2005

A New Strategy to Improve Proactive Route Updates in Mobile Ad Hoc Networks

Mehran Abolhasan
mehran.abolhasan@uts.edu.au

Tadeusz A. Wysocki
University of Wollongong, wysocki@uow.edu.au

Justin Lipman
University of Wollongong, jlipman@uow.edu.au

Recommended Citation

Abolhasan, Mehran; Wysocki, Tadeusz A.; and Lipman, Justin: A New Strategy to Improve Proactive Route Updates in Mobile Ad Hoc Networks 2005, 828-837.
https://ro.uow.edu.au/infopapers/2792

A New Strategy to Improve Proactive Route Updates in Mobile Ad Hoc Networks

Abstract

This paper presents two new route update strategies for performing proactive route discovery in mobile ad hoc networks (MANETs). The first strategy is referred to as minimum displacement update routing (MDUR). 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. The second strategy is called minimum topology change update (MTCU). In this strategy, the route updating rate is proportional to the level of topology change each node experiences. We implemented MDUR and MTCU on top of the fisheye state routing (FSR) protocol and investigated their performance by simulation. The simulations were performed in a number of different scenarios, with varied network mobility, density, traffic, and boundary. Our results indicate that both MDUR and MTCU produce significantly lower levels of control overhead than FSR and achieve higher levels of throughput as the density and the level of traffic in the network are increased.

Disciplines

Physical Sciences and Mathematics

Publication Details

Abolhasan, M., Wysocki, T. A. & Lipman, J. (2005). A New Strategy to Improve Proactive Route Updates in Mobile Ad Hoc Networks. EURASIP Journal on Wireless Communications and Networking, 828-837.

EURASIP Journal on Wireless Communications and Networking 2005:5, 828–837

A New Strategy to Improve Proactive Route Updates in Mobile Ad Hoc Networks

Mehran Abolhasan
Telecommunications and Information Technology Research Institute, University of Wollongong, NSW 2522, Australia
Email: mehran@titr.uow.edu.au

Tadeusz Wysocki
Telecommunications and Information Technology Research Institute, University of Wollongong, NSW 2522, Australia
Email: wysocki@uow.edu.au

Justin Lipman
Telecommunications and Information Technology Research Institute, University of Wollongong, NSW 2522, Australia
Email: justin@titr.uow.edu.au

Received 12 January 2005; Revised 7 April 2005; Recommended for Publication by Phillip Regalia

This paper presents two new route update strategies for performing proactive route discovery in mobile ad hoc networks (MANETs). The first strategy is referred to as minimum displacement update routing (MDUR). 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. The second strategy is called minimum topology change update (MTCU). In this strategy, the route updating rate is proportional to the level of topology change each node experiences. We implemented MDUR and MTCU on top of the fisheye state routing (FSR) protocol and investigated their performance by simulation. The simulations were performed in a number of different scenarios, with varied network mobility, density, traffic, and boundary. Our results indicate that both MDUR and MTCU produce significantly lower levels of control overhead than FSR and achieve higher levels of throughput as the density and the level of traffic in the network are increased.

Keywords and phrases: MDUR, MTCU, proactive route updating, MANETs, routing, GPS-based route updating.

1. 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 enables 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 PDAs or laptops) makes routing in these networks a challenging task. A routing protocol 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 [1, 2], each data packet carries 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 of 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, so does the probability of route failure. To show this let $P(f) \alpha \sum_{i=1}^{n} a_i$, 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 $n \to \infty$, $P(f) \to 1$. Secondly, as the number of intermediate nodes in each route grows, the amount of overhead carried in each header of each data packet will grow as well. Therefore, in large networks with significant levels of multihopping and high levels of mobility, these protocols may not scale well.

In hop-by-hop routing (also known as point-to-point routing) [3, 4], 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 receive 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 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 [5, 6, 7, 8, 9, 10]. 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 proactive routing strategy (such as distance vector or link state) and maintain approximate routes or on-demand routes for faraway nodes.

