Cut detection in Wireless sensors network using Distributed Source Separation Detection (DSSD) Approach

Jagdish Pimple*, Prof.Yogadhar Pandey **

* Department of Computer Sci & Engg, Sagar Institute of Research & Technology, Bhopal
** Department of Computer Sci & Engg, Sagar Institute of Research & Technology, Bhopal

Abstract- A classical problem caused by separation of network is partitioning. Predicting those positioning from where the network get separated into the different partition could be a very useful feature that can be provided to applications in a wireless sensors network environment. Indeed, being aware of a future disconnection in the network can help to ensure a better quality of service by adapting the application behavior. 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. In the absence of cuts, the states converge to values that are equal to potentials in a fictitious electrical network. When a set of nodes gets separated from a special node, that we call a "source node", their states converge to 0 because "current is extracted" from the component but none is injected. These trends are used by every node to detect if a cut has occurred that has rendered it disconnected from the source. 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.

Index Terms- Wireless networks, sensor, positioning cut, DOS,CCOS,DSSD.

I. INTRODUCTION

However, several challenges have to be overcome to achieve the potential of WSNs. One of the challenges in the successful use of WSNs come from the limited energy of the individual sensor nodes. Significant current research has therefore been directed at reducing energy consumption at the sensor nodes. In the hardware front, energy efficient components have been developed, and in the software front, power aware routing, low complexity coding, and low power data processing algorithms have been examined. 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. Wireless sensor networks (WSNs) have emerged as a promising new technology to monitor large regions at high spatial and temporal resolution. Virtually any physical variable of interest can be monitored by equipping a wireless device with a sensor and networking these sensors together with the help of their on-board wireless communication capability.

Although these advances are expected to increase the lifetime of the wireless sensor nodes, due to their extremely limited energy budget and environmental degradation, node failure is expected to be quite common. This is especially true for sensor networks deployed in harsh and dangerous situations for critical applications, such as forest fire monitoring.

In addition, the nodes of a sensor network deployed for defense applications may be subject to malicious tempering. When a number of sensors fail, whether due to running out of energy, environmental degradation, or malicious intervention, the resulting network topology may become disconnected.
That is, as a result of failure of a set of nodes, a subset of nodes that have not failed become disconnected from the rest of the network.

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. We call it the Distributed Source Separation Detection (DSSD) algorithm.

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.

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; the reason for this particular name is the electrical analogy introduced . 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 Disconnected frOm Source (DOS) 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 Connected, but a Cut Occurred Somewhere (CCOS) 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. 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.

II. Distributed cut detection in WSN

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 convergence rate of the computation is independent of the size and structure of the network.

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. 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.

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.

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.

III. Distributed Source Separation Detection (DSSD)

DSSD known as an distributed source separation detection . algorithm that 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.

We call it the Distributed Source Separation Detection (DSSD) algorithm. 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. the 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 little attention. Kleinberg et. al. have studied the problem of detecting network failures in wired networks, and proposed schemes for the case when k edges fail independently

To the best of our knowledge, the work by Shrivastava et. al. is the only one that addresses the problem of detecting cuts in wireless sensor networks. They developed an algorithm for detecting ϙ linear cuts, which is a linear separation of on nodes from the base station. The reason for the restriction to 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 to sensor networks deployed in the 2D plane. The algorithm developed in needs a few nodes called sentinels that communicate with a base station either directly or through multi-hop paths.

The base station detects ϙ-cuts by monitoring whether it can receive messages from the sentinels. In contrast to the algorithm in the DSSD algorithm proposed in this paper is not limited to ϙ-linear cuts; it can detect cuts that separate the network into multiple components of arbitrary shapes. Furthermore, the DSSD algorithm is 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 neighbor communication, which eliminates the need of routing messages to the source node. This feature makes the algorithm applicable to 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. Simulations are reported in that illustrate the capability of the algorithm to detect cuts in mobile networks, and also its ability to detect if a “reconnection” occurs after a cut. The DSSD algorithm has been demonstrated in an wireless testbed with MicaZ motes

Even though the proposed algorithm is iterative and involves only nearest neighbor communication, the convergence rate of the algorithm is quite fast and is independent of the size of the network. The assumptions are that the source node 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 power, have physical damage or environmental interference.

The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to sensor node failures The reliability fault tolerance of a sensor node is modelled in using the Poisson distribution to capture the probability of not having a failure within the time interval (0; t): where kk and t are the failure rate of sensor node k and the time period, respectively.

Note that protocols and algorithms may be designed to address the level of fault tolerance required by the sensor networks. If the environment where the sensor nodes are deployed has little interference, then the protocols can be more relaxed. For example, if sensor nodes are being deployed in a house to keep track of humidity and temperature levels, the fault tolerance requirement may be low since this kind of sensor networks is not easily damaged or interfered by environmental noise. On the other hand, if sensor nodes are being deployed in a battlefield for surveillance and detection, then the fault tolerance has to be high because the sensed data are critical and sensor nodes can be destroyed by hostile actions. As a result, the fault tolerance level depends on the application of the sensor networks, and the schemes must be developed with this in mind.
The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. Depending on the application, the number may reach an extreme value of millions.

