Journal of
Computer Science & Systems Biology

Pallavi et al., J Comput Sci Syst Biol 2018, 11:3
DOI: 10.4172/jcsb.1000276

Research Article Open Access

An Efficient Approach to Preserve the Network Connectivity of WSN by Cautiously Removing the Crossing Edges Using COLS

Pallavi R*, Srinivas BC and Banu Prakash GC

Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka, India

Abstract

In the present scenario, Wireless Sensor Network (WSN) is widely used in applications such as Disaster Relief operations, Biodiversity mapping, Intelligent Buildings or Bridges, Machine Surveillance and Preventive maintenance, Precision Agriculture, Medicine and Health care etc. which has led to the deployment of enormous sensor nodes leading to the complexity of the network. Extensive research work has been carried out for monitoring these sensor devices for connectivity, coverage, load balancing, network structure etc. Study on these complex networks is a challenging task. Such networks can be modelled with the help of a graph, which exhibits the properties of a nonplanar graph. In this paper, we would like to propose an algorithm "Coordinate theory On Line Segment" (COLS) to reduce a nonplanar graph to a planar graph by removing the crossing edges carefully. The proposed algorithm preserves the topological structure without compromising Quality of Service of the original.

Keywords: Wireless sensor networks; Network connectivity; Nonplanar graph; Planar graph

Introduction

WSN is a wireless network consisting of spatially distributed sensors to monitor the physical and environmental conditions. WSN's are emerging as a promising technology for information extraction from the surrounding environment. Small-sized electronic devices, the sensor nodes have sensing capability with limited processing and storage. Sensor nodes can sense and record any changes within a limited range. Information collected at the source is aggregated and communicated to the destination (sink node) via multi-hop routing paradigm. Numerous such nodes are deployed within an area of interest either through an Unmanned Aerial Vehicle (UAV) or manual deployment. Post deployment, the nodes communicate amongst each other forming a WSN [1].

Network Connectivity is the crucial part and also an important QoS of WSN [2-5]. For an information to travel from the source node to the sink node, the whole network with the sink node and sensors should be connected throughout the lifetime of the network. Two nodes are said to be connected if they are within each other's transmission radius. In Figure 1, the nodes are connected and adjacent nodes are within the nodes sensing radius δ.

The design-complexity of WSN can be modelled using Graph Theory concepts. The topological structure of a WSN can be viewed as a graph [6,7]. A graph G is denoted as G=(V, E), where V is the set of vertices and E the set of edges. If a topology of WSN consist of x nodes then,

={n, n2, n3 … nx} where ni is the node number 1 ≤ i ≤ x,

E={eij} where eij is defined as a possible communication between node ni and nj,

i.e., o (i, j) Є E iff there exists an edge between I and j, 1 ≤ i, j ≤ x and i ≠ j.

Such graphs are simple by construction. They are loop-free and do not exhibit any parallel edges. The graphs are always by default bi-directional as the sensor ranges are identical. In this paper we assume that the sensor nodes are static and the graphs are thus static graphs (Figure 1).

There are many types of graph formations *viz.* simple graph, peterson graph, partite graph, bipartite graph, planar and nonplanar Graphs etc. [8]. If a graph can be embedded on a plane, i.e., the edges only intersect at the vertices, then such a graph is called a planar graph (Figure 2a). The characteristics of these graphs exhibit no crossing edges or intersecting edges. A graph which is not planar is a nonplanar

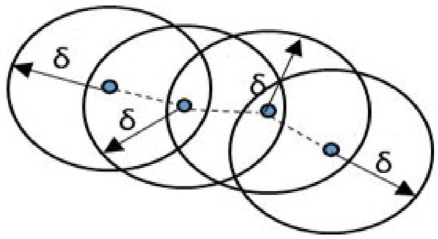

Figure 1: Wireless Sensor Network.

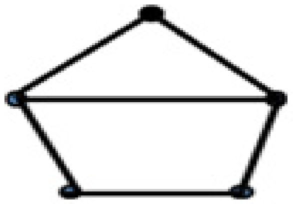

Figure 2a: Planar graph.

