# An Extended AODV Protocol for Multipath Routing in MANETs

N.Jaisankar [1] and R.Saravanan [2]

*Abstract*—**Mobile ad hoc networks (MANETs) consist of a collection of wireless mobile nodes which dynamically exchange data among themselves without the need of fixed infrastructure or a wired backbone network. Due to limited transmission range of wireless network nodes, multiple hops are usually needed for a node to exchange information with any other node in the network. Thus routing is a crucial issue in the design of MANET. On-demand routing protocols for mobile ad hoc networks discover and maintain only the needed routes to reduce routing overheads. They use a flood-based discovery mechanism to find routes when required. Since each route discovery incurs high overhead and latency, the frequency of route discoveries must be kept low for on-demand protocols to be effective. The wide availability of wireless devices requires the routing protocol should be scalable. But, as the size of the network increases the on-demand routing protocols produce poor performance due to large routing overhead generated while repairing route breaks. The proposed multipath routing scheme provides better performance and scalability by computing multiple routes in a single route discovery. Also, it reduces the routing overhead by using secondary paths. This scheme computes combination of the node-disjoint path and fail-safe paths for multiple routes and provides all the intermediate nodes of the primary path with multiple routes to destination.**

*Index Terms*—**Mobile ad hoc networks; Multipath routing; Fail-safe multiple path; Primary path; Secondary path**

## I. Introduction

Mobile Ad hoc Networks (MANETs) are autonomous networks, which operate without any fixed infrastructure or wired backbone. In MANETs, nodes typically communicate over multiple hops while the intermediate nodes act as routers by forwarding data. Because of mobility and limited battery power of nodes, topology of ad hoc network is highly dynamic. Hence routing protocols should adapt to such dynamic nature and continue to maintain connection between the communicating nodes even if path breaks due to mobility and or node failures.

The objective of this paper is to develop multiple routes in order to improve scalability. Byfinding multiple paths in a single route discovery, reduce the routing overhead incurred in maintaining the connection between source and destination nodes. The secondary paths can be used to transmit data packets, in case the primary path fails due to node mobility or battery failure, which avoids extra overhead generated by a fresh route discovery. These multiple paths are more advantageous in larger networks, where he number of route breaks are high.

When a source node needs to send data to destination and does not have a valid path to destination, it starts a timer and relays a route request (RREQ) for destination with unique route request identifier. When source node receives a feasible reply for the destination, it updates its route table and starts sending a data packet. If the timer expires in between, then source node increments the route request identifier and initiates a new request for the destination.

Multipath routing can increase end-to-end throughput and provide load balancing in MANETs by the use of multiple paths. The concept of multipath routing motivated to design a multipath routing for mobile ad hoc networks.

1. To avoid the overhead of additional route discovery attempts.
2. To minimize the routing overhead by the use of secondary paths.
3. To reduce the route error transmission during route break recovery.

## II. Related Works

In this section, we have given a brief review of routing protocols which is for multipath routing. Mobile ad hoc networks (MANETs) are characterized by a dynamic topology, limited channel bandwidth and limited power at the nodes. Because of these characteristics, paths connecting source nodes with destinations may be very unstable and go down at any time, making communication over ad hoc networks difficult. On the other hand, since all nodes in an ad hoc network can be connected dynamically in an arbitrary manner, it is usually possible to establish more than one path between a source and a destination. When this property of ad hoc networks is used in the routing process, then it is called multipath routing.

In most cases, the ability of creating multiple routes from a source to a destination is used to provide a backup route. When the primary route fails to deliver the packets in some way, the backup is used. This provides a better fault tolerance and efficient recovery from route failures. Multiple paths can also provide load balancing and route failure protection by distributing traffic among a set of paths. Multiple paths between a source and a destination can be disjoint in two ways: (a) link-disjoint paths and (b) node-disjoint paths. Node-disjoint paths do not have any nodes in common, except the source and destination hence they do not have any links in common .Link-disjoint paths, in contrast, do not have any links in common.

Many on-demand multipath routing protocols have been proposed for mobile ad hoc networks, including Split Multipath Routing (SMR), Multipath Dynamic Source Routing (Multipath DSR), Temporally Ordered Routing Algorithm (TORA), Routing On-demand Acyclic Multipath (ROAM), Ad hoc On-demand Multipath Distance Vector (AOMDV), AODV-BR Ad hoc On-demand Distance Vector Backup Routing (AODV-BR) and Cooperative Packet Caching and Shortest Multipath (CHAMP). SMR and multipath DSR are based on source routing and are based on DSR while TORA, ROAM, AOMDV are distance-vector based. AODV-BR and AOMDV routing protocols are based on AODV.

