doi: 10.20532/cit.2016.1002674

Construction of Pipelined Strategic Connected Dominating Set for Mobile Ad Hoc Networks

Ceronmani Sharmila[1] and George Amalanathan[2]

[1]Hindustan Institute of Technology and Science, Department of Information Technology, India
[2]Periyar Maniammai University, Department of Mathematics and Computer Applications, India

Efficient routing between nodes is the most important challenge in a Mobile Ad Hoc Network (MANET). A Connected Dominating Set (CDS) acts as a virtual backbone for routing in a MANET. Hence, the construction of CDS based on the need and its application play a vital role in the applications of the MANET. The PipeLined Strategic CDS (PLS-CDS) is constructed based on strategy, dynamic diameter and transmission range. The strategy used for selecting the starting node is any source node in the network, which has its entire destination within a virtual pipelined coverage, instead of the node with maximum connectivity. The other nodes are then selected based on density and velocity. The proposed CDS also utilizes the energy of the nodes in the network in an optimized manner. Simulation results showed that the proposed algorithm is better in terms of size of the CDS and average hop per path length.

ACM CCS (2012) Classification: Networks → Network Protocols;
Networks → Network types → Ad hoc networks → Mobile ad hoc networks

Keywords: Mobile ad hoc networks, connected dominating set, strategy, virtual pipe, transmission range, density, velocity

1. Introduction

Ad hoc network is an infrastructure less network where each node acts as a router or an access point in the network. Each node is also a mobile computing device through which wireless communication is established. Any node can move with an arbitrary velocity, in any direction at any time. Message transmission is achieved either through single-hop (if nodes share their transmission range) or through multi-hop (if nodes do not share their transmission range). The following features of ad hoc networks, such as dynamic topology, interference, multi-hop message transmission, less bandwidth and limitation of resource, make routing a challenging task during the establishment of a connection.

Ad hoc networks have been the focus of research in tactical networks, acoustic networks, sensor networks, social networks, disaster relief, etc. Traditional routing protocols of wired networks cannot be utilized in ad hoc networks, as the nodes are mobile and this allows higher mobility causing rapid changes in the topology. Routing is very difficult in ad hoc networks due to frequent link failure. In contrast, the wired networks have a static topology and the frequency of link failure is considerably low. Mobile Ad hoc Network (MANET) is a temporary network formed by a set of mobile nodes which communicate with each other using wireless communication. In the past, researchers Perkins [1] and Toh [2] proposed routing protocols for MANET and these protocols were classified based on their performance and are outlined below.

The first generation of routing protocol was formed based on data collection. The protocols under this category were reactive and proactive routing protocols. In reactive routing protocols, as pointed by Bhaskar et al. [3], routes were established between source and destination only when there arose a need to transmit the message. They were further categorized as Ad hoc On-demand Distance Vector (AODV) routing protocol as discussed by Perkins and Royer [4] and Xu et al. [5], and Dynamic Source Routing (DSR) protocol as given in [6], [7]. In the proactive routing protocols routes were maintained up-to-date from one node to another node in the network. They were further categorized as Destination Sequence Distance Vector (DSDV) routing protocol according to [8], [9] and Optimized Link State Routing (OLSR) protocol as pointed out by Baayer et al. [10]. These flood-based protocols required a network with large bandwidth and energy to establish communication within the network.

The second generation of routing protocol was formed based on coordinator based identifier. These identifiers were used to locate nodes and perform searching and routing task. The protocols under this category are Beacon Vector Routing (BVR) protocol as given in [11] and Greedy Perimeter Stateless Routing (GPSR) protocol as discussed in [12]. These geographic routing-based protocols were found to be efficient in establishing communication within the network.

The third generation of routing protocol was formed based on Connected Dominating Set (CDS). A CDS acts as a virtual backbone of the network. More research work in this area showed that the construction of CDS could be categorized as Centralized algorithms as given in [13] and Decentralized algorithms. The Decentralized algorithms were then classified as Localized algorithm, as projected in [14], and Distributed algorithm, as discussed in [15]. The centralized CDS approach obtained a CDS with small size having a better performance ratio. But this approach cannot be used for real time applications because, the topology of the entire network should be known in advance. The localized CDS approach was found to be good at maintaining the CDS formed, but required a constant number of communication rounds. The distributed CDS approach was further categorized as Prune-based and Maximal Independent Set (MIS)-based, as discussed in [16], was more practical for a dense network.

The performance of CDS-based routing algorithms was better compared to other protocols because the message was broadcasted only within a smaller sized backbone network to reduce the retransmission of messages. Hence, the main objective of this work is to construct a strategic connected dominating set using a virtual pipe in a mobile ad hoc network. This virtual pipe will be used for routing from the starting node (designated as source) to various destinations within the pipe. Once the CDS is constructed, it can be used for routing between any source to any destination where the searching area for a route in the network is reduced to a smaller set.

This paper is organized as follows: Section 2 gives the details of related work. Section 3 describes the pipelined strategic connected dominating set algorithm. Section 4 gives the performance metrics. Section 5 focuses on conclusion and future enhancement.

2. Related Work

A MANET can be characterized by a graph *G* (*V*, *E*), where *V* represents the vertices such as nodes in the graph and E represents the time varying edges, such as communication links between the vertices in the graph. These communication links exist when any two nodes are within the same transmission range *R*. If a graph is a tree, then the number of edges is given by *V*-1, V is the number of vertices in the graph. Flooding is the method mostly used for broadcasting in MANET. But flooding incurs overhead due to redundant transmissions. To avoid the above problem, the researchers proposed a Connected Dominating Set approach using the concepts of graph theory. This approach constructed a virtual backbone network, which proved to be a better network in most of the applications.

