LOCATION-BASED POINT-TO-POINT ADAPTIVE ROUTING PROTOCOL FOR MOBILE AD HOC NETWORKS

Mehran Abolhasan[1], *Tadeusz Wysocki*[1], *Eryk Dutkiewicz*[2]

[1] Telecommunication and Information Research Institute,
University of Wollongong, NSW 2522, Australia
E-mail: mehran@titr.uow.edu.au, wysocki@uow.edu.au
[2] Motorola Australia Research Centre,
12 Lord St, Botany, NSW 2525, Australia
E-mail: Eryk.Dutkiewicz@motorola.com

ABSTRACT

This paper presents a new Global Positioning System (GPS)-based routing protocol, called Location-based Point-to-point Adaptive Routing (LPAR) for mobile ad hoc networks. This protocol utilises a 3-state route discovery strategy in a point-to-point manner to reduce routing overhead while maximising throughput in medium to large mobile ad hoc networks. In LPAR, data transmission is adaptable to the changing network condition. This is achieved by using a primary and a secondary data forwarding strategy to transfer data from the source to the destination when the condition of the route is changed during data transmission. A simulation study is performed to compare the performance of LPAR with a number of different routing algorithms proposed. Our results indicate that LPAR produces less overhead than the other simulated routing strategies, while maintaining high levels of throughput. Furthermore, a number of optimizations have been proposed to further improve LPAR's performance.

1. INTRODUCTION

Over the past decade the growing interest in mobile communication and the Internet has opened many new avenues of research in telecommunications. One research area is to provide Internet type application over Mobile Ad hoc Networks (MANETs). Unlike cellular networks, MANETs are made up of a number of end-user nodes, which are capable of determining routes to different parts of their networks, without using a base-station or a centralised administrator. This desirable feature of such networks, makes them useful in many different applications, particularly in areas where an infrastructure is not available or cannot not be implemented. These areas include, the highly dynamic battlefields environment where rapid exchange of crucial information can give significant advantage to one side or in search and rescue operations where the rescue team can use these networks to coordinate their efforts to search more effectively. Other areas where MANETs are useful are in exhibitions, conferences or concerts where a temporary network structure provided by MANETs can reduce implementation cost and time when compared to wired networks. However, MANETs have a number of limitations when compared to wired networks. These include, limitations in bandwidth, battery power and storage space. Other constraints include achieving different levels of QoS under a dynamic network topology and maintaining an acceptable level of data throughput as the number of users and traffic in the network increase. In previous literature, a number of different routing protocols have been proposed for MANETs. These protocol can be classified into three groups. These are proactive, reactive and hybrid routing protocols. The evolution in design of mobile ad hoc network routing protocols began from the traditional link state and distance vector algorithms, which are commonly used in wired networks. Routing protocols such as DSDV[14] and WRP[12] are among some of the early proactive protocols designed for MANETs [3]. However, due to the periodic updating strategy used in these protocols, they are not scalable in large networks, as the cost of maintaining full network topology will consume a significant part of the available bandwidth, power and storage space available at each node. Reactive protocols were designed to reduce the cost of maintaining up-to-date routes in proactive protocols at a cost of introducing extra delays during route discovery. This is done by determining routes when they are required via a route discovery strategy, rather than periodically exchanging topology information. The route discovery for most on-demand protocols proposed to date, such as DSR[9], AODV[5](recently expanding ring search was introduced to limit the scope of the search area) are based on pure-flooding. This means that every time a source requires a route to a particular destination, it will broadcast a RREQ packet throughout the network. This strategy lacks scalability, as the size of the network and the number of source/destination pairs increase under a dynamic network topology. As a result, a number of hybrid protocols such as ZRP[7], ZHLS[8] and SLURP[16] were introduced to reduce the effect of flooding in the network. In ZRP, each node defines a zone radius in which the network topology is maintained proactively and routes to destination nodes outside of the zone radius is determined reactively by bordercasting[7]. In ZHLS and SLURP, the network is divided into a number of non-overlapping zones. The topology within each zone is maintained proactively and the routes to the nodes in the interzone are determined reactively. The main disadvantage of ZHLS and SLURP is that they rely on a static zone map, which must be defined for each node at the design stage.

In this study, we propose a number of different strategies to reduce the overheads during route discovery under a dynamically changing network topology, and minimise the power consumed at each node. The rest of this paper is organised as follows. In Section 2, we describe our routing strategy. Section 3, the simulation environment and parameters used are described. In Section 4, we present a discussion on the results we obtained for our simulations. Section 5 presents a number of improvements and optimisations which can be implemented for LPAR and Section 6 presents the concluding remarks of our study.

2. LOCATION-BASED POINT-TO-POINT ADAPTIVE ROUTING PROTOCOL (LPAR)

Fundamentally, mobile ad hoc networks are dynamic in nature. These network may consists of a number of nodes with different levels of mobility, which may constantly create different node configurations and topologies. This means that during data transfer, source nodes may require a number of route recalculations[1] to successfully transmit the data. As discussed earlier, determining routes proactively over the entire network may use significant amount of the networks available bandwidth. Furthermore, reactive route discovery strategies based on flooding lack scalability as the size of the network increases [13]. Previous work has been done in [10][4] [5] to reduce the effects of flooding in source routing protocols (discussed in the following section). In this study, we use propose different strategies to reduce overheads under point-to-point routing. In Point-to-point routing, each node along the path to destination can make routing decision, which means that they are more adaptable to changing topology and reduce route recalculations at the source. In the following sections, we describe previous strategies proposed in the literature to reduce routing overheads in reactive routing and propose a number of new strategies to increase the performance of point-to-point routing strategies.

2.1. Reactive Route Discovering Strategies

The most common routing strategy used in on-demand routing protocol is pure flooding-based route discovery. In pure flooding the source node generates a Route Request (RREQ) packet which is broadcast and propagated globally through the network. When a RREQ packet reaches the required destination or an intermediate node with knowledge about the destination, a Route Reply (RREP) is generated and sent back to the source. If the RREQ has travelled through bi-directional links, then link reversal can be used to send the reply back to the source, otherwise, the destination may piggy back the route (if source routing) in a route reply packet, which is also flooded to reach the source.Protocols such as DSR and AODV are based on the flooding algorithm. The main differ-

