Dynamically Adaptive Multipath Routing based on AODV

Perumal Sambasivam   Ashwin Murthy   Elizabeth M. Belding-Royer
Department of Computer Science
University of California, Santa Barbara
{peru, ashwin, ebelding}@cs.ucsb.edu

Abstract— **Mobile ad hoc networks are typically characterized by high mobility and frequent link failures that result in low throughput and high end-to-end delay. To reduce the number of route discoveries due to such broken paths, multipath routing can be utilized so that alternate paths are available. Current approaches to multipath routing make use of pre-computed routes determined during route discovery. These solutions, however, suffer during high mobility because the alternate paths are not actively maintained. Hence, precisely when needed, the routes are often broken. To overcome this problem, we present an adaptive multipath solution. In this approach, multiple paths are formed during the route discovery process. All the paths are maintained by means of periodic update packets unicast along each path. These update packets measure the signal strength of each hop along the alternate paths. At any point of time, only the path with the strongest signal strength is used for data transmission. In this paper, we present two variations of our protocol and evaluate both with respect to two previously published multipath routing protocols. Simulation results show that the proposed solutions result in significant performance improvement.**

I. INTRODUCTION AND MOTIVATION

Ad hoc networks are characterized by dynamic topology, high node mobility, low channel bandwidth and limited battery power. In these scenarios, it is essential to perform routing with maximal throughput and, at the same time, with minimal control overhead. Several on-demand routing protocols have been proposed for ad hoc networks. In such protocols, nodes build and maintain routes as they are needed. Examples of these protocols include the Dynamic Source Routing (DSR) [1] and the Ad hoc On-Demand Distance Vector (AODV) routing protocol [2]. These protocols initiate a route discovery process whenever a node needs a route to a particular destination. In AODV, for example, the source broadcasts a Route Request (RREQ) packet in the network to search for a route to the destination. When a RREQ reaches either the destination or an intermediate node that knows a route to the destination, a Route Reply (RREP) packet is unicast back to the source. This establishes a path between the source and the destination. Data is transfered along this path until one of the links in the path breaks due to node mobility. The source is informed of this link failure by means of a Route Error (RERR) packet from the node upstream of the failed link. The source node then re-initiates a route discovery process to find a new route to the destination.

One observation of AODV is that, though the source actually discovers multiple paths during the route discovery process, it chooses only the best route and discards the rest. Also, frequent route breaks cause the intermediate nodes to drop packets because no alternate path to the destination is available. This reduces the overall throughput and the packet delivery ratio. Moreover, in high mobility scenarios, the average end-to-end delay can be significantly high due to frequent route discoveries.

Multipath on-demand protocols try to alleviate these problems by computing and caching multiple paths obtained during a single route discovery process. The performance of these protocols tends to increase with node density; at higher node densities, a greater number of alternate paths are available. In such protocols, link failures in the primary path, through which data transmission is actually taking place, cause the source to switch to an alternate path instead of initiating another route discovery. A new route discovery occurs only when all pre-computed paths break. This approach can result in reduced delay since packets do not need to be buffered at the source when an alternate path is available.

Current multipath routing protocols cache multiple routes obtained during the route discovery process [1], [3]. The best path, i.e., the path with the shortest hop count, is chosen as the primary path for data transfer while other paths are used only when the primary path fails. These protocols do not perform any kind of maintenance of the alternate paths. Path maintenance is important because node mobility, which results in the failure of the primary path, might also affect the alternate paths. As a result, when an alternate path is selected after the failure of a primary path, the alternate is likely to also be invalid. Using such stale or invalid paths results in more dropped packets while each of the alternate routes is tried in succession.

To remedy this problem, this paper presents a new multipath routing protocol called the *Mobility Prediction Ad hoc On-Demand Multipath Distance Vector (MP-AOMDV)* routing protocol. This protocol attempts to solve the above mentioned problems by periodically revalidating each of the alternate paths, while introducing a minimum of control overhead. In fact, since the solution prevents the use of broken alternate paths, our solution actually generates less control overhead in some scenarios than previously published approaches. The path maintenance process of our protocol also provides information on the quality of the paths. Based on this information the source can choose the best available path for data transmission. The source does not wait for its current path to break in order to switch to a different path. Instead, it constantly monitors each of its alternate paths and always selects the best among them for transmitting data. This selection is based on a metric we have developed to represent the overall strength of a path.

Specifically, the contributions of this paper are as follows:

- Two effective and simple multipath route discovery and maintenance mechanisms.
- A metric to quantify the quality of the overall path.
- Evaluation of our implementation and comparison with two other multipath routing protocols.

The rest of this paper is organized as follows. Section II discusses the related work in the area of multipath routing. In section III, we present the details of our protocol and describe its operation. We then evaluate our protocol and present the results in section IV. Finally, section V provides our conclusions.

II. RELATED WORK

A number of solutions for multipath routing in ad hoc networks have been proposed. Similar to the work proposed here, the Ad hoc On-Demand Multipath Distance Vector Routing protocol [3] and AODV-Backup Routing (AODV-BR) [4] are both based on the AODV protocol. In [3], multiple link-disjoint paths are computed from the source to destination through a modified route discovery process. The destination responds to only those unique neighbors from which it receives a route request. Each node in the network maintains a list of alternate next hops that are sorted based on the hop count. If, during routing, one of the links between two nodes break, then the immediate upstream node switches to the next node in its list of next hops. If the upstream node does not have an alternate next hop, it sends a RERR to its upstream neighbor. The source node initiates a route request when all its alternate paths fail. The main drawback of this protocol is that the alternate paths that are computed during route discovery are not maintained during the course of data transfer. Thus the paths could become stale and outdated by the time they are actually utilized. The multipath approach in this protocol is therefore not adaptive to the changes in the network topology. As we show in Section IV, the effectiveness of this protocol decreases as the mobility increases. While the route discovery mechanisms of our solutions are based on AOMDV, we modify both the manner in which alternate paths are selected and how those paths are maintained.

