VOL. 10, NO. 14, AUGUST 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ENERGY EFFICIENT MCDS ALGORITHM FOR MANETS BASED ON CONVEX HULLS

Chakradhar P. and P. Yogesh
Department of Information Science and Technology, Anna University, Chennai, India
E-Mail: chakradhari13@gmail.com

ABSTRACT

This paper proposes and analyzes energy efficient Convex Hull based Minimum Connected Dominating Set [CHMCDS] algorithm. The algorithm achieves energy efficiency, bandwidth efficiency, reliability and robustness by reducing redundant rebroadcasts of control packets in the network. The Connected Dominating Set (CDS) is widely used as a virtual backbone or Spine in mobile ad-hoc networks for the purpose of routing and broadcasting. Here the MCDS is based on distributed Convex Hull algorithm and Unit Disk Graph. In this paper we use a quick hull algorithm with incremental sweeping which is more suitable for the MANETs than other algorithms. Mobility and Residual energy of the nodes are considered as parameters in the maintenance of MCDS. The resultant CHMCDS has relatively longer lifetime than other MCDS algorithms. This is a distributed algorithm with a time complexity of *O(nh log n)* and the message complexity *O(nh log n)*, where ‘n’ is the number of nodes in the network and ‘h’ is the number of convex hull nodes in the network. The performance evaluation of this algorithm yields better results in dense networks as well as sparse networks. The cardinality and the performance ratio of this CHMCDS algorithm are optimal compared with existing MCDS algorithms. The simulation results show that the proposed algorithm performs better.

Keywords: broadcast storm problem, convex hull, MANET, minimum connected dominating set (MCDS), mobility, residual energy.

1. INTRODUCTION

Mobile ad-hoc networks appear in a wide variety of applications such as military, disaster relief, surveillance, sensing, and monitoring. Mobile ad hoc network (MANET) is a special kind of wireless network environment. It is different from the traditional wireless networks. Ad-hoc network is a collection of autonomous and arbitrarily located wireless nodes. It is an infrastructure less network with dynamic topology, limited link bandwidth, variation in links, node capabilities and energy-constrained resources. The nodes need to be more intelligent as they have to act as sender, receiver and intermediate forwarder. Because of dynamic topology they frequently advertise the control information using a simple blind flooding. Blind flooding leads to Broadcast Storm Problem [1]. Broadcast Storm refers to the problem of exhausting the limited resources of the network due to excessive transmission of packets in the network. Considering this environment, several routing protocols have been proposed in order to find out and maintain the multi-hop route with reliability.

Generally ad-hoc routing protocols in MANETs are classified into two types namely reactive and proactive. Proactive (table driven) routing protocols like DSDV [12] and OLSR [9] frequently broadcasts routing information in the network to update the tables. In case of on-demand (reactive) ad-hoc routing protocols such as DSR [10] and AODV [11] simple flooding is used in route discovery procedure by broadcasting route request (RREQ) packets to the network. Generally in MANETs flooding is also used for *alarm signals, paging* etc. However reactive routing protocols do not involve in flooding of data packets but flooding of control packets takes place. Simple flooding consumes energy of the nodes and bandwidth of the network, which is called as Broadcast Storm Problem.

In simple flooding a host on receiving a broadcast message for the first time will rebroadcast the message into network. A straight-forward approach of simple flooding in spite of the fact that size of control packets are small, suffers from the Broadcast Storm Problem which leads to following problems [1].

- **Redundant rebroadcasts:** When a mobile host decides to rebroadcast a broadcast message to its neighbours, all its neighbours already have the message.
- **Contention for medium:** After a mobile host broadcasts a message, if many of its neighbours decide to rebroadcast the message, then they contend with each other.
- **Collision:** collisions are more likely to occur due to lack of efficient back-off mechanism.

The effect of the broadcast storm problem can be reduced efficiently and effectively using Connected Dominating Set (CDS). A Connected Dominating Set is constructed assuming the MANET as a graph, generally Unit Disk Graph [2]. CDS is analogous to the backbone network of traditional wired networks. The activity of broadcasting is confined to the nodes of CDS. The Dominating Set (MCDS) and the problem of constructing

MCDS is an NP hard problem and we call the algorithm as convex hull based minimum connected dominating set [CHMCDS] algorithm. Many standard methods exist in the literature to construct the Minimum Connected Dominating Set [3-7]. Each has its own merits and demerits. In this paper we propose a Convex Hull [26] based algorithm to construct the MCDS. The convex hull of a geometric object (polygon) is the smallest convex set containing that object. Since convex hull has smallest perimeter and convex hull nodes are the farthest neighbours, it reduces the complexity in construction of MCDS.

The rest of the paper is organized as follows: The definitions are given in section 2. The existing solutions are reviewed in section 3. The proposed work is discussed in section 4. The performance analysis is provided in section 5. The simulation results are shown in section 6. The conclusion and future work is provided in section 7.

2. PRELIMINARIES

This section contains the basic definitions.

Unit Disk Graph (UDG)

- A unit disk graph is an intersection graph of family of unit disks in the Euclidean plane.
- A graph is a Unit Disk Graph if and only if its vertices can be put in one to one correspondence with equalized circles in a plane in such a way that two vertices are joined by an edge if and only if the corresponding circles intersect.

Dominating set (DS): Dominating Set for a graph G = (V, E) is a subset D of the Vertex Set V such that each vertex *u* €V is either in D or adjacent to some vertex *v* in D. The elements of dominating set are called dominators.

Independent set (IS): Independent Set of a graph G is a subset of the set of vertices such that no two vertices are adjacent in the subset.

Maximal independent set (MIS): Maximal Independent set is an independent set, which is not a subset of any other independent set, i.e. it is a set S such that every edge of the graph has at least one end point not in S and every vertex not in S has at least one neighbour in S.

Connected dominating set (CDS): Given a graph G = (V, E) where V is the set of vertices and E is set of edges or links. Then CDS is a set of vertices with two properties; D is a dominating set in G, D induces a connected sub-graph of G

Minimum connected dominating set (MCDS): A Minimum Connected Dominating Set is a connected dominating set with smallest possible cardinality among all the CDSs of the graph G.

MCDS lifetime: The time taken between creation of MCDS and its disconnection, calculated and averaged over the time.

Convex Hull (CH)

- The convex hull of a finite point set *S = {P}* is the smallest convex polygon that contains *S*. Also, this convex hull has the smallest area and the smallest perimeter of all convex polygons that contain *S*.
- A set *S* is convex if whenever two points *P* and *Q* are inside *S*, then the whole line segment is also in S.