Guha and Khuller [17] proposed two centralized polynomial time algorithms for constructing a CDS in a graph. In the first algorithm, a greedy CDS was constructed for solving the problem. In the second algorithm, the first algorithm was improved by finding a dominating set in the first phase and constructing a CDS in the second phase. The parameter used for evaluation was performance guarantee. Ruan et al. [18] proposed a greedy centralized approximation algorithm for constructing a CDS. In this algorithm, it was proved that the greedy potential solution produced an approximate result for minimum CDS. The parameter used for evaluation was maximum degree.

Alzoubi et al. [19] proposed a localized approximation algorithm for constructing a Minimum CDS (MCDS) in a unit disk graph. In this algorithm, an algorithm was developed which maintained the same approximation ratio even after the change in topology. The parameters used for evaluation were CDS size, time complexity and message complexity. Li et al. [20] proposed a completely localized algorithm for constructing a CDS with constant performance ratio. A CDS constructed without a spanning tree or leader election was smaller in size. The parameter used for evaluation was the size of the CDS.

Therefore, the distributed CDS approach is the best for MANET and sensor networks as the above networks lack centralized administration. Kim et al. [21] proposed two centralized algorithms for constructing a CDS with constant performance ratio. Finally, a distributed version of the above algorithms was developed which could be implemented in real circumstances. The CDS developed utilized minimum energy so as to extend the lifetime of the network. The parameters used for evaluation were the size of the CDS, diameter and average backbone path length. Ding et al. [22] proposed an efficient distributed algorithm for constructing a CDS with minimum routing cost. Here, a CDS was developed such that the routing path between any pair of nodes was the shortest path in the network. The parameters used for evaluation were the maximum routing path length and the average routing path length.

Huang et al. [23] proposed three approximation algorithms for constructing Minimum Latency Broadcast Scheduling (MLBS) algorithm in wireless ad hoc networks. In MLBS, the radius of the network was assigned as the lower bound value, and the maximum distance of all the nodes from the broadcasting source as the upper bound value. The broadcasting proceeded in synchronous time-slots. The parameter used for evaluation was latency.

De Souza et al. [24] proposed pipelined scheduling for sensor networks. The schedule was constructed before determining the data aggregation tree with a condition that every node could reach the sink. The parameters used for evaluation were scheduled length, aggregate throughput, collection delay and maximum degree.

Meghanathan [25] proposed a Maximum Density-based CDS (MaxD-CDS) algorithm for Mobile Ad hoc Networks (MANET). This algorithm generates a Minimum Connected Dominating Set (MCDS) in a static graph. The Max-D-CDS algorithm selects the nodes with larger number of uncovered neighbors for constructing the CDS. The construction of the Max-D-CDS starts with the inclusion of the node having the highest density (uncovered nodes) into the CDS. Once a node is added to the CDS, all its neighbours are said to be covered. The node with the highest density in the covered list is selected as the next node for inclusion into the CDS. This procedure is repeated until all the nodes in the network are covered. The parameters used for evaluation were CDS node size and CDS edge size. Simulation results showed that the proposed algorithm achieved an average number of nodes and edges in the CDS.

Ceronmani and George [26] have proposed four Strategic CDS algorithms for mobile ad hoc networks. The strategy used in all the algorithms was to select any node in the network as the starting node for constructing the CDS, providing the selected node as a specific application and need. The four algorithms proposed were Strategic Maximum Density CDS (ST-MaxD CDS), Strategic Strong Neighbourhood CDS (ST-SN CDS), Strategic Minimum Velocity Minimum Density CDS (ST-MV-MD CDS) and Strategic Minimum Density Minimum Velocity CDS (ST-MD-MV CDS). The construction of the above algorithms starts with inclusion of any node in the network into the CDS. Once a node is added to the CDS, all its neighbours are said to be covered. The node with highest density or strong neighbourhood or minimum velocity (in case of a tie a node with minimum density) or minimum density (in case of a tie a node with minimum velocity) in the covered list is selected as the next node for inclusion into the CDS. This procedure is repeated until all the nodes in the network are covered. The parameters used for evaluation were CDS node size, CDS edge size and CDS circuit size. When comparing the four algorithms, ST-MD-MV CDS has a moderate number of CDS nodes, less number of CDS edges and no circuits.

3. PipeLined Strategic Connected Dominating Set (PLS-CDS)

The PipeLined Strategic Connected Dominating Set (PLS-CDS) finds a minimal CDS in a MANET in a distributed manner. The proposed algorithm finds a virtual pipelined based strategic connected dominating set which can be used as a backbone for a mobile ad hoc network. It does not require a predefined routing mechanism and it is totally distributed. The basic application of the algorithm is to select the starting node based on the existence of virtual pipe in the network. The strategy used in all the algorithms was to select any node in the network as the starting node for costructing the CDS, providing the selected node as a specific application and need. The other nodes are selected based on density and velocity. The proposed algorithm is better in terms of CDS node size and average hop count per path.

The mobile ad hoc network area is considered as a virtual pipe. The selection of starting node and existence of virtual pipe is explained through the following lemma.

Lemma 1. There exists a virtual pipe in the mobile ad hoc network (as shown in Figure 1) under the following assumptions:

(i) The starting point of the virtual pipe is the vertex position of S_n, where S_n is the starting node which has received a specific source of information from any external network.

(ii) The upper bound and lower bound distances from S_n in the vertical direction of the virtual pipe are obtained by

$$D_u = \frac{1}{2}(R) + D_A\{U(t)\} \tag{1}$$

$$D_d = -\left[\frac{1}{2}(R) + D_B\{V(t)\}\right] \tag{2}$$

The breadth of the virtual pipe in the vertical direction (denoted as br) is given by

$$br = R + D_A\{U(t)\} + D_B\{V(t)\} \tag{3}$$

where R is the transmission range and D_A and D_B are the distances covered by the node per unit time at a particular velocity in the upper bound and lower bound respectively.

