International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-2 Issue-5, April 2013

Distributed Fault Tolerant Algorithm for Identifying Node Failures in Wireless Sensor Networks

Navneet N Tewani, Neeharika Ithapu, K Raghava Rao, Sheik Nissar Sami, B. Sai Pradeep,V.Krishna Deepak

Abstract*—A Wireless Sensor Network is a set of multiple connected components. Sometimes due to the failure of some of its nodes, the sensor network communication fails. So that we consider this problem of node(s) failure termed as "cut" from the remaining nodes of a wireless sensor network. We propose an algorithm that allows (i) every node to detect when the connectivity to a specially designated node has been lost, and (ii) one or more nodes (that are connected to the special node after the cut) to detect the occurrence of the cut. The algorithm we proposed is distributed and asynchronous i.e. every node needs to communicate with only those nodes that are within its communication range. The algorithm is based on the iterative computation of the nodes. The convergence rate of the underlying iterative scheme is independent of the size and structure of the network. In this algorithm we devised a way to solve the problem of redundant information at the destination which arises due to availability of information at every node that is initially sent from the source node. We demonstrate the effectiveness of the proposed algorithm through simulation.

Index Terms*--- Cut, iterative computation, redundancy, simulation, Wireless sensor networks.

I. INTRODUCTION

A Wireless Sensor Network consists of spatially distributed sensors in order to monitor the physical or environmental conditions such as temperature, sound, pressure so that it can pass their data from the source to the destination. The WSN is buildup of hundreds and thousands of "nodes" where each node is connected to sensor(s). Small size and low cost of the nodes make them attractive for widespread deployment and also causes a disadvantage of low-operational reliability. Multihop paths in the network are reduced due to failure of nodes which is quite common due to limited energy budget of the nodes supplied by small batteries. Mechanical/Electrical problems, environmental degradation, hostile tampering, battery depletion are some of the factors that cause node failure. Two nodes are said to be disconnected if there is no path between them.

Due to node failure the subset of nodes gets disconnected from each other resulting in a "cut", and to detect a "cut" we assume a specially designated node in the network which is known as the "source node". It acts as a base station and serves as an interface between the network and its users. We have two distinct outputs of a cut for a particular node because a cut may or may not separate a node from the source node. Let us consider a node "p" which may undergo two events.1) If "p" is disconnected from the source node then we say that a DOS (Disconnected from Source) event has occurred for "p".2) If a cut occurs in the network and that does not separate "p" from the source node then we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred for "p". By cut detection we mean, 1) Detection by each node of a DOS event when it occurs, and 2) Detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. The "approximate location" of a cut means the location of one or more active nodes that lie at the boundary of the cut and that are connected to the source. Nodes that detect the occurrence and approximate locations of the cuts can then alert the source node or the base station. To detect these two outcomes we propose a distributed algorithm that allows every node to monitor the topology of the (initially connected) graph and detect if a cut occurs. In this article we propose a distributed algorithm to detect cuts, named the Distributed Cut Detection (DCD) algorithm. The algorithm proposed here is asynchronous and distributed which involves only local communication between the neighboring nodes, and is robust to temporary communication failure between node pairs. We demonstrate the implementation of the algorithm in JAVA.

II. BACKGROUND

A. Node Failure

The Node failure is expected to be quite common in WSN**,** due to their extremely limited energy budget and environmental degradation. This scenario is mostly true for the sensor networks that are deployed in harsh and dangerous environments for such as forest fire monitoring. When a number of sensors fail for whatever may be the reason the resulting network topology may be disconnected which in result is considered as a failure of set of nodes. The nodes that have not failed become disconnected from the rest of the network.

Manuscript published on 30 April 2013.
***Correspondence Author(s)
Navneet N Tewani, Department of Electronics and Computer Engineering , K. L .University, Vijayawada, India.
Neeharika Ithapu, Department of Electronics and Computer Engineering,K.L.University,Vijayawada,India.
K. Raghava Rao, Professor, School of Computing,, K L University, Vijayawada, India. Research areas include WSN, Data Mining and Sensor Web Technologies and also worked in Singapore for 3 years on Java Technologies.
Sheik Nissar Sami, Department of Electronics and Computer Engineering, K.L.University, Vijayawada, India.
B. Sai Pradeep, Department of Electronics and Computer Engineering, K.L.University,Vijayawada,India.
V. Krishna Deepak, Department of Electronics and Computer Engineering , K. L .University, Vijayawada, India.

