International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2510-2513 ISSN: 2249-6645

A Distributed Cut Detection Method for Wireless Sensor Networks

Meer Khursheeda Sulthana[1], Syed Sadat Ali[2]

[1]M. Tech, Nimra College of Engineering & Technology, Vijayawada, A.P., India.
[2]Assoc. Professor & Head, Dept.of CSE, Nimra College of Engineering & Technology, Vijayawada, A.P., India.

Abstract: *Wireless sensor networks (WSNs) consisting of large numbers of low- power and low- cost wireless nodes, have recently been employed in many applications: military surveillance, disaster response and medical care among others. The inherent nature of sensor networks such as unattended operation, battery-powered nodes, and harsh environments pose major challenges. One of the challenges is to ensure that the network is always connected. The connectivity of the sensor 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. For example, ill-informed decisions to route the 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. This paper presents a distributed cut detection method for WSNs.*

Keywords: CCOS, Cut, DOS, WSN.

I. Introduction

Wireless sensor network (WSNs) (Figure 1) is composed of a powerful base station and a set of low-end sensor nodes. Base station and the sensor nodes have wireless capabilities and communicate through a wireless, multi-hop, ad-hoc network[1]. WSNs have emerged as an important new technology for purpose of instrumenting and observing the physical world. Sensor networks are a capable scenario for sensing large areas at high spatial and positive resolution. However, the small size and low cost of the processing machines that makes them attractive for large deployment also causes the loss of low operational reliability[2]. The basic building block of these networks is a small microprocessor integrated with one or more MEMS (micro-electromechanical system) sensors, actuators, and a wireless transceiver.[3]

A WSN is usually collection of hundreds or thousands of different sensor nodes. These sensor nodes are often densely deployed in the sensor field and have the ability to gather data and route data back to a base station (BS). A sensor has 4 basic parts: a processing unit, a sensing unit, a transceiver unit, and a power unit [4]. Most of the WSNs routing techniques and sensing tasks require knowledge of location, which is provided by a location finding system. WSNs contains large number of nodes and each node may be very close to each neighbor. Since WSN should use multi- hop techniques because it consume less power than single hop techniques.

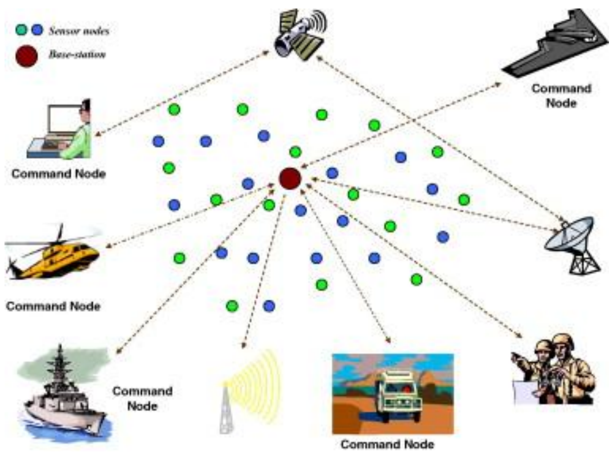

Figure 1 : Wireless sensor network example

II. Cuts in WSNs

One of the unique challenges in wireless networking environments is the phenomenon of network partitioning, which is the breakdown of a connected network topology into two or more separate, disconnected topologies[1]. Similarly sensor nodes become fail for several reasons and the network may breaks into two or more divided partitions so can say that when a number of sensor fails so the topology changes. A node may fail due to a variety of conditions such as electrical or mechanical problems, environmental degradation, and battery reduction. In fact, node failure is expected to be quite common anomaly due to the typically limited energy storage of the nodes that are powered by small batteries. Failure of a set of sensor nodes will reduce the number of multi-hop paths in the network. Such failures can cause a subset of sensor nodes –

www.ijmer.com Vol. 3, Issue. 4, Jul - Aug. 2013 pp-2510-2513 ISSN: 2249-6645

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 sensor nodes said to be disconnected if there is no path between them [2].And as we know that sensor nodes 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 data loss, unreliability , performance degradation. Because of cuts in WSNs many problems may arise like a wired network means data loss problem arises, means data reach in a disconnected route. Due to cuts, if any sensor node breaks down then the network is separated into different parts so the topology of the network changes but still network works. But because partition affects reliability, QOS of the network, data loss, efficiency, data processing speed. Because if any data passes unfortunately in the wrong route so data loss occurs this also shows unreliability of the network.