3. RELATED WORK

3.1 Major solutions

There are various solutions [1, 13] to alleviate this broadcast storm problem. Important of them are Probability based scheme, Area based scheme, Counter based Scheme, Neighbour knowledge based scheme and CDS based scheme. In probabilistic scheme [15, 14] on receiving a broadcast message for the first time, a host will rebroadcast it with probability *P*. This probability is of two types *Static* and *Dynamic.* Static probability scheme [15], generally broadcasts with a constant value initiated by the source of the broadcast. Clearly, when P = 1, this scheme is equivalent to flooding. In Dynamic probability scheme [14] the value of the probability changes based on various parameters such as number of duplicate messages received, neighbour list, density of the network etc. In Area based approach [1, 13], a neighbour is selected as a broadcasting node based on extra area it is going to cover. If a node is not covering new area then it is discarded. In counter-based scheme [1, 13, 16], upon receiving a previously unseen broadcast message, the mobile node initializes a counter with a value of one and set a Random Defer Time (RDT). During this deferring time the counter is incremented by one for each redundant message received. If the counter is less than a predetermined threshold and the deferring time expires, the message will be relayed. Otherwise, it is simply discarded. Neighbour knowledge methods [3, 13, 16] usually utilize the one-hop or two-hop neighbourhood information to reduce redundant transmissions. Neighbourhood information is obtained by periodically exchanging "HELLO" messages among neighbour nodes. In 2-hop information methods [17, 19], backward information is used for minimizing the number of forwarding nodes and reducing the collisions in the network. However none of these schemes is able to form the virtual backbone network.

3.2 Connected Dominating Set (CDS) [3-8, 35]:

CDS belongs to graph based method. It can be used as a virtual backbone or spine of wireless ad-hoc networks. CDS not only provides efficiency in broadcasting but also in multicasting and power management. Our proposed method is based on the construction of minimum CDS. Most of the above CDS algorithms aimed at small size of CDS but not considered the lifetime of CDS. In this paper we consider the lifetime of CDS. Generally they are classified into global, quasi global, local, quasi local based on the type of information it gathers. The CDS construction algorithms are classified into two types; Centralized and Distributed.

3.2.1 Centralized CDS algorithms

Centralized algorithms require entire network topology of MANETs. But they produce small size CDS and better performance ratio when compared to distributed CDS algorithms. Guha and Khuller [6] propose two polynomial time algorithms to construct a CDS in a general graph G. These algorithms are greedy and centralized. The first one has an approximation ratio of $2(H(\Delta)+1)$, where H is a harmonic function and Δ is the degree of the node that has highest number of neighbours in G. Here a spanning tree is developed with maximum degree node as root. The non-leaf nodes in the tree form the CDS. The second algorithm constructs a Steiner tree that connects all dominator (black) nodes to form CDS. The size of CDS is at most $(ln(\Delta)+3)|OPT|$, where $|OPT|$ is the size of an optimal MCDS. The message complexity is $O(n^2)$ and time complexity is $O(n^2)$. Ruan *et al.* [18] propose a one-step greedy approximation algorithm with performance ratio at most $3+ln(\Delta)$. Min *et al.* [20] propose a two phase algorithm, in first phase MIS is constructed and second phase forms the CDS based on Steiner tree with minimum number of Steiner nodes. It has a performance ratio of $6.8 \mid MCDS \mid$. Cheng *et al.* [7] proposed a three phase algorithm to construct MCDS in UDG. It has time complexity of $O(n)$ and message complexity of $O(n)$. Butenko *et al.* [21] proposed a pruning based CDS construction. It has time complexity of $O(|V|. |E|)$ and message complexity of $O(n^2 log^3 n)$.

3.2.2 Distributed CDS algorithms

CDS is constructed based on localized information. These are further classified into prune based and MIS based. Alzoubi *et al.* [4] is a distributed algorithm based on MIS, this algorithm has time complexity of $O(n)$, and message complexity of $O(n \log(n))$. The resulting CDS has a size of at most $8|OPT| + 1$. Wu and Li's algorithm [3] is very simple. The localized property makes the CDS maintenance easier. The algorithm has a linear performance ratio. This algorithm needs at least two-hop neighbourhood information. It is presented based on the general graph model and it has time complexity of $\Theta(m)$, and message complexity of $O(\Delta^3)$. They analysed the movement of nodes with an on model or off model as leaving from one small management domain and entering another management domain. Das *et al.* [8] propose a three staged algorithm, first stage identifies dominating set, second stage constructs spanning forest and third stage constructs the spanning tree. It has a time complexity of $O(n^2)$, message complexity of $O(n^2)$ and cardinality is at most $3H(\Delta)$. Stojmenovic *et al.* [22] propose a cluster based CDS construction which is based on 2-hop neighbourhood knowledge. Here ranking is based on degree and location of the node. It has time complexity of $\Omega(m)$, and message complexity of $O(n^2)$. Dai *et al.* [5] proposed a subtraction based CDS algorithm that consists of two stages. In the first stage, 2-hop neighbouring information is obtained. A node subtracts itself from the consideration of the CDS if it has a direct link between any pair of its 1-hop neighbours. In the second stage, efficient heuristics are applied to further reduce the size of CDS. It has a time complexity of $O(\Delta^2)$, message complexity of $O(\Delta)$, where Δ is the maximum degree of G. Alzoubi *et al.* [19] proposed a distributed algorithm based on 1-hop neighbour information. It has the time complexity of $O(n)$, and message complexity of $O(n)$. Li *et al.* [23] proposed an algorithm with two phases. At the first phase, an Maximal Independent Set (MIS) is formed. At the second phase, a Steiner tree algorithm is used to connect the MIS. The Steiner tree algorithm uses the property that any node in an UDG is adjacent to at most five independent nodes [4]. It has a time complexity of $O(\Delta)$, message complexity of $O(n\Delta^2)$. Cheng *et al.* [7] propose a multi leader algorithm. Initially, it selects the node with minimum ID in its 1-hop neighbourhood as a leader, build a tree rooted at each leader and connect two adjacent trees through one or two nodes. It has a message complexity of $O(n \log n)$ and time complexity of $O(n)$.

