Displacement-based Route Update Strategies for Proactive Routing Protocols in Mobile Ad Hoc Networks

Mehran Abolhasan[1] and Tadeusz Wysocki[1]
[1] University of Wollongong, NSW 2522, Australia
E-mail: mehran@titr.uow.edu.au, wysocki@uow.edu.au

***Abstract*—This paper presents a new route update strategy for performing proactive route discovery in Mobile Ad hoc Networks (MANETs). In this strategy, the rate at which route updates are sent into the network is controlled by how often a node changes its location by a required distance. We refer to this updating strategy as Minimum Displacement Update Routing (MDUR). We implemented MDUR on top of the Fisheye State Routing (FSR) protocol and investigated its performance by simulation. The simulations where performed in a number of different scenarios, with varied network mobility, density, traffic and boundary. Our results indicate that MDUR has lower levels of control overhead than FSR and achieves higher levels of throughput as the density and the level of traffic in the network is increased.**

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs), are made up of a number of nodes, which are capable of performing routing without using a dedicated centralised controller or a base station. This key feature of these networks enable them to be employed in places where an infrastructure is not available, such as in disaster relief and on battle grounds. However, the dynamic nature of these networks and the scarcity of bandwidth in the wireless medium, along with the limited power in mobile devices (such as PDA's or laptops) makes routing in these networks a challenging task. A Routing protocols designed for MANETs must work consistently as the size and the density of the network varies and efficiently use the network resources to provide each user with the required levels of quality of service for different types of applications used.

With so many variables to consider in order to design an efficient routing protocol for MANETs, a number of different types of routing strategies have been proposed by various authors. These protocols can be classified into three groups; global/proactive, on-demand/reactive and hybrid. Most proactive routing protocols are based on the link state and distance vector algorithms. In these protocols, each node maintains up-to-date routing information to every other node in the network by periodically exchanging distance vector or link state information using different updating strategies (discussed in the following section).

In on-demand routing protocols each node only maintains active routes. That is, when a node requires a route to a particular destination, a route discovery is initiated. The route determined in the route discovery phase is maintained while the route is still active (i.e. the source has data to send to the destination). The advantage of on-demand protocols is that they reduce the amount of bandwidth usage and redundancy by determining and maintaining routes when they are required. These protocols can be further classified into two categories: source routing and hop-by-hop routing. In Source routed on-demand protocols [12][18], each data packets carry the complete source to destination address. Therefore, each intermediate node forwards these packets according to the information kept in the header of each packet. This means that the intermediate nodes do not need to maintain up-to-date routing information for each active route in order to forward the packet towards the destination. Furthermore, nodes do not need to maintain neighbour connectivity through periodic beaconing messages. The major drawback with source routing protocols is that in large networks they do not perform well. This is due to two main reasons; firstly as the number of intermediate nodes in each route grows, then so does the probability of route failure. To show this let $P(f) = a.n$, where $P(f)$ is the probability of route failure, a is the probability of a link failure and n is the number of intermediate nodes in a route. From this[1], it can be seen that as $n \rightarrow \infty$, then $P(f) \rightarrow \infty$. Secondly, as the number of intermediate nodes in each route grows, then the amount of overhead carried in each header of each data packet will grow as well. Therefore, in large networks with significant levels of multihoping and high levels of mobility, these protocols may not scale well.

In hop-by-hop routing (also known as point-to-point routing) [6], each data packet only carries the destination address and the next hop address. Therefore, each intermediate node in the path to the destination uses its routing table to forward each data packet towards the destination. The advantage of this strategy is that routes are adaptable to the dynamically changing environment of MANETs, since each node can update its routing table when they receiver fresher topology information and hence forward the data packets over fresher and better routes. Using fresher routes also means that fewer route recalculations are required during data transmission. The disadvantage of this strategy is that each intermediate node must must store and maintain routing information for each active route and each node may require to be aware of their surrounding neighbours through the use of beaconing messages.

