IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, p-ISSN: 2278-8719
Vol. 3, Issue 9 (September. 2013), ||V2|| PP 13-16

Censored Discovery in Wireless Device Nets

G. Veeranagamma, J. Balanjaneyulu
1. M.Tech student, 2.Assistant Professor
Global College of Engineering & Technology, Kadapa

Abstract: - In this paper; we consider the problem of detecting cuts by the remaining nodes of a wireless sensor network. Network partitioning is a form of network failure. A single connected network topology breaks apart into two or more network topologies separated from each other. An algorithm which enables each node in the network to detect whether a cut has occurred anywhere in the network is demonstrated. The algorithm is based on the iterative computation of a fictitious "electrical potential" of the nodes. The algorithm specifies that every node to detect when the connectivity to a specially designated node has been lost.

Index Terms: - *Sensor networks, partition detection, Distributed Cut Detection, Wireless Sensor Networks.*

I. INTRODUCTION

Wireless sensor networks (WSN) have emerged as an important new technology for instrumenting and observing the physical world. In this paper, we address a different kind of challenge for sensor networks, which does not seem to have received adequate attention. Wireless sensor network can get separated into multiple connected components due to the failure of some of its nodes, which is called a "cut". Existing cut detection system deployed only for wired networks. In fixed infrastructure based networks, the occurrence of network partition is highly unlikely and only possible if big parts of the infrastructure fail simultaneously. How to monitor the sensor network itself and how to detect when the network has suffered a significant "cut". In this paper we compare two approaches to detect network partitioning, a centralized and a distributed approach. Both approaches have unique advantages which are compared and analyzed. On detection of network partitioning, the nodes could extend their transmission range to reconnect the separate network topologies. ". Two nodes are said to be disconnected if there is no path between them. We consider the problem of detecting cuts by the nodes of a wireless network.

ADVNATAGES

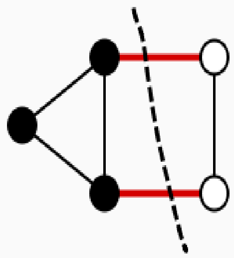
1. Comes with provable characterization on the DOS detection accuracy
2. CCOS events detection can be identified
3. DCD algorithm enables base station and also every node to detect if it is disconnected from the base station.

Network cut's in sensor networks:

The 'cut' is the process of dividing the network into subset of individual networks. Generally there are two types of cut's introduced in sensor networks. They are

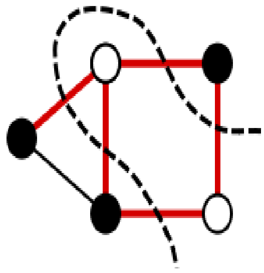
Minimum cut:

A cut is minimum if the size of the cut is not larger than the size of any other cut. The illustration on the below shows a minimum cut: the size of this cut is 2, and there is no cut of size 1 because the graph is bridgeless.



Maximum cut:

A cut is maximum if the size of the cut is not smaller than the size of any other cut. The illustration on the right shows a maximum cut: the size of the cut is equal to 5, and there is no cut of size |E| because the graph is not bipartite (there is an odd cycle). In general, finding a maximum cut is computationally hard. The max-cut problem is one of Karp's 21 NP-complete problems. The max cut problem is also APX-hard, meaning that there is no polynomial-time approximation scheme for it unless P = NP.



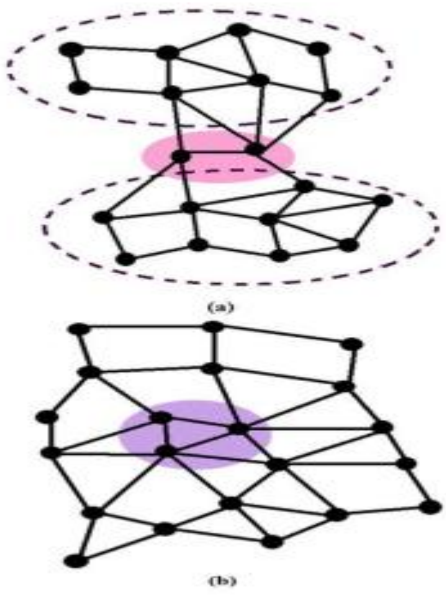
The fallowing are the mislanious cut's in the sensor networks.
We focus on a particular kind of failure:
network partition by a linear cut.
e-cut: network partition where e-fraction of sensors cut off from the base station.
Linear e-cut: partition defined by a line.Two equivalent views: physical disabling of sensors, or communication disruption along the cut. The fallowing is algorithm for detecting cuts in sensor networks.

Algorithm 2 Cut Detection

Input: $p_s, p_t, \mathbb{P}$ and UPDATE_INDEX
upon receiving the broadcast packet:
if update type **then**
    $\mathbb{P} \leftarrow \mathbb{P} \setminus \{P_i\}, \forall i \in$ UPDATE_INDEX
    $\mathbb{P} \leftarrow \mathbb{P} \cup P.$
else
    $\mathbb{P} \leftarrow \mathbb{P} \cup P.$
end if
upon having a packet to destination at p_t:
if $\forall P \in \mathbb{P}, (PIP(P, p_s) \cdot PIP(P, p_t) > 0)$ **then**
    t is reachable.
else
    t is not reachable.
end if

II. DISTRIBUTED CUT DETECTION

Loss of connectivity in deployed wireless sensor networks can be quite disastrous for the network. A "cut" (which separates the network into two or more components incapable of communicating with each other) is usually hard to detect. An algorithm which enables each node in the network to detect whether a cut has occurred anywhere in the network is demonstrated. We propose a distributed algorithm to detect cuts in sensor networks, i.e., the failure of a set of nodes that separates the networks into two or more components. The algorithm consists of a simple iterative scheme in which every node updates a scalar state by communicating with its nearest neighbors. Although the algorithm is iterative and involves only local communication, its convergence rate is quite fast and is independent of the size of the network. The below is examples of cut's and holes in distributed detection algorithm.



