Emergence of Ad-hoc On-demand Distance Vector Protocol (AODV) as an Efficient On-demand Routing Protocol (EORP) Using Bayesian Approach

Dr. Anubhuti Khare[1], Manish Saxena[2], Raghav Shrivastava*[3]
1. Dr. Anubhuti Khare, Reader, Department of Electronics and Communication, University Institute Of Technology, Rajeev Gandhi Technical University, Bhopal,
2. Manish Saxena, Head of Electronics and Communication Department, Bansal Institute Of Science and Technology Bhopal,
3. Raghav Shrivastava, Student, Mtech (Digital Communication), B.I.S.T. Bhopal.

Abstract— **In this paper, we are converting AODV (Ad-hoc On-demand Distance Vector) protocol in to an efficient routing algorithm called EORP (Efficient on-demand routing protocol) for mobile ad-hoc networks with a route establishment technique using Bayesian approach. A mobile ad-hoc network (MANET) is a self-configuring infrastructure less network of mobile devices connected by multihop-communication paths or wireless links with no fixed administrative support. One of the typical routing methods in mobile ad-hoc networks use on-demand distance vector, e.g. Ad-hoc On-demand Distance Vector (AODV). The major issue in such a protocol is the route establishment cost. The initial results show that there is significant improvement in delivery ratio, control packets overhead w.r.t. mobility and control packet overhead w.r.t. network size.**

***Index Terms*— Bayesian, Mobile, Networks, Routing, protocol.**

I. INTRODUCTION

A Mobile Ad hoc Network (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies allowing people and devices to inter-network without
any preexisting communication infrastructure. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic

unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet or we can say mobile ad-hoc networks (MANETs) are used typically in environments where there is no fixed infrastructure that can be deployed. On top of lack of infrastructure, there is also the challenge of ensuring that communication between any two nodes is still possible, even if the mobile nodes themselves are moving. One of the simple ways for routing is to send packets to the destination from the source node through intermediate nodes using the geometric information of all the nodes in the network. Getting accurate geometric information is still not easy. Another technique is to determine the route by means of actively asking all the neighbors and their neighbors for information regarding path to the destination.
For computing the route from source to destination, many of the protocols like AODV [1] flood in all directions, while there are some protocols like FRESH [2] which employ a directional search. In FRESH, historic information regarding when two nodes have been in direct contact is maintained in the form of encounter ages. By maintaining time information also, they were able to steer the direction of the route requests. In our approach, we are improving the performance of route discovery by improving the cost of route establishment using a history based Bayesian method, along with the relative region of the destination node.
In case of on-demand routing, the source of overhead comes from route establishment and maintenance. The relative costs of these two components vary from one protocol to another. Whenever a route has to be discovered, the protocols have to perform some form of flooding of route request packets until the destination node is reached. Route maintenance involves re-establishment of a route, especially in the scenario of link failure or node failure.

II. RELATED WORK

In table-driven routing protocols, such as Destination Sequenced Distance-Vector (DSDV) [3], every node maintains a routing table consisting of topology information that is updated frequently using flooding. Unlike table-driven routing protocols, the on-demand routing protocols determine the path to node during the connection establishment process, alleviating the need to store entire topology in each node. These protocols don't need to send periodic beacon messages to exchange route information. Dynamic Source Routing (DSR) [4] protocol

which is an on-demand protocol establishes a route to the destination by flooding Route Request packets in the networks.