In this paper, we propose two new route updating strategies to perform proactive route discovery in mobile ad hoc networks. These are minimum displacement update routing (MDUR) and minimum topology change update (MTCU). In MDUR, 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 MTCU, the rate at which updates are sent is proportional to the level of topology change experienced by each node. In [10], we briefly mentioned MDUR; in this paper we give a full description of this strategy and investigate its performance, along with MTCU, under different network scenarios using a simulation tool. The rest of this paper is organised as follows. In Section 2, we describe a number of different route update strategies proposed in the literature. Section 3 describes our route updating strategies. Section 4 describes the simulation environment, parameters, and performance metric used to investigate the performance of our route updating strategies. Section 5 presents the discussion of our simulation results and Section 6 presents the conclusions of the paper.

2. RELATED WORK

Proactive route discovery provides predetermined 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 [11, 12, 13], 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. Furthermore, we also describe a number of different updating strategies proposed for wireless cellular networks.

2.1. 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 significantly, especially as the number of nodes in the network is increased. Two such protocols are DSDV [14] and WRP [15].

2.2. Localised updates

To reduce the overheads in global updates, a number of localised updating strategies were introduced in protocols such as GSR [16] and FSR [12, 17]. In these strategies, route update propagation is limited to a localised region. For example, in GSR each node exchanges routing information with

---

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

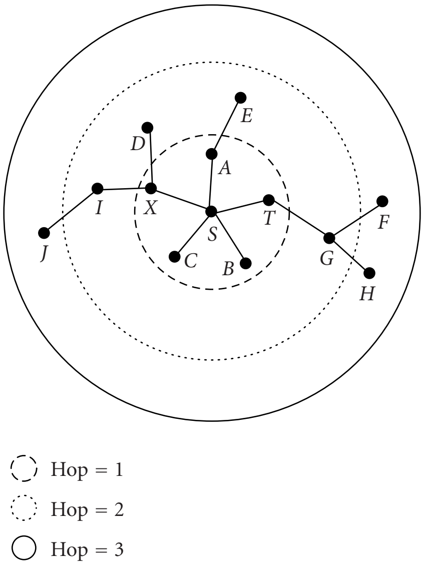


Figure 1: Illustration of the fisheye scope in FSR.

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 that of updating the nodes which are located faraway. 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 in Figure 1. The update messages which contain routing information to the nodes outside of the fisheye scope are disseminated to the neighbouring nodes 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. In OLSR, a two-hop neighbour knowledge is maintained proactively to determine a set of MPR (or multipoint relay) nodes. These nodes are used during the flooding of globally propagating route updates in order to minimise the number of rebroadcasting nodes (i.e., redundancy).

2.3. Mobility-based updates

Another strategy which can be used to reduce the number of update packets is introduced in DREAM [13]. The author proposes that routing overhead can be reduced by making the rate at which route updates are sent proportional 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 faraway.

2.4. 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 [14, 18]. In this strategy a node sends an update if certain different events occur at any time. Some events which can trigger an update are 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, these eliminating redundant periodic update dissemination into the network.

2.5. Updating strategies for cellular networks

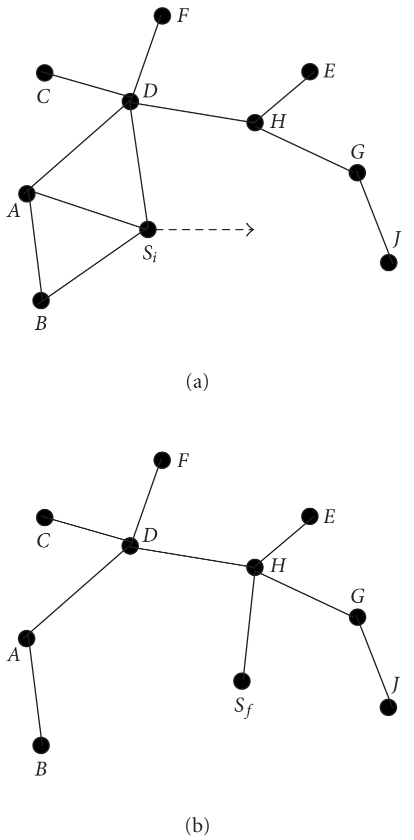
Previous sections described a number location and route updating strategies proposed for ad hoc networks. In cellular networks, a number of updating strategies have been proposed for cellular networks. These include movement-based updates, distance-based updates, and timer-based updates. In movement-based updates [19, 20], a location update is transmitted when the number of cell boundary crossings exceeds a predetermined value. In distance-based updates [21, 22], a location update is transmitted when a node's distance (in terms of number of cells) from the last updating time, exceeds a predetermined limit. In timer-based [23] each node transmits an update packet periodically (similar to the periodic updating used in ad hoc networks).