---

[1] Depending of the level of mobility and how often the topology changes

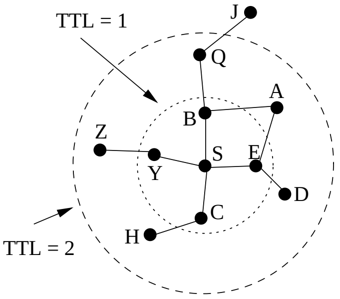
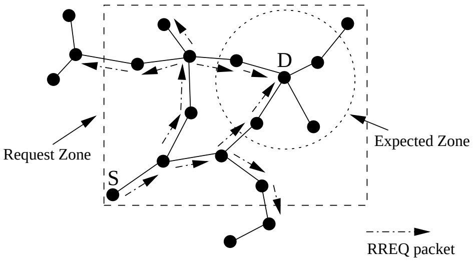
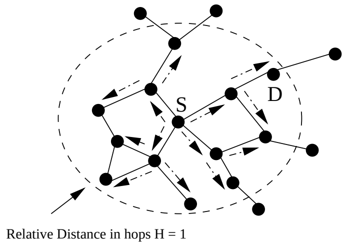


Figure 1: Controlled flooding using expanding ring search

ence between the two is in the way routes are created and used. DSR is based on source routing, which means, each data packet carries the complete source to destination address. AODV is a point-to-point routing protocol, which means that the data packets only carry the next hop address and the destination address. A number of different strategies have been proposed to reduce the routing overheads of pure flooding. Two such strategies are, Expanding Ring Search (ERS) and Restricted Search Zones (RSZ). In ERS, the source node incrementally increases the search area until the entire network is searched or the destination has been found. For example, if node S (see Figure 1) wants to find a route to node A, it will create a RREQ packet with a Time To Live (TTL) of one, which means that only the neighbouring nodes Y, B, F and C will see the packet. Now, since nodes F and B have a link to node A, they can send back a RREP to node S. As a result a route between node S and node A can be established without flooding the entire network. If node A was more that one hop away, then node S will timeout if no route reply is received and generate another RREQ packet with a higher TTL value. In RSZ, given that the source node has some idea of the current location of the destination or knows approximately how many hops away it is, it can calculate a region in which the destination node can currently reside and flood within that region only. Two such protocols which use RSZ are LAR1 and RDMAR. In LAR1, if the source node has a location information (through a GPS) about a particular node, it can calculate a region called the Expected Zone, in which the destination node can reside. If the source node is outside of the Expected Zone, a Request Zone (which is a region surrounding the expected zone) is also calculated.



(a) Localized RREQ propagation in LAR1



(b) Localized RREQ propagation in RDMR

Figure 2: Controlled flooding using restricted search zone

The source node will then restrict RREQ packet to the nodes within the request zone only (see Figure 2a). In RDMAR, the source nodes estimate the number of hops the destination is away from it (assuming a moderate velocity), thus restricting the route discovery within the calculated number of hop (see Figure 2b).

2.2. 3-State Route Discovering Strategy

As discussed earlier, LPAR is a point-to-point based routing strategy, which has been built on top of AODV. However, in LPAR, each node also exchanges location information (using GPS coordinates) in their hello message beaconing. In LPAR, if a node has location information for a required destination, it will use different route discovery strategies to determine a route, depending on the recored location and velocity of the destination. The aim of our 3-state route discovery strategy is to minimise routing overhead introduced into the network for each route discovery, while selecting relatively stable routes (This will be discussed later). We have defined 3 different routing scenarios and describe what strategies are used to determine a route for each scenario. Hence, the name 3-state routing. The routing discovery strategies used in our 3-state algorithm are as follows:

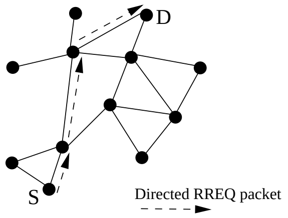


Figure 3: Directed Unicast RREQ propagation

(i) Directed Unicast Route Discovery (DURD)

(ii) Restricted Search Zone

(iii) Expanding Ring Search

When a node has data to send to a particular node and a location information is available, it will initiate the 3-state route discovery algorithm. Otherwise ERS route discovery will be used to determine a route. In our 3-state RD algorithm, the source node will first attempt to find a route to the required destination using our DRUD algorithm. If the discovery was unsuccessful, RSZ strategy will be used to search over a wider scope. Finally, if the RZS strategy fails ERS will be used to determine a route. To illustrate how the 3-state algorithm works, suppose that node S (see Figure 3) wants to send data to node D and the known route had expired. Now assume that node S has recorded location information (x,y) and velocity information V for D at t_0, and the current time is t_1. Then, the possible migrating distance for D is $d_m = V(t_1 - t_0)$. Furthermore, a Maximum Migration Distance (MMD) is assigned[2], if $d_m \leq MMD$ and $d_m \leq d_s d$, then DURD will be initiated. The aim here is to increase the accuracy of the DURD algorithm, since only one packet is forwarded. Therefore, we will use DRUD if the destination has not migrated too far from its known location and it has not migrated to the opposite side of the source. In DURD, the source node will attempt to send one packet through a selected node towards the destination. The selected node must lead towards the destination, must have at least one outgoing link and meet the stability criterion (this is discussed later). Each intermediate node will follow the same procedure until the destination is reached. The *DURD* algorithm is outlined below[3].

Algorithm *DURD*

```
(* The DURD algorithm *)

$N \leftarrow$ set of neighbours
$C_d \leftarrow 10000$ (* Closest distance found so far *)
$F_N \leftarrow NULL$ (* chosen forwarding neighbour so far *)
$D_d \leftarrow dist(node, destination)$
$D_i$ distance between neighbour $N_i$ and destination
$D_f$ distance between neighbour $N_i$ and this node
for $i \leftarrow 1$, $N_i \neq NULL$, $i++$
    $D_i \leftarrow dist(N_i, destination)$
    $D_f \leftarrow dist(N_i, node)$
    if $Deg(N_i) > 1$ and $D_f < \tau$
        if $D_i < D_d$ and $D_f < C_d$
            $F_N \leftarrow N_i$
            $C_d \leftarrow D_f$
return $D_F$
```

If DURD fails to find a route to the destination or if $d_m > MMD$, the source will calculate a RREQ propagation region (similar to LAR1), and attempt to find a route using RSZ. If unsuccessful, the source will increase the RSZ and another localised route discovery is initiated. Finally, if DURD and RSZ both fail, or location information is not available, then ERS will be initiated (note that the radius of ERS will be adjusted to cover the previously calculated propagation region in RSZ, if RSZ was used prior to ERS).

