Network Based - Cut Detection in Wireless Sensor Networks

Fathi Abdulhakim

Master student & Computer Science and Technology & UESTC Sichuan, China

Abstract : *Wireless sensor networks is one of the most important for communication is used in many application and consist of multiple sensors ,low power and low cost make wireless sensor networks is distributed sensors to monitor at different area and applied in many application such as Temperature ,Humidity Pressure ,medical care , This Paper proposes modification algorithm to detect the cuts by the help of remaining nodes , The rule of algorithmic is distributed and not synchronized, Each node has to communicate with solely those nodes that are among its communication limit, a number of protocols have been proposed to efficiently detect network cuts, they focus solely on a cut that disconnects nodes from the base station. Here the proposed modified DCD algorithm that allows every node to detect when the connectivity to a specially designated node has been lost, and one or more nodes to detect the occurrence of the cut. The algorithm is distributed and asynchronous as it requires only neighbor nodes communication, 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.*

Keywords: *Wireless Networks, Sensor Networks, Cut, Detection, Distributed and asynchronous.*

I. Introduction

Wireless sensor networks (WSNs), In 1999 it was named as one of "21 ideas for the 21st Century", and in 2003 was presented as one of "10 new technologies that will change the world" ,the "Wireless Sensor Networks" introduces and combines the three different words, i.e., wireless, sensor and networks, Wireless sensor networks is promising technology to monitoring religion consisting of large numbers of low-cost and low-power wireless nodes. The inherent nature of WSNs such as unattended operation, battery powered nodes, and harsh environments pose major challenges. One of the challenges is to ensure that the network is connected. The connectivity of the network can easily be disrupted due to unpredictable wireless channels, early depletion of node's energy, and physical tampering by hostile users. Network disconnection, typically referred to as a *network cut*, and may cause a number of problems. For example, ill-informed decisions to route data to a node located in a disconnected segment of the network might lead to data loss, wasted power consumption, and congestion around the network cut.

In this paper we consider the problem of detecting cuts by the nodes of a wireless network. We assume that there is a specially designated node in the network, which we call the source node. The source node may be a base station that serves as an interface between the network and its users. Since a cut may or may not separate a node from the source node, we distinguish between two distinct outcomes of a cut for a particular node, when a node u is disconnected from the source, we say that a DOS (Disconnected frOm Source) event has occurred for u. When a cut occurs in the network that does not separate a node u from the source node, we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred for u. By cut detection we mean (i) detection by each node of a DOS event when it occurs, and (ii) detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. By "approximate location" of a cut we mean 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.

1.2 Problem Statement

Consider a sensor network modelled as an undirected graph G = (V , E), whose node set V represents the sensor nodes and the edge set E consists of pairs of nodes (u, v) such that nodes u and v can exchange messages between each other. Note that we assume inter-node communication is symmetric. An edge (u, v) is said to be incident on both the u and v. The nodes that share an edge with a particular node u are called the neighbors of u. A cut is the failure of a set of nodes V cut $\subset$ V such that the removal of the nodes in V cut and the edges that are incident on Vcut from G results in G being divided into multiple connected components. Recall that an undirected graph is said to be connected if there is a way to go from every node to every other node by traversing the edges, and that a component Gc of a graph G is a maximal connected sub graph of G . We are interested in devising a way to detect if a subset of the nodes has been disconnected from a distinguished node, which we call the source node, due to the occurrence of a cut. In terms of these definitions, a cut event is formally defined as the increase of the number of components of a graph due to the failure of a subset of nodes). The number of cuts associated with a cut event is the increase in the number of components after the event. The problem we seek to address is twofold. First, we want to enable every node to detect if it is disconnected from the source (i.e., if a DOS event has occurred). Second, we want to enable nodes that lie close to the cuts but are still connected to the source (i.e., those that experience CCOS events) to detect CCOS events and alert the source node

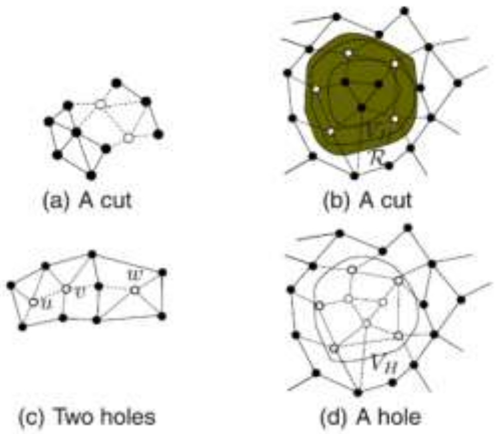

Fig. 1.1 *Examples of cuts and holes*

.

