University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Computer & Electronics Engineering Faculty Publications

Computer & Electronics Engineering, Department of

1-1-2003

A Performance Study of Dynamic Zone Topology Routing Protocol

Mehran Abolhasan
University of Wollongong, mehran@titr.uow.edu.au

Tadeusz A. Wysocki
University of Nebraska-Lincoln, wysocki@uow.edu.au

Follow this and additional works at: http://digitalcommons.unl.edu/computerelectronicfacpub

Part of the Computer Engineering Commons

Abolhasan, Mehran and Wysocki, Tadeusz A., "A Performance Study of Dynamic Zone Topology Routing Protocol" (2003). *Computer & Electronics Engineering Faculty Publications.* Paper 54.
http://digitalcommons.unl.edu/computerelectronicfacpub/54

This Article is brought to you for free and open access by the Computer & Electronics Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Computer & Electronics Engineering Faculty Publications by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

A Performance study of Dynamic Zone Topology Routing Protocol

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*— In this paper we present a simulation study of a hybrid routing protocol we proposed in our previous work [3][4]. Our hybrid routing strategy is called Dynamic Zone Topology Routing protocol (DZTR). This protocol has been designed to provide scalable routing in a Mobile Ad hoc Networking (MANET) environment. DZTR breaks the network into a number of zones by using a GPS. The topology of each zone is maintained proactively and the route to the nodes in other zones are determined reactively. DZTR proposes a number of different strategies to reduce routing overhead in large networks and reduce the single point of failure during data forwarding. In this paper, we propose a number of improvements for DZTR and investigate its performance using simulations. We compare the performance of DZTR against AODV, LAR1 and LPAR. Our results show that DZTR has fewer routing overheads than the other simulated routing protocols and achieves higher levels of scalability as the size and the density of the network is increased.**

I. Introduction

Mobile Ad hoc Networks (MANETs) are comprised of end user nodes, which are capable of performing routing in a distributed fashion. This means that these networks do not require a central coordinator or a base station to perform and establish routes. These networks are particularly useful in areas where an infrastructure is not available or difficult to implement. Such areas include the highly dynamic battle field environment, which requires a mobile networking solution and in the search-and-rescue operations where a large rescue team may be searching through a remote area such as a jungle or a desert.

Similar to most infrustructured or wired networks such as the Internet, MANETs employ a TCP/IP networking model. However, the need to provide end-to-end communication in a dynamic environment, along with the limited resources such as bandwidth and power, demands a redefinition of the layers used in the TCP/IP. Currently, research is being carried out across all layers of the TCP/IP model, to design an infrastructure, which will provide reliable and efficient end-to-end communication for MANETs. One challenging, yet highly researched area in MANETs is routing. In MANET, an intelligent routing strategy is required to provide reliable end-to-end data transfer between mobile nodes while ensuring that each user receives certain level of QoS. Furthermore, the routing strategy must minimise the amount of bandwidth, power and storage space used at each end user node. Therefore, traditional routing strategies, such as the link-state and distance vector algorithm, which where intended for wired or infrastructured networks will not work well in dynamic networking environment.

To overcome the problems associated with the link-state and distance-vector algorithms a number of routing protocols have been proposed for MANETs. These protocols can be classified into three different groups: Global/Proactive, On-demand/Reactive and Hybrid. In proactive routing protocols such as FSR[8], DSDV[13] and DREAM[6], each node maintains routing information to every other node (or nodes located in a specific part) in the network. The routing information is usually kept in a number of different tables. These tables are periodically updated and/or if the network topology changes. The difference between these protocols exists in the way the routing information is updated, detected and the type of information kept at each routing table. Furthermore, each routing protocol may maintain different number of tables. On-demand routing protocols such as AODV[7], DSR[11] and LAR[12] were designed to reduce the overheads in proactive protocols by maintaining information for active routes only. This means that routes are determined and maintained for nodes that require to send data to a particular destination. Route discovery usually occurs by flooding a route request packets through the network. When a node with a route to the destination (or the destination itself) is reached, a route reply is sent back to the source node using link reversal if the route request has travelled through bi-directional links, or by piggy-backing the route in a route reply packet via flooding. Hybrid routing protocols such as ZHLS[10], ZRP[9] and SLURP[14] are a new generation of protocol, which are both proactive and reactive in nature. These protocols are designed to increase scalability by allowing nodes with close proximity to work together to form some sort of a backbone to reduce the route discovery overheads. This is mostly achieved by proactively maintaining routes to nearby nodes and determining routes to far away nodes using a route discovery strategy. Most hybrid protocols proposed to date are zone-based, which means that the network is partitioned or seen as a number of zones by each node. Others group nodes into trees or clusters. Hybrid routing protocols have the potential to provide higher scalability than pure reactive or proactive protocols. This is because they attempt to minimise the number of rebroadcasting nodes by defining a structure (or some sort of a backbone), which allows the nodes to work together in order to organise how routing is to be performed. By working together the best or the most suitable nodes can be used to perform route discovery. For example, in ZHLS only the nodes which lead to the gateway nodes rebroadcast the route discovery packets. Collaboration between nodes can also help in maintaining routing information much longer. For example, in SLURP, the nodes within each region (or zone) work together to maintain location information about the nodes, which are assigned to that region (i.e. their home region). This may potentially eliminate the need for flooding, since the nodes know exactly where to look for a destination every time. Another novelty of hybrid routing protocols is that they attempt to eliminate single point of failure and creating bottleneck nodes in the network. This is achieved by allowing any number of nodes to perform routing or data forwarding if the preferred path becomes unavailable.

