International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume2, Issue 8 (August '2016)

# AN ENHANCED ROBUST ENERGY EFFICIENCY ROUTING IN WIRELESS SENSOR NETWORKS

**Dr.M.Sughasiny,**
Head-PG & Research Department of Computer Science,
Srimad Andavan Arts & Science College (Autonomous),
Tiruchirapalli-620005, Tamilnadu, India.

**Abstract:** A wireless sensor network can be separated into multiple connected components due to the failure of some of its nodes, which is called a "cut." Therefore, we consider the problem of detecting cuts by the remaining nodes of a wireless sensor network. We propose a distributed algorithm can be to detect cuts, named the Distributed Cut Detection (DCD) algorithm. This algorithm allows each node to detect DOS events and a subset of nodes and also to detect CCOS events separation and reconnection to the source in mobile networks is a topic of ongoing research. These algorithms are centralized and detect only a linear cut. Therefore we also propose RECDP (Lightweight and Robust Cut Detection Algorithm) protocol that to detect cuts while providing energy-efficiency and robustness to attack. Using distributed, cluster-based algorithms recognizes and determines the scope of disrupted connectivity while examining available data for evidence of an attack. A result indicates that the energy efficiency can be improved by an order of magnitude in denser networks while malicious nodes are detected at variation of 1% from expected behavior

***Keywords: WSN, DCD, DOS, RECDP, Clustering , Energy Efficiency***

## I.INTRODUCTION

Wireless sensor networks (WSNs) are a promising technology for monitoring large regions at high spatial and temporal resolution which with small, low-power, inexpensive radios, have attracted a large amount of research leading to interesting and innovative applications in disaster response, military surveillance and medical care among others[1] . The most challenge problem in WSN is to maintain a network connectivity which used to reliably communicate between peers or deliver data to a specified point, or sink, in an energy efficient manner.

A node failure can be happen due to various factors like mechanical/electrical problems, environmental degradation, battery depletion, or hostile tampering and for all typically limited energy budgets of the nodes that are powered by small batteries. These failure nodes can reduce the number of multi hop paths in a network which will cause of a subset of nodes that have not failed to become disconnected from the rest and resulting will be a "cut." Two nodes are said to be disconnected if there is no path between them.

To see the benefits of a cut detection capability, imagine that a sensor that wants to send data to the source node has been disconnected from the source node. Without the knowledge of the network's disconnected state, it may simply forward the data to the next node in the routing tree, which will do the same to its next node, and so on. However, this message passing merely wastes precious energy of the nodes.The cut prevents the data from reaching the destination.

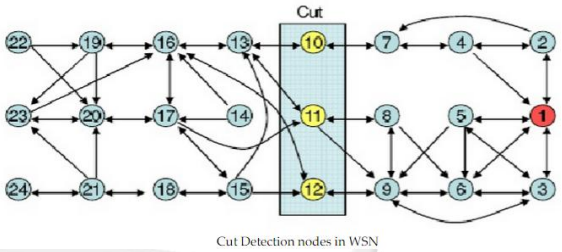


**Figure 1: Cut Detection nodes in WSN**

Hence we look into figure 1 the problem of detecting cuts by the nodes of a wireless network and assume that there is a specially designated node in the network, which we call the source node. A key component of the Distributed Cut Detection (DCD) algorithm is a distributed iterative involving computation step through which the nodes compute their electrical potentials and the convergence rate of the computation is independent of the size and structure of the network. However, these algorithms are centralized and detect only a linear cut only.

We also propose RECDP algorithm which is designed for resource constrained situation to built on top of the DSSD (Distributed Source Separation Detection) algorithm a RECDP incorporates outlier detection is a statistical data analysis technique used to detect a masquerade or impersonation attack.

A detection can enables the identification of statistically improbable data at extreme deviation, a possible indicator of malicious activities, and provides a light weight mechanism to validate neighbor data.

 

## II.RELATED WORK

The identification of holes in a wireless sensor network is of primary concern because the breakdown of sensor nodes in a larger area often indicates one of the special events to be monitored by the network in the first place. This task of identifying holes is especially challenging since typical wireless sensor networks consist of lightweight, low-capability nodes that are unaware of their geographic location. There was also a secondary interest in detecting holes in a network that is the routing schemes do not assume knowledge of the geographic location of the network nodes but rather perform routing decisions based on the topology of the communication graph. Holes are salient features of the topology of a communication graph.