Further research is required for determining the usefulness of these strategies in mobile ad hoc networking models which use a static grid (similar to cells) or zone-based maps [5, 6]. Such work is beyond the scope of this paper.

3. PROPOSED STRATEGIES

In this section, we propose minimum displacement update routing (MDUR) and minimum topology change update (MTCU). This strategy attempt disseminates route update packets into the network when they are required rather than using purely periodic updates. In MDUR, 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). Note that the rate of displacement is different to speed, which is used in DREAM [13] routing protocol. This is because speed measurement does not take into account displacement but rather distance. In MTCU, the rate at which route update packets are sent is proportional to the level of topology change detected by each node, using its topology table. Note that this strategy does not require a GPS.

The following section describes the idea behind displacement-based updates and illustrates the advantage



(a)

(b)

Figure 2: Illustration of node migration in MDUR: (a) initial position for node S, (b) final position for node S.

of using displacement as a route update section criteria rather than speed (or distance). This is then followed by the description of MTCU.

3.1. Minimum displacement update routing

3.1.1. Overview and definition of MDUR

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.

```
(*The MDUR algorithm *)
$L_p$ ← previous location
$L_c$ ← current location
$L_p$ ← $L_c$
$D_T$ ← the threshold distance
Disseminate update packet
$V$ ← speed of node
$T_c$ ← current time
if ($V = 0$)
    $V \leftarrow V_{\max}$
$\tau \leftarrow \left(\frac{D_T}{V}\right) + T_c$
while (node is online)
    wait until $T_c = \tau$
    $L_c$ ← current location
    if dist $(L_c, L_p) \geq D_T$
        Disseminate update packet
        $L_p \leftarrow L_c$
        $\tau \leftarrow \left(\frac{D_T}{V}\right) + T_c$
    else
        $\tau \leftarrow \left(\frac{D_T - \text{dist}(L_c, L_p)}{V}\right) + T_c$
```

Algorithm 1: MDUR.

3.1.2. 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 outlined in Algorithm 1.

Displacement updates are more beneficial than using updates based purely on mobility (i.e., speed [13]). 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 $\text{dist}(L_c, L_p) < D_T$. 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, the topology of node S will not change. Therefore, an update is not required in this network. However, in the case a strategy is purely based on mobility such as in [13], an update may be disseminated and it may continue

Table 1: Fisheye state routing simulation parameters.

Number of scopes	2
Intrascope update interval	5 s
Interscope update interval	15 s
Neighbour timeout interval	15 s

Table 2: Hierarchical MDUR simulation parameters.

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

to send updates even if node S moves back and forward between these two points. On the contrary, in this scenario in MDUR no updates will be sent.

3.1.3. Implementation decisions for MDUR

To evaluate the performance and benefits of MDUR, it was implemented on top of FSR, which we refer to as hierarchical MDUR (HMDUR). Recall that FSR disseminates two types of update packets: intrascope update packets which propagate within the fisheye scope and interscope packets which propagate through the entire network. Therefore, 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}. Tables 1 and 2 illustrate the parameters used in FSR[2] and HMDUR.

The HMDUR algorithm is outlined in Algorithm 2.

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

Algorithm 2: HMDUR.

3.2. Minimum topology change updates

3.2.1. Description of MTCU

One way to increase the scalability of proactive routing protocols is by maintaining approximate routes to each destination rather than exact routes. In [12, 13], each node maintains approximate (or less accurate) information to faraway destinations, since the updates from faraway nodes are received less frequently. Similarly, in HMDUR, nodes maintain approximate routing information to nodes located faraway by using the interscope displacement metric.