Most hybrid routing protocols proposed to date are zone-based. In zone-based routing protocols, the network is divided into a number zones, which can be overlapping ones, such as in ZRP, or non-overlapping such as in ZHLS. The disadvantage of ZRP is that if the zone radius is too large the protocol can behave like a pure proactive protocol, while for a small zone radius it behaves like a reactive protocol. Furthermore, the zones are overlapping, which means that each node can belong to a number of different zones, which increases redundancy. The disadvantage of a non-overlapping zone-based protocols such as ZHLS is that the zone partitioning is done at the design stage. This means that all nodes must have preprogrammed zone maps, which are identical for all nodes in the network, or they must obtain a copy of the zone map before routing can occur. Static zone maps can be used in environments where the geographical boundaries of the network are known (or can be approximated). Such environments include: shopping malls, universities or large office buildings, where physical boundaries can be determined and partitioned into a number of zones. However, in environments where the geographical boundaries of the network are dynamic (i.e. can change from time to time as nodes may travel to different regions), a static zone map cannot be implemented. Examples of such networks include: the battlefield where the battle scene may constantly move from one region to another or in search-and-rescue operations in remote areas. In these environments, a dynamic zone topology is required.

In our previous study [4], we proposed DZTR, where we introduced two dynamic zone creation algorithms, which use a number different location tracking strategies to determine routes with the least amount of overheads. In this paper, we propose a number of improvements for DZTR and investigate its performance using simulation technique. We also compare the performance of DZTR with AODV, LAR1, and LPAR[5], under a number of different network scenarios and comment on their scalability in large networks. The rest of this paper is organised as follows. Section II briefly describes the DZTR routing protocols. Section III describes the simulation tool and the parameters used in our simulations. Section IV presents a discussion on the results we obtained from the preliminary simulations. and section V presents the concluding remarks for the paper.

II. Dynamic Zone Topology Routing

DZTR is a zone based routing protocol is designed to provide scalable routing in large networks with high levels of traffic. The advantage of DZTR over some of the other zone-based routing protocols described in the previous section includes:

- Zones are created dynamically rather than using a static zone map such as in ZHLS. This means that a preprogrammed zone map is not required.
- Each zone only belongs to one zone, which means that information redundancy is reduced, while a more collaborative environment is defined.
- Single-point of failure is reduced, since there is no clusterhead or a root-node. All nodes within each zone work together to determine the best routes with the least amount of overheads, and data forwarding between each zone can still occur without a route failure as long as there is one gateway connecting the two zones.
- A number of location tracking strategies is proposed to determine routes with minimum amount of overheads for a number of different scenarios.

The DZTR routing protocol is made up of three parts. These are Zone Creation, Topology Determination and Location Discovery. The following sections describe each part.

A. Zone Creation

In DZTR two different zone creation algorithms are proposed. These are referred to as DZTR1 and DZTR2. In DZTR1, all nodes in the network start off by being in single state mode, which means that they are not members of any zone. When two nodes come within each others transmission range and form a bi-directional link, a zone is created if the following conditions are satisfied:

- Neither node has a zone ID which maps within their transmission range.
- At least one of the nodes are not a gateway node of another zone[1].