In [4], the authors propose a scheme to calculate alternate paths such that when a link failure occurs, the intermediate node searches for an alternate path to circumvent the broken link. The basic assumption made in this protocol is that all the nodes are in promiscuous mode and that they can overhear every transmission within their range. This protocol, however, has a number of limitations. First, it assumes that several nodes are within transmission range of each other. Also, constant mobility of the nodes is not taken into account. The protocol assumes that a node that offers the alternate route around a broken link does not move away and remains within range of the two nodes between whom the link has broken. Moreover, the utilization of promiscuous mode greatly increases the power consumption of each node.

There are other routing protocols, such as DSR and TORA [5], that have inherent facilities to support multipath routing. DSR caches multiple routes received during the route discovery process, based on the hop count of each route. In the event of route failure along the primary path, the source node starts data transmission along an alternate path chosen from the route cache. This approach is not effective under high mobility conditions where the alternate paths become stale very often, as shown in

[6]. In [5], the authors propose a distributed loop-free routing protocol, called TORA, that is based on diffusing computations. Here, multiple routes are computed mainly to alleviate congestion on links. TORA, however, requires reliable, in-order delivery of control messages.

Another multipath routing protocol used for load balancing is the Adaptive Multipath Source Routing protocol [7]. This is an extension to DSR. Here load is distributed among multiple paths based on RTT measurements. Routing On-demand Acyclic Multipath (ROAM) [8] is another protocol that makes use of diffusing computations. This requires coordination between nodes and state information to be maintained at each node during route discovery. This increases the overhead considerably. ROAM is better suited for wired networks and ad hoc networks with low node mobility.

Another interesting work in this area is presented in [9]. The authors propose an on-demand routing scheme, called Split Multipath Routing (SMR), that establishes and utilizes multiple routes of maximally disjoint paths. In this paper, the authors attempt to build maximally disjoint routes to prevent certain links from becoming congested and to efficiently utilize the available network resources. The SMR protocol is a variation of the DSR protocol and makes use of source routing to cache pre-computed alternate routes.

In [10], the authors propose a mechanism to analyze alternate path routing and its impact on load balancing. In addition, Nasipuri et. al. propose analytical models to study the effect of multiple paths on routing performance [11].

Thus, we find that most approaches to multipath routing lack a mechanism to expire stale cached routes. We propose a solution that can determine the freshness and validity of the alternate routes before utilizing them for routing.

Finally, while not a multipath protocol, the Signal Stability based Adaptive (SSA) routing protocol [12] performs on-demand route discovery by selecting longer-lived routes based on signal strength. The signal strength criteria allows the protocol to differentiate between strong and weak channels. Though our protocol also uses signal strength information for routing, the manner in which it is utilized is different from the SSA protocol. SSA is a distributed protocol that uses the signal strength information on a per link basis, whereas, our solution uses the signal strength information accumulated over an entire path.

III. ADAPTIVE MULTIPATH ROUTING: A MOBILITY PREDICTION BASED APPROACH

In the proposed protocol, predictions are made regarding the overall stability of routing paths based on the relative signal strength of the links along those paths. Based on these predictions, the various paths are prioritized so that the most stable path is chosen for routing before any other path. We measure the stability of a route based on the strength of the individual links in that route and not on the hop count.

We propose the use of the signal strength metric over the usual hop count metric because the hop count of a route is not sufficient to determine the quality and stability of the path. A very weak link, even if on a low hop count route, could lead to a significant number of dropped packets. In contrast, since our signal strength metric is based on the signal strength of each individual link in the path, it provides information about both the quality and reliability of the path. A single poor link will yield a very low signal strength for the entire path, thus making it less favorable for routing. Previous work has shown that using hopcount to select routes can be problematic [13], and that using signal strength often yields more reliable routes [12], [14]. In [14], signal strength is used to counter communication gray zones by identifying and discarding control packets received with a weak signal.

Mobile ad hoc environments can vary greatly in terms of the number of nodes, density of the network and bandwidth constraints. We propose and investigate the tradeoffs between two variations of the MP-AOMDV protocol:

- Node-Disjoint MP-AOMDV
- Link-Disjoint MP-AOMDV

The node-disjoint version is a more strict variation of the link-disjoint protocol and thus produces fewer alternate routes. Because the node-disjoint protocol discovers completely independent paths, each path will fail independently. With the link-disjoint protocol, individual node movement can cause the loss of multiple paths. However, the node-disjoint approach is less useful in sparse environments where there are few node-disjoint paths. In such scenarios, the link-disjoint version will prove to be more useful, since it can produce a greater number of alternate paths. Hence, we investigate both variations in this paper.

A. Node-Disjoint MP-AOMDV

Discovery of Multiple Node-Disjoint Paths: The purpose of computing alternate paths for a source node is that when the primary path breaks due node movement, one of the alternate paths can then be chosen as the next primary path and data transmission can continue without initiating another route discovery. One way to increase the likelihood that the alternate paths themselves are valid is to make sure that the alternate paths do not have any nodes in common with each other. Two routes that do not have any nodes in common are said to be node-disjoint. Previous studies suggest that such paths typically fail independently [3].

MP-AOMDV modifies the base AODV protocol's route discovery mechanism, in a manner similar to [3], to enable discovery of multiple node-disjoint paths for a particular source node. The RREQ packet is modified to contain the address of the neighbor of the source through which it has been forwarded. The destination node uses this information to reply to only those RREQs that come from distinct neighbors of the source. Since every intermediate node forwards only one RREQ toward the destination, each RREQ arriving at the destination has traveled along a unique path from source to destination. Thus when the destination replies only to RREQs from distinct neighbors of the source, these RREPs arrive at the source via node-disjoint paths. The source node then stores, in its route table, multiple next hops for each destination. Refer to [3] for a proof of correctness of the route discovery mechanism. The formation of node disjoint paths can be seen in an example shown in figure 1.