Hybrid routing protocols have been proposed to increase the scalability of routing in MANETs [11][19][13][9] [15]. These protocols often can behave reactively and proactively at different times and they introduce a hierarchical routing structure to the network to reduce the number of retransmitting nodes during route discovery or topology discovery. Each node periodically maintains the nearby topology by employing a proac-

---

[1] Assuming that the intermediate nodes have a probability of a link failure of $a > 0$

tive routing strategy (such as distance vector or link state) and maintain approximate routes or on-demand routes for far away nodes.

In this paper, we propose a new route updating strategy to perform proactive route discovery in mobile ad hoc networks. In this strategy, the rate at which route updates are sent is controlled by the rate of displacement of each node. This is determined by using the services of a GPS. In [3], we briefly mentioned MDUR, in this paper we give a full description of this strategy, and investigate its performance under different network scenarios using a simulation tool.The rest of this paper is organised as follows. In section II, we describe a number of different route update strategies proposed in the literature. Section III describes our route updating strategy. Section IV describes the simulation environment, parameters and performance metric used to investigate the performance of our route update strategy. Section V presents the discussion of our simulation results and section VI presents the conclusions of the paper.

II. RELATED WORK

Proactive route discovery provides pre-determined routes for every other node (or a set of nodes) in the network at every node. The advantage of this is that end-to-end delay is reduced during data transmission, when compared to determining routes reactively. Simulation studies [5][8][4],which have been carried out for different proactive protocols show high levels of data throughput and significantly less delays than on-demand protocols (such as DSR) for networks made up of up to 50 nodes with high levels of traffic. Therefore, in small networks using real-time applications (e.g. video conferencing), where low end-to-end delay is highly desirable, proactive routing protocols may be more beneficial. In this section, we describe a number of different route update strategies proposed in the literature to perform proactive routing.

A. Global updates

Proactive routing protocols using global route updates are based on the link state and distance vector algorithms, which were originally designed for wired networks. In these protocols, each node periodically exchanges its routing table with every other node in the network. To do this, each node transmits an update message every T seconds. Using these update messages, each node then maintains its own routing table, which stores the freshest or best route to every known destination. The disadvantage of global updates is that they use significant amount of bandwidth. Since they do not take any measures to reduce control overheads. As a result data throughput may suffer suffer significantly, especially as the number of nodes in the is increased. Two such protocol are DSDV [16] and WRP [14].

B. Localised updates

To reduce the overheads in global updates, a number localised updating strategies where introduced in protocols such as GSR[10] and FSR[8]. In these strategies, route update propagation is limited to a localised region. For example, in GSR each node exchanges routing information with their neighbours only, thereby eliminating packet flooding methods used in the global routing. FSR is a direct descendent of GSR. This protocol attempts to increase the scalability of GSR by updating the nearby nodes at a higher frequency than the node which are located far away. To define the nearby region, FSR introduces the fisheye scope (as shown in Figure 1). The fisheye scope covers a set of nodes which can be reached within a certain number of hops from the central node shown figure 1. The update messages with greater hop counts are sent at a lower frequency. This reduces the accuracy of the routes in remote locations, however, it significantly reduces the amount of routing overheads disseminated in the network. The idea behind this protocol is that as the data packets get closer to the destination the accuracy of the routes increases. Therefore, if the packets know approximately what direction to travel, as they get close to the destination, they will travel over a more accurate route and have a high chance of reaching the destination.

