RESEARCH ARTICLE OPEN ACCESS

Cut Detection in Wireless Sensor Network

Vignesh Menon, Pratiksha Mhase, Shwetali Kanekar
Prof: Nilima Nikam
Department of Computer Science and Engineering
Yadavrao Tasgaonkar Institute of Engineering & Technology
Bhivpuri Road
Karjat, Raigad,
Maharashtra – India

ABSTRACT

A wireless sensor network can get separated into multiple connected components due to the failure of some of its nodes, which is called a "cut". Here consider the problem of detecting cuts by the remaining nodes of a wireless sensor network. Here 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 is distributed and asynchronous: every node needs to communicate with only those nodes that are within its communication range. The algorithm is based on the iterative computation of a fictitious "electrical potential" of the nodes. The convergence rate of the underlying iterative scheme is independent of the size and structure of the network. All cut nodes information is been updated and created into a log file of system and filtered log records of cut nodes are maintained.

Keywords:- WSNs

I. INTRODUCTION

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. 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. Wireless sensor networks (WSNs), consisting of large numbers of low-cost and low-power wireless nodes, have recently been employed in many applications: disaster response, military surveillance, and medical care among others. 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 as a *network cut*, may cause a number of problems.

II. LITERATURE SURVEY

In paper [1], Nisheeth Shrivastava propose a low overhead scheme for detecting a network partition or cut in a sensor network. Consider a set S of n sensors, which are modeled as points in the two-dimensional plane. An adversary can make a linear cut through the sensor network, disabling all the sensors on one side of the line; the base station is assumed to lie on the other (safe) side. Formally, given a line L, let

$L^-$and L^+ denote the two half-planes defined by L, and let $L^-(S)$ and $L^+(S)$ denote the subset of sensors that lie in these half-planes. Alternatively, the adversary can disrupt the communication so that sensors on one side of the line cannot communicate with sensors on the other side, including the base station. Here it is call a linear cut an *ε-cut* if at least *ε* fraction of the sensors are cut off, where $0 < \varepsilon < 1$ is a user-specified parameter. The Distributed Source Separation Detection (DSSD) algorithm proposed in paper [2] is not limited to ε-linear cuts; it can detect cuts that separate the network into multiple components of arbitrary shapes. Furthermore, the DSSD algorithms not restricted to networks deployed in 2D, it does not require deploying sentinel nodes, and it allows every node to detect if a cut occurs. The DSSD algorithm involves only nearest neighbour communication, which eliminates the need of routing messages to the source node. This feature makes the algorithm applicable to mobile nodes as well. The cut detection problem was first considered in a wired network [3]. Kleinberg et al. [3] introduce the concept of (ε, k) -cut, which is defined as a network separation into two sets of nodes, namely $(1 - \varepsilon)n$ nodes and εn 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. Ritter et al. [4] proposed a cut detection algorithm where a sink node broadcasts an *alive message*. A cut is detected by *border* nodes, which are located on the border of network, if these nodes fail to receive the alive message more than a certain number of times. In paper [5], Myounggyu Won propose solutions for a more general cut detection problem – the *destination-based cut detection* problem. Unlike the traditional cut detection problem, Here an attempt to find a network cut between a sender and any node in a set of given destinations. Point-to-Point Cut Detection protocol (P2P-CD) is used. P2P-CD allows source node to identify a cut with respect to any destination node. In this protocol, the boundary of a cut is compactly represented as a set of linear segments. The compact representation of a cut allows the information on existing cuts to be efficiently distributed throughout the network with small overheads. A source node, using the distributed information, locally determines whether any potential destination is reachable.

III. PROPOSED METHODOLOGY

In our project ,we have two sections :

1.Network node controller

2.Network nodes

Firstly we will enter the nodes and make it as active and inactive according to which the path will be decided from the source to destination.All the node details will be updated In network node controller frame. Data will be sent from source to destination for this random path will be selected if there is inactive node in the random path then the cut will be detected and this will be shown in the form of graph.

IV. METHODOLOGY

Past decade has seen a surge of research activities in the field of wireless Communication. Emerging from this research thrust are new points of view on how to communicate effectively over wireless channels. Here our complete mechanism is divided into the major domain.

1. Wireless Transmission Channel
2. Routing Algorithm
3. Cut Detection

4.1 Route Discovery: The selection of path for data transmission is done based on the availability of the nodes in the region using the ad-hoc on demand distance vector routing algorithm. By using the Ad hoc On demand Distance Vector (AODV) routing protocol, the routes are created on demand, i.e. only when a route is needed for which there is no "fresh" record in the routing table. In order to facilitate determination of the freshness of routing information, AODV maintains the time since when an entry has been last utilized. A routing table entry is "expired" after a certain predetermined threshold of time. Consider all the nodes to be in the position. Now the shortest path is to be determined by implementing the Ad hoc on Demand Distance Vector routing protocol in the wireless simulation environment for periodically sending the messages to the neighbors and the shortest path.