III. RELATED WORK

In [5], the authors developed a partitionable group communication service which allows so called "partition-aware applications" to operate in separated network topologies and, after two or more partitions merge, reconfigure themselves. The partitioning problem is handled by a simple PING or ACK mechanism. A sensor node sends a PING message to another node. If it does not receive an ACK in a certain period of time, that node is added to a list of suspects. A dynamic timeout mechanism is helpful which leads to a reasonably accurate suspect list. This scheme lacks the ability to distinguish between the node failure and partitioning which for most applications is desirable. In [3], the authors challenges posed by the possibility of network partitioning in WSNs has been recognized in several papers but the problem of detecting when such partitioning occurs seems to have received a little attention. To the best of our knowledge, the work in [3] is the only one that addresses the problem of detecting cuts in WSNs. They developed an algorithm for detecting the linear cuts, which is a linear separation of sensor nodes from the base station. The reason for the restriction to the linear cuts is that their algorithm relies critically on a certain duality between straight line segments and points in 2D, which also restricts the algorithm in [3] to sensor networks deployed in the 2D plane.

In contrast to the algorithm designed in [3], the DSSD algorithm proposed in [6] is not limited to linear cuts; it can detect cuts that separate the network into multiple components of arbitrary shapes. Furthermore, the DSSD algorithm is not restricted to sensor 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 the nearest neighbor communication, which eliminates the need of routing messages to the source node. This feature makes the algorithm applicable to the mobile nodes as well. Since the computation that a node has to carry out involves only averaging, it is particularly well suited to wireless sensor networks with nodes that have limited computational capability. In this paper, the proposed algorithm is an extension of the previous work [6], which partially examined the DOS detection problem.

IV. PROPOSED WORK

A. *Problem Statement*

Consider a WSN as a time-varying graph G(k) = (V(k), E(k)), whose node set V(k) represents the sensors active at time k and the edge set E(k) consists of pairs of nodes (u, v) such that nodes u and v can directly exchange messages between each other at time k. By an "active" node we mean a node that has not failed permanently. All graphs considered here are called undirected, i.e., (i, j) = (j, i). The neighbors of a node i is the set Ni of nodes connected to i, i.e. Ni = {j|(i, j) € E}.

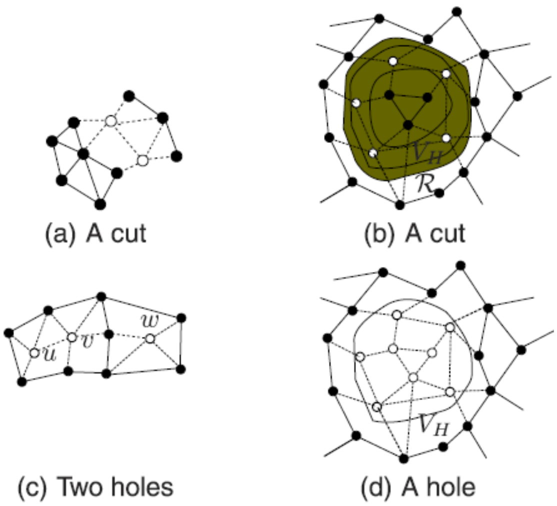

Figure 2: Examples of Cut and Holes

The number of neighbors of i, |Ni(k)|, is known its degree, which is denoted by di(k). A path from nodes i to j is a sequence of edges connecting i and j. A graph is called connected if there is a path between every pair of the nodes. A component Gc of a graph G is a maximal connected sub- graph of G (i.e., no other connected subgraph of G contains Gc as its subgraph). 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 (Figure 2). The number of cuts associated with a cut event is the increase in the number of the components after the event. The problem we seek to address is a twofold. First, we have to enable every sensor node to detect if it is disconnected from the source.