A simple distributed procedure used to identify nodes near the boundary of the sensor field as well as near hole boundaries.

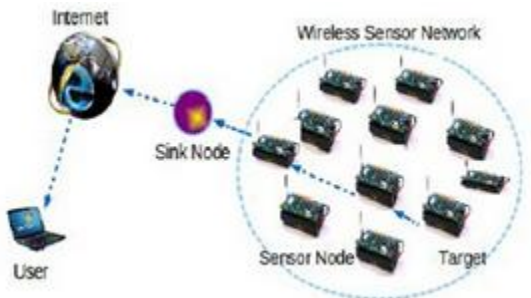


**Figure 2 : Illustration of Wireless Sensor Networks**

This hole detection algorithm is based purely on the topology of the communication graph which is a nodes can communicate with each other. We illustrate the secondary interest of our hole detection procedure using several examples as in figure 2.

Networked Embedded Systems play an important role and which will affect many aspects of our lives. New applications are being enveloped in areas such as health-care, industrial automation, smart building and rescue operations. The European Integrated design as "Reconfigurable Ubiquitous Networked Embedded Systems". By the way of reasoning upon the degree of connectivity with neighbors, a sensor node finds the proper position where to stop in order to re-establish connectivity. Factors influencing the method performance are singled out and criteria for their selection are discussed. Simulations show that the proposed method is effective and efficient notwithstanding packet loss. The conceptual implementation of the Verify function and the function estimates the connectivity degree of the sensor node with the safe partition and many good links are to be used in sensor node has with safe nodes[2].

Assume a network S of n sensors, modeled as points in a two-dimensional plane. An "-cut, for any 0 < " < 1, is a linear separation of "n nodes in S from a distinguished node, the base station. We make that the base station can detect whenever an "-cut occurs by monitoring the status of just O( 1" ) nodes in the network.

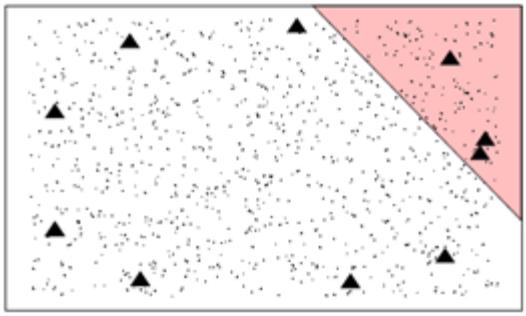

**Figure 3: 1000 sensor nodes scattered randomly and a linear cut**

Our scheme is deterministic and it is free of false positives: no reported cut has size smaller than 12 "n. Besides this combinatorial result, we also propose efficient algorithms for finding the O( 1 ) nodes that should act as sentinels, and report on our simulation results, comparing the sentinel algorithm with two natural schemes based on sampling.

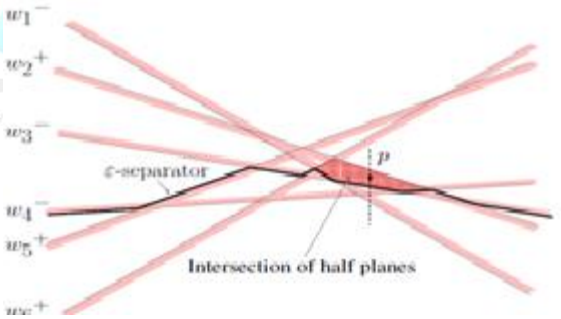


**Figure 4: The intersection of half -planes**

A vast amount of applications and mechanisms recently developed for mobile ad-hoc networks could greatly benefit from the utilization of network status information. That includes, but is not limited to the detection of network partitioning. 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. Nodes within each partition are still able to communicate with each other but nodes in other partitions are unreachable.

A typical centralized partition detection system structure. Obviously the server is the most vulnerable element of successful partition detection[3]. For our simulations we chose the first node that activates itself to become the server. In a real network we would have to use a mechanism similar to a cluster head election phase used in hierarchical routing algorithms as already mentioned or a node initiating the

application would serve as the beacon source (e.g. the game initiator).

