Performance of Dominating and Adaptive Partial Dominating Sets in AODV Routing Protocol for MANETs

A. Nagaraju *, Dr. S. Ramachandram**, B. Eswar***
* Head & Associate Prof in I.T , Kamala Institute of Tech & Science, Huzurabad, Karimnagar, A.P
** Head & Prof in CSE , Osmania University , Hyderabad. A.P, INDIA
***Assistant Prof in I.T, Kamala Institute of Tech & Science, Huzurabad, Karimnagar, A.P. INDIA

Abstract— **A mobile ad hoc network (MANET) is a wireless network that does not rely on any fixed infrastructure (i.e., routing facilities, such as wired networks and access points), and whose nodes must coordinate among themselves to determine connectivity and routing. The broadcast can target a portion of the network (e.g. gathering neighborhood information), or the entire network (e.g., discovering routes on demand). Broadcasting of signaling and data in MANETs raise redundant transmission of control packets to overcome these problems we applied dominating set and Adaptive partial Dominating (APDP) approach to existing routing protocols such as Ad-hoc On-demand Distance Vector (AODV). The focus of this paper is to apply the concept of DS and APDP to AODV and evaluate the performance of dominating sets in AODV that improve broadcasting, End-to-End Delay, Network load, Packet Latency, and also maintains secure packet transmission.**

IndexTerms—**Adaptive partial dominating, Dominating sets, AODV**

I. Introduction

Mobile IP and wireless networks accessing the fixed networks have provided support for the mobility. But it is still restrictive in forcing the connectivity at least to the core network. It puts impediments on supporting the true mobility in the network. In this connection, one area which is getting much attention in last couple of years is Mobile Ad Hoc Networks (MANETs). A MANET (Mobile Ad-Hoc Network) is a type of ad-hoc network with rapidly changing topology. ***S***ince the nodes in a MANET are highly mobile, the topology changes frequently and the nodes are dynamically connected in an arbitrary manner.

As defined in [1] a MANET is an autonomous system of mobile nodes. MANET nodes are equipped with wireless transmitters and receivers using antenna which may be omni directional, highly-point-to-point, possibly steerable, etc. At a given point in time, depending on the positions of the nodes and their transmitters and receivers coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multi hop graph or ad hoc network exists among the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters. There is current and future need for dynamic ad hoc networking technology. The emerging field of mobile and nomadic computing, with its current emphasis on mobile IP operation, should gradually broaden and require highly-adaptive mobile networking technology to effectively manage multi hop, ad-hoc network clusters which can operate autonomously or, more than likely, be attached at some point(s) to the fixed Internet.

There are two broad categories of unicast routing protocols for MANETs, proactive and reactive. With *proactive routing* (e.g., OLSR [21]), nodes keep routing information to all nodes in the network, not subject to any existing data flow. OLSR is a link state protocol using an optimized broadcast mechanism for the dissemination of link state information. In *reactive routing* (e.g., AODV [3]), routes are found on demand and nodes find routes to their destinations as they are needed. Route discovery starts by broadcasting a *route request* (RREQ) message throughout the network. This message is relayed until it reaches a node with a valid route to the destination, or the destination itself. Once this happens, a *routereply* (RREP) message is sent back to the source by reversing the path traversed by the RREQ message. Only after receiving the corresponding RREP message can the source start sending packets to the destination. Reactive and proactive routing can be combined, resulting in *hybrid protocols* (e.g., ZRP 20]). In this case, routes to some nodes (usually the nearest ones) are kept proactively, while routes to the remaining nodes are found on-demand.

Neighbor-knowledge-based methods mainly depend on the following idea: To avoid flooding the whole network, a small set of forward nodes is selected such that the forward node set forms a connected dominating set (CDS). A node set is a connected dominating set if every node in the network is either in that set or the neighbor of a node in that set. Then the challenge is to select a small set of forward nodes in the absence of global network information

In our proposed paper we focused on applying APD to AODV and to evaluate the performance of Dominating sets in Ad-hoc On Demand Distance Vector Routing algorithm(AODV) that improve broadcasting, End-to-End Delay, Network load, Packet Latency, and also Maintains secure packet transmission . To do this, concepts from ***domination in graphs*** have been explored (i.e.

doi:10.4304/jetwi.2.2.80-85

Dominating sets), The rest of the paper is organized as follows. Section 2 is the related work. Section 3 comprises Dominating Sets (Domination in graph theory) as in [9]. Section 4 is Route Request Algorithm using Dominant Pruning, Section 5 presents simulation results of this method and Section 6 presents Conclusions and future scope.

