Multipath Node-Disjoint Routing Protocol to Minimize End To End Delay and Routing Overhead for MANETs

Jayshree Tajne[1] (Student), Veena Gulhane[2] (Asst. Prof.)
Department of Computer Science and Engineering
G.H. Raisoni college of Engineering, Nagpur university, Maharashtra, India

ABSTRACT
Today, node-disjoint multipath routing becomes an essential technique in communication of packets among various nodes in network. Mobile ad hoc networks typically having the high mobility and frequent link failures, so multipath routing protocol is crucially important. In this paper, we proposed the multipath node-disjoint routing based on AODV protocol. This routing finds three node-disjoint routes from source to destination. The main goal is to discover multiple node disjoint paths with a low routing overhead during a route discovery, also improve the end-to-end delay and packet delivery ratio. The performance of the proposed protocol investigated and compared against the single path AODV and multipath NDMP-AODV protocols through simulation using .NET. Results have shown that the proposed multipath routing protocol outperforms both protocols in terms of routing overhead, end to end delay and packet delivery ratio.

Keywords- AODV, MANET, Multipath Routing, Node-disjoint

I. INTRODUCTION

A mobile ad hoc network (MANET) represents a system of wireless mobile nodes that can self-organize into temporary network topologies, allowing devices to internetwork in areas without any pre-existing communication infrastructure. Mobile ad hoc networks are characterized by high node mobility, dynamic topology and low channel bandwidth. In these scenarios, it is essential to perform routing with maximal throughput and, at the same time, with minimal control overhead. In the networking research community, there is tremendous interest in MANETs Routing [1]. Routing protocols can be classified as either unipath or multipath based on the number of routes between the source-destination pair. Intuitively, network resources can be better utilized by multipath routing and multipath routing can offer performance improvements over unipath routing [2]. Now days there are many researches on multipath routing protocols for mobile ad hoc networks [3].

Multipath routing protocols discover and store more than one route in their routing table for each destination node. In wireless scenarios, routes are broken due to mobile nature of node. Also, the wireless links used for data transmission are inherently unreliable. Therefore multipath routing has been used as an attractive alternative for shortest path routing protocols. Multipath routing provides the support for fault tolerance and load balancing. But the existing multipath routing protocols have some demerits such as larger routing overhead, less multipath route and more difficult in search for maximum relevant path. Node disjoint multipath routing allows the establishment of multiple paths, each consisting of a unique set of nodes between a source and destination. We know that MANETs consist of mobile nodes that cause frequent link failures. This link failure causes two main problems. Firstly, when a route break occurs, all packets that have already been transmitted on that route are dropped and it decreasing the average packet delivery ratio (PDR). Secondly, the transmission of data traffic is halted for the time till a new route is discovered and it increasing the average end-to-end delay.

In this paper, we develop multipath node-disjoint routing protocol to minimize end to end delay and routing overhead. The proposed approach minimizes the effect of link failure. Hence, the above mentioned two problems caused by frequent link failures are addressed. This protocol ensures that after a route is broken, the node can continuously send data without any delay, using one of the backup routes stored in its routing table during route discovery process.

The remainder of the paper is structured as follows. In Section II, related work is given by providing a brief description of existing multipath extensions of AODV routing protocol. Section III presents the proposed multipath protocol. Simulation and performance evaluation is presented in Section IV. Finally, the conclusion is provided in Section V.

II. RELATED WORK

In this section, we discuss the previous work done on multipath routing protocol. Multipath routing creates multiple paths between a source-destination pair. In case of the failure of first route, the backup routes are used for continues data transmission. In multipath routing protocols, the paths between a source and destination can be link-disjoint, node-disjoint or non-disjoint.

AOMDV [14] is a multipath extension of AODV [4] protocol to find out multiple disjoint loop-free paths between source and destination. It relies on the routing information already available in the AODV protocol, thereby limiting the control overhead incurred in discovering multiple paths. Long alternate paths are avoided by ignoring alternate paths that are more than one hop distance away. It has less number of route discoveries and more packet delivery ratio than AODV due to availability of alternate paths. The use of additional RREPs to form multiple forward paths to the destination increases the overhead for each route discovery but the overall overhead is less as compared to AODV due to less number of route discoveries. It does not provide scalability.