III. EFFICIENT ON–DEMAND ROUTING PROTOCOL

A. Introduction

The proposed Efficient On–demand Routing Protocol for MANET, using Bayesian Approach (EORP) is a novel way of finding route to the destination based on summing up the probabilities (affinity) of each node towards the particular destination, which is calculated using Bayes Theorem[5][6]. The protocol also makes sure that the data travels through shortest route only, thereby minimizing the time delays between sending and receiving of data packet. Another important feature of EORP is that it sends data through two disjoint paths, and the data is sent through both the paths alternatively. This is done so as to reduce traffic through one path and also to avoid the loss of battery power that may occur a result of standby mode of nodes in second path, as they might only be waiting for the data packet to be received. In EORP, each node maintains a history table, which contains the destination node's id and the value of the attributes used for calculating the affinity, along with the status whether a route reply (RREP) was received or not for every route request (RREQ) sent. This history is used in calculating own affinity while sending or rejecting a RREQ. Each intermediate node upon receiving RREQ checks their routing table (RT) to find if the path to the destination is known or not. If known, a RREP is generated back to the node which generated the RREQ otherwise, node first compares the hop count in RREQ with last known hop count for same destination. If hop count in RREQ is higher; the RREQ is discarded (to ensure minimum hops route). Then node compares the stored affinity value for that particular destination in their RT with the affinity contained in RREQ. If affinity in RT is greater, RREQ is rejected (to ensure only highest affinity requests are forwarded and stale routes are avoided) otherwise node will add its own affinity in the RREQ and will broadcast the RREQ. Since the affinity of destination for itself will be 1(highest), hence, upon receiving a RREQ, destination replies back by adding 1 to the affinity contained in the RREQ. Upon receiving RREP, intermediate nodes will again check their RT to see if route affinity in RREP is higher than RT affinity. If it is less, RREP is rejected, otherwise it sent to the node from which it received RREQ with highest affinity. When a route reply is received by the source, it accepts best two RREPs based upon the affinity contained in them.
Upon route failure, intermediate node first tries to repair route locally. If route couldn't be repaired locally, route error (RERR) is generated and sent back to the previous node, which forwards it to be sent to the source. The source upon receiving RERR will start a fresh route discovery if it has lost both the paths to destination (ensuring reduction in control packet overhead). In case if other path is still exists, it will only mark its backup path as INVALID, and will start sending data through only one path.

B. Calculating Affinity

Affinity index (AI) is a probability based upon historical data. Through this we can find out how much likely it is for a particular node to transfer the data packet to the desired destination. It is calculated by using the Bayes Theorem [5][6], which is $P(C_i/X) = \{ P(X/C_i) P(C_i) \} / P(X)$
Here, Ci is the class showing whether reply was received for the RREQ sent. And $X = \{ X_1, \ldots\ldots, X_N \}$ i.e. the various attributes upon which the probability will depend. P(X) does not depend upon C_i is used only for normalization. So $P(C_i/X)$ will be maximum when $P(X/C_i) P(C_i)$ is maximum[6]. Hence,

$$AI = P(C_{yes})\Pi viP(attributei/C_{yes}).$$

Assuming that every attribute xi is independent of other attribute x_j ($j \neq i$), and then we can say that

$$P(X/C_i) = \Pi_k{}^{n}=1 \; P(x_k/C_i) \text{ and}$$
$$AI = P(C_i) \; \Pi_{vk} \; P(x_k/C_i)$$

Now since we are multiplying the probabilities of each and every attribute hence; even if one of the attributes has a zero probability; the whole index will become zero. Because of this; zero probability will be replaced with a very low probability (0.001) [8].

C. Example

In the example below, source S broadcasts a request for destination D. It calculates its own affinity and puts it in the Route Request (RREQ).