***Corresponding author:** Pallavi R, Associate Professor, Department of Computer Science and Engineering, Sir M Visvesvaraya Institute of Technology, Bangalore, Karnataka, India, Tel: +918971834256; E-mail: pallavi_cs@sirmvit.edu

Received March 20, 2018; **Accepted** April 10, 2018; **Published** May 10, 2018

Citation: Pallavi R, Srinivas BC, Banu Prakash GC (2018) An Efficient Approach to Preserve the Network Connectivity of WSN by Cautiously Removing the Crossing Edges Using COLS. J Comput Sci Syst Biol 11: 224-228. doi:10.4172/jcsb.1000276

graph (Figure 2b). In a Nonplanar graph, edges can intersect anywhere on a given plane. Concepts of Planar graph and Nonplanar graph can be used in WSN for building the topology.

In WSN, the information travels from source node to the sink node via intermediate nodes. this exchange of information between the source node to the sink node is done through multi-hop routing paradigm. Various routing protocols [9,10] are used to calculate the best path from the source to the destination. The best path for the data transmission is computed with the help of routing tables. The fundamental requirement of all these routing protocol is that the topological structure of a network should always be connected. As WSN can also be represented as a nonplanar graph, it can be easily simplified to a planar graph and yet retain the original connectivity.

Muhammad and Egerstedt [11] proposed a mathematical model using the technique Maximal Simplicial Complex. The authors used the concept of crossing generators, graph amalgamation and isomorphism to simplify a given graph. Wang and Wu [12] used the concepts of to deduce a nonplanar graph to a planar one. In this paper, we would like to propose a technique called Coordinate theory On line segment (COLS) to reduce a given nonplanar graph (complex) to a planar graph (simple).

Location information can be obtained by using either the Position Sensor or a Global Positioning System (GPS) embedded sensor.

The rest of the paper is organized as follows: Section II deals with the mathematical concept which brings out different formal theories. Section III explains the COLS algorithm. Section IV describes the simulation setup. Section V tabulates the results of various topologies used in the experiment.

Mathematical Concept

In a nonplanar graph as shown in Figure 3, the path taken by a data packet to traverse from the source to the sink is shown with a dashed arrowed line. The path between the source and destination modifies based on the density of packets in the network. At a given instance of time, only one edge is used for traversing between node *v1* to node *v2*. Number of edges incident from a given node *vi* is defined as *deg (vi)*.

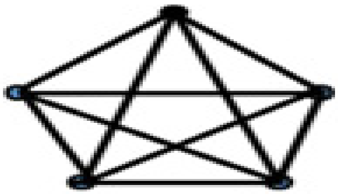

Figure 2b: Nonplanar graph.

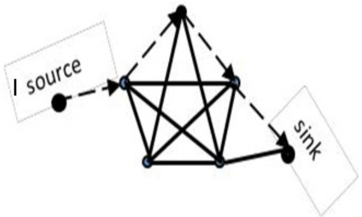

Figure 3: Path taken on a Nonplanar graph.

At any instance of time only one edge is selected for communication between the two nodes and thus fewer edges can be removed from the graph without disturbing the topology. This further helps in removing crossing edges from the graph and thus simplifying it. Mathematical concepts play a vital role in the simplification of a given graph. In this paper we would like to use the mathematical concept of Coordinate theory on line segments for removing the crossing edges.

Axiom/Postulate

Any non-planar graph G=(*V, E*) can be reduced to a planar graph P=(*V, E`*) where *E` E* and is obtained by carefully removing the crossing edges and yet retain the cardinality of the graph i.e., *V=V.*

Using Direction of a reference point: Two line segments $\overline{\overline{\ }}$ and $\overline{\ }$ intersect if (*p, q, r*) and (*p, q, s*) have different directions and (*r, s, p*) and (*r, s, q*) have different directions [13].

For a given ordered triplet *(x, y, z)*, the different directions of a reference point *z* with respect to the line segment $\overline{\overline{\ }}$ is as shown in Figure 4.

Using slope of a given line: Let $\overleftrightarrow{\ }$ and $\overleftrightarrow{\ }$ be two lines. The slope of $\overleftrightarrow{\ }$ can be computed by:

$$m_1=(q-b)/p \tag{1}$$

$$m_2=(s-b)/r \tag{2}$$

From (2) and (3) if $m_1=m_2$ then the slopes of the lines $\overleftrightarrow{\ }$ and $\overleftrightarrow{\ }$ are same. Thus the lines are parallel as shown in Figure 5. Similarly, if $m_1 \neq m_2$ then the slopes of the lines $\overleftrightarrow{\ }$ and $\overleftrightarrow{\ }$ are different i.e., the lines are not parallel and they intersect at some coordinate [14-19].