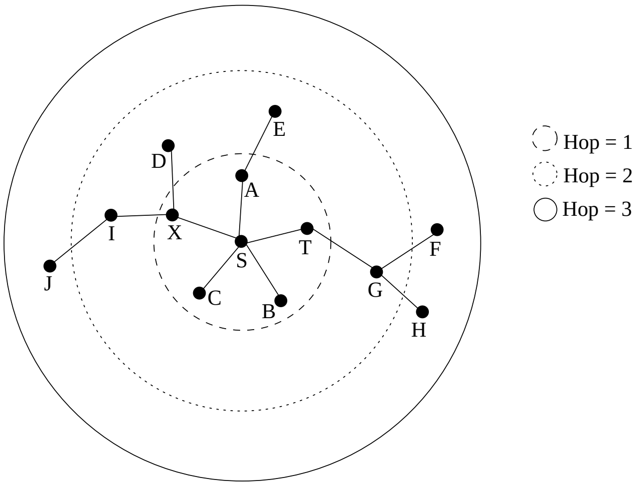


Fig. 1. Illustration of the fisheye scope in FSR

C. Mobility based updates

Another strategy which can be used to reduce the number of update packets is introduced in DREAM [4]. The author proposes that routing overhead can be reduced by making the rate at which route updates are sent to the speed at which each node travels. Therefore, the nodes which travel at a higher speed disseminate more update packets than the ones that are less mobile. The advantage of this strategy is that in networks with low mobility this updating strategy may produce fewer update packets than using a static update interval approach such as DSDV. Similar to FSR, in this protocol, updates are sent more frequently to nearby nodes than the ones located far away.

D. Conditional or Event-driven updates

The number of redundant update packets can also be reduced by employing a conditional (also known as event-driven) based update strategy [16][7]. In this strategy a node sends an update if certain different events occur at any time. Some events which can trigger an update is when a link becomes invalid or when a new node joins the network (or when a new neighbour is detected). The advantage of this strategy is that if the network topology or conditions are not changed, then no update packets are sent. Therefore, eliminating redundant periodic update dissemination into the network.

III. PROPOSED STRATEGY

In this section, we propose Minimum Displacement Update Routing (MDUR). This route update strategy attempts disseminate route update packet into to the network when they are required rather than using purely periodic updates. This is achieved by making the rate at which updates are sent proportional to the rate of displacement. That is, the more a node changes location by a threshold distance the more updates are transmitted into the network. The rate of displacement can be measured using a Global Positioning System (GPS).

A. Overview and Definition

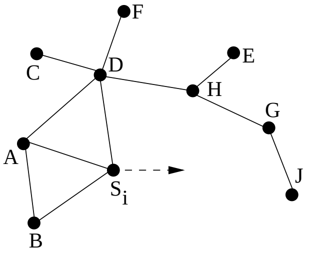
The idea behind this strategy is to reduce the amount of periodic route updates by restricting the update transmission to nodes which satisfy the following conditions:

1) A node experiences or creates a significant topology change.
2) A node has not updated for a minimum threshold time.

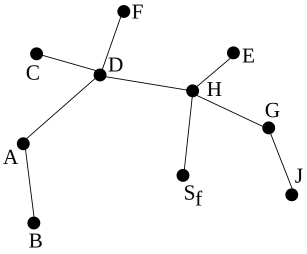
In the first condition we assume that a node experiences a significant topology change if it has migrated by a minimum distance from one location to another location. By migrating from one location to another the routes connected to the migrating node (and the route to the migrating node itself) may significantly change. Therefore, the migrating node is required to transmit an update packet through the network (or parts of the network) to allow for recalculation of more accurate routes. To illustrate how MDUR works, suppose node S (see figure 2) migrates from one location to another as shown. From this migration it can be seen that the neighbour topology of node S has changed, which has also significantly changed the topology of the network. Therefore, the dissemination of an update packet at this time will be beneficial as each node in the network can rebuild their routing tables and store more accurate routes.

B. Description of MDUR Algorithm

With MDUR, each node starts by recording its current location and sets it as its previous location. They will also record their current velocity and time. Using this information each node determines when the next update should be sent. When this update time is elapsed the nodes check to see if their migration distance is greater than the required threshold distance. If yes, an update is sent. Otherwise, no update is sent and the next update time is estimated according to the current location and velocity of the node. If the current velocity is zero the node can assume a maximum velocity or set a minimum wait time according to an update time constant, which has been used in the MDUR algorithm. The *MDUR* algorithm is outlines below.



