies this by sending a hop count change request.

Steps for updating response time:

1. Send hop count change request to node [i]
2. Do
   Increment response time
3. While (received mess id= = id of send node[i]);
4. If(oldresponsetime node[i]<newresponsetime)
1. Store responsetime node[i]=newresponsetime
2.Use multiple paths for node[i] ,send message
5. If(oldthresholdtime node[i]<newthreshold time) then
Store thresholdtime node[i]=newthresholdtime)
6. End

5.1.2. Messaging

During communication, node will not respond properly due to network failures or power shortage. In that case, we have to detect whether the node is dead or alive. If the node is dead, it will be replaced by a new node depending on requirements. If node is alive and communication link breakage had occurred then new link will be established. This is done by verifying the response and threshold times of node during messaging phase. The advantage is that it will not waste the much energy of the node.

Steps for identifying dead nodes and taking necessary action:

1. Message is send to destination node[i]
2. Do
   3. Decrement response time
   4. if (received mess id= = id of send node[i])
      Begin
   1. Declare destination node and all nodes in that path are also alive
      2. Exit
      End
5 while (response time of node[i]= = 0);
6. Declare one of the nodes[i] in the path is dead or link failure.
7. for each node[i] in the path
   Begin
   1. Check response time for each node by Alternating path available
   2. Identify the failed node [i]/nodes
   End
8. Do
   1. Decrement threshold time of node[i]/nodes
   2. Wait for the nodes reply
   3. If (received mess id= = id of send node[i])

Begin
   1. Communication link failure
   2. Use alternative communication links
End
9. While (threshold time of node[i]= = 0);
10. Declare the node is dead
11. If (new node = = available) then
   12 initialize the new node with old nodes values along with response time
   13. Else
   14. Return the dead node and use alternating path for communications
   15. End

Let us consider a network with different communication links and malicious nodes activities as shown in figure1.

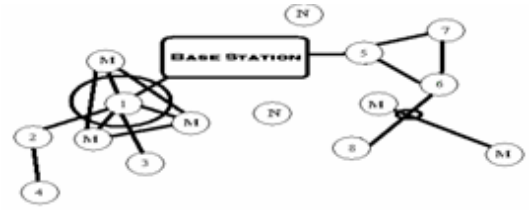
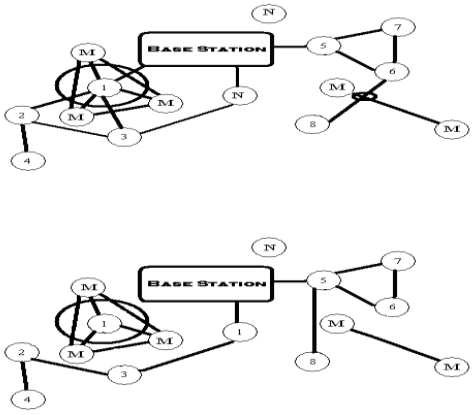


Figure 1: network with valid nodes, malicious nodes M, new nodes N

Here node 8 has link breakage; node 1 is dead due to malicious node activities. We apply the algorithm as shown in figure 2.The link breakage of node 8 is identified during message phase and the link between nodes 6and 8 is de-allocated, new link between nodes 5 and 8 is established. The dead node 1 and its old links are de-allocated and a new node is initialized and new links are established.



.

Figure 2: Network free from failures.

5.2. Fault tolerance:

The success of a WSN lies in realization of fault tolerance (FT)[3]. To achieve high quality of data (QoD), the requirements for FT[5] in WSNs are awareneness of network main operation, status of network resources and adaptability to frequent changes in WSNs conditions. We have proposed an algorithm which can be implemented using decentralized architecture which satisfies above requirements. Assumptions made for dynamic node events are even applicable for fault tolerance in the network. The power of this leader node is checked from time to time to ensure that its power is sufficient to carry out the process.

Steps for detection of fault nodes in network:

1. for all nodes
BEGIN
   1. Select node[i] if hopcount[i] = =1;

```
  2. Group [1] =node[i];
  3. For j=2 to n do
       BEGIN
          1. Group[j] =pathestablishment (i);
          2 . Increment j;
       END
   4. for all nodes in group
         Power [i] = (d*d)/(j*j);
   5. Lnode=max (power);
END
2. for each group
  BEGIN
   1. If (power (Lnode)>minpowreq)
      BEGIN
        1. Set timermin=0;
        2. Node[i].data <- datapackets;
        3. Qinsert (responsetime[i]);
           Wait till threshold time;
        4. Lnodebuffer=qdelete (responsetime[i]);
        5. For all nodes in node buffer,
         BEGIN
          1. f(responsetime[i]==responsetime[i+1])
            BEGIN
              1. Busy[i] = Busy [i+1] = 1;
              2. ENABLE RTS (i,i+1);
              3. until response[i];
                  Wait for CTS
              4. If CTS received
                 Busy[i] = Busy [i+1] =0;
              5.  Else
                  Wait threshold time
                    1. If CTS received
                       Busy[i] =Busy [i+1] =0;
                    2. Else
                          Declare hidden terminals
                          are i, i+1
          HENCE FAULT NODES ARE   i,i+1;
            END
     2. Else Allow the concurrent messaging and
        i,i+1 are correct nodes
     3. Power [lnode] =power[lnode]-
                          Sqrt ( ( log j+γ (j) )/∏ j );
    END
  2. Else Repeat selection of leadernode among
all  nodes in respective groups
  END
3. END //end of algorithm
```

Let us consider the network shown in figure 3 which is assumed to be free from any malicious activities in the beginning. But in the course of communication, an adversary may take control over a node and inject false data. This results in a faulty node which may end up in erroneous information to the entire network which is not desirable.