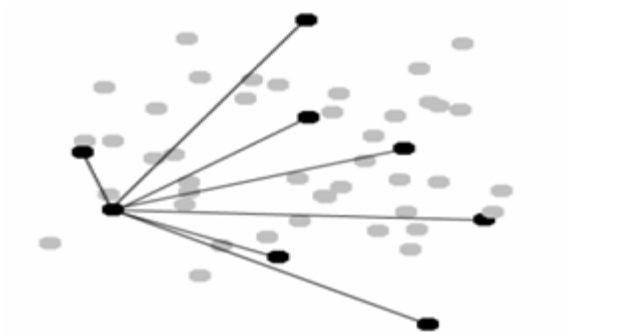

**Figure 5: Typical centralized partition detection system structure**

The distributed approach of course generates much more traffic since much more beacon messages have to be exchanged and also every active node needs a buddy. The load generated settings chosen are the same as for the distributed approach. Furthermore, every active node should communicate with 3 other active nodes (temporarily less).

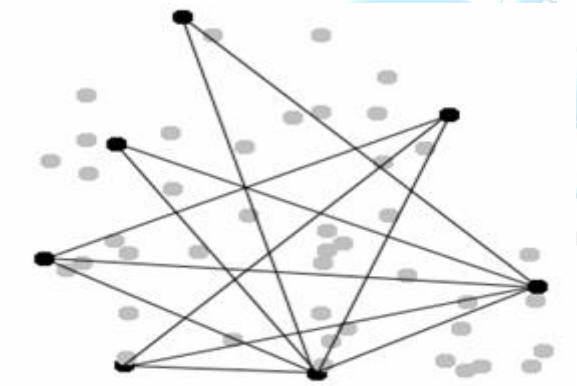

**Figure 6 : Typical partnerships amongst nodes in the distributed approach**

Although the separation of a buddy node from its corresponding active node during a network partition is a rare event, it might result in the network partition going undetected. Therefore, we want to evaluate a buddy election process based on average signal strength or another perhaps more suitable metric[4] . 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 [4]. These trends are used by every node to detect if a cut has occurred that has rendered it disconnected from the source. The source started transmitting data from packets toward the sink. For the reason the sink is at the limit of radio range, it gets very high packet loss from the source. We call this situation a communication hole as in figure 7.

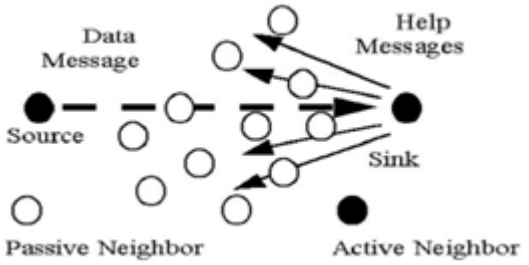


**Figure 7: Communication hole**

When a neighbor can receives a help message, it may decide to join the network. This situation is illustrated. When a node joins the network, it starts transmitting and receiving packets, i.e., it becomes an active neighbor. As soon as a node decides to join the network, it signals the existence of a new active neighbor to other passive neighbors by sending an neighbor announcement message.

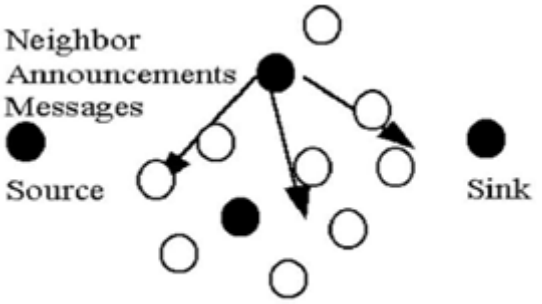


**Figure 8 : Transition**

When the process completes, the group of newly active neighbors that have joined the network make the delivery of data from source to sink more reliable. The process will restart when some future network event (e.g., node failure) or environmental effect (e.g., new obstacle) causes packet loss again. In this section, we describe the ASCENT algorithm and their components.

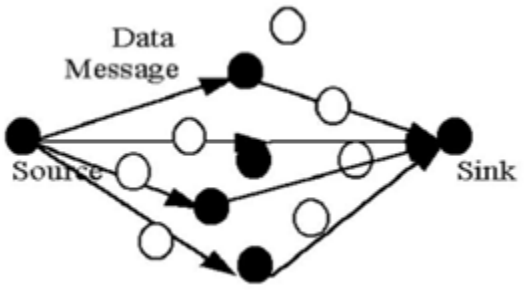


**Figure 9 : Final State**

There are several effects which is related to the DSSD algorithm that need to be examined initially. 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[5] . 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.