II. Related Work

Several broadcasting techniques have been proposed, differing among each other on the heuristics applied to reduce the redundancy on broadcast transmissions. Broadcasting protocols can be categorized into the following four classes [1]:

Blind flooding [9]: Each node broadcasts a packet to its neighbors whenever it receives the first copy of a broadcast packet; therefore, all nodes in the network broadcast the packet exactly once.

Probability-based methods [12]: A node re-broadcasts a packet with a given probability p (if $p = 1$, we have blind flooding).

Area-based methods [12]: A node broadcasts a packet based on the information about its location and the location of its neighbors (e.g., if a node receives the packet from a neighbor really close to it, probably it will not reach other nodes other than the nodes reached by the first broadcast).

Neighbor information methods [15]: In these methods, a node has partial topology information, which typically consists of the topology within two hops from the node (two hop neighborhood). There are two main classes of methods in this category. In a *neighbor designated method* a node that transmits a packet to be flooded specifies which one-hop neighbors should forward the packet. In a *self-pruning method* a node simply broadcasts its packet, and each neighbor that receives the packet decides whether or not to forward the packet. Williams and Camp [1] have shown that *neighbor information* methods are preferred over other types of broadcast protocols. Between the two classes of neighbor information methods, Lim and Kim [6] show that the simplest form of neighbor-designated algorithm out performs the simplest form of self-pruning, and Wu and Dai [7] show that an improved self-pruning technique outperforms the most efficient neighbor-designated algorithm based on the two-hop neighborhood information

Dominating sets play a major role in deciding the forwarding list in neighbor designated algorithms. Extensive work has been done on finding good approximations for computing the *minimum cardinality* CDS (MCDS). An algorithm with a constant approximation of eight has been proposed by Wan et al. [13]. However, their approach requires that a spanning tree to be constructed first in order to select the dominating nodes (forwarding nodes), and only after the tree has been constructed a broadcast can be performed.

The forwarding nodes are selected using the *greedy set cover* (GSC) algorithm. GSC recursively chooses one-hop neighbors that cover the most two hop neighbors, repeating the process until all two-hop neighbors are covered. The identifiers (IDs) of the selected nodes are piggy-backed in the packet as the forwarding list. A receiving node that is requested to forward the packet again determines the forwarding list.

In our proposed method, we apply Dominating Set model to identify the best RREQ forwarding nodes among the existing neighbors. Here the RREQ are transferred using the forwarder list information. This kind of mechanism controls the overhead of Route Request Phase (RREQ) of AODV by eliminating the redundant RREQ forwarding towards the destination.

III. Dominating Sets (Domination in Graph Theory)

As our project involves the computation of dominating sets, we provide a brief introduction to domination in graph theory below.

In our notation, the undirected graph G = (V, E) consists of a set of vertices V represents a set of wireless mobile nodes and E represents a set of edges. A set $D \subseteq V$ of vertices in graph G called a dominating set (DS) if every n_i Є V either an element of D or is adjacent to an element of D [15]. If the graph induced by the nodes in D is connected, we have a connected dominating set (CDS). The problem of computing the minimum cardinality DS or CDS of any arbitrary graph is known to be NP-complete [15].

In dominant pruning (DP) the sending node decides with adjacent nodes should relay the packet. The relaying nodes are selected using the distributed CDS algorithm, and the identifiers (IDs) of the selected nodes are piggybacked in the packet forward list. A receiving node that is requested to forward the packet again determines the forwarder list. The flooding ends when there is no more relaying nodes.