Sung-Ju Lee and Mario Gerla proposed AODV-BR [2] routing protocol. The AODV-BR protocol uses the route discovery process as AODV [1]. When a source needs a route to a destination, and there is no route to that destination in its route cache, it searches a route by flooding a route request (RREQ) packet. Each of these packets has a unique ID so intermediate nodes can detect and drop duplicates. When an intermediate node receives a RREQ, it records the previous hop and the source node information and then broadcasts the packet or sends a route reply (RREP) packet back to the source if a route to the desired destination is known. The destination sends a RREP via the selected route when it receives the first RREQ or later RREQs that traversed a better route (with fewer hops).

The alternate route creation part is established during the RREP phase, and uses the nature of wireless communications. When a node that is not part of the selected route overhears a RREP packet not directed to it. It records the sending neighbor as the next hop to the destination in its alternate route table. In this way a node may receive numerous RREPs for the same route, select the best route among them and insert it into the alternate route table.

When an RREP finally reaches the source of the route, a primary route between that source and destination has been established. All the nodes that have an alternate route to the destination in their alternate route table form a fish bone. The properties of AODV-BR are is an extension of AODV. They floods RREQs with unique ID so duplicates can be discarded. Each node maintains backup route(s) in an alternate table. No multiple complete routes available. No multiple route(s) information known at source.

Mahesh K. Marina Samir R. Das proposed AOMDV [3] routing protocol. Like AODV-BR, the AOMDV uses the basic AODV route construction process. In this protocol some extensions are made to create multiple loop-free, link-disjoint paths. The main idea in AOMDV is to compute multiple paths during route discovery. It consists of two components: (i) A route update rule to establish and maintain multiple loop-free paths at each node. (ii) A distributed protocol to find link-disjoint paths. In AODV, when a source needs a route to a destination, it initiates a route discovery process by flooding a RREQ for destination throughout the network. RREQs should be uniquely identified by a sequence number so that duplicates can be recognized and discarded. Upon receiving a non-duplicate RREQ, an intermediate node records previous hop and checks whether there is a valid and fresh route entry to the destination in routing table. If such case, the node sends back a RREP to the source if not rebroadcasts the RREQ by incrementing the hopcount. A node updates its routing information and propagates the RREP upon receiving further RREPs only if a RREP contains either a larger destination sequence number (fresher) or a shorter route found.

In AOMDV each RREQ, respectively RREP arriving at a node potentially defines an alternate path to the source or destination. Just accepting all such copies will lead to the formation of routing loops. In order to eliminate any possibility of loops the “advertised hopcount” is introduced. The advertised hopcount of a node *i* for a destination *d* represents the maximum hopcount of the multiple paths for *d* available at *i.* The protocol only accepts alternate routes with hopcount lower than the advertised hopcount, alternate routes with higher or the same hopcount are discarded. The advertised hopcount mechanism establishes multiple loop-free paths at every node. These paths still need to be disjoint.

In AOMDV duplicate copies of a RREQ are not immediately discarded. Each packet is examined to see if it provides a node-disjoint path to the source. For node-disjoint paths all RREQs need to arrive via different neighbor of the source. This is verified with the first hop field in the RREQ packet and the first hop list for the RREQ packets at the node. At the destination a slightly different approach is used, the paths determined are link-disjoint or node-disjoint. In order to do this, the destination replies up to k copies of the RREQ, regardless of the first hops. The RREQs only need to arrive via unique neighbors.

S.Lee and Mario Gerla proposed SMR [4] protocol. It provides way of determining maximally disjoint paths. Paths are maximally disjoint when they are node disjoint, but when there are no node-disjoint paths available, the protocol minimizes the number of common nodes. Multiple routes are discovered on demand, one of which is the path with the shortest delay. The routes established by the protocol are not necessarily equal in length. Saleem et. al [10] proposed the model of self-optimized multipath routing algorithm. Fujian Qin [11] a multipath source routing protocol with bandwidth and reliability guarantee is proposed. In the routing discovery pahse, the protocol selects several multiple alternate paths which meet the QoS requirements and the ideal number of multipath routing is achieved to compromise between load balancing and network overhead. In the routing maintenance phase, it can effectively deal with route failures similar to DSR. Furthermore, the per-packet granularity is adopted in traffic allocation phase. Simulation results show that the proposed protocol remarkably increases the packet delivery rate and life-span of network with lower routing overhead. Yuwang Yang, et.al[12] presents network coding based reliable disjoint and braided multipath routing (NC-RMR) for sensor networks, which forms multipath by hop-by-hop method and only maintains local path information of each node without establishing end-to-end paths.

