

November 2005

Efficient and highly scalable route discovery for on-demand routing protocols in ad hoc networks

Mehran Abolhasan

University of Wollongong, mehran.abolhasan@uts.edu.au

Justin Lipman

University of Wollongong, jlipman@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Abolhasan, Mehran and Lipman, Justin: Efficient and highly scalable route discovery for on-demand routing protocols in ad hoc networks 2005.
<https://ro.uow.edu.au/infopapers/35>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Efficient and highly scalable route discovery for on-demand routing protocols in ad hoc networks

Abstract

This paper presents a number of different route discovery strategies for on-demand routing protocols, which provide more control to each intermediate node make during the route discovery phase to make intelligent forwarding decisions. This is achieved through the idea of selfselection. In self-selecting route discovery each node independently makes Route Request (RREQ) forwarding decisions based upon a selection criterion or by satisfying certain conditions. The nodes which do not satisfy the selection criterion do not rebroadcast the routing packets. We implemented our self-selecting route discovery strategies over AODV using the GloMoSim network simulation package, and compared the performance with existing route discovery strategies used in AODV. Our simulation results show that a significant drop in the number of control packets can be achieved by giving each intermediate node more authority for self-selection during route discovery. Furthermore, a significant increase in throughput is achieved as the number nodes in the network is increased.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Abolhasan, M and Lipman, J, Efficient and highly scalable route discovery for on-demand routing protocols in ad hoc networks, The IEEE 30th Anniversary Conference on Local Computer Networks, 15-17 November 2005, 358-366. Copyright IEEE 2005.

Efficient and Highly Scalable Route Discovery for On-demand Routing Protocols in Ad hoc Networks

Mehran Abolhasan and Justin Lipman
Telecommunications and IT Research Institute (TITR)
University of Wollongong, NSW 2522, Australia
E-mail: mehran@titr.uow.edu.au

Abstract— This paper presents a number of different route discovery strategies for on-demand routing protocols, which provide more control to each intermediate node make during the route discovery phase to make intelligent forwarding decisions. This is achieved through the idea of *self-selection*. In self-selecting route discovery each node independently makes Route Request (RREQ) forwarding decisions based upon a selection criterion or by satisfying certain conditions. The nodes which do not satisfy the selection criterion do not rebroadcast the routing packets. We implemented our self-selecting route discovery strategies over AODV using the GloMoSim network simulation package, and compared the performance with existing route discovery strategies used in AODV. Our simulation results show that a significant drop in the number of control packets can be achieved by giving each intermediate node more authority for self-selection during route discovery. Furthermore, a significant increase in throughput is achieved as the number nodes in the network is increased.

I. INTRODUCTION

Ad hoc networks are seen as the next step towards designing networks which are instantly deployable and self manageable. Consequently, in an ad hoc network each end-user node is capable of sending, receiving and routing data packets in a distributed manner. Moreover, such networks can be configured to allow for mobility and perform routing over multiple hops. These networks are commonly referred to as Mobile Ad hoc networks (MANETs) Constraints such as low bandwidth, limited energy, mobility, non-deterministic topology and the broadcast nature of wireless communication make the efficient routing of data a critical element of ad hoc networks.

Existing research in ad hoc network routing has contributed significantly to improved routing through maximising the usage of prior knowledge of nodes, improving stability of routes and creating a collaborative environment between nodes (Hybrid routing). However, little work has been done in improving the process of route discovery when no prior node or topology knowledge is available. In reactive routing protocols, improving the efficiency of route discovery is one key to providing higher scalability as network density increases.

More importantly, if only a Blind flood is performed then the route determined is generally the shortest path (as all routes are determined in parallel during a Blind flood) and is not necessarily the best route in terms of resources. In this paper, we present a number of different self-selecting route discovery strategies, which allow for intermediate nodes to selectively participate in route discovery. The aim of these strategies is to reduce the Broadcast Storm Problem [16] in terms of the number of control packets and the level of medium contention in the network. Thereby, achieving higher levels of scalability. Additionally, such strategies are able to provide more control to individual nodes to better manage their limited resources (such as battery power) and to determine more effective routes between end nodes.

The rest of this paper is organised as follows. Section II describes the previous work performed in MANET routing. Section III, describes the proposed self-selecting route discovery strategies. Section IV describes the simulation environment, parameters and metrics used to investigate the performance of the proposed route discovery strategies with AODV. Section V presents a discussion of the obtained simulation results. Section VI concludes our paper.

