IMPROVING MANET ROUTING PROTOCOLS THROUGH THE USE OF GEOGRAPHICAL INFORMATION

Vasil Hnatyshin

Department of Computer Science, Rowan University, Glassboro, NJ, USA
hnatyshin@rowan.edu

ABSTRACT

This paper provides a summary of our research study of the location-aided routing protocols for mobile ad hoc networks (MANET). This study focuses on the issue of using geographical location information to reduce the control traffic overhead associated with the route discovery process of the ad-hoc on demand distance vector (AODV) routing protocol. AODV performs route discovery by flooding the whole network with the route request packets. This results in unnecessarily large number of control packets traveling through the network. In this paper, we introduced a new Geographical AODV (GeoAODV) protocol that relies on location information to reduce the flooding area to a portion of the network that is likely contains a path to destination. Furthermore, we also compared GeoAODV performance with that of the Location Aided Routing (LAR) protocol and examined four mechanisms for reducing the size of the flooding area: LAR zone, LAR distance, GeoAODV static, and GeoAODV rotate. We employed OPNET Modeler version 16.0 software to implement these mechanisms and to compare their performance through simulation. Collected results suggest that location-aided routing can significantly reduce the control traffic overhead during the route discovery process. The comparison study revealed that the LAR zone protocol consistently generates fewer control packets than other location-aided mechanisms. However, LAR zone relies on the assumption that location information and traveling velocities of all the nodes are readily available throughout the network, which in many network environments is unrealistic. At the same time, the GeoAODV protocols make no such assumption and dynamically distribute location information during route discovery. Furthermore, the collected results showed that the performance of the GeoAODV rotate protocol was only slightly worse than that of LAR zone. We believe that even though GeoAODV rotate does not reduce the control traffic overhead by as much as LAR zone, it can become a preferred mechanism for route discovery in MANET.

KEYWORDS

Mobile Ad-Hoc Networks; MANET Routing Protocols; Ad Hoc On Demand Distance Vector Routing; Location-Aided Routing; Geographical AODV; OPNET Modeler

1. INTRODUCTION

In this paper we summarize the results of our research endeavors in the area of location-aided routing protocols for mobile ad hoc networks (MANET). In particular, our study focuses on the issue of using geographical location information to reduce the control traffic overhead associated with the route discovery process. MANET routing protocols often rely on flooding to find a route to destination. Flooding is a simple and effective technique for route discovery where each node broadcasts route request message to all of its neighbors. The neighboring nodes repeat the process until a route to destination is found. The message broadcasts are usually "heard" by all the neighboring nodes, including those that have already "seen" (i.e.,

DOI : 10.5121/ijwmn.2013.5201

forwarded) that message. To deal with such situations, MANET routing protocols typically include a mechanism that allows the nodes to identify and discard message duplicates. However, flooding often leads to unnecessary large control message overhead since the whole network is being searched, including the parts of the network that do not contain a route to destination.

In our study we examined how the route discovery process of Ad-Hoc On-demand Distance Vector (AODV) protocol [1-3] can be enhanced through the use of geographical location information. AODV is a stateless reactive routing protocol for MANET, which employs flooding to discover a route to destination. There have been numerous studies that attempted to improve the performance of MANET route discovery through the use of geographical location information [4-14]. This study concentrates on Location-aided routing (LAR) protocol [7-8] and its improvements [5-6]. LAR is a modification of AODV protocol which relies on Global Positioning System (GPS) coordinates and traveling velocity of the nodes to limit flooding to a small area which is likely to contain a path to destination. This approach reduces the amount of control traffic traveling through the network during the route discovery process because only a portion of the network is being searched. Geographical AODV (GeoAODV) [5-6] is a variation of LAR protocol which we developed and studied over the past several years. GeoAODV also relies on the GPS coordinates to reduce the size of the search area during route discovery. However, unlike LAR, GeoAODV assumes that the nodes only know their own location information, while the GPS coordinates of all the other nodes in the network are dynamically distributed during route discovery. GeoAODV defines the search area differently from LAR and it allows expanding the area if the initial attempt to find a route to destination fails. The main contributions of this paper are (1) introduction of a new location-aided protocol called GeoAODV and (2) summary of a simulation study that compares the performance of the AODV protocol and its location-aided variations: LAR zone, LAR distance, GeoAODV static, and GeoADV rotate.

We used OPNET Modeler [15] version 16.0 to implement LAR and GeoAODV protocols and to conduct our simulation studies. OPNET Modeler is leading commercial software for simulation and modeling of computer networks. It supports a wide range of network protocols, technologies, and device models which allows the users to model almost any of today's computer networks. OPNET Modeler relies on combination of the C/C++ code, state transition diagrams, and discrete event simulation engine to model various devices, communication mediums, applications, protocols, and network technologies. While the accuracy of the results generated by OPNET products is very high [15], it comes at the expense of the underlying implementation complexity. Typically, an OPNET model of network protocols such as IP, consists of thousands and thousands lines of C/C++ code distributed through numerous external files and process model modules. Even though the implementation is mostly well documented, identifying the location of the code responsible for modeling certain aspects of the simulated system often is a very challenging task. This paper includes an overview of our endeavors implementing GeoAODV and LAR protocols using OPNET Modeler software.

The rest of the paper is organized as follows. In Section 2 we provide an overview of related work, followed by introduction of GeoAODV protocol in Section 3. We describe our implementation of LAR and GeoAODV protocols in Section 4. Description of the simulation study and analysis of collected results are presented in Sections 5 and 6, respectively. The paper concludes and presents the plans for the future work in Section 7.

2. RELATED WORK OVERVIEW

2.1. Ad-hoc On-demand Distance Vector (AODV)

As the name implies, the Ad-hoc On-demand Distance Vector (AODV) routing protocol is an on-demand protocol that begins a route discovery process only when source has data to send but does not know a route to destination. There are two main parts of the AODV protocol: route discovery and route maintenance. We are primarily interested in the route discovery phase that performs network-wide flooding to discover a path to destination. The route maintenance part of the AODV protocol deals with the removal of the outdated or broken path entries from the routing table [1-3]. However, this part of AODV has little or no influence on the route discovery process and thus is not consider in this study.

The AODV route discovery phase is conducted as follows. The sources node, also referred to as an originator, initiates the process by broadcasting the Route Request (RREQ) packet. The RREQ packet is re-broadcast further by the intermediate nodes until it reaches the destination node or a node that knows a path to destination. The request packet header carries such information as the RREQ packet ID, the originator IP address, the originator sequence number, the destination IP address, the destination sequence number, and others. As RREQ travels through the network, the intermediate nodes update/build their routing tables by recording an id of the hop from which RREQ arrived, along with the originator's IP address and sequence number. These routing table entries in the intermediate nodes form a reverse path to the originator node. The intermediate nodes also keep track of recently rebroadcast RREQ packets, by recording the originator IP address, the originator sequence number, and the RREQ ID field values. The intermediate nodes identify and discard all duplicate and outdated request packets. The duplicate packets are identified via the originator IP address and the RREQ packet ID, while outdated RREQs are identified via the originator's IP address and the sequence number [1-3].