## III. Proposed Multipath Routing Scheme

This paper proposes a multipath routing scheme called Multipath On-demand Routing (MORT), in order to minimize the route break recovery overhead. This scheme provides multiple routes on the intermediate nodes on the primary path to destination along with source node. The primary path is the first path received by the source node after initiating the route discovery, which is usually the shortest path. Having multiple routes at the intermediate nodes of the primary path, avoid overhead of additional route discovery attempts, and reduce the route error transmitted during route break recovery.

Multipath routing protocols work on the principle that higher performance can be achieved by recording more than one feasible path. When multiple routes are known, even if the primary path fails data forwarding can continue uninterrupted on the alternate available paths without waiting for a new route to be discovered. In this scheme, the single-path AODV has been extended for multipath routing. This scheme is used for infrastructureless networks in which communication failure occurs frequently and designed to calculate node-disjoint paths and fail-safe paths. In node –disjoint path do not have any particular nodes in common, except the source and destination, whereas fail-safe is a path between source and destination if it bypasses at least one intermediate node on the primary path, which is the shortest path between the source and destination pair. Thus fail-safe path is different from node-disjoint and link-disjoint paths, in the sense that fail-safe path can have both nodes and links in common.

On-demand routing scheme that computes fail safe multiple paths reduces the route recovery time and path maintenance overhead more effectively than the node-disjoint multipath routing scheme. When node-disjoint multiple paths are used, only the source can correct the route disconnections, as alternate paths exist only at that node. In effect, route error packets have to be sent to the source node for every link break. In large networks, these error packets are likely to take considerable amount of time to reach the source node from the point of route break. Besides, the number of route errors communicated may also be high, as more number of nodes transmits these packets. Alternatively, usage of fail-safe paths has the advantage that route disconnection gets corrected at an intermediate node itself, thereby reducing the route recovery time and the number of route error transmissions.

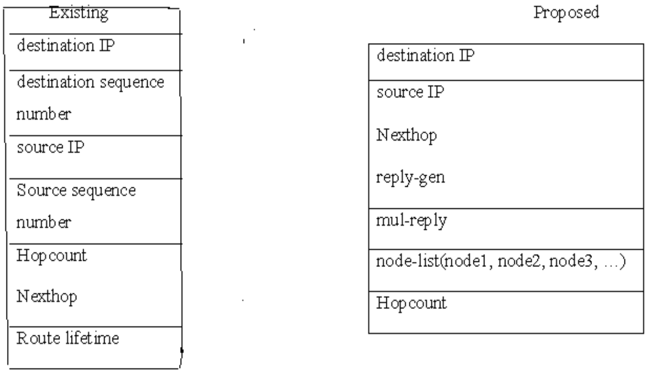
The proposed scheme provides multiple alternative paths using the combination of the node-disjoint path and fail-safe paths. This scheme has more alternative paths than node-joint or link-disjoint paths. Each MANETs node keeps and maintains tables—routing table, and neighbor node table. The propsoed scheme has two basic phases:

- route discovery
- route maintenance

### A. *Route discovery process*

First, to find routes for a destination node, a source node broadcasts an RREQ packet. When an intermediate node receives the first RREQ packet, it records a node address in route request table to relay RREP. When an intermediate node receives another RREQ packet again, then the node checks a node list field in the packet. The packet would be discarded immediately when the field contains the same node's IP address that of in the RREQ packet, else stores a node details into the request received table. After storing the node details it checks for route to a destination is exist in its routing table. If this check is passed then creates a RREP and send to the source using request received table entry. If not it re-broadcasts the RREQ packet by incrementing the hopcount. When a RREQ is received by destination node itself it stores the node address which relayed the RREQ in the request received table and creates a RREP, updates its routing table and send the RREP to its upstream nodes using request received table information.If a node receives a RREQ for the first time, it searches for a reverse route to the source. If no reverse route exists, then it will create a new route.