II. ROUTING IN AD HOC NETWORKS

Design an efficient routing protocol for Ad hoc network has proven to be a very challenging task. Up to now, a range of routing protocols have been proposed for ad hoc networks. However, designing a scalable routing protocol is still an open research issue. The proposed Ad hoc network routing protocols from the literature may be classified into three groups: proactive, reactive and hybrid. In this section, routing protocols are briefly described with an emphasis on how they disseminate control information and perform route discovery.

Proactive routing was the first attempt at designing routing protocols for MANETs. Early generation proactive protocols such as DSDV [17] and GSR [5] were based on the traditional distance vector and link state

algorithms, which were originally proposed for wired networks. These protocols periodically maintain and distribute route information to all nodes within the network. The disadvantage of these strategies were their lack of scalability due to exceedingly large overhead produced due to Blind flooding. Blind flooding is shown to result in the Broadcast Storm Problem [16] and is thus not efficient. Other proactive routing protocols such as Fisheye State Routing (FSR) [9] limit the rate at which they update route information depending on the distance. Routes to closer nodes are maintained more regularly, whereas routes to remote nodes are maintained less regularly. Source-Tree Adaptive Routing (STAR) [8] eliminates periodic dissemination of control information in favour of conditional dissemination, thus reducing the constant overhead. However, Blind flooding is still required. In Cluster-head Gateway Switch Routing (CGSR) [6] a hierarchy is created based upon node clustering. Clusterheads control the flow of route information within their cluster and between clusters, thus reducing the amount of route information and limiting the dissemination of route information. More recent attempts at reducing control overhead in proactive routing can be seen in protocols such as OLSR [10] and TBRPF [4]. These protocols attempt to reduce the control overhead by reducing the number of rebroadcasting nodes in the network through optimised flooding.

Reactive (on-demand) routing protocols attempt to reduce the amount of control overhead disseminated in the network by determining routes to a destination only when it is required. This is usually achieved through a two phase route discovery process initiated by a source node. The first phase of route discovery starts by the propagation of Route Request (RREQ) packets throughout the network using a simple Blind flooding approach. The second phase is initiated when a RREQ packet reaches a node which is the destination or has a route to the destination, in which case a Route Reply (RREP) packet is generated and transmitted back to the source node. Reactive routing protocols produce significantly lower amounts of routing overhead when compared with proactive routing protocols when the number of flows in the network are low. However, for large number of flows reactive protocols experience a significant drop in data throughput. This is because routing control packets are usually Blind flooded (globally) throughout the entire network to find a route to the destination - resulting in the Broadcast Storm Problem.

To limit the effects of Blind flooding in reactive routing a number of different strategies have been proposed. The Routing On-demand Acyclic Multi-path (ROAM) [18] protocol limits the effects of flooding by using

directed acyclic subgraphs based upon distance between the source and destination for the propagation of a flood. This eliminates the propagation of a flood in a direction along a subgraph if the destination is not reachable along that subgraph. In Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) [2], overhead associated with route discovery is reduced and localised by limiting each RREQ packet to a certain number of hops. However, this localisation of route requests can only occur if the source and destination node have communicated before and exchanged position information. If the nodes have not communicated before, then the route request is not localised. Location Aided Routing (LAR) [13] requires that nodes have a GPS device and therefore are aware of their location. Thus overhead associated with route discovery is reduced by limiting the direction and scope of flooding. This protocol defines zones specifying which direction a RREQ packet may travel towards. RREQ packets therefore only travel in the approximate direction of the intended destination. In LPAR [1] a combination of prior location knowledge and unicasting is used to reduce the number of re-broadcasting nodes within a search zone. Cluster-Based Routing Protocol (CBRP) [11] is a hierarchical routing protocol based upon clustering. Clusterheads are defined and responsible for the nodes within each cluster. To reduce the effects of route discovery, only clusterheads exchange and propagate RREQ packets. Both Dynamic Source Routing (DSR) [12] and Ad hoc On-Demand Distance Vector Routing (AODV) [7] protocols utilise Blind flooding as a means of performing route discovery. However, they differ in the way they maintain routes to destination nodes and also in the amount of information required to route packets. To reduce the effects of Blind flooding, these protocols use route caching as well as limiting the number of hops for route discovery. In AODV the source nodes use Expanding Ring Search (ERS) to search nearby nodes first, thereby reducing the number of globally propagated control packets.