(a) Illustrating Initial position for node S



(b) Illustrating final position for node S

Fig. 2. Illustration of node migration in MDUR

```
Algorithm MDUR
(∗ The MDUR algorithm ∗)
$L_p \leftarrow$ Previous location
$L_c \leftarrow$ Current location
$L_p \leftarrow L_c$
$D_T \leftarrow$ The threshold distance
Disseminate update packet
$V \leftarrow$ speed of node
$T_c \leftarrow$ current time
$\tau \leftarrow (\frac{D_T}{V}) + T_c$
while (node is online)
    wait until $T_c = \tau$
    $L_c \leftarrow$ current location
    if $(dist(L_c, L_p) \geq D_T$
        Disseminate update packet
        $L_p \leftarrow L_c$
        $\tau \leftarrow (\frac{D_T}{V}) + T_c$
    else
        $\tau \leftarrow (\frac{D_T - dist(L_c, L_p)}{V}) + T_c$
```

Displacement updates are more beneficial than using updates based purely on mobility (i.e. speed [4]). This is because this strategy attempts to send an update when a topology change occurs. To show this, suppose node S (figure 2) moves rapidly towards node A for a short time such that $dist(L_c, L_p) < D_T$. Furthermore, it moves in such a way that it maintains its links to nodes B and D. Now, assuming that there are no interference during this time and nodes A, B and D stay stationary, then the topology of node S will not change. Therefore, an update is not required in this network. However, in this case a purely mobility based strategy such as in [4] an update may be disseminated and it may continue to sent updates even if node S moves back and forward between these two point. On contrary, in this scenario in MDUR no updates will be sent.

C. Implementation Decision

To evaluate the performance of our route updating strategy, we implemented MDUR on top of FSR, which we refer to as Hierarchical MDUR (HMDUR). Recall that FSR disseminates two types of update packet; Intrascope update packets which propagate within the fisheye scope and Interscope packets which propagate through the entire network. Therefore,

TABLE I

FISHEYE STATE ROUTING SIMULATION PARAMETERS

Number of scopes	2
Intrascope update interval	5S
Interscope update interval	15S
Neighbour timout interval	15S

TABLE II

HIERARCHICAL MDUR SIMULATION PARAMETERS

Number of scopes	2
Intrascope max timeout interval	10S
Interscope max timeout interval	30S
Minimum intrascope migration	30M
Minimum interscope migration	200M

we introduced two types of displacement updates; one for the intrascope and one for the interscope, and we modified the MDUR algorithm to disseminate these two updates. To initiate each of these updates we also used two different threshold distances; D_{intra} and D_{inter} for the intrascope and interscope updates respectively. To initiate the intrascope updates more frequently than interscope updates, we set D_{intra} to be significantly less than D_{inter}. Table I and II illustrate the parameters used in FSR[2] and HMDUR.

The *HMDUR* algorithm is outlined below.

```
Algorithm HMDUR
(* The HMDUR algorithm *)
$L_{intra} \leftarrow$ Location at last inter-update
$L_{inter} \leftarrow$ Location at last inter-update
$L_c \leftarrow$ Current location
$L_{intra} \leftarrow L_c$
$L_{inter} \leftarrow L_c$
$D_{intra} \leftarrow$ The intrascope threshold distance
$D_{inter} \leftarrow$ The interscope threshold distance
Disseminate intrascope update packet
Disseminate interscope update packet
$V \leftarrow$ speed of node
$T_c \leftarrow$ current time
$\tau_{intra} \leftarrow (\frac{D_{intra}}{V}) + T_c$
$\tau_{inter} \leftarrow (\frac{D_{inter}}{V}) + T_c$
while (node is online)
    wait until a timer expires
    if ($\tau_{intra} = expired$)
        if ($dist(L_c, L_{intra}) \geq D_{intra}$
            Disseminate intrascope update
            $L_{intra} \leftarrow L_c$
            $\tau_{intra} \leftarrow (\frac{D_{intra}}{V}) + T_c$
        else
            $\tau_{intra} \leftarrow (\frac{D_{intra} - dist(L_c, L_{intra})}{V}) + T_c$
    if ($\tau_{inter} = expired$)
        if ($dist(L_c, L_{inter}) \geq D_{inter}$
            Disseminate interscope update
            $L_{inter} \leftarrow L_c$
            $\tau_{inter} \leftarrow (\frac{D_{inter}}{V}) + T_c$
        else
            $\tau_{inter} \leftarrow (\frac{D_{inter} - dist(L_c, L_{inter})}{V}) + T_c$
```