Maintenance of Alternate Paths: To ensure that the alternate paths stored at each source node remain up-to-date with the changes in the network topology, a separate mechanism is needed. The source node periodically sends a special update message, called a *heartbeat*, to the destination along each of its alternate paths. As the heartbeat packets propagate through the alternate paths, every node along that path updates the packet with a mobility prediction metric (MP). The MP is a measure of the relative signal strength with which a node receives a packet from its upstream neighbor and is given by:

$$MP = \frac{P_{AB} - P_{min}}{P_{min}} \qquad (1)$$

where P_{AB} is the power of the signal from node A as received by node B and P_{min} is the minimum threshold power with which the signal must be received for it to be considered as a valid transmission. Thus, the MP is a normalized representation of the signal strength. The source initializes the MP to one and as the packet traverses through the path, each node multiplies its MP with the value in the heartbeat. The destination unicasts the heartbeat packet back to the source along the same path after initializing the MP to one. In this process, all nodes obtain the signal strength information about the paths to both the source and destination. Hence when the heartbeat reaches the source, the value of the MP in the heartbeat is a cumulative product of the MPs of all links along that path. The path MP is given by:

$$MP_{path} = \prod_{i \in path} MP_i \qquad (2)$$

This product gives a measure of the relative stability of the path because links with higher signal strength are less likely to break. As the value of the MP increases, so does the stability of the path.

Another approach to measure the quality of the path, called the MaxMin approach, is to consider its weakest link. In this approach the MP of the entire path is just the MP of the weakest link. The source node chooses the path with the maximum MP for routing. We have performed simulations to determine which approach results in more stable paths. Simulations show that first approach is better in terms of packet delivery ratio. This is further discussed in the section IV.

The RREQ and RREP packets are modified to carry this MP value during the route discovery process. The source is thus able to learn the stability of the multiple paths during the route discovery itself. Once the source receives the RREPs, it sorts its next hop information in decreasing order of MPs and chooses the path with the greatest MP as its primary path. The source node thus maintains a priority list of next hop information. Based on this information, it uses the most stable path for data transmission.

As a result of the above mechanism, data packets always travel along the most stable path. Whenever the signal strength of the current primary path becomes lower than one of its alternate paths, the primary path is switched. At all times, the best available path is the primary path. In this way, the source switches routes to a better alternative when it sees the primary path growing weaker. Since the

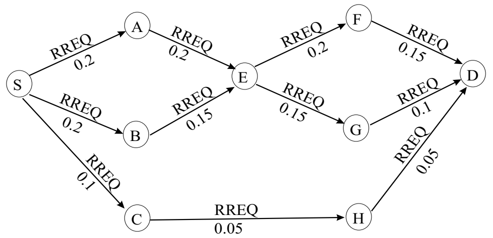
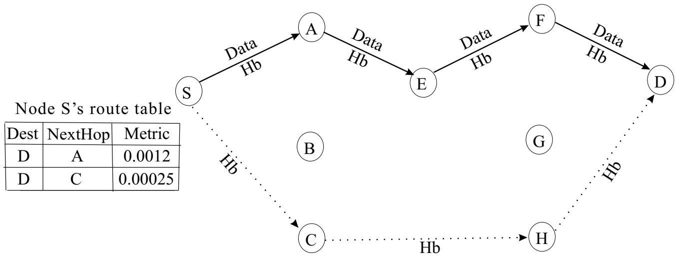


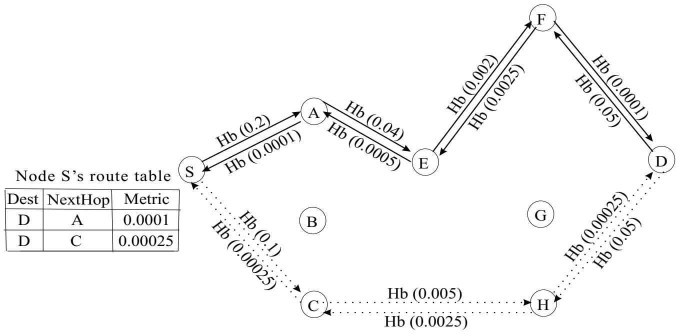
Fig. 1. Finding Node-Disjoint paths: Node E receives RREQs from nodes A and B, in that order. It forwards only the RREQ from A and discards the one from B. Destination D receives RREQs from nodes F, G and H. It sends RREPs only to F and H and discards the RREQ from G since it has come from the same neighbor (node A) of the source as has the RREQ from node F. The numbers on each link indicate the signal strength of that link.



Node S's route table

Dest	NextHop	Metric
D	A	0.0012
D	C	0.00025

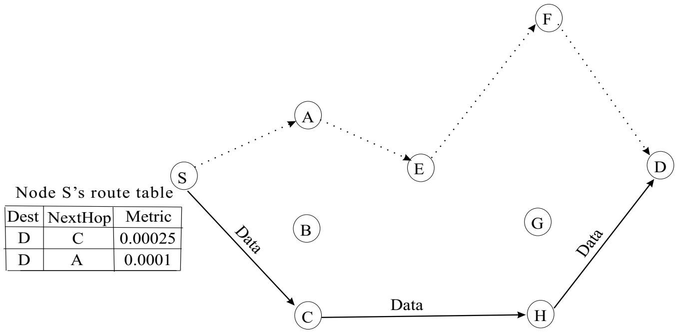
Fig. 2. Finding Node-Disjoint paths: Source S has received RREPs via two node-disjoint paths: S-A-E-F-D and S-C-H-D. Since path S-A-E-F-D has a higher MP value, it is selected as the primary path. The other path becomes the alternate path. The source sends periodic heartbeat packets (denoted by 'Hb') along both paths.



Node S's route table

Dest	NextHop	Metric
D	A	0.0001
D	C	0.00025

Fig. 3. Suppose node F moves away from nodes E and D. The signal strength of links E-F and F-D thus become very weak. 'Hb' denotes the heartbeat message. The number in parenthesis indicates the cumulative MP value that the receiving node will record in its route table.