3.3 Convex Hulls

Irrespective of the fact that the above CDS based algorithms are able to form a virtual back bone network for the given MANET they do not consider the geometrical attributes like perimeter of the given area. Convex hull [26, 32, 33] based algorithms consider the geometric attributes and they are able to reduce the physical area in which the broadcasting could take place. Jarvis's march algorithm [28] wraps up a piece of string around the points. This algorithm is usually called as *gift-wrapping* algorithm. It starts by computing the leftmost point (smallest x-coordinate), because by the property of convex set the left most point must be a convex hull vertex. This is an output sensitive algorithm whose time complexity is $O(nh)$ and the worst-case running time is $O(n^2)$. Graham's Scan algorithm [29] starts by finding the leftmost point suppose 'x' and sorts the points in counter clockwise order around 'x'. Usually a comparison-based sorting algorithm has a time complexity of $O(n \log n)$. To compare two points p and q, whether the triple x, p, q is oriented clockwise or counter clockwise is checked. Divide and Conquer algorithm [33] resembles quick sort. Choosing a *pivot* point '*p*' partitions the input points into two sets L and R, containing the points to the left of p, including p itself, and the points to the right of p. By comparing the x-coordinates, recursively compute the convex hulls of *L* and *R*. Finally, merge the two convex hulls into the final output. The runtime of this algorithm is *O(n log n)*. Chan's Algorithm [34] is an output-sensitive algorithm with a combination of divide-and-conquer and gift-wrapping. Here the set of inputs are divided into n/h subsets with size of 'h' each. Then it converges in *O(n log h)*. But the value of 'h' is a guess, it may progress in exponential way. Incremental Insertion algorithm [27, 31] begins by sorting the points by their *x*-coordinates. To insert new point, it is connected to the rightmost point in the convex hull, as well as one of that point's neighbour. Repeatedly remove concave corners from this polygon. The run time of this algorithm is *O(n log n)*.

In Randomized Incremental Insertion algorithm [30], each point maintains a pointer to an arbitrary visible edge (null if point is inside the existing hull). This can be used for repairing the convex hull algorithm. Each point *p* maintains a pointer *e*(*p*) to some *visible* edge of the evolving convex hull, nothing but the *conflict edge* for *p*. If *p* is already inside the evolving convex hull, then *e*(*p*) is a null pointer. Insert each new point *p* by interweaving it into the endpoints of edge *e*(*p*) and then removing concave vertices. If the points are inserted in random order then the time complexity is *O(n log n)*.

Quick Hull Algorithm [25] runs in *O*(*n* log *n*) time for favourable inputs and performs well than most of the others. Here Quick Hull begins by computing the points with the maximum and minimum, *x* and *y* coordinates. Clearly these points must be on the convex hull. Horizontal and vertical lines passing through these points are support lines for the convex hull, and so define a bounding rectangle, within which the hull is contained. The points lying within this quadrilateral can be eliminated from further consideration. Consider some hull edge *ab* then the task is to identify outermost point *c* from outside hull points and from an imaginary triangle *abc*, and split the remaining points into two subsets, those that lie outside *ac* and those than lie outside of *cb*. This will be continued until all convex vertices are identified. Table-2 shows the run time complexity of various convex hull algorithms.

Table-2. Performance of convex hull algorithms.

Convex Hull algorithm	Run time complexity
Jarvis's march	O(nh)
Graham's scan	O(n log n)
Divide and conquer	O(n log n)
Chan's	O(n log h)
Incremental insertion	O(n log n)
Quick hull	O(n log n)

Since convex hull based algorithms are able to work in terms of the physical area, we build our algorithm for constructing the minimum connected dominating set to reduce the broadcast storm problem. Here we propose a specialized convex hull algorithm, which is a Quick hull algorithm with Incremental sweeping. This algorithm prunes away a large subset of the interior points (nodes) before recursion, which ultimately helps in yielding the MCDS.

4. PROPOSED WORK

The MANET is assumed to be distributed in a two dimensional plane. Each node has an equal transmission range. Initially the energy of the nodes are assumed to be maximum and equal. Here the topology of wireless ad hoc network is modelled as a unit-disk graph [2]. This is a Convex Hull Based Minimum Connected Dominating Set algorithm (CHMCDS), which is distributed in nature and depends on local 1-hop information. The convex hull plays an important role in design of this algorithm. Convex Hull is very much adaptable to topological changes in the MANET due to mobility. In highly mobile MANETs, convex hull helps in reconstruction of CDS which in turn makes the CDS a fault tolerant. The topology changes are mainly due to addition of new nodes or moving out of existing nodes. It has four phases: initial dominating set phase, convex hull phase, MCDS construction phase and maintenance phase which handles mobility and energy depletion of the nodes. Here a generalized graph colouring process is used in Maximal Independent Set (MIS) formation. Four different colours are used here, WHITE indicates the initial and newly joined nodes, BLACK indicates the Dominating nodes, RED indicates the convex hull nodes and GREY indicates the neighbours of dominating nodes.

4.1 Initial dominating set phase

In this phase the dominators are selected based on 1-hop neighbour information. This is done in a distributive manner.

Algorithm phase 1

```
1. Initially every node is in White and exchanges its ID, Position (i.e X,Y co-ordinates) in its transmission range {Nodes are assumed to have GPS}
2. Every node gets 1-hop neighbour information $N_1$ (v) and stores in its table.
3. Node with lowest ID and maximum number of neighbours advertises itself as Dominator in its range and waits for some time ΔT.
4. Within this time,
5. If no other advertisements come
6. Then
```

7. *It will become the Dominator and converts itself as Black.*
8. *Else*
9. *The node with highest number of neighbours will become the Dominator.*
10. *Now its neighbours become Grey.*

This phase is illustrated in Figure-1 and Figure-2. Figure-1 shows the initial Mobile ad-hoc network a general scenario; here all nodes are in White colour. After obtaining 1-hop knowledge, the nodes with highest number of neighbours become dominators and turns Black. Here nodes 5, 13 and 18 become dominators.

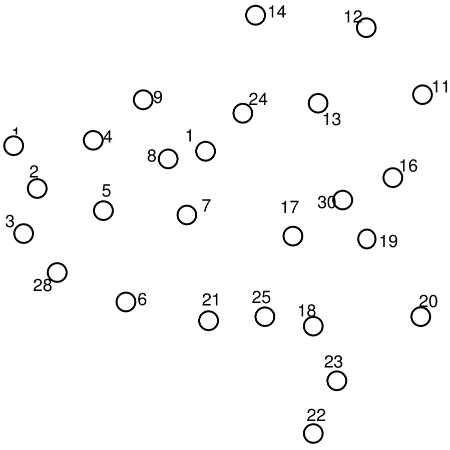
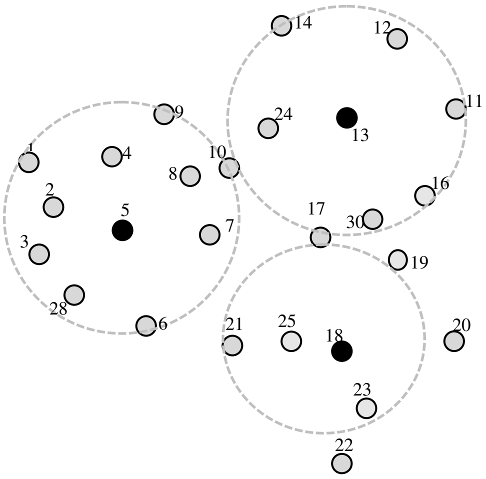


Figure-1. Initial mobile ad-hoc network.