B. Disconnected from source(DOS) Detection

We say that a Disconnected from Source (DOS) event has occurred for node u. The algorithm allows each node to detect the DOS events. The sensor nodes use the computed potentials to detect if DOS events have occurred (i.e., if they are disconnected from the source node). The approach here is to exploit the fact that if the state is close to zero then the node is disconnected from the source, otherwise not. In order to reduce the sensitivity of the algorithm to variations in network size and structure, we use a normalized state. DOS detection part consists of normalized state computation, steady-state detection and connection or separation detection. A sensor node keeps track of the positive steady states seen in the past using the following method. Each sensor node i computes the normalized state difference δ xi(k) as follows:

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

Where $\in$ zero is a small positive number.

Each sensor node keeps an estimate of the most recent "steady state" observed, which is denoted by $x_i^{ss}(k)$. This estimate is updated at every time "k" according to the following rule: if PSSR(k)=1, then $x_i^{ss}(k) \leftarrow x_i$ k; otherwise $x_i^{ss}(k) \leftarrow x_i$ k− 1 . It is initialized as $x_i^{ss}(0) = \infty$. Every sensor 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. This information is kept in a vector $x_i^{ss}(k)$, which is initialized to be empty and is updated as follows: If PSSR(k) = 1 but PSSR(k −1) = 0, then $x^{ss}(k)$ is appended to $x_i^{ss}(k)$ as a new entry. If steady state reached was detected in both k and k - 1 (i.e., PSSR(k) = PSSR(k − 1) = 1, then the last entry of $x^{ss}(k)$ is updated to $x_i^{ss}(k)$.

C. Connected Cut Occurred Somewhere (CCOS) Detection

The algorithm for detecting CCOS events relies on finding a shortest path around a hole, if it exists, and is partially inspired by the jamming detection algorithm [7]. The method utilizes sensor node states to assign the task of hole-detection to the most appropriate nodes. When a sensor 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. The probe messages that are initiated by certain nodes that encounter failed neighbors, and are forwarded from one node to another in a way that if a short path exists around a "hole" created by node failures, the message will reach the initiating node. The pseudo code for the algorithm that decides when to initiate a probe message is included. Each probe message "p" contains the following information:

- A unique probe ID,
- Source Node id S
- Destination node,
- Path traversed (in chronological order), and
- The angle traversed by the probe message.

The list of probes is the union of the probe messages it received from its neighbors and the probe it decided to initiate, if any probe is forwarded in a manner such that if the probe is triggered by the creation of a cut or small hole(with circumference less than lmax) the probe traverses a path around the hole in a counter clockwise (CCW) direction and reaches the node that initiated the probe. Since it is only used to compute destinations of the probe messages.

V. CONCLUSION

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial as well as temporal resolution. The failure of some of its sensor nodes, which is called "cut" can separate the network into multiple connected components. The ability of detecting the cuts by the disconnected nodes and source node of a wireless sensor network will lead to the increase in the operational lifetime of the network. The Distributed Cut Detection (DCD) algorithm proposed here enables every sensor node of a sensor network to detect Disconnected from Source events if they occur. Second, it enables a subset of sensor nodes that experience CCOS events to detect them and estimate the approximate location of the cut in the form of a list of active sensor nodes that lie at the boundary of the cut. The algorithm is based on ideas from both electrical network theory and parallel iterative solution of linear equations. A key strength of the DCD method is that the convergence rate of the iterative scheme is quite fast and independent of the size and structure of the network.

REFERENCES

[1] A Partition Detection System for Mobile Ad-Hoc Networks: Hartmut Ritter, Rolf Winter, Jochen Schiller

[2] Akyildiz, I.F., Su, W., ankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: A Survey. Computer Networks Journal (Elsevier), Vol. 38, No.4 (2002)pp. 393-422.

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

[4] A Survey on Sensor Setworks,‖ : I. F. Akyildiz et al IEEE Commun. Mag., vol. 40, no. 8, Aug. 2002, pp. 102–112.

[5] Ö. Babaoglu, R. Davoli, A. Montresor, Group Communication in Partitionable Systems: Specification and Algorithms, IEEE Transactions on Software Engineering, 2001, vol. 27, no.4.

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

[7] 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.