IV. SIMULATION MODEL

The aim of our simulation studies were to investigate the performance of our route update strategy under different levels of node density, traffic, mobility and network boundary. We simulated FSR-DU and FSR for each scenario in order to differentiate their performance. The simulations parameters and performance metrics are described in the following sections.

A. Simulation Environment and Scenarios

The Glomosim simulation tool was used to carry out our simulations [1]. 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 and 100 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 used for the propagation model. The antenna hight is set to 1.5m, 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 five 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, the simulation was run for 10 different client/server pairs and each session was set to last for the duration of the simulation.

B. 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.
- Normalised routing overhead (O/H): The amount of routing overhead transmitted through the network for each data packet successfully delivered to the destination.
- 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 last metric compares the amount of delay experienced by each data packet to reach their destination.

V. SIMULATION RESULTS

This section presents our simulation results. The aim of this simulation analysis is to compare the performance of HMUR with FSR under different network scenarios.

---

[2] The FSR parameters were set according to the ietf internet draft number 3

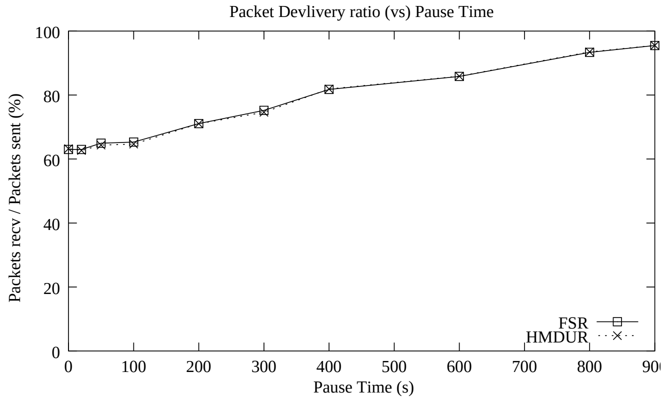


Fig. 3. PDR for 10S and 50N

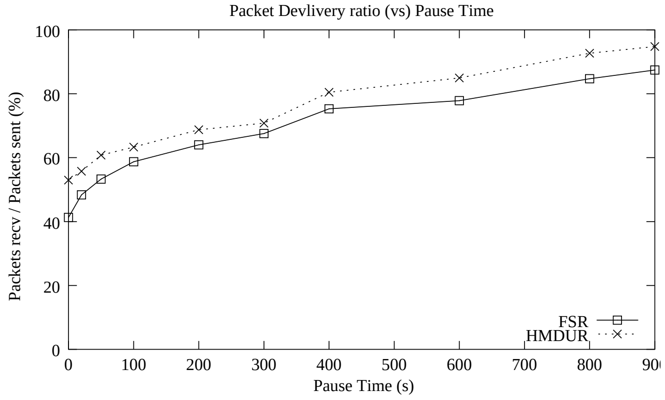


Fig. 4. PDR for 10S and 100N

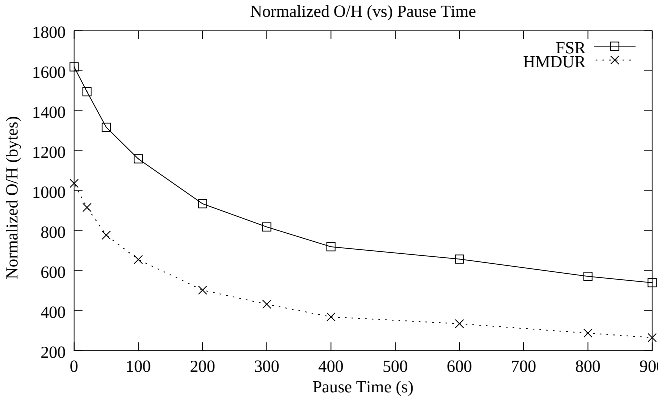


Fig. 5. Normalised O/H for 10S and 50N

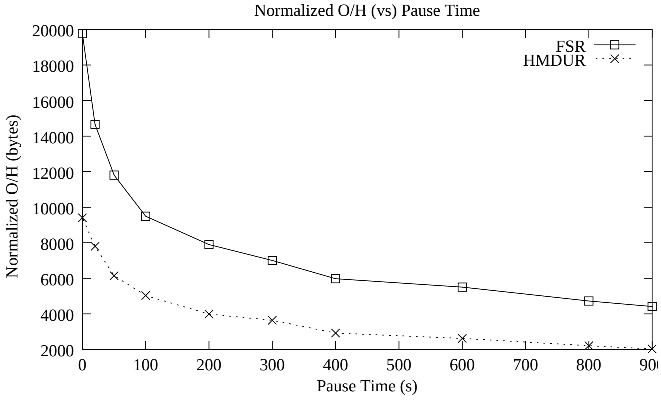


Fig. 6. Normalised O/H for 10S and 100N

A. Packet Delivery Ratio

The graphs in Figure 3 and 4 illustrate the PDR results obtained for the 1000m x 1000m boundary. In the 50 node scenario, FSR and HMDUR show similar levels of PDR However, in the 100 node network scenario, HMDUR starts to outperform FSR. This is because HMDUR still maintains a similar level of PDR as in the 50 node scenario, whereas FSR has shown a significant drop in performance when compared to the 50 node scenario. This drop in performance is evident across all different levels of pause time. This is because under high node density the periodic updating strategy in FSR starts to take away more of the available bandwidth for data transmission than HMDUR. Furthermore, more updates may increase channel contention, which can result in more packets being dropped at each intermediate node.