Hybrid routing protocols combine both reactive and proactive routing characteristics to achieve high levels of scalability. Generally in hybrid routing protocols, proactive routing is used within a limited region. These regions can be a cluster, a tree or a zone, which may contain a number of end-user nodes. Reactive routing is used to determine routes, which do not lie within a source node's local region. The idea behind this approach to routing is to allow nearby nodes to collaborate and reduce the number of re-broadcasting nodes. Therefore, during a route discovery only a selected group of nodes within the entire network may rebroadcast packets.

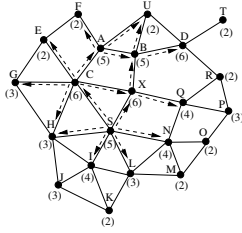


Fig. 1. Illustration of Utility-based Self-Selection for Route Discovery

III. SELF-SELECTING ROUTE DISCOVERY

In this section, we propose a number of different self-selecting route discovery strategies. The idea behind these strategies is to reduce the number of redundant Route Request (RREQ) packets rebroadcast by allowing each intermediate node to selectively participate in route discovery. To illustrate the benefits of self-selecting strategies, assume a node S (see Figure 1) is required to discover a route to node D without any prior knowledge (e.g. hop count, location information). Now, assume each node maintains a utility function, U , (e.g. based on mobility, topology and power). To minimise the number of route request retransmissions, we can modify the route discovery procedure to allow the nodes with the highest levels of utility to rebroadcast in the first route discovery attempt. Lets assume in this case, only nodes with a utility level greater than 4 may rebroadcast. In this scenario, only five nodes rebroadcast whereas using a pure flooding approach 21 nodes may rebroadcast. Hence, a reduction of 17 nodes is achieved. In networks with high node density and traffic, such strategies may significantly improve data throughput and allow each node to conserve resources if required.

In the following sections, we introduce two different self-selection strategies: Source-Driven Self-Selection (SDSS) and Pure Self-Selection (PSS). In SDSS, the source node specifies a utility metric in each RREQ packet. Therefore, all nodes that do not meet this metric may choose not to participate in route discovery. In PSS, each intermediate node calculates its own utility and based on this decides whether or not take part in route discovery.

A. SDSS based on Mobility (SDSS-M)

The idea of using mobility to minimise the number of control packets was introduced in DREAM [3]. DREAM is a proactive routing strategy, which optimises the frequency at which route updates are sent by nodes to the speed at which they travel. Therefore, the nodes, which travel at high speeds send update packets more frequently. In this strategy, we use mobility to reduce route discovery redundancy in on-demand routing. To do this, we modify

the route discovery strategy to restrict the RREQ rebroadcasts packets to occur over more stationary nodes first. A utility function is introduced, which determines a maximum allowable node speed during each route discovery phase. Therefore, only nodes which are travelling at a lower speed than the one specified in the utility function will rebroadcast. The benefits of the SDSS-M strategy include:

- Route stability may be higher than using pure blind flooding as selecting least mobile nodes could result in fewer route failures.
- Since fewer nodes will be rebroadcasting during each route discovery the Broadcast Storm Problem is reduced thus limiting channel contention.
- The total number of control packets may be reduced significantly, especially in dense networks.

The SDSS-M route discovery algorithm is outlined below:

Algorithm SDSS-M

(* SDSS based on Mobility algorithm *)

1. $RREQ_{max} \leftarrow$ Maximum number of route request retries
2. $V_{max} \leftarrow \tau$ Maximum speed at which a node can travel
3. $V_u \leftarrow$ Maximum allowable node speed
4. $V_{NoMax} \leftarrow$ Flag used for pure flooding
5. $P \leftarrow \{0.125, 0.25, 0.5, 0.75, 1.0\}$ (* Used to select different speed levels *)
6. $RREQ_{max} \leftarrow 6$
7. **for** $i \leftarrow 0, i \neq RREQ_{max}, i++$
8. $V_u \leftarrow V_{max} \cdot P_i$
9. Forward_RREQ(P_i, V_u)
10. wait for reply
11. **if** $Route = found$
12. break loop
13. initiate data transmission
14. **if** $Route = not\ found$
15. Forward_RREQ($0, V_{NoMax}$)
16. wait for reply
17. **if** $Route = found$
18. initiate data transmission
19. **else**
20. return route not found

In the SDSS-M algorithm, the source node begins by calculating the mobility utility function (V_u), which selects a value for maximum allowable velocity at each intermediate node during a route discovery phase. This value is then passed to the *Forward_RREQ* function where it is attached to the RREQ packet and disseminated to the network. When an intermediate node receives a RREQ packet and it does not have a route to the required destination, it checks to see if its current speed is less than V_u . If yes, the it will rebroadcast the RREQ packet. Note that in the SDSS-M algorithm, we have selected 5 different mobility levels, which are used to increase V_u when a route discovery fails to determine a route. If a route is still not found, then a final route discovery is initiated, which

allows all nodes to rebroadcast resulting in a Blind flood.