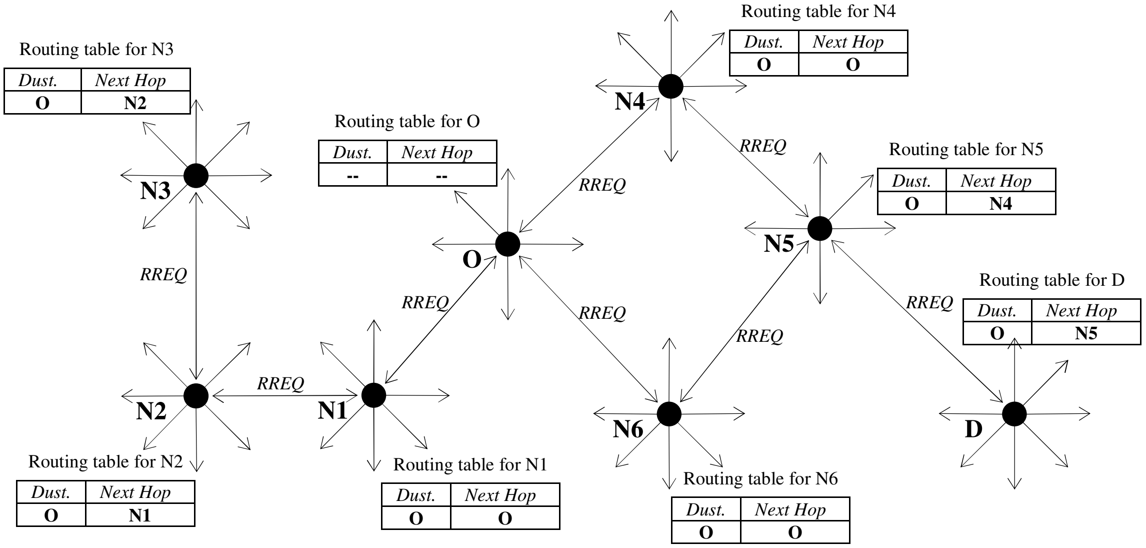
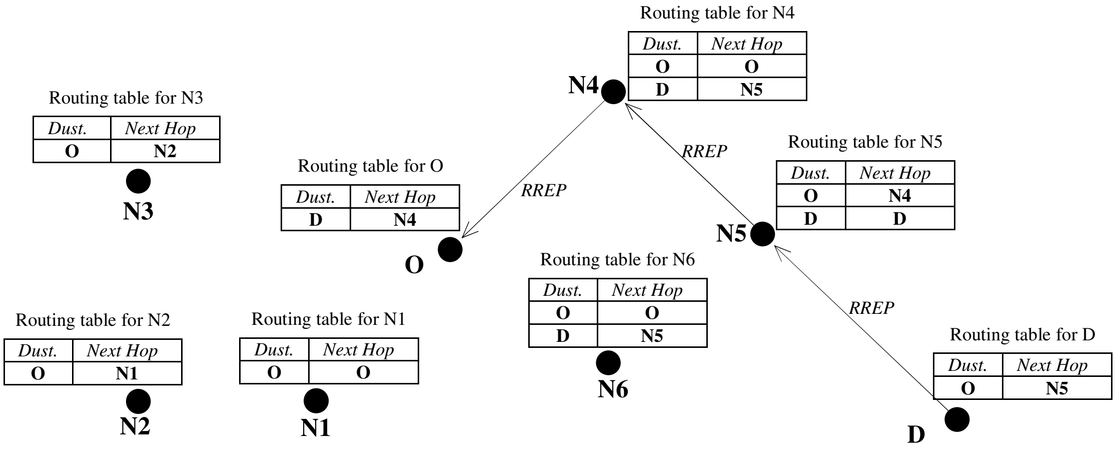


Figure 1. Establishing path to originator during the RREQ flooding

The AODV sequence numbers represent the "freshness" of information and are used to identify the outdated RREQ packets and to find "newer" routes to destination. When an intermediate node that already has a route to destination, receives the RREQ packet, it compares the destination sequence number recorded in its routing table with that carried in the RREQ packet header. If the routing table's destination sequence number is greater than that retrieved from the RREQ packet, then an intermediate node has a "fresh" route to destination, which can be advertised in the network. In this case, the intermediate node generates the Route Reply (RREP) packet and sends it to originator. Similarly, the Route Reply (RREP) packet is also sent when the RREQ packet arrives at the destination node. The RREP packets are unicast back to the originator node using the reverse route recorded during the RREQ flooding.

As RREP travels through the network, the nodes update their routing tables by recording the IP address of the node from which RREP arrived. This process creates a forward path from originator to destination. The destination sequence number carried in each RREP packet is also recorded in the routing table. The destination sequence number is used to check if another RREP that arrives at the node carries a fresher route to destination. Specifically, if the node receives the RREP packet with the value of destination sequence number higher than that recorded in its routing table then the node updates its routing table entry by replacing the destination sequence number with the value carried in the RREP packet and setting the next hop field to the id of the node that sent RREP. Route discovery completes when the originator node receives the RREP packet and starts transmitting data [1-3].

Figures 1 and 2 illustrate the AODV route discovery process in which node **O** attempts to discover a route to node **D**. As shown in Figure 1, when node **O** initiates route discovery and floods the network with the RREQ packets, all the nodes receive generated RREQ and update their routing tables. For example, when node **N2** receives the RREQ packet originated by node **O** and rebroadcasted by node **N1**, **N2** adds an entry into its routing table recording that to reach **O** the data should be forwarded to **N1**. Please note that when **N2** rebroadcasts RREQ, node **N1** will receive that RREQ but will discard it as a duplicate. When destination node **D** receives the RREQ packet it unicasts RREP back to **O**. However, since MANET is a broadcast environment, the neighbors of node **D** and of all the intermediate nodes on the path to **O** will hear the RREP packet and will update their routing tables accordingly. For example, when node **N6** (which is not part of the path between **O** and **D**) overhears the RREP message forwarded by node **N5**, it adds a routing table entry which states that to reach **D** the data has to be forwarded to **N5**.