Using cross product on the coordinates

Two line segments $\overline{\overline{\ }}$ and $\overline{\ }$ are said to intersect each other if the ends of one segment i.e., $\overline{\overline{\ }}$ are on the either side of the other segment $\overline{\ }$ and vice-versa as shown in Figure 6. i.e., if point *r* is towards the left of point *p* and point *s* is towards the right of point *q*, then $\overline{\ }$ is intersecting the

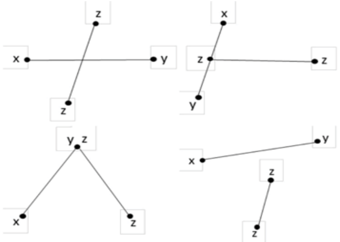

Figure 4: Orientation of a reference point *z* in the ordered triplet *(x, y, z)*.

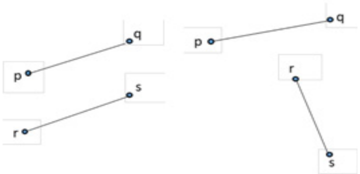

Figure 5: Using the slopes *m1=(q-b)/ p* and *m2=(s-b)/ r*.

line $\equiv$.. This can be computed by calculating the cross products on the coordinates of the points [20-25]. Computing the cross products, one for *p* and the other for *q*:

In this paper, we have used the concept of "Using the Direction of a reference point" i.e., Orientation technique for eliminating the crossing edges in a network (Figure 6).

Algorithm for implementing the orientation technique

```
Algorithm nonplanar to planar (G: Graph)
Step 1: initialize k
Step 2: repeat for each of the node n in a Graph G
Step 3: repeat for each of the one-hop neighbor of a given node in G
Step 4: //store the edges of each node into the list EEN:
//Edge of Each Node
EENk0=Edgei0
EENk1=Edgeij
Step 5: increment k
Step 6: do for all subsequent edges
//Algorithm do Intersect returns true if the two given line segments intersect
if do Intersect(EENx0, EENx1, Edgei0, Edgei1) then decrement k
break
end if
end for
Step 7: loop to Step 3
Step 8: loop to Step 2
```

The time complexity of the algorithm is O(k).

Simulation

If the results have the same sign, the coordinates are on the same side of the line, the segments don't intersect. If one is positive and the other negative, then the coordinates are on the opposite sides up connected networks. Each node had a transmission radius of 250 m. We also changed the node transmission range while keeping the number of deployed nodes same. The results of the individual experiments are averaged over 20 trials for different network topologies.

Experimental setup

The experiment was conducted using a Network Simulator NS2. Different sets of topology were considered where the nodes ranged from 20, 50, 100 to 150. We assume that the topology created was a strongly connected network. For e.g., in a topology of 5 Nodes as shown in Figure 6, every node is connected to every other node in the network. The coordinates or positional information and the final result matrix where only the non-intersecting edges are considered are shown in result 1 [26-29].

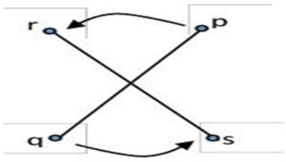

Figure 6: Using cross product on the coordinates.

Topology 1:

Result 1: Information of a network with 5 nodes

Nodes		Connected to
N0	(100.00, 100.00)	1 2 3 4
N1	(200.00, 100.00)	0 2 3 4
N2	(200.00, 200.00)	0 1 3 4
N3	(100.00, 200.00)	0 1 2 4
N4	(45.00, 45.00)	0 1 2 3

After eliminating crossing edges in the graph, the remaining edges are:

(v01, v02, v03, v04, v12, v14, v23 and v34)

For the above topology, G=(V, E)

V={0, 1, 2, 3, 4} and

E={(0,1)(0,2)(0,3)(0,4)(1,2)(1,4)(2,3) (3,4)}

Topology 2:

