AN IMPROVED CUT DETECTION ALGORITHM FOR DATA TRANSMISSION IN WSN

[1] M. Dhivyasri, [2]P.E. Prem, [3] S. Nisha
[1]PG scholar, Department of Information Technology, VCEW
[2]Assistant Professor, Department of Information Technology, VCEW
[3]Assistant Professor, Department of Information Technology, KONGUNADU COLLEGE
[1]dhivyasridhivakar@gmail.com, [2] pepremkumar@gmail.com, [3] nithi.becse@gmail.com

Abstract: In this paper consider the problem of detecting cuts by the nodes of a wireless network. 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 DOS (Disconnected from Source) event has occurred to you. When a cut occurs in the network that does not separate a node you from the source node, we say that CCOS (Connected, but a Cut Occurred Somewhere) event has occurred to you. By cutting 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.

Keywords: Wireless network, base station, CCOS , WSN.

1. INTRODUCTION

Routing in WSNs is very challenging due to the inherent characteristics that distinguish these networks from other wireless networks like mobile ad hoc networks or cellular networks Due to relatively large number of sensor nodes, it is not possible to build a global addressing because overhead of ID maintenance is high. In contrast to typical communication networks, almost all the applications of sensor networks require the flow of sensed data from multiple sources to a particular Base station (BS). This, however, does not prevent the flow of data to be in other forms (e.g., multicast or peer to peer). Sensor nodes are tightly constrained interms of energy, processing, and storage capacities. Thus, they require careful resource management, which is not supported by other network protocol. Position awareness of sensor nodes is important since data collection is normally based on the location which is not addressed by legacy protocols. Hence traditional IP-based protocols may not be applied to WSNs. Many new algorithms have been proposed for the routing problem in WSNs.

These routing mechanisms have taken into consideration the inherent features of WSNs along with the application and architecture requirements. The task of finding and maintaining routes in WSNs is nontrivial since energy restrictions and sudden changes in node status (e.g., failure) cause frequent and unpredictable topological changes. To minimize latency and energy consumption, routing techniques proposed in this literature employ some well-known routing tactics e.g., data aggregation and clustering. The DCD algorithm is based on the following electrical analogy. Imagine the wireless sensor network as an electrical circuit where current is injected at the source node and extracted out of a common fictitious node that is connected to every node of the sensor network. Each edge is replaced by a resistor. When a cut separates certain nodes from the source node, the potential of each of those nodes becomes 0, since there is no current injection into their component. The potentials are computed by an iterative scheme which only requires periodic communication among neighboring nodes. The nodes use the computed potentials to detect if DOS events have occurred (i.e., if they are disconnected from the source node). To detect CCOS events, the algorithm uses the fact that the potentials of the nodes that are connected to the source node also change after the cut. However, a change in a node‟s potential is not enough to detect CCOS events, since failure of nodes that do not cause a cut also leads to changes in the potentials of their neighbors. Therefore, CCOS detection proceeds by using 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 nodes that detect CCOS events then alert the source node about the cut. Every node keeps a scalar variable, which is called its state. The state of node i at time k is denoted by x. Every node i initializes its state to 0. During the time interval between the kth and k þ 1th iterations, every node i broadcasts its current state and listens for broadcasts from its current neighbors. Let N be the set of neighbors of node i at time k. Assuming successful reception, i has access to the states of its neighbors, i.e., x for j, at the end of this time period. The node then updates its state according to the following state where di is the degree of node i at time k, and 1AðiÞ is the indicator function of the set A. That is, 1 if i < (source node). After the state is updated, the next iteration starts. At deployment, nodes go through a neighbor discovery and every node I determines its initial neighbor set Nið0Þ. After that, i can update its neighbor list as follows: If no messages have been received from a neighboring node for the past drop iterations, node i drops that node from its list of neighbors. The integer parameter drop is a design choice.

Each PROBE message p contains the following information:

- a unique probe ID,
- probe centroid Cp
- destination node,
- path traversed (in chronological order), and
- The angle traversed by the probe around the centroid.

The probe is forwarded in a manner such that if the probe is triggered by the creation of a small hole or cut (with circumference less than max), the probe traverses a path around the hole in a counter-clockwise (CCW) direction and reaches the node that initiated the probe. In that case, the net angle traversed by the probe is 360 degree. On the other hand, if the probe was initiated by the occurrence of a boundary cut, even if the probe eventually reaches its node of initiation, the net angle traversed by the probe is 0. Nodes forward a probe only if the distance traveled by the probe (the number of hops) is smaller than a threshold value „max. Therefore, if a probe is initiated due to a large internal cut/hole, then it will be absorbed by a node (i.e., not forwarded because it exceeded the distance threshold constraint), and the absorbing node declares that a CCOS event has taken place.