2.3. Adaptive Data Forwarding

Another way to reduce routing overheads in the network is by reducing the effects of link breakage during data transmission. A number of different strategies have been proposed to reduce the overhead costs of link failure, these include:

(i) Localised route maintainance [AODV,ABR]

(ii) Storing multiple routes [DSR,LAR1]

(iii) Backup routing using promiscuous overhearing [AODV-BR[11]]

Localised route maintainance, reduces routing overheads by repairing the route at the point of failure, by initiating a controlled flooding (similar to a RSZ) around the point of failure rather than initiating another route discovery at the source. Storing multiple routes (commonly used in source routing protocols such as DSR) can also be used to reduce the number of route recalculations at the source. However, this method still requires a RERR to be send back to the source. Furthermore, there is no guarantee that the source

---

[2] MMD is defined as a simulation parameter, we set $MMD = R/2$ where R is the maximum transmission range. Also, $d_s d$ is the distance between the source and the destination

[3] τ = max allowable distance between two nodes

will have alternate route or whether it will still be valid. Link failure overhead can be also reduced by maintaining backup routes at every intermediate node in the route. For example, in AODV-BR, the node detecting the link failure broadcasts the data packets to the neighbours. The receiving neighbours with a route to the next hop unicast the data to the next hop. The disadvantage of this strategy is the redundancy, as multiple nodes maybe sending the same data to the next hop. Additionally, the forwarding nodes can alter the data, which introduces further security problems. We propose a GPS-based Alternate Route Selection (GARS) strategy, where each node can select another node as the secondary route, if the primary route fails. Similar to [AODV-BR], in GARS the alternate routes are calculated during a route reply phase. However, instead to building backup routes using promiscuous overhearing at each neighbouring node, the node sending the route reply also selects another neighbour, which can be used as a secondary route in case this node is no longer available. For example, during RREP, node B (see figure 4) can select[4] node A as the secondary route to connect node L and E. This is done by calculating the distance between E and A and also L and A, if both these distances are less than the maximum allowable transmission range, then node B assign node A as an alternate path. Node L will accept node A as a secondary route if it forms a direct link with node A. Note that the RREP packet also contains the node id of the next node which leads to the destination. If a secondary route is used the node id of the second-hop is passed (using the IP options field at the moment), to the node in the secondary route. Therefore, the node in the secondary route can forward the data packet to the next hop which leads to the destination. For example, node L (figure 4) passes the node id, E, to node A, if the secondary route is used. Therefore, node A will then know that it should forward the data packet to node E unless it knows a better route. The *GARS* algorithm is outlined below.

Algorithm *GARS*
(∗ The GARS algorithm ∗)

```
$N \leftarrow$ set of neighbours
$SN \leftarrow NULL$ neighbour used as secondary route
$d_f \leftarrow 10000$
$d_r \leftarrow 10000$
$d_T prev \leftarrow d_f + d_r$
$d_T curr \leftarrow 0$
$T_x \leftarrow$ max transmission range
for $i \leftarrow 1,\ N_i \neq NULL,\ i++$
    $d_f \leftarrow dist(N_i, uplinknode)$
    $d_r \leftarrow dist(N_i, downlinknode)$
    if $d_f < T_x$ and $d_r < T_x$
        then $d_T curr = d_f + d_r$
            if $d_T curr < d_T prev$
                $S_N \leftarrow N_i$
                $d_T prev \leftarrow d_T curr$
return $SN$
```

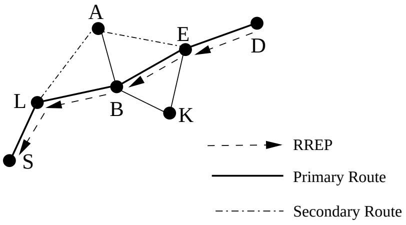


Figure 4: Alternate route selection using GARS strategy

The advantage of GARS compared to AODV-BR is that we eliminate data redundancy by specifying which node can be used as the secondary relay point if the primary relaying node is no longer available. Furthermore, security is increased since a known node is selected as a secondary relay point rather than relying on an unknown nodes to forward the data.

2.4. Stable route selection

Stable route selection can also contribute in reducing the total amount of routing overhead transmitted in the network. By selecting routes which last longer, the number of route recalculations due to link failure can be reduced. Most of the previous work done to provide stable routes in MANETs have been carried out with source routing protocols. ABR[15] and SSA[6] are two such protocols which attempt to provide stable routes using source routing. In these protocols the destination selects the route, which has travelled over the most stable links. We explore the effects of selecting stable routes in point-to-point routing. One way to select stable routes in a point-to-point manner is to restrict the flooding of RREQ packets over strong links only. To select strong link, we allow only the nodes which receive a RREQ packet over a strong link to further broadcast the packet. Therefore, the RREQ packets which reach the destination (or an intermediate node with a route to the destination) have travelled over strong route. This means that the destination (or the interme-

---

[4] Assuming all nodes have equal transmission range

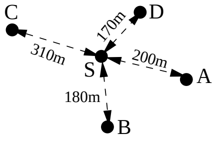


Figure 5: Illustration of stable route selection in LPAR

diate node) can send back a RREP over strong links, and a stable route between the source can be established. We define a link as being strong if the distance between the edges (nodes) in the link are less than a predefined Threshold Transmission Range[5](TTR). For example, in Figure 5, for $TTR = 300m$, nodes A, B and D form a strong link with node S. Therefore, these nodes can further broadcast any RREQ packets received from node S.

3. SIMULATION MODEL

The aim of our simulation study is to measure the performance of our routing strategy under changing network topology and investigate what levels of successful data delivery (and throughput) can be achieved under different network conditions. We compare the performance of LPAR under network scenarios, which have different levels of mobility, traffic and node density with a number of existing routing protocols and discuss how each protocol performs under each scenario.

3.1. Simulation Environment and Scenarios

The simulations were carried out in GloMoSim[1] simulation package. GloMoSim is an event driven simulation tool designed to carry out large simulations for mobile ad hoc networks. Our simulations were carried out for 50, 100, 200, 300, 400 and 500 node networks, migrating in a 1000m x 1000m boundary. IEEE 802.11 DSSS (Direct Sequence Spread Spectrum) was used with maximum transmission power of 15dbm at 2Mb/s data rate. In the mac layer IEEE 802.11 was used in DCF mode. The radio capture effects were also taken into account. Two-ray path loss characteristics was for the propagation model. The antenna hight is set to 1.5m and the radio receiver threshold is set to -81 dbm and the receiver sensitivity was set to -91 dbm according to the Lucent's wavelan card[2]. A random way-point mobility model was used with the node mobility ranging from 0 to 20m/S and pause time varied from 0 to 900S. The simulation was run for 900S for 10 different values of pause time and each simulation was averaged over eight different simulation runs using different seed values.
Constant Bit Rate (CBR) traffic was used to establish communication between nodes. Each CBR packet was 512 Bytes and the simulation was run for 10 and 20 different client/server pairs and each session was set to last for the duration of the simulation.