A. Dominant Pruning Algorithm

Selection Process:

Step 1 Let F(u,v) = [ ] (empty list)

Z = Ø (empty set) and K = U S_i

Where S_i = N(v_i) ∩U(u,v) for , v_i Є B(u,v)

Step 2 Find Set S_i whose size is maximum in K

(In case of a tie, the one with smallest identification I selected)

Step 3 F(u,v) = F(u,v) || v_k , Z = Z U S_i , K = K - S_i and

S_j = S_j- S_i for all S_j Є K

Step 4 If Z = U(u,v) exit ; Otherwise , go to Step 2

Nodes maintain the information about their two-hop neighborhood, which can be obtained by the nodes exchanging their adjacent node list with their neighbors. DP is the distributed algorithm that determines a set cover based

B. Route Request Algorithm Using DP Algorithm

On-demand route discovery is based on *route request* (RREQ) and *route reply* (RREP) messages (e.g., AODV [3] and DSR [4]). The way in which these messages are handled may differ among different protocols, but their functionality remains the same: a request is relayed until it reaches a node with a valid route to the destination or the destination itself, which triggers a reply message sent back to the originator. Several parameters (such as how long to keep requests in a cache, timeouts for requests, timeouts for hellos) are subject to tuning, and the choices made may result in improvements in the protocol performance. However, RREQs are propagated using either an unrestricted broadcast or an expanding ring search. In either case, the resulting flooding operation causes considerable collisions of packets in wireless networks using contention-based channel access.

In addition to applying DP to reduce the number of nodes that need to propagate RREQs transmitted on broadcast mode, information regarding prior routes to a destination is used to unicast RREQs to a region close to the intended destination, so that broadcast RREQs are postponed as much as possible and occur (if necessary) only close to the destination, rather than on network-wide basis. This RREQ Algoritm presents the pseudo-code for the modified RREQ. A route request (RREQ) is handled as follows:

• If the source of a RREQ does not have any previous knowledge about the route to the destination or is retrying the RREQ, it calculates its forwarder list using DP, and broadcasts the packet (Lines 8, 9, and 14).

• On the other hand, if the source of a RREQ has knowledge about a recently expired route to the destination, and there is a valid route to the next hop towards the destination (Lines 2, 3, and 4), the node calculates the forwarder list using DP (Line 9), but instead of broadcasting the RREQ packet, the node unicasts the packet to the last known next hop towards the destination (Line 12).

Upon receiving a route request, a forwarder that cannot respond to this request calculates its own forwarder list using the information provided in the RREQ packet (i.e., forwarder list, second to previous forwarder list, and source node) and broadcasts or unicasts the packet (depending on which one of the two first cases apply) after updating it with its own forwarder list.

RREQ Algorithm

Data : n_i, destination D, B_i, U_i
Result : Unicast the RREQ, or Broadcast the RREQ
Begin

```
1   if recently expired route to D and not retrying then
2   NextHop ← previous_nextHop(D)
3     if validRoute(NextHop) then
4        result←Unicast
5     else
6        result←Broadcast
7   else
8      result←Broadcast
9   Fi←DP( ni, Bi, Ui )
10  Update RREQ packet with Fi
11  if result== Unicast then
12     Unicast the RREQ packet to NextHop
13  else
14     broadcast the RREQ packet
    end
```

Eventually, the RREQ reaches a node with a route to the destination or the destination itself. Our approach attempts to reduce the delay of the route discovery by unicasting a RREQ towards the region where the destination was previously located. The success of this approach depends on how fresh the previous known route to the destination is, and how fast the destination node is moving out of the previous known location. If an intermediate node has completely removed any route to the destination, the RREQ is then broadcasted. The intended effect is to postpone the broadcast of a RREQ to the region closest to the destination. In the case that the unicast approach fails, or there is no previous route to the destination, the source broadcasts by default.

Because of topology changes, nodes may not have correct two-hop neighborhood information, which may result in forwarding lists that do not cover all nodes in the neighborhood. However, this is not a major problem when the request is broadcasted, because a node incorrectly excluded from the forwarder list may also receive the request and is able to respond in the case it has a route to the destination.

C. Enhanced Dominant Pruning Algorithm

In their paper [6], wei Lou and Jie Wu proposed two enhanced dominant pruning algorithms: the Total Dominant pruning (TDP) algorithm and Paratial Dominant Pruning (PDP).

In TDP algorithm, if node v can receive a packet piggybacked with N(N(v)) from node u , the 2-hop neighbour set that needs to be covered by v's forward node list F is reduced to U = N(N(v)) – N(N(u)). The main objective of the TDP algorithm is that 2-hop neighbourhood information of each sender is piggybacked in the broadcast packet which results in consumption of more bandwidth.