To create a zone ID, each node records its current location, speed and battery power and exchange it with the other using a Zone-Query packet. The coordinates of the node with the lower speed will be used as the zone centre point, which is used to create and reserve the zone boundary. If the nodes have the same velocity, then the node with the higher battery power will be used as the centre point. The aim here is to select the node which is expected to last the longest in the calculated zone. This means that the calculated zone will be active for a longer time.

When the node which has the higher stability of the two is determined, each node will then calculate the boundary using the centre point and the transmission range of that node. Note that when a node sends a Zone-Query Packet, it also keeps a copy of this packet and waits for the other node to send its Zone-Query Packet. When the neighbours Zone-Query packet is received, it uses the two packet to create the zone. The node will then exchange the calculated zone ID to ensure that they have agreed on the same zone ID. If the zone IDs are different the zone ID of the least mobile node is used based on the mobility information exchanged during the zone ID exchange phase. The zone ID will be a function of the centre point and the zone radius. We have chosen the zone ID to be the concatenation of the zone centre point and the zone radius. [2]

$$Z_{ID} = f(C, R) = C|R \tag{1}$$

Similar to DZTR1, the zones are geographically bounded by a zone radius. However, in DZTR2, the boundary of the zone is chosen in such a way that all nodes are within transmission range [4]. The advantage of this strategy is that there is no partitioning in each zone. Therefore, there is all nodes within each zone are aware of each other. Another advantage is that each

[1] One of the two nodes have a neighbouring node which is a zone member

[2] C = coordinates of the centre node (x,y,z), R = transmission range and the | means concatenation. Note that if we assume that R for all nodes are equal, then $ZID = C$

node can update its intrazone with just one beacon message, as there is no need for further rebroadcast to reach all different parts of each zone. However, the zones created in DZTR2 are smaller than DZTR1, which means that the number of zones in DZTR2 maybe significantly higher than DZTR1. This can increase the number of interzone migration when mobility is high, which will require each node to become affiliated with different zones more frequently. Hence, processing overhead and intrazone update may be higher than in DZTR1.

B. Topology Determination

In DZTR[3], once each nodes determines its zone ID, it will start to build its intrazone and interzone routing tables. The intrazone topology of each dynamic zone is maintained proactively and the topology and/or routes to the nodes in the interzone is determined reactively.

1) Intrazone routing: The intrazone network topology is maintained proactively. Each node in the network periodically broadcasts its location information to the other nodes in its intrazone. However, we minimise the number of control packets propagated through the intrazone by setting the frequency at which each node broadcasts its location to be proportional to its mobility and displacement. That is, each node broadcasts its location information through its intrazone if it has travelled (displaced) a minimum distance. This distance is called Minimum Intrazone Displacement (MID). To determine their displacement, each node starts by recording its current location at the startup using a GPS device. It will then periodically check its location (if the node is mobile), and compare it with the previously recorded location. If the distance between the current and the previous location is greater than or equal to MID, then the node will broadcast its location information through the intrazone and set its current location as the new previous location.

We call this updating strategy, Minimum Displacement Update (MDU). The advantage of this updating strategy is that updates are sent more frequently if the location of a node has changed significantly. The disadvantage of sending updates based on mobility alone is that if a node travels back and forward in a small region update packets are still disseminated, however, the topology may have not necessarily changed. Therefore, sending an update packet will be wasteful.

Intrazone update packets will also be sent if any of the following conditions occur:

1) New node comes online
2) Node enters a new zone
3) Node travels more than MID within a zone
4) Intrazone-Update Timer (IUT) expires

2) Interzone topology creation: The nodes that are situated near the boundary of each zone can overhear update or data packets travelling through the nodes in their neighoubouring zones. These nodes may also be in transmission range of other nodes which are members of another zone. These nodes are referred to as gateway nodes. When a gateway node learns about an existence of another zone, it will broadcast the zone ID of the new zone through its intrazone. This packet is called an Interzone-Update packet (IEZ). This packet includes the gateway nodes node ID, zone ID, location, velocity and learnt zone ID. Therefore, since the gateway includes its velocity and location information, other member nodes can update the information stored in their intrazone table about that gateway node. Hence, the gateways can reset their IUT timer each time they send one of these packets

3) Interzone migration: When nodes migrate from one zone to another they send a control packet to the previously visited zone, thus leaving behind a trail. The trail information includes the node's current zone ID, location and velocity. The nodes which receive this trail information update their routing tables. Therefore, the nodes in previously visited zone can forward the location request or data packets for the migrating zone to its current zone.