1.3. LITERATURE REVIEW

Cut detection problem was first considered in a wired network [5]. Kleinberg et al. [5] introduced the concept of (ϵ , k)-cut, which is defined as a network separation into two sets of nodes, namely (1 − ϵ)n nodes and ϵ n nodes (n refers to the total number of nodes), caused by k independently disabled edges. A set of agents, denoted by a set D, is strategically deployed in the network to detect the (ϵ , k)-cut. Each agent exchanges a control packet with other agents periodically. A cut is assumed to be present if the control message loss exceeds some threshold. The authors are interested in the size of D, and prove that the size of the set D [6], the cut present when the packet lost between agents.
Shrivastava et al. [1] recently introduced a protocol to detect a cut in wireless sensor networks. Their work is largely based on [4]. The base station detects ϵ -cuts by monitoring whether it can receive messages from the sentinels. The protocol deploys sentinels, a counterpart of agents in [4], to detect ϵ -cut, which is defined as a linear cut that separates the network into two parts, where one part has at least ϵ -fraction of total nodes. The paper aims to minimize the number of sentinels based on the assumption that in sensor networks, linear-shaped or other geometric shaped cuts are more likely to happen, rather than the cut with k independent edge failures. They prove that O(1/ϵ) sentinels are required to detect ϵ -cut with $\epsilon < 1$. The limitation of their cut detection algorithm is that they consider only the linear cuts, being unable to detect arbitrarily shaped cuts. Additionally, their algorithm is a centralized solution, requiring global topology information. As noted by Shrivastava et. al. [1], the challenges posed by the possibility of network partitioning in WSNs has been recognized in several papers (see [2], [3], [4]) but the problem of detecting when such partitioning occurs seems to have received little attention. Barooah et al. [7][8] addressed the issues that previous cut detection algorithms have. The Distributed Source Separation Detection (DSSD) algorithm is fully distributed and detects arbitrarily shaped cuts. A positive scalar value, called state, is maintained by each node. The state of each node is updated based on the states of its immediate neighbors. If a node is connected to a sink, its state converges to some positive value. Otherwise, its state converges to zero. The DSSD algorithm, however, suffers from control message overhead, since the algorithmic iterations for the convergence depends on the degree of the network. A node k is deemed locally critical node if the corresponding sub graph of k-hop neighbors of that node is disconnected. The relatively lower communication overhead of these protocols come at the cost of high rate of incorrect detection; especially false positives [11][12].

1.4. PROPOSED SOLUTION

Failure of a set of nodes will reduce the number of multi-hop paths in the network. Such failures can cause a subset of nodes that have not failed to become disconnected from the rest, resulting in a **"cut".** Two nodes are said to be disconnected if there is no path between them, When a node u is disconnected from the source, we say that a DOS (Disconnected frOm Source) event has occurred for u. When a cut occurs in the network that does not separate a node u from the source node, we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred for u. By cut detection we mean (i) detection by each node of a DOS event when it occurs, and (ii) detection of CCOS events by the nodes close to a cut, and the approximate location of the cut. In this article we propose a distributed algorithm to detect cuts, named the Distributed Cut Detection (DCD) algorithm. The algorithm allows each node to detect DOS events and a subset of nodes to detect CCOS events. The algorithm we propose is distributed and asynchronous: it involves only local communication between neighboring nodes, and is robust to temporary communication failure between node pairs The convergence rate of the computation is independent of the size and structure of the network

2.1 Module Description

2.1.1 Distributed Cut Detection

The algorithm allows each node to detect DOS events and a subset of nodes to detect CCOS events. The algorithm we propose is distributed and asynchronous: it involves only local communication between neighboring nodes, and is robust to temporary communication failure between node pairs. A key component of the DCD algorithm is a distributed iterative computational step through which the nodes compute their (fictitious) electrical potentials. The modification algorithm DCD acts when the cut occurs and separate the network into arbitrary ships, the algorithms is saving the energy of nodes and increase the networks life time The convergence rate of the computation is independent of the size and structure of the network.