**(DCD): Distributed Cut Detection Algorithm**

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 [4] . DOS detection part consists of steady-state detection, normalized state computation, and connection/separation detection. Every node i maintains a binary variable DOS, 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.

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 $\delta x_i(k)$ as follows:

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

**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 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 ALERT message[4][6]. The only requirement is that the source node never be a part of it. Therefore, even if the graph keeps changing with time, e.g., due to node mobility, the states of the nodes that are disconnected from the source will converge to 0. The DOS detection part of the proposed algorithm comes with a guarantee on the maximum delay incurred, which is stated in the following Lemma.

**RECDP: Lightweight and Robust Cut Detection**

RECDP is a cut detection protocol designed for resource constrained situations where ER-CD is too heavy. This protocol provides a good energy-efficiency, and robustness against a particular security threat, the masquerade attack. A lightweight protocol (RECDP) used enhances the state of the art cut detection algorithm with robustness against the masquerade attack at low computation and memory cost. Extensive simulations that verify and validate performance of the protocols across a variety of network sizes and densities and an implementation on Epic wireless sensor motes extending and confirming the simulation results. We propose RECDP, the Lightweight and Robust Cut Detection protocol.

These protocol overlays DSSD with a mechanism to detect malicious nodes. It does this by using statistical measures to detect outliers in the states of neighboring nodes. Hence it may seem tempting to directly use the states of neighbors as samples for the construction of a distribution for outlier detection. However, these naive approach is insufficient. Because of some cases, it is hard to assume a bell-shaped distribution based on samples of received states from neighbors for the below reasons: The sample size of the received states is too small for some nodes, the states in close proximity are not always similar and regional variations in the sample set exist for nodes that are located close to the source node and the range of the state value is relatively large (depending on the source strength)[7]. Therefore a straightforward outlier rejection algorithm based on the samples of neighbors' states is not sufficient. To this end, an application-specific outlier detection algorithm is proposed that mitigates the aforementioned problems (Algorithm 5). The main idea is to derive a new sample set from the sample set of neighbor states by taking the sum of differences to p nearest takes for each sample. By representing a new sample as a difference of only a few nearest states, regional variations in states can be canceled out, and the range of sample becomes much smaller, thereby leading to a better sampling even for small data set. Since RECDP builds on the DSSD protocol, only enhancements were investigated. In particular, experiments were conducted to measure malicious node detection latency, the number of iterations required to detect a malicious source. For this experiment, a cut was made by turning off some nodes at iteration k, after the network had converged. And then, at iteration k + 1, a malicious source node began to inject false state into the network from a location in the disconnected segment. Elapsed time in the number of iterations between malicious source injection and detection by nodes in the disconnected [8] segment was measured. For different thresholds (critical value in the Student t-table), the experiment was repeated using different W values, a representation of how much the state of a malicious source deviates from the average state of its neighbors in percentage,

$$\Psi = (x_i(k) - \textstyle\sum_{j \in N_i(k)} x_j(k)/|N_i(k)|)/100, \quad \ldots [2]$$

Where xi(k) is the state of the malicious node (node i) and Ni(k) is the set of its neighbors at iteration k.

## III.PERFORMANCE EVALUATION

Our algorithm was tested using NS-2 simulation and we have taken the performance by varying number of nodes inside the network .we have varied nodes from 50 to 250 and check the performance with parameters like packet delivery ratio, Average delay ,Energy consumed, Overhead, Average Throughput and packet loss . we have consider our simulation parameters as mentioned in Table 1.

**Table 1 : Simulation Parameters**

| PARAMETER | VALUE |
|---|---|
| Number of nodes | 50,100,150,200,250 |
| Routing protocol | AODV |
| Source strength | 100 |
| $\epsilon_{zero}$ | 10^-10 |
| $\epsilon_{DOS}$ | 10 ^-3 |
| $\tau_{guard}$ | 3 |
| $\tau_{drop}$ | 4 |
| $\overline{\ell_{max}}$ | 15 |
| $\overline{r^{\Delta ss}}$ | 0.35 |
| Initial energy | 50 joules |