Node S's route table

Dest	NextHop	Metric
D	C	0.00025
D	A	0.0001

Fig. 4. Source S detects this change through the low MP value of 0.0001. Since the other path now has a higher cumulative signal strength, source switches its primary path to S-C-H-D.

route is switched before the primary path is broken, fewer data packets are dropped and the end-to-end delay is also minimized. To prevent path oscillations, a hysteresis mechanism is adopted. Here, the source node switches from its current primary path to an alternate path only if the difference in the corresponding path stabilities is greater than some predefined threshold.

Computation of the most stable paths, based on the signal strength, involves only a marginal increase in computation at the source nodes. We utilize the signal strength, as opposed to the hop count, as our path selection metric because previous experimental results have demonstrated that the use of weak links can lead to routing path oscillations and numerous dropped data packets [14], [13].

Example of Node-disjoint Paths: As previously stated, the quality of each path is measured in terms of the signal stability of each of its links. Thus, the source node considers only the cumulative MP value, and not the hop count, when determining its primary path. A small hop count does not necessarily mean a good path since one of the links could be on the verge of breaking. On the other hand, the signal stability, while giving importance to the strength of each individual link, depends also on the number of intermediate links. Thus a path with fewer hops would obviously have a better MP value than an equally strong path with more hops.

We explain the operation of the protocol with an example shown in figures 1 to 4. Node S is the source and node D is the destination. When the source node S needs to send data to destination D and does not have any route, it initiates a route discovery process. Node S broadcasts a RREQ to its neighbors, nodes A, B and C. Nodes A and B relay the RREQ to node E. In this example, assume node E receives the RREQ from node A before it receives the RREQ from node B. Since each node forwards only the first RREQ received, it discards the duplicate RREQ from node B. Node E relays the RREQ from node A on both its outgoing links, i.e., to nodes F and G. Nodes F and G relay RREQs to the destination D. Similarly, node C relays its RREQ to destination D via node H. The destination sends RREPs to nodes F and H and discards the RREQ from node G. This is because the RREQ from node G originated from the same neighbor (node A) of the source as the RREQ from node F. The RREPs travel towards the source via the forward paths set up by the RREQs. Thus, the source learns of two node-disjoint paths to the destination.

There are two other reception possibilities. If node E receives a RREQ from node B before it receives one from node A, the node-disjoint path that is formed is still S-A-E-F-D. This is because, on receiving the RREQ from node A, E will switch its reverse path to S from B to A since the RREQ from A has higher signal strength. On the other hand, if node B receives a RREQ from node E before it can relay the RREQ it received from S, the multiple paths will again still be setup as previously described. Node B will discard the duplicate RREQ that it receives from E and relay the RREQ that it received from S. This RREQ will set up the reverse link at node E.

During the relay of RREQs from the source towards the destination, each node computes the MP based on the signal strength with its immediate upstream neighbor from which it received the RREQ. For example, consider the relay of the RREQ from node E to G. Node G computes the MP based on the signal strength with which it received the RREQ from node E. The signal strength information is obtained from the MAC layer. Node G now computes its MP as follows:

$$MP = \frac{P_{EG} - P_{min}}{P_{min}} \tag{3}$$

where P_{EG} is the power of the signal from node E as received by node G.

Suppose, for example, the MP for link E-G is 0.15, as indicated in figure 1. Node G then multiplies its MP (0.15) into the current product stored in the RREQ (0.04). Similarly, node D computes its MP with respect to node F as 0.15 and multiplies it into the current contents of the RREQ from node F (0.0008). Node D thus stores in its route table a MP value of 0.0012 corresponding to the path S-A-E-F-D.

Similarly, during the route reply phase, each node along the path computes its MP with respect to the node from which it received the RREP and multiplies it into the current product stored in the RREP. Both RREQs and RREPs need to carry the MP information so that all intermediate nodes learn the signal strength of their respective forward and backward paths to the destination and source. Node S sorts all the received paths and stores them in its route table in decreasing order of MP. The source chooses the best available path as its primary path and initiates data transfer along that path.

Figure 2 shows the data transfer along the primary path. The source node performs maintenance of all the alternate paths along with transfer of data via the primary path. At periodic intervals, the source unicasts the heartbeats, with the MP value initialized to one, via each of the node-disjoint paths, S-A-E-F-D and S-C-H-D. Each of the nodes along these paths multiplies its MP into the cumulative MP in the heartbeat just as in the case of the RREQ and RREP. When the destination receives a heartbeat from one of its neighbors, it records the MP for that path. It then creates a new heartbeat message, initializes the MP to one and unicasts it back along the same path towards the source. Based on the MP of the paths received, the source node sorts the next hops so that the best available path in terms of signal strength is always chosen as the primary path for data transmission.

Figure 3 shows the scenario when the signal strength of one of the links in the primary path becomes weak due to node movement. In this case, as node F moves away from node E, the MP of the link E-F becomes very low, for example 0.05. Similarly, the MP of the link F-D also decreases. The resulting MP of the entire path S-A-E-F-D is now 0.0001. The source S thus switches its primary path to the path via next hop C, as shown in figure 4.

Thus Node-Disjoint MP-AOMDV makes use of multiple node disjoint paths to provide alternate paths for data. Moreover, the use of heartbeats ensures that the best available path is always chosen as the primary path for data transmission.

B. *Link-Disjoint MP-AOMDV*

Discovery of Multiple Link-Disjoint Paths: The link-disjoint approach results in paths that are less disjoint than the node-disjoint approach. Here paths are allowed to have nodes in common, but the links still must be unique. Since it has fewer limitations than the node-disjoint version, we can obtain more alternate paths using this approach than with the node-disjoint approach.