2.1.2 Cut

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial and temporal resolution. In fact, node failure is expected to be quite common due to the typically limited energy budget of the nodes that are powered by small batteries. Failure of a set of nodes will reduce the number of multi-hop paths in the network; failures can cause a subset of nodes that have not failed to become disconnected from the rest of the network, resulting in a partition of the network also called a “cut”. Two nodes are said to be disconnected if there is no path between them, and as we know that sensors has disconnectivity from the network is normally referred as a partition of the network of cut in the wireless sensor network, which arise many problems like unreliability, data loss, performance degradation, because of cuts in wireless sensor network many problems may arise like a wired network means data loss problem arises, means data reach in a disconnected route.

2.1.3 Source Node

We consider the problem of detecting cuts by the nodes of a wireless network. We assume that there is a specially designated node in the network, which we call the source node. The source node may be a base station that serves as an interface between the network and its users. Since a cut may or may not separate a node from the source node, we distinguish between two distinct outcomes of a cut for a particular node.

2.1.4 CCOS AND DOS

When a node u is disconnected from the source, we say that a DOS (Disconnected frOm Source) event has occurred for u. When a cut occurs in the network that does not separate a node u from the source node, we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred for u. By cut detection we mean (i) detection by each node of a DOS event when it occurs, and

(ii) Detection of CCOS events by the nodes close to a cut, and the approximate location of the cut.

3.1 The Modified Distributed Cut Detection (DCD) Algorithm

The DCD algorithm used here is a modified version of the original algorithm proposed by Prabir the modified DCD algorithm allows every node to monitor the topology of the (initially connected) graph and detect if a cut occurs. For reasons that will be clear soon, one node of the network is denoted as the “source node”. The algorithm consists of every node updating a local state periodically by communicating with its nearest neighbors. The state of a node converges to a positive value in the absence of a cut. If a node is rendered disconnected from the source as a result of a cut, its state converges to 0. By monitoring its state, therefore, a node can determine if it has been separated from the source node. In addition, the nodes that are still connected to the source are able to detect that, one, a cut has occurred somewhere in the network, and two, they are still connected to the source node. Since the algorithm is iterative, a faster convergence rate is desirable for it to be effective. The convergence rate of the proposed algorithm is not only quite fast, but is independent of the size of the network. As a result, 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. This last feature makes the algorithm highly scalable to large sensor networks.

Two Functionalities of the DCD algorithm

1- **DOS Detection**
2- ***CCOS detection***

Part 1: DOS Detection

DOS disconnected from source it detection part consists of steady-state detection, normalized state computation, most steady state and separation detection. The assumptions are that the source node or base station never fails, the sensor network is initially connected, and the communication between the sensor nodes is bidirectional. Some sensor nodes may fail or be blocked due to lack of battery power, have physical damage or environmental interference

$$\delta x_i(k) = \begin{cases} \frac{x_i(k) - x_i(k-1)}{x_i(k-1)} & \text{if } x_i(k-1) > \epsilon_{zero} \\ \infty & \text{otherwise} \end{cases}$$

Where ϵ_{zero} is a small positive number

Part 2: *CCOS detection:*

The algorithm for detecting CCOS events relies on finding a short path around a hole, if it exists, when a node detects a large change in its local state as well

as failure of one or more of its neighbors, and both of these events occur within a (predetermined) small time interval, the algorithm use potentials of the nodes that are connected to the source node also change after the cut, so the failure of nodes cause a cut and leads to changes in the potential of neighbors the node initiates a PROBE message.
The pseudo-code for the algorithm that decides when to initiate a probe is described in Algorithm (PROBE INITIATION).
Each PROBE message p contains the following information:

- unique probe ID
- destination node ID
- path traversed
- Time stamp

3.2Algorithm: PROBE INITIATION

- Executed by node i at every k to decide if new probes are to be initiated.
- **Input:** The parameter $r^{\Delta ss}$, most recent unsteady interval, list of neighbors that failed in that interval.
- **Output:** new probe to be initiated, if any.

If the vector X_i^{ss} (k) = [x1, . . . , *xl*] *has more than one entry.*

Then,

If each of the following is true:

(i) $\delta x_i(k) > \varepsilon_{\Delta x}$

(ii) At least one neighbor failed during the last unsteady interval

(iii) xl-xl-1/xl> $r^{\Delta ss}$

Then,

Initiate a new probe with IDp, and add p to the list of probes to be processed

End

End

4.1 Simulation Results

The Simulations are applied using the following networks,
In software the algorithm was implemented using the java language running on windows xp operating system. The system executes in two phases: the Reliable Neighbor Discovery (RND) phase and the DCD Algorithm phase In the RND phase each node is connected to the source node. Upon receiving the message, the mote updates the number of beacons received from that particular sender. To determine whether a communication link is established, each mote first computes for each of its neighbors the Packet Reception Ratio (PRR), defined as the ratio of the number of successfully received beacons and the total number of beacons sent by a neighbor, the figure 4.3 shows the cut occurrence in node F .