C. Location Discovery

When a node has data to send to a particular destination, if the location of the destination is known, DZTR will attempt a number of different location tracking strategies to determine a fresh route to the destination. The location tracking strategy chosen for a known destination will depend on its physical location, velocity and the time of the last previous communication. If the location of the destination is not know, DZTR will initiate Limited Zone-hop Search with Multizone Forwarding (LZS-MF) to determine a route while minimising overhead. To initiate different location tracking strategies, DZTR introduces four different routing scenarios:

(i) Destination is in the intrazone or is a temporary member.
(ii) Destinations ZID or location is known, and it is expected to be in its current zone.
(iii) Destinations ZID or location is known, but its velocity and location information suggest that it could currently lie a number of different neighbouring zones.
(iv) The location or the ZID of the destination is unknown.

When a source has data to send to a particular destination it firstly starts by checking if the destination is located in the intrazone or it is a temporary member. If the destination is found in one of these tables (i.e. case (i)), the source can start sending data since the route to the destination has been predetermined proactively.

If the destination is not found in the intrazone, then the source will consult its Destination History Table (DHT, described further in [4] . If an entry is found in the DHT, the source will check if the destination still maps in its current zone (using the destinations location, velocity and expiration time in the DHT), if the mapping suggests that the destination is still in its current zone (i.e. case (ii)), the source node will use its interzone table to forward the data packet towards the next zone, which leads to the destination zone.

In (iii), the destination's velocity indicates that it may not be in its recorded zone. In this case the destination node can lie in any number of zones. To find the current zone ID (or location) of the destination, the source node unicasts a Zone Request packet with destination's previously recorded location information (i.e. ZREQ-L), to the zone in which the destination was last suspected to be in, using its interzone topology table. When the

[3] When we say DZTR, we refer to both DZTR1 and DZTR2

ZREQ-L packet reaches the destination's suspected zone, the gateway node which have received this packet will first check to see if the destination is still in the intrazone (or a temporary member). If the destination was not found and no location trail is available, the gateway node will calculate a region in which the destination could have migrated to. We call this the Destinations Expected Region (DER)[4], and it is calculated using the destination's previously known velocity and location information. When the DER is calculated, the gateway node will create a new packet, which includes the source node ID and zone ID, destination ID, a sequence number and the DER. This packet is called a Localised Zone Request (LZREQ). The gateway node forwards this packet to all the neighbouring zones which map into the DER. Each gateway node in the receiving zones will check their tables for the destination, if the destination is not found, they will forward this packet to their outgoing (neighbouring) zones which map into the DER. Note that each node only forward the same LZREQ (or ZREQ) packet once. However, each zone may be queried more than once from different entry points (i.e. gateways). This way if there is clustering within each zone, the zones can still be effectively searched. If the destination is found, the destination will send a ZREP packet back towards the source.

In (iv), the destination's current zone is not known. In this case to search the network effectively while ensuring that overheads are kept low, we introduce a new zone searching strategy called Limited Zone-hop Search with Multizone Forwarding (LZS-MF). In this strategy the source node generates a ZREQ-N packet (N denotes no location information is available for the destination). This packet includes the source node ID, zone ID, location, sequence number, neighbouring zone list and a Zone-Hop (ZH) number. The zone hop number defines the number of zones which the ZREQ-N packet can visit before it expires. To search for an unknown destination, the source node begins by setting $ZH = 1$, which means that only the neighbouring zones can be searched. Each time the ZREQ-N discovery produces no results, the source node increments the value of ZH to increase the search area, and the search is initiated again. This search strategy continues until ZH = MAX-COVERAGE-AREA. The advantage of our limited zone-hop search is that if one of the nearby zones has a trail to the destination (or hosts the destination), we avoid searching all the zones in the network. Now, to ensure that not all nodes within each zone are involved in the routing, each time a gateway node in each zone receives a ZREQ-N packet, it uses its interzone topology table to forward the ZREQ-N packet to the nodes, which lead to the neighbouring zones. We call this Multizone Forwarding (MF). In this strategy the source node starts by consulting its interzone topology table to determine the list of neighbouring zones. It will then store the list of neighbouring zones, along with the neighbouring nodes which lead to one of these neighbouring zones. These nodes are the only nodes, which can forward the ZREQ-N packet towards the next neighbour leading to a neighbouring zone. When a ZREQ-N packet reaches a new zone, the receiving node (i.e. the gateway), will first check its routing tables to see if it has a location information about the destination. If no location destination is found and it has not seen the packet before, it will consult its interzone table and forward the ZREQ-N packet with a new list of neighbouring zones and forwarding nodes. The process continues until the ZH limit is reached, the packet timer expires or the destination is found. When the destination is found, it will send a ZREP packet back towards the source node, indicating its current zone, location and velocity. In DZTR, a link failure may not necessarily lead to route failure. This is because data packets can still be forwarded to their destination if there exists a node which leads towards the destination. A route failure will occur and returned back to the source if no such node can be found.

