Enhanced DSR by using MCDS for Ad hoc network

Nirmala Chouhan*, Prof. Deepali Kelkar, Antim Bala chouhan*****

(MIT Ujjain,m.p.,India)

***Abstract*--Blind broadcast in a mobile ad hoc network is a common problem. Blind broadcast in a wireless ad hoc network means any mobile node will rebroadcast all received broadcast messages. One node may receive the same copy of a message from more than one neighbor, so unnecessary overhead is introduced. We can minimize this overhead by using MCDS algorithm**.

1. Introduction to MCDS

A connected dominating set (CDS) is used to reduce broadcast overhead. A common source of overhead in a wireless ad hoc network comes from blind broadcasts. Assuming the worst case, nodes in a wireless ad hoc network rebroadcast all received broadcast messages. Nodes may receive multiple copies of the same message from more than one neighbor. Therefore reducing redundant broadcast messages can reduce channel bandwidth consumption and increase bandwidth efficiency.

It is possible to significant reduction of overhead by using the minimal connected dominating set (MCDS) to reduce redundancy due to these blind broadcasts. In a simple graph G = (V, E) where V is the set of nodes and E is the set of edges Assume a node set T is subset of V such that for all 'X' in V-T, there exist 'Y' belongs to V, such that edge (x,y) belongs to E .

This is the cover property for a CDS (connected dominating set). Set T is called a dominating set. Set T is called a connected dominating set (CDS) when T forms a connected graph. This is the connectivity property for a CDS. Figure 1.1 gives an example of a CDS.

Black nodes 2 and 3 are connected and cover all nodes in the network. They form a CDS for this graph. Minimal set of CDS is known as minimal connected Dominating Set (MCDS). Since in given example CDS is already minimal so MCDS includes node 2 and node 3.

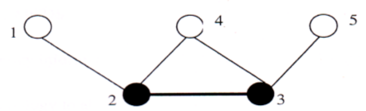


Fig 1.1 MCDS in mobile ad hoc network is treated as a virtual backbone for whole network.

A virtual backbone structure on the ad-hoc network will useful, in order to support unicast, mulitcast, and fault-tolerant routing within the ad-hoc network. This virtual backbone differs from the wired backbone of cellular network. The hosts in the MCDS maintain local copies of the global topology of the network, along with shortest paths between all pairs of nodes.

Findings a minimal CDS for a connected graph is an NP-hard problem. For a small graph, we can enumerate all possible cases to find a minimal solution. However, this approach is not feasible for larger graphs. Different approximation algorithms to find a minimal CDS have proposed. Generally distributed in to two classes:

- Global information based algorithms
- Local information based algorithms

Algorithms that use all information in the network are called global algorithms. The global algorithms assume that nodes keep identical copies of the entire topology. This is always true in a proactive link state routing protocol. Local algorithms utilize only local neighbor information.

The global algorithms, finds a dominating set 'c' in the first phase by selecting the node with the largest effective degree and stopping when "C" covers all nodes, (The effective degree is the number of neighbors not in c".) C' may contain several disconnected components. In the second phase, CDS tries to use the minimum number of extra nodes to connect the components of C' to form a CDS. In this phase, links are assigned weights. If a link connects two non-CDS nodes that are in the same component, this link is assigned a weight of infinity. The weight for other links is the number of end points that are not in C". The lightest links are chosen to connect components. This phase ends when a CDS is found.

The different global algorithms, start at the node with the highest degree. It extends the set of selected CDS nodes by including the nodes adjacent to the current set that have the largest number of uncovered neighbors or two-hop uncovered neighbors until no uncovered nodes is left.

The local algorithm is implemented in a distributed manner. This algorithm assumes that all nodes know all the other nodes that are within their two-hop range and that nodes have unique IDs. In the first phase, a node is selected as a potential member of the CDS if and only if it has two non-adjacent neighbors. Nodes broadcast if they elect themselves as members of the potential CDS. Two extensions are used in the second phase to reduce the size of the CDS. A node stays in the CDS unless a neighbor CDS node with a larger ID covers its entire neighbor set. As an extension, if two adjacent CDS neighbors with larger IDs cover the neighbor set of a node, this node may change itself to a non-CDS node.

To determine routes with the MCDS, global knowledge of G is gathered into all the MCDS nodes and compute shortest paths based on local copies of G (akin to the link state approach restricted to MCDS Nodes. In general, the routes determined by the MCDS nodes do not pass through the MCDS. However, the MCDS can handle routes in two situations:

- When a non-MCDS edge or node fails, the MCDS provides an immediate backup route to use while another shortest path is found, and
- The MCDS can be used for multicast and broadcast routing.

At a high level of abstraction, the MCDS based routing algorithm consists of the following steps to determine routes:

- Compute the MCDS.
- Gather topology information from non-MCDS nodes to MCDS nodes.
- Broadcast topology to all MCDS nodes.
- Determine routes. Each node runs an all pairs shortest path algorithm on its local copy of graph.
- Propagate information out to non- MCDS nodes.
- Send periodic maintenance update. Every T second for some large value of T, the MCDS nodes repeat the topology broadcast step. This periodic broadcast ensures that the MCDS nodes recognize drastic topology changes.