2. SYSTEM MODEL

2.1 Connection establishment:

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.

2.2 Data Transmission:

The basic idea is to autonomously learn unknown and possibly random mobility parameters and to group mobile nodes with similar mobility pattern into the same cluster for data transmission. The nodes in a cluster can then interchangeably share their resources for overhead reduction and load balancing, aiming to achieve efficient and scalable routing.

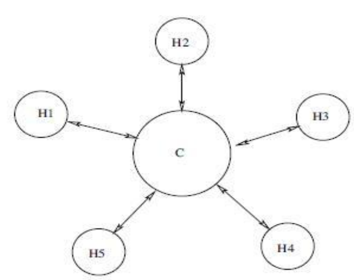

Figure 1: Connection Establishment

The key bottleneck lies in the transmission part, which requires collecting excessive amount of feedback information from all users and also needs complex operations to make decisions on BS coordination at every time slot. To overcome such complexity, the original optimization problem is decomposed into many sub-problems and these are solved with different time scales. The complexity becomes much lower than that of the optimal algorithm, yet sustains high efficiency in data transmission.

3. ALGORITHM

3.1 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. 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.

3.2 CCOS Detection

- **Jamming Detection Algorithm**

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.

Assumption 1:

- The sensor network is a two dimensional geometric graph, with Pi 2 R2 denoting the location of the i-th node in a common Cartesian reference frame;
- Each node knows its own location as well as the locations of its neighbors.

3.3 DOS Detection.

The DOS detection part of the algorithm is applicable to arbitrary networks; a node only needs to communicate a scalar variable to its neighbors. The CCOS detection part of the algorithm is limited to networks that are deployed in 2D Euclidean spaces, and nodes need to know their own positions. The position information need not be highly accurate.The approach here is to exploit the fact that if the state and based on binaryVariables is close to 0 then the node is disconnected from the source, otherwise not (this is made precise)

Theorem 1

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.

3.4 Sink

All data originated from sensors in a network are received by their corresponding cluster head, which aggregates them with its own data into only one single data (i.e., taking the average). Precisely, each sensor sends its data directly to its head, where data are aggregated. We distinguish two types of aggregation. In the first scenario, called local data aggregation, aggregation occurs only within clusters and all data aggregated by cluster heads are forwarded to the sink without further aggregation. Thus, the sink receives data from each head in each round. In the second scenario, called global data aggregation, the sink receives only one data packet in each round that represents the aggregation of all data aggregated by cluster heads. Precisely, each cluster head also aggregates its own aggregated data with the aggregated data it has received from a cluster head and forwards the result to another cluster head.

4. CONCLUSION

This paper, described the cover-sense-inform framework WSNs using DDOS algorithm, where k-coverage, sensor scheduling, and data forwarding are jointly considered. It proposed energy-efficient geographic forwarding protocols on duty-cycled, where sensed data are forwarded to the sink through cut details. The joint coverage and geographic forwarding protocols can be used for applications that demand a high - coverage degree, such as intruder detection and tracking. It is also useful for applications that require data aggregation and those where all data originated from sources should reach the sink without prior aggregation.

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´ the, "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.

[1] 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.

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

[3] 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.

[4] P. Bose and P. Morin, "Online Routing in Triangulations," SIAM J. Computing, vol. 33, no. 4, pp. 937-951, 2004.

[5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," ACM/ Kluwer Wireless Networks, vol. 7, no. 6, pp. 609-616, 2001.

[6] B. Bolloba´ s, The Art of Mathematics: Coffee Time in Memphis. Cambridge Univ. Press, 2006.

[7] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," IEEE Personal Comm., vol. 7, no. 5, pp. 28-34, Oct. 2000.

[8] B. Chen, K. Jameson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc WirelessNetworks,"

[9] ACM Wireless Networks, vol. 8, no. 5, pp. 481-494, 2002.

[10] X. Du and F. Lin, "Maintaining Differentiated Coverage in Heterogeneous Sensor Networks," EURASIP J. Wireless Comm. and Networking (WCN), vol. 5, no. 4, pp. 565-572, 2005.

[11] H. Gupta, Z. Zhou, S. Das, and Q. Gu, "Connected Sensor Cover:Self-Organization of Sensor Networks for Efficient Query Execution,"IEEE/ACM Trans. Networking, vol. 14, no. 1, pp. 55-67, Feb.2006.

[12] P. Hall, Introduction to the Theory of Coverage Processes. John Wiley& Sons, Inc., 1988.