Routing table for N4

Dust.	*Next Hop*
O	**O**
D	**N5**

Routing table for N3

Dust.	*Next Hop*
O	**N2**

Routing table for O

Dust.	*Next Hop*
D	**N4**

Routing table for N5

Dust.	*Next Hop*
O	**N4**
D	**D**

Routing table for N6

Dust.	*Next Hop*
O	**O**
D	**N5**

Routing table for N2

Dust.	*Next Hop*
O	**N1**

Routing table for N1

Dust.	*Next Hop*
O	**O**

Routing table for D

Dust.	*Next Hop*
O	**N5**

Figure 2. Establishing path to destination via RREP

To reduce the negative effects of network-wide flooding, AODV relies on the expanding ring search mechanism, where the scope of the flooding is controlled by varying the value of the Time-to-Live (TTL) field in the IP header. The originator node first attempts to find a route to destination by setting the TTL field to some initial value, typically 1. If originator does not receive a RREP message within certain amount of time then it assumes that a route to destination was not found and it repeats the process again using large TTL value. This process continues until a route to destination is found or the route discovery process with the TTL field in the RREQ packet set to a certain maximum value times-out (i.e., fails to find a route to destination) [1-3].

2.2. Location-Aided Routing (LAR)

The Location-Aided Routing (LAR) protocol [7-8] is an extension of AODV, which attempts to limit the flooding area during the route discovery process. LAR assumes that all the nodes know the Global Positioning System (GPS) locations and the travelling velocities of all the other nodes in the network. LAR relies on this information to identify a portion of the network which is likely to contain a path to destination. Only the nodes that belong to the identified part of the network participate in route discovery and rebroadcast the RREQ messages.

2.2.1 LAR Zone

There are two primary variations of the LAR protocol which we will refer to as *LAR zone* and *LAR distance*. LAR zone computes the area of the network where the destination node is likely to be located using the last know GPS coordinates and the traveling velocity of that destination node. This area is called *the expected zone* and it is defined as a circle with radius R, centered in the last-known GPS location of the destination node recorded at certain time t_0. The value of radius **R** is computed according to equation (1), where v is the last-known traveling speed of destination and t_1 is the current time:

$$R = v \times (t_1 - t_0) \qquad (1)$$

Next, LAR zone computes the area, called *the request zone*, which is likely to contain a path to destination. LAR zone defines the request zone as a smallest rectangle that encompasses the expected zone so that the sides of the rectangle are parallel to the X and Y axes. During route discovery, the LAR zone protocol confines the RREQ flooding to the request zone, i.e., only the nodes within the request zone rebroadcast the RREQ messages. Figure 3 illustrates possible arrangements of the expected and request zones in the LAR zone protocol.

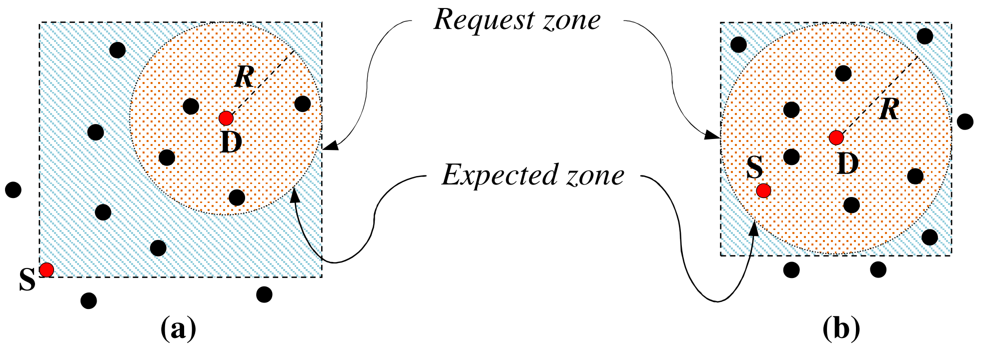


Figure 3. Arrangement of the expected and request zones of the LAR zone protocol:
(a) source is located <u>outside</u> of the destination's expected zone
(b) source is located <u>inside</u> of the destination's expected zone

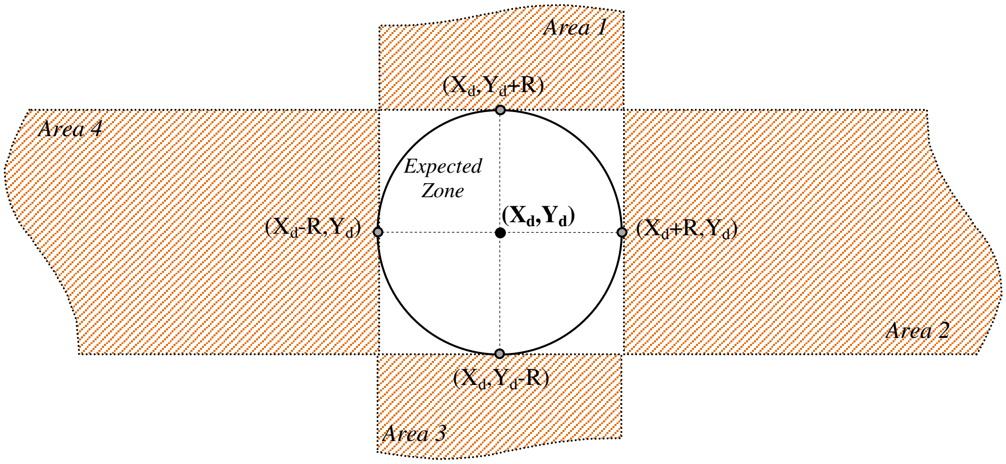


Figure 5. Special Cases for the definition of the request zone in LAR zone scheme

While studying the LAR protocol, we discovered several special cases, not cover in the literature, which pertain to the request zone definition. To simplify the explanation we use (X_s, Y_s) and (X_d, Y_d) to denote the source and destination node coordinates, respectively and R to denote the radius of the expected zone. Imagine the expected zone circle surrounded by a square with the sides of length R that parallel to X and Y axes. Now imagine that each side of that square is a part of a rectangular area that extends to infinity in the direction away from the expected zone circle. Please note that all the sides of these four rectangles are parallel to X and Y axes. Figure 5 illustrates such situation and identifies the rectangular areas as *Area 1*, *Area 2*, *Area 3*, and *Area 4*. We observed that if the source node is located in any of the rectangular areas 1 – 4, then the request zone, as defined in the LAR protocol, is be unable to cover all of the expected zone area, i.e., it is impossible to create the requested zone rectangle with the sides tangent to the expected zone circle and parallel to Y and Y axes if the source node is located in any of the rectangular areas 1 – 4. This situation occurs if the X, Y coordinates of the source node satisfy one of the following rules:

- $X_s \in\ (X_d - R, X_d + R)\ and\ Y_s > Y_d + R$ – the source node is located in *Area 1*
- $X_s > X_d + R\ and\ Y_s \in (Y_d - R, Y_d - R)$ – the source node is located in *Area 2*
- $X_s \in\ (X_d - R, X_d + R)\ and\ Y_s < Y_d - R$ – the source node is located in *Area 3*
- $X_s < X_d - R\ and\ Y_s \in (Y_d - R, Y_d - R)$ – the source node is located in *Area 4*