Our approach for discovering link-disjoint paths is based on a modified version of AOMDV [3]. To discover link-disjoint paths, each node forwards only one route request towards the destination during the route discovery

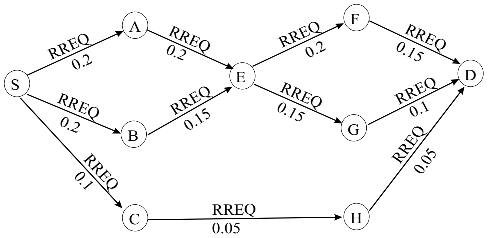
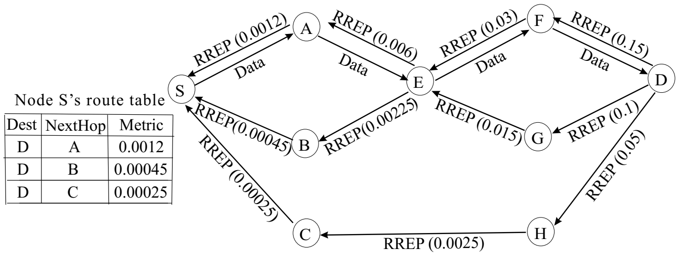


Fig. 5. Link-Disjoint MP-AOMDV: Node E receives RREQs from nodes A and B. It forwards only the RREQ received from node A. However, it enters node B into its route table as an alternate route to node S. The numbers on each link indicate the signal strength of that link.

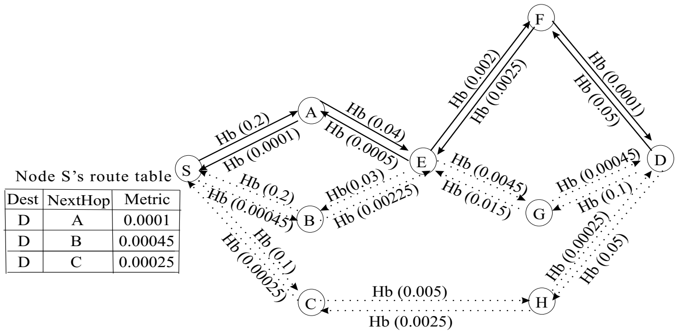


Node S's route table

Dest	NextHop	Metric
D	A	0.0012
D	B	0.00045
D	C	0.00025

Fig. 6. Link-Disjoint MP-AOMDV: Destination D replies to nodes F, G and H. Node E forwards the RREP from node F to node A and from node G to node B. Source S thus obtains three alternate routes to D and selects the path S-A-E-F-D as its primary route.

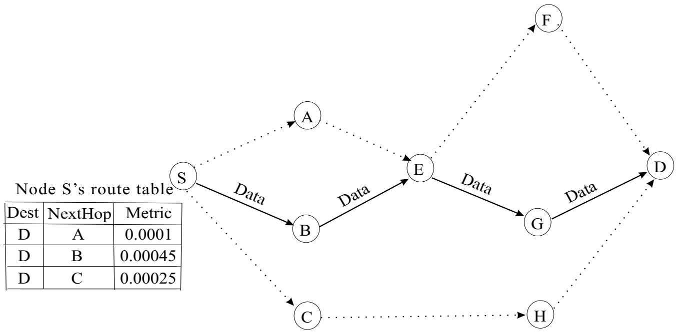
process; however, it maintains a queue of the previous hop nodes for each RREQ received from a unique neighbor of the source. As in node-disjoint MP-AOMDV, the RREQ contains a field indicating the first hop neighbor of the source through which the RREQ passed. The destination node sends an RREP to each of the unique previous hops from which it received an RREQ. When an intermediate node receives an RREP, it forwards the RREP to the node at the head of its next hop queue, and then removes this node from the queue. This forwarding scheme is repeated for subsequent RREPs received at this intermediate node. Once the queue is empty, all received RREPs are dropped. This method ensures that an intermediate node forwards each RREP it receives through a different upstream link. When there is an unequal number of upstream and downstream links, the excess links are not utilized in the routing and are expired. Thus, each intermediate node maintains a one-to-one mapping between its upstream and downstream neighbors. All packets are forwarded using this mapping. By this mechanism, each node now maintains a list of alternate next hops. Since the destination only sends one RREP to each neighbor and since the RREPs are transmitted across unique hops, all paths obtained at the source are link-disjoint. Refer to [3] for a proof of correctness of this route discovery mechanism.



Node S's route table

Dest	NextHop	Metric
D	A	0.0001
D	B	0.00045
D	C	0.00025

Fig. 7. Link-Disjoint MP-AOMDV: Heartbeat messages are sent periodically along all three paths. The number in parenthesis indicates the cumulative MP value that the receiving node records in its route table. Node F moves away from nodes E and D, thus reducing the strength of links E-F and F-D.



Node S's route table

Dest	NextHop	Metric
D	A	0.0001
D	B	0.00045
D	C	0.00025

Fig. 8. Link-Disjoint MP-AOMDV: The source learns of the movement through the drop in the MP value of the path (0.0001) and hence changes its primary path to S-B-E-G-D.

When a link failure occurs, the node upstream of the link detects the failure, invalidates its routing table entry for that destination and unicasts an RERR message towards the source. Each node along the active path receives the RERR and invalidates its corresponding route table entry. Once the source node receives the RERR, it switches its primary path to the next best alternate link-disjoint path. If no alternate path is available at the source, it initiates a route discovery.

Maintenance of Alternate Paths: Just as in the case of the Node-Disjoint MP-AOMDV, *heartbeat* messages are sent periodically along each of the alternate paths. Each node uses the one-to-one mapping between its upstream and downstream nodes, created during the route discovery process, to forward the heartbeat messages. The heartbeat message accumulates the signal strength information both towards the destination and back towards the source. This accumulated value is used by the source, destination and all intermediate nodes to order their multiple paths towards the source and destination.

Example of Link-Disjoint Paths: Consider the same example as shown earlier; figure 5 is identical to figure 1. In this case, node E receives RREQs from nodes A and B, with the one from A being received first. Node E forwards the RREQ from A onto each of its outgoing links. It also notes that a RREQ was received from node B in its route table but does not forward this RREQ. The destination thus receives RREQs from nodes F, G and H. The destination sends RREPs to each of its distinct neighbors, nodes F, G and H. Node E receives RREPs from node F and G. It then forwards the RREP received from node F to node A and the RREP from G to node B. As described before, it does this by maintaining two lists, namely a next hop list and a previous hop list, and a mapping between these two. Similarly, a RREP is also transmitted via the path D-H-C-S. Thus the source node obtains three link disjoint paths to the destination D. Figure 6 shows the propagation of the RREPs, as well as the data transmission along the selected path. Note that this process is similar to that in the Node-Disjoint MP-AOMDV, except that here more paths are discovered and maintained.