4.2 Route Maintenance: The next step is the maintenance of these routes which is equally important. The source has to continuously monitor the position of the nodes to make sure the data is being carried through the path to the destination without loss. In any case, if the position of the node change and the source doesn't make a note of it then the packets will be lost and eventually have to be resent.

4.3Data Transmission: The path selection maintenance and data transmission are consecutive process which happen in split seconds in real-time transmission. Hence the paths allocated priory is used for data transmission. The first path allocated previously is now used for data transmission. The data is transferred through the highlighted path. The second path selected is now used for data transmission. The data is transferred through the highlighted path. The third path selected is used for data transmission. The data is transferred through the highlighted path. 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 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. 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 All cut nodes information is been updated and created into a log file of system and filtered log record of cut nodes are maintained .

4.4 DOS Detection:

The approach here is to exploit the fact that if the state is close to 0 then the node is disconnected from the source, otherwise not. In order to reduce sensitivity of the algorithm to variations in network size and structure, we use a normalized state. DOS detection part consists of steady-state detection, normalized state computation,and connection/separation detection. Every node i maintains a binary variable $DOS_i(k)$, which is set to 1 if the node believes it is disconnected from the source and 0 otherwise. This variable, which is called the DOS status, is initialized to 1 since there is no reason to believe a node is connected to the source initially.

Each node i computes the normalized state difference $\delta x_i(k)$ as follows:

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

where εzero is a small positive number.

Each node computes a normalized state $x_i^{norm}(k)$ as

$$x_i^{\text{norm}}(k) := \begin{cases} \dfrac{x_i(k)}{\hat{x}_i^{ss}(k)}, & \text{if } \hat{x}_i^{ss}(k) > 0, \\ \infty, & \text{otherwise}, \end{cases}$$

where $x_i^{ss}(k)$ is the last steady state seen by i at k, i.e., the last entry of the vector $X_i^{ss}(k)$. If the normalized state of i is less than $^{\varepsilon}$DOS, where $^{\varepsilon}$DOS is a small positive number, then the node declares a cut has taken place: DOS*i* = 1. If the normalized state is ∞, meaning no steady state was seen until k, then DOS*i*(k) is set to 0 if the state is positive (i.e., $x_i(k) >$ εzero) and 1 otherwise.

4.5 CCOS Detection:

The algorithm for detecting CCOS events relies on finding a short path around a hole, if it exists, and is partially inspired by the jamming detection algorithm proposed in [9]. The method utilizes node states to assign the task of hole-detection to the most appropriate nodes. 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 node initiates a PROBE message.

Each PROBE message p contains the following information:

1 A unique probe ID

2. Probe centroid Cp

3. Destination node

4. Path traversal (in chronological order)

5. The angle traversed by the probe around the centroid

V. PERFORMANCE EVALUATION

Two important metrics of performance for the DCD algorithm are 1) detection accuracy, and 2) detection delay. Detection accuracy refers to the ability to detect a cut when it occurs and not declaring a cut when none has occurred. DOS detection delay for 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 DOS_i flag from 0 to 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 disconnection from source (DOS) events, two kinds of inaccuracies are possible. 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. A CCOS0/1 error is said to occur when cut (or a large hole) has occurred but not a single node is able to detect it. A CCOS1/0 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.

5.1 Communication Cost:

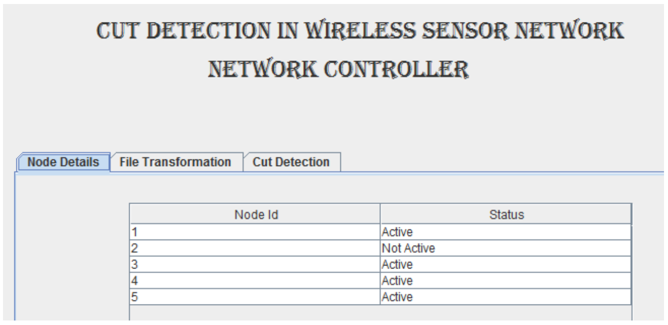
A reasonable metric of communication cost is the number of packets transmitted per node per unit of time. The epoch between two successive iterations k and k + 1 is taken as one unit of time. It is important to recognize that a single message (some quanta of information) may require multiple packet transmissions.