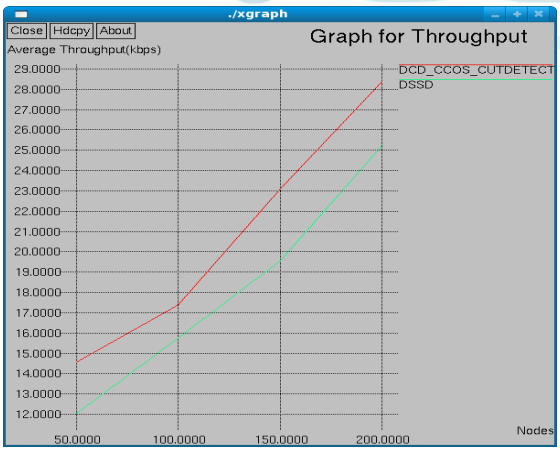


**Figure 10: Number of nodes vs Average throughput**

In figure 10 shows that our proposed DCD_CCOS detection gives better throughput compared to DSSD method and speed of transmission will be high in our method.

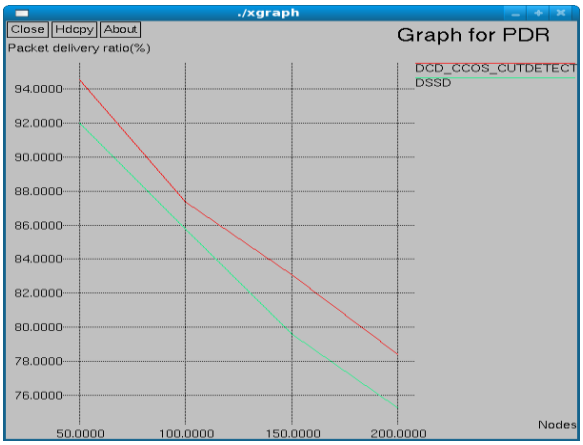


**Figure 11: Number of nodes vs Packet delivery Ratio**

In figure 11 shows that our proposed DCD_CCOS detection gives better delivery ratio compared to DSSD method and we have achieved good improvement in pdr.

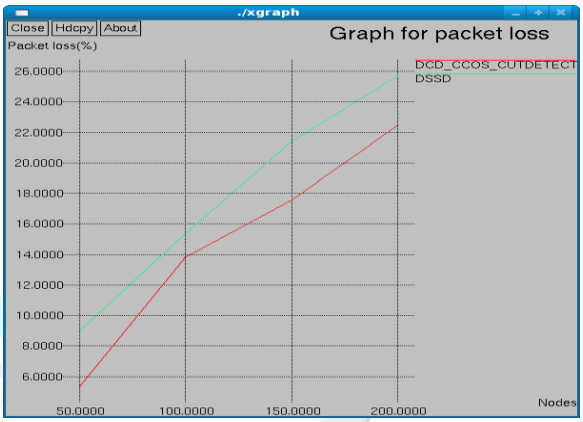


**Figure 12: Number of nodes vs Packet loss ratio**

In figure 12 shows that our proposed DCD_CCOS detection have less packet loss compared to DSSD method and this proves we have overcome the cut efficiently.

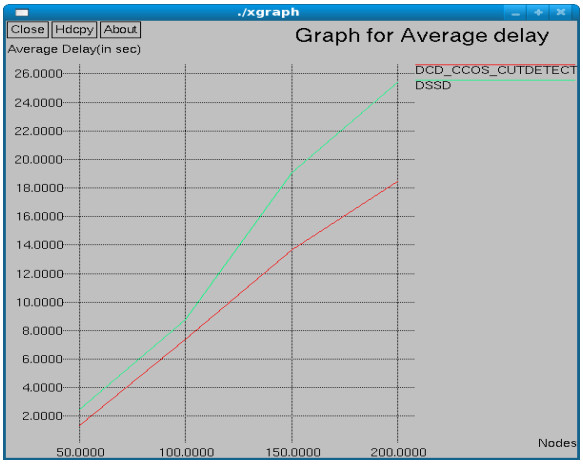


**Figure 13 Number of nodes vs Average Delay(in ms)**

In figure 13 shows that our proposed DCD_CCOS detection gives less delay compared to DSSD method and it proves that our method completes the data transmission with less delay.