3.2. Performance Metrics

To investigate the performance of the routing protocols the following performance metrics were used:

- Packet Delivery Ratio (PDR): Ratio of the number of packet sent by the source node to the number of packets received by the destination node.
- Control (O/H): The number of routing packets transmitted through the network for the duration of the simulation.
- Packet Delivery Ratio (vs) Number of nodes: The percentage of packets successfully delivered as the number of nodes is increased for a chosen value of pause time.
- Control (O/H) (vs) Number of nodes: The number of control packet introduced into the network as the number of nodes is increased for a chosen value of pause time.
- End-to-End Delay: The average end to end delay for transmitting one data packet from the source to the destination

The first metric is used to investigate the levels of data delivery (data throughput) achievable each protocol under different network scenarios. The second metric will illustrate the levels of routing overhead introduced. The third and the forth metric are used to investigate the scalability of the protocols as the network grows in size. The last metric compares the amount of delay experienced by each data packet to reach their destination.

---

[5] TTR < maximum possible transmission range.

4. RESULTS

This sections gives a discussion on the simulation results we obtained for our routing strategies. To investigate the performance of LPAR with and without stable link strategy (in section 2.4), we ran two different versions of LPAR. These are: LPAR, which consists of sections 2.2 and 2.3, LPAR-S which sections 2.2, 2.3 and 2.4. The performance of our LPAR strategies where compared with LAR1 and AODV.

4.1. Packet Delivery Ratio Results

Figure 6, 7, 8 and 9 show the PDR achieved by each routing protocol as the number of nodes in the network was increased, for 10 CBR sources. In this scenario, for all node density levels, the PDR of all routing protocols are greater than 95%. The performance of each protocol converges to 100% when the mobility is reduced to zero (i.e. 900S pause time). LAR1 has the highest level of PDR. This is more evident in Figure 6 where the node density is lower than the other scenarios. This is because LAR1 stores multiple routes, where as the other protocols store a single route. The disadvantage of storing a single route when node density is low is that the nodes in the path to the destination have less chance of learning about a fresher route to the destination. This means that link failure between the intermediate nodes leading to the destination, may cause another route discovery. As a results some data packets maybe dropped, which means that PDR will be reduced. LPAR-S has the lowest delivery ratio in the 50N scenario. However, as the number of nodes are increased, LPAR-S performs as well as LAR1. This is because when the node density is low, the number routes found (or available) is less. Therefore, if route selection is done over strong links only, then the number of routes found will be less and in some situations the RREQ packets may not reach the destination (or an intermediate node to the destination).

Figure 10, 11, 12 and 13 show the PDR for 20 CBR sources. In this scenario, LPAR shows the best performance under low node density (i.e. 50 node scenario), and as the node density is increased, LPAR maintains over 95% PDR. LPAR-S, still under performs in the 50 nodes scenario, however, as the node density is increased its performance increases and performs as well as LPAR and AODV.

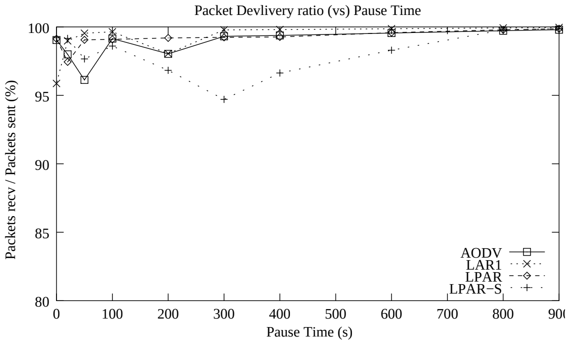


Figure 6: PDR for 50N and 10S

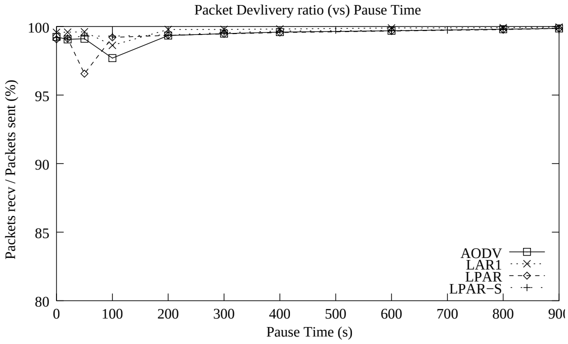


Figure 7: PDR for 100N and 10S

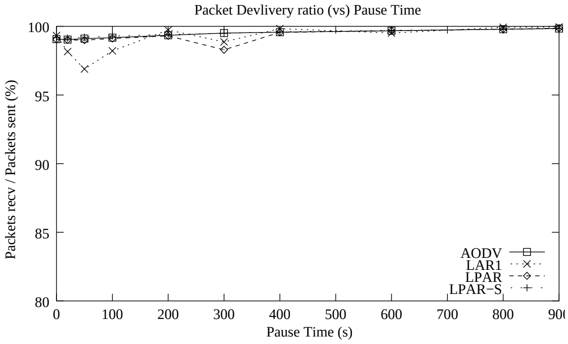


Figure 8: PDR for 200N and 10S

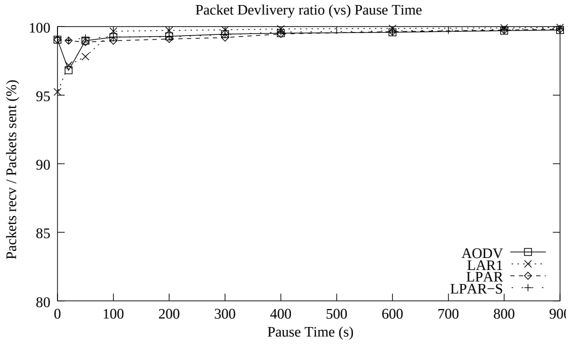


Figure 9: PDR for 300N and 10S

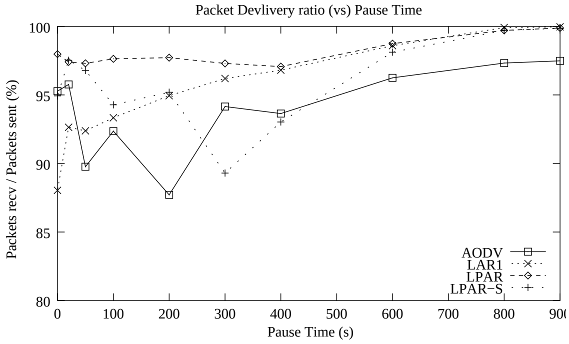


Figure 10: PDRfor 50N and 20S

Furthermore, in the high node density scenario (i.e. 13) it begins to out perform the other routing protocols. This increase in performance is due to the availability of more stable routes when compared to the least dense scenarios. AODV also performs well across all ranges of node density. However, it start to under perform LPAR and LPAR-S in the 300 node network scenario. LAR1 achieves the lowest levels of PDR in this scenario. This is more evident under the higher mobility (i.e. smaller pause times), where link failure rate is higher. Therefore, in this scenario, the point-to-point routing protocols clearly out perform the source routing protocol (i.e. LAR1).