BLACK- Dominators
GREY- Neighbours

Figure-2. Dominating Set in MANET.

4.2 Convex Hull phase

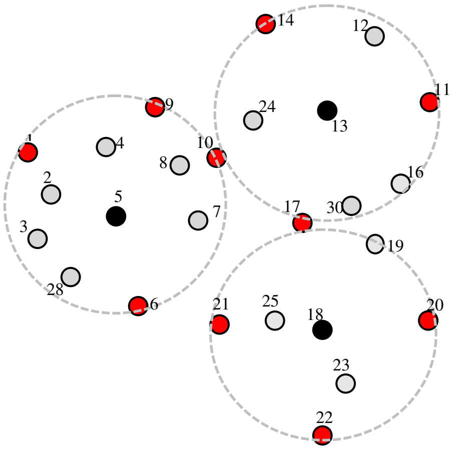
In this phase each Dominator constructs a convex hull of its neighbours. As the nodes are MANET nodes, a new convex hull algorithm that suits the MANTET is proposed in this paper. It is capable of adapting to topology changes that occur due to addition of new nodes or mobility of existing nodes or depletion of nodes energy. It can also be considered as one of the best optimal MCDS and its approximation ratio is very less when compared to existing MCDS algorithms. Also it has 100% reachability. The new algorithm proposed in this work is a combination of Quick hull and Randomized Incremental sweep algorithms. Initially Quick hull algorithm uses quickly sort to sort the nodes according to their position. After this Incremental sweep is used to form final convex hull.

Algorithm Phase 2

1. *Initialize a convex set, $C_H = \{\Phi\}$ and begin the Quick hull process.*
2. *Every Dominator sorts their 1-hop neighbour's list, $N_1(v)$ according to their $X=\{x_1, x_2, ...x_n\}$, $Y=\{y_1, y_2, ...y_n\}$ co-ordinates using Quick sort.*
3. *Now the nodes with lowest and highest 'X' co-ordinates as well as 'Y' co-ordinates will fall on convex hull. Insert these nodes in to C_H.*
4. *Join these nodes with edges to form a quadrilateral.*
5. *Now the nodes inside the polygon (Concave nodes) will be of no effect. Nodes above the edges are considered for further construction.*
6. *Find the farthest node from each of these edges and join to form an imaginary triangle and insert this node in C_H.*
7. *The nodes inside the triangle are concave nodes of no effect.*
8. *Repeat step 7 recursively.*
9. *Continue the process until no further neighbour nodes exist.*
10. *Finally C_H forms the convex hull.*
11. *Turn all the C_H nodes to Red colour.*

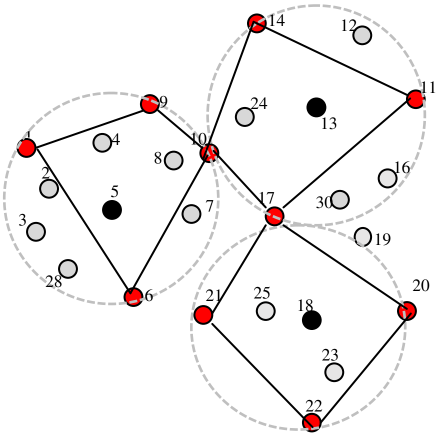
Phase 2 is illustrated Figure-3. Here nodes 5, 13 and 18 are the dominating set {Black} nodes and each dominator distributively constructs the convex hull. For node 5(dominator) the initial convex hull nodes are $C_{H5} = \{1, 6, 9, 10\}$. Similarly for nodes 13 and 18, $C_{H13}=\{10, 11, 14, 17\}$ and $C_{H18}=\{17, 20, 21, 22\}$ respectively. Figure-3 shows the convex set C_H (Red nodes) for each dominator. Figure-4 shows the initial formation of convex hull polygon by applying phase of the algorithm i.e lowest and highest X co-ordinate node and Y co-ordinate node at each region. Here three convex polygons are formed. Now with incremental sweep, the nodes that are farthest from each convex edge are joined to form imaginary triangles. Here nodes {3, 7}, {12, 16} and 19 are farthest from each of its initial convex edges. The imaginary triangles from convex edges are shown in Figure-5. Now concave nodes have no impact. Concave edges are also removed. Node 28 in C_{H5} and node 30 in C_{H13} falls out of the triangle. So, the incremental sweep is applied to these nodes as shown in

Figure-6. Here the edges $\overline{1,6}$, $\overline{11,14}$, $\overline{11,17}$, $\overline{17,20}$ (concave edges) are removed. The final convex hull for each dominator after removing concave edges is shown in Figure-7.



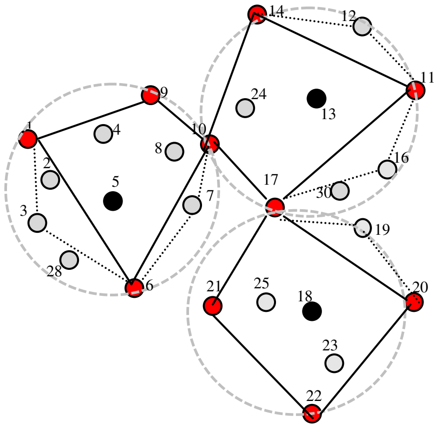
BLACK- Dominators
GREY- Neighbours
RED- Convex Hull Nodes

Figure-3. The initial convex hull nodes.



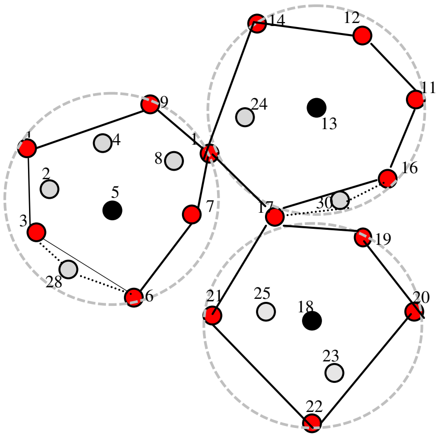
BLACK- Dominators
GREY- Neighbours
RED- Convex Hull Nodes
—— Convex Edge

Figure-4. Initial convex polygon.



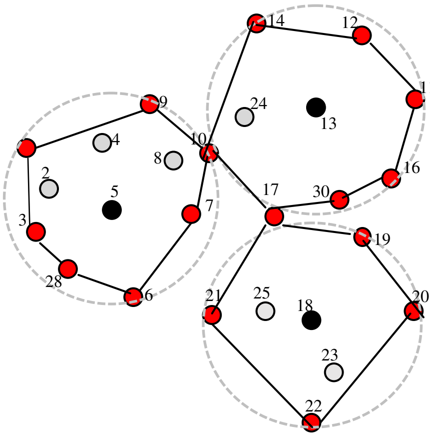
BLACK- Dominators
GREY- Neighbours
RED- Convex Hull Nodes
—— Convex Edge
······ Imaginary Edge

Figure-5. Convex hull formation with Incremental sweep.