To resolve this issue we extended the definition of the request zone to include a portions of the expected zone that otherwise is left out. Specifically, we specified the new (X, Y) coordinates of the lower-left and upper-right corners of the redefined request zone as shown in equations (2) and (3):

$$Lower - left\ corner: (\min(X_d - R, X_s), \min(Y_d - R, Y_s)) \qquad (2)$$

$$Upper - right\ corner: (\max(X_d + R, X_s), \max(Y_d + R, Y_s)) \qquad (3)$$

2.2.2 LAR Distance

The LAR distance variation of the LAR protocol does not require the knowledge of the node velocities as it does not compute the request zone area. Instead, LAR distance only assumes that the GPS location of the destination node is known network-wide and then relies on this information to limit the RREQ rebroadcast to only those nodes that are closer to destination than the node which forwarded the RREQ packet [7-8]. This rule is generalized through equation 4, where α and β are configuration parameters, **D** is the destination node, $\mathbf{N_0}$ is the node that broadcasts RREQ, $\mathbf{N_1}$ is the node that receives RREQ from $\mathbf{N_0}$, and **|A B|** denotes the distance between nodes **A** and **B**.

$$\alpha \times |N_0 D| + \beta \geq |N_1 D| \qquad (4)$$

Figure 6 illustrates an example of the LAR distance protocol operation. Suppose the source node **S** initiates the route discovery process by broadcasting the RREQ packet. At some point, node $\mathbf{N_0}$ receives this RREQ and rebroadcasts it farther. When node $\mathbf{N_1}$ receives RREQ from node $\mathbf{N_0}$ it participates in route discovery and rebroadcasts the packet again since $\mathbf{N_1}$ is located closer to destination than node $\mathbf{N_0}$, i.e., $\mathbf{|N_1\, D| \leq |N_0\, D|}$. However, nodes $\mathbf{N_2}$ and $\mathbf{N_3}$ discard the RREQ packet received from $\mathbf{N_0}$ because both of these nodes are located father away from destination **D** than node $\mathbf{N_0}$, i.e., $\mathbf{|N_2\, D| > |N_0\, D|}$ and $\mathbf{|N_3\, D| > |N_0\, D|}$.

Even though the LAR protocols achieve the goal of reducing the control traffic overhead during the route discovery process, they both suffer from two deficiencies. First, the LAR protocols assume a network-wide availability of GPS coordinates, which often is not the case. Second, both protocols do not account for a possibility that a route to destination cannot be found even though it does exist, i.e., in the case of LAR zone the route could be located outside the request zone and in the case of LAR distance a portion of the route could require traveling away from destination. Our approach, called Geographical AODV, attempts to address these issues by only assuming that the nodes know their own GPS coordinates (the coordinates of other nodes in the network are dynamically distributed during the route discovery process) and by expanding the request zone area and re-starting the route discovery process again if a path to destination was not found using a smaller request zone area.

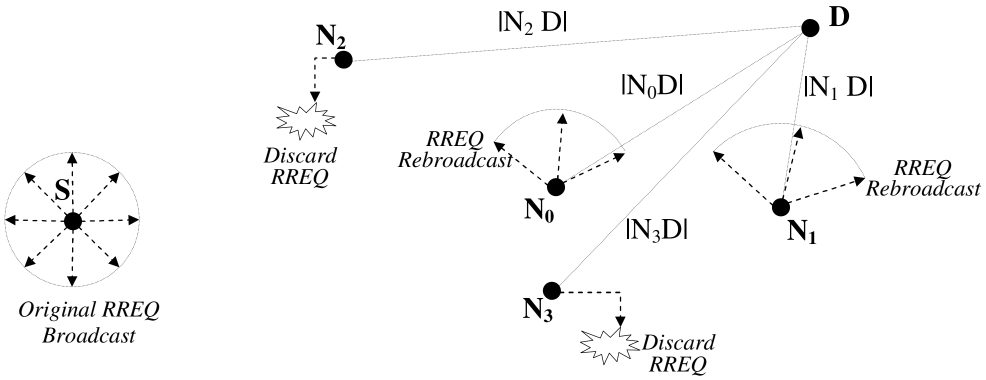


Figure 6. Example of the LAR distance protocol operation

3. Geographical AODV (GeoAODV)

Geographical AODV (GeoAODV) [4-6] is our proposed improvement of LAR zone protocol: it defines the request zone differently than LAR zone, it allows the request zone area to expand after route discovery failures, and it dynamically distributes the node coordinates during route discovery. In this section we introduce the GeoAODV protocol and describe its operation in detail.

3.1 GeoAODV Protocol

GeoAODV defines the request zone in a shape of an isosceles triangle, where the source node is the vertex of the triangle, located in the top corner opposite to the base (i.e., the source node is the origin point of the equal sides of the triangle). The destination node is located on the line that originates at the source node and is perpendicular to the base of the triangle. The width of the GeoAODV request zone (i.e., the isosceles triangle) is controlled via the angle between the equal sides. We refer to this protocol configuration parameter as *the flooding angle* and denote it as α. Only the nodes located within the confines of the GeoAODV request zone participate in route discovery. All the other nodes discard arriving RREQs. Intermediate node **N** computes angle θ formed between the source node, itself, and destination (as shown in Figure 7) to determine if it belongs to the request zone. Since the source-destination vector always divides flooding angle evenly, node **N** belongs to the request zone if angle θ is not larger than one half of flooding angle α.

$$\theta \leq {}^{1}/_{2} \times \alpha \qquad (5)$$

Thus, if inequality (5) holds then node **N** is located within the request zone and will rebroadcast the RREQ packet, otherwise **N** is outside of the request zone and RREQ is discarded. Generally we compute the value of angle θ according to equation (6), where we use $\overrightarrow{SD}$ to denote a vector between source node **S** and destination node **D**, $\overrightarrow{SN}$ to denote vector between source node **S** and node **N**, while $|SD|$ and $|SN|$ are the absolute values of vectors $\overrightarrow{SD}$ and $\overrightarrow{SN}$, respectively.

$$\theta = \cos^{-1}\left(\frac{\overrightarrow{SD}\cdot\overrightarrow{SN}}{|SD|\times|SN|}\right) \qquad (6)$$

At the start of the route discovery process GeoAODV sets flooding angle α to some initial value. This initial value could be determined by the “freshness” of the destination’s GPS coordinates, i.e., the value of α increases proportionally to t_d, the time passed since the last update of the destination’s location information. Once t_d crosses certain threshold, α is set to 360 degrees and GeoAODV performs the same way as regular AODV. Alternatively, the initial value of the flooding angle could be a function of the expected zone radius defined in the equation (1).