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 startup 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 for failure notifications and the number of changes recorded. These changes can include discovery of a 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 counts 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 each node maintains its neighbour

---

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

```
(* 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
$PN_{\text{change}} \leftarrow$ percentage of neighbour change required
$PT_{\text{change}} \leftarrow$ percentage of topology change required
$N_{\text{change}} \leftarrow$ neighbour changes recorded
$T_{\text{change}} \leftarrow$ 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_{\text{change}} += $ number of changes
            if ($N_{\text{change}} \geq PN_{\text{change}} * NT_p$)
                Disseminate intrascope update
                $NT_p \leftarrow NT_c$
                $N_{\text{change}} \leftarrow 0$
        if (update = intrascope)
            update topology table
            $T_c \leftarrow$ total number of neighbours
            $T_{\text{change}} += $ number of changes
            if ($N_{\text{change}} \geq PT_{\text{change}} * T_p$)
                Disseminate interscope update
                $T_p \leftarrow T_c$
                $T_{\text{change}} \leftarrow 0$
        if (update = interscope)
            update topology table
```

Algorithm 3: HMDUR.

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] thus reducing the number of redundant updates if no changes occur. This also means that fewer periodic updates may be 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 of changes after the startup phase and triggers an update event if a certain number of changes occur. The MTCU algorithm is outlined in Algorithm 3. Note that the algorithm only shows the maintenance phase of MTCU.

In Algorithm 3, 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 may be 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.

Table 3: MTCU simulation parameters.

Number of scopes	2
Intrascope max timeout interval	10*S*
Interscope max timeout interval	30*S*
Neighbour change threshold	10%
Intrascope change threshold	30%

3.2.2. *Implementation decisions for MTCU*

Similar to MDUR, MTCU was also implemented on the top of FSR. Table 3 illustrates the simulation parameters of MTCU. Note that the neighbour change threshold and the intrascope thresholds represent the required level of topology change in the neighbouring and intrascope topology, respectively, before an intrascope or an interscope update is disseminated.

4. SIMULATION MODEL

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

4.1. *Simulation environment and scenarios*

The GloMoSim simulation tool was used to carry out our simulations [24]. 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 1000 m × 1000 m boundary. IEEE 802.11 DSSS (direct sequence spread spectrum) was used with maximum transmission power of 15 dBm at 2 Mb/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 were used for the propagation model. The antenna height is set to 1.5 m, the radio receiver threshold is set to −81 dBm, and the receiver sensitivity was set to −91 dBm according to the Lucent wavelan card [25]. A random way-point mobility model was used with the node mobility ranging from 0 to 20 m/s and pause time varied from 0 to 900 s. The simulation was run for 900 s for 10 different values of pause time and each simulation was

---

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

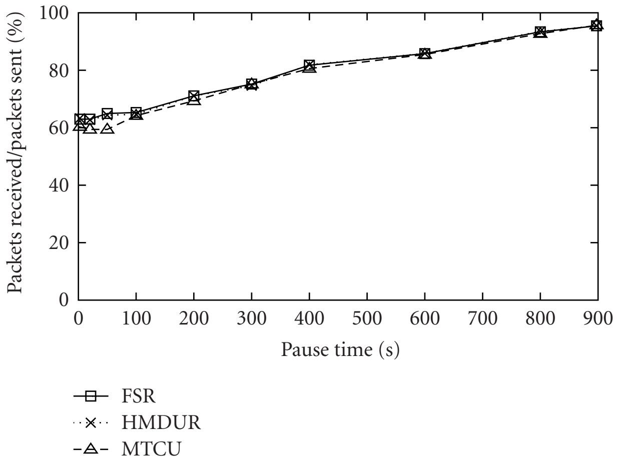


Figure 3: PDR for 10*S* and 50*N*.

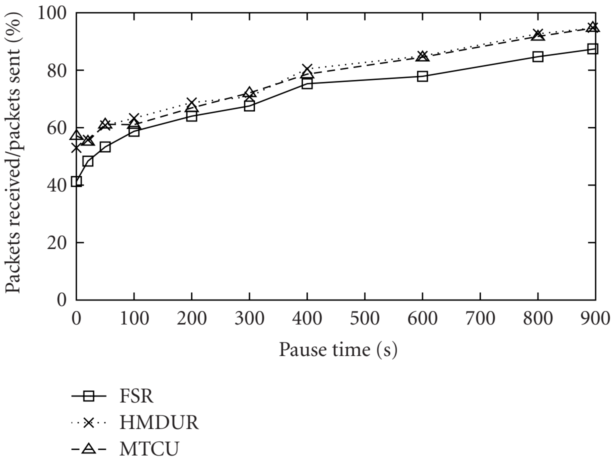


Figure 4: PDR for 10*S* and 100*N*.