B. SDSS based on Reachability (SDSS-R)

This strategy attempts to prioritise the propagation of RREQ packets over nodes with high levels of reachability to nodes which are away from the source node. We define reachability in this context as the ability to forward route request packets to destinations which are more than one hops away while increasing the search coverage area at each hop. To do this, we limit the route request propagations to nodes which satisfy the following conditions:

- 1) The forwarding node knows a minimum number of nodes, which are more than one hop away.
- 2) The distance between the forwarding node and the source node is greater than the distance between the previous forwarding node and the source.
- 3) The forwarding node is significantly far from the previous forwarding node.

To show how the above conditions are used to limit the number of forwarding nodes, we define the following parameters:

- M , number of nodes which are more than one-hop away and N , the number of destination nodes known at each node (route table size).
- D_{FS} , distance between the forwarding node and the source and D_{PS} , distance between the previously forward node and the source.
- D_{min} , minimum required between the forwarding node and the previously forwarding node.
- D_{FP} be the distance between the forwarding node and the previously forwarding node.
- T_R be the maximum transmission range of a node.

Using the above metrics, we define a reachability metric, R , such that:

$$R \propto \frac{D_{FP}}{T_R} \quad \text{For } T_R > D_{FP}$$

and

$$R \propto \frac{M}{N} \quad \text{For } N > 0$$

We combine the above two equations to get:

$$R = \left(\frac{D_{FP}}{T_R} + \frac{M}{N} \right)$$

or

$$R = \frac{ND_{FP} + MT_R}{NT_R} \quad (1)$$

Next, we find the limit of our equation in order to normalise equation 1 to vary between 0 and 1, with 0 being no reachability and 1 being 100% reachability.

$$\lim_{\substack{D_{FP} \rightarrow T_R \\ M \rightarrow N}} \frac{ND_{FP} + MT_R}{NT_R} = 2$$

Therefore, normalising with the above limit we get:

$$R_n = \frac{ND_{FP} + MT_R}{2NT_R} \quad (2)$$

In on-demand routing protocols such as AODV, the M and N parameters of the reachability equation can be determined from the routing table. Moreover, the distance calculation between the nodes (e.g. D_{FP}) can be determined by appending GPS coordinates of the source nodes and the currently forwarding node to the RREQ packet¹.

Algorithms *SDSS-R-I* and *SDSS-R-R* illustrate the route discovery initiation and the relay strategy used in SDSS-R.

Algorithm *SDSS-R-I*

(* SDSS based upon Reachability - Initiation algorithm *)

1. $RREQ_{max} \leftarrow$ Maximum number of route request retries
2. $RREQ_c \leftarrow$ current number of route retries by the source
3. $L_S \leftarrow$ Location of the source
4. $D_{min} \leftarrow$ Minimum Distance required between forwarding nodes
5. $P \leftarrow \{0.75, 0.65, 0.5, 0.25, 0.125\}$ (* Used by the forwarding nodes to compare their reachability *)
6. **for** $i \leftarrow 0, i \neq RREQ_{max}, i++$
7. $RREQ_c \leftarrow i$
8. Forward_RREQ($P_i, D_{min}, RREQ_c$)
9. wait for reply
10. **if** $Route = found$
11. break loop
12. initiate data transmission
13. **if** $Route = not\ found$
14. Forward_RREQ($0, 0, RREQ_{max}$)
15. wait for reply
16. **if** $Route = found$
17. initiate data transmission
18. **else**
19. return route not found

Algorithm *SDSS-R-R*

(* SDSS based upon Reachability - Relay algorithm *)