B. Normalised Control Overhead

The graphs in Figures 5 and 6 illustrate the Normalized routing overhead experinced in the 1000m x 1000m boundary. In our simulation, the maximum update intervals for the intrascope and interscope is set to be half of that of FSR. Therefore, under high mobility (i.e. 0 pause time) if purely periodic updates where used in HMDUR, the routes produced would have been less accurate, which may have resulted in a drop in throughput. However, adapting the rate of updates by each node to the rate of its displacement, allows the nodes to send more updates when they are required (i.e. during high mobility). This means that the accuracy of the routes will be high during high mobility where nodes are more likely to migrate more frequently and when mobility is low less updates are sent. From the results shown in Figures 5 and 6 it can be seen that HMDUR produces less overhead than FSR, across all different levels of pause time and node density.

C. Delays

The graphs in Figure 7 and 8 illustrate the end-to-end delay experinced in the 1000m x 1000m boundary. These results show that in HMDUR each data packet experinces lower end-to-end delay than in FSR. The lower delay experienced is due to the higher level of accessability to the wireless medium. This is because in HMDUR each node generates less route updates than it FSR, which means there is less contention for the channel when a data packet is received. Therefore, each node can forward the data packet more frequently.

VI. ALTERNATE STRATEGIES AND IMPROVEMENTS

One way to increase the scalability of proactive routing protocols is by maintaining approximate routes to each destination rather than exact routes. In [8] and [4], each node maintains approximate (or less accurate) information to far away destinations, since the updates from far away nodes are received less frequently. Similarly, in HMDUR, nodes maintain approximate routing information to nodes located far away by using the interscope displacement metric. In this section, we propose