BLACK- Dominators
GREY- Neighbours
RED- Convex Hull Nodes
—— Convex Edge
······ Imaginary Edge

Figure-6. Convex hull formation with Incremental sweep.



BLACK- Dominators
GREY- Neighbours
RED- Convex Hull Nodes
—— Convex Edge

Figure-7. Final convex hull formation.

4.3 MCDS formation phase

Only the convex hull nodes plays important role here in formation of CDS which is one of the optimal MCDS.

Algorithm Phase 3

```
1. Every Dominator (Black node) exchanges its convex hull list $C_H$ with neighbour Dominators.
2. The common convex hull node (Red node) is selected as the connector.
3. If there are more than one common convex hull node Then
```

The tie is resolved using three parameters: Residual energy of the node, Mobility (Speed) of the node and Position of the node (Short distance) i.e. the node with high residual energy, low mobility and short distance between the Dominators is selected as connector.

```
4. Convert the common convex hull node (Red) to Black // It is the connector
5. All the Black nodes form the MCDS.
```

As illustrated in Figure-7, the Red nodes 10 and 17 are the common convex hull nodes between the dominators 5, 13 and 18. Hence these nodes are turned to Black (Connectors) and edges are established as shown in below Figure-8. Now the MCDS= {5, 10, 13, 17, 18} which is an optimal MCDS. Hence these MCDS nodes are involved in broadcasting, which reduces broadcast storm problem.

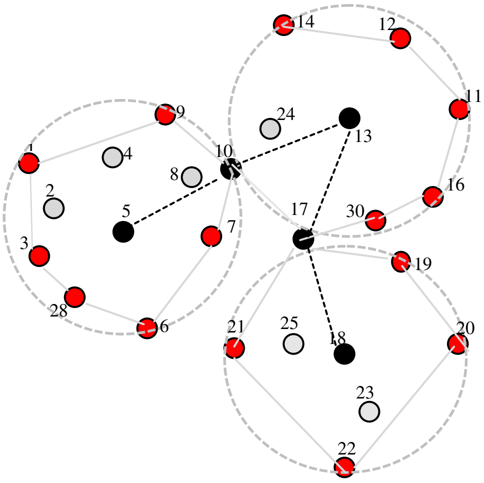


Figure-8. Optimal MCDS formation.

4.4 Maintenance phase

After MCDS is established, the maintenance phase improves the life time of CHMCDS by considering *Mobility* and *Residual energy* of the nodes. The topology changes mainly occur for two reasons, first one is node unavailability due to node's energy depletion or node moving out of the network and second one is due to addition of new nodes to the network. All the cases are analyzed in the following phase which is executed at regular intervals based on the density of the network.

Algorithm Phase 4

```
1. Every Convex hull (Red) node checks its neigbour's list $N_1(v)$ at regular intervals T.
2. If ( $N_1(v) < N_1(v)^1$ ) // New nodes (White) are added
3. Then
4. Red node waits for $\Delta T_1$.
5. If New nodes (White) are converted to Grey // i.e. white nodes become neighbours of some dominator
6. Then
7. No change.
8. Else
9. The Red node will become Black. // i.e. New Dominator
10. End If
11. Else
12. No change.
13. End If
14. Every Dominator (Black) checks its neighbour's list $N_1(v)$ at regular intervals T.
15. If ( $N_1(v) < N_1(v)^1$ )
16. Then
17. If New White nodes are concave nodes
18. Then
19. No change
```

```
Else
Update the convex hull of the dominator using incremental sweep
End if
Else
No change
If a Convex hull node moves out of the network
Then
Immediate concave node will become the convex hull node.
Else
No change.
If a Connector moves out of the network
Then
An edge is established between the immediate concave nodes of either side
If a Dominator moves out of the network
Then
Repeat algorithm phase 2 locally.
```

Phase4 is illustrated in Figure-9. Here the convex hull node 9 is moved out of region, Node 4 is the immediate concave node which will become the convex hull node. The connector node 10's energy is depleted, then the two immediate concave nodes 8 and 24 are turned to Black and an edge is established. Some new nodes 31 and 49 are joined as neighbours to node 20. In this case node 20 is turned to Dominator (Black). Similarly, node 67 has added as neighbour to dominator 13 near nodes 12 and 11. So, the convex hull is updated. Now the new MCDS = {5, 8, 24, 13, 17, 18, 20} which can be converged in less time when compared to other maintenance algorithms. This is illustrated in figure 10.

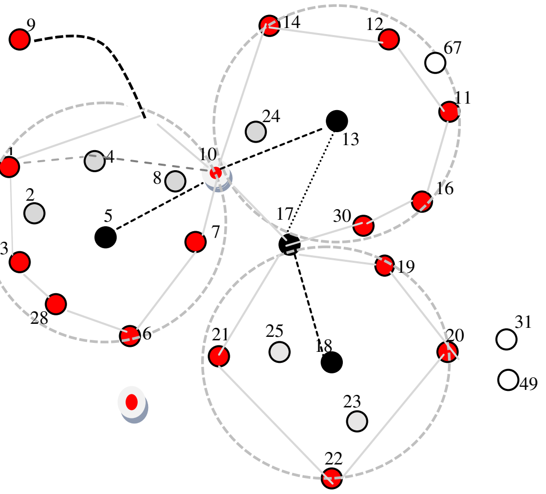


Figure-9. Maintenance of CHMCDS.

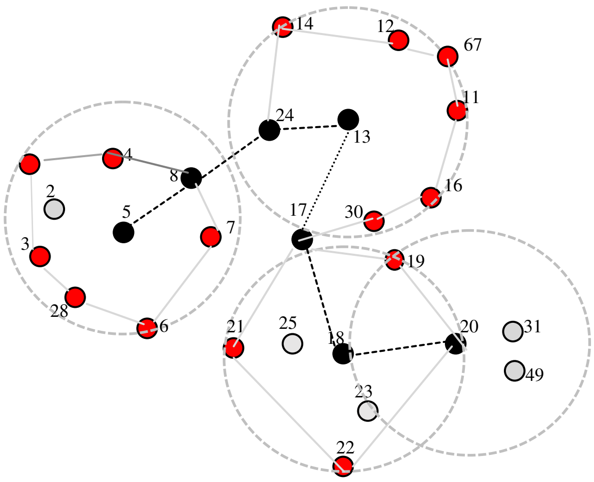


Figure-10. CHMCDS maintenance phase.