Guha and Khuller in proposed two approximation algorithms for the connected dominating set problem. The first approach is to develop a greedy algorithms for solving problem, it executes in connected dominating set manner. The second algorithm is an improvement of the first algorithms, and it executes in dominating set manner.

2 Algorithm-I(Connected Dominating Set Approach)

This approach is a greedy algorithm for solving the MCDS Problem. The idea behind the algorithms is the following grow a tree T, starting from the vertex of maximum degree. At each step a vertex v in T is picket and scanned it. Scanning a vertex, adds edges to T from v to all its neighbors not in T. In the end it obtains a spanning tree T, and will pick the non-leaf nodes as the connected dominating set.

Initially all vertices are unmarked (white). When a vertex is scanned (color it black), all its neighbors that are not in T is marked and add them T (color them gray). Thus marked nodes that have not been scanned are leaves in T (gray nodes). All unmarked nodes are white. The algorithm continues scanning marked nodes, until all the vertices are marked (gray or black). The set of scanned nodes (blank nodes) will from the connected dominating set (CDS) in the end. A vertex is picked that has the maximum number of unmarked (white) neighbors.

In figure 1.1 (a) and 1.1 (b) black nodes are obtained as CDS using algorithm I, and gray nodes are dominated nodes by dominating set. It is clear that no white nodes are left in the given graphs.

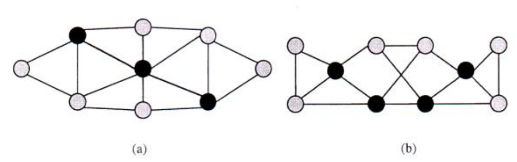


Fig2.1 (a-b): an example of Algo. I

3 Algorithms II (Dominating Set Approach)

This algorithm is an improvement of the first algorithm. The algorithm finds a dominating set in the first phase and in the second phase connects the dominating set.

At the start of the first phase all nodes are colored white. Each time it include a vertex in the dominating set, it is colored with black. Nodes that are dominated are colored gray (once they are adjacent to a black node). In the first phase the algorithm picks a node at each step and colors it black, coloring all adjacent white nodes gray. A piece is defined as a white node or a black connected component. At each step we pick a node to color black that gives the maximum (non-zero) reeducation in the number of pieces.

It the second phase, it has a collection of black connected components that need to connect. Recursively connect pairs of black components by choosing a chain of vertices until there is one black connected component.

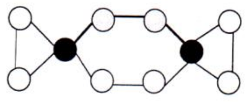


Fig 3.1 an example of Algo II

In figure 2.1 black and gray nodes are generated by algorithm II in first and second phases respectively. Since black nodes are disconnected so gray nodes are needed to connect these black nodes. CDS is both black and gray nodes.

3.1 DRAWBACK:-

Algorithm starts by selecting a node which has maximum degree. Since it is a heuristic approach for selection of a node ,it can increase the size of MCDS .

Proof : If CDS based algorithm discussed in is applied for graph like figure 2.1 (a) then obtained MCDS is like in figure 3.2 (b) where, all black nodes from CDS.

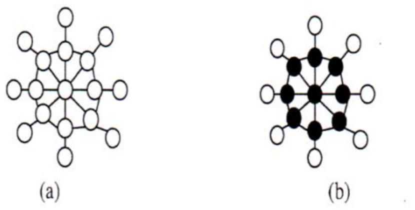
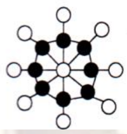


Fig 3.2: MCDS generated by Guha and Khuller algo

Here the Size of obtained MCDS is 9. But optimal solution is 8 only like figure 3.1 Hence, heuristic like maximum degree for starting node can lead to increased size of



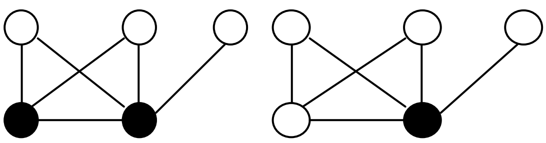
.

Fig 3.3: MCDS generated by optimal solution

4 Wu and Li's Algorithm

Wu and Li in proposed are the localized algorithm. This is localized algorithm which is implemented in a distributed manner. This algorithm assumes that all nodes know all the other nodes that are within their two-hop range and that nods have unique IDS.

In the first phase, a node is selected as a potential member of the CDS if and only if it has two non-adjacent neighbors. Nodes broadcast if they elect themselves as member of the potential CDS. Two extensions are used in the second phase to reduce the size of the CDS. A node stays in the CDS unless a neighbor CDS node with a larger ID covers its entire neighbor set. As an extension, if two adjacent CDS neighbor with larger IDs cover the neighbor set of a node, this node may change itself to a non- CDS node.



(a) (b)

Figure 4.1 (a-b): An example for Wu and Li's algorithm.

In Figure 4.1 (a) black nodes are generated after execution of first phase of Wu and Li's algorithm. If ID of vertex V is lower than the ID of vertex W then from second phase redundant node V is discarded from DS, and then resultant CDS is W only as shown in figure 4.1 (b).