(iii) If velocities of the nodes are varying drastically, then D_A and D_B are chosen as

$$D_A = \max\{u_1(t),\ u_2(t)\ldots u_i(t)\} \tag{4}$$

$$D_B = \max\{v_1(t), v_2(t)\ldots v_j(t)\} \tag{5}$$

where i and j are the numbers of edge nodes (nodes in the circumference of the transmission range of S_n) in the upper bound and lower bound of the virtual pipe.

(iv) The length of the virtual pipe from S_n in the horizontal direction (denoted as ln)

$$ln = d_y d + D_C\{X(t)\} \tag{6}$$

where $d_y d$ is the dynamic diameter of the virtual pipe and D_C is the distance covered by the node per time unit at a particular velocity in the horizontal direction of the virtual pipe.

(v) If velocities of the nodes are varying drastically, then D_C is chosen as

$$D_C = \max\{x_1(t),\ x_2(t)\ldots x_k(t)\} \tag{7}$$

where k is the number of edge nodes (nodes in the circumference of the diameter coverage) in the horizontal direction of the virtual pipe (towards the end of the pipe).

(vi) The distance in the horizontal direction is always greater than the distance in the vertical direction of the virtual pipe.

$$ln > br \tag{8}$$

(vii) The number of nodes located inside the virtual pipe will be greater than a fixed minimum.

For example (as shown in Figure 2), if the snapshot of the mobile ad hoc network is considered as a hex derived network, as discussed by Ceronmani and George [27], then ln and br are given by

$$ln = (2n-2) + x \tag{9}$$

$$br = R + u + v \tag{10}$$

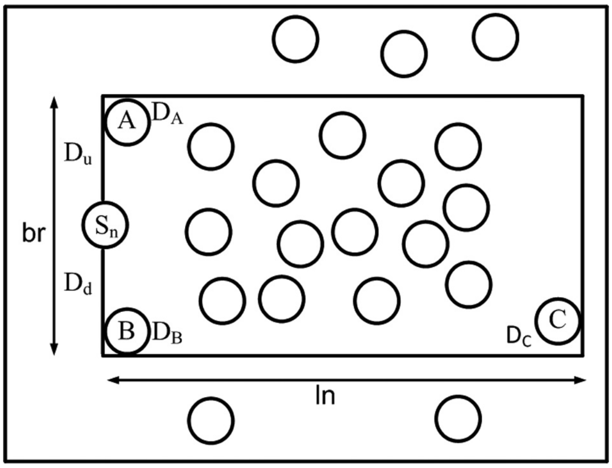


Figure 1. Pipelined structure for CDS.

where n is the number of nodes located inside the virtual pipe, x is the maximum velocity of an edge node in the horizontal direction of the pipe and u and v are the maximum velocities of an edge node in the upper bound and lower bound of the pipe.

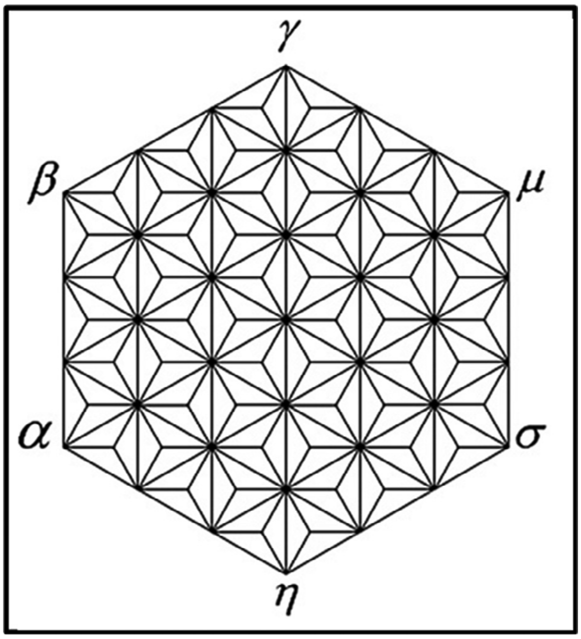


Figure 2. Hex derived network.

3.1. Data Structures

The following data structures are used in the PLS-CDS algorithm.

(i) *Node-Velocity-Function:* It is a function which has a range of velocities from which the nodes in the network select the velocities randomly.

(ii) *CDS-Node-List:* It is a list which includes all the nodes that are selected in each round of the PLS-CDS algorithm.

(iii) *Covered-Node-List:* It is a list which includes the nodes that are in CDS-Node-List and the nodes covered by the CDS nodes.

(iv) *CDS-Edge-List:* It is a list which includes the edges that exist between any two CDS nodes in the CDS-Node-List.

3.2. Algorithm for Constructing PLS-CDS

In this paper, networks in which nodes may have different transmission range are studied. Density, velocity, etc. are not used for selecting the starting node. In the literature proposed by Meghanathan [25], a node with maximum density was selected as the starting node. But in the proposed algorithm a node with specific application and need was selected as the starting node and the other nodes were selected based on density and velocity.

The input to the algorithm is a sample network as shown in Figure 3(a). This snapshot is considered as a graph $G=(V, E)$, where each node is assigned a velocity from the *Node-Velocity-Function*.

The general steps for PLS-CDS algorithm are:

Step 1 Select a node S_n from the network, which has specific information as the start node.

Step 2 Apply Lemma 1 and add all the nodes which are inside the virtual pipe into the *Uncovered-Node-List* (including S_n).

Step 3 Add S_n to *CDS-Node-List* and *Covered-Node-List*.

Step 4 Add the entire one hop neighbours of S_n to *Covered-Node-List*.

Step 5 Remove S_n and entire one hop neighbours of S_n from *Uncovered-Node-List*.