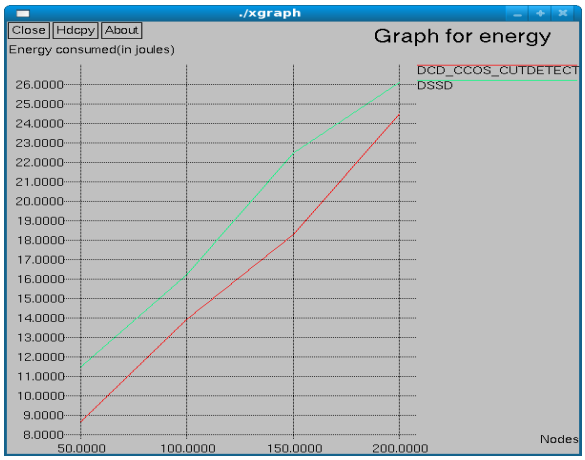


**Figure 14: Number of nodes vs Average Energy consumed**

In figure 14 shows that our proposed DCD_CCOS detection achieves less energy consumption to detect cut and for data transmission compared to DSSD method.

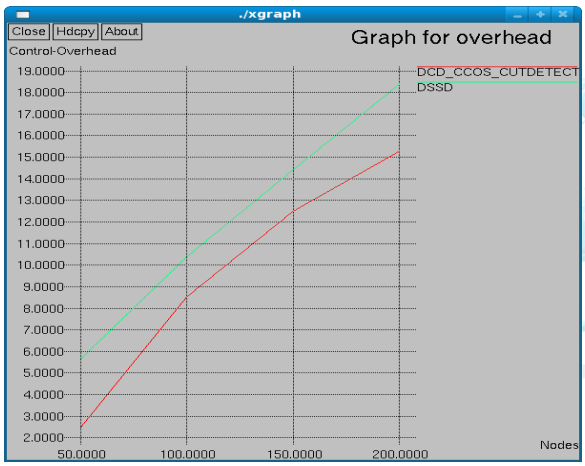


**Figure 15: Number of nodes vs Overhead**

In figure 15 shows that our proposed DCD_CCOS detection have less overhead (i.e) it spends less number of route request, reply and broadcast packets spend to identify the route for data transmission while compared to DSSD method.

## IV.CONCLUSION AND FUTURE WORK

In this paper we proposes DCD algorithm to enables every node of a wireless sensor network to detect Disconnected from Source events if they occur. Second, it enables 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. Also we propose a new robust, energy-efficient algorithms to enhance the detection of disrupted network connectivity in harsh, unattended low-security environments using a network composed of resource-constrained nodes. By the way an adoption of a clustered, leader-based convergence algorithm shown that the possible way to greatly reduce the energy required to detect a cut. This algorithm also used to enhances security by enabling detection of malicious source nodes even at very low thresholds. A same algorithm also can offer that the security without incurring the additional overhead needed for clustering and leader election which with the parameters examined include cluster size, node density, and deviation thresholds. For further we need to improve the speed of trade off energy uses and malicious source detection speed for optimal results in arbitrary networks.

## V.REFERENCES

[1] G. Dini, M. Pelagatti, and I.M. Savino, "An Algorithm for Reconnecting Wireless Sensor Network Partitions," Proc. European Conference, Wireless Sensor Networks 2008, pp. 253-26.

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

[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, Oct. 2004), pp. 489-497.

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

[5]A. Cerpa, D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, IEEE Transactions on Mobile Computing 3 (3) (2004) 272–285.

[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] H. Ritter, R. Winter, J. Schiller, A partition detection system for mobile ad-hoc networks, in: Proceedings of IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON), 2004, pp. 489–497.

[8] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, SIGMOD Record 29 (2) (2000) ,pp 427–438.

[9] B. Karp, H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, in: Proceedings of Conference on Mobile Computing and Networking (MOBICOM), 2000, pp. 243–254.

[10] D. Niculescu, B. Nath, Trajectory based forwarding and its applications, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom), 2003, pp. 260–272.

**AUTHOR PROFILE**

Dr.M. Sughasiny obtained her Ph.D in Computer Science from Bharathiar University, Tamil Nadu ,India. She has 14 years of teaching experience. She is currently working as Head PG and Research Department of Computer Science Srimad Andavan Arts & Science College. Besides that she is Microsoft Professional certified and acting as an ISO internal auditor and member of Board of Studies in Computer Science in autonomous college. She presented papers in various international and national conferences and acted as an chief guest in various colleges. She participated in various national and international seminars and workshops.