If an attempt to find a route to destination using certain value of the flooding angle fails (i.e., a time out occurs) then GeoAODV repeats route discovery again using a larger value of the flooding angle, which effectively expands the size of the request zone. GeoAODV continues this process until a path to destination is found or until GeoAODV searched the whole network (i.e., route discovery with flooding angle value of 360 degrees failed to find a route to destination). Please note that when the flooding angle value reaches 360 degrees, GeoAODV operates exactly the same way like the AODV protocol. Since GeoAODV may eventually search the whole network, it guarantees that a route to destination will be found if one exists [5-6].

Figure 7 illustrates an example of the GeoAODV protocol operation where source node **S** initiates the route discovery process in an attempt to find a path to destination node **D**. Initially, **S** uses the flooding angle with the value $\boldsymbol{\alpha_1}$. The request zone defined by $\boldsymbol{\alpha_1}$ is shown in Figure 7 as an isosceles triangle of a lighter grey color. During this round of route discovery only intermediate node **N_1** rebroadcasts the RREQ packets, while the remaining nodes are outside the request zone defined by **α_1** and are excluded from participating in route discovery. These nodes (i.e., **N_2**, **N_3**, and **N_4**) discard all arriving RREQs during this initial round of route discovery. If the first round of route discovery fails, then source increases the flooding angle to some new value **α_2** and repeats the process again. During the second round of route discovery, the request zone is extended (i.e., shown in Figure 7 as a darker color isosceles triangle) and intermediate nodes **N_1**, **N_2**, and **N_3** rebroadcast RREQs, while intermediate node **N_4** discards all arriving RREQ packets since it is located outside the request zone defined by flooding angle **α_2.**

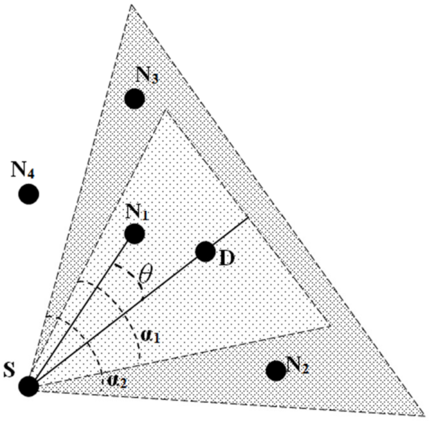


Figure 7. Example of GeoAODV operation

3.2 GeoAODV Rotate

In our study we considered two variations of the GeoAODV protocol: *GeoAODV static* and *GeoAODV rotate*. GeoAODV static operates as described above: the request zone remains unchanged during each round of route discovery, i.e., the source node is always a vertex opposite to the base of the isosceles triangle. GeoAODV rotate operates a bit differently. It re-orients the request zone towards destination at each intermediate node by making the previous node a new vertex of the triangle, i.e., each intermediate node re-computes the request zone based on the location of the previous hop, instead of the source node. Figure 8 illustrates the idea of GeoAODV rotate: node **N_1** belongs to the request zone computed based on location of node **S** while node **N_2** belongs to the new, re-oriented request zone computed based on the location of node **N_1**. Both nodes **N_1** and **N_2** participate in route discovery even though they belong to different request zones. On the other hand, node **N_3**, which receives RREQ from **N_1**, will not participate in route discovery because node **N_3** is located outside the new request zone computed based on the location of **N_1**, its previous hop. On the other hand, when GeoAODV static is used, all the node in the Figure 8 will participate in route discovery because they all belong to the request zone computed based on location of source node **S**.

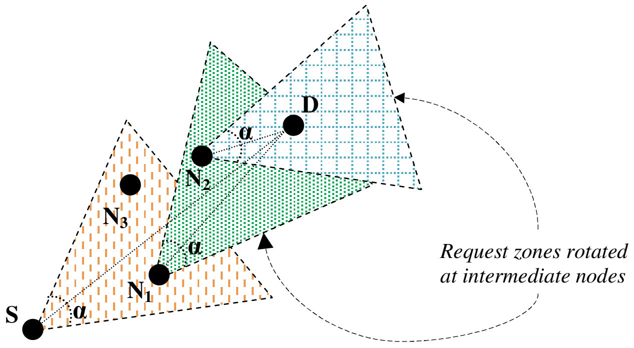


Figure 8. Example of GeoAODV rotate operation

3.3 Distributing Location Information in GeoAODV

GeoAODV assumes that the nodes use GPS to find their own coordinates, while the locations of all the other nodes is dynamically distributed in the network during route discovery. Each GeoAODV node stores the node location data in a structure we call *the geo-table*. An entry in the geo-table consists of the location information (e.g., GPS coordinates), the “freshness” timer, the AODV sequence number, and the identity of the node (e.g., IP address). The “freshness” timer is keeps track of when the node coordinates were updated the last time, while the AODV sequence number allows an intermediate node to identify if the arriving control packet (e.g., RREQ or RREP) carries new location information. This process is similar to the way the AODV protocol differentiates between new and old control packets. However, the geo-table entries remain valid for longer periods of time than the entries in the AODV routing table because the location information can help determine the general direction where the destination node may be located even if a route to that destination has changed [5-6].

We modified format of the RREQ and RREP packets to carry additional information such as locations of the source and destination nodes, and the flooding angle. This information is used to populate the geo-table and to determine if an intermediate node should participate in route discovery. At the start of the route discovery process the source node consults it geo-table and generates a RREQ packet that will carry node’s own location information, the initial value of the flooding angle, and the last known location of the destination node. If the source node does not contain a geo-table entry for the destination node then the flooding angle is set to 360 degrees and GeoAODV operates the same way as regular AODV.

Upon the RREQ message arrival, the nodes (even those that will discard the RREQ packet) update their geo-tables with the source node’s location information. An intermediate node updates its geo-table with the destination’s location information only if the destination sequence number carried in RREQ is larger than that stored in its geo-table. Otherwise, an intermediate node may update the destination coordinates carried in the RREQ packet. Similar processing occurs when RREP is sent back: each intermediate node updates its geo-table with the source and destination location information carried in the packet. GeoAODV also utilizes periodic AODV hello messages (which have the same header format as the RREQ packets) to distribute location information among the neighboring nodes [5-6].

4. GeoAODV Implementation in OPNET Modeler

We used OPNET Modeler [15] version 16.0 to implement the LAR and GeoAODV protocols. OPNET Modeler is the leading industrial software for modeling computer networks. It employs layered architecture which consists of the process, node, network, and simulation domains. The process domain allows the developer to model behavior of various processes such as applications, network protocols, physical environments, etc. This domain includes such items as external C/C++ code files and various process models. A process model consists of the finite state machine that describes the modeled process, implementation of various actions using C or C++ programming language, and various configuration parameters. At the node domain layer individual process models are combined together to create the node models of various networking devices such as servers, switches, routers, WLAN devices, etc. Figure 9 shows OPNET node model of MANET station [16].