Next step is comparing the communication cost of the DCD algorithm with that of the bidirectional flooding scheme. Let

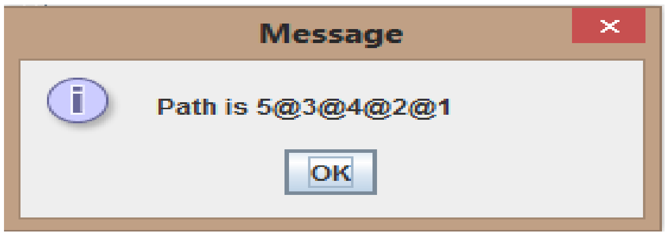
T be the *flooding time-period* for both nodes-to-source flooding and source-to-nodes flooding. For the nodes-to-source flooding, this means at intervals of T time units, every node generates a new time-stamped message intended for the source that contains its ID. These messages will be routed to the source by a multihop routing protocol. Assuming that a node aggregates the messages received from its neighbors before forwarding it to the next node in the routing tree, the number of messages to be transmitted in every T time units is n−1. In the source-to-nodes flooding, the source node generates a new time-stamped message that is intended for all the nodes in the network every T time units. A node i that receives this packet has to forward it to its active neighbors,

VI. RESULTS

We have successfully completed the project. By assuming five nodes in which source nod is 5 and destination node is 1. As node2 is inactive so the cut is detected at node 2as shown in below fig.


CUT DETECTION IN WIRELESS SENSOR NETWORK
NETWORK CONTROLLER

Node Id	Status
1	Active
2	Not Active
3	Active
4	Active
5	Active

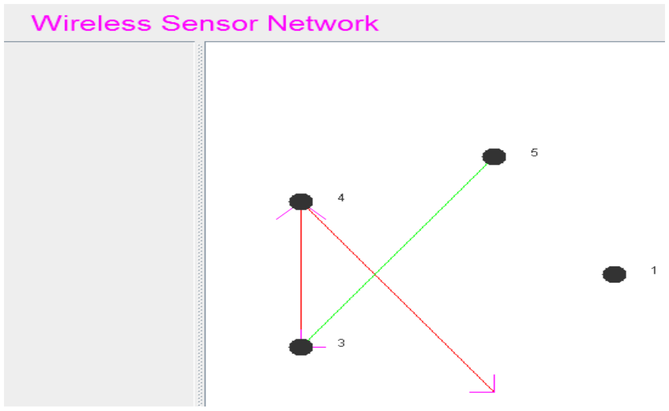

Message

Path is 5@3@4@2@1

which it can do efficiently by broadcasting. If ηd is the number of nodes that are at a graphical distance of d from the source node, then the number of copies of this message broadcasted during a flooding time period is 1+η1 +η2 + · · · +ηT−1 = |VT−1|, where VT−1 is the set of nodes in G that are within a distance of T − 1 hops from the base station. The number of messages broadcasted per node per unit time is therefore:

$$\frac{1}{nT}(n + |\mathcal{V}_{T-1}|) = \frac{1}{T} + \frac{1}{T}\frac{|\mathcal{V}_{T-1}|}{n}.$$

In a 2D-grid, |VT−1| = O(T 2), so that the communication cost is 1/T + O(T/n).



VII. CONCLUSION

In this report, a distributed algorithm is proposed 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 propose here is distributed and asynchronous: it involves only local communication between neighboring nodes, and it can be 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 electricapotentials. The convergence rate of the computation is independent of the size and structure of the network. The DCD algorithm is used for every node of a wireless sensor network to detect Disconnected from Source events if they occur. Second, it is also used for a subset of nodes that experience CCOS 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. The DOS and CCOS events are defined with respect to a specially designated source node. The algorithm is based on ideas from electrical network theory and parallel iterative solution of linear equations. All cut nodes information is been updated and created into a log file of system and filtered log records of cut nodes are maintained

REFERENCE

[1] N. Shrivastava, S. Suri, and C. Toth, "Detecting cuts in sensor networks, " ACM Transactions on Sensor Networks, vol. 4, no. 2, pp. 1–25, 2008.

[2] J. Kleinberg, "Detecting a network failure," Proceedings of the 41st Annual Symposium on Foundations of Computer Science ,p. 231, 2000.

[3] J. Kleinberg, M. Sandler, and A. Slivkins, "Network failure detection and graph connectivity," in Proc. of ACM SODA , 2004.

[4] H. Ritter, R. Winter, and J. Schiller, partition detection system for mobile ad-hoc networks," in Proc. of IEEE SECON, 2004.

[5] 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

[6] A.D. Wood, J.A. Stankovic, and S.H. Son, "Jam: A Jammed-Area Mapping Service for Sensor Networks," Proc. IEEE Real Time Systems Symp., 2003.

[7] M. Won, M. George, and R. Stoleru, "Towards robustness and Energy efficiency of cut detection in wireless sensor networks,"

[8] 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.

[9] P. Barooah, "Distributed Cut Detection in Sensor Networks," Proc. 47th IEEE Conf. Decision and Control, pp. 1097-1102, Dec. 2008.