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.

4.2. *Performance metrics*

To investigate the performance of the routing protocols, the following performance metrics were used.

(i) 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.
(ii) Normalised routing overhead (O/H): the amount of routing overhead transmitted through the network for each data packet successfully delivered to the destination.
(iii) 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 by 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.

5. SIMULATION RESULTS

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

5.1. *Packet delivery ratio*

The graphs in Figures 3 and 4 illustrate the PDR results obtained for the 1000 m × 1000 m boundary. In the 50-node scenario, all routing strategies show similar levels of PDR.

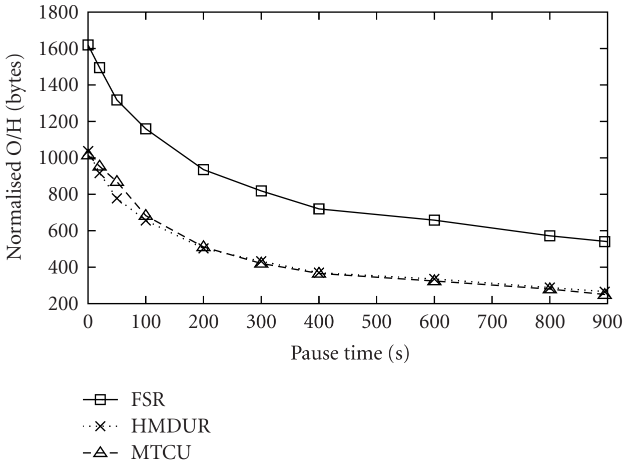


Figure 5: Normalised O/H for 10*S* and 50*N*.

However, in the 100-node network scenario, HMDUR and MTCU start to outperform FSR. This is because HMDUR and MTCU still maintain 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 our proposed strategies. Furthermore, more updates may increase channel contention, which can result in more packets being dropped at each intermediate node.

5.2. *Normalised control overhead*

The graphs in Figures 5 and 6 illustrate the normalised routing overhead experienced in the 1000 m × 1000 m 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

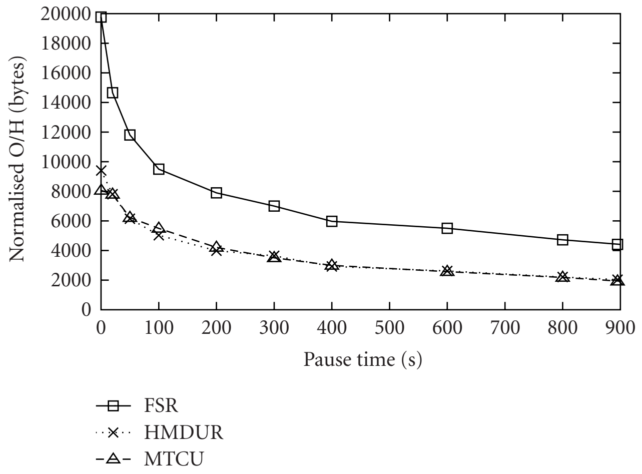


Figure 6: Normalised O/H for 10S and 100N.

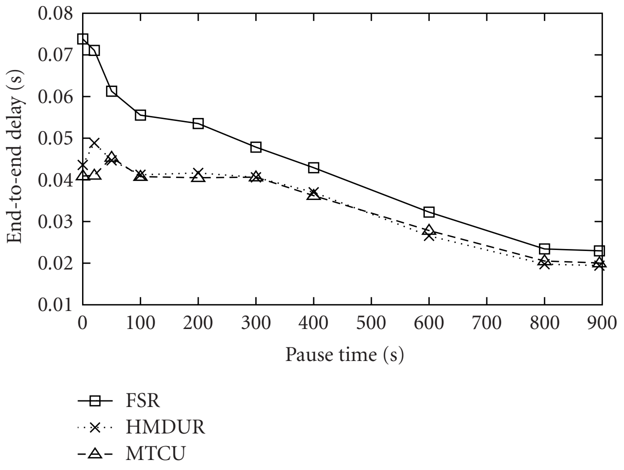


Figure 7: Delays O/H for 10S and 50N.

periodic updates were used in HMDUR and MTCU, 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 experience topology changes, and when mobility is low, less updates are sent. From the results shown in Figures 5 and 6, it can be seen that both HMDUR and MTCU produce less overhead than FSR, across all different levels of pause time and node density.