The network domain combines individual node and link models to create and configure representation of simulated network. At the network domain level the developer can specify various characteristics of the network such as the types, configuration, and location of individual nodes and links, the traffic generation sources, the traveling speed of individual nodes, parameter settings for various network protocols, etc. The simulation domain is responsible for configuring simulation model as a whole and for specifying the values of various parameters such as the duration of the simulation, statistics to be collected during the simulation, the seed value for the random number generator, etc.

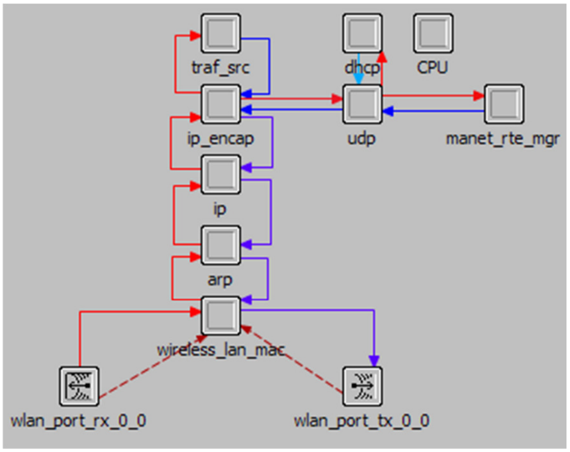


Figure 9. OPNET node model of MANET station

To implement the LAR and GeoAODV protocols we modified OPNET *aodv_rte* process model and several external C files including *manet_support*, *aodv_suport*, *aodv_pkt_support*, *aodv_request_table*, *aodv_route_table*, and *aodv_packet_queue* all of which are responsible for modeling the AODV protocol. Figure 10 illustrates *aodv_rte* process model. The finite state machine representation of the AODV protocol consists of two states: *init* which initializes state variables, configuration parameters, supporting data structures, etc., and *wait* state which models operation of the AODV protocol. Wait state is responsible for processing various arriving AODV control packets, generating new AODV control packets and preparing them departure, forwarding and queuing (if necessary) the data packets, and generating periodic AODV hello messages. Upon the packet arrival the process model identifies the type of an incoming packet (e.g., data, RREQ, RREP, etc) and then calls the corresponding function that handles processing of the packets of identified type.

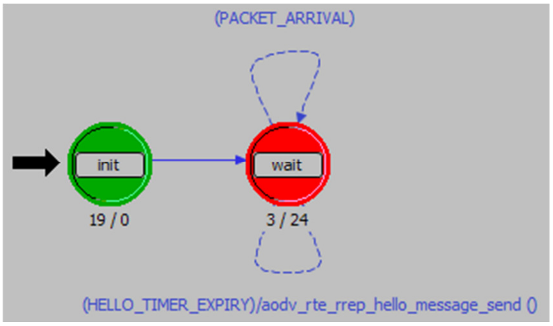


Figure 10. OPNET Modeler's *aodv_rte* process model

When modifying OPNET's implementation of the AODV protocol, we first added several parameters to simplify configuration of the LAR and GeoAODV protocols. In OPNET Modeler, configuration parameters for MANET routing protocols are defined via *manet_mgr* process model. However, the parameter values are parsed in the corresponding routing protocol process models, such as *aodv_rte*. Next, we modified the structure of the RREQ and RREP headers, defined in *aodv_pkt_support.ex.h* file, to carry additional information. After that we created several new data structures to store location information. We created a simulation-wide table, which we refer to as *global_location_table,* which helps us to model the network-wide availability of geographical coordinates and velocities of the nodes in the LAR protocol. This table is periodically updated by individual nodes that store their precise location information and traveling speed. The frequency of these periodic updates is controlled via configuration parameter which by default is set to 1 second. During the LAR route discovery process individual nodes consult this table to retrieve location information about other nodes in the network. We also implemented the geo-table which stores such information as the IP address, the insertion time, location coordinates, and the sequence number needed for the GeoAODV protocol. The information stored in a node's geo-table is only accessible by that node itself and is not available network-wide.

Finally we modified *aodv_rte* process model and several external files to implement the LAR and GeoAODV route discovery processes. Specifically, we added the code responsible for

- processing arriving control packets,
- generating the RREQ, RREP, and HELLO messages using new GeoAODV format,
- determining if a node should retransmit an RREQ packet based on LAR zone, LAR distance, GeoOADV static, or GeoADOV rotate approach, and
- updating location information in *global_location_table* and *geo-table* data structures.

We verified the accuracy of our implementation by conducting a line-by-line code trace under various configuration settings.

5. SIMULATION SET-UP

We compared the performance of the AODV, LAR, and GeoAODV protocols in the network environment that consisted of 50 MANET node randomly placed within a 1500 meters x 1500 meters network domain area. In this study we varied the following two parameters: the number of nodes that generate traffic and the node traveling velocities. Specifically, we conducted simulation studies with 2, 5, 10, 20, and 30 communicating nodes, where all the source-destination pairs were selected randomly. Furthermore, each simulation set with a different number of communication nodes was also examined in the environment where the nodes were traveling with the following velocities: 0 meters/second (all the nodes were stationary), 5 meters/second, 10 meters/seconds speed, and a random value computed using the uniform distribution function with the outcome between 0 and 20 meters/second. All the nodes in the network were traveling within the confines of the simulated network domain according to the Random Waypoint model. The pause time between two consecutive legs of the node's journey was computed according to the exponential distribution function with the mean outcome of 10 seconds.

TABLE 1. Summary of Node Configuration

Configuration Parameter	**Value**
Channel Data Rate	*11 Mbps*
Transmit Power	*0.0005 Watts*
Packet Reception Power Threshold	*-95 dBm*
Start of data transmission	*normal (100, 5) seconds*
End of data transmission	*End of simulation*
Duration of Simulation	*300 seconds*
Packet inter-arrival time	*exponential (1) second*
Packet size	*exponential (1024) bytes*
Mobility model	*Random Waypoint*
Pause Time	*exponential(10)*
Destination	*Random*

We configured the LAR protocol to have the nodes publish their location and traveling velocities in *global_location_table* once every second. We set the values of configuration parameters α and β for the LAR distance protocol to 1 and 0, respectively. Furthermore, we modified the LAR protocols to operate like regular AODV if they fail to find a path to destination after the initial round of route discovery, i.e., if the LAR protocol does not find a route to destination then it tries again using the AODV protocol. We configured the GeoAODV protocol to have the initial value of the flooding angle set to 90 degrees if the destination coordinates are known and to 360 otherwise. The value of the flooding angle was incremented by 90 degrees after each failed attempt to find a route to destination. The wireless LAN configuration parameters of each node were set to their default values as defined in OPNET Modeler. The summary of simulation set-up is provided in Table I.