AODV-ABR [7] is an extension to AODV-BR which in turn is an extension to AODV. AODV-ABR tries to overcome the problems occurred in AODV-BR routing protocol. Overhearing of RREP and data packets makes the routing protocol more adaptable to changing topology without transmitting many extra control messages. Route maintenance is done by using a handshake process, which is accomplished by using two networks control signals: BRRQ and BRRP. The BRRP packet contains hop count field to solve the problem of choosing longer alternate path. Based on this hop count field, the node selects the shortest path available among the many alternate paths available. This can solve the congestion and collision problem occurring in AODV-BR. An aging technique is also going to be used for alternate route maintenance. AODV-ABR repairs the link failure by only using immediate neighbour nodes. So, it has less routing overhead and better throughput as compared to AODV.

SMORT [8] minimizes the route break recovery overhead by providing intermediate nodes on the primary path with multiple paths to the destination. It uses the idea of fail- safe multiple paths. There would be more fail safe paths as compared to node and link disjoint paths. Due to the usage of fail-safe paths, a link failure can be corrected at the intermediate node itself, thereby reducing the route recovery time and the number of route error packet transmissions. Fall safe multiple paths have higher fault tolerance to route breaks due to their higher availability

MP-AODV [9] discovers two routes for each pair of source-destination, a main route and a back-up route. Two RREQ messages are used to discover routes, each for one route. Whenever one route is broken, the other route is used for data packet transmission and a RREQ is flooded to maintain the broken route. This approach has two drawbacks: (1) MP-AODV contains higher overhead than the traditional AODV because it requires one RREQ flooding for one route and additional RREPs for node-disjoint paths and, (2) This approach is not able to find all the available node disjoint paths between a source-destination pair

NDM-AODV [10] finds all node-disjoint paths between source and destination also considers the residual energy of the nodes while selecting the routes. Multiple paths are created by using minimum routing overhead by making use of Destination Source Routing (DSR) protocol like source routing in route discovery process. Local connectivity is maintained by using Periodic Hello messages for all active routes during the route maintenance stage. The main disadvantage of this approach is that, as the size of the network increases, the size of the RREQ and RREP messages also increases because of the path accumulation function. Therefore, the size of routing table at destination node also increases due to the storage required to store multiple paths.

AODVM-PSE [11] presents multipath versions of AODV protocols, but the multiple paths identified in this approach are link-disjoint rather than node-disjoint. In this method, data transfer is started only after all multiple paths are discovered. Therefore there is an initial delay in data packet transmission. AODVM [12], AOMDV [13] presents multipath versions of AODV protocols. The multiple paths identified in these approaches are link-disjoint. The links which are created does not match with each other but may have node in common. Data transmission is started only after the discovery of all multiple paths therefore there is an initial delay in data packet transmission.

III. PROPOSED MULTIPATH ROUTING PROTOCOL

In this section, the proposed protocol is described. The main goal of this protocol is to find three available node-disjoint routes between a source and destination with minimum routing overhead and low end to end delay. To achieve this goal, this protocol works in the following phases: (i) Route Discovery Phase, (ii) Route Selection Phase and data packet transmission (iii) Route Maintenance Phase.

A. Route Discovery Phase

When a source node wants to transmit a data packet to destination, it checks its routing table for the next-hop towards the destination of the packet. If there is an active route entry for the destination in the routing table, then source forwards data packet to the next hop. Otherwise, the route discovery phase begins. In the route discovery phase, routes are discovered using two types of control messages: (i) Route request messages (RREQs) and (ii) Route reply messages (RREPs). The source node broadcasts the RREQ message into the network. Each intermediate node after receiving a RREQ packet, checks whether it is a fresh or a duplicate one by searching an entry in the Seen Table [6]. Seen Table stores three entries (i.e. source *IP* address, RREQ broadcasting ID (bi*d*_RREQ) and seen flag) that uniquely identify a RREQ message in the network. If an entry of received RREQ message is present in the Seen Table, then it is considered as a duplicate RREQ message and discarded without broadcasting to its neighbor. Otherwise, the node creates an entry in the Seen Table and updates its routing table for forward path before broadcasting the RREQ message.