[4]Note the size of the DER is calculated in a similar manner to[12]

III. Simulation Model

In this section we describe the scenarios and parameters used in our simulation. We also describes the performance metrics used to compare our routing strategy with a number of existing routing strategies.

A. Simulation Environment and Scenarios

Our simulations were carried out using the GloMoSim[1] simulation package. GloMoSim is an event driven simulation tool designed to carry out large simulations for mobile ad hoc networks. The simulations were performed for 50, 100, 200, 300, 400 and 500 node networks, migrating in a 1000m x 1000m area. IEEE 802.11 DSSS (Direct Sequence Spread Spectrum) was used with maximum transmission power of 15dbm at a 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 considered as the propagation model. The antenna hight was set to 1.5m, the radio receiver threshold was set to -81 dbm and the receiver sensitivity was set to -91 dbm according to the Lucent wavelan card[2]. Random way-point mobility model was used with the node mobility ranging from 0 to 20m/s and pause time was set to 0 seconds for continuous mobility. The simulation was ran for 200s[5] 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 contained 64 Bytes and each packet were at 0.25s intervals. The simulation was run for 20 and 50 different client/server pairs[6] and each session begin at different times and was set to last for the duration of the simulation.

B. Implementation Decisions

The aim of our simulation study was to compare the route discovery performance of DZTR under different levels of traffic and node density with a number of different routing protocols. In our simulations, we compare DZTR with LPAR[7], AODV and LAR1. We implemented DZTR on the top of AODV using AODV's existing error recovery strategy, sequence numbering

[5]We kept the simulation time lower than the previous chapter due to a very high execution time required for the 50 Flow scenario

[6]Note that the terms Client/Server, src/dest and Flows are used interchangeably

[7]With stable routing enabled[5]

TABLE I
DZTR SIMULATION PARAMETERS

IUT	**30s**
Location Check Timer	**3s**
Zone Query Intervals	**30s**
MID	**150m**
Maximum Zone Hops	**7**

and broadcast ID strategies. The DZTR2 cluster strategy was implemented as the zone creation strategy in order to eliminate partitioning within each zone and also to allow topology maintenance messages (such as Intrazone, Interzone, Trail updates) to occur by using beaconing messages only. Therefore, each packet is exchanged between neighbouring nodes. For example, when a node sends a trail update packet, this packet is also used by its current intrazone members to update their intrazone table (i.e. it is seen as an intrazone update). Similarly, the nodes in the neighbouring zones update their interzone table and the closest gateway to the node which sent the trail update then broadcasts this trail update in its intrazone.

To reduce the number of intrazone updates in DZTR2, each time a node initiates a ZREQ-N, it also uses this packet to update its intrazone and resets its IUT. Furthermore, to minimise the number of interzone updates propagating through each zone, only the closest known gateway rebroadcasts a learnt zone ID. Similarly, during the zone creation phase, a zone reply is only sent by the node which is closest to the zone which sent a zone query. To minimise the routing overhead when location information is not available at the source, we modified the LZS-MF strategy so that during the first cycle of route discovery (i.e. first attempt at route discovery), each retransmitting node only select one node to represent each known zone in the interzone table during further rebroadcasts and each packet cannot re-enter the same zone. Furthermore, the chosen nodes must be further away from the source than the current hop. For example, if there are 6 neighbouring zones, then each retransmitting node will choose at most 6 other retransmitting nodes to further rebroadcast the control packets away from the source. If the first cycle fails, then in the second cycle, all nodes in the interzone table are chosen, which are further away from the source than the current hop. Finally, in the third cycle, all nodes in the interzone table are chosen regardless of their position.

Table I illustrates the simulation parameters used for DZTR.

C. Performance Metrics

The performance of each routing protocol is compared using the following performance metrics.