D. Partial Dominant Pruning Algorithm:

Just like the DP algorithm, in PDP, no neighbourhood information of the sender is piggybacked with the broadcast packet. Apart from excluding N (u) and N (v) from N (N (v)) as in the DP algorithm, we can here exclude some more nodes from neighbours of each node in N (u) ∩ N (v). Such a node set is denoted by P (u,v) (or simply P) = N(N(u) ∩ N(v)). Then 2-hop neighbour set U in the PDP algorithm can be given by U = N(N(v)) – N(u) – N(v) – P since P is a subset of N(N(u)), we can easily see P can be excluded from N(N(v)). Also it can be proved that U is subset of N(B), when P = N(N(u)∩N(v)), U = N(N(v)) – N(u) – N(v)-P and B = N(v) – N(u).

E Adaptive Partial Dominant Pruning Algorithm[19]

Adaptive dominant pruning algorithm (APDP) is similar to PDP. However, besides excluding N (u), N (v) and P from N (N (v)) as mentioned in PDP algorithm, adjacent nodes of U are eliminated from U.

APDP Algorithm

1. Node v uses N(N(v)) ,N(u), and N(v) to obtain P = N (N(u) ∩ N(v)), U^1 = U – E where U = N(N(v)) – N(u) – N(v) – P and E is the set of equivalent and adjacent nodes in U and B = N(v) – N(u)
2. Node v calls the selection process to determine F.

Now consider Fig 1 in the example of sample ad-hoc network with 12 nodes. Table 1 shows each node in Fig 1 1-hop and 2-hop neighbor nodes. Here we illustrate the difference between PDP and APDP algorithms.

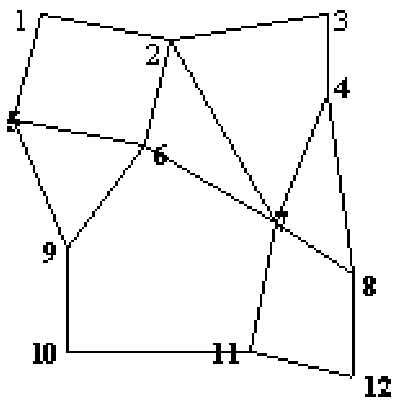

Fig 1 An Ad-hoc Network with 12 nodes.

Table 1: Two hop neighbors of each node

V	N(v)	N(N(v))
1	1,2,5	1,2,3,5,6,7,9
2	1,2,3,6,7	1,2,3,4,5,6,7,8,9,11
3	2,3,4	1,2,3,4,6,7,8
4	3,4,7,8	2,3,4,6,7,8,11,12
5	1,5,6,9	1,2,5,6,7,9,10
6	2,5,6,7,9	1,2,3,4,5,6,7,8,9,10,11
7	2,4,6,7,8,11	1,2,3,4,5,6,7,8,9,10,11,12
8	4,7,8,12	2,3,4,6,7,8,11,12
9	5,6,9,10	1,2,5,6,7,9,10,11
10	9,10,11	5,6,7,9,10,11,12
11	7,10,11,12	2,4,6,7,8,9,10,11,12
12	8,11,12	4,7,8,10,11,12

For PDP algorithm, node 6 again has same forward node list F (Ø,6) = [7,2,9]. From P(6,7) = {1,3,6,7 }, we have U(6,7) = N(N(7)) – N(6) – N(7) – P(6,7) = { 10,12 }. The forward node list for 7 is F(6,7) = {11 }, Similarly , from P(6,2)= { 2,4,6,8,11} , we have U(6,2) = N(N(2)) – N(6) –N(2) –P(2,6) = Ø and , then , F(6,2) = {10 }. Therefore, the total number of forward nodes is 1+3+2 = 6. The details of P, U, B and F are represented in the following Table 2.

The total number of forwarding nodes according to PDP is in the give example is 6 including source node i.e. { 6,2,7,9,11,10 }.