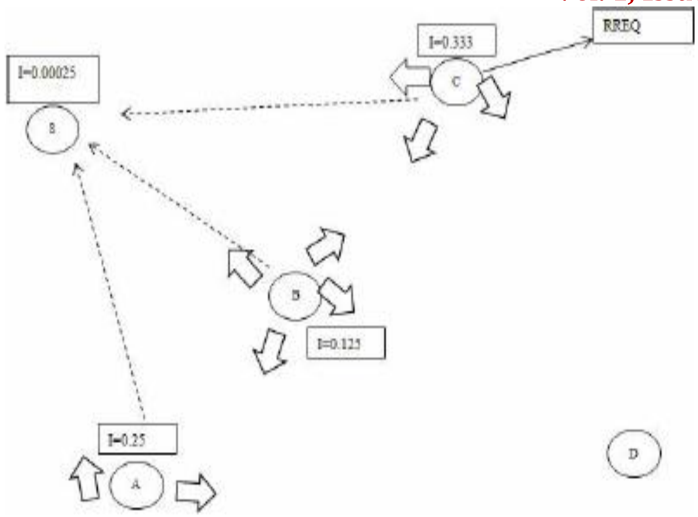

Fig. 1: Example to show forwarding of RREQ

Upon receiving RREQ, nodes A , B and C compare the affinity in RREQ with affinity in their Routing Table (RT). Since affinity in RT was less, they calculate their own affinity and add it in RREQ, and then they broadcast RREQ as shown in Fig.1. Now the destination upon receiving the route requests, replies by adding 1 to the affinity in RREQ received. Now each intermediate node (i.e. A, B, C) will forward route reply (RREP) to previous node if RREP contains higher affinity than that in their RT as shown in Fig.2 Source S upon receiving RREP, chooses route through C and A , while discards RREP from B since it had affinity lower than best two replies received by S. Now S starts sending data
through both paths alternatively.

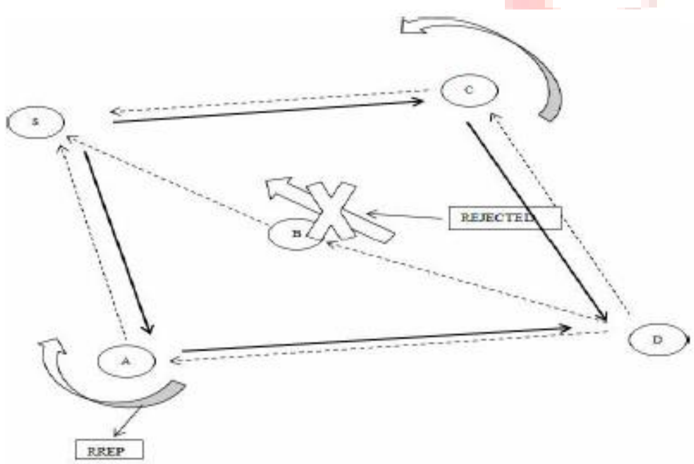

Fig. 2: Example showing forwarding of RREP and path for transferring data

**Solid line means the route used for sending the data packets.

**Dotted lines mean path to previous node (to which reply must be sent)

** “ I ” is the each node’s affinity for destination D

IV. ALGORITHM

In our algorithm, local repair works upon similar principle as that of AODV, except that we use Bayesian Approach to find routes as compared to destination sequence number.

```
A. Route Generation
1) Sending RREQ (route request)
for (each node in Neighborhood)
begin
if ( received (data_pkt)) AND ( rout flag marked INVALID)
            // route to destination does not exists
i. affinity(rreq)←affinity(rreq)+affinity(current_node);
ii. broadcast (RREQ (destination_id, affinity(rreq)));
iii. insert (history(destination_id, region, time, status));
else
        send (data_pkt );
end
```

```
  2) forwarding RREQ
for (each node in Neighborhood) //upon receiving RREQ
begin
a.  if( route flag marked VALID) // route exists
     1. send (rrep); //RREP stands for route reply
     2. end
b.  if ( destination_id(rreq) == current_node_id)
// I am the destination
     i. affinity (rrep) □affinity (rreq) + 1;
     ii. send (rrep(destination_id, affinity(rrep)));
    iii. end
c. else
    if(affinity(rtable) > affinity (rreq))
    i. affinity (rtable)□affinity (rreq);
    ii. affinity(rreq)□affinity(rreq)+affinity(current_node);
    iii. broadcast ( RREQ (destination_id, affinity (rreq)));
    iv. insert (history(destination_id, region, time, status));
else
       discard rreq;
end


3) on receiving RREP (route reply)
for(each node in Neighbourhood)
begin

a.  if source_id(rrep) != current_node_id
       if affinity (rrep) > affinity(rtable)
       i. mark route entry VALID;
       ii. affinity(rtable)□□ affinity(rrep);
       iii. forward(rrep <did, affinity(rtable)>);
       iv. update_status (history, destination_id);
       else
            discard RREP;
b.  else    // current node is the source
        if(affinity(rrep)> affinity(rtable))
        i. mark route entry VALID;
       ii. backup_hop(rtable) □□ nexthop(rtable);
       iii. backup_affinity(rtable) □□ affinity(rtable);
       iv. nexthop(rtable) □□ sender_ip;
        v. affinity(rtable) □□ affinity(rrep);
       else if (affinity(rrep) <= affinity(rtable) ) AND
             (affinity(rrep) > backup_affinity(rtable)) )
      //only source needs to maintain backup routes
         i. mark route entry VALID;
        ii. backup_hop(rtable)□sender_ip;
       iii. backup_affinity(rtable)□□ affinity(rrep);
end
```