The network topology changes resulted by the node mobility and node state transitions due to the use of power management or energy efficient schemes may be detected as node failures. A highly dynamic network greatly increases the complexity of failure management. Unexpected node failure is handled through redundancy in the network and backup pointers to reestablish damaged links. When a node failure occur the subset of nodes get disconnected from the network which results in a "cut". The cut occurred prevents the data from reaching the destination, the subset of nodes that gets disconnected from the source form a cut area which is known as a "hole".

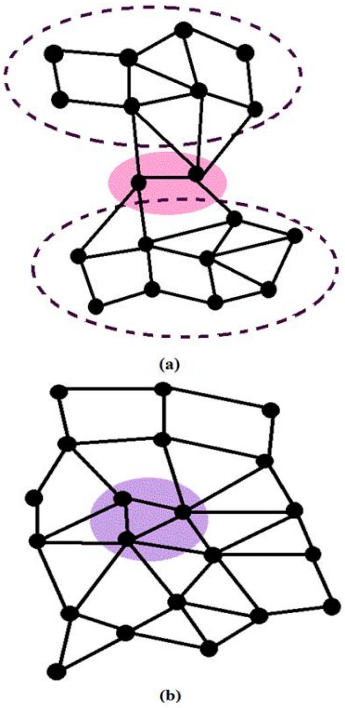

Fig.1 Example of cut and hole. Here filled circles represent nodes and solid lines represent edges. (a)Nodes in the colored circle are failed nodes. (b)The

B. Related Work

In [1] it is described that a Wireless sensor network (WSN) typically consists of a large number of small, low-cost sensor nodes distributed over a large area with one or possibly more powerful sink nodes gathering readings of sensor nodes. The sensor nodes are integrated with sensing, processing and wireless communication capabilities. To detect the node failures we assign a specially designed node known as source node through which we detect the DOS and CCOS events. The challenges due to the partitioning the networks have come forward in various papers. A paper by Kleinberg et al [2] considers the cut detection problem in wired network. In one of the paper G. Dini et al [3] has proposed a method of repairing the disconnected network by using mobile nodes. In another paper [4] Shrivastava et al proposed a randomized and deterministic algorithm to detect network separation using a set of sentinel nodes to monitor linear cuts in a network. In [5], [6] the authors select a source node and make it broadcast a message throughout the network. Boundary nodes detect a cut if they miss that message more than specified number of times. Using the concept in [7] we have proposed an algorithm that not only detect an arbitrarily shaped cut but also enables every node in the network to detect a cut in distributed manner. During the transmission of the data from source to destination the copy of data is sent to each and every node which may cause the problem of redundancy and memory wastage. Here we propose the algorithm in such a way that even though there is a cut in the network the transmission takes place successfully without any redundancy issues. The algorithm proposed is extension to the proposed algorithm in [8]

III. FAILURE NODE DETECTION IN WIRELESS SENSOR NETWORKS

A. Problem Definition

A canonical problem caused by separation of network is partitioning caused due to node failures. Failure of set of nodes will reduce the number of multi-hop paths in the network. Due to this reason a subset of nodes that have not failed will get detached from the rest causing a **cut**. In this paper we have deployed an algorithm to detect these cuts to allow the continuous flow of the data among the nodes. The problem we seek to address is diploid. Here firstly we need to enable each and every node to detect if it is disconnected from the source (i.e. a DOS event has occurred). Second we need to enable nodes that lie close to the cuts but are still connected to the source to detect the CCOS events and alert the source node.. We proposed a way to determine a subset of nodes that are disconnected from the assumed source node. We are also interested in devising a way to solve the problem of redundancy at the destination which arises due to availability of information at every node that is initially sent from the source node.