In our proposed model APDP, an enhanced version of PDP, the definition of existing U has been broadened to a new U to check and exclude if it contains any adjacent nodes. The results of the proposed model are presented in Table 3

Table 2: PDP algorithm

u	v	P	U	B	F
Ø	6	Ø	1,3,4,8,10,11	2,5,7,9	7,2,9
6	7	1,3,6,7	10,12	4,8,11	11
6	2	2,4,6,8,11	Ø	1,3	[ ]
6	9	1,6,9	9	10	10
7	11	Ø	9	10,12	10
9	10	Ø	7,12	11	11

Table 3: APDP algorithm

U	V	P	U	B	F
Ø	6	Ø	1,4,8,11	2,5,7,9	7,2
6	7	1,3,6,7	10,12	4,8,11	11
6	2	2,4,6,8,11	Ø	1,3	[ ]
6	9	1,6,9	9	10	10
7	11	Ø	9	10,12	10
9	10	Ø	7,12	11	11

The lower bound as per the AMCDS (Approximation Minimum Connected Dominating Set) reduces the minimum connected dominating set to {2, 6, 7, 11} minimizing the number of forward nodes to 4. According to our proposed model number of total forwarding nodes including source node is 5 i.e. {2, 6, 7, 10, 11} where as it is 6 in PDP.

IV. SIMULATIONS RESULTS AND SIMULATION PARAMETERS

We used the simulator glomosim-2.03,[8] to run the simulation Table 3 summarizes the simulation parameters we used. The simulation time was 15 minutes according to simulator clock. A total of 45 nodes were randomly placed in field of 500 X 500 m^2 and in field of 2000 X 2000 m^2. Power range of each node is 250m. We performed the simulation for AODV with DP algorithm.

Table 3 Simulation Parameters

Parameter	Value	Description
Number of nodes	160 and 40	Simulation Nodes
Field range x	500m and 2000	X-Dimension
Field range y	500m and 2000	Y-Dimension
Power range	250m	Nodes power range
Mac protocol	IEEE 802.11	MAC layer protocol
Network Protocol	AODV &Rough AODV	Network Layer
Transport Layer Protocol	UDP	Transport Layer
Propagation function	FREE-Space	Propagation Function
Node placement	Random	Nodes are distributed in random manner
Simulation time	15M	According to simulation clock
Mobility Interval	10-30sec	Pause time of node

We run each simulation four times with different node pause time varying from 15 to 30sec with a step interval of 5 sec. The graphs are presented in results section.

Results

The performance of proposed protocol is evaluated using the following metrics:

Packet delivery ratio (Throughput): Packet delivery fraction is the ratio between the number of packets originated by the application layer CBR sources and the number of packets received by the CBR sinks at the final destinations. Packet delivery Ratio is higher for AODV with Dominating Sets than that of conventional AODV (figure 3 & 7)

Number of Control Packets: The total number of control packets occurred by different nodes is less in AODV with Dominating Sets than that of conventional AODV(figure 4 & 8).

Number of Route requests: It is the number of control packets generated by all the nodes in the simulation. The number of Route Request packets is less in AODV with Dominating Sets (figure 2 & 6) compare to conventional AODV.

End-to-End Delay: The End-to-End Delay of AODV with Dominating Sets is better when compared with conventional AODV (figure 5 & 9).

Performance of AODV with Dominatin sets and AODV in the Terrain Dimensions (500, 500)