- Packet Delivery Ratio (vs) Number of nodes
- Normalised control overhead (O/H) (vs) Number of nodes
- End-to-End Delay (vs) Number of nodes

PDR is the Ratio of the number of packet sent by the source node to the number of packets received by the destination node. Normalised control overhead (O/H) presents the ratio of the number of routing packets transmitted through the network to the number of data packets received at the destination. for the duration of the simulation. This metric will illustrate the levels of the introduced routing overhead in the network. Therefore, Packet Delivery Ratio (vs) Number of nodes represents the percentage of data packets that were successfully delivered as the number of nodes was increased for a chosen value of pause time, and Normalised control overhead (O/H) (vs) Number of nodes shows how many control packet were introduced into the network to successfully transmit each data packet to its destination as the number of nodes is increased for a chosen value of pause time. The last metric is used to investigate the changes in end-to-end delay as the number of nodes is increased. Using these metrics, the level of scalability can be determined by the level of PDR or normalised overhead experienced and the shape of the curves. For example, the protocol which have the highest level of PDR and also maintains the flattest curve, has the highest scalability. For normalised overhead we look for the protocol which has the lowest amount of overhead throughout all different node densities. The last metric is used to investigate the changes in end-to-end delay as the number of nodes is increased.

IV. RESULTS

In this section we present the worst case (i.e. zero pause time and constant mobility) scenario results we obtained from our simulation. The results for other levels of mobility can be seen in Appendix A. To investigate the worst case scenario behaviour of each routing protocol, we recorded the PDR, normalised routing overhead and the end-to-end delay introduced into the network. We recorded this behaviour for up to 500 nodes in the network.

A. Packet Delivery Ratio Results

Figure 1 and 2 illustrate the PDR for the 20 Flows and 50 Flow scenarios. In the 20 Flow scenario all routing protocols achieved over 95% packet delivery across all node density levels. This is because the total number of control packets introduced into the network consumes a small portion of the available bandwidth which still leaves a reasonable level of bandwidth for data transmission. However, in the 50 Flow scenario, DZTR outperform all the other routing strategies through all levels of node density. This becomes more evident as the number of nodes are increased to 500 nodes, where the gap between the curve for DZTR and the curve for the other routing protocols becomes wider. It can be seen that at the 500 node density level, AODV, LAR1 and LPAR achieve less than 50% PDR, whereas DZTR achieves over 80%. This is because in DZTR, the increase in the number of nodes may not increase the number of zones in the network. This means that the number of neighbouring zones for each zone may not increase significantly. As a results, the number of retransmitting nodes chosen from the interzone table will remain reasonably low. In contrast, in AODV, LAR1 and LPAR, the increase in node density will increase the number of retransmitting nodes. This will reduce the available bandwidth for data transmission and increase channel contention, which will result in further packet losses due to buffer overflows. Furthermore, in DZTR, a link