Figure 7 shows the scenario when node F moves away, thereby reducing the signal strength of the links E-F and F-D. This reduces the MP of these paths and causes the source node to switch data transmission to the path S-B-E-G-D, thus making it the primary path, as shown in figure 8.

IV. PERFORMANCE OF THE PROTOCOL

The main goal of the simulations is to evaluate the performance of both the variations of the proposed protocol. In addition, we compare the performance of our solutions with that of AOMDV. We also include DSR in our comparison because of its inherent multipath support. Our primary aim is to improve the packet delivery ratio as well as reduce the delay, particularly in high mobility scenarios, without introducing significant control overhead.

Since our protocol is an optimization to AODV, no comparisons are performed with the latter. When multiple paths do not exist, the protocol stops the heartbeat mechanism, and hence becomes basic AODV. Therefore, the worst case performance of the protocol is equal to AODV's performance.

TABLE I

SIMULATION PARAMETERS

Parameter	*Value*
Network size	1000m x 1000m
Number of nodes	50
Simulation time	200sec
Mobility model	Random Waypoint
Node speed	0, 1, 2, 5, 10, 15 and 20 m/s
Pause time	10sec
Node placement	random
Node transmission range	250m
Channel capacity	2Mbps

A. Simulation Environment

The simulation of the protocol has been performed using the GloMoSim network simulator [15]. The various simulation parameters are as shown in Table 1. The number of routes cached at each node is limited to three. Previous studies [11], [16] have shown this to be the optimal number of cached routes for multipath routing. Each route table entry thus consists of up to three next hops and a signal strength field for each of these next hops. All other fields are the same as in AODV.

IEEE 802.11 was the MAC protocol used. The results are averages of ten simulation runs. The traffic load used in all the simulations was Constant Bit Rate (CBR) data sessions between ten different pairs of nodes. Each data session consisted of 1000 packets of 512 bytes each sent at a rate of 10 packets per second. The predefined threshold used for switching paths, as discussed in section III-A, was 20%. In other words, the source node will switch to a new path if its signal strength is at least 1.2 times that of the current path.

B. Performance Metrics

We evaluate the following key performance metrics:

- Packet Delivery Ratio
- Average End-to-End Delay
- Control Overhead

The packet delivery ratio is calculated as the ratio of the data packets delivered to the destination to those transmitted by the CBR source. The average end-to-end delay includes all delays during the data transmission, including the buffering of packets during a route discovery after a link failure, retransmission delays and delays at the MAC layer. The control overhead is a measure of the amount

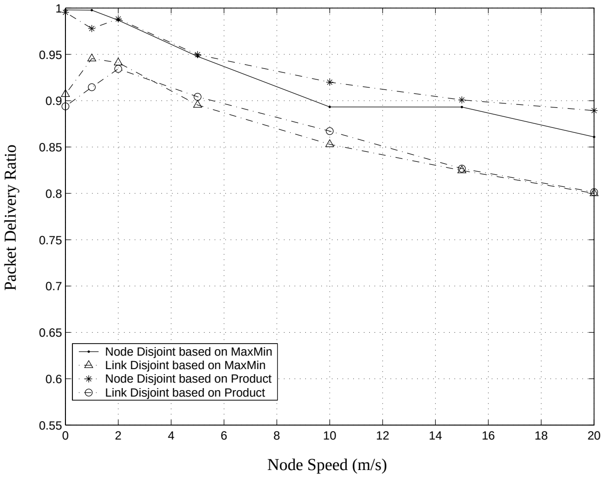


Fig. 9. Performance of MP calculation.

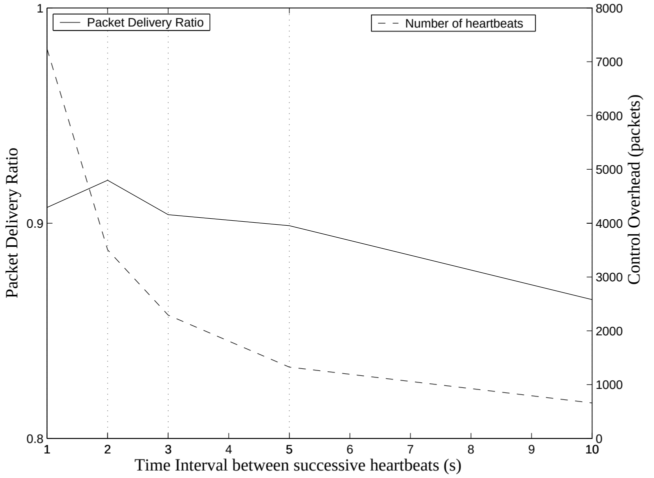


Fig. 10. Frequency of Route Updates.

of traffic generated by the control packets in the network. In the case of basic AOMDV, this includes the number of RREQs, RREPs and RERRs. In the case of MP-AOMDV, the control overhead also includes the heartbeats sent along each of the alternate paths.

C. Results

Calculation of MP_{path}: As discussed in section III-A, simulations were performed to determine the best approach to calculate the MP of a path. The results are shown in Figure 9.

The results show that the performance of both approaches is nearly equal. The MaxMin approach performs better at very low speeds, but the Product approach outperforms MaxMin at higher speeds. At higher speeds, all links, irrespective of their currently measured signal strengths, are more or less equally likely to break. Consequently, a more holistic approach that takes into account the signal strengths of all of the links in a path is a better measure than an approach that considers only the weakest link. Hence the Product approach is utilized for all further simulations.