5. PERFORMANCE EVALUATION

5.1 Time and message complexity

5.1.1 Time complexity

Initially a convex quadrilateral is formed based on minimum and maximum, *x*- and *y*- co-ordinates which is done in parallel. This can be achieved in *O(n)* where 'n' is number of nodes in graph [24][25]. Recursively it forms the imaginary triangles around the quadrilateral to achieve the final convex hull. The total run time for quick hull is of *O(n log n)*. After the convex hull is formed, only the convex hull nodes involve in further communication to form the MCDS. This can be achieved in *O(h)*. So, the total time complexity of CHMCDS is *O(nh log n)* where '*n*' is number of nodes in graph and '*h*' is the number of convex hull nodes.

5.1.2 Message complexity

This distributed algorithm is based on 1-hop neighbour information. If '*n*' is the number of nodes in the graph. For every colouring operation of a node, a *hello* message is sent for *n*-1 adjacent neighbours, message complexity will be of *O*(*n*). The message complexity for the quick hull algorithm is *O*(*n log n*) [25][33]. After the construction of convex hull only the convex hull nodes are involved in further communication. If 'h' is the number of convex hulls nodes in the graph, then the complexity has an upper bound of *O(h)*. Therefore total messaging complexity of our algorithm has an upper bound of *O(nh log n)*.

5.2 Simulation results

In this section, the simulation results of our CHMCDS algorithm is compared with energy efficient distributed CDS algorithms [37] [38]. We present simulations that illustrate the results and analyse the behaviour of our algorithm in various scenarios. We run simulations under the *ns*-2 (version 2.34). The simulation parameters are listed in Table-3. Various aspects like Size of the CDS, Lifetime of the CDS and Message Overhead are observed at various speeds and transmission ranges and also compared with existing energy efficient CDS algorithms [37] [38]. Performance is analysed at sparse as well as dense networks. A sparse network is created with 100 nodes with transmission range of 25mts at two different speeds 5mts/s and 10mts/s. Similarly a dense network is created with 500 nodes with transmission range of 50mts at two different speeds 5mts/s and 10mts/s. A Random walk mobility model is used. The below graphs, Figure-11 shows the size of CDS Vs Number of nodes in sparse network with node's transmission range= 25mts, at node speed of 5mts/s. Figure-12 shows the size of CDS Vs Number of nodes in sparse network with node's transmission range= 25mts, at anode speed of 10mts/s. In both the cases CHMCDS performed well irrespective of speed and obtained optimal CDS Size in sparse network when compared to the existing CDS algorithms. Figure-13 shows the size of CDS Vs Number of nodes in dense network with node's transmission range of 50mts, at node speed of 5mts/s. Figure-14 shows the size of CDS Vs Number of nodes in dense network with node's transmission range of 50mts, at node speed of 10mts/s. In these cases also CHMCDS performed well when compared to other algorithms. Figure-15 shows the Message overhead Vs Number of nodes with node's transmission range of 50mts and speed 10mts/s. Here the CHMCDS exchanged optimal number of messages when compared to the existing standard CDS algorithms. The graph in Figure-16 shows the Lifetime of MCDS Vs Mobility of nodes with node's transmission range= 50mts. Lifetime of CHMCDS is observed at different speeds of the nodes 5mts/s, 10mts/s, 15mts/s and 20mts/s. The lifetime of CHMCDS in case of mobility is also good when compared to the existing standard CDS algorithms. Figure-17 shows the delivery ratio Vs Number of nodes with transmission range of 25mts and speed of 10mts/s. The lifetime of CHMCDS is more compared to the other existing CDS algorithms. So the CHMCDS is robust, reliable and fault tolerant.

6. CONCLUSION AND FUTURE WORK

The major advantage of using convex hull algorithm is fast convergence and longer life even if CDS nodes moved out due to mobility. Only convex set node are involved in further communication to establish the back bone and in maintenance of MCDS. These convex hull nodes reduce the time and message complexity. As the convex hull nodes are farthest nodes from the dominator, it proves that it gives shortest path to connect the dominating set, i.e. CDS size. Major advantage of this CHMCDS is, when a MCDS is broken then the convex hull nodes are used for broadcasting before reconstruction of MCDS. In future work we will extend this to directed graphs, general graphs and vehicular ad-hoc networks.

Table-3. Simulation environment.

Simulator	**Ns2 (Version 2.34)**
Simulation area	1000 × 1000m
Propagation	Two-ray Ground Reflection
MAC Protocol	IEEE 802.11
Bandwidth	2 Mbps
Traffic	CBR
Transmission range	25~100 meters
Number of nodes	100~500
Maximum speed	5~25 m/sec
Mobility model	Random Walk
Broadcast sessions	50
Broadcast rate	2–5 pkts/s
Message size	128 bytes
Hello interval	2 sec
Simulation time	100 minutes
Number of trials	50

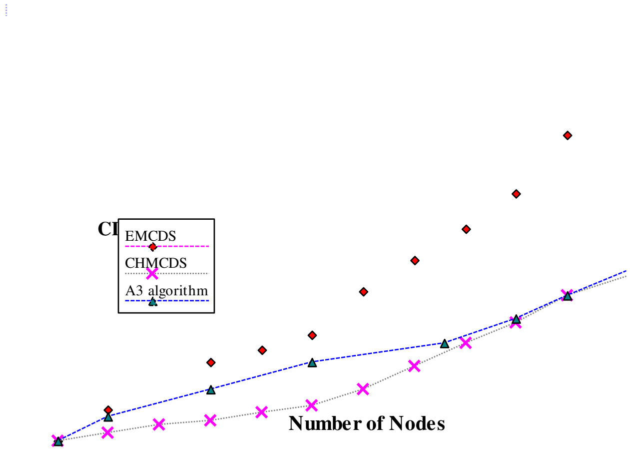


Figure-11. Number of nodes Vs CDS size (Transmission Range = 25mts and Speed= 5mts/s).

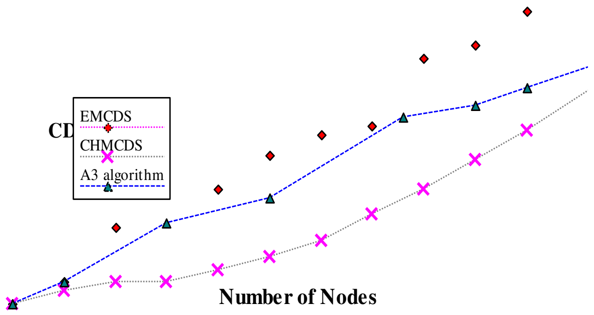


Figure-12. Number of nodes Vs CDS size (Transmission Range = 25mts and Speed= 10mts/s).