Step 6 Change the density of S_n to 0.

Step 7 Compute the density of covered nodes of S_n.

Step 8 Choose a node from the *Covered-Node-List* which satisfies the next node criterion (as described below).

Step 9 Add the node chosen from step 8 to *CDS-Node-List* and *Covered-Node-List*.

Step 10 Add the entire one hop neighbours of chosen node (from step 8) to *Covered-Node-List*.

Step 11 Remove the chosen node and entire one hop neighbours of the node chosen (in step 8) from *Uncovered-Node-List*.

Step 12 Change the density of the chosen node to 0.

Step 13 Compute the density of all the nodes in the network (except for the nodes whose density is 0 and CDS nodes).

Step 14 Go to step 8.

Step 15 Stop when the *Uncovered-Node-List* is empty.

Step 16 Compute *CDS-Edge-List*.

The next node criteria are as follows:

(i) Do not choose a node from *CDS-Node-List*.

(ii) Choose a node from *Covered-Node-List*.

(iii) Choose a node which has at least one uncovered neighbor.

(iv) Choose a node which has a minimum density (greater than one). If there is a tie, choose a node with minimum velocity.

(v) Otherwise, choose a node randomly.

The PLS-CDS algorithm is illustrated below with an example. The symbols which are used for identifying the nodes in the above figures are as follows:

(i) Nodes which are represented by dotted circles with a plus symbol are designated as CDS nodes.

(ii) Nodes which are represented by dotted circles are designated as covered nodes.

(iii) Nodes which are represented by circles are designated as uncovered nodes.

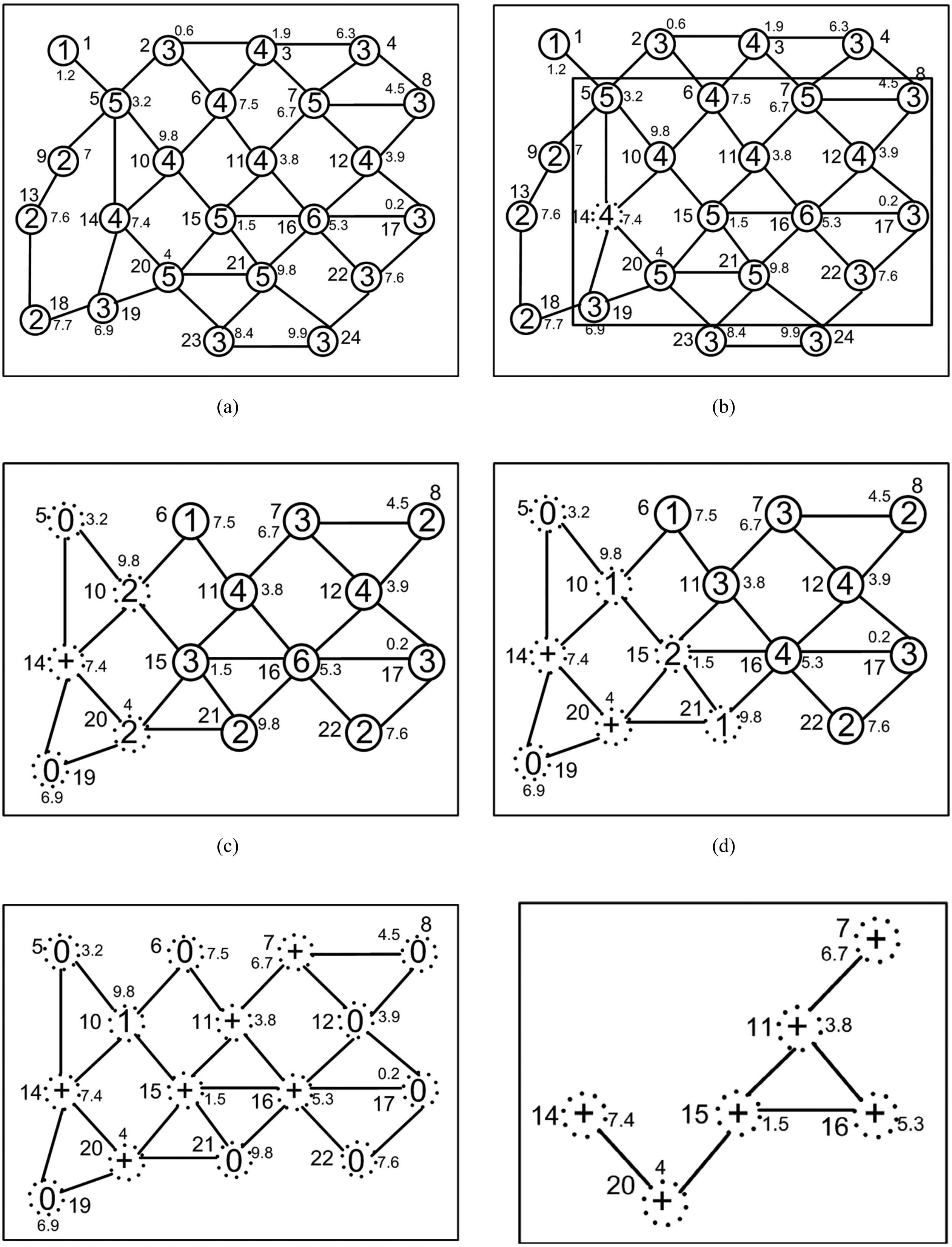
The number inside the circle represents the density of the node and the number outside the circle represents the identity of the node. The decimal value outside the circle represents the velocity of the node which is randomly assigned from the *Node-Velocity-Function*.

Any node in the initial sample network (Figure 3(a)) can be selected as the starting node, provided that it has a specific source of information from other network and needs to be broadcasted. For example, node 14 is selected and a pipeline is drawn using Lemma 1 with node 14 as the origin (as shown in Figure 3(b)). Include the nodes which are inside the virtual pipe into the *Uncovered-Node-List*.