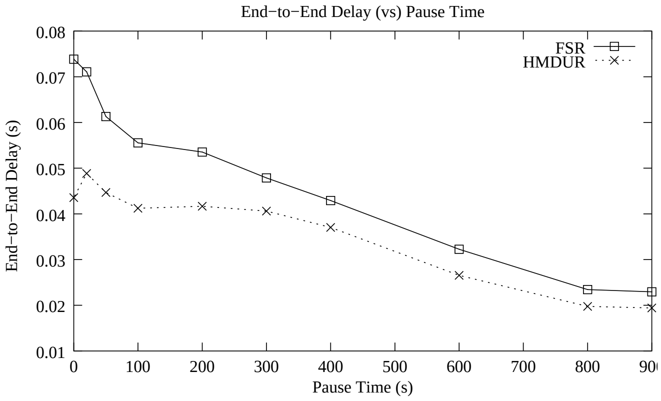


Fig. 7. Delays O/H for 10S and 50N

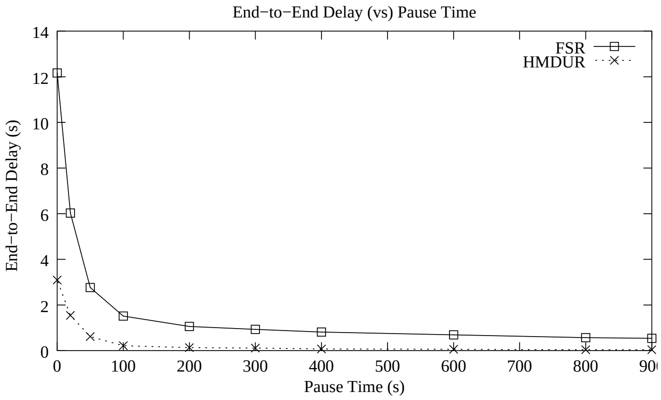


Fig. 8. Delays O/H for 10S and 100N

a number of alternative strategies and improvements for MDUR and HMUR based on approximate routing.

A. Minimum Topology Change Updates

Another way to determine if an update is required is by monitoring the nearby topology and disseminating update packets only when a minimum level of topology change occurs.To do this we introduce Minimum Topology Change Updates (MTCU). This strategy assumes that each node maintains an intrascope and interscope topology like FSR. However, instead of using purely periodic updates, the rate at which updates are sent is proportional to a topology metric. MTCU is made up of two phases: these are startup phase and maintenance phase. The start up phase is initiated when a node enters the network (or when it comes online). During this phase, each node starts by recording its location and sends three updates, which are: neighbour update, intrascope update and interscope update. Each node then counts the number of neighbouring nodes and the number of nodes in their intrascope. During the maintenance phase, the neighbouring topology is periodically monitored and the number of changes is recorded. These changes can include: discovery of new neighbour or the loss of a link. If a significant change in the neighbouring topology is experienced an intrascope update is sent. Furthermore, each node monitors its intrascope topology and count the number of changes, such as the number of nodes in the intrazone and the number of route changes for each destination. If the intrascope has changed significantly then an interscope update is sent. Note that the each node maintains its neighbour connectivity through beaconing messages. However, the rate at which intrascope and interscope updates are disseminated is dependent on the rate at which neighbouring or intrascope topology changes, and periodic updates can be used only if each node has not sent an intrascope or interscope update for long time[3]. Therefore, reducing the number of redundant updates if no changes occur. This also means that fewer periodic updates maybe transmitted when compared to protocols which use a purely periodic update strategy (such as FSR). To detect if a significant neighbour or intrascope topology change has occurred a topology metric can be used. In this case, two topology metrics are required to be kept, one for the neighbouring topology and one for the intrascope topology. The topology metric counts the number changes after the startup phase and triggers an update event if a certain number of changes occur. The *MTCU* algorithm is outlines below. Note that the algorithm only shows the maintenance phase of MTCU.