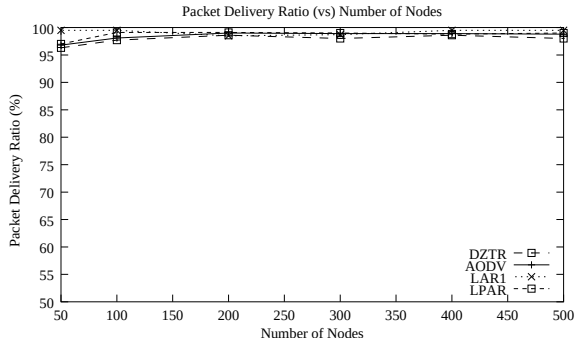

Fig. 1. PDR for 20 Flows

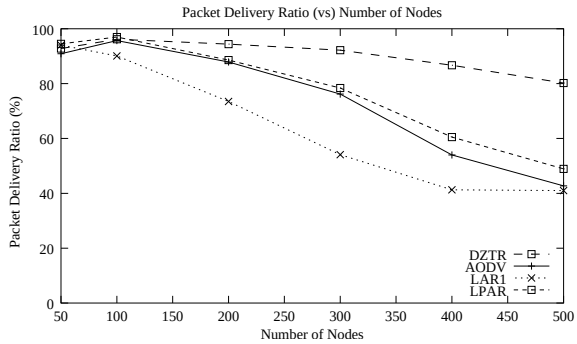

Fig. 2. PDR for 50 Flows

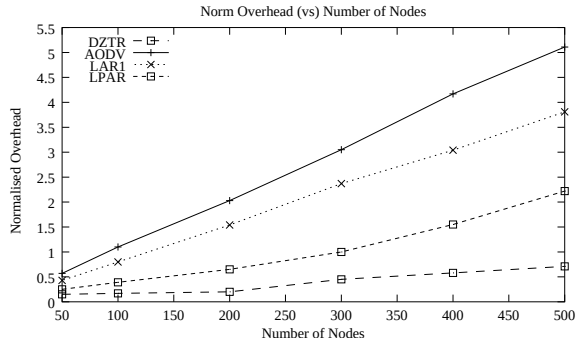

Fig. 3. Normalised overhead for 20 Flows

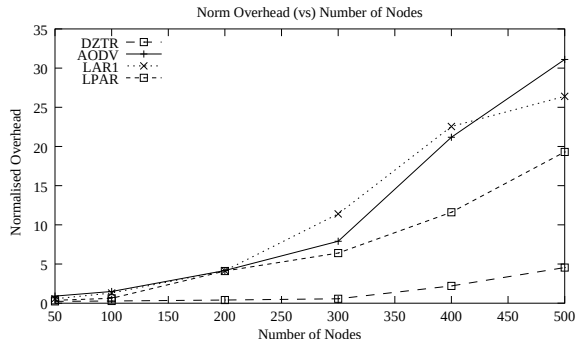

Fig. 4. Normalised overhead for 50 Flows

failure may not initiate a re-discovery of another route, if another gateway node can successfully transmit the data packets. Whereas in AODV, LAR1 and LPAR a link failure may require an alternate route to be discovered. LAR1, attempts to reduce the number of route recalculations by storing multiple route in a route cache (DSR based). However, since the best route is always used first, then storing alternate route may not be beneficial when mobility is high. Since this route may already be expired or broken when it is required. Hence, in this case, recalculation on alternate route may not be avoided by storing multiple routes. Similarly, in LPAR, the secondary route may expire before a link breaks in the primary route. This means that the alternate route in LPAR may not be always available or valid, especially during high levels of mobility. Therefore, the source nodes may be required to make frequent route recalculations, which will increase the level of bandwidth consumed by routing packets throughout the network.

B. *Normalised Routing Overhead Results*

Figure 3 and 4 demonstrate the normalised control overhead for the 20 Flows and the 50 Flows scenarios. In both scenarios DZTR produces the least amount of overhead per packet. Note that as the node density is increased, DZTR maintains the flattest curve when compared to the other three routing strategies, which shows that number of retransmitting nodes do not significantly increase in DZTR. Therefore, the total number of control packets disseminated into the network remains reasonably low as the node density is increased. This shows that DZTR scales significantly better than the other strategies. AODV produces more overhead that the other strategies across all different levels of node density in the 20 Flow scenario. However, in the 50 Flow scenario AODV and LAR1 produce similar levels of overhead. This is because LAR1 performs source routing rather than point-to-point routing, which means the rate at which route failures occur will be higher than the point-to-point based routing protocols (i.e. AODV, LPAR and DZTR), since the routes are not adaptable to the changes in network topology. Therefore, link failures in LAR1 will initiate more route recalculations at the source than in the point-to-point routing protocols. LPAR produces fewer routing packet than AODV and LAR1 in both of the 20 Flow and the 50 Flow scenario. This reduction is achieved by using the 3-state route discovery strategy, which attempts to find a route to a required destination by unicasting if location information about the destination is available. Thus reducing the need for broadcasting during route discovery. Furthermore, LPAR reduces the number of control packet retransmission by flooding over stable links only.

C. *Delay Results*

Figure 5 and 6 illustrate the end-to-end delay experienced by each data packet for the 20 Flows and the 50 Flows scenarios. In the 20 Flow and 50 Flow scenario for the 50 node network DZTR produces longer delays than the other strategies. Two factor contribute to this extra delay, firstly when the node density is low, the nodes may be engaged in zone creation more frequently as the chance for network partitioning to occur is much higher. This means nodes may go in and out of single-state mode or may become temporary members. Therefore, the information kept in each interzone table may not be very accurate, and the first cycle of route discovery may not always be successful.