B. Distributed Cut Detection algorithm to identify Node failure

While the transformation of data some of the nodes stop working which results in a cut. To avoid and detect this problem here we propose a distributed and asynchronous algorithm known as Distributed Cut Detection. The algorithm consists of nodes, updating their local state periodically by communicating with their nearest neighbors. The state of a node converges to a positive value in the absence of a cut. If a node is disconnected from the source as a result of a cut, its state converges to 0 (deactive). The state of node determines whether it is connected to source or not. The nodes that are still connected to the source will be able to detect that a cut has occurred somewhere in the network. It has not only fast convergence rate but also independent of size of the network, as the delay between the occurrence of a cut and its detection by all the nodes can be made independent of the size of the network. The algorithm proposed in Distributed Cut Detection, in this paper it is not limited to linear cuts [4], but also solves the issue of redundancy and also enables every node to detect if a cut occurs. The DCD algorithm eliminates the need of routing messages to the source node as it involves only nearest neighbor communication. The DCD algorithm in this paper has been demonstrated under JAVA environment. The assumptions we have taken are that (1) The source node never fails, the sensor network is initially connected, (2) Communication between nodes is symmetric, (3) If a node fails permanently, and each of its neighbours can detect its failure within a fixed time period.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The failure of sensor nodes should not affect the overall task of the sensor network. This defines the reliability or fault tolerance issue. [9] Fault tolerance is nothing but the ability to sustain sensor network functionalities without any interruption due to sensor node failures. As this is a distributive algorithm whenever there is a cut in the network or is a set of nodes fail in the network, the energy which is used to make the failure nodes function is distributed and consumed among the finely working nodes which increases their reliability and makes the network to function properly until the failure nodes are repaired or restored. This results in effective and reliable functioning of the network. [10]Due to this quality the algorithm is robust to temporary communication failure between the node pairs.

C. DOS Detection

The DOS detection part of the algorithm is applicable to the arbitrary networks; a node only requires communicating a scalar variable to its neighboring nodes. The potential of certain nodes becomes '0' (deactive) when they are separated from the source node; result is that they stop functioning. The state of the node is computed using an iterative scheme which requires only periodic communication among the neighboring nodes. These nodes keep the state of their neighboring nodes to detect a DOS event.

D. CCOS Detection

The CCOS events are detected on the basis that the states of nodes that are connected to the source node also change after the cut. However this is not enough to detect the CCOS event. Therefore the CCOS detection proceeds by using probe messages that are initiated by certain nodes that encounter failed neighbors. These probe messages are forwarded from one node to another in such a way that if a short path exists around a 'hole' created by the node failures, the message will reach the initiating node. The nodes that detect CCOS event alert the source node about the cut.

E. Algorithm description

Here we briefly describe the proposed DCD algorithm. One of the nodes of the network is a specially designed node which is always active called as "source node". Let G = (V, E) denote the undirected sensor network that consists of all the nodes and edges of G that are active at time k, where k = 0, 1, 2 . . . is an iteration (repetitive) counter. Every node p of node set V maintains a scalar state x_p (k) that is iteratively updated. Let the nodes of the graph G execute the DCD algorithm with initial condition as $x_u(0) = 0 \ \forall p \in V$.

1. If no cut occurs or else no node fails then state of every node converges to a positive number.
2. If a cut occurs at a time T > 0 which separates the graph G into N connected components G_s,.. . . , G_N, where the component G_s (V_s,, E_s) contains the source node, then

(a) the state of every node disconnected from the source node converges to 0 (deactive) and

(b) the state of every node in V_s converges to a positive number.

Hence by monitoring the states of the nodes one can know about the status of the network connection. For effectiveness we proposed a prototype model by taking small number of nodes and their corresponding edges in the graph G. Hence the nodes can effectively detect first if there is any cut occurred and second they are still connected to source. We modified this algorithm by adding additional parameters to reduce redundant information at destination. We designed it in such a way that once the file is sent from a node, it is sent to its respective neighbors so that each and every node has the information. If there is any node failure from where information cannot be forwarded and a cut is detected, the information at the nodes is combined and then sent to the destination without the occurrence of redundancy. This approach is simulated successfully in JAVA environment and the expected results have found.