Optimal Frequency of Route Updates: We first need to determine the optimal rate at which the heartbeat packet should be transmitted. This frequency plays an important role in the overall performance of the protocol. Sending the periodic heartbeat packets at large time intervals may result in more dropped data packets, since the routes may become stale while not in use. The loss of the path would not be discovered by the source until the next heartbeat is sent or a RERR is received. Frequent updates, on the other hand, produce a large number of control packets. This could affect the transmission of data packets if the medium becomes congested. There is thus a tradeoff between achieving a high packet delivery ratio and minimizing the control overhead. We performed simulations to determine the time interval that achieves the best balance between the two. Figure 10 plots both the packet delivery ratio and control overhead of the node-disjoint MP-AOMDV protocol against the time interval between successive heartbeat packets. It was found that sending heartbeats every two seconds yielded the best balance between the metrics. Hence, we use this sending interval for all further simulations.

Packet Delivery Ratio: Figure 11 compares the packet delivery ratio of each of the four protocols in varying mobility conditions. In the simulations, all nodes moved at the same specified speed. The graph demonstrates that node-disjoint MP-AOMDV performs the best among

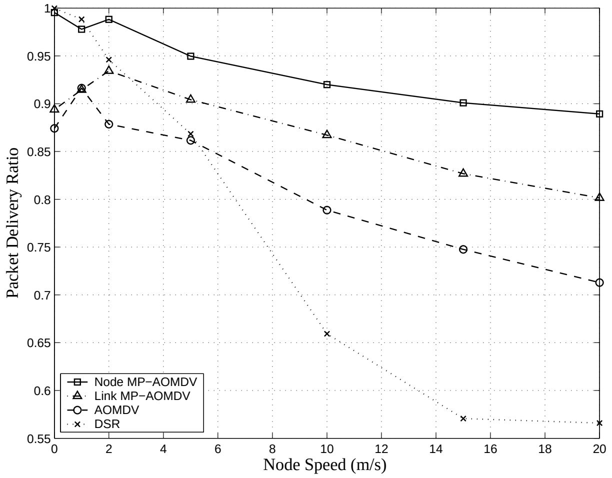


Fig. 11. Packet Delivery Ratio.

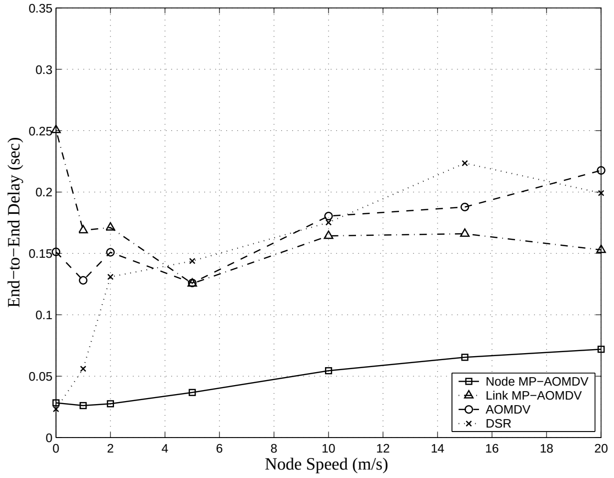


Fig. 12. End-to-End Delay.

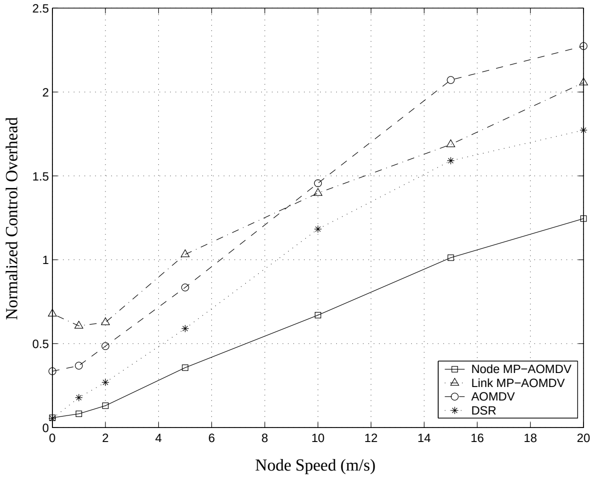


Fig. 13. Normalized Control Overhead.

the four protocols at nearly all speeds. This is mainly attributed to the fact that the paths in node-disjoint MP-AOMDV fail independently due to their node-disjoint property. As expected, both AOMDV and DSR perform well at low speeds but degrade at high speeds. Both variations of our proposed protocol perform better than AOMDV; the node disjoint solution achieves a minimum 10% improvement over AOMDV, up to a maximum of almost 20%. As the speed increases, the difference in performance becomes more evident due to the fact that routes become stale in AOMDV and DSR. The use of stale routes is detrimental to performance since it results in a larger number of dropped packets. This is even more evident at high speeds where routes become invalid frequently. The use of such invalid paths causes the packet delivery ratio to drop quite drastically at higher speeds.

End-to-End Delay: Figure 12 presents the average end-to-end delay as a function of the mobility speed. Once again, node-disjoint MP-AOMDV outperforms the other protocols comfortably at all speeds. It is able to achieve a near 75% reduction in end-to-end delay as compared to AOMDV and even more when compared with DSR. The regular maintenance of the paths in MP-AOMDV leads to an increased availability of valid alternate paths when the primary path breaks. Stale routes can increase the end-to-end delay since packets are transmitted along the invalid paths. Hence, the time between when the primary is lost and a new valid path is discovered increases. Link-disjoint MP-AOMDV also performs well at higher speeds but not as well the node-disjoint version. This is due to the fact that the link-disjoint paths are affected when a node common to multiple paths moves away. DSR has a good delay at low speeds due to the effectiveness of its multiple paths. However, when these alternate paths break very often at high speeds, DSR's performance degrades considerably.

Control Overhead: Figure 13 plots the normalized control overhead of each protocol against the mobility speed. The control packets counted for our protocols include the periodic heartbeat packets. Observe that node-disjoint MP-AOMDV has a considerably lower control overhead than AOMDV. Though AOMDV does not send periodic update packets as in MP-AOMDV, its normalized control overhead is still considerably higher. At low speeds, this is primarily due to its lower packet delivery ratio compared to that of node-disjoint MP-AOMDV. At higher speeds, the increased number of route discoveries initiated by the source contributes significantly to the control overhead. It is desirable to reduce the number of such broadcasts since they introduce load on the entire network. MP-AOMDV is able to achieve this by means of update packets, while still keeping the control overhead lower than AOMDV.