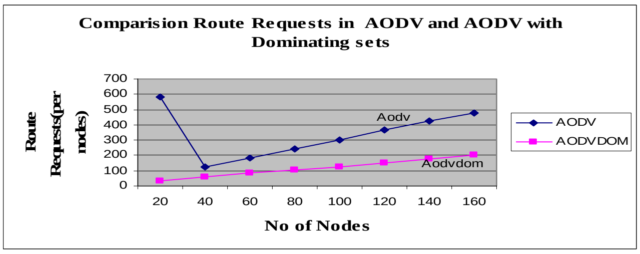

Fig 2 Route Request packet Comparison

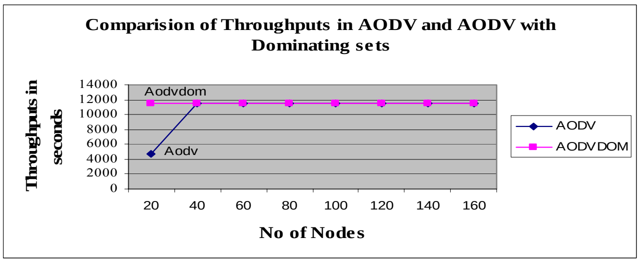

Fig 3 Throughput comparison

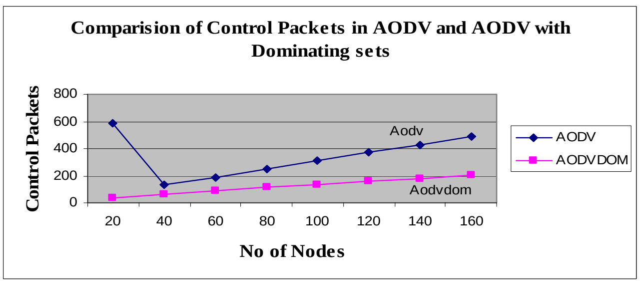

Fig4.Control Packets Comparison

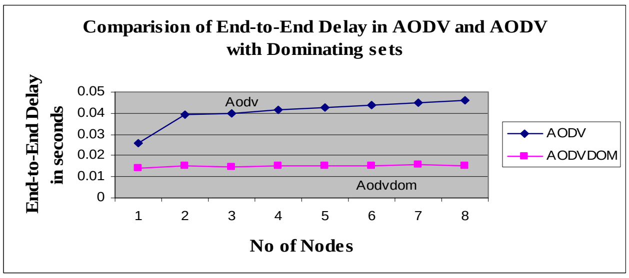

Fig 5 End-to-End Delay in Sec

Performance of AODV with dominating sets and AODV in the Terrain (2000, 2000) up to 40 nodes

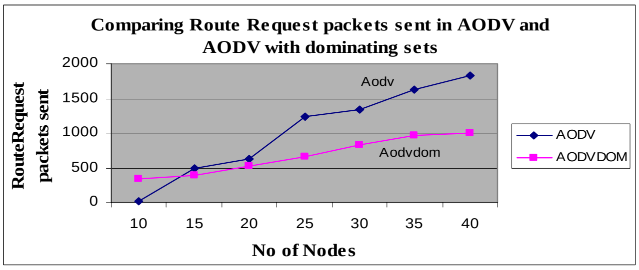

Fig 6 Route Request packet Comparision

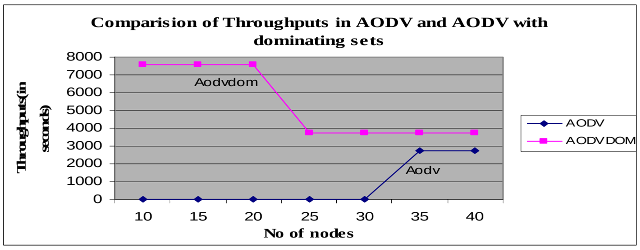

Fig 7 Throughput comparison

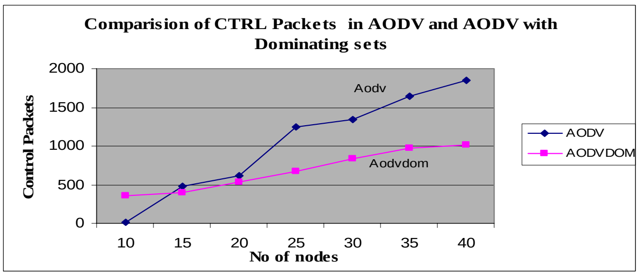

Fig 8.Control Packets Comparison

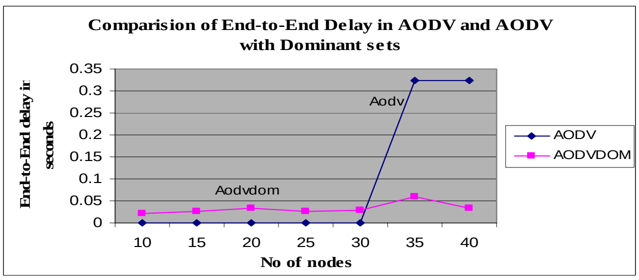

Fig 9 End-to-End Delay in Sec

V. Conclusions and Future Work

We presented an enhanced dominant pruning approach that allows pruning redundant broadcasts even more than the conventional dominant pruning heuristic. Redundant broadcasts increase the number of packet collisions, and consequently delay the response for RREQs in the route discovery process. DP is shown to reduce the number of broadcast transmissions when compared to standard DP. Because DP requires the two-hop neighborhood to determine the forwarder list, we built a neighbor protocol as part of AODV. By making the neighbor protocol part of AODV, the result is a more accurate view of the local topology, and therefore more accurate is the determination of the forwarder list. We also proposed a APDP algorithm which is an enhanced version of PDP. Here we have shown the performance comparison of AODV with DP algorithm. Future scope of the work is to compare to the performance of AODV with PDP algorithm and compare the performance of both.