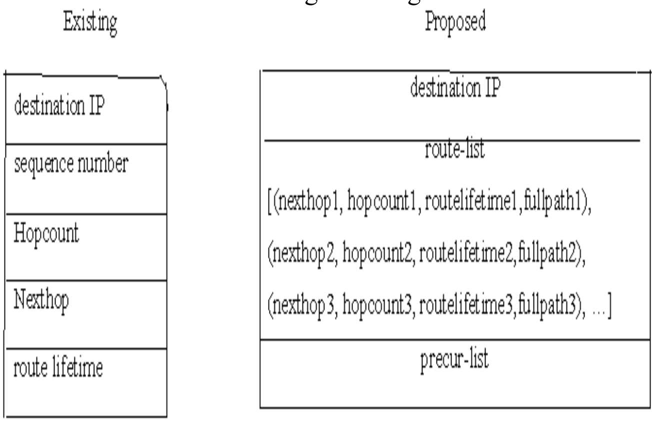
The extension of the RREP packet structure is given below:



| Existing | Proposed |
|---|---|
| destination IP | destination IP |
| destination sequence number | source IP<br>Nexthop<br>reply-gen |
| source IP | mul-reply |
| Source sequence number | node-list(node1, node2, node3, ...) |
| Hopcount<br>Nexthop | Hopcount |
| Route lifetime | |

TABLE 1. RREP PACKET STRUCTURE

The structure of the routing table is given below:



| Existing | Proposed |
|---|---|
| destination IP | destination IP |
| sequence number | route-list<br>[(nexthop1, hopcount1, routelifetime1,fullpath1),<br>(nexthop2, hopcount2, routelifetime2,fullpath2),<br>(nexthop3, hopcount3, routelifetime3,fullpath3), ...] |
| Hopcount | precur-list |
| Nexthop | |
| route lifetime | |

TABLE 2: ROUTE TABLE ENTRY STRUCTURE

The following details of different packet information have been given below:

The RREQ packet is created based on the IETF format specification. The fields are:

| | |
|---|---|
| PacketType: | To identify the type of packet |
| SrcAddr | : The node address which generates RREQ |
| SrcSeqNo | : Sequence number of source node |
| BcastId | : Request Id of RREQ |
| DestAddr | : Destination node address |
| Dest SeqNo | : Sequence number of destination node |
| Hopcount | : Number of hops from source |

The RREP packet is created with three additional fields. The format of RREP is:

| | |
|---|---|
| PacketType | : To identify the type of packet |
| SrcAddr | : The node address which generates RREQ |
| DestAddr | : Destination node address |
| Repgen | : Address of the node which generates the RREP |
| Mulreply | : Is a Boolean value. Set TRUE for the first Reply |
| Nodelist | : List of node address which relayed the RREQ |
| Hopcount | : Number of hops to reach the source node |

The fields of RERR packet is:

PacketType : To identify the type of packet

NodeAddr : Address of node where link failure is occurred.

The route table consists of the following information:

DestAddr : Address of the destination node.
RouteList : This filed holds multiple routes with the
values of nexthop, hopcount, lifetime, and fullpath.
PrecurList : Holds list of nodes that relayed a RREQ Packet

In this scheme, the destination is responsible for discovering primary path, node-disjoint paths and fail-safe paths from all the received routes as well as defining the route labels. The destination receives the RREQ for the first time, which stores the route path of RREQ and sets it with route label. Then the destination node creates route reply (RREP) in which route path is included. Once created, RREP will be unicast to the next hop according to route path towards the source S and the hop_cnt also incremented by one at each hop. Hence the intermediate nodes can forward this packet using path information in RREP. As the RREP reaches the source the hop count represents the distance, in hops, of the destination from the source. When the destination receives a duplicate RREQ, it will compare route path of RREQ to that of the routing table, then the path will be selected. The number of multiple paths between source node S and destination D can be discovered using selective RREQ forwarding scheme during route discovery process.

The number of RREP packets generation is limited to MAX_REPLY. The intermediate node that receives the first RREP packet forwards it to any neighbors using request received table that forms a reverse routes toward a source node and updates its routing table. Routing loop can be easily avoided by using the node list attached. If the node receives a delayed RREP packet, it updates routing table similar to the RREQ extension case, discarding the RREP packet. In this route accumulation process, nodes are adding their neighbor node route information as well as which type of paths are used in the route discovery process. If the destination nodes don't have the reverse route, it finds one new reverse route to the source.