We executed a total of 100 different simulation scenarios (i.e., 5 different values for the number of communicating nodes and 4 different traveling velocity values). Each simulation scenario was executed for 300 seconds with the communicating nodes starting data transmission at simulation time of 100 seconds. Each scenario was executed six times using different seed value for the random number generator. We executed a total of 600 simulation runs, which took over 72 hours to complete. Collected results from this study were exported into a comma separated text file and then processed using a Python programming language script.

6. RESULTS

It should be noted that in this simulation study we made few simplifying assumptions. Specifically, we did not account for the delay associated for the retrieval of GPS coordinates; in our evaluation of the GeoAODV protocols we disregarded the overhead introduces by addition of new fields in the RREQ and RREP headers; and in the LAR protocols we made an assumption that location information and traveling velocities are available everywhere in the network at no additional cost. Our study primarily focused on the total amount of control traffic generated by each of the examined protocols. The results of this study suggest that all location-aided routing protocols outperform AODV by generating significantly fewer control packets during route discovery. The summary of collected results is in presented in Figures 11 – 15 which illustrate the total number of control packets (i.e., the number of RREP + RREP) generated by each protocol versus the node traveling speed.

The results show that the LAR zone protocol consistently generates the smallest number of control packets while GeoAODV rotate is a close second. This can be attributed to the fact that the simulation does not account for the cost associated with retrieval of node coordinates and traveling speeds in LAR zone (i.e., these values are assumed to be available as needed). GeoAODV on the other hand makes no such assumption, and distributes location information during route discovery. Additionally, before location-aided improvements of GeoAODV kick-in, the nodes require some time to gather location information about other nodes in the network. As a result, initially, GeoAODV operates the same way as regular AODV, and only after location information has been distributed in the network it can take advantage of the limited flooding. Furthermore, GeoAODV may have to go through up-to 3 rounds of route discovery using different values of the flooding angle before conducting route discovery using the AODV protocol. The LAR protocols, on the other hand, revert to AODV after a single failure.

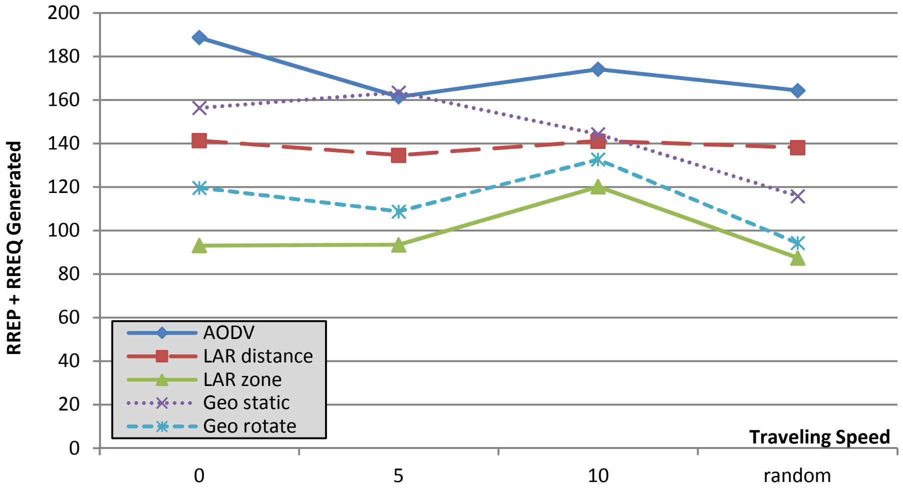


Figure 11. The control traffic overhead in scenarios with 2 communicating nodes

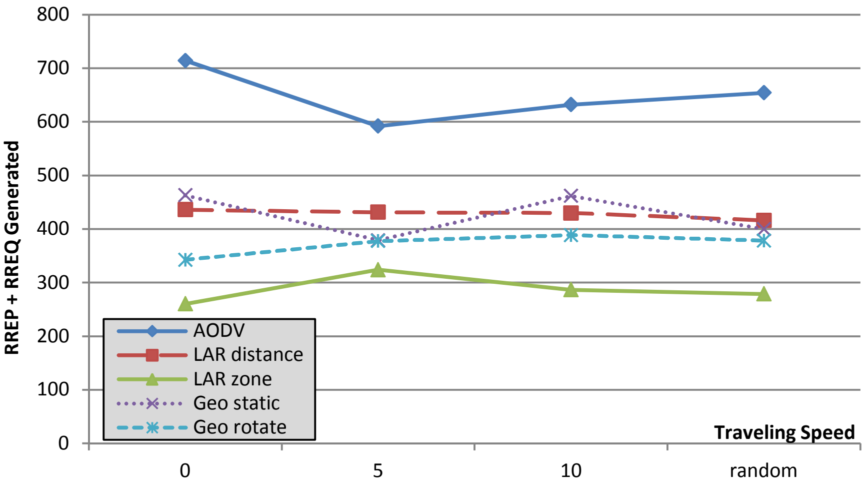


Figure 12. The control traffic overhead in scenarios with 5 communicating nodes

The difference in performance between two LAR protocols can be attributed to how often the protocols do not find a route and are forced to conduct route discovery as AODV. The search area of the LAR distance protocol is very limited: the current node rebroadcasts the RREQ message only if it is closer to destination than the previous node. Thus, it is not surprising that LAR distance fails to find a route to destination more frequently than LAR zone. As result, LAR distance often behaves like AODV and generates a large number of control packets. LAR zone, on the other hand, conducts route discovery over a wide area and thus is less likely to revert to AODV. This results in LAR zone consistently outperforming the LAR distance protocol.

When comparing the GeoAODV protocols we observed that GeoAODV rotate consistently outperforms GeoAODV static. While both variations of GeoAODV fail to find a route to destination roughly the same number of times, GeoAODV rotate dynamically reorients the direction of the request zone and thus excludes the nodes that likely are not a part of a route to destination. This results in GeoAODV rotate forwarding fewer RREQ packets through the network and thus introducing lower control traffic overhead than the GeoAODV static protocol.