Number of Route Requests Initiated: To further understand the control overhead of the approaches, we also examine the total number of route requests initiated. The periodic maintenance of all paths reduces the use of invalid paths for routing. Apart from giving a better packet delivery ratio, it also helps in reducing the number of route requests initiated by the sources. Ideally, we would like this number to be as low as possible. Figure 14 compares the number of route requests initiated in each protocol at various speeds. DSR performs the best in terms of the number of route requests initiated. However, the normalized control overhead of DSR is higher than that of the node-disjoint MP-AOMDV, as is evident in

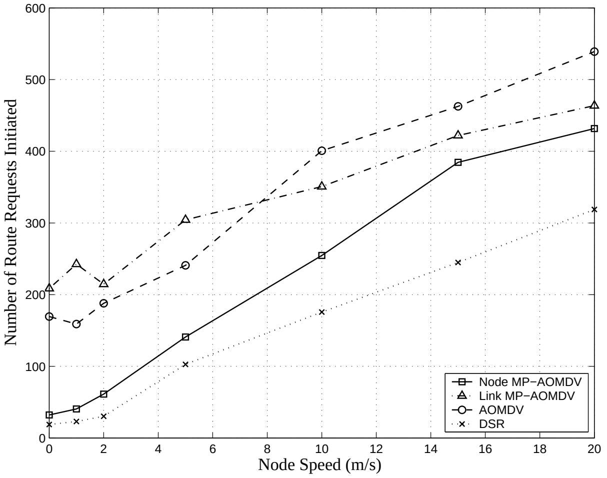


Fig. 14. Number of Route Discoveries.

figure 13. This is primarily due to the low packet delivery ratio of DSR at higher speeds, as seen in figure 11. Node-disjoint MP-AOMDV performs better than AOMDV and link-disjoint MP-AOMDV. Though AOMDV has fewer route discoveries compared to link-disjoint MP-AOMDV at lower speeds, this number increases more rapidly at high mobility speeds.

From the results in this section, it is evident that our protocol performs very well in both low and high mobility environments. MP-AOMDV significantly outperforms both AOMDV and DSR in terms of packet delivery ratio and delay, two important metrics in ad hoc networks. We have also shown that the control overhead incurred is much lower for our protocol than AOMDV and DSR.

V. CONCLUSION

In this paper, we present a new approach for multipath routing in mobile ad hoc networks. The primary characteristic of this approach is that it dynamically adapts to varying network topology by monitoring the quality of each path to the destination and always using the best path. It is able to eliminate stale routes and thereby reduce the number of data packets dropped due to the use of these invalid paths. Though control overhead is introduced through the periodic heartbeat packets, results prove that the overall overhead is still lower than other approaches. Our results show that the performance of the node-disjoint approach is superior to that of the link-disjoint approach. Hence, node-disjoint MP-AOMDV is the preferred protocol in networks that offer multiple node-disjoint paths. However, in networks where very few multiple paths exist, MP-AOMDV performs no worse than AODV. Both versions of MP-AOMDV hence offer a viable solution for a more reliable communication system in ad hoc networks.

REFERENCES

[1] David B. Johnson and David A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *ACM SIGCOMM*, 1996.

[2] 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, February 1999, pp. 90–100.

[3] Mahesh K. Marina and Samir R. Das, “On-demand multipath distance vector routing in ad hoc networks,” in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Riverside, California, November 2001, pp. 14–23.

[4] Sung-Ju Lee and Mario Gerla, “AODV-BR: Backup routing in ad hoc networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2000)*, Chicago, IL, Sepember 2000.

[5] Vincent D. Park and M. Scott Corson, “A highly adaptive distributed routing algorithm for mobile wireless networks,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997, pp. 1405–1413.

[6] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer, “Performance comparison of two on-demand routing protocols for ad hoc networks,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Tel Aviv, Israel, March 2000.

[7] Lei Wang, Yantai Shu, Miao Dong, Lianfang Zhang, and Oliver W.W. Yang, “Adaptive multipath source routing in ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001, vol. 3, pp. 867–871.

[8] Jyoti Raju and J.J. Garcia-Luna-Aceves, “A new approach to on-demand loop-free multipath routing,” in *Proceedings of the 8th Annual IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 522–527.

[9] Sung-Ju Lee and Mario Gerla, “Split multipath routing with maximally disjoint paths in ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications (ICC)*, Helsinki, Finland, June 2001, pp. 3201–3205.

[10] Marc R. Pearlman, Zygmunt J. Haas, Peter Sholander, and Siamak S. Tabrizi, “On the impact of alternate path routing for load balancing in mobile ad hoc networks,” in *Proceedings of the ACM MobiHoc*, Boston, MA, August 2000, pp. 3–10.

[11] Asis Nasipuri, Robert Castaneda, and Samir R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” in *ACM Mobile Networks and Applications (MONET) Journal*, 2001, vol. 6, pp. 339–349.

[12] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi, “Signal stability based adaptive routing (SSA) for ad-hoc mobile networks,” in *IEEE Personal Communication*, February 1997, vol. 4.

[13] Ian D. Chakeres and Elizabeth M. Belding-Royer, “The utility of hello messages for determining link connectivity,” in *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Honolulu, Hawaii, October 2002.

[14] Henrik Lundgren, Erik Nordstrom, and Christian Tschudin, “Coping with communication gray zones in IEEE 802.11b based ad hoc networks,” in *Proceedings of the 5th ACM international Workshop on Wireless Mobile Multimedia*, Atlanta, GA, September 2002, pp. 49–55.

[15] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Rajive Bagrodia, and Mario Gerla, “Glomosim: A scalable network simulation environment,” Technical report 990027, UCLA Computer Science Department, 1999.

[16] Israel Cidon, Raphael Rom, and Yuval Shavitt, “Analysis of multi-path routing,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 885–896, December 1999.