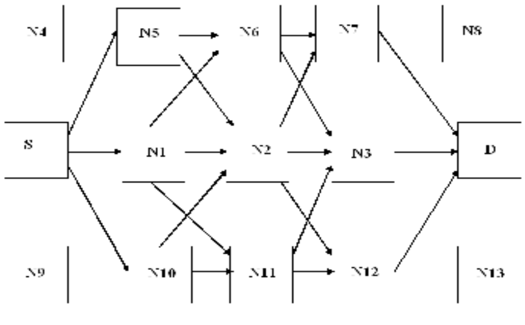


Fig .1 Discovering multiple paths during route discovery

Finally, the fastest RREP for the source node provides a primary route. The others are examined in the source node as well as in intermediate nodes, and some of the routes are accepted as backup routes according to the full path information. Data transfer begins just after the primary route is established. When the destination receives the duplicate RREQ packet, it will compare route path of RREQ of that routing table. If the source and destination nodes are same, then the path is said to be a node- disjoint path and the destination determines it as path type two.

If at least one of intermediate nodes in the route path in the routing table is different from nodes in the route path of the RREQ, a route is said to be a fail-safe path and destination determines it as path type three. After setting appropriate route label in RREP, the destination sends it to the source along the path information in it.

As shown in Figure 1, number of multiple paths between source node S and destination D can be discovered using selective RREQ forwarding scheme during route discovery process. After completion of route discovery process, there will be a primary path <S–N1–N2–N3–D>; two node-disjoint paths <S–N5–N6–N7–D>, and <S–N10–N11–N12–D>; and a number of fail-safe paths <S–N5–N2–N7–D>, <S–N1–N6–N3–D>, <S–N1–N2–N12–D>, <S–N10–N11–N3–D>.

### B. *Route maintenance*

When a node cannot receive HELLO messages from neighbors, the node detects link break. If neighbor nodes do not have any backup routes, the nodes invalidate their current routing tables and find precursor lists to send RERR packets to its neighbor nodes. Otherwise, the nodes immediately change a current route to a backup route. Avoidance of re-route discovery contributes to reduction of packet delay and the amount of routing packets in network. In addition, HELLO packets detecting link failure can update the backup route expiration timer and extend its life cycle.

## Results and Discussion

The proposed scheme has been implemented in NS2. The simulation environment consists of different number of nodes in a rectangular region of varying size. The nodes are randomly placed in the region and each of them has a radio range of 150 meters. Five sessions of Constant Bit Rate flows are developed for data transmission. The random waypoint model is chosen as the node mobility model. Simulation time is 300 seconds. Each scenario is simulated five times and an average is taken for the performance analysis. The random waypoint model is chosen as the node mobility model. All data packets are 512 bytes. Table **1** shows the simulation parameters used.

| | |
|---|---|
| Simulation time | 300 seconds |
| Number of nodes | 100 to 1000 |
| Bandwidth | 2Mbps |
| MAC layer protocol | IEEE 802.11 |
| Application type used | CBR (Constant Bit Rate) |
| Mobility model used | Random Waypoint Model |

TABLE 3.SIMUALTION PARAMETERS

The following metrics is used to analyze the scalability and performance of AODV by increasing the number of nodes in the network from 100 to 1000 nodes. Five CBR sessions are generated between randomly selected source-destination pairs. Averages of five sessions are taken for analysis.

The following three scenarios are considered for the analysis.

1. Mobility is kept constant at a minimum speed of 0 m/s, a maximum speed of 10 m/s, and a pause time of 30 m/s.
2. Varying the mobility speed from 10 m/s to 50 m/s
3. Varying the network load from 5 sessions to 30 sessions.

The following metrics are used to analyze the performance of the proposed scheme.

*A. Network Throughput*

This value represents the ratio of the total number of packets that reach their destination, to the total number of packets sent by the source. It is calculates according to this formula: Throughput = Packets Received / Packets Sent.

*B. Average end-to-end delay of Data Packets*

This is the average delay between the sending of the data packet by the constant bit rate source and its receipt at the corresponding constant bit rate receiver.

*C. Routing overhead*

Routing overhead is the total number of control packets transmitted by nodes while establishing and maintaining routes. Each hop-wise transmission of the control packet is considered.

In order to evaluate and compare the performance of proposed technique, a most widely used unipath on demand protocol AODV is chosen.

Three scenarios are considered for the performance evaluation.