IV. Experiments

a. Route Discovery

The first criterion in wireless medium is to discover the available routes and establish them before transmitting. To understand this better let us look at the example below. The below architecture consists of 11 nodes in which two being source and destination others will be used for data transmission. 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 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.

In the MANET, the nodes are prone to undergo change in their positions. Hence the source should be continuously tracking their positions. By implementing the AODV protocol in the simulation scenario it transmits the first part of the video through the below shown path. After few seconds the nodes move to new positions.

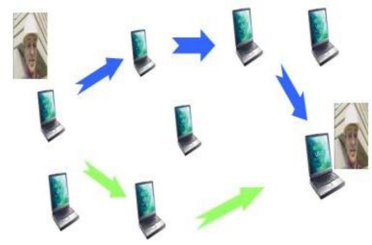

Fig. Route Discovery

b. 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 nodes change and the source doesn't make a note of it then the packets will be lost and eventually have to be resent.

The modified proposed algorithm

Threshold = 50%; success = 0; cutoff = 10%
A := S;
Repeat
If g(A) >= threshold then
B := A;
Let A be neighbor of B that minimizes
pc(B,A) = power-cost(B,A) + v(s)f'(A);
Send message to A;
success = 1;
Until A = D (* Destination reached *)
or if success <> 1 then
if threshold > cutoff then
threshold = threshold /2;
or A = B (* Delivery failed *);

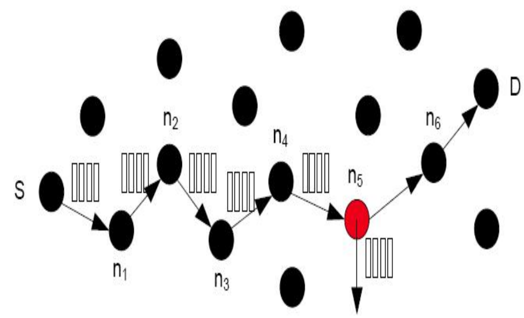

Fig. Route Maintenance

c. Data 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.

d. Minimum-energy multicast tree Module

Our main objective is to construct a minimum-energy multicast tree rooted at the source node. We explore the following two problems related to energy-efficient multicasting in WANETs using a source-based multicast tree wireless multicast and the concept of wireless multicast advantage . Because the problem of constructing the optimal energy-efficient broadcast/multicast tree is NP-hard, several heuristic algorithms for building a source-based energy-efficient broadcast/multicast tree have been developed recently .Wieselthier et al. presented BIP/MIP algorithm which is a centralized source-based broadcast/ multicast tree building centralized algorithm

V. Conclusion

Although we only discussed cut detection in this paper proposed algorithm can also be used for detection of "reconnection". If a component that is disconnected due to a cut gets reconnected later (say, due to the repairing of some of the failed nodes), the nodes can detect such reconnection from their states. Simulations are reported in that illustrate the capability of the algorithm to (i) detect cuts in mobile networks and (ii) detect re-connections after cuts.

There are several issues related to the DSSD algorithm that need to be examined. The first is the appropriate choice of the design parameter s (source strength) in the algorithm.

The potentials of nodes far away from the source typically become smaller as the network size increases. To keep the state values become too small – which will affect cut detection – the parameter s has to be chosen as a large number for a large network. Guidelines for choosing s depending on the size of the network, and how much a-priori knowledge of the network structure is needed to make the appropriate choice, is being investigated.

The states of the nodes computed by the DSSD algorithm are affected by even those node failures that do not lead to cuts. This is intuitive, since the electrical potential of a node in a resistive electrical network is a function of the network structure which changes due to node failures. This feature raises the possibility of designing algorithms to compute "electrical potentials" of nodes in a wireless network so as to detect structural changes that are more complex than simply cuts. While a protocol that enables nodes to detect cuts is useful, there is also a need for protocols that allow a base station to detect when and where a cut has occurred. We envision a protocol that lies on top of the DSSD algorithm to determine the location of a cut when it occurs. This will the subject of future investigation. Another related issue that merits investigation is secure cut detection, 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.

References

[1] Prabir Barooah,Harshavardhan Chenji,Student, Radu Stoleru, and Tama´s Kalma´r-Nagy," Cut Detection in Wireless Sensor Networks".PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. X, XXX 2012

[2] 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.

[3] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configurings EnsorNetworks Topologies," in IEEE Infocom. New York, NY: IEEE, June 2002.

[4] 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.

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

[6] J. Kleinberg, "Detecting a network failure," Internet Mathematics, vol. 1, pp. 37–56, 2003.

[7] J. Kleinberg, M. Sandler, and A. Slivkins, "Network failure detection and graph connectivity," in the 15th ACM-SIAM Symposium on Dis-crete Algorithms, 2004.

[8] 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.

[9] N. Shrivastava, S. Suri, and C.D. To´ th, "Detecting Cuts I Sensor Networks," ACM Trans. Sensor Networks, vol. 4, no. 2, pp. 1-25, 2008.

[10] 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.

[11] 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.

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

[13] 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.

Authors

First Author – Jagdish Pimple, Department of Computer Sci & Engg, Sagar Institute of Research & Technology, Bhopal. Email: pimplejagdish@gmail.com

Second Author – Prof.Yogadhar Pandey, Department of Computer Sci & Engg, Sagar Institute of Research & Technology, Bhopal, Email: p_yogadhar@yahoo.co.in