The nodes which are outside the virtual pipe such as nodes 1, 2, 3, 4, 9, 13, 18, 23 and 24

Pseudo Code for constructing PLS-CDS

```
Begin
      for i to n
            if (node(i).message != NULL and node(i).diameter = $d_y d$)
            begin
                  strategic_node = node(i);
                  construct_PLS_CDS (strategic_node);
            end
   construct_PLS_CDS(node)
   begin
      if (node = = NULL)
            return;
      else
      begin
            find_all_neighbour_node (node);
            add neighbour_node with min_density to CDS;
            if (there is more than one neighbour with min_density)
                  add neighbour_node with min_density and min_velocity to CDS;
      end
      construct_PLS_CDS (neighbour_node);
   end
End
```



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3. Example to construct PLS-CDS, (a) Initial sample network graph, (b) Pipelined Structure for CDS, (c) Round # 1, (d) Round # 2, (e) Round # 6 (Last Round), (f) CDS nodes and CDS edges.

are not included into the *Uncovered-Node-List*. Then, having this list as the base, the CDS is constructed.

Node 14 is included into the *CDS-Node-List* and *Covered-Node-List*. Include all the one hop neighbours of node 14 such as nodes 5, 10, 19 and 20 into the *Covered-Node-List*. Exclude node 14 and nodes 5, 10, 19 and 20 from *Uncovered-Node-List*. The density of node 14 is changed to 0 (since it is a CDS node, it is represented by a circle with a plus symbol inside it). The density of all the nodes is recomputed again, such that only the nodes which are uncovered are aggregated as density (as shown in Figure 3(c)).

Using the next node criterion, the node for the next round is chosen from the *Covered-Node-List*. In the example, nodes 10 and 20 have minimum density value, but node 20 is selected because of low velocity when compared to node 10 (as shown in Figure 3(d)).

The process is repeated until *Uncovered-Node-List* is empty and all the nodes are moved into *Covered-Node-List* (as shown in Figure 3(e)). The number of CDS nodes in *CDS-Node-List* is 6 and the number of CDS edges in *CDS-Edge-List* is 6 (as shown in Figure 3(f)). The final CDS list has one circuit.

4. Simulations and Analysis

We present simulations that illustrate the results of the algorithm and analyze the behaviour of the algorithm in various scenarios. The proposed algorithm was implemented on an ad hoc network by placing nodes randomly in a 150 m × 150 m area. Nodes were connected with each other if they shared their transmission range *R*. The transmission range used during the simulation is 75 meters. The Random Waypoint Mobility model was used for simulating the model in MANETs. In this model each node is assigned a random speed in the range of [5, 10, 15, 20, 25, 30, 35, 40, 45, 50] m/s. Initially a low mobility network is created by choosing the maximum speed limit as 10 m/s and at a later stage a high mobility network is created by choosing the maximum speed limit as 50 m/s for simulation. A homogeneous network was created by having the same transmission ranges of the nodes in the network. The PLS-CDS algorithm was tested in the Visual C++ express tool. The network was generated with sizes of 100~250 nodes respectively. The Breadth First Algorithm (BFS) was used to check the connectedness of the constructed CDS.

The PLS-CDS has been evaluated with respect to the three metrics: CDS node size, CDS edge size and Average Hop per Path Length (AHPL). The average size of the virtual pipe has reduced the network area to 82 M × 142 M. The comparisons of PLS-CDS with existing algorithms MAXD-CDS and ST-MD-MV-CDS are shown in Figure 4 through Figure 9. The PLS-CDS is best in terms of all the parameters. The average value of 15 simulation results was used to estimate the parameter values. In the previous research work, it was found that if the numbers of CDS nodes were less, then the average hop count per path was more. But in PLS-CDS this has been disproved, as the AHPL is less when compared to other algorithms. This makes the source to reach the destination with the least number of hops. The smaller number of CDS nodes reduces the overhead in broadcasting the messages within the network.

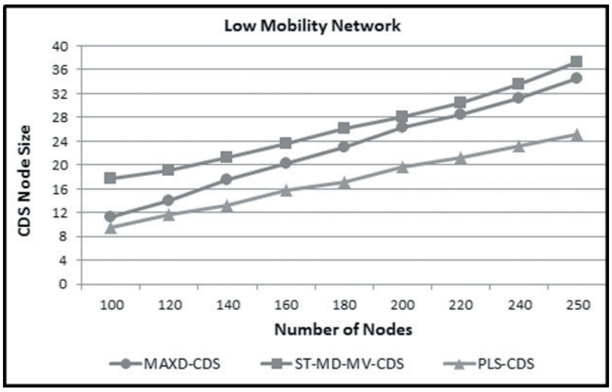


Figure 4. CDS Node Size in low mobility network.

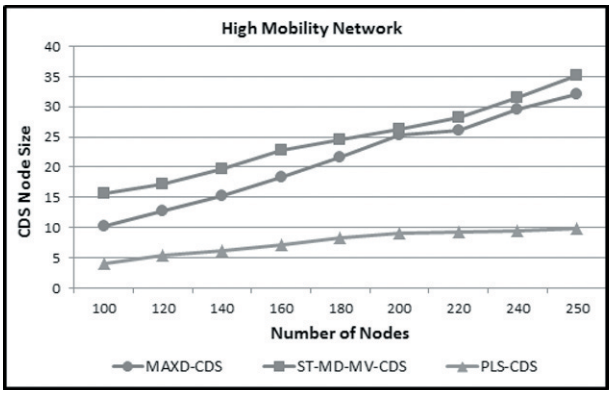


Figure 5. CDS Node Size in high mobility network

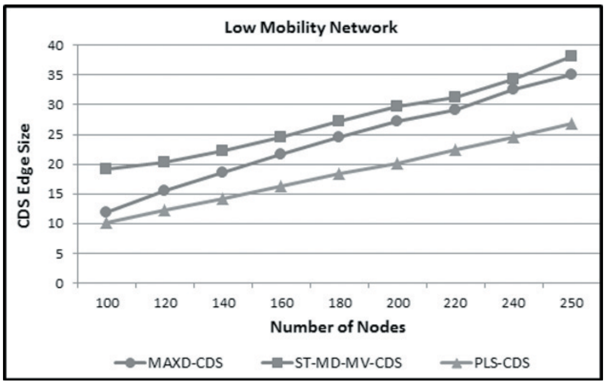


Figure 6. CDS Edge Size in low mobility network.