Source IP Address	Broadcasting ID	**Seen flag**
---	---	---

Fig. 1. Seen Table structure

Type	R	A	Reserved	Prefix size	Hop count
Destination IP Address					
Destination Sequence Number					
Source IP Address					
Source Sequence Number					
Broadcasting ID					

Fig. 2. RREP structure

In this protocol, only the destination node can send RREPs upon reception of a RREQ message. The intermediate nodes are restricted to send RREPs even if they have an active route to destination. We have changed the data structure of Seen Table and RREP message as shown in figures 1, 2. This is done so as to get the node-disjoint routes. The destination node has to send a RREP message packet for each RREQ received, even if the RREQ message is a duplicate one. We change the data structure of both Seen Table and RREP message, by adding an extra field that works as a flag known as seenflag. At the beginning this flag is set to FALSE. The RREP messages initiated by destination node contain one extra field known as broadcast ID (*bid*_RREP).

The route discovery process is used to discover node disjoint paths. When a destination node receives a RREQ message packet, it creates the corresponding reply as RREP message. The destination node copies the bi*d*_RREQ from the received RREQ message into the *bid*_RREP field of sent RREP message. This RREP is sent towards the source of the RREQ using the reverse route to construct the forward route. For every RREQ message received, the destination does the above mentioned process. When the RREP message has been received by the intermediate nodes in the reverse path, the intermediate nodes check the seenflag value in their Seen Table. If the seenflag is set to FLASE, this indicates that this is the first RREP message packet on the reverse path towards the source node. So, the intermediate node forwards the RREP towards the source and reset the value of seenflag. When the intermediate node receives a RREP message for the same RREQ message it got earlier, the node simply discards the RREP message on the basis of seenflag value. Due to this, the intermediate node's cannot take part in more than one route from the existing multiple routes.

Algorithm 1: Route discovery method when a node receives RREQ message

N = Node
S = Source Node
D = Destination Node
I = Intermediate Node
S_flag = FALSE //Initial value of seen flag in seen table
*bid*_RREP= broadcast ID of RREP
bi*d*_RREQ= broadcast ID of RREQ
X = FALSE
Y = FALSE
Possible= FALSE
Count = 0
n_routes=3.

if (*S* has data to send)
{
  if (*S* has route for *D*)
  {
    Y=*start data transmission*
    if (Y==TRUE)
    {
      Find next possible node
      Possible= check for possible communication
      if (Possible== TRUE)
        Start data transmission
      else
        Get secondary path.
    }
  }
else
  Initiate RREQ broadcasting
}
if (*N* receives a RREQ message)
{
  if (*N* = I or N=S)
  {
    X= check value of seen table for duplicate RREQs
    if (X==TRUE)
      Discard RREQ without rebroadcasting
    else
      Rebroadcast RREQ

```
   }
 else
  {
    N is the destination
     bid_RREP= bid_RREQ
    Destination node unicast RREP on reverse route
    to create forward route
  }
}
```

Algorithm 2: Route discovery method when a node receives RREP message

```
if (N receives RREP)
{
  if ( N != S )
 {
    S_flag=check and return the value of seenflag
    from the seen table
    if (! S_flag )
   {
       Insert first route in routing table
       Reset the value of seenflag in seen table to
       detect duplicate RREPs
       Forward RREP to next hop towards source
    else
         Drop the duplicate RREP
    }
  }
 else
  {
       Count=count the numbers of active routes for
       destination in routing table
       if (Count < n_routes)
            Insert secondary routes and sort them in
            ascending hop count
       else
           Drop the RREP message
   }
 }
```

A node follows the procedure as shown in algorithm 1 after getting a RREQ message. When a source node wants to send a data packet to destination, it checks its routing table for any active route available for destination. If an active route is available, data packet is forwarded to the next hop towards its destination. Else, it creates a RREQ message packet and inserts the entry in seen table about the request packet. The re-sending of RREQ messages is avoided by using the above process. Each node does updating in its seen table to avoid duplicate broadcasting of the RREQ message. When a node receives RREQ message, the algorithm checks the node whether it is a source, intermediate or destination node. If the node is a source or intermediate node, then RREQ message is processed same as is done in the traditional AODV protocol. A destination node creates a RREP message after receiving the RREQ and copies bid_RREQ value from RREQ into the extra field provided in RREP. Destination node does not check whether the received RREQ message is fresh or duplicate as is done in traditional AODV protocol. It replies to every received RREQ message to establish multiple paths.