Result 2: Information of a network with 8 nodes

Nodes	Connected to
N0(100.00, 100.00)	1 2 3 4 5 6 7
N1(200.00, 100.00)	0 2 3 4 5 6 7
N2(200.00, 200.00)	0 1 3 4 5 6 7
N3(100.00, 200.00)	0 1 2 4 5 6 7
N4(45.00, 45.00)	0 1 2 3 5 6 7
N5(20.00, 140.00)	0 1 2 3 4 6 7
N6(120.00, 20.00)	0 1 2 3 4 5 7
N7(200.00, 60.00)	0 1 2 3 4 5 6

After eliminating crossing edges in the graph, the remaining edges are:

(*v*01, *v*02, *v*03, *v*04, *v*05, *v*06, *v*07, *v*12, *v*17, *v*23, *v*35, *v*45, *v*46, and *v*67)

For the above topology, G=(V, E)

V={0, 1, 2, 3, 4, 5, 6, 7} and

E={(0,1)(0,2)(0,3)(0,4)(0,5)(0,6)(0,7) (1,2)(1,7)(2,3)(3,5)(4,5)(4,6)(6,7)}.

Topology 3:

Figure 8 is a network of 8 nodes similar to Figure 7. Here node 2 is moved out of the sensing range and thus disconnects itself with the rest of the nodes, i.e., node 2 is not a one-hop neighbor to any of the remaining nodes. Thus edge (1 3) is considered because it no more intersects edge (0 2) as in Figure 7. The positional information and the final result of non-intersecting edges are tabulated in result 3 (Figure 9).

Result 2: Information of a network with 8 nodes in which node 2 is out of the transmission range

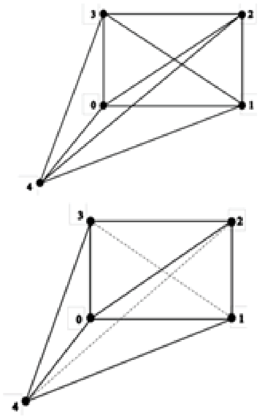

Figure 7: Topology depicted before and after removing the intersecting edges.

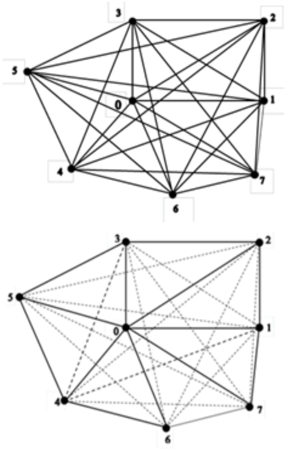

Figure 8: Topology depicted before and after removing the intersecting edges.

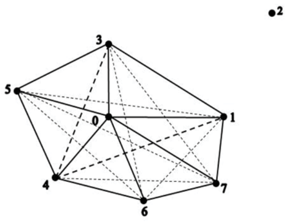

Figure 9: Scenario with node 2 moved beyond the sensing range.

Nodes	Connected to
N0(100.00, 100.00)	1 3 4 5 6 7
N1(200.00, 100.00)	0 3 4 5 6 7
N2(350.00, 350.00)	
N3(100.00, 200.00)	0 1 4 5 6 7
N4(0.00, 0.00)	0 1 3 5 6 7
N5(20.00, 140.00)	0 1 3 4 6 7
N6(120.00, 20.00)	0 1 3 4 5 7
N7(200.00, 60.00)	0 1 3 4 5 6

After eliminating crossing edges in the graph, the remaining edges are:

(v01, v03, v04, v05, v06, v07, v13, v17, v35, v45, v46, and v67)

For the above topology, $G=(V, E)$

$V=\{0, 1, 2, 3, 4, 5, 6, 7\}$ and

$E=\{(0,1)(0,3)(0,4)(0,5)(0,6)(0,7)(1,3)\ (1,7)(3,5)(4,5)(4,6)(6,7)\}$

Conclusion

WSN can be modelled as a graph. A graph is said to be connected if the sink and the entire sensor nodes are connected at any given point of time throughout the network's lifetime. A WSN thus reduced on a graph exhibits the properties of a nonplanar graph. Any nonplanar graph can be reduced to a planar graph by carefully removing the crossing edges. The reduced planar graph is also connected and preserves the original topological structure of WSN. Coordinate geometry over line segments COL plays a vital role in reducing a complex graph to a simple one. Our algorithm scales up with the topology and provides the most efficient solution.