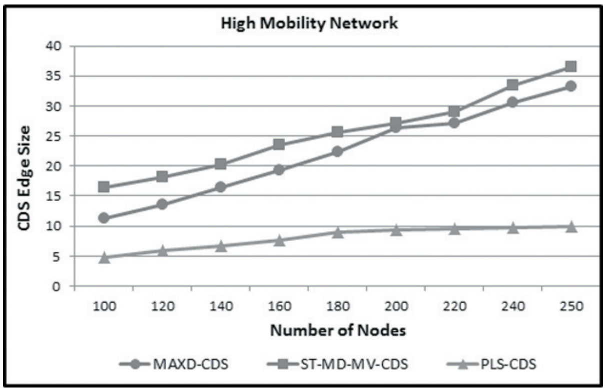


Figure 7. CDS Edge Size in high mobility network.

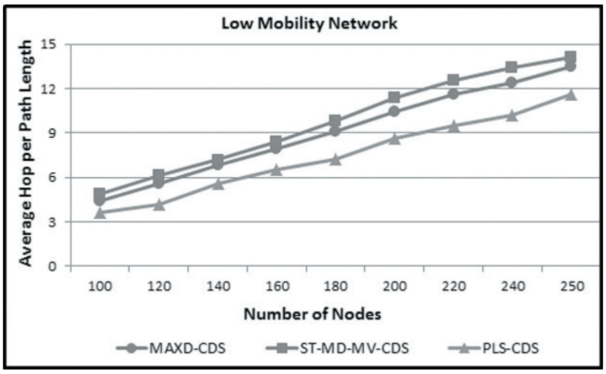


Figure 8. AHPL in low mobility network.

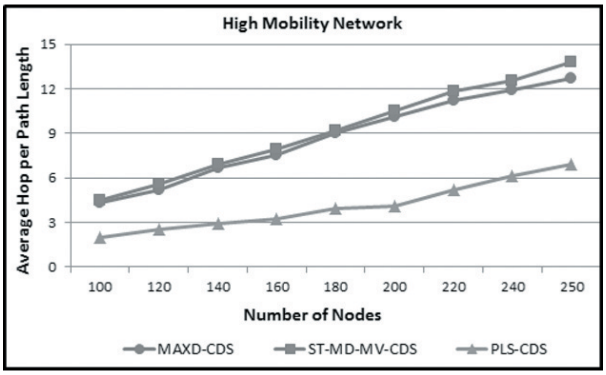
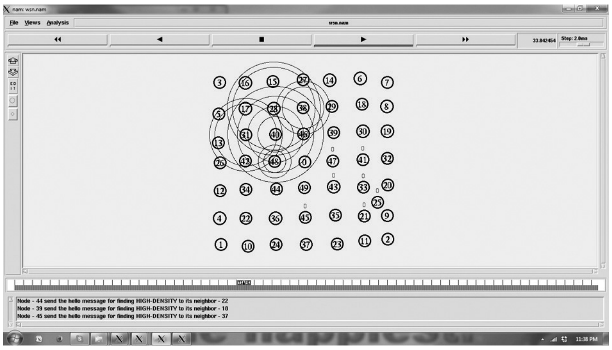


Figure 9. AHPL in high mobility network.

But during high mobility, it is observed that the nodes move very fast and they cross the virtual pipe. So the number of CDS nodes and CDS edges are found to be minimum in the high mobility network. Thus, it also reduces the average hop per path length. While choosing the starting nodes, care was taken that different nodes were selected each time during the 15 trials. It was observed that if a node with very low density was selected as the starting node, the number of CDS nodes was by 0.2% larger when compared to the node with moderate density.

Simulation of the proposed algorithm was also implemented in NS2 to estimate the energy utilization of the nodes in the virtual pipe. During the simulation 15 trace files for mobility were used. The covered nodes are represented by a circle with a number 0 (as shown in Figure 10).



```
Node - 44 send the hello message for finding HIGH-DENSITY to its neighbor - 22
Node - 39 send the hello message for finding HIGH-DENSITY to its neighbor - 18
Node - 45 send the hello message for finding HIGH-DENSITY to its neighbor - 37
```

Figure 10. Snapshot of simulation run in NS2.

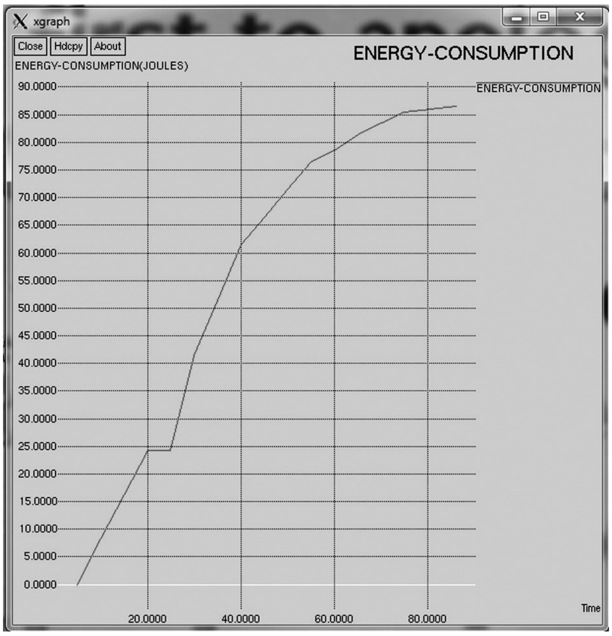


Figure 11. Energy utilization for constructing the CDS.