In discovery process, when a node receives a RREP message, Algorithm 2 is used to discover multiple node-disjoint routes. The RREP message is received by the node then the algorithm checks whether the node is an intermediate or source node. If the node is an intermediate node then its seenflag is checked from its seen table. In a Seen Table if seenflag is FALSE then this indicates that it is the first RREP message for this particular source-destination pair. The algorithm resets the value of seenflag for particular node corresponding to the source-destination pair and the primary route for the destination node is inserted. The RREP message is then forwarded to the next hop towards source. The duplicate RREP message is discarded. This ensures that all the discovered routes are node-disjoint. If the RREP message is received by the source node then the discovered node-disjoint path is inserted as primary or secondary, based on the value of seenflag and the number of routes already present for this destination in routing table.

Route discovery process of traditional AODV protocol is shown in figure 3. In Figure 4, we demonstrate with an example how the route discovery process in MND-AODV gets all node-disjoint routes between a source-destination pair. Suppose, node S is the source node and node D is the destination node. When node S has data to send, it initiates the route discovery process by broadcasting RREQ in the network.

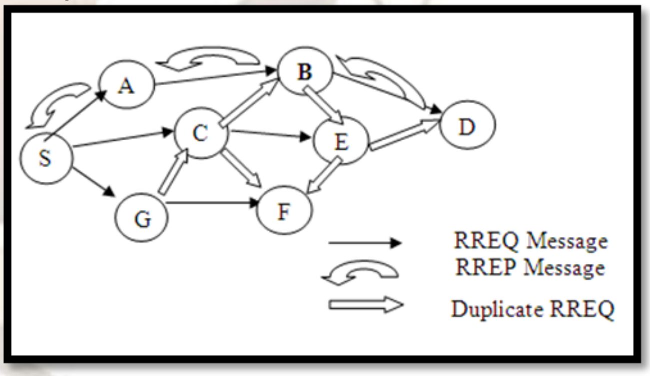

Fig. 3. AODV route discovery process

Let us assume that destination D receives its first RREQ from intermediate node H at time $t1$ and D initiates the RREP1 message. RREP1 is unicast towards source S by creating the reverse path D→H→G→F→S. When RREP1 is received by an intermediate node along the reverse route each intermediate node resets the value of seenflag in their Seen Table. Suppose, D receives the first duplicate RREQ message from E at time $t2$. Again node D initiates a RREP2 for this duplicate RREQ and sends it back towards node S through the same path it came to D (i.e. S→A→B→C→E→D) to make the reverse route D→E→C→B→A→S. This helps to create a forward route towards node D. Finally, say at time *t*3, node D receives the third duplicate RREQ message from node I. Node D initiates RREP3 for this duplicate RREQ and sends it towards S through I. The RREP3 reaches node H through I. Node H checks the value of seenflag for RREP3 before forwarding it to next hop. Node H determines that the seenflag is set to TRUE. So node H considers RREP3 as a duplicate message and drops it. This helps to maintain the node-disjoint property of our method.

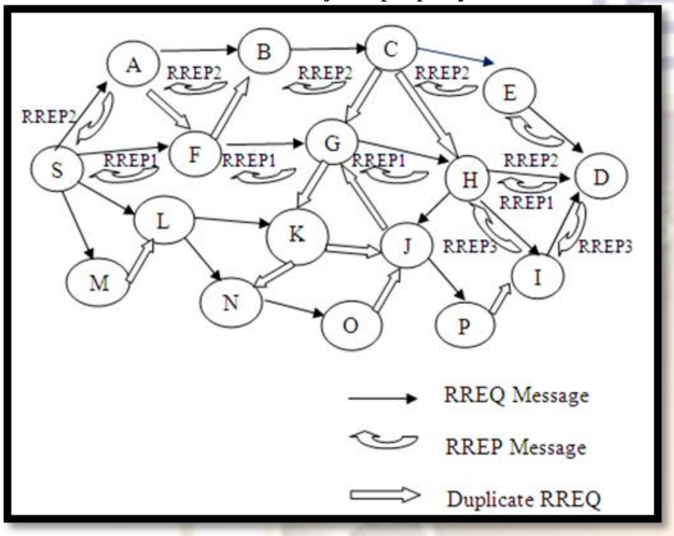

Fig. 4. MND- AODV route discovery process

B. Route Selection Process and Data Packet Transmission

If the source node has data packets to send and there is no route available in routing table, the node starts the route discovery phase. The data packet transmission is started as soon as it gets the first route for destination node known as primary route. All the other node-disjoint routes that are discovered will be stored in the routing table as secondary routes. After storing the primary route and specified number of secondary routes in the routing table, all the other routes are not stored. All the other routes that have lower hop count for destination as compared to existing secondary paths can replace the existing ones. The route selection phase works in such a way that whenever a route is required for data packet transmission, it always selects the primary route if it is available in routing table. If the primary route is not active, then the route selection process selects the route with lowest hop count from the available secondary routes in the routing table.