1. $RREQ_c \leftarrow$ current number of route retries by the source
2. $N \leftarrow$ Route Table Size
3. $M \leftarrow$ Num of dest 2 or more hops away
4. $T_R \leftarrow$ Max Tx range of nodes
5. $L_S \leftarrow$ Location of the source
6. $L_F \leftarrow$ Location of current node
7. $L_P \leftarrow$ Location of the previously forwarding node
8. $D_{min} \leftarrow$ Minimum Distance required between forwarding nodes

¹Each forwarding node over-writes its own GPS coordinates replacing the GPS position of the previously forwarding node in the RREQ packet

9. $R_n \leftarrow$ Reachability function
10. $D_{FP} \leftarrow \text{dist}(L_F, L_P)$ Distance between the current and previous node
11. $D_{FS} \leftarrow \text{dist}(L_F, L_S)$ Distance between the current node and source
12. $D_{PS} \leftarrow \text{dist}(L_P, L_S)$ Distance between the previous node and source
13. $P \leftarrow$ Minimum reachability required (assigned by the source)
14. **if** $P > 0$ **and** $D_{FS} \leq D_{PS}$ Drop RREQ packet
15. **if** $RREQ_c < 1$ **and** $D_{min} > D_{FP}$
16. Drop RREQ packet
17. **else**
18. $R_n \leftarrow \frac{N \cdot D_{FP} + M \cdot T_R}{2 \cdot N \cdot T_R}$
19. **if** $R_n < P$
20. Drop RREQ packet
21. **else**
22. Forward RREQ packet

As in the *SDSS-M* algorithm, in the *SDSS-R-I* algorithm the source node uses P to select different utility levels. However, in this algorithm the utility level is adjusted in descending order. Therefore, during the first route discovery, only the nodes with reachability levels of $R_n > 0.75$ will retransmit. If the first route discovery phase is unsuccessful then the nodes with lower levels of reachability will also be allowed to rebroadcast when another route discovery initiated. Also note, to further reduce redundancy during the first phase of route discovery phase only the nodes, which are at least D_{min} away from the previously forwarding node will re-broadcast. Furthermore, during all other route discovery attempts (instead of the last one, where $P = 0$), only the nodes which are further away from the source than the previously forwarding node will rebroadcast.

C. Pure Self-Selection based on Reachability (PSS-R)

In PSS, each intermediate node selects its own utility metric and decides whether to further rebroadcast a RREQ packets. In this section we propose two different strategies, which illustrate how route discovery redundancy can be reduced by using Pure Self-Selection strategies.

Similar to the *SDSS-R-R* algorithm, in PSS-R algorithm, a utility metric is calculated based on a reachability. However, in this strategy, each forwarding node determines a probability function to selectively drop RREQ packets based on the reachability metric at each node (i.e. nodes with higher levels of reachability will have a probability of forwarding the data packets). To show how this is done, let U_A be the probability of rebroadcasting and P_d be the probability of dropping a packet at each node. Then, using the reachability metric (from the previous section), R_n , which varies from 0 to 1, we let $U_A = R_n$. Therefore, the probability at which each packet is dropped is $P_d \propto (1 - U_A)$ (or simply

$P_d \propto (1 - R_n)$). The PSS-R algorithm is outlined below:

Algorithm PSS-R

(* PSS based on Reachability algorithm *)

1. $N \leftarrow$ Route Table Size
2. $M \leftarrow$ Num of dest 2 or more hops away
3. $T_R \leftarrow$ Max Tx range of nodes
4. $L_S \leftarrow$ Location of the source
5. $L_F \leftarrow$ Location of current node
6. $L_P \leftarrow$ Location of the previously forwarding node
7. $D_{min} \leftarrow$ Minimum Distance required between forwarding nodes
8. $R_n \leftarrow$ Reachability function
9. $D_{FP} \leftarrow \text{dist}(L_F, L_P)$ Distance between the current and previous node
10. $P_d \leftarrow$ Probability of RREQ being dropped
11. $Seed \leftarrow$ A random seed
12. $D_n \leftarrow$ dropping value
13. $R_n \leftarrow \frac{N \cdot D_{FP} + M \cdot T_R}{2 \cdot N \cdot T_R}$
14. $P_d \leftarrow (1 - R_n) \times 100$
15. $D_n \leftarrow \text{Random}(0, 100)$ Assign a random number between 0 and 100
16. **if** $P_d \geq D_n$
17. drop RREQ packet
18. **else**
19. rebroadcast RREQ packet