1- When the source node sends file to the neighbors nodes A,B , C the node A ,B and C receive the file and send it to the neighbors nodes ,D,E
2- The node C receive the file from Source node,
3- Node D receives file from node A or from C if the node A is failed
4- Node E receive file from node B or C if the node B is failed
5- Node F receives file from node D or E if the node D is failed
6- If the node E ,C, and D is failed, then they will form a cut

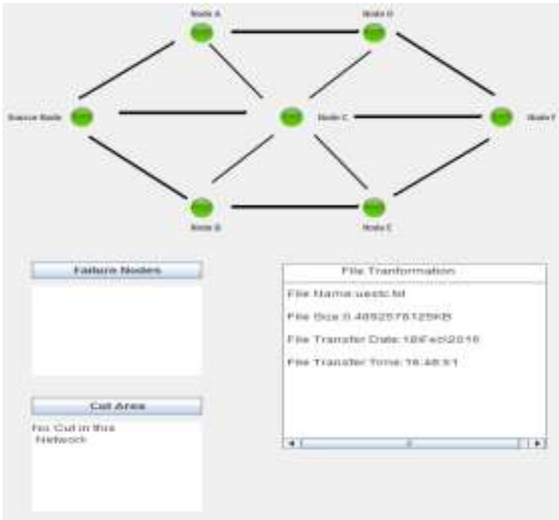

Figure 4.1 Network before failure nodes

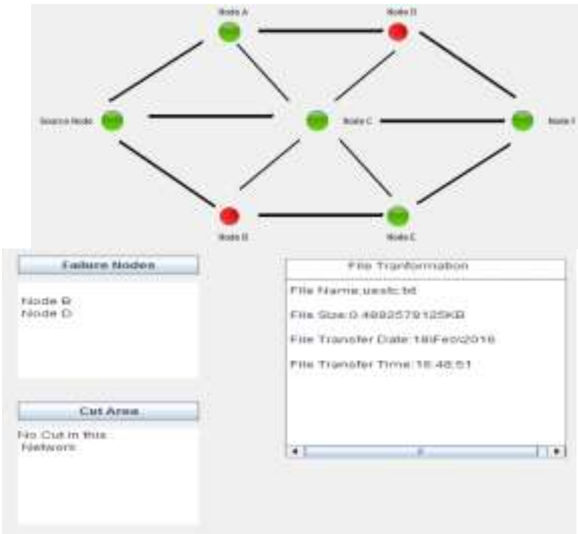

Figure 4.2 Network after failure nodes

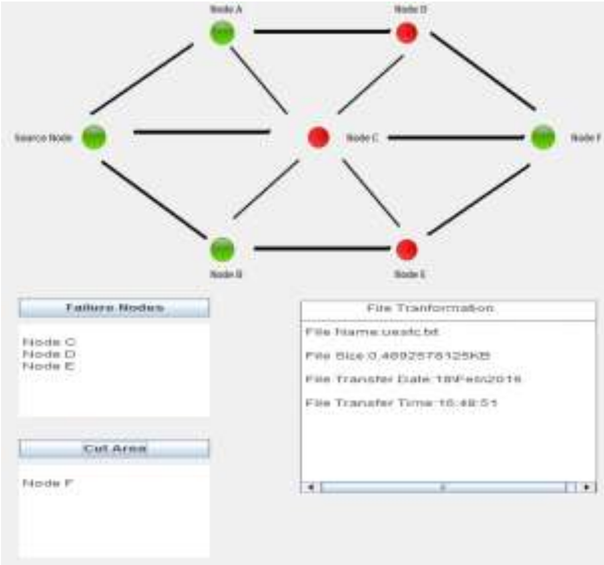

Figure 4.3 Network after failure nodes and appears cut on node F

5. Performance measures

There are two important metrics of performance for the DCD algorithm are (1) detection accuracy, and (2) detection delay.