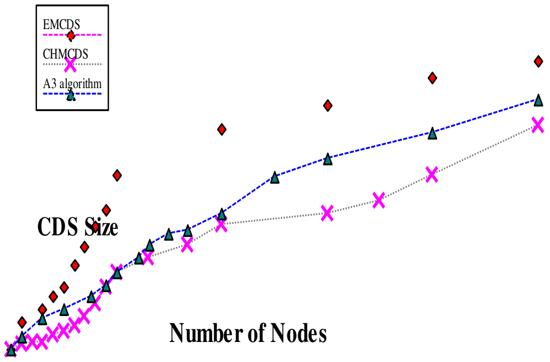


Figure-13. Number of Nodes Vs CDS Size (Transmission Range = 50mts and Speed = 5mts/s).

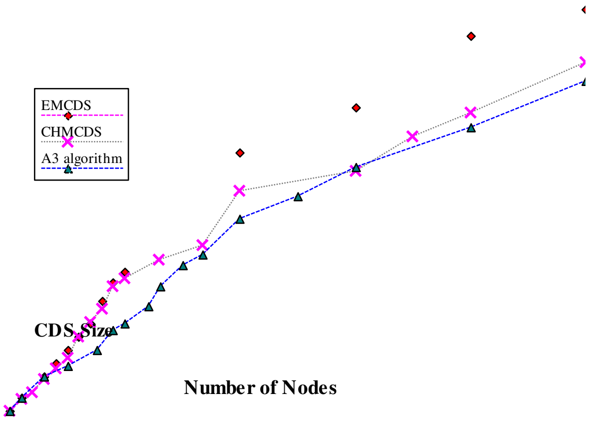


Figure-14. Number of Nodes Vs CDS Size (Transmission Range = 50mts and Speed= 10mts/s).

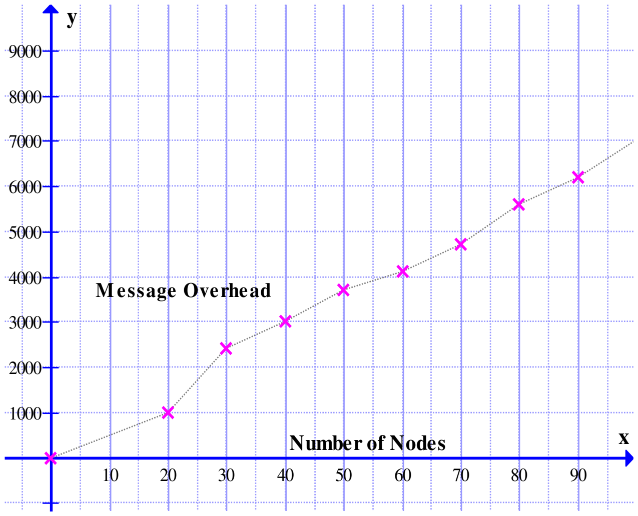


Figure-15. Number of Nodes Vs Message Overhead node’s transmission range of 50 m.

Figure-16. Mobility of Nodes (mts/s) Vs MCDS Lifetime (minutes) with node's transmission range of 25 m.

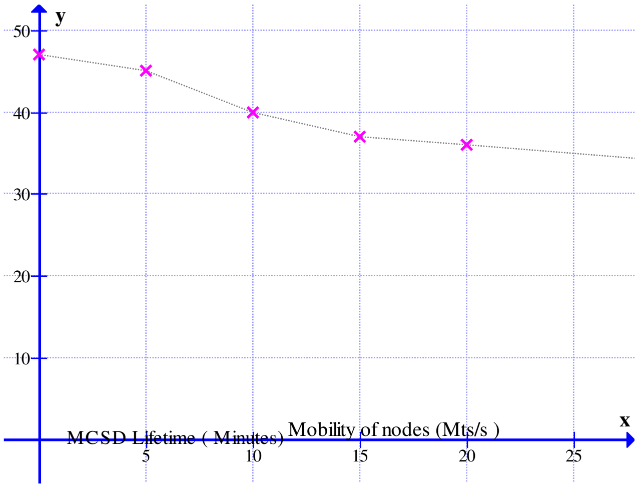


Figure-17. Mobility of Nodes (mts/s) Vs MCD Lifetime (minutes) with node's transmission range of 50 m.

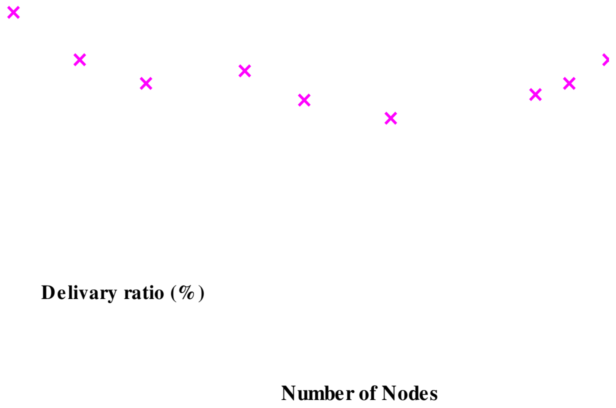


Figure-18. Number of Nodes Vs Delivery Ratio (%) with node's transmission range of 25 m.

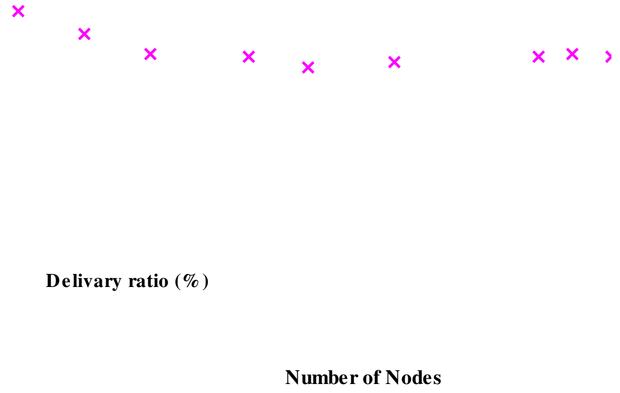


Figure-19. Number of Nodes Vs Delivery Ratio (%) with node's transmission range of 50 m.

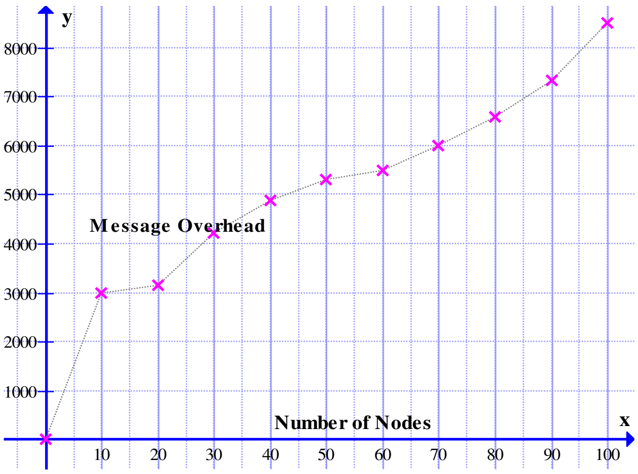


Figure-20. Number of nodes Vs Message overhead at 25.