D. Pure Self-Selection based on Mobility (PSS-M)

The PSS-M algorithm is similar to the PSS-R algorithm. However, instead of using reachability, node speed is used to determine the probability of dropping each control packet. To do this, we let τ equal to a maximum speed threshold (as in the *SDSS-M* algorithm). Using the current speed of the node V_c , we let $P_d \propto \frac{V_c}{\tau}$ (where $V_c \leq \tau$). Then, the probability of forwarding, $U_A \propto (1 - \frac{V_c}{\tau})$. Therefore, the nodes with higher level of mobility drop RREQ packets more frequently than the more stationary ones. The PSS-M algorithm is outlined below:

Algorithm PSS-M

(* PSS based on Mobility algorithm *)

1. $\tau \leftarrow$ Maximum speed at which a node can travel
2. $V_c \leftarrow$ current node speed
3. $U_A \leftarrow$ Probability of a RREQ being forwarded
4. $P_d \leftarrow$ Probability of RREQ being dropped
5. $Seed \leftarrow$ A random seed
6. $D_n \leftarrow$ dropping value
7. $P_d \leftarrow \frac{V_c}{\tau}$
8. $U_A \leftarrow (1 - P_d) \times 100$
9. $D_n \leftarrow \text{Random}(0, 100)$ Assign a random number between 0 and 100
10. **if** $U_A \leq D_n$
11. drop RREQ packet
12. **else**
13. rebroadcast RREQ packet

IV. SIMULATION SCENARIOS AND PERFORMANCE METRICS

The GloMoSim[14] simulation package was chosen to run the simulations. GloMoSim is an event driven simulation tool designed to carry out large simulations for mobile ad hoc networks. The simulations were performed for 50 and 100, 300 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 height 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[15]. 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 simulations ran for 200s 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 512 Bytes and each packet were at 0.25s intervals. The simulation was run for 10 and 20 different client/server pairs² and each session begin at different times and was set to last for the duration of the simulation. We implemented our proposed Self-Selecting strategies over the AODV algorithm.

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

- Packet Delivery Ratio (PDR)
- Control (O/H)
- End-to-End Delay

PDR is the Ratio of the number of number of packets received by the destination to the number of packets sent by the source. Control overhead (O/H) presents the number of control packets transmitted through the network. The End-to-End Delay represents the average delay experienced by each packet when travelling from the source to the destination.

V. RESULTS

This sections presents the results obtained for AODV and the proposed Self-Selecting strategies described in section III.

A. Packet Delivery Ratio

Figures 2, 3 and 4 illustrate the Packet Delivery Ratio (PDR) for the 50, 100 and 300 node scenarios respectively. In the 50 node scenario, where the node and traffic density is low, all routing strategies achieve over 97%

²Note that the terms Client/Server, src/dest and Flows are used interchangeably

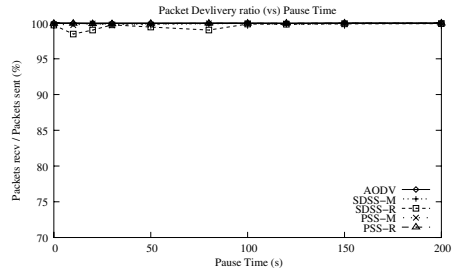


Fig. 2. PDR: 50 Nodes and 10 Flows

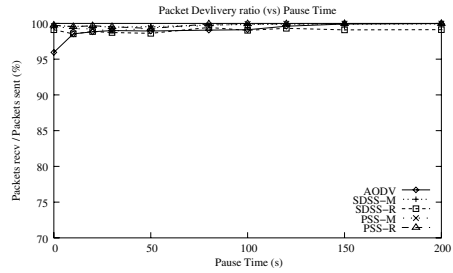


Fig. 3. PDR: 100 Nodes and 20 Flows

PDR. In the 100 node scenario, AODV begins to show signs of drop in performance when mobility is high (i.e. zero pause time). For lower levels of mobility all routing strategies continue to achieve over 97% PDR. In the 300 node scenario, AODV experiences up to 15% drop in performance compared to the 100 and 50 node scenarios. Other routing strategies show a slight drop in performance, however, they continue to achieve over 95% PDR. This is due to fewer control packets disseminated into the network by the self selecting strategies when compared to AODV, which means that there is less medium contention and more available bandwidth for data packet transmission. This is particularly apparent during high levels of mobility where fewer data packets are dropped by the self selecting strategies when compared to AODV.