4.2. Control Overhead Results

Figure 14, 15, 16 and 17 show the number of control packets introduced into the network by each routing protocol, for 10 CBR sources. In AODV, more overhead is introduced into the network than the other routing strategies. This is because, AODV does not take any measurements to reduce the route discovery region if the source and

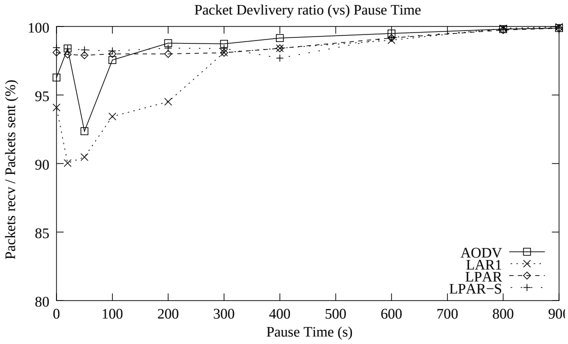


Figure 11: PDR for 100N and 20S

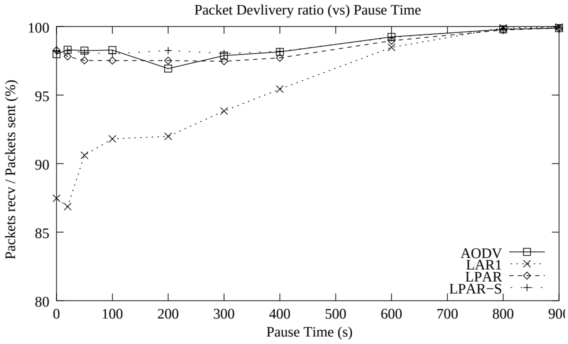


Figure 12: PDR for 200N and 20S

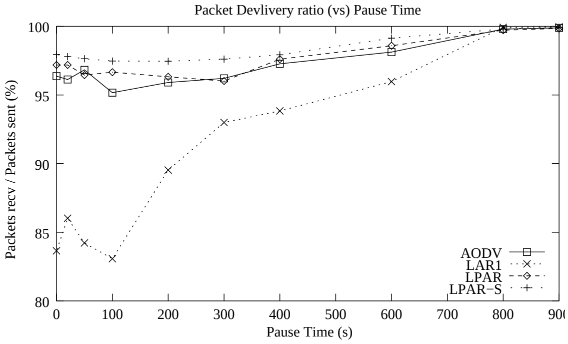


Figure 13: PDR for 300N and 20S

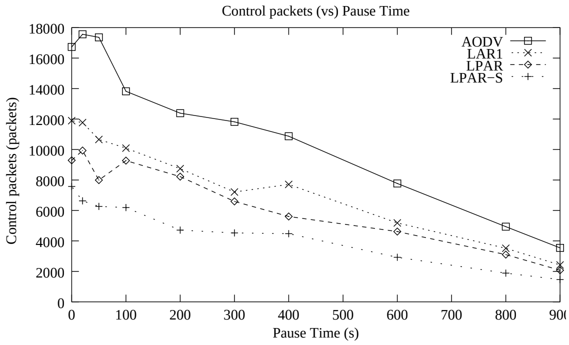


Figure 14: CTRL packets for 50N and 10S

the destination have recently communicated (or the source has location information about the destination). On the contrary, in LAR1 two factors contribute to reducing routing overhead. Firstly, nodes can have multiple routes to destinations (as discussed earlier), which may reduce the number of route discoveries initiated for each src/dest pair, whereas in AODV, each node only stores a single route. Secondly, in LAR1, if source nodes have location information about the required destination, they can use RZS (as described earlier), which minimises (or localises) the search area to a particular region. The advantage of this is that the number of nodes involved in broadcasting RREQ packets are reduced, which means that fewer control packets are transmitted. This also means more bandwidth to be available for the nodes that are not in the search area and reduce channel contention. LPAR and LPAR-S, which use the 3-state route discovery algorithm, produce less overhead than LAR1, despite only storing single routes. This is because in our 3-state route discovery algorithm, if unexpired location information is available, the source node will first attempt to discovery a route by unicasting rather in broadcasting (as previously described in the DURD algorithm in section 2.2). This means that fewer control packets are transmitted through the network. LPAR-S further reduces this overhead by flooding over links which have certain level of stability. The advantage of this is that route may last longer, which means fewer route recalculations will be required and fewer data packet will be dropped.

Figure 18, 19, 20 and 21 show the number of control packets introduced into the network by each routing protocol, for 20 CBR sources. It this scenario, it can be seen that LPAR-S continues