Fig 3. Route Generation

V. PERFORMANCE RESULTS

We have done all the simulations using the network simulator NCTUNs (version 6.0) using 802.11 wireless network [7]. The paths of all moving nodes were generated randomly, and the payload used was 1400 bytes. For calculating delivery ratio, network size used was 10 nodes, as shown in fig 6. For the results shown in fig 7 and 8, network size was 32. For results in fig 9, mobility used was 10 m/s. All simulations were done for 30 seconds. A sample simulation run is shown in fig 5.

```
Route Maintainance

1)  Sending and Forwarding Route Error (RERR)
for (each node in Neighborhood)
begin
a.    if(( received (data_pkt)
          AND (route flag marked VALID))
                previous_node(rtable) ←send ( rerr );
b.    else if ( received (rerr))
        1. if ( backup_hop (rtable) != NULL )
          i. if(backup_hop(rtable) != sender_ip )

              A affinity(rtable)←backup_affinity (rtable);
              B nexthop(rtable) ←backup_hop(rtable);
          ii backup_affinity(rtable) ← 0;
          iii backup_hop (rtable) ← NULL;
          iv mark route flag VALID;
2     else
          i mark route flag INVALID; // no backup path exists
ii previous_node(rtable) ←send ( rerr );
end
```

Fig. 4: Route Maintenance

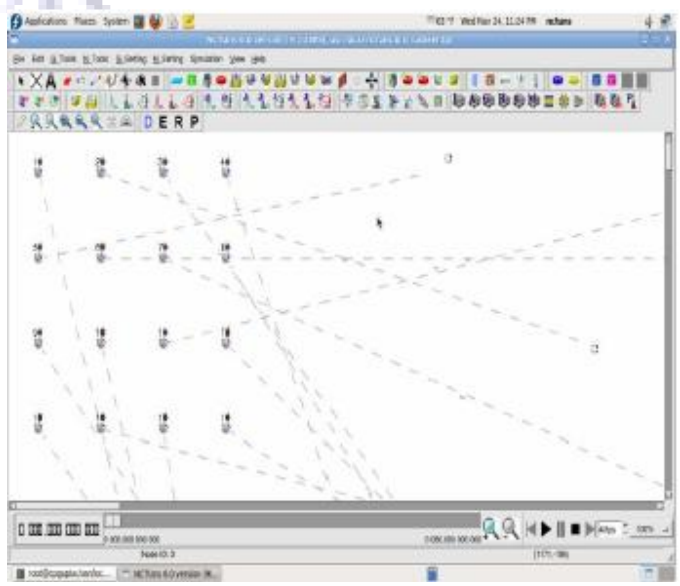

Fig 5. A sample simulation run

B. Delivery Ratio

Delivery ratio is the number of data packets received by destination upon number of packets sent by the source. Any protocol aims to have a higher delivery ratio so that there is sminimum loss of data packets.