The energy utilization for constructing the PLS-CDS with 50 nodes was 85 Joules (as shown in Figure 11). The simulation parameters are listed in Table 1.

Table 1. Simulation parameters.

Simulator	Ns2 (version 2.28)
Simulation area	1000 × 1000 m
Propagation	Two-ray ground reflection
MAC protocol	IEEE 802.11
Bandwidth	2 Mbps
Traffic	CBR
Transmission range	25 meters
Number of nodes	100~250 nodes
Maximum speed	5~20 m/sec
Mobility model	Random Waypoint
Broadcast sessions	35
Broadcast rate	0.5~5 pkts/s
Message size	128 bytes
Hello interval	2 secs
Simulation time	60~100 minutes
Number of trials	15

The energy utilization of PLS-CDS algorithm was compared with MAXD-CDS and ST-M-D-MV-CDS algorithms as shown in Figure 12. The PLS-CDS algorithm utilizes less energy for CDS formation when compared to other algorithms. As the node count increases from 50 to 250, the energy utilization increases drastically for MAXD-CDS and ST-MD-MV-CDS algorithms. But in case of PLS-CDS algorithm, the energy utilization increases gradually, as virtual pipe has been used for CDS formation.

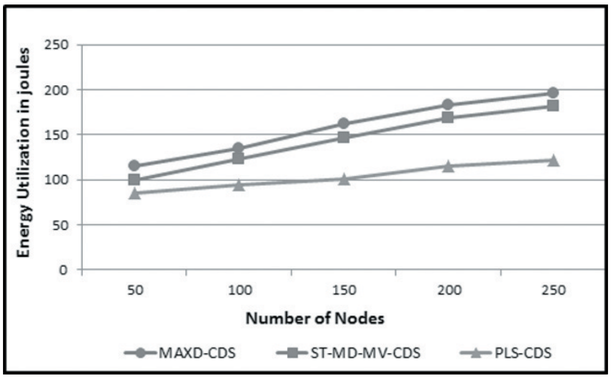


Figure 12. Energy utilization comparison.

5. Conclusion

It can be concluded from the study that the virtual pipe based strategic connected dominating set appears to be more suitable for low mobility network. This PLS-CDS algorithm helps to discover the routes easily. The algorithm has improved the performance factors such as the CDS node size, CDS edge size and AHPL. The energy utilization is less because, nodes with minimum density and minimum velocity are chosen during the construction of virtual pipelined based CDS. This algorithm can be improved further by constructing a virtual pipe in a heterogeneous network. The impact on the high mobility network can also be studied further.

References

[1] C. E. Perkins, *Ad Hoc Networking*, 2nd Ed. Boston: Addison-Wesley, 2001.

[2] C. K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, Ed. NJ: Prentice Hall PTR, Upper Saddle River, 2001.

[3] R. Bhaskar *et al.*, "Efficient authentication for reactive routing protocols", in *Proc. of 20th IEEE Int. Conf. on Advanced Inform. Networking and Applications*, Washington, DC, USA, 2006, pp. 57–61.
http://dx.doi.org/10.1109/AINA.2006.162

[4] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", in *Proc. of 2nd IEEE Workshop on Mobile Computing Syst. and Applications*, New Orleans, LA, 1999, pp. 90–100.
http://dx.doi.org/10.1109/MCSA.1999.749281

[5] S. Xu *et al.*, "Authenticated AODV routing protocol using one-time signature and transitive signature schemes", *J. Networks*, vol. 1, no. 1, pp. 47–53, 2006.
http://dx.doi.org/10.4304/jnw.1.1.47-53

[6] D. B. Johnson *et al.*, *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad hoc Networks*, Ed. Boston: Ad hoc networking, 2001, pp. 139–172.
http://dx.doi.org/10.17487/rfc4728

[7] M. Alilou and M. Dehghan, "Upgrading performance of DSR routing protocol in mobile ad hoc networks", *Int. J. Elect. Comput. Energetic, Electron. and Commun. Eng.*, vol. 1, no. 5, pp. 762 –764, 2007.

[8] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector routing (DSDV) for Mobile Computer", in *ACM Sig-*

comm 94: Proc. of the Conf. on Commun. Architecture, Protocols and Applications, NY, USA, 1994, pp. 234–244.
http://dx.doi.org/10.1145/190809.190336

[9] K. Khan *et al.*, “An efficient DSDV routing protocol for wireless mobile ad hoc networks and its performance comparison”, in *Proc of 2nd IEEE UKSIM European Symp. on Comput. Modeling and Simulation*, Washington, DC, USA, 2008, pp. 506–511.
http://dx.doi.org/10.1109/EMS.2008.11

[10] A. Baayer *et al.* “A new criterion for MPR selection in OLSR”, in *Proc. of the ACM 2007 Euro American Conf. on Telematics and Inform. Syst.*, NY, USA, 2007.
http://dx.doi.org/10.1145/1352694.1352739

[11] R. Fonseca *et al.*, “Beacon vector routing: Scalable point-to-point routing in wireless sensornets”, in *Proc. of 2nd Symp. on Networked Syst. Design and Implemenation*, 2005, pp. 329–342.