1) Detection accuracy refers to the capable to detect a cut when it occurs and not declaring a cut when none has occurred.
2) Detection delay for DOS a node i that has undergone a DOS event is the minimum number of iterations (after the node has been disconnected) it takes before the node switches its flag from DOSi = 0 toDOSi = 1.

CCOS detection delay is the minimum number of iterations it takes after the occurrence of a cut before a node detects it.

In detecting (DOS) events, two kinds of inaccuracies are possible.

1) DOS 0/1
2) DOS 1/0

❖ **A DOS0/1 error** is said to occur if a node concludes it is connected to the source while it is in fact disconnected, i.e., node i declares DOSi to be 0 while it should be 1.

❖ **A DOS1/0 error** is said to occur if a node concludes that is disconnected from the source while in fact it is connected.

In CCOS detection, again two kinds of inaccuracies are possible:

1) CCOS 0/1
2) CCOS 1/0

❖ **A CCOS 0/1 error** is said to occur when cut (or a large hole) has occurred but not a single node is able to detect it.

❖ **A CCOS 1/0 error** is said to occur when a node concludes that there has been a cut (or large hole) at a particular location while no cut has taken place near that location.

CONCLUSIONS:

The DCD algorithm proposed here enables every node of a wireless sensor network to detect DOS (Disconnected from Source) events if they occur so that the node can go into a sleepy state and based on that saving the battery life until the network get repaired then the node goes into an active state. Second, it enables a subset of nodes that experience CCOS (Connected, but Cut Occurred Somewhere) events to detect them and estimate the approximate location of the cut in the form of a list of active nodes that lie at the boundary of the cut/hole.

FUTURE WORK

A topic for future research is how to enable secure cut detection, for the case when some of the nodes may “fail” in a malicious mode, such as when nodes are hacked by an adversary to send incorrect state data.

ACKNOWLEDGMENT

I am greatly thankful to my advisor, I would like to thank my colleague for giving me important suggestions and feedback on my work, Special thanks to my family whose unending love and support sustained and encouraged me throughout my graduate studies.

REFERENCES

[1] N. Shrivastava, S. Suri, and C. D. T´oth, “Detecting cuts in sensor networks,” in *IPSN ’05: Proceedings of the 4th international symposium on Information processing in sensor networks*, 2005, pp. 210–217.

[2] A. Cerpa and D. Estrin, “ASCENT: Adaptive Self-Configuring sensor Networks Topologies,” in *IEEE Infocom*. New York, NY: IEEE, June 2002.

[3] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *SenSys03*, Los Angeles, California, USA, November 57 2003.

[4] X. J. Du, M. Zhang, K. E. Nygard, S. Guizani, and H.-H. Chen, “Self healing sensor networks with distributed decision making,” *International Journal of Sensor Networks (IJSNET)*, vol. 2, no. 5/6, 2007.

[5] J. Kleinberg, “Detecting a network failure,” *Internet Mathematics*, vol. 1, pp. 37–56, 2003.

[6] J. Kleinberg, M. Sandler, and A. Slivkins, “Network failure detection and graph connectivity,” in *the 15th ACM-SIAM Symposium on Discrete Algorithms*, 2004.

[7] P. Barooah, H. Chenji, R. Stoleru, and T. Kalm´ar-Nagy, “Detecting separation in robotic sensor networks,” 2008, submitted to the IEEE wireless communications magazine.

[8] H. Chenji, P. Barooah, R. Stoleru, and T. Kalm´ar-Nagy, “Demo abstract: Distributed cut detection in sensor networks,” in *6th ACM Conference on Embedded Networked Sensor Systems (SenSys’08)*, November 2008.

[9] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, "Distributed recovery of actor failures in wireless sensor and actor networks," in *WCNC*, 2008, pp. 2480–2485.
[10] X. Liu, L. Xiao, A. Kreling, and Y. Liu, "Optimizing overlay topology by reducing cut vertices," in *Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video*, ser. NOSSDAV '06. New York, NY, USA: ACM, 2006, pp.17:1–17:6.
[11] M. Jorgic, N. Goel, K. Kalaichevan, A. Nayak, and I. Stojmenovic, "Localized detection of k-connectivity in wireless ad hoc, actuator and sensor networks," in *Proceedings of 16th International Conference on Computer Communications and Network (ICCCN)*, 2007,
[12] M. Jorgic, I. Stojmenovic, M. Hauspie, and D. Simplot-ryl, "Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks," in *in proceedings of the 3rd Annual IFIP Mediterranean Ad Hoc Networking Workshop*, 2004.