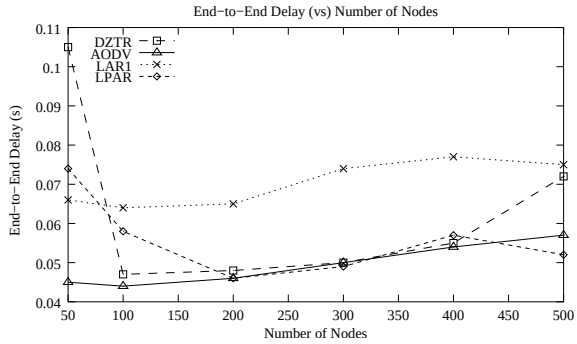

Fig. 5. End-to-end delay for 20 Flows

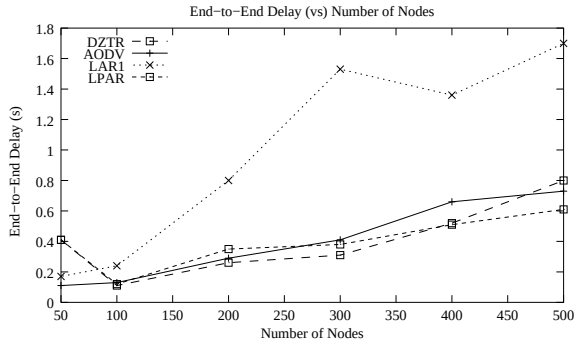

Fig. 6. End-to-end delay for 50 Flows

The second factor is due to stable routing. In DZTR, a source nodes attempts of find a route over stable links, similar to LPAR, which limits the number of nodes which can rebroadcast. Therefore, more attempts maybe required to determine a route over less stable links. This increase in extra delay can be also seen in LPAR. AODV (which uses Expanding Ring Search, ERS) produces the lowest delay in the 50 node scenario, and maintains similar levels of overhead when compared with DZTR and LPAR. This is because, AODV the flooding nature of AODV allows every node to rebroadcast (if the RREQ packet has not expired). Therefore, it calculates the path between the source to the destination more quickly. When the node density is increased to 100, DZTR's end-to-end delay drop dramatically. This is because the higher node density allows DZTR to calculate the required routes more quickly as the LZS-MF strategy becomes more effective in its first route discovery cycle). The delay experienced by all protocols increases slowly as the number of nodes is increased. AODV, LPAR and DZTR experience similar levels of delay for all node density levels greater than 500. However, LAR1 continues to produce larger delays than the other routing protocols during higher node density levels. This is because when mobility is high, more packets may travel over non-optimal routes with larger hop counts, which may be stored in a route cache. Therefore, these packets will experience longer end-to-end delay than the ones travelling over the shortest path. Furthermore, as the node density is increased, the number of routes stored in the route cache may also increase. This means that more non-optimal routes with large hop counts may be available for each required destination. Hence, the probability of longer (non-optimal) end-to-end delay experienced by each packet also increase.

V. Conclusions

This paper presented a new routing protocol for mobile ad hoc networks, which is called Dynamic Zone Topology Routing (DZTR). The idea behind this protocol is to group nodes that are in close proximity of each other into zones. By grouping nodes together and allowing routing and data transmission to be carried out by a group of nodes, we eliminate single points of failure during data transmission, distribute network traffic through a set of nodes and avoid frequent route recalculation. The topology of each routing zone is maintained proactively and each zone member node is aware of the neighbouring zones through the gateway nodes. DZTR reduces routing overheads by reducing the search zone and allowing only selected nodes to forward the control packets. Each node that migrates between zones also leaves transient zone trails, which assist our proposed search strategy to find the destination more quickly and with fewer overheads. Our theoretical overhead analysis and simulation studies showed that DZTR significantly reduces the number of control packets transmitted into the network and achieves higher levels of packet delivery under worst case network conditions when compared to AODV, LAR1 and LPAR.

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] M. Abolhasan, T. Wysocki, and E. Dutkiewicz. Zone-Based Routing Algorithm for Mobile Ad Hoc Networks. In *proceedings of AD-HOC NetwOrks and Wireless (ADHOC-NOW) FIELDS INSTITUTE September 20-21*, Canada, Toronto, 2002.

[5] Mehran Abolhasan, Tadeusz Wysocki, and Eryk Dutkiewicz. LPAR: An Adaptive Routing Strategy for MANETs. *In Journal of Telecommunication and Information Technology*, pages 28–37, 2/2003.

[6] 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.

[7] 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.

[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] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Seung-Chul Woo and Suresh Singh. Scalable Routing Protocol for Ad Hoc Networks. *accepted for publication in Journal of Wireless Networks (WINET)*, 2001.