1. Keeping the mobility of a node at a constant speed
2. Varying the mobility speed
3. Varying the network load

*Scenario – I: Keeping the mobility of a node constant*

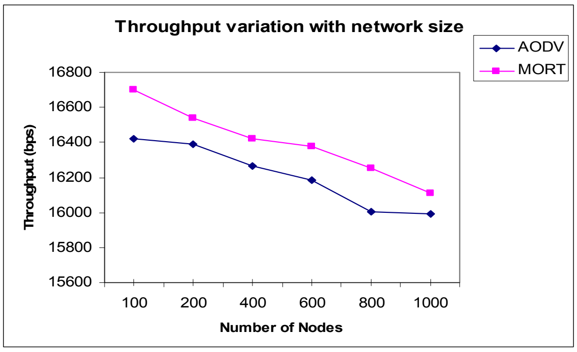


Fig. 2. Variation of throughput with network size.

Figure 2 shows the throughput comparison of MORT and AODV. Packet delivery capacity of all these routing techniques decreases as the number of nodes in the network increases. This is due to the increasing number of route breaks as the size of network increases. However, the propsoed scheme outperforms AODV in packet delivery capability for all sizes of network because most of the route breaks are corrected with secondary paths at intermediate nodes. This avoids packet drops at all the upstream nodes of the intermediate node that detected the route break. On the other hand, in AODV, all upstream nodes of the broken link drop packets to the disconnected destinations as they do not have secondary paths. Some of the packet drops are also due to the congestion caused by high routing overhead in AODV.

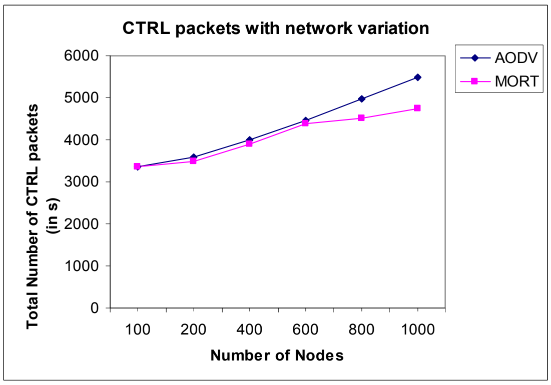


Fig. 3. Variation of routing overhead with network size.

Figure 3 shows the variation of routing overhead of two routing techniques. The value increases with network size because, the number of nodes communicating control packets and number of route computations increase as the network size increases. Number of route computations increase with network size because of increase in number of route breaks. AODV has higher routing overhead than MORT at all network sizes. This is because, AODV involves additional route computations and route error packet transmission for recovering route breaks. Where as in MORT route breaks can be resumed through the secondary paths and only a limited number of route breaks cause fresh route discoveries. Hence the proposed scheme has lower routing overhead that of AODV.

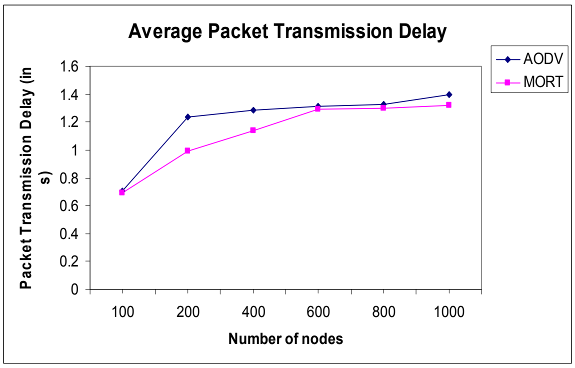


Fig. 4. Variation of packet transmission delay with network size.

Figure 4 shows the comparison of average packet transmission delay experienced by data packets for AODV and MORT. This metric reflects the delay involved in resuming the sessions after route breaks have occurred. The delay is high for AODV than MORT. But MORT has the lowest delay value at all network sizes, as it finishes the session with lowest number of route computations when compared to AODV. The proposed scheme increases throughput when compared to AODV. Reduction in routing overhead enables MORT to scale to double the number of nodes that AODV supports.