C. Route Maintenance Process

Due to node behavior as random mobile and the rapid change in network topology, link interruption may occur frequently in mobile ad hoc network. Therefore there is a need to consider the route maintenance. Route maintenance process is invoked when an active route is broken during transmission of data packets. We implement and analyze the performance of the route maintenance method in case of route breaks. In this method, when the primary route is broken due to the failure or mobility of an intermediate node, transmission of data is continued using the next possible node in the same path. This method decreases the RERR messages in network caused by intermediate nodes due to link breaks, thus increasing the network capacity. It also increases the PDR by not dropping the packets that are already on the broken route. If the next possible node is not in the range of the intermediate node previous to the failure node then the secondary path is used. To keep the secondary routes active while using the primary route, we increase the lifetime of each active secondary route after a fixed amount of time. When all the secondary routes are also broken, the source starts a new route discovery process. In this way, we can minimize the routing overhead caused in finding and maintaining multiple routes. Only one RREQ is used to find all available node-disjoint paths as compared to one RREQ required for each path, as in the case of traditional AODV and other existing multipath extensions of AODV.

IV. SIMULATION AND PERFORMANCE EVALUATION

In this section, we evaluate and compare the performance of the modified AODV scheme to the NDMP-AODV and the conventional AODV using .net for simulation.

The schemes are evaluated using the following three performance metrics:

1) *Routing Control overhead* — the total number of routing control messages per the total number of packets in the network (i.e. control messages plus data packets).

2) *Average end-to-end delay* — the average time taken for all data packets to be transmitted across a network from source to destination.

3) *Packet delivery ratio* — the number of received data packets divided by the number of generated data packets.

A. Simulation Setup

Our simulation modeled a network of 100 nodes placed randomly within a 1200 X 1200 sq. meter area. The random waypoint mobility model was used. Each node randomly selects a new position and moves towards that location with a pause time 0 and 300s. Once nodes reach the position, they become stationary for a predefined pause time and then select another position after a delay. The data rate is of 2mbps. This process continues until the end of simulation. Simulations were performed for 700 seconds with data transmission. We compared with the simulation results of AODV, NDMP-AODV and MND-AODV.

TABLE I
SIMULATION PARAMETERS

Parameters	**Values**
Number of nodes	100
Simulation time	700sec
Scenario Dimension	1200x1200 sq. meter
Transport protocol	UDP
Routing protocol	MND-AODV
Mobility model	Random way-point
Pause Time	0 to 300 sec
Radio type	802.11b
Data rate	2mbps
Number of Primary Routes	1
Number of Secondary Routes	2

B. Results and Analysis

In this section, we discuss the results obtained from simulation that have been performed to show the effectiveness of proposed route discovery and route maintenance method. The simulation results include the average packet delivery ratio (PDR), average end-to-end delay (EED), and routing control overhead caused by route discovery and route maintenance process. The effectiveness of proposed method is checked against the effect of node mobility. Figure 6 shows the overhead caused by routing control messages during route discovery process. Low routing overhead saves the bandwidth of the network, thus increasing the network capacity. The number of routes stored in routing table for a destination from the available node-disjoint routes greatly depends on the mobility of network. If the network mobility is high, the probability that the secondary route is expired with the primary route is high. As shown in Figure 5, AODV has the highest routing overhead because only one route for destination is stored in the routing table. Due to this, AODV has to broadcast the RREQ messages whenever route is broken to maintain the data transmission at all times.