IV. IMPLEMENTATION

A. UML Diagram

Sequence Diagram

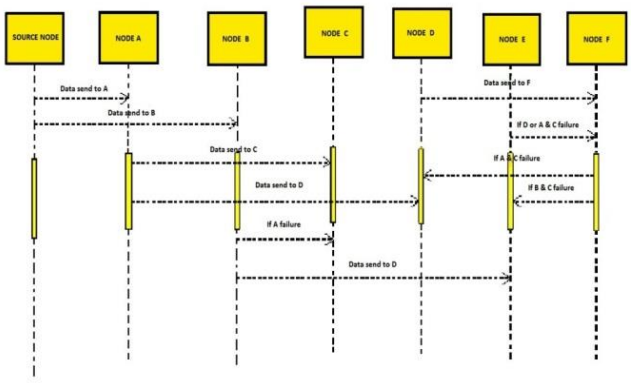

Fig 2.The above sequence diagram depicts the transmission of data among the nodes with respect to time.Messages are denoted as labeled horizontal arrows between the life-lines (Lines which denote the scope of objects (like source node))

Class Diagram

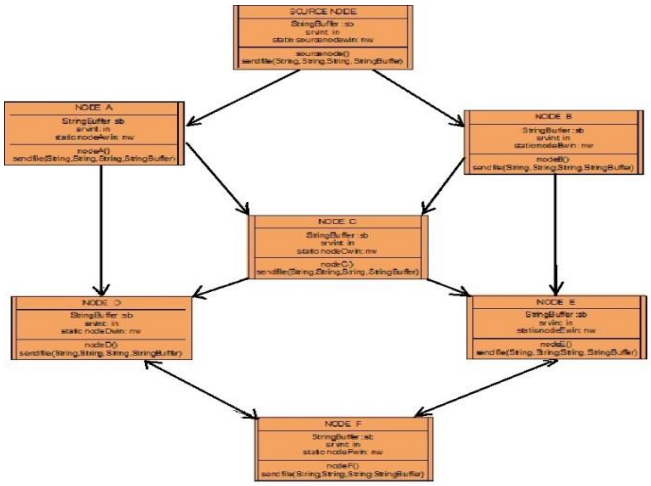

Fig 3 The above class diagram shows the existence of classes (like source node, node A….etc.) and their relationship in a logical view. Each class consists of certain variables and methods declared.

B. DCD Algorithm Implementation

Consider S=Source node; Neighbors of node S are A, B.
ack=active; dack=inactive

1. If the node A is active i.e. ack state
2. Wait for 500 ms
3. Send file to node A

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

4. Else if the node A is deactive (node failed) i.e. dack state then file sending to A failed.
5. If the node B is active i.e. ack state
6. Wait for 500 ms
7. Send file to node B.
8. Else if the node B is deactive (node failed) i.e. dack state then file sending to B failed.

Sample Code

```
String a=in.active ();
String a1=in1.active ();
String ack="Node Active";
String dack="Node Deactive";
//file sending to node A
if (a.equals (ack)){try{
for (int j=0;j<=2;j++){
Thread.sleep (500);}}
catch (Exception et){
System.out.println (et);}
in.sendfile (fn1, fs1, pat1,sb1);
sw.node1_Text.append ("\n \n File Sent to Node A ");}else
if(a.equals(dack)){
sw.node1_Text.append ("\n \n File Sending Failed to Node
A ");}
//file sending to node B
if (a1.equals(ack)){try{
for (int j=0;j<=2;j++){
Thread.sleep (500);}}
catch(Exception et){
System.out.println (et);}
in1.sendfile (fn1, fs1, pat1, sb1);
sw.node1_Text.append ("\n \n File Sent to Node B ");}else
if(a1.equals(dack)){
sw.node1_Text.append ("\n \n File Sending Failed To Node
B ");}}
```

In this way each node sends it information to its corresponding neighboring nodes. But due to this we may encounter the problem of redundancy at destination. So we solved it by using data binding methods of java like rebind () method at every node which binds the information of two nodes and transfers it without commence of the duplication.

```
try {nodeD na=new nodeD();
Naming.rebind ("nodeD",na);
}catch(Exception e){}}
```