References

[1] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile adhoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international symposiumon mobile ad hoc networking & computing*, pages 194–205. ACM Press, 2002.

[2] C. Perkins. Ad-hoc on-demand distance vector routing. In *Proceedings of the IEEE workshop on mobile computing systems and applications*, pages 90–100, 1999.

[3] Charles E. Perkins and Elizabeth M.Royer. Ad-hoc on-demand distance vector routing. Technical report, Sun Micro Systems Laboratories, Advacnced Developmenet Group, USA.

[4] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks.In *Mobile Computing*, volume 353, pages 153–179. Kluwer Academic Publishers,1996.

[5] Dipti Joshi, Sridhar Radhakrishnan, and Chandrasekheran Narayanan. A fast algorithm for generalized network location problems. In *Proceedings of the ACM/SIGAPP symposium on applied computing*, pages 701–8, 1993

[6] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*, 24(3–4):353–363, February 2001

[7] Jie Wu and Fei Dai. Broadcasting in ad hoc networks based on self-pruning. In *IEEE INFOCOM*, pages 2240–2250, 2003.

[8] Jorge Nuevo, "A comprensible glomosim tutorial", March4,2004 http://appsrv.cse.cuhk.edu.hk\~hndai\research.

[9] Katia Obraczka, Kumar Viswanath, and Gene Tsudik. Flooding for reliable multicast in multi-hop ad hoc networks. *Wireless Networks*, 7(6):627–634, 2001.

[10] Marco A. Spohn and J. J. Garcia-Luna-Aceves. Enhanced dominant pruning applied to the route discovery process of on-demand routing protocols. In *Proceedings of the 12th IEEE international conference on computer communications and networks*, pages 497–502, October 2003

[11] Mobile Ad Hoc Networking:Routing Protocol Performance Issues and Evaluation Considerations Request For Comments 2501 available at http://www.ietf.org

[12] Ning.P. and Sun.K. How to misuse aodv: a case study of insider attacks against mobile ad-hoc routing protocols. Technical report, Comput. Sci. Dept., North Carolina State Univ., Raleigh, NC, USA, 2003.

[13] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *IEEE INFOCOM*, pages 1597–1604, June 2002.

[14] Rajive Bagrodia. README GloMoSim Software. University of California, Los Angeles - Department of Computer Science. Box 951596, 3532 Boelter Hall, Los Angeles-CA 90095-1596 / rajive@cs.ucla.edu.

[15] Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater, editors. *Fundamentals of Domination in Graphs*. Marcel Dekker, Inc., 1998.

[16] Xiaoyan Hong, Kaixin Xu, and Mario Gerla. Scalable routing protocols for mobile ad hoc networks. 2002.

[17] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8:153–167, 2002. H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications*,24(3–4):353–363, February 2001-04-15).

[18] Z. Haas. A new routing protocol for the reconfigurable wireless network. In *Proceedingsof the IEEE international conference on universal personal communications*, pages 562–566, October 1997.

[19] A Nagaraju , Dr S.Ramachandram "Adaptive Partial Dominating Set Algorithm For Mobile Ad-Hoc Networks" , Presented in the international conference ACM ,COMPUTE-09, Jan 9th -11th,2009,Banglore.

[20] J.Hass and M.R.Pearlman, "The Zone Routing Protoco(ZRP) for Ad Hoc Networks," draft-ietf-manet-zone-zrp-02.txt,work in progress,June 1999

[21] T.Clausen, P.Jacquet,A.Laouiti, P.Muhlethaler,A. Q and L.Viennot. Optimized link state routing protocol for ad-hoc networks. In Proceedings of the IEEE internat multi topics conferences. 2001