Algorithm *MTCU*

```
(∗ The MTCU algorithm ∗)
$NT_c \leftarrow$ Total current number of neighbours
$NT_p \leftarrow$ Total previous number of neighbours
$T_c \leftarrow$ Total number of destinations in the intrascope
$T_p \leftarrow$ Total intrascope destinations previously recorded
$N \leftarrow$ Total intrascope destinations previously recorded
$PN_{change} \leftarrow$ percentage of neighbour change required
$PT_{change} \leftarrow$ percentage of topology change required
$N_{change} \leftarrow$ neighbour changes recorded
$T_{change} \leftarrow$ Topology changes recorded
while (node is online)
    wait for an update
    if ($update = neighbour$)
        update neighbour table
        $NT_c \leftarrow$ total number of neighbours
        $N_{change} += $ number of changes
        if ($N_{change} \geq PN_{change} * NT_p$)
            Disseminate intrascope update
            $NT_p \leftarrow NT_c$
            $N_{change} \leftarrow 0$
    if ($update = Intrascope$)
        update topology table
        $T_c \leftarrow$ total number of neighbours
        $T_{change} += $ number of changes
        if ($N_{change} \geq PT_{change} * T_p$)
            Disseminate interscope update
            $T_p \leftarrow T_c$
            $T_{change} \leftarrow 0$
    if ($update = Interscope$)
        update topology table
```

In the above algorithm, the rate at which updates are sent also depends on the percentage of changes experienced (i.e. PT_{change} and PN_{change}). The percentage of change value can be a static parameter between 0 and 100% and preprogrammed into each device. However, it maybe beneficial to dynamically change its value according to the network conditions. One way to do this is by estimating the available bandwidth at each node and also for the intrascope, then varying the percentage change values according to the level of available bandwidth. Therefore, in times where the level of traffic (e.g. data and control) is low, more updates can be sent to increase the accuracy of the routes.

---

[3] That is, when the network is static then updates are sent at a lower frequency when compared to purely periodic updates

VII. CONCLUSIONS

This paper presents a new proactive route update strategy for mobile ad hoc networks. We present Minimum Displacement Update Routing (MDUR) and Hierarchical MDUR (HMDUR). In these strategies, the rate at which route updates are sent is proportional to the rate at which each node changes its location by a threshold distance. We implemented HMDUR and compared its performance with FSR. Our results indicate that HMDUR produces fewer routing overheads than FSR while maintaining high levels of data throughput across different network scenarios. Furthermore, the results show that when the node density is high, reducing routing overhead can result in higher levels of data packet delivery and lower end-to-end delay for each packet. In the future, we plan to simulate MDUR and HMDUR with a simple geographic data forwarding (such as those those described in [17]) and compare its performance with shortest path routing.

REFERENCES

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

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

[3] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. Scalable Routing Strategy for Dynamic Zone-based MANETs. In *Proceedings of IEEE GLOBECOM*, Taipei, Taiwan, November 17-21 2002.

[4] S. Basagni, I. Chlamtac, V.R. Syrotivk, and B.A. Woodward. A Distance Effect Algorithm for Mobility (DREAM). In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'98)*, Dallas, TX, 1998.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y-Ch Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[6] 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.

[7] J.J. Garcia-Luna-Aceves and Cruz Marcelo Spohn. Source-Tree Routing in Wireless Networks. In *Proceedings of the Seventh Annual International Conference on Network Protocols Toronto, Canada*, page 273, Oct 1999.

[8] M. Gerla. Fisheye State Routing Protocol (FSR) for Ad Hoc Networks. In *Internet Draft, draft-ietf-manet-aodv-03.txt*, work in progress, 2002.

[9] 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.

[10] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen. Scalable Routing Strategies for Ad Hoc Wireless Networks. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks*, pages 1369–79, Aug. 1999.

[11] 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.

[12] 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.

[13] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00)*, pages 120–130, August 2000.

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

[15] G. Pei, M. Gerla, X. Hong, and C. Chiang. A Wireless Hierarchical Routing Protocol With Group Mobility. In *Proc. Wireless Communications and Networking*, New Orleans, 1999.

[16] 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.

[17] I. Stojmenovic. Position based routing in ad hoc networks. *IEEE Communications Magazine*, pages 128–134, July 2002.

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

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