C. Screen Shots

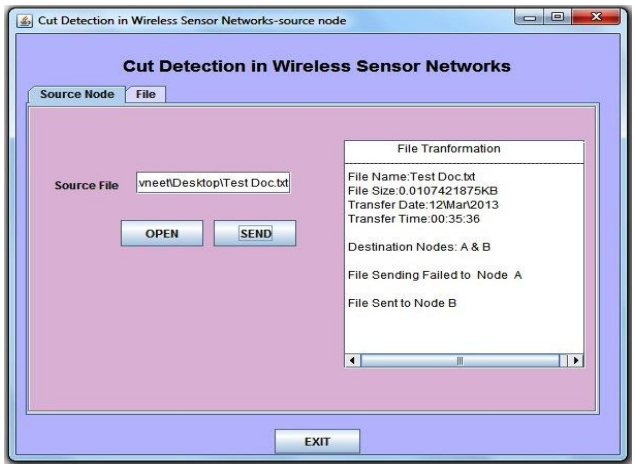

Fig 4. Screenshot representing Source Node from which file was sent.

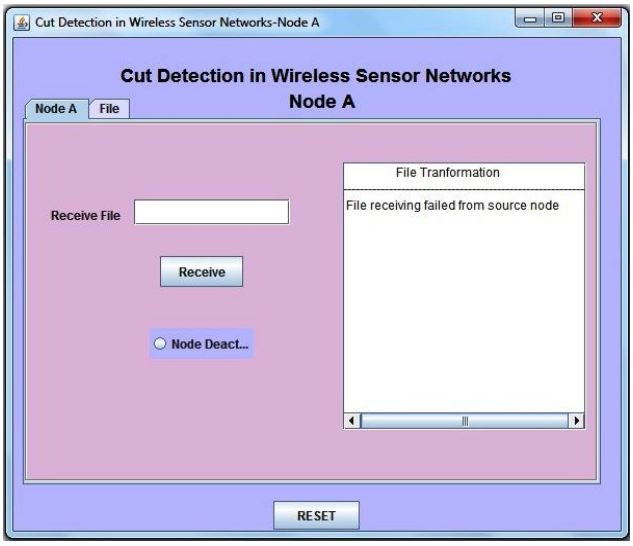

Fig 5. Screenshot representing Node A where file receiving failed.

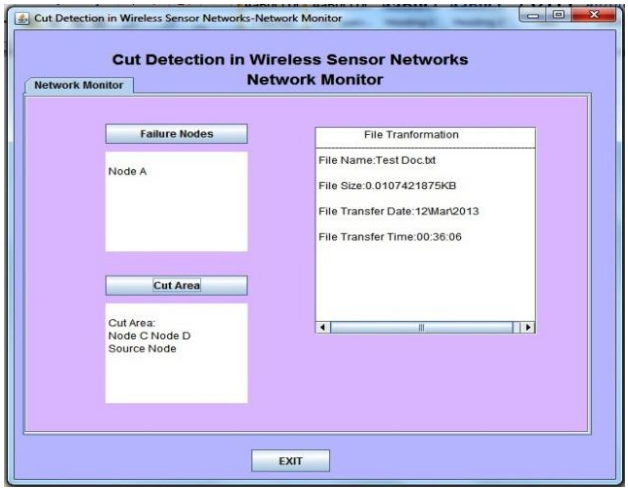

Fig 6. Screenshot representing Network Monitor which indicates failure nodes and cut area. Here Node A is failed and Cut area is Node C node D Source Node.