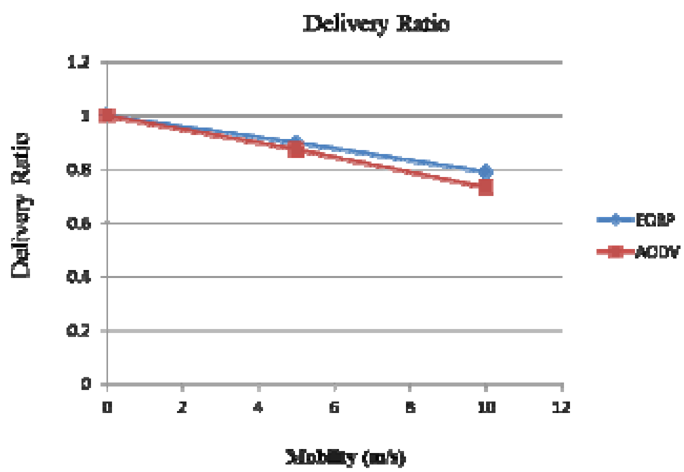

Fig. 6: Deliver Ratio of EORP v/s AODV

With zero mobility i.e. static network, both protocols have nearly 100% transfer of data, but with the increase in mobility, EORP is able to maintain a higher delivery ratio as compared to AODV.

C. Effect of Mobility on Packet Overhead

The number of control packets flooded into the network is the major reason of having unnecessary congestion and collision of data packets. Unnecessary flooding of control packets can cause collision of data packets, causing extensive increase in the number of data packets being dropped because of which any genuine packet may also not be able to reach the destination. Hence we present experimental analysis of the packet overhead between EORP and AODV.

Initially the overhead is more in both protocols, which decreases once the initial route is established. Later on, whenever a route breaks, control packets are again flooded into the network for finding the new path.

From the results it can be seen that, with the increase in mobility, number of control packets that are broadcasted increase manifolds in AODV whereas in EORP the increase is very less. In EORP there is an increase of only 13.68% in total control packet overhead[8], when mobility was increased from 5 m/s to 20 m/s whereas AODV showed a jump of 18.78%.

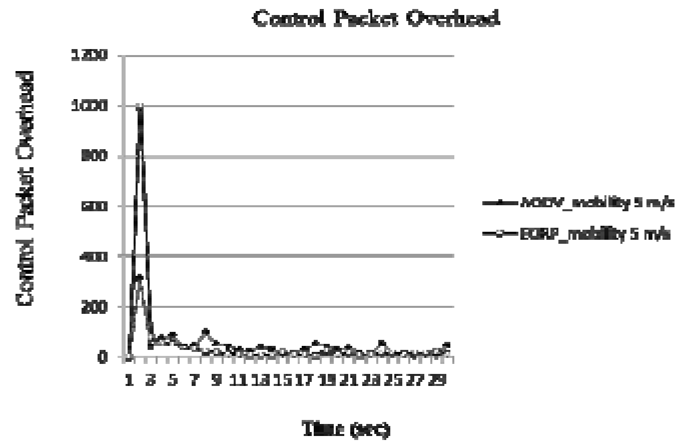

Fig. 7: comparison of control packet overhead w.r.t. mobility of 5 m/s

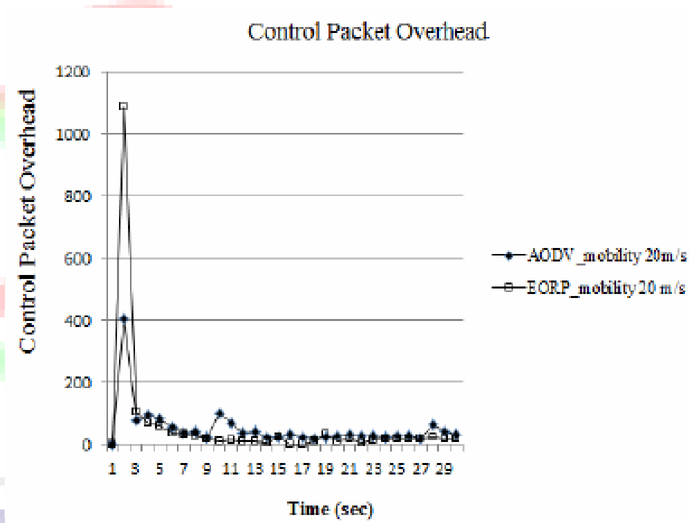

Fig. 8: comparison of control packet overhead w.r.t. mobility of 20 m/s

The reason why there is very high number of control packets in EORP initially as compared to AODV is that, in the very beginning of data transfer, we need to access secondary memory, so that the stored history can be loaded into the primary memory. This produces an additional time delay in starting, which causes the nodes to broadcast more number of control packets, because at that time they have no path to destination. But after history is loaded into the secondary memory, for the remaining part of the data transfer, control packet overhead reduces drastically as compared to AODV. This clearly shows that, by using the Bayesian approach in finding the route, EORP is able to find such routes to destination which have better lifetime.