[12] B. Karp and H. T. Kung, “GPSR: Geedy Perimeter Stateless Routing for Wireless Networks”, in *Proc. of the 6th ACM Annual Int. Conf. on Mobile computing and Networking*, NY, USA, 2000, pp. 243–254.
http://dx.doi.org/10.1145/345910.345953

[13] B. Das *et al.*, “Routing in Ad Hoc Networks Using a Spine”, in *Proc. of 6th Int. Conf. on Comput. and Commun. Networks*, Las Vegas, NV, 1997, pp. 34–39.
http://dx.doi.org/10.1109/ICCCN.1997.623288

[14] J. Wu and H. Li, “On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks”, in *Proc. of the 3rd Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Commun.*, NY, USA, 1999, pp. 7–14.
http://dx.doi.org/10.1145/313239.313261

[15] J.-H. Lin *et al.*, “A distributed virtual backbone development scheme for ad-hoc wireless networks”, *Wireless Personal Commun.*, vol. 27, no. 3, pp. 215–233, 2003.
http://dx.doi.org/10.1023/B:WIRE.0000010135.75404.23

[16] J. Blum *et al.*, *Combinatorial Optimization*, Connected dominating set in sensor networks and MANETs, Kluwer Academic Publishers, Washington, DC, US, 2004, pp. 329–369.
http://dx.doi.org/10.1007/0-387-23830-1_8

[17] S. Guha and S. Khuller, “Approximation algorithms for connected dominating set”, *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.
http://dx.doi.org/10.1007/PL00009201

[18] L. Ruan *et al.*, “A greedy approximation for minimum connected dominating sets”, *Theoretical Comput. Sci.*, vol. 329, no. 1, pp. 325–330, 2004.
http://dx.doi.org/10.1016/j.tcs.2004.08.013

[19] K. M. Alzoubi *et al.*, “Distributed Heuristics for Connected Dominating Set in Wireless Ad Hoc Networks”, *J. Commun. and Networks*, vol. 4, no. 1, pp. 22–29, 2002.
http://dx.doi.org/10.1109/JCN.2002.6596929

[20] Y. Li *et al.*, “Localized Construction of Connected Dominating Set in Wireless Networks," in *Proc. of US National Science Foundation Int. Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks*, Chicago, USA, 2004.

[21] D. Kim *et al.*, “Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks”, *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 2, pp. 147–157, 2009.
http://dx.doi.org/10.1109/tpds.2008.74

[22] L. Ding *et al.*, "Distributed construction of connected dominating sets with minimum routing cost in wireless networks”, in *Proc. of 30th IEEE Int. Conf. on Distributed Computing Syst.*, Washington, DC, USA, 2010, pp. 448–457.
http://dx.doi.org/10.1109/ICDCS.2010.17

[23] S. C.-H. Huang *et al.*, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *Proc. of 26th IEEE Int. Conf. on Computer Commun.*, Hong Kong, 2007, pp. 733–739.
http://dx.doi.org/10.1109/INFCOM.2007.91

[24] E. De Souza and I. Nikolaidis, "On the application of pipelining in aggregation convergecast scheduling," in *Proc. of 14th IEEE Int. Symp. and Workshops on World of Wireless, Mobile and Multimedia Networks*, Edmonton, Canada, 2013, pp. 1–9.
http://dx.doi.org/10.1109/WoWMoM.2013.6583392

[25] N. Meghanathan, “Graph Theory Algorithms for Mobile Ad hoc networks”, *Informatica*, vol. 36, no. 2, pp. 185–200, 2012.
http://dx.doi.org/10.4018/978-1-4666-0080-5.ch004

[26] C. Sharmila and A. George, “Construction of Strategic Connected Dominating Set for Mobile Ad Hoc Networks”, *J. Comput. Sci.*, vol. 10, no. 2, pp. 285–295, 2014.
http://dx.doi.org/10.3844/jcssp.2014.285.295

[27] C. Sharmila and A. George, “A Survey on area planning for Heterogeneous Networks”, *International Journal on applications of graph theory in wireless ad hoc networks and sensor networks*, vol. 5, no. 1, pp. 11–21, 2013.
http://dx.doi.org/10.5121/jgraphoc.2013.5102

Received: April, 2015
Revised: November, 2015
Accepted: February, 2016

Contact addresses:
Ceronmani Sharmila
Hindustan Institute of Technology and Science
India
Rajiv Gandhi Salai (OMR)
Padur
Kelambakam
Chennai
Tamil Nadu 603103
India
e-mail: csharmila@hindustanuniv.ac.in

George Amalanathan
Department of Mathematics and Computer Applications at
Periyar Maniammai University
Periyar Nagar
Vallam
Thanjavur
Tamil Nadu 613403
India
e-mail: amalanathangeorge@gmail.com

Ceronmani Sharmila joined Hindustan Group of Institutions in 2003 and is currently working as Assistant Professor (SG) in the Information Technology Department. She received her Master's degree from Hindustan College of Engineering, Anna University, Chennai, India in 2007. She is currently pursuing a doctoral program in Hindustan Institute of Technology and Science (formerly Hindustan College of Engineering), Chennai, India. She has three years of industrial experience and thirteen years of teaching experience in engineering colleges, both undergraduate and postgraduate levels. She has 18 international journal/conference publications. Her areas of interest are computer networks, applications of Graph theory in mobile ad hoc networks and very large scale integration (VLSI).

George Amalanathan is a Professor and Head of the Department of Mathematics and Computer Applications at Periyar Maniammai University. He received his PhD degree from IIT Madras. He has more than 20 years of teaching experience in colleges, both undergraduate and postgraduate levels. He is mentoring four research scholars. He has 40 international journal/conference publications and around 700 citations. His research interests are applications of Graph theory in mobile ad hoc networks, fuzzy theory and combinatorial algorithms.