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. A node keeps track of the positive steady states seen in the past using the following method. Each node I computes the normalized state difference ∂xi(k) as follows:

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

Every node i also keeps a list of steady states seen in the past, one value for each unpunctuated interval of time during which the state was detected to be steady. Each node computes a normalized state xNorm i (k) as:

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

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
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.

Related Work:

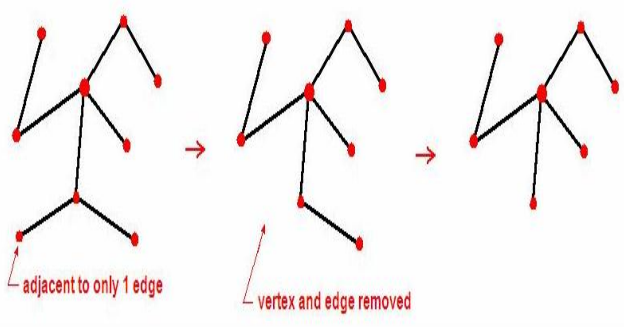
The sentinel model is inspired by work of Kleinberg. His setting is a wired network: detect an e-cut that results from cutting off at most k edges in the graph. Designate O(poly(k), 1/e) nodes as sentinels, who engage in pairwise communication. The partitioning problem is handled by a simple *PING/ACK* mechanism. A node sends a *PING* message to another node. If it does not receive an *ACK* in a certain amount of time, that node is added to a list of suspects.

Geometric Cuts:

- Spatially correlated cuts are more natural in sensor networks.
- Geometric cut complexity (linear, circular etc.) more meaningful than edge failures.
- A major drawback of Kleinberg scheme is the presence of False Positive.
- With high probability, it catches all e-cuts, but many of the reported cuts can be quite small (False Alarms).
- In remotely deployed sensor networks, checking a false alarm is expensive.

Geometry of Network Cuts:

- Think of sensors as points in the plane.
- A linear cut is a line that partitions the point set.
- The point-line duality: point (a,b) ⇔ line (y = ax - b).

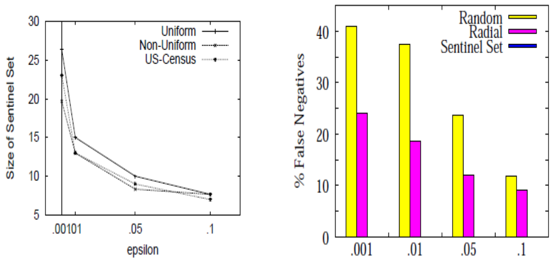


III. PERFORMANCE AND EXPERIMENTAL EVALUTION

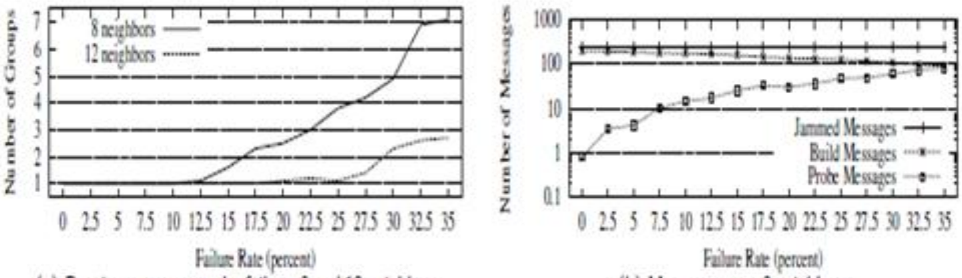
The geometric distribution of sensors is likely to vary widely from application to application. We, therefore, generated several random and non-random distributions of points in the plane to model a variety of sensor networks. We used three main data sets in our simulation:

(1) Uniform, (2) non-uniform, and (3) US census data.

In this experiment, we evaluated the behavior of our scheme with different values of the cut threshold ″. These experiments were performed with networks of a fixed size $n = 5000$.



The below diagram show how the failure performance of affected node will detected in wireless sensor networks.



(a) Dominant groups under failure, 8 and 12 neighbors (b) Messages sent, 8 neighbors

IV. CONCLUSION

Among the security problems that wireless sensor networks face is the prospect of relatively simple denial-of- service. The DCD algorithm we propose here enables every node of a wireless sensor network to detect Disconnected from Source events if they occur. In this paper we have evaluated two different approaches to detect network partitioning. Both approaches are based on the notion of border nodes and their successful identification. Both our approaches have unique advantages. Multiple partnerships make sure that a single or more failing nodes only reduce the monitored area of the affected nodes temporarily. The centralized approach generates a by far lower message overhead compared to the distributed approach. The DCD algorithm we propose here enables every node of a wireless sensor network to detect Disconnected from Source events if they occur. A key strength of the DCD algorithm is that the convergence rate of the underlying iterative scheme is quite fast and independent of the size and structure of the network, which makes detection using this algorithm quite fast.

REFERENCES

[1] 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.

[2] N. Shrivastava, S. Suri, and C.D. To´ Th, “Detecting Cuts in Sensor Networks,” ACM Trans. Sensor Networks, vol. 4, no. 2, pp. 1-25, 2008.

[3] 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.

[4] 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.

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