4.1 DRAWBACK:

This algorithm applies only to wireless ad hoc networks whose unit disk graph is not a complete graph.

Proof : For example if a wireless ad hoc network is in from of a graph as shown in figure 4.2 (a) then MCDS obtained by using algorithm in is like figure 4.2 (b) where, black nodes is minimal connected dominating set. But practically it is not possible case for figure 4.2 (a). Figure 4.3

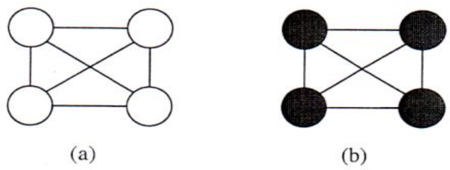


fig 4.2(a-b): MCDS generated by Wu and Li's algorithm.

Shows an optimal MCDS for the given graph (fig. 3.6(a)) where, black nodes represent the MCDS. Since the given graph is complete this proves drawback.

4.2 An example of Guha and Khuller algorithm

Assumptions of this algorithm

- All nodes are located in a two-dimensional plane.
- All nodes have an equal transmission range unit.
- Communication is bidirectional and hence edges of UDG are undirected.
- Each host sends message by local broadcast and an underlying MAC algorithms perfectly schedules message to prevent any collision.

Notations: The various notations are:-

- Circles denote the nodes.
- Lines represent the link between the nodes ie they are within each others wireless transmission range
- Black node represent the selected MCDS member
- Gray node represent the neighbors of MCDS member.

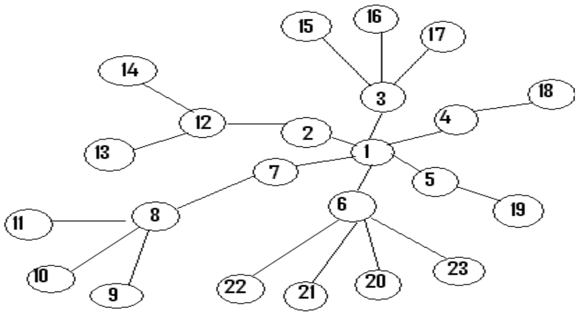


FIG 4.3(a)

The figure 4.3(a) represent the ad hoc network which consist of 23 nodes .On applying Guha and khuller algorithm in this fig MCDS can be obtained.

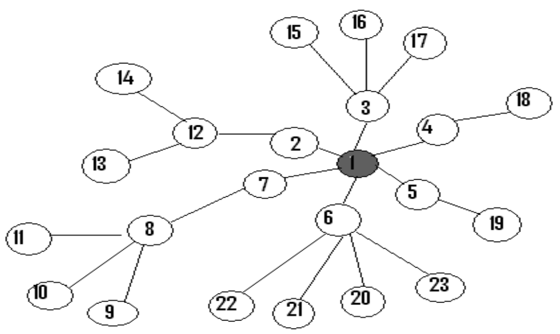


Fig 4.3(b)

First the node with maximum degree is selected .ie here node 1 is selected and is colored black. All the neighbours of node 1 ie nodes 2,3,4,5,6,7 are colored gray.Now among these gray colored nodes node with the maximum degree is selected again and is colored black .the 2nd node selected is 6 in fig 4.3(d).

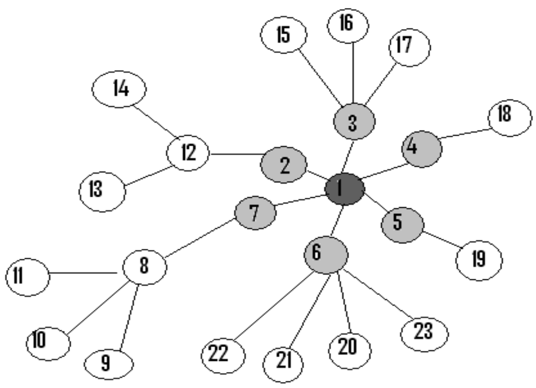


FIG 4.3(c)

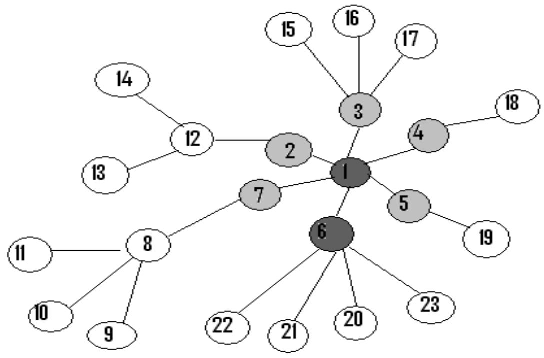


Fig 4.3 (d)

Now, all the neighbors of node 6 i.e. 20,21,22,23 are colored gray as shown in fig **4.3** (e). Among all the gray nodes in fig. **4.3** (e), the node 3 is colored black and all its neighbors i.e. 15, 16, 17 are colored gray which is shown in fig **4.3** (g).