5.3. *Delays*

The graphs in Figures 7 and 8 illustrate the end-to-end delay experienced in the 1000 m × 1000 m boundary. These results show that in HMDUR and MTCU each data packet experiences lower end-to-end delay than in FSR. The lower delay experienced is due to the higher level of accessibility to the wireless medium. This is because in our proposed strategies each node generates less route updates than in 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.

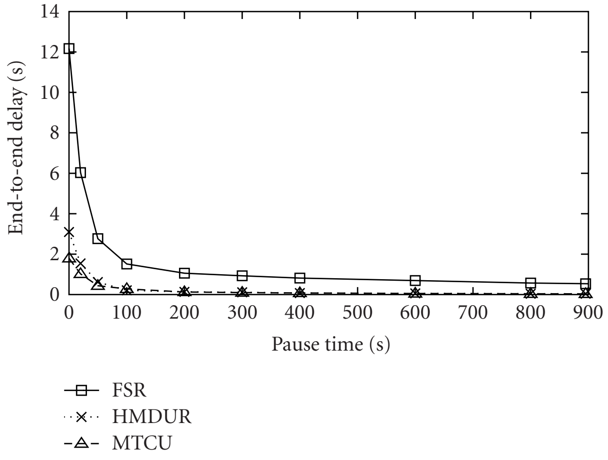


Figure 8: Delays O/H for 10S and 100N.

6. CONCLUSIONS

This paper presents new proactive route update strategies 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. Furthermore, we introduced minimum topology change update (MTCU). In this strategy, update packets are sent only when a minimum topology change is experienced by each node. We implemented HMDUR and MTCU in GloMoSim and compared their performance with FSR. Our results indicate that both HMDUR and MTCU produce 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 described in [26]) and compare its performance with shortest path routing.

REFERENCES

[1] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," in *Internet Draft*, work in progress, 2002, http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-dsr-07.txt.

[2] C.-K. Toh, "A novel distributed routing protocol to support Ad Hoc mobile computing," in *Proc. 15th IEEE Annual International Phoenix Conference on Computers and Communications (IPCCC '96)*, pp. 480–486, Scottsdale, Ariz, USA, March 1996.

[3] S. Das, C. Perkins, and E. Royer, "Ad Hoc On-Demand Distance Vector (AODV) Routing," in *Internet Draft*, work in progress, 2002, http://tools.ietf.org/wg/manet/draft-ietf-manet-aodv/draft-ietf-manet-aodv-11.txt.

[4] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "LPAR: an adaptive routing strategy for MANETs," *Journal of Telecommunications and Information Technology*, vol. 2, pp. 28–37, 2003.

[5] J.-N. Mario and I.-T. Lu, "A peer-to-peer zone-based two-level link state routing for mobile Ad Hoc networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1415–1425, 1999.

[6] S. C. M. Woo and S. Singh, "Scalable routing protocol for Ad Hoc networks," *Wireless Networks*, vol. 7, no. 5, pp. 513–529, 2001.

[7] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic Ad Hoc routing," in *Proc. 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 120–130, Boston, Mass, USA, August 2000.

[8] Z. J. Haas, R. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," in *Internet Draft*, work in progress, 1999, http://www.ietf.org/proceedings/99jul/I-D/draft-ietf-manet-zone-zrp-02.txt.

[9] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC '99)*, vol. 3, pp. 1538–1542, New Orleans, La, USA, Septemper 1999.

[10] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, "Scalable routing strategy for dynamic zones-based MANETs," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 173–177, Taipei, Taiwan, November 2002.

[11] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless Ad Hoc network routing protocols," in *Proce. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 85–97, Dallas, Tex, USA, October 1998.

[12] M. Gerla, "Fisheye State Routing Protocol (FSR) for Ad Hoc Networks," in *Internet Draft*, work in progress, 2002, http://www.ietf.org/proceedings/03mar/I-D/draft-ietf-manet-fsr-03.txt.

[13] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proce. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pp. 76–84, Dallas, Tex, USA, October 1998.

[14] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. Conference on Communications Architectures, Protocols and Applications (ACM SIGCOMM '94)*, pp. 234–244, London, UK, August–September 1994.