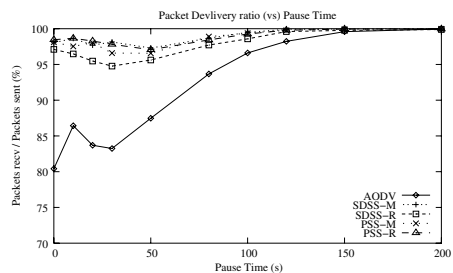


Fig. 4. PDR: 300 Nodes and 20 Flows

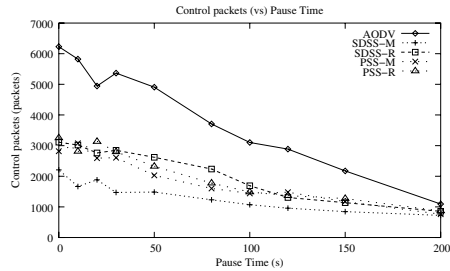


Fig. 5. O/H: 50 Nodes and 10 Flows

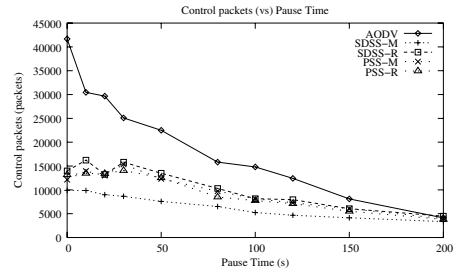


Fig. 6. O/H: 100 Nodes and 20 Flows

B. Control Packets

Figures 5, 6 and 7 illustrate the number of control packets introduced into the network for the 50, 100 and 300 node scenarios respectively. In all scenarios it can be seen that the self-selecting strategies produce significantly fewer control packets than AODV. In the 50 node scenario, SDSS-M produces significantly fewer control packets than the other self-selecting strategies. This is because when the node density is low the reachability based flooding algorithms (i.e. SDSS-R and PSS-R) are less effective, since the reachability information available at each node is also low. Therefore, the first iteration of the reachability algorithm where only the nodes with highest reachability information rebroadcast the RREQ packets, may not always be successful. Thus, more iterations with lower reachability requirements may be needed to determine a route, which results in more control packets being disseminated into the network. The PSS-M algorithm also produces more control packets than SDSS-M in this scenario. This is because the nodes with higher levels of mobility may still have a chance to rebroadcast during the first iteration of the route discovery process. However, in SDSS-M, only the nodes with the least level of mobility can rebroadcast during the first route discovery attempt. In the 100 node scenario, the SDSS-M continues to outperform all other strategies. However, the gap between the SDSS-M curve and the other self selecting strategies is smaller than in the 50 node scenario. This is because each node has more reachability information due to an increase in both the number of nodes and flows, which improves the effectiveness of the reachability metric calculated at each node. Therefore, the reachability based algorithms have a higher probability of determining a route to the destination during the first iteration of route discovery. In the 300 node scenario, all self selecting strategies show similar levels of performance. Furthermore, the self selecting strategies show higher levels of improvement over AODV when compared to the 50 and 100 node scenarios. This is because the first attempt at route discovery by each self selecting strategy has a greater probability of success, due to higher node density.

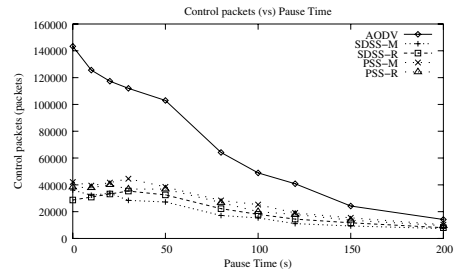


Fig. 7. O/H: 300 Nodes and 20 Flows

C. Delays

Figures 8, 9 and 10 illustrate the end-to-end delay introduced for the 50, 100 and 300 node network scenarios respectively. In the 50 node scenario, the SDSS-R strategy experiences higher levels of delay than all other strategies. This is due to the low levels of reachability information available at each node, which means that the first iteration of the route discovery phase is less successful than the other self selecting strategies. In the 100 node scenario, the pure self selective strategies slightly outperform the source driven strategies. This is because the source driven strategies may require more than one route discovery process to determine a route, as they are more selective than the pure self selection strategies. AODV produces the highest level of delay during the zero pause time. This is mainly due to the high number of control packets produced by AODV, which drastically increases channel contention and queueing at each node. Therefore, each data packet may experience longer queueing compared to the other strategies. In the 300 node scenario, all self selecting strategies produce similar levels of delays. However, AODV begins to show a significant drop in performance, as it produces far more delays in the mid to high level mobility range. This is also due to a significant increase in the number of control packets produced by AODV in the 300 node scenario compared to the 100 node scenario.