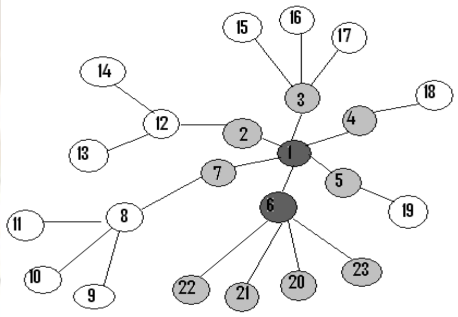


Fig 4.3 (e)

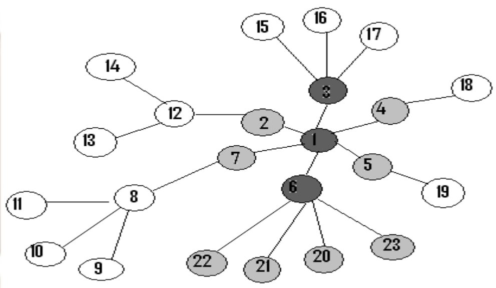


Fig 4.3 (f)

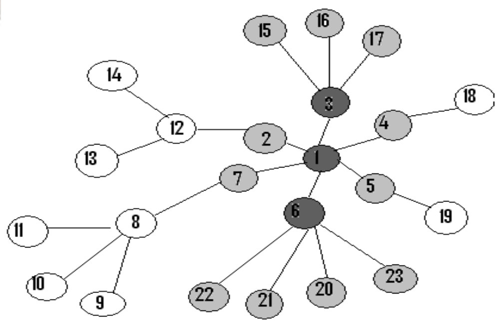


Fig 4.3 (g)

Now again, we select a gray node with a maximum degree among a set of gray nodes obtained in fig. **4.3** (g).Since node 2, 7, 4, 5 have the same degree, thus any one of these nodes can be selected.

So, we select node 7 and is colored black (shown in fig **4.3** (h)) and its only white neighbor 8 is colored gray as shown in fig **4.3** (i).

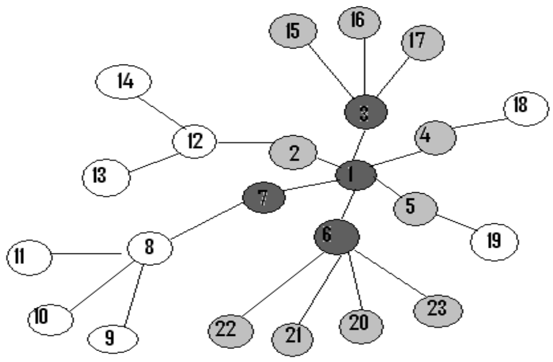


Fig 4.3 (h)

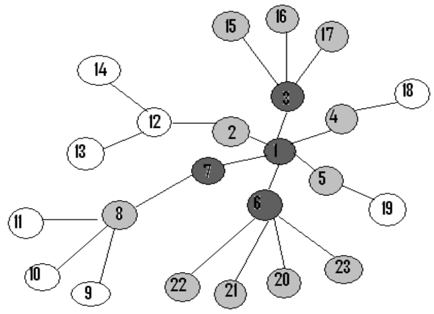


Fig 4.3 (i)

In fig. **4.3 (i)**,the maximum degree node is 8 and is colored black and its neighbors 9, 10, 11 are colored gray as depicted in fig **4.3** (j).

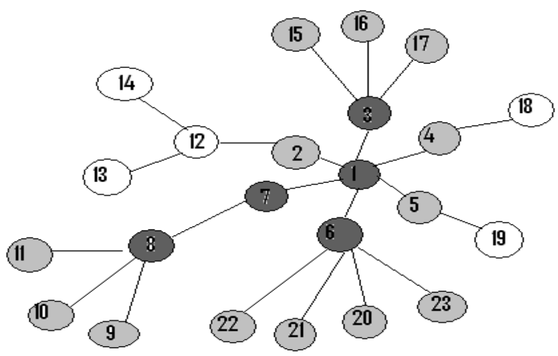


Fig 4.3 (j)

Now, the nodes 2, 4, and 5 have the same degree, thus the node 2 is chosen and colored as black, and its neighbor node 12 is colored gray shown in fig **4.3** (k).Now in this fig, we select node 12 to be the member of MCDS and is colored black, and its neighbors 13 and 14 are colored gray which is shown in fig **4.3** (l).

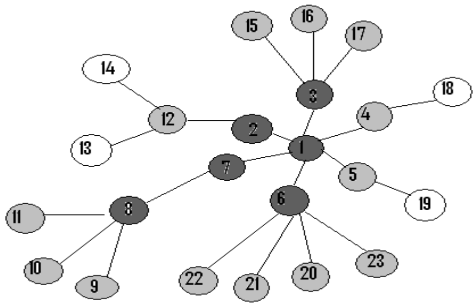


Fig 4.3 (k)

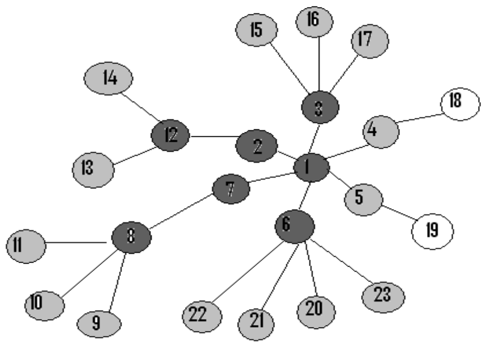


Fig 4.3 (l)

Since the only white nodes are 18 and 19 and the gray nodes connected with them are node 4 and 5 respectively, thus both the nodes are colored black and the nodes 18 and 19 are colored gray.