REFERENCES

[1] S. Ni, Y. Tseng, Y. Chen and J. Sheu. 1999. The broadcast storm problem in a mobile ad hoc network. Proc. of ACM/IEEE MOBICOM'99. pp. 151-162.

[2] B. N. Clark, C. J. Colbourn and D. S. Johnson. 1990. Unit Disk Graphs. Discrete Mathematics. 86: 165-177.

[3] J. Wu, H. Li. 1999. On calculating connected dominating set for efficient routing in ad hoc wireless networks. Proceedings of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. pp. 7-14.

[4] P. Wan, K. Alzoubi, O. Frieder. 2002. Distributed construction of connected dominating set in wireless

networks. Proceedings of IEEE Conference on, Computer Communications (INFOCOM). pp. 141-149.

[5] F. Dai, J. Wu. 2004. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. IEEE Transaction on Parallel Distributed Systems. 15(10): 908-920.

[6] S. Guha, S. Khuller. 1998. Approximation algorithms for connected dominating sets. Algorithmica. 20, pp. 374-387.

[7] X. Cheng, M. Ding, D. Chen. 2004. An approximation algorithm for connected dominating set in ad hoc networks. Proceedings of the First International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer, Networks.

[8] B. Das, V. Bharghavan. 1997. Routing in Ad hoc Networks using Minimum Connected Dominating Sets. ICC'97, Montreal, Canada. pp. 376-380.

[9] Jacquet P, Muhlethaler T, Clausen T, Laouiti A, Qayyum A, Viennot L. 2003. Optimized Link State Routing Protocol ad Hoc Networks. Hipercom Project, INRIA Roquencourt, BP 105, 78153 Le Chesnay Cedex, France.

[10] D. B. Johnson and D. A. Maltz. 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing, Kluwer Academic Publishers. 353: 153-181.

[11] C. E. Perkins and E. M. Royer. 1999. Ad Hoc On-Demand Distance Vector Routing. Proceedings of IEEE Workshop on Mobile Computing Systems and Applications. pp. 90-100.

[12] C. E. Perkins and P. Bhagwat. 1994. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proceedings of ACM SIGCOMM. pp. 234-244.

[13] B. Williams, T. Camp. 2002. Comparison of broadcasting techniques for mobile ad-hoc networks. Proc. MobiHoc, Lausanne, Switzerland.

[14] Q. Zhang and D. P. Agrawal. 2005. Dynamic probabilistic broadcast in MANETs. Journal of Parallel and Distributed Comp. 65: 220-233.

[15] Y. Sasson, D. Cavin and A. Schiper. 2002. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In Swiss Federal Institute of Technology, Technical Report IC/2002/54.

[16] Justin Lipman, Hai Liu and Ivan Stojmenovic. 2009. Broadcast in Ad-hoc networks. Computer Communications and Networks, Springer-Verlag London Limited.

[17] K.M. Alzoubi, P.J. Wan and O. Frieder. 2002. Distributed Heuristics for Connected Dominating Sets in Wireless Ad Hoc Networks. Journal of Communications and Networks. 4(1).

[18] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li and K. Ko. 2004. A greedy approximation for minimum connected dominating sets. Theoretical Computer Science. 329: 325-330.

[19] K.M. Alzoubi, P.J. Wan and O. Frieder. 2002. Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks. MOBIHOC, EPFL Lausanne, Switzerland.

[20] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang and W.Wu. 2006. Improving construction for connected dominating set with steiner tree in wireless sensor networks. Journal of Global Optimization. 35: 111-119.

[21] S. Butenko, X. Cheng, C. Oliveira and P.M. Pardalos. 2004. A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks. Re-cent Developments in Cooperative Control and Optimization, pp. 61-73, Kluwer Academic Publishers.

[22] I. Stojmenovic, M. Seddigh, J. Zunic. 2001. Dominating Sets and Neighbor Elimination Based on Broadcasting Algorithms in wireless networks, Proc. IEEE Hawaii Int. Conf. on System Sciences.

[23] Y. Li, M.T. Thai, F. Wang, C.W. Yi, P.J. Wan, D.Z. Du. 2005. On greedy construction of connected dominating sets in wireless networks. Wireless Communications and Mobile Computing. 5(8): 927-932.

[24] Preparata F.P. and Hong S.J. 1977. Convex Hulls of Finite Sets of Points in two and Three Dimensions. Communications of ACM. 20(2).

[25] Barber C. B., D.P. Dobkin and H.T. Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. ACM Transactions on Mathematical Software. 22(4): 469-483.

[26] Joseph O'Rourke. 1998. Computational Geometry in C (2nd Edition). Chapter 3. "Convex Hulls in 2D".

[27] Marton Trencseni, Istvan Csabai. 2012. Plane-Sweep Incremental Algorithm: Computing Delaunay Tessellations of Large Datasets. Cornell Univ. Library, Computational geometry, arXiv: 1210.3595.

[28] Jarvis R. A. 1973. On the identification of the convex hull of a finite set of points in the plane. Information Processing Letters. 2(1): 18-21.

[29] R. L. Graham. 1972. An efficient algorithm for determining the convex hulls of a finite point set. Information Processing Letters. 1: 132-133.

[30] Clarkson K.L., K. Mehlhorn and R. Seidel. 1993. Four results on randomized incremental construction. Computational Geometry: Theory and Applications. 3: 185-211.

[31] M. Kallay. 1984. The Complexity of Incremental Convex Hull Algorithms in R^d. Information Processing Letters. 19, 197.

[32] G.N. Purohit and Usha Sharma. 2010. Constructing Minimum Connected Dominating Set: Algorithmic approach. International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC). 2(3).

[33] David M. Mount, © Lecture Notes on. 2002. Computational Geometry. Deptartment of Computer Science, University of Maryland, College Park, Fall.

[34] T. Chan. 1996. Optimal output-sensitive convex hull algorithms in two and three dimensions. Discrete and Computational Geometry. 16, 361-368.

[35] Jiguo Yu, Nannan Wang, Guanghui Wang, Dongxiao Yu. 2013. Connected dominating sets in wireless ad hoc and sensor networks - A comprehensive survey. Computer Communications. 36, 121-134.

[36] Pedro M. Wightman, Miguel A. Labrador. 2011. A family of simple distributed minimum connected dominating set-based topology construction algorithms. Journal of Network and Computer Applications. 34: 1997-2010.

[37] Qiang Tang, Kun Yang, Ping Li, Jianming Zhang, Yuansheng Luo and Bing Xiong. 2012. An energy efficient MCDS construction algorithm for wireless sensor networks. EURASIP Journal on Wireless Communications and Networking. p. 83