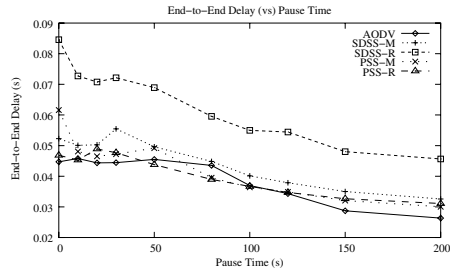


Fig. 8. Delays: 50 Nodes and 10 Flows

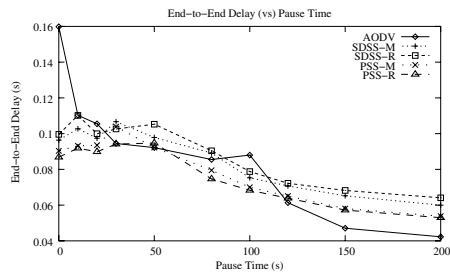


Fig. 9. Delays: 100 Nodes and 20 Flows

VI. CONCLUSIONS

In this paper, we introduce the idea of self-selection to improve the route discovery performance of on-demand routing protocols. Two different types of self-selection strategies are introduced: Source-Driven and Pure Self Selection. In the source driven strategies, each intermediate node makes forwarding decisions based on a utility metric specified by the source during each route discovery phase. The pure self selection strategies make routing decisions independently of the source or any other node. We implemented our self selection strategies over AODV and compared their performance with each other and AODV. Our results show that the self selecting strategies significantly reduce the number of control packets and the end-to-end delay. when the node density is high. They also achieve higher levels of data packet delivery than AODV. In the future we plan to further investigate the performance of our self selection strategies in larger

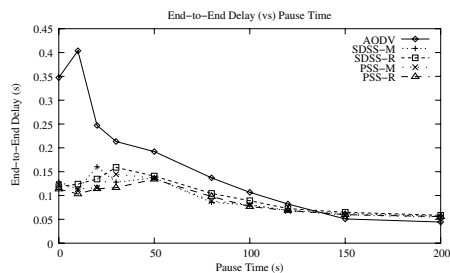


Fig. 10. Delays: 300 Nodes and 20 Flows

and more dense networks.

REFERENCES

- [1] 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.
- [2] George Aggelou and Rahim Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *ACM International Workshop on Wireless Mobile Multimedia (WoW-MoM)*, pages 26–33, 1999.
- [3] 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.
- [4] B. Bellur, R.G. OGIER, and F.L. Templin. Topology broadcast based on reverse-path forwarding routing protocol (tbrpf). In *Internet Draft, draft-ietf-manet-tbrpf-06.txt, work in progress*, 2003.
- [5] T-W. Chen and M. Gerla. Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks. *Proc. IEEE ICC*, 1998.
- [6] C-C. Chiang. Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. In *Proc. of IEEE SICON*, pages 197–211, april 1997.
- [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] 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.
- [9] M. Gerla. Fisheye State Routing Protocol (FSR) for Ad Hoc Networks. In *Internet Draft, draft-ietf-manet-fsr-03.txt, work in progress*, 2002.
- [10] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks, *iee inmic pakistan*. 2001.
- [11] M. Jiang, J. Ji, and Y.C. Tay. Cluster Based Routing Protocol. In *Internet Draft, draft-ietf-manet-cbrp-spec-01.txt, work in progress*, 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] 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.
- [14] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. Glomosim scalable simulation environment for wireless and wired network systems. In <http://pcl.cs.ucla.edu/projects/gloimosim/>, 2003.
- [15] Lucent. Orinoco pc card. In <http://www.lucent.com/orinoco>, 2003.
- [16] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, USA*, pages 151–162. ACM Press, 1999.
- [17] 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.
- [18] J. Raju and J. Garcia-Luna-Aceves. A new approach to on-demand loop-free multipath routing. In *A New Approach to On-demand Loop-Free Multipath Routing, Proceedings of the 8th Annual IEEE International Conference on Computer Communications and Networks (ICCCN), Boston, MA, October 1999, pp. 522-527.*, 1999.