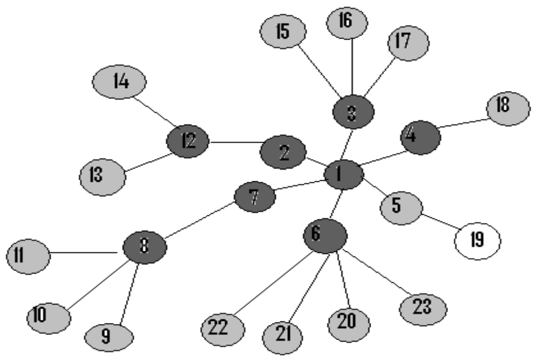


Fig 4.3 (m)

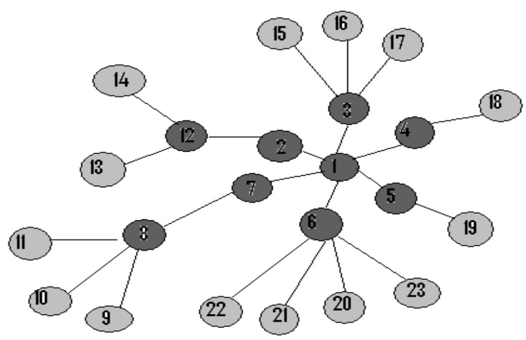


Fig 4.3 (n)

In fig **4.3** (n), all the nodes are either colored black or gray, thus we have obtained the final Minimum Connected Dominating Set (MCDS).Thus the MCDS of the given example of ad-hoc network obtained contains the following nodes: 1, 6, 3, 7, 8, 2, 12, 4, 5.

4.3 Praposed new Improved MCDS algo for DSR protocol

Algorithm 1:

Populating MCDS adjacency list: As soon as a node enter in the network.

Step1: Broadcast IP.
If (Any reply comes from any node, say Y)
Then
{Node will update its MCDS adjacency list.}

Step2: reply will contain address of Y and degree of Y. Accordingly MCDS list of X will be update.

Step3: if degree of nearly added node is >1.then MCDS flag is set to true.

Step4: as soon as new entry is done or a node MCDS list then its degree is increment as one.

Step5: as soon as new node updates its list, it will broadcast its degree.

Algorithm 2:

Message flow in network: this will work on reduced before.

Step1: source node will check for destination address in MCDS adjacency list.
If (address available)
Then {send request packet directly.}

Step2: Else
{Send request packet to the MCDS nodes (node with MCDS flag set).}

Step3: similarly entry node in network will find the packet.

Step4: finally
If (packet reach destination)
Then {destination node generates RREP message (full path) to source.}

Else
{Run algorithm I.}
Step5: Source send DATA packet to destination.

5 Comparison of MCDS over DSR

5.1 Finding of MCDS

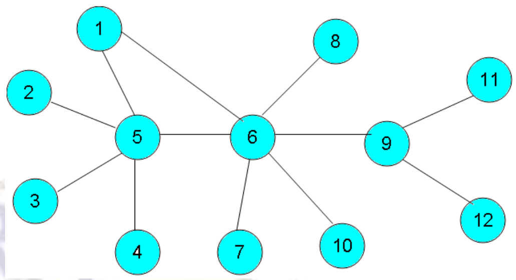


Fig 5.1(a)

Fig 5.1 (a) represents the ad hoc network. Our aim is to find the MCDS members in the given network. Fig 5.1(b) shows the result of MCDS members (colored orange).

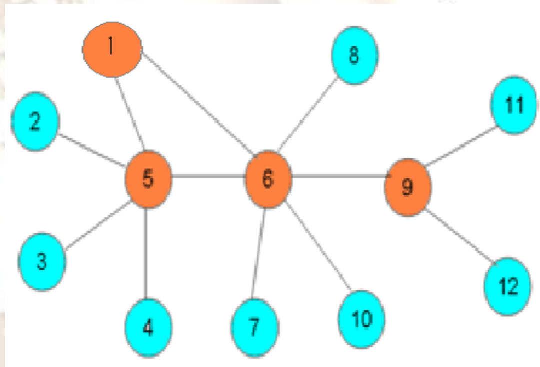


Fig 5.1(b)

5.2 Applying DSR protocol

Here we apply DSR on fig 5.2 (a) .We assume that node 2 is a source node and node 11 is a destination node. We will implement ROUTE DISCOVERY to find the route from source to destination node.