*Scenario – II: Varying the mobility speed*

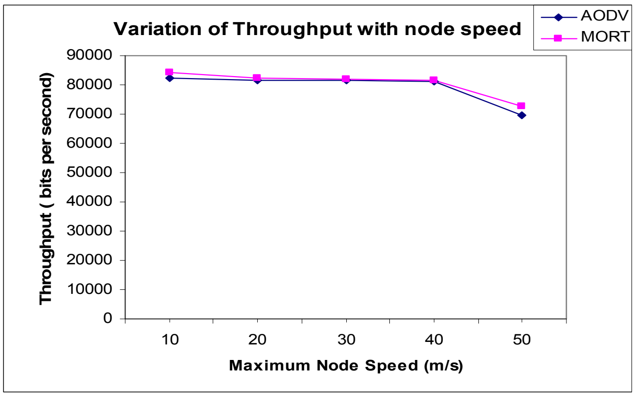


Fig. 5. Variation of throughput with node speed.

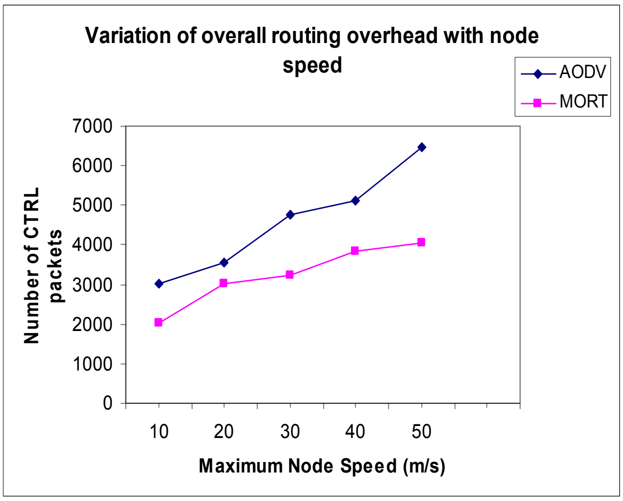


Fig. 6. Variation of routing overhead with node speed.

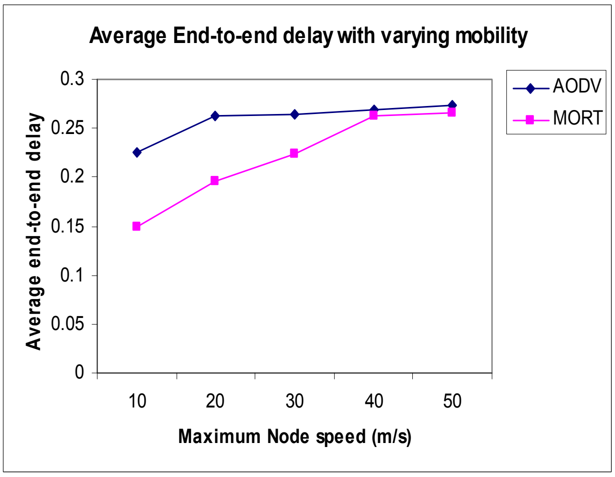


Fig. 7. Variation of packet transmission delay with node speed.

The comparative results of throughput, control overhead and end-to-end delay are shown in Figures 5, 6, and 7 respectively. As mobility increases, the protocol behaves as expected. Routing overhead and number of packet drops of these protocols increases with mobility, because of larger number of route breaks at higher speeds. But, the proposed scheme achieves improvement over AODV due to usage of secondary paths. Drastic increase of routing overhead in AODV at higher speeds show the need for methods to repair the route breaks with minimal routing overhead. Routing overhead and packet drops slightly increase with mobility, MORT outperforms AODV due to usage of fail-safe multiple paths to repair route breaks.

*Scenario – III: Varying the network load*

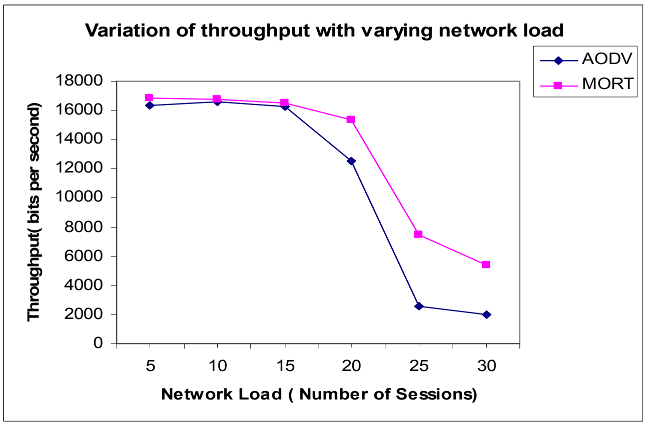


Fig. 8. Variation of throughput with network load.