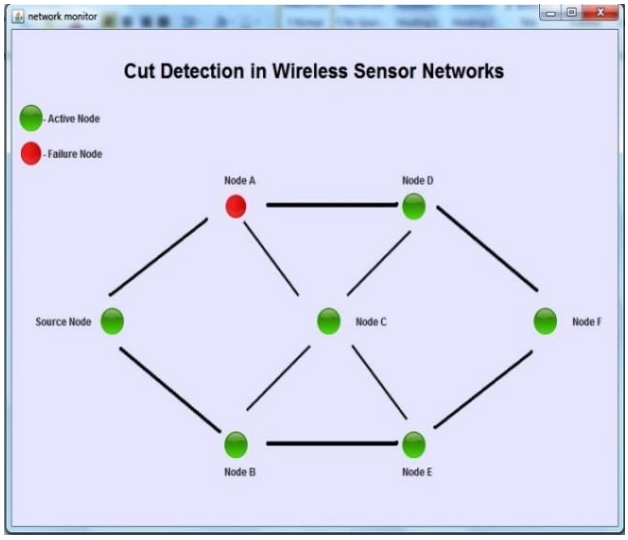

Fig 7. Screenshot representing the status of network connection which shows that Node A has been failed.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

D. Advantages of Proposed System

The DCD algorithm is distributed and asynchronous. It is robust to the temporary communication failure between the node pairs. The algorithm is iterative and has a fast convergence rate which makes it independent of size of network. Elimination of redundant information at destination node. It saves on-board energy of multiple nodes and prolongs their lives. The source node has the ability to detect the occurrence and location of a cut which will allow it to undertake network repair. The ability to detect cuts by both the disconnected nodes and the source node will lead to the increase in the operational lifetime of the network as a whole

V. CONCLUSIONS

The DCD algorithm that is proposed enables every node to detect DOS events and CCOS events and determines the approximate location of the cut. The algorithm is successfully and effectively simulated in JAVA platform. For certain scenarios the algorithm is assured to detect connection and disconnection to the source node without any error. By simulation of the proposed algorithm is effective in case of disconnecting probability, network connections and communication overhead. If a component that is disconnected due to a cut gets reconnected later the nodes can detect such reconnection from their states. Though we have got expected results we have drawback of detection of CCOS events only limited to 2D Euclidean spaces. We proposed an additional approach to the actual algorithm by adding additional parameters which solve the occurrence of duplication of information at the destination node and expected results have found through which the efficiency of the algorithm has been enhanced. Application of the DCD algorithm for node separation and reconnection to the source node in mobile networks is a ongoing research.

REFERENCES

1. Jagdish Pimple, Prof.Yogadhar Pandey **"**Cut Detection in Wireless Sensor Network using Distributed Source Separation Detection (DSSD) Approach.", International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012 1 ISSN 2250-3153.
2. "Detecting a Network Failure" by Jon Kleinberg, Internet Mathematics Vol. 1, No. 1: 37-56.
3. G. Dini, M. Pelagatti, and I.M. Savino, "An Algorithm for Reconnecting Wireless Sensor Network Partitions," Proc. European Conf. Wireless Sensor Networks, pp. 253-267, 2008..
4. N. Shrivastava, S. Suri, and C.D. To´ the, "Detecting Cuts in Sensor Networks," ACM Trans. Sensor Networks, vol. 4, no. 2, pp. 1-25, 2008.
5. H. Ritter, R. Winter, and J. Schiller, "A Partition Detection System for Mobile Ad-hoc Networks," Proc. First Ann. IEEE Comm. Soc.Conf. Sensor and Ad Hoc Comm. and Networks (IEEE SECON '04), pp. 489-497, Oct. 2004.
6. M. Hauspie, J. Carle, and D. Simplot, "Partition Detection in Mobile Ad-Hoc Networks," Proc. Second Mediterranean Workshop Ad-Hoc Networks, pp. 25-27, 2003.
7. "Cut Detection in Wireless Sensor Networks" Prabir Barooah, Member, IEEE, Harshavardhan Chenji, Student Member, IEEE, Radu Stoleru, Member, IEEE, and Tama´s Kalma´r-Nagy IEEE Transactions on Parallel and Distributed Systems, vol.23, no.3, March 2012.
8. P. Barooah, "Distributed Cut Detection in Sensor Networks, "Proc. 47th IEEE Conf. Decision and Control, pp. 1097-1102, Dec.2008.
9. "Fault Tolerant Algorithms/Protocols in Wireless Sensor Networks" Hai Liu, Amiya Nayak, Ivan Stojmenović.
10. "Towards robustness and energy efficiency of cut detection in wireless sensor networks Myounggyu Won", Stephen M. George, Radu Stoleru.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)