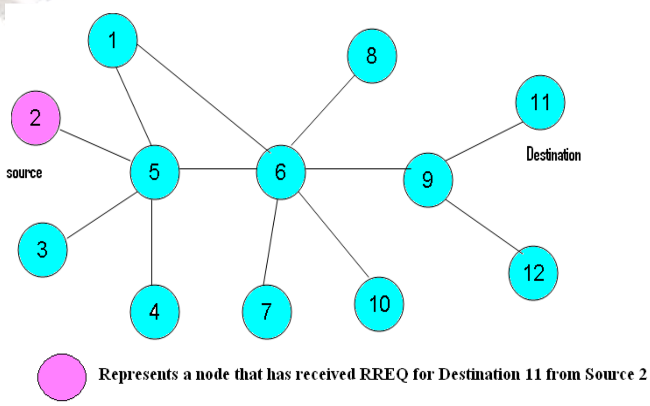


Fig 5.2 (a)

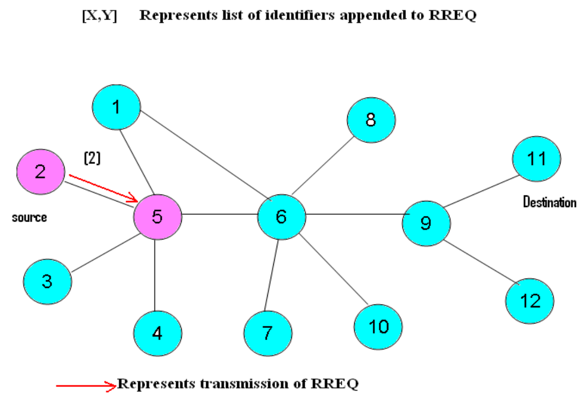


Fig 5.2 (b)

Node 2 broadcast ROUTE REQUEST packet to its neighbor i.e. node 5. The RREQ packet header contains the pair <originating address, request id> as shown in fig 5.2 (b). Each node receiving this packet adds its own address in the header and forwards the packet to its neighbors.

Node 5 broadcasts the RREQ packet to nodes 1 , 2, 3 ,4, 6 as shown in fig 5.2 (c).

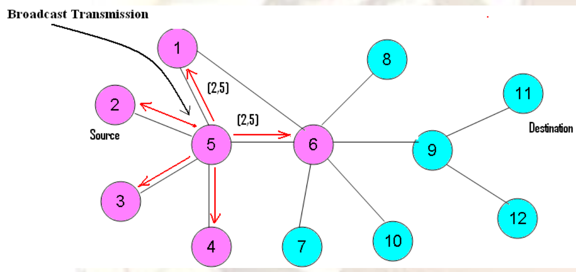


Fig 5.2 (c)

Node 1 forwards the RREQ [2,5,1] to node 6.Node 6 receives the two RREQ packets[2,5] and [2,5,1], one from node 5 and another from node 1.The node then broadcasts both these packets to its neighbor nodes 5,7,8,9,10 as shown in fig. 5.2 (d). Node 5 will discard the packets since it already contains the packet with same request id.

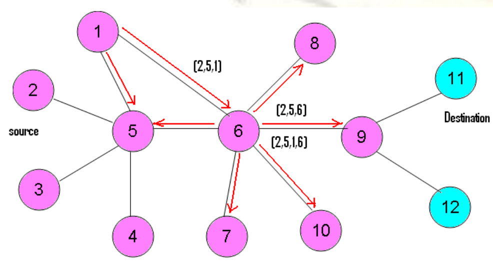


Fig 5.2 (d)

Node 9 sends the RREQ to node 6, 11,12 as shown in fig 5.2 (e).Node 6 discards the packet. Since the destination node 11 receives the RREQ packet, it sends RREP REQUEST REPLY to the source node through the path as described in the packet as shown in fig 5.2 (f).

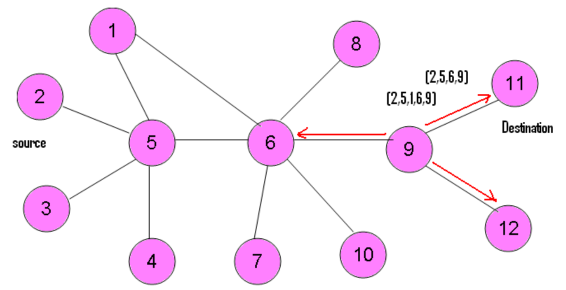


Fig 5.2 (e)

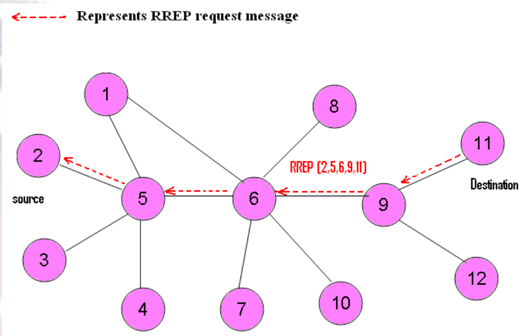


Fig 5.2 (f)