Future Work

The graphs reduced from nonplanar to planar don't have a unique solution. Solutions depend on the node sourcing the data to the sink. For e.g., if a network consists of *n* nodes, then there are *n* different solutions. These different solutions, or paths can be further utilized to load balance a given network. Load balancing is a concept used to increase the capacity and reliability of concurrent applications by distributing the load on different systems (nodes) available, rather than burdening the server. The experiment can also be conducted on mobile nodes and their impact on the algorithm can be recorded.

Acknowledgements

I would like to thank my institution Sir M. Visvesvaraya Institute of Technology for providing me an opportunity to carry out my research work under the Department of Computer Science and Engineering Research Center. My sincere thanks to Prof. Kini K. R., Principal and Prof. Dilip K. Sen, HOD of CSE for their constant support and encouragement.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks a survey. Computer Networks 38: 393-422.
2. Sultan A, Merabti M, Askwith B, Kifayat K (2009) Network connectivity in wireless sensor networks: a survey.
3. Tamboli N, Younis M (2009) Coverage-aware connectivity restoration in mobile sensor networks. Journal of Network and Computer Applications.
4. Imran M, Younis M, Said M, Hasbullah H (2010) Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks.
5. Imran M, Younis M, Said M, Hasbullah H (2012) Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks. Journal of Network and Computer Applications 35: 844-856.
6. Saber RO, Murray RA (2003) Problems in networks with directed graphs and switching.
7. Mesbahi M (2002) On a dynamic extension of the theory of graphs.
8. Scribd (2018) Graph theory IP.
9. Antiagingsurvival (2018) Start the checklist.
10. Handa H (2017) Role of NGO'S and corporate social responsibility in protection of environment.
11. Muhammad A, Egerstedt M (2005) Connectivity graphs as models of local interactions. Mathematics and Computation 168: 43-269.
12. Wang L, Wu X (2014) Distributed prevention mechanism for network partitioning in wireless sensor networks. Journal of Communications and Networks 16: 1-6.
13. Purdue (2018) Undergraduate curricular practical training.

14. Basagni S, Marco C, Silvia G, Ivan S (2004) Mobile ad hoc networking.
15. Abbasi AA, Akkaya K, Younis M (2007) A distributed connectivity restoration algorithm in wireless sensor and actor networks.
16. Basu P, Redi J (2004) Movement control algorithms for realization of fault tolerant ad hoc robot networks. IEEE Network 18: 36-44.
17. Wu C, Yang Z, Liu Y, Xi W (2012) WILL: Wireless indoor localization without site survey. IEEE Infocom 24: 64-72.
18. Chen H, Shi Q, Tan R, Sezaki K (2010) Mobile element assisted cooperative localization for wireless sensor networks with obstacles. IEEE Transactions on Wireless Communications 9: 956-963.
19. Hauspie M, Carle J, Simplot D (2003) Partition detection in mobile ad-hoc networks using multiple disjoint paths set. International Workshop on Objects Models and Multimedia Technologies.
20. Ritter H, Winter R, Schiller J (2004) A partition detection system for mobile ad-hoc networks.
21. Jorgic M (2007) Localized detection of k-Connectivity in wireless ad hoc, actuator and sensor networks.
22. Cardei M, Yang S, Wu J (2008) Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks. IEEE Transactions on Parallel and Distributed Systems 19: 545-558.
23. Gao L, Nelson J (2010) An algorithm for k-connectivity topology in heterogeneous wireless sensor networks.
24. Bai X, Xuan D, Yun Z, Ten H, Lai L (2008) Complete optimal deployment patterns for full-coverage and k-connectivity (k _ 6) wireless sensor networks. IEEE/ACM Transactions on Networking 18: 401-410.
25. Wang Y, Li K, Xu Q (2007) A distributed topology control algorithm for k-connected dominating set in wireless sensor networks.
26. Du J, Li X, Sun X, Zheng R (2012) Application-oriented fault detection and recovery algorithm for wireless sensor and actor networks. International Journal of Distributed Sensor Networks 12: 01-10.
27. Wang YC, Hu CC, Tseng YC (2005) Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks.
28. Jinko K, Selmc RR, Phova V (2009) Detecting coverage holes in wireless sensor networks.
29. Ghrist R, Muhammad A (2005) Coverage and hole-detection in sensor networks via homology.