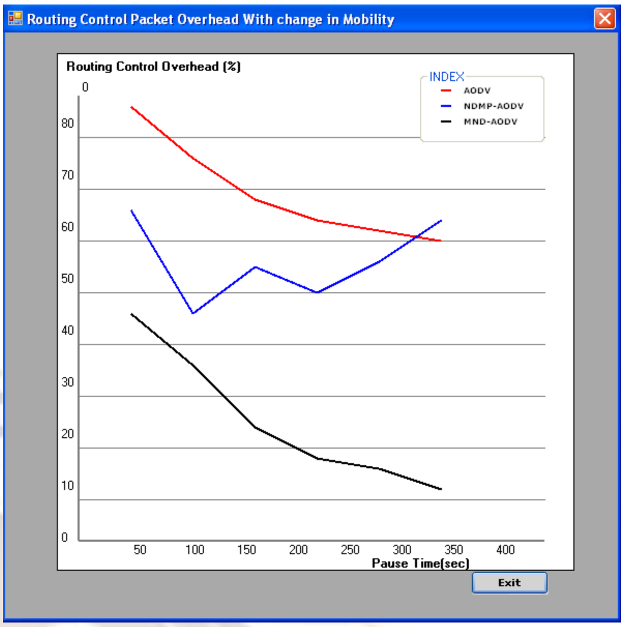

Fig. 5. Routing control overhead with change in node mobility

Effect of mobility of the node on end to end delay and packet delivery ratio are shown in Figure 6 and Figure 7. The delay in MND-AODV is less as compared to other protocols. This is because MND-AODV keeps a backup routing path more than 50% of the time when the primary route fails with the lowest routing overhead. We can observe from Figure 6, that end to end delay of all routing protocols decreases with increase in node pause time. Packet delivery ratio increases with increase in pause time of MND-AODV as shown in Figure 7. This is because the on-route data packets that are currently on the broken route are rerouted using the backup route from the point of route break.

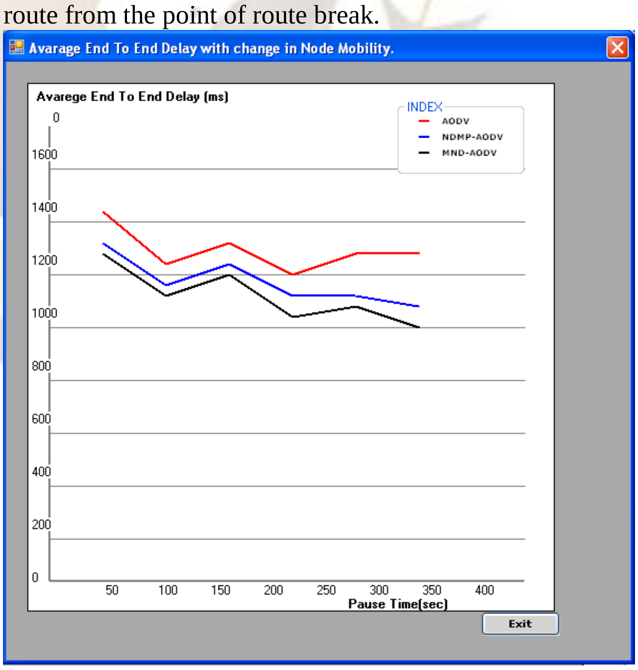

Fig. 6. Average end to end with change in node mobility.

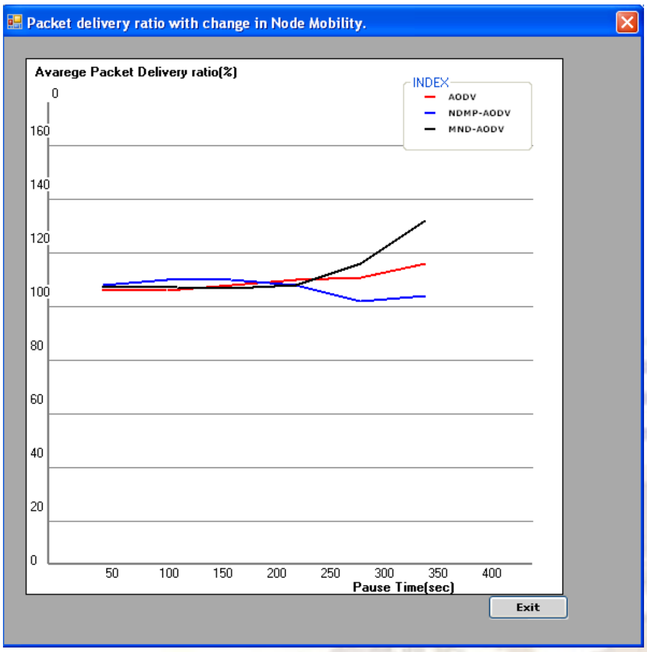

Fig. 7. Packet delivery ratio with change in node mobility.