Now, the source node 2 can send the data packet to destination node 11 through the path as came through RREP packet as depicted in fig 5.2 (g)

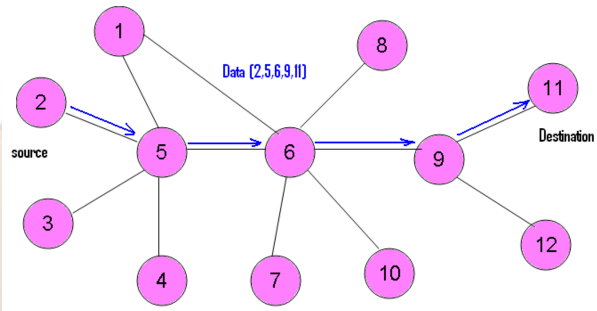


Fig 5.2 (g)

5.3 Implementation of DSR on MCDS

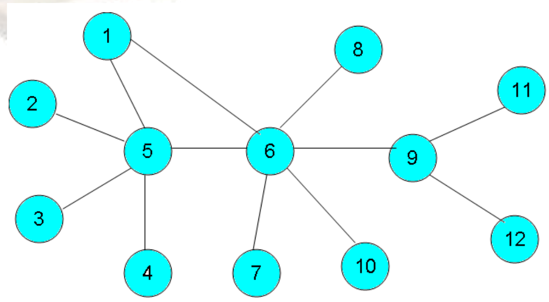


Fig 5.3 (a)

Now, we will implement Dynamic Source Routing on the obtained MCDS and compare the performance.The MCDS members of fig 5.3 (a) have been shown in fig. 5.3(b)

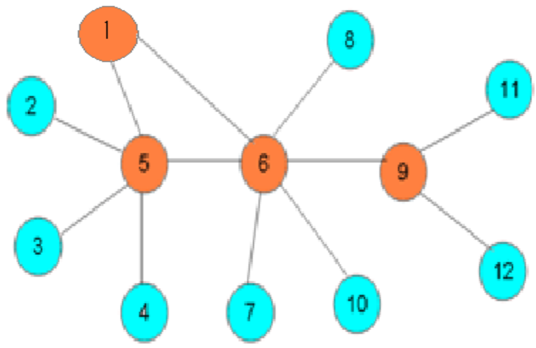


Fig 5.3 (b)

Source node 2 will forward the RREQ packet to its MCDS neighbor node 5 as shown in fig 5.3 (c). Node 5 will forward the packet only to MCDS neighbor node 6 as shown in fig 5.3 (d).

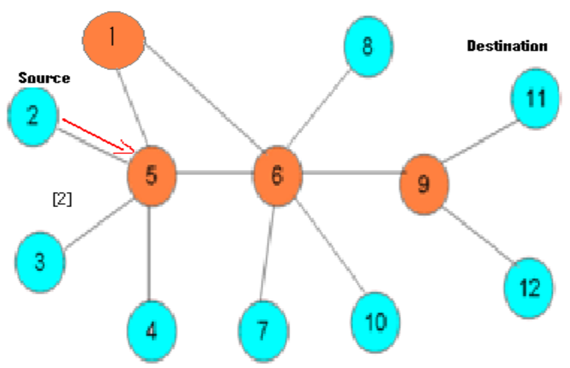


Fig 5.3 (c)

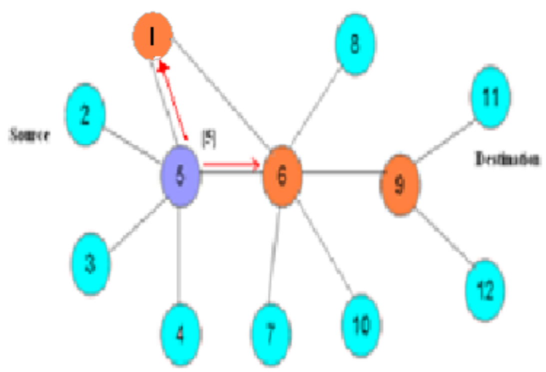


Fig 5.3 (d)

Similarly, node 6 will forward the packet to its MCDS neighbor node 9 adding its own address in packet as shown in fig 5.3(e).Since MCDS node 9 is directly connected to non-MCDS destination node 11,so it sends a RREP message [5,6,9,11] to node 5 through this path as shown in fig 5.3(f).

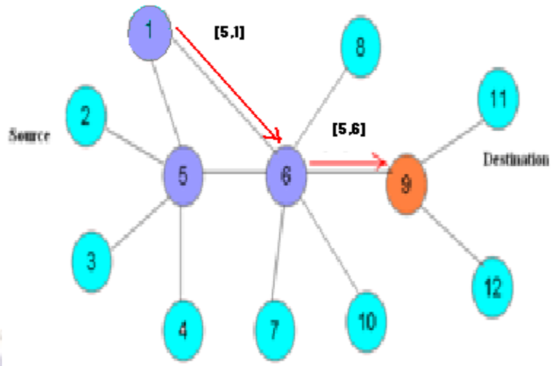


Fig 5.3 (e)