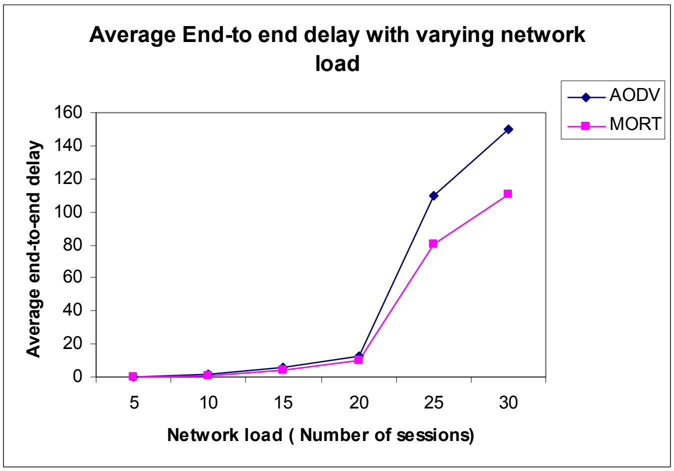


Fig. 9. Variation of packet transmission with network load.

Figures 8 and 9 show the throughput and end-to-end delay variation with offered load respectively. It is observed that the throughtput has been improved and delay has been reduced in the proposed scheme by varying network load. At higher loads, number of false route breaks increases due to congestion created by more number of active sessions. False route breaks occur as nodes falsely assume that a route break as occurred, when there are lots of packet drops due to collisions created by congestion is intact. So, AODV's overhead increases as it initiates fresh route discovery for every route break. MORT outperforms AODV by using secondary paths to repair route breaks.The number of route breaks in the network increase with the offered load. At higher loads, the proposed scheme shows better performance than AODV, because they minimize the routing overhead incurred in repairing route breaks using secondary paths.

## IV. Conclusion

This paper proposed a multipath routing scheme, in order to improve scalability and provides efficient multipath routing. Simultaneously, finding multiple paths in a single route discovery reduces the routing overhead incurred in maintaining the connection between source and destination nodes. Multipath routing can provide load balancing and reduce the frequency of on-demand route discovery. The simulation results show that the proposed scheme is better than AODV in discovering and maintaining routes. The performance analysis shows that the frequency of an on-demand route discovery for multipath routing is less than that for single path routing. Our future work intends to be in the direction of simulating the protocol for secure multipath routing.

## REFERENCES

[1] C.E. Perkins, E.M. Royer, "*Ad-hoc On-Demand Distance Vector Routing*", Proceedings of IEEE WMCSA, Feb 1999.

[2] S.J.Lee, M.Gerla, "*AODV-BR: Backup Routing in ad hoc networks*", IEEE WCNC, September 2000.

[3] Mahesh K. Marina Samir R. Das, "*AOMDV: On-demand Multipath Distance Vector Routing in Ad Hoc Networks*", 2000.

[4] Sung-Ju Lee and Mario Gerla, "*Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks*", Proceedings of IEEE International Conference on Communications, June 2001.

[5] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Ken Tang, Rajive Bagrodia, Mario Gerla,"*GloMoSim: A Scalable Network Simulation Environment*", 2000.

[6] Valera, W. Seah, and S. Rao, "Cooperative Packet Caching and Shortest Multipath Routing in Mobile Ad hoc Networks", INFOCOM 2003, San Francisco, CA, USA, 2003.

[7] Johnson, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)", IETF Internet Draft, draft-ietf-manet-dsr-09.txt, April 2003.

[8] Tasman Networks Inc. Routing basics: Protocol evolution in enterprise and service provider networks. Technical report, 2004

[9] V. Anantharaman, S.-J. Park, K. Sundaresan, and R. Sivakumar. "TCP performance over mobile ad hoc networks: a quantitative study", Wireless Communications and Mobile Computing, 4:203–222, 2004.

[10] K. Saleem et.al, "A Self-Optimized Multipath Routing Protocol for Wireless Sensor Networks", International Journal of Recent Trends in Engineering, Vol 2, No. 1, November 2009.

[11] Fujian Qin, Youyuan Liu, Multipath based Qos routing in MANETs", Journal of Networks, Vol 4, No 8, 2009.

[12] Yuwang Yang, Chunshan Zhong, Yamin Sun, Jingyu Yang, "Network coding based reliable disjoint and braided multipath routing for sensor networks", Journal of Network and Computer Applications, Volume 33, Issue 4, July 2010, Pages 422-432.