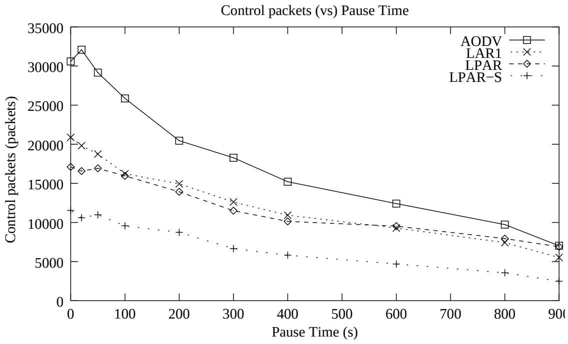


Figure 15: CTRL packets for 100N and 10S

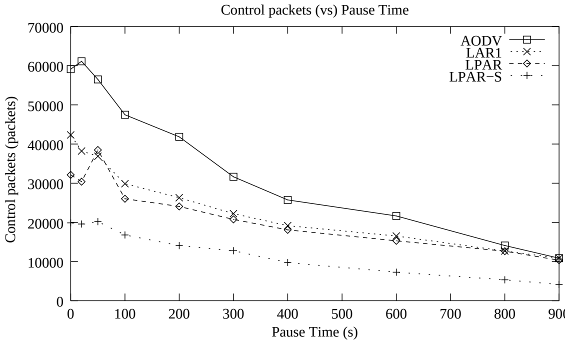


Figure 16: CTRL packets for 200N and 10S

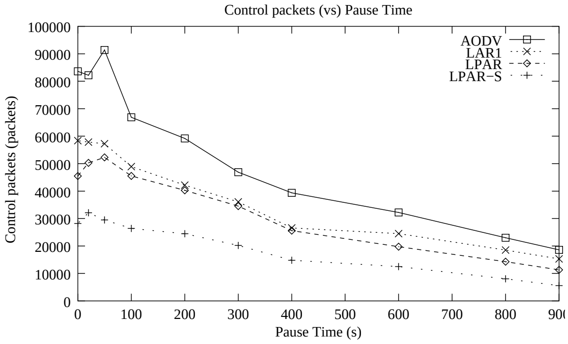


Figure 17: CTRL packets for 300N and 10S

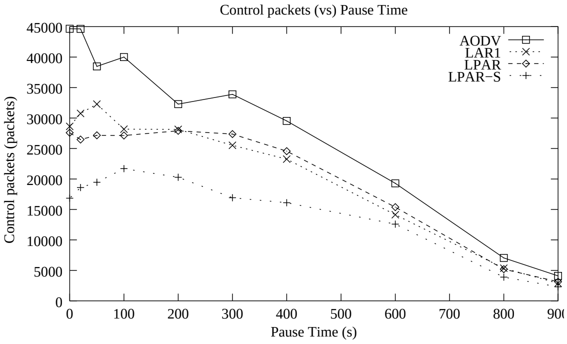


Figure 18: CTRL packets for 50N and 20S

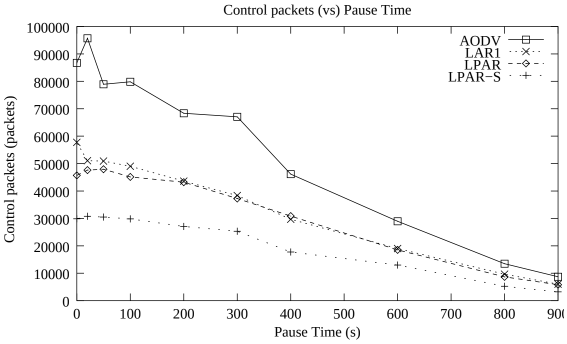


Figure 19: CTRL packets for 100N and 20S

to produce the least amount of overhead. Both LPAR and LAR1 show similar levels overhead in low density scenario, with LPAR performing better under higher mobility and LAR1 performing better during mid range mobility. LPAR starts to out perform LAR1 at higher node density. This is because at higher node density the DURD algorithm will have a better chance of forwarding the RREQ packet to the destination, which means that it will have a higher success rate for finding a route to the destination. Therefore, fewer control packets are transmitted when compared to using ERS or RSZ during route discovery. AODV continues to produce the highest level of control overhead in all scenarios. This is more evident during high mobility where AODV produces three times more control overhead than LPAR-S and two times more overhead than LPAR and LAR1. This result illustrates the valuability of exploiting location information during route discovery.

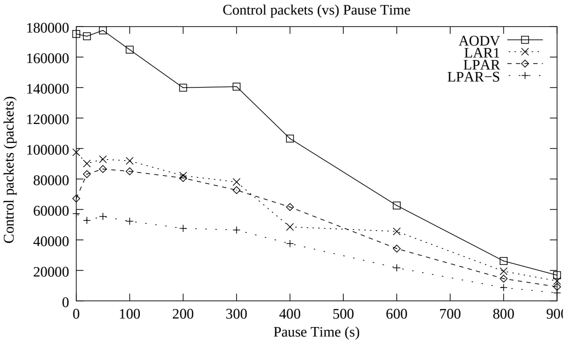


Figure 20: CTRL packets for 200N and 20S

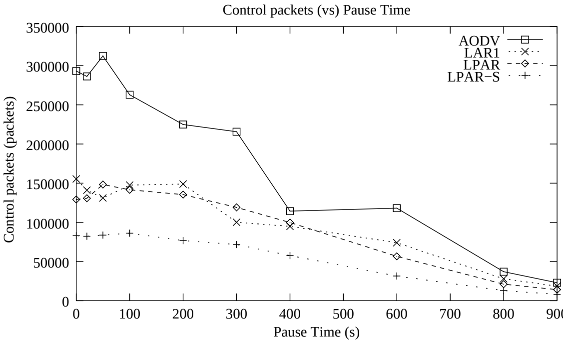


Figure 21: CTRL packets for 300N and 20S

4.3. Scalability Results

To further investigate the scalability of each routing protocol, PDR and control overhead was recorded for the worst case network scenario (i.e. under constant node mobility, 0 pause time), for up to 500 nodes. Figure 22 and 23 shows the PDR achieved for 10 source/dest pairs and 20 sources/dest pairs respectively. For the 10 source/dest scenario, LPAR and LPAR-S achieve over 98% PDR for all node density levels. LAR1 achieves its highest PDR for up to 200 nodes, after this is performance begins to drop. AODV also performs well across all node density levels. In the 20 source/dest scenario, LPAR, LPAR-S and AODV clearly out perform LAR1. LPAR-S shows the highest PDR and it maintains over 97% PDR . LPAR's performance is sightly less than LPAR-S during high node density. However, they both out perform AODV across all range of node density. Furthermore, AODV's performance starts to drop after 200 nodes. LAR1's highest PDR occurs at 100 nodes where it achieves 94%. However, after 100 nodes its performance continues to drop significantly, and by 500 nodes