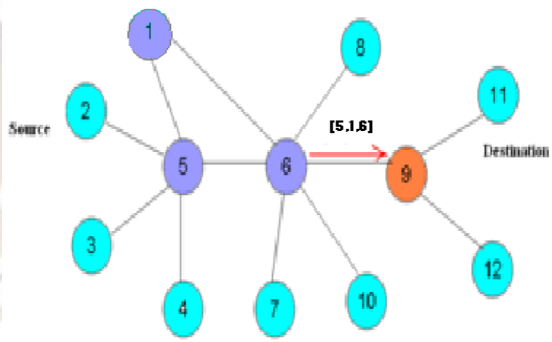


Fig 5.3 (f)

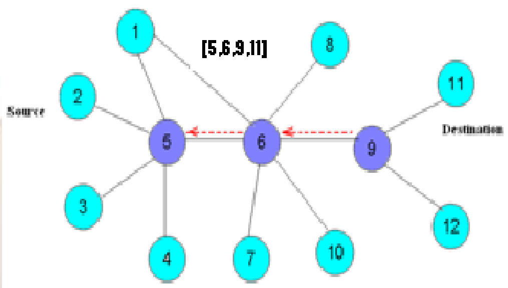


Fig 5.3 (g)

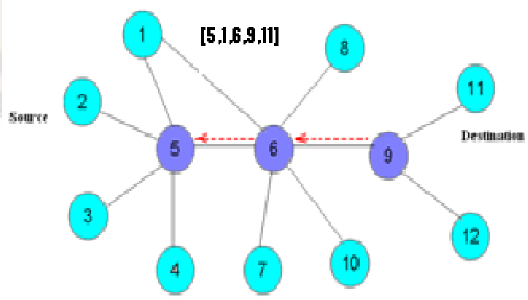


Fig 5.3 (h)

Now, the node 5 forwards the reply to the source node 2 as shown in fig 5.3(g).Node 2 will send the data packet to node 11 through this path .

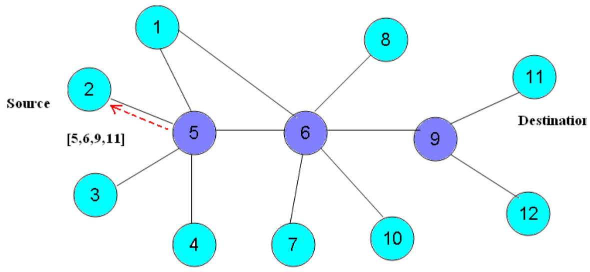


Fig 5.3 (i)

5.4 Conclusion

When the performance of DSR, in terms of packet loss rate and routing protocol overhead, is compared with that of three other routing protocols designed for use in multi-hop ad hoc networks, DSR significantly outperforms the other protocols across a wide range of scenarios.

5.4.1 Various advantages of applying DSR on MCDS are:-

1. Total number of Nodes:- Fig 5.3(g) shows that the number of nodes in reaching from source to destination can be reduced

2. Time:- The above example helps to reduce the extra time required during route discovery to find the path from source to destination .Instead the path is already maintained by the MCDS members of the graph.

3. Broadcasting:- The Source need not broadcast the route request to all its neighbors, instead it can send the request only to the MCDS neighbors.

4. Bandwidth consumption:- The packet header size grows with the route length resulting in more bandwidth utilization. Thus, by applying DSR over MCDS, there is a significant reduction in total number of nodes, thereby reducing the bandwidth consumption.

REFERENCES

[1] A Distributed approach for computing the minimum connected dominating set in ad hoc networks *by Kais Mnif , Bo Rond and Michel Kadoch*

[2] David B. Johnson, David A. Maltz, and Josh Broch. DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. In *Ad Hoc Networking*, edited by Charles E. Perkins, chapter 5, pages 139–172. Addison-Wesley, 2001.

[3] David B. Johnson, David A.Maltz, Yih-Chun Hu, and Jorjeta Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsr-04.txt, November 2000. Work in progress.

[4] David A.Maltz. Resource Management in Multi-hop Ad Hoc Networks. Technical Report CMU CS TR00-150, School of Computer Science, Carnegie Mellon University, November 1999. Available from http://www.monarch.cs.cmu.edu/papers.html.

[5] M. Scott Corson and Anthony Ephremides. A Distributed Routing Algorithm for Mobile Wireless Networks. *Wireless Networks*, 1(1):61–81, February 1995.

[6] National Science Foundation. Research Priorities in Wireless and Mobile Communications and Networking: Report of a Workshop Held March 24–26, 1997, Airlie House, Virginia. Available at http://www.cise.nsf.gov/anir/ww.html.

[7] RajendraV. Boppana and Satyadeva P. Konduru. An Adaptive DistanceVector Routing Algorithm for Mobile, Ad Hoc Networks. In *Proceedings of IEEE INFOCOM 2001*, pages 1753–1762, Anchorage, Alaska, April 2001.

[8] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std 802.11-1999. The Institute of Electrical and Electronics Engineers, New York, New York, 1999.

[9] John Jubin and Janet D. Tornow. The DARPA Packet Radio Network Protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.