[15] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proc. 1st Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '95)*, pp. 86–94, Berkeley, Calif, USA, November 1995.

[16] T.-W. Chen and M. Gerla, "Global state routing: a new routing scheme for Ad Hoc wireless networks," in *Proc. IEEE International Conference on Communications (ICC '98)*, vol. 1, pp. 171–175, Atlanta, Ga, USA, June 1998.

[17] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for Ad Hoc networks," in *Proc. IEEE International Multi Topic Conference (INMIC '01)*, pp. 62–68, Lahore, Pakistan, December 2001.

[18] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *Proc. 7th IEEE International Conference on Network Protocols (ICNP '99)*, pp. 273–282, Toronto, Ontario, Canada, October–November 1999.

[19] V. W.-S. Wong and V. C. M. Leung, "Location management for next-generation personal communications networks," *IEEE Network*, vol. 14, no. 5, pp. 18–24, 2000.

[20] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: to update or not to update?" *Wireless Networks*, vol. 1, no. 2, pp. 175–185, 1995.

[21] U. Madhow, M. L. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking," *IEEE/ACM Trans. Networking*, vol. 3, no. 6, pp. 698–707, 1995.

[22] V. W.-S. Wong and V. C. M. Leung, "An adaptive distance-based location update algorithm for next-generation PCS networks," *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 1942–1952, 2001.

[23] C. Rose, "Minimizing the average cost of paging and registration: a timer-based method," *Wireless Networks*, vol. 2, no. 2, pp. 109–116, 1996.

[24] UCLA Parallel Computing Laboratory Wireless Adaptive Mobility Laboratory, "GloMoSim—scalable simulation environment for wireless and wired network systems," 2003, in http://pcl.cs.ucla.edu/projects/glomosim/.

[25] Lucent Technologies, ORINOCO PC card, 2003, in http://www.lucent.com/orinoco.

[26] I. Stojmenovic, "Position-based routing in Ad Hoc networks," *IEEE Commun. Mag.*, vol. 40, no. 7, pp. 128–134, 2002.

Mehran Abolhasan received the B.E. degree in computer engineering with honours from the University of Wollongong, in December 1999. He completed his Ph.D. degree in School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, in June 2003. During the course of his Ph.D., he has authored a number of different journal and conference papers. He has also been a technical referee for various conferences and journals and is currently a Member of IEEE. In 2003, he joined CRC-SIT, where he worked closely with a number of government organisations in proposing new innovative strategies and projects to improve the telecommunications infrastructure of emergency services in New South Wales. In 2004, he joined the Telecommunications and Information Technology Research Institute (TITR) in University of Wollongong and the Desert Knowledge CRC, where he is currently leading a project focusing on developing new telecommunications services for remote desert communities. His research interests are ad hoc, mesh, and sensor networking.

Tadeusz Wysocki received the M.S. Eng. degree with the highest distinction in telecommunications from the Academy of Technology and Agriculture, Bydgoszcz, Poland, in 1981. In 1984, he received his Ph.D. degree, and in 1990, he was awarded a D.S. degree (habilitation) in telecommunications from the Warsaw University of Technology. In 1992, he moved to Perth, Western Australia, to work at Edith Cowan University. He spent the whole 1993 at the University of Hagen, Germany, within the framework of Alexander von Humboldt Research

Fellowship. After returning to Australia, he was appointed as a Wireless Systems Program Leader within Cooperative Research Centre for Broadband Telecommunications and Networking. Since December 1998, he has been working as an Associate Professor at the University of Wollongong, New South Wales, within the School of Electrical, Computer and Telecommunications Engineering. The main areas of his research interests include indoor propagation of microwaves, code division multiple access (CDMA), space-time coding, and MIMO systems, as well as mobile data protocols including those for ad hoc networks. He is the author or coauthor of four books, over 150 research publications, and nine patents. He is a Senior Member of IEEE.

Justin Lipman received a B.E. degree in computer engineering and a Ph.D. degree in telecommunications engineering from the University of Wollongong in 1999 and 2004, respectively. He is currently the Project Manager for Research and Innovation at Alcatel Shanghai Bell telecommunications labs in Shanghai, China. His research interests are diverse but focus mainly on mesh, ad hoc, sensor, and 4G networks.