D. Effect of network Size on Control Packet Overhead

As the size of network increases, the packet overhead increases because more number of nodes will now generate and forward control packets. In Figs. 7 and 8, it can be seen that EORP has less increase in overhead (once the history is loaded into primary memory of all nodes) as compared to AODV when network size was doubled from 16 to 32 nodes.

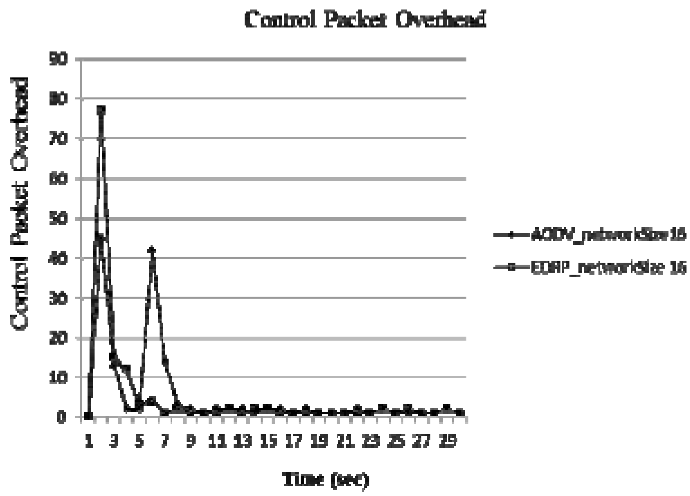

Fig 9: Control Packet Overhead wr.t. Network size of 16 & mobility 20m/s

VI. CONCLUSION

In our algorithm, we are using both time and space information to compute the route from source to destination. We have maintained the historic traffic information in each node along with the details on relative region from which the requests had come from by just expanding the current broadcast cache used in AODV. By using a Bayesian method, we have limited the flooding of broadcast requests.

VII. ACKNOWLEDGMENT

Dr. Anubhuti Khare one of the authors is indebted to Director UIT RGPV Bhopal for giving permission for sending the paper to the journal. HOD of E.C. Dept. Manish Saxena is also thankful to the Chairmen, Bansal Institute of Science & Technology Bhopal for giving permission to send the paper for publication. Last but not least, I would also like to thanks our colleagues for supporting us.

REFERENCES

[1] C. E. Perkins, and E. M. Royer, "Ad-hoc on-demand distance vector routing," 2nd IEEE Workshop on Mobile Computing Systems and Applications, Monterey, California, USA: Feb 25 – 26, 1999: 90-100.

[2] H D-Ferriere, M Grossglauser, and M Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages," 4th ACM International Symposium on MANET and Computing, 2003.

[3] C. E. Perkins, and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," ACM SIGCOMM Computer Communication Review, London, UK. Oct 1994. 23:234-244.

[4] D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Internet draft, draft-ietf-manet-dsr-00.txt, March 1998.

[5] Shabbir Ahmed, "Message Forwarding In People Centric Delay Tolerant Networks," Wireless Communications and Networking Conference (WCNC), 2010 IEEE

[6] George F Luger, "Artificial Intelligence", ed. 4, p. 333-338

[7] S. Y. Wang, C. L. Chou, C. H. Huang, C. C. Hwang, Z. M. Yang, C. C. Chiou, and C. C. Lin, "The design and implementation of the NCTUns 1.0 network simulator," Comput. Netw., vol.42, no.2, pp.175–197,2003.doi:10.1016/S1389 1286(03)00181-6

[8] Rusheel Jain, Murali Parameswaran, and Chittaranjan Hota "An Efficient On-Demand Routing Protocol for MANETs using Bayesian Approach" 978-1-4244-8953-4/11/$26.00 .2011 IEEE