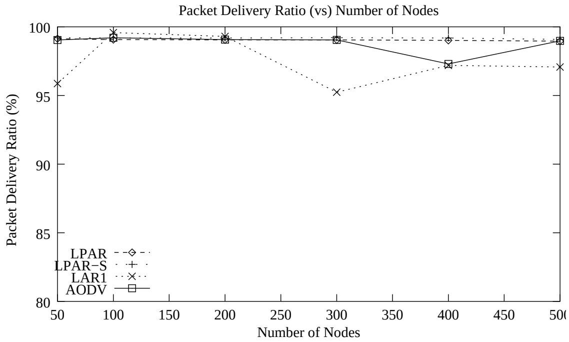


Figure 22: PDR for pause time = 0 and 10S

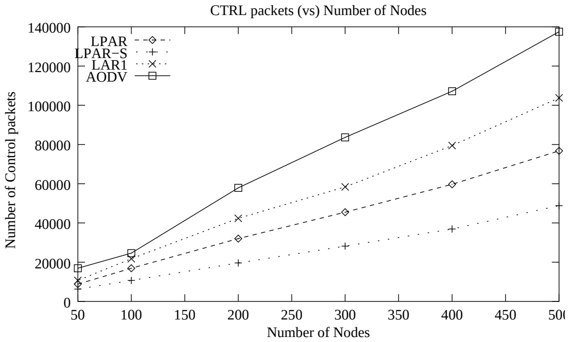


Figure 24: CTRL for pause time =0 and 10S

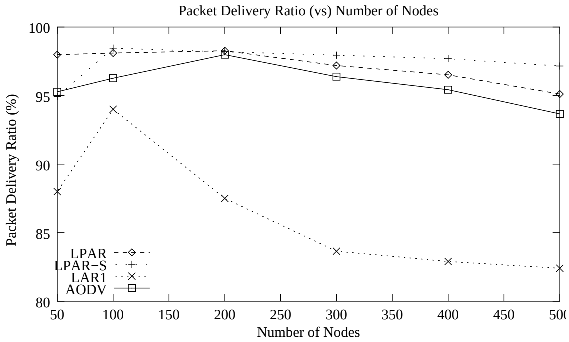


Figure 23: PDR for pause time =0 and 20S

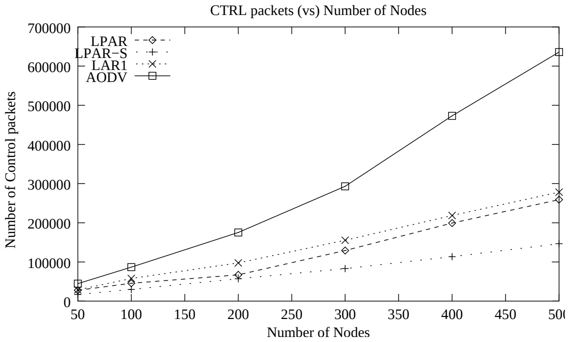


Figure 25: CTRL for pause time =0 and 20S

it performance has dropped to 83%.

Figure 24 and 25 shows the number of control packets introduced into the network for 10 src/dest pairs and 20 sources/dest pairs respectively. From these figures it can be seen that as the node density is increased the performance difference between each routing strategy becomes more significant. AODV has higher control overhead than LAR1, LPAR and LPAR-S for both the 10 src/dest scenario and the 20 src/dest scenario where it produces three times more overhead than LPAR-S and over two times more overhead than LPAR and LAR1. LPAR-S continues to produce the least amount of overhead for all node density scenario. LPAR also shows fewer overheads than LAR1 and AODV. Therefore, from these results it can be seen that both LPAR and LPAR-S are more scalable than AODV and LAR1 as the level of traffic and node density increases in the network.

4.4. Delay Results

Figure 26 and 27 shows the average end-to-end delay experienced by each data packet for 10 src/dest pairs and 20 src/dest pairs in a 100 node network respectively. As expected, all protocols experienced larger delays during high mobility, since more frequent link failure may cause route recalculation. This means than each packet may experience longer delays before they reach their destination. AODV has lowest end-to-end delay compared to the other protocols. This is because, AODV always uses the shortest route to the destination and it only maintains a single route, whereas LAR1 can store multiple route. This means that if optimal route fails (the one with the shortest src/dest path), an alternate route from the route cache may be use. This means that some packet may travel over longer routes to reach the destination. Similarly in LPAR and LPAR-S if the primary route fails, some packet may travel over the secondary route, which may be longer in length. Therefore, they may experience slightly longer delay. From the figures we can see that LPAR and LPAR-S have on average about 5ms more delay across all range of mobility. However, by using a secondary route, LPAR and LPAR-S are able to successfully transmit more data packets, and

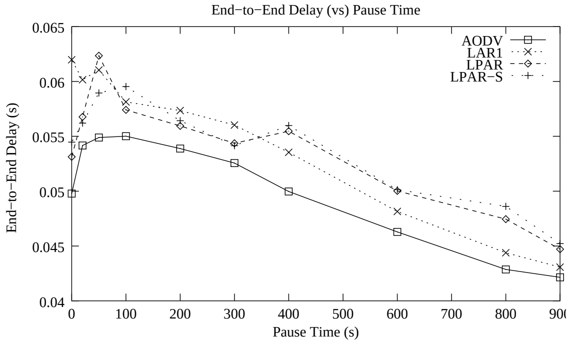


Figure 26: Average End-to-end delay for 10S

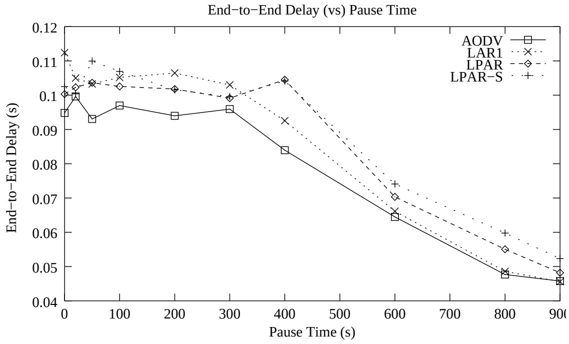


Figure 27: Average End-to-end delay for 20S

reduce the number of route recalculations, which means fewer control packets.

5. OPTIMISATIONS AND IMPROVEMENTS

In this paper, we have introduced a number of different strategies to reduce routing overhead and power used during route discovery. In this section, we describe a number of different optimisation and Alternative strategies, which can further improve the performance of LPAR.

5.1. Forwardibility Algorithm for DURD

In the DURD algorithm (section 2.2), RREQ packets are forwarded towards the destination using the intermediate nodes which have a minimum required degree (i.e. node density, we use $deg > 1$ in this case) and the intermediate distance[6] is less than τ. To further increase the possibility of the RREQ packets reaching the destination (in the DURD algorithm), we introduce a forwardability algorithm, to allow the forwarding node to make further decisions about whether the next forwarding node chosen is able to forward the RREQ towards the destination. To do this, each node calculates a forwardibility factor (instead of a node degree) and exchanges this information with the neighbouring nodes[7]. To describe the forwardability algorithm, we refer to Figure 28. To calculate the forwardibility, each node defines starts by defining four point at a distance R away from its current location. These points are at north, east, south and west as shown in Figure 28. Using these points the nodes will determine if they have any neighbour which lead to (or approximately lead to) those directions, using their neighbour table. This is done by mapping each neighbour in the coordinate plane, and determining which neighbours are closest to which one of the four points. This can be done either by comparing the distance between each neighbour with each of the four points or calculating the angle θ between the neighbour and one of the four points, neighbours which have $\theta < 45$ degrees can be assigned for one of the four directions[8]. For example, if node B is closest to point E (or $\theta_{BE} < 45$), then node B will be a forwarding node for the destinations in the direction of E. Once all the neighbouring nodes have been sorted for each direction, a forwardability factor will be determined for each direction. One way to do this is to count the number of nodes for each

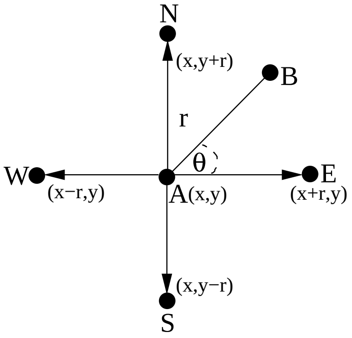


Figure 28: Dimensions used to Calculate the forwardability factor

---

[6] The intermediate distance is the distance between the forwarding node and the next node which is chosen as the next forwarding node

[7] This is done using the existing hello beacon messages and the routing packets

[8] In networks with high node density θ can be further reduced to further increase the accuracy of the forwardability algorithm

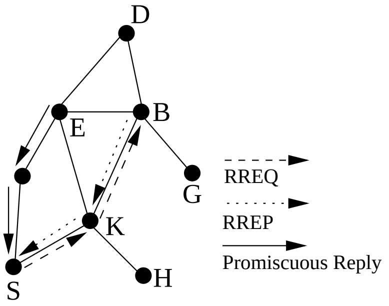


Figure 29: Illustration of promiscuous route reply

direction and exchange them with the neighbouring nodes. Therefore, upon receiving a directed RREQ packet, the receiving node can determine if the forwarding nodes have outgoing links which lead to the destination.