TABLE II
COMPARATIVE ANALYSIS OF PROTOCOLS

Comparative Graph analysis

Parameter	AODV	NDMP-AODV	MND-AODV
Routing Overhead	69.71%	55.42%	31.42%
Delivery Ratio	54.09%	55.4%	60.4%
End to End Delay	49.95%	48.36%	42.75%

V. CONCLUSION

In this paper, we proposed multipath node-disjoint routing protocol to minimize end to end delay and routing overhead. Frequent link failure occurs in mobile ad hoc networks because of its features like dynamic topology and resource constraints. The proposed protocol provides the improvement of on-demand multipath routing method in terms of packet delivery ratio, average end-to-end delay, and routing control overhead. The proposed routing finds three node-disjoint routes from source to destination. The performance of the proposed protocol investigated and compared against the single path AODV and multipath NDMP-AODV protocols through simulation using .NET. Results show that our multipath routing protocol performs better than both protocols in terms of routing overhead, end to end delay and packet delivery ratio. MND-AODV causes approximately 24% less routing overhead in moderate or low mobility networks (i.e. when node pause time is greater than 100 sec) as compared to NDMP-AODV. Packet delivery ratio increases by 4% and end to end delay decreases by 6% as compared to NDMP-AODV.

In future work, we will extend the proposed method to work efficiently in high mobility networks by dynamically updating the backup route status. The applications like video on demand have high transmission rates as compared to the available channel bandwidth. Therefore a rate adaptation scheme should be combined with our multipath protocol.

REFERENCES

[1] M. Abolhasan, T. Wysocki, E. Dutkiewicz, "A review of routing Protocol for mobile ad hoc networks, Ad Hoc Networks," Vol. 2, No. 1, pp.1-22, 2004.

[2] Shafqat Ur Rehman1, Wang-Cheol Song2 and Gyung-Leen Park. "Associativity-Based On-Demand Multi-Path Routing In Mobile Ad Hoc Networks". KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 3, NO. 5, pp.475-496, October 2009.

[3] S. Mueller, D. Ghosal, "Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges", Lecture Notes in Computer Science, Springer, Berlin, pp. 209-2342004.

[4] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, pp. 90–100, 1999.

[5] David B. Johnson, David A. Maltz and Yih-Chun Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), INTERNET-DRAFT, 15 April 2003.

[6] Chhagan Lal, V.Laxmi, M.S.Gaur, "A Node-Disjoint Multipath Routing Method based on AODV protocol for MANETs", 2012 26th IEEE International Conference on Advanced Information Networking and Applications, pages 399-405, 2012.

[7] Wei Kuang Lai, Sheng-Yu Hsiao, and Yuh-Chung Lin, "Adaptive Backup Routing for Ad-Hoc Networks," Elsevier, Computer Communications, vol. 30, pp. 453-463, 2007.

[8] L.Reddeppa Reddy and S.V. Raghavan, "SMORT: Scalable Multipath On-Demand Routing for Mobile Ad Hoc Networks," Elsevier, Ad Hoc Networks, vol. 5, pp. 162-188, 2007.

[9] Chang-Woo Ahn, Sang-Hwa Chung, Tae-Hun Kim, and Su-Young Kang," A Node-Disjoint Multipath Routing Protocol Based on AODV in Mobile Ad hoc Networks", Information Technology: New Generations

(ITNG), 2010 Seventh International Conference on, pages 828 –833, April 2010.
[10] Shunli Ding and Liping Liu, "A Node-Disjoint Multipath Routing Protocol Based on AODV", Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on, pages 312 –316, aug. 2010.
[11] Fubao Yang and Baolin Sun, "Ad hoc On-demand Distance Vector Multipath Routing Protocol with Path Selection Entropy", Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on, pages 4715 –4718, april 2011.
[12] Tsung-Chuan Huang, Sheng-Yu Huang and Lung Tang, "AODV-Based Backup Routing Scheme in Mobile Ad Hoc Networks", International Conference on Communications and Mobile Computing, 2010.
[13] YuHua Yuan, HuiMin Chen, and Min Jia., "An Optimized Ad-hoc On Demand Multipath Distance Vector (AOMDV) routing protocol", Communications, 2005 Asia-Pacific Conference on, pages 569 –573, oct. 2005.
[14] Mahesh K. Marina and Samir R. Das, "Ad Hoc On-Demand Multipath Distance Vector Routing," Wiley InterScience, Wireless Communications and Mobile Computing, vol. 6, pp.969-988, 2006.