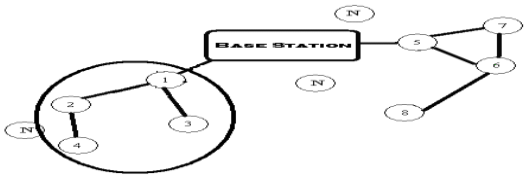


Figure 3: Valid nodes, new nodes, group

As per the proposed algorithm, node 1 has a direct link with the base station it forms a group with nodes 2,3,4. Node 2, having more transmission capacity is the leader node. It establishes links with all nodes (1, 3, 4) and interlinks them. It then sends data packets followed by time stamp request to all nodes in its group. These response times are stored in an array buffer and are compared with one another. Nodes 1 and 4 have the same response time. So the Leader node assumes them to be busy, enables RTS (Request to send) and monitors their response times. If CTS (Clear to send) is not yet received, then it waits for the response from these nodes until threshold time. If it does not receive any response from the nodes within the time bound, nodes 1 and 4 are detected as faulty nodes in the network and are replaced by new nodes.

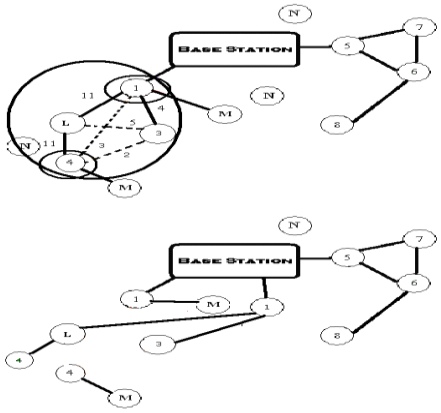


Figure 4: Network free from fault tolerance

6. Conclusion:

The Dynamic node events and fault nodes detection in Sensor Networks can be solved by implementing the algorithms provided, in any unicast light weight protocols in order to provide uninterrupted communication.

7. References

[1.]C. Karlof and D. Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures". In *Proc. of First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[2] D. Liu and P. Ning. "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks". In *Proc. of 10th Annual Network and Distributed System Security Symposium (NDSS 03)*, San Diego, CA, 2003.

[3] F.Koushanfar, M.Otkonjak, A.Sangiovanni Vincentelli. Fault tolerance in wireless adhoc sensor networks. In IEEE Sensors, pages 1491-1496,june 2002.

[4] Hwang, J.; He, T.; Kim, Y. Detecting Phantom Nodes in Wireless Sensor Networks INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE Volume , Issue , May 2007 Page(s):2391 – 2395 Digital Object Identifier 10.1109/INFCOM.2007.287

[5] I.Saleh, M.Eltoweissy, A.Agbaria H. El-sayed a fault tolerance management framework for wireless sensor networks, journal of communications, vol.2.no.4, june 2007

[6]. J. Benaloh and M. de Mare. "One Way Accumulators: A Decentralized Alternative to Digital Signatures". In *Proceedings of Advances in Cryptology (EUROCRYPT'93)*, Vol. 765, *Lecture Notes in Computer Science*, *Springer*, 1993.

[7] Jeff Janies, Chin-Tser Huang, Nathan L. Johnson SUMP: A Secure Unicast Messaging Protocol for Wireless Ad Hoc Sensor Networks *Department of Computer Science and Engineering*

[8] R. Merkle. "Protocols for Public Key Cryptosystems". In *Proceeding of the IEEE Symposium on Security and Privacy*, 1980

8. About Author

V.Srikanth, working as Assistant Professor in Department of Information Science and Technology. He has done his Masters and Bachelors in Engineering from Madras University. He is currently pursuing his Ph.D from Acharya Nagarjuna University. His area of interests include Mobile Ad Hoc Networks, Embedded Systems and Computer Networks. He is having Industrial and Research experience in Computer Networks.

I.Ramesh Babu,professor in Computer Science and Engineering in Acharya Nagarjuna University. He has done his Masters and Bachelors from Andhra University and Mysore University. He has done his Ph.D in Nagarjuna University during 1994. He is a senior member of IEEE. He has published 41 papers in international and National Journals/Conferences. He has guided four Ph.D and two Phil's projects. His area of interests include Wireless Sensor Networks and Adhoc Networks.

D.Swathi, student doing her third year of Bachelor of Technology course in Koneru Lakshmaiah College of Engineering. She has presented papers at national level. She is actively pursuing her research on security in wireless sensor networks. Her area of interest is wireless sensor networks and Adhoc Networks.

M.Tabassum, student doing her third year of Bachelor of Technology course in Koneru Lakshmaiah College of Engineering. She has presented papers at national level. She is actively pursuing her research on security in wireless sensor networks. Her area of interest is wireless sensor networks and Adhoc Networks