5.2. Promiscuous Reply for DURD

In the DURD algorithm, only the nodes which receive the RREQ packet (in the path leading to the destination), can send a reply back to the source. To further increase the number of replys sent back while minimising a route reply storm, we can allow the nodes which overhears (promiscuously) the directed RREQ packet to send a RREP back to the source, if it has a route to the required destination and also to the source, and also if it is using a different downlink than the one used in the directed RREQ[9]. The advantage of selecting this type of node for a promiscuous route reply is that not only they increase the chance of receiving a route reply at the source, but also they can provide an alternate path (if a number of route replies are received at the source). Therefore, the source can select the best route or distribute its load between a number of different path. For example, if node S (see Figure 29) send a directed RREQ towards node D. Assume that node B has a route to node D, it will the send a RREP back to the source. Now assume that node E also has a route to node D and also to the source node S. In this case when node E will send a promiscuous route reply back to the node S.

5.3. Alternate Route Selection using a Suggestion List

In the GARS algorithm (section 2.3), when a route reply is sent back, each intermediate node will select one of its neighbours to be used as secondary link if the primary link fails. This is done by selecting a neighbour, which is expected to be in transmission range with both the uplink and the downlink node. The node in the downlink can accept the secondary route if they are in transmission range (e.g. node L accepting node A as a secondary route in figure 4). In this strategy it is possible that in some scenarios the secondary route is not in transmission range with the uplink (e.g. node B assumes that nodes A and E are in transmission range from their distance), this can be due to factors such as interference or barriers. To increase the availability of the uplink and the downlink, each intermediate can include a Suggestion List in the RREP packet. The suggestion list is a list of nodes in the direction of the next hop in the RREP. For example, node E (see Figure 4) can select node A and K for the suggestion list. Node B will then choose node A from this list since it is expected to be within transmission range of node L, and forward it (along with its own suggestion list) to node L. Therefore, if node L chooses node A to be a secondary route, the probability of the link between node A and E being available will be higher. The suggestion list can be further optimised, by selecting the neighbours which have certain levels of mobility, stability and/or load.

5.4. Dynamic Power Allocation

In current protocols such as AODV, DSR, LAR1, transmissions for all types of packets (e.g. data and control) is carried out with maximum transmission power. Transmission of control packets at the maximum possible power may have the advantage of allowing the nodes in the MANETs to be aware of their surrounding node topology. However, transmission of data packets at maximum power may not have any advantage, but rather just consume extra battery power. We can reduce the amount of battery power consumed during data transmission by estimating the minimum amount of power required to successfully transmit a packet between two intermediate nodes. To do this, we calculate the power required for a given distance between two nodes (using GPS coordi-

---

[9] The node which overhears the directed RREQ packet can check to see which node has forwarded the RREQ. If the forwarding node is the same as the route it knows about the source, then reply is not sent

nates) and add the power losses (based on the propagation model). This will then give us the minimum power required to transmit a packet between two chosen nodes.

6. CONCLUSIONS

This paper describes a new routing strategy for mobile ad hoc networks. We present LPAR routing protocol, which introduces a number of different strategies to reduce route discovery overhead and the power consumed by each node. We compared LPAR with LAR1 and AODV using simulations. Our results show that LPAR and LPAR-S produce fewer overhead than LAR1 and AODV, while still maintaining high levels of data delivery when node density is low. In high node density both LPAR and LPAR-S produce fewer overheads and maintain higher levels of data throughput than AODV and LAR1.

REFERENCES

[1] Glomosim scalable simulation environment for wireless and wired network systems. In *http://pcl.cs.ucla.edu/projects/glomosim/*.

[2] Orinoco pc-card http://www.lucent.com/orinoco.

[3] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. A Review of Current On-Demand Routing Protocols. In *Networking - ICN 2001, First International Conference*, Colmar, France, 2001.

[4] George Aggelou and Rahim Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *ACM International Workshop on Wireless Mobile Multimedia (WoWMoM)*, pages 26–33, 1999.

[5] S. Das, C. Perkins, and E. Royer. Ad Hoc On Demand Distance Vector (AODV) Routing. In *Internet Draft, draft-ietf-manet-aodv-11.txt*, work in progress, 2002.

[6] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (ssa) for ad hoc mobile networks. In *IEEE Personal Communication, pp. 36-45 Feb.*, 1997.

[7] Z.J. Hass and R Pearlman. Zone Routing Protocol for Ad-Hoc Networks. In *Internet Draft, draft-ietf-manet-zrp-02.txt*, work in progress, 1999.

[8] Mario Joa-Ng and I-T Lu. A Peer-to-Peer Zone-based Two-level Link State Routing for Mobile Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8), 1999.

[9] D. Johnson, D. Maltz, and J. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. In *Internet Draft, draft-ietf-manet-dsr-07.txt*, work in progress, 2002.

[10] Yong-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, 1998.

[11] S. Lee and M. Gerla. Aodv-br: Backup routing in ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Chicago, IL, Sepember.*, 2000.

[12] Shree Murthy and J. J. Garcia-Luna-Aceves. A routing protocol for packet radio networks. In *Mobile Computing and Networking*, pages 86–95, 1995.

[13] S-Y. Ni, Y-C. Tseng, Y-S. Chen, and J-P. Sheu. The broadcast problem in a mobile ad hoc network. In *Fifth annual ACM/IEEE international conference on Wireless Networks*, 2002.

[14] C.E. Perkins and T.J. Watson. Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94 Conference on Communications Architectures*, London, UK, 1994.

[15] C. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. In *IEEE 15th Annual International Pheonix Conf. pp 480-86*, 1996.

[16] Seung-Chul Woo and Suresh Singh. Scalable Routing Protocol for Ad Hoc Networks. *accepted for publication in Journal of Wireless Networks (WINET).*