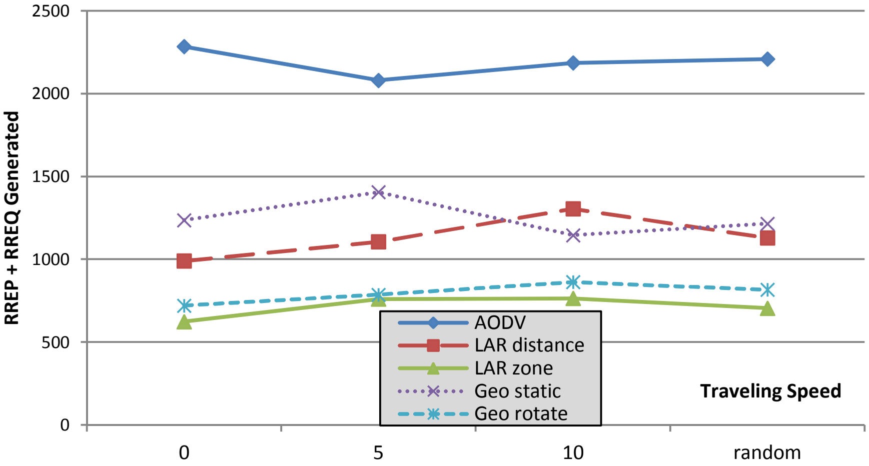


Figure 13. The The control traffic overhead in scenarios with 10 communicating nodes

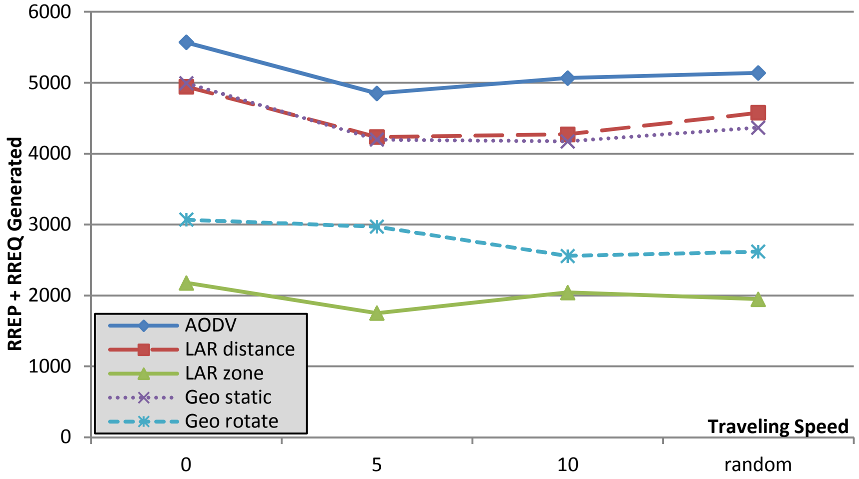


Figure 14. The control traffic overhead in scenarios with 20 communicating nodes

As expected, collected results report that the control traffic overhead increases with the increase in the number of communicating nodes. What was surprising is how all the protocols, except for LAR zone, performed in the simulation scenario with 30 communication nodes. As Figure 15 shows, LAR distance, GeoAODV static, and GeoAODV rotate generated almost the same amount of control traffic as AODV. Such behavior could be attributed to the fact what when there are many communicating nodes, the chance of failing to find a route using limited broadcast increase causing these protocols to revert to regular AODV more frequently. As a result, all of the advantage gained by successfully employing limited flooding is lost when the protocols fail to find a route and have to conduct a network-wide flooding. LAR zone appears to be less susceptible to this problem and performs the best as shown in Figure 15. Nevertheless, the GeoAODV rotate protocol consistently remains a second-best option, outperforming all the protocols except for LAR zone in all evaluated scenarios.

7. Conclusions and Future Work

This paper introduced a new routing protocol called GeoAODV and presented comprehensive comparison study of the AODV-based location-aided routing protocols. Simulation results suggest that even though GeoAODV rotate does not always reduce the control traffic overhead by as much as LAR zone, it can become a preferred mechanism for route discovery in MANET. The main advantages of GeoAODV are its build-in mechanism for distributing location information and its simplicity of deployment. LAR zone requires the location coordinates and the traveling speed of individual nodes to be readily available in the network, which may not be always possible. However, the LAR zone protocol has an advantage in the environments where location information and node traveling speed is distributed with the help of some additional facilities such as described in [17-21].

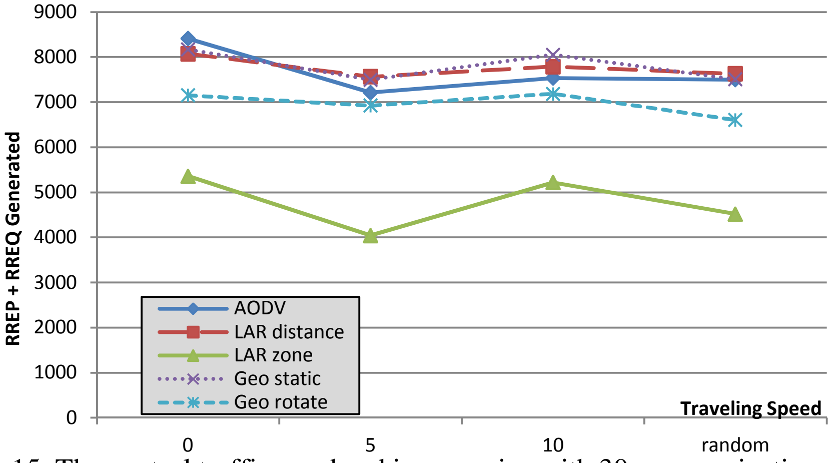


Figure 15. The control traffic overhead in scenarios with 30 communicating nodes

Currently we continue our study focusing on other aspects of protocol performance such as the number of route discovery failures and the time to find a route to destination. We are also studying the accuracy of location information available in the network during GeoAODV route discovery and how that accuracy influences the protocol performance. We are investigating other possible improvements of the GeoAODV protocol including selection of initial value for the flooding angle, dynamically adjusting the flooding angle at intermediate nodes (i.e., increasing the flooding angle value when an intermediate node knows that there are no neighboring nodes within the request zone defined by the flooding angle carried in arriving RREQ), and few others. We are also examining improvements to the LAR protocols which will allow them to increase the search area after the route discovery failure, instead of immediately reverting to AODV. For example, LAR distance can adjust the values of configuration parameters α and β used in (4), while LAR zone can increase the request zone by "advertising" the source node coordinates as if source is located farther away on the line between source and destination than it actually is. Finally, we plan to expand our study by comparing the performance of LAR and GeoAODV with other routing protocols, such as the Greedy Perimeter Stateless Routing (GPSR) protocol [22] and the Geographical Routing Protocol (GRP) [23].

ACKNOWLEDGEMENTS

I would like to thank Hristo Asenov, Remo Cocco, Malik Ahmed, Dan Urbano, Rob Hussey, Earl Huff, and Zabih Shinwari, the students from Rowan University, Computer Science Department, who have been tirelessly working with me on this project over the past several years. Thank you all for your dedication, diligence, and hard work!

REFERENCES

[1] C. Perkins, E. Belding-Royer, S. Das. (July 2003). *Ad hoc On-Demand Distance Vector (AODV) Routing*. IETF RFC 3561. Last accessed on 2013-01-26

[2] E. M. Royer and C. E. Perkins. *An Implementation Study of the AODV Routing Protocol*, Proc. of the IEEE Wireless Communications and Networking Conference, Chicago, IL, September 2000

[3] C. E. Perkins and E. M. Royer. *Ad hoc On-Demand Distance Vector Routing*, Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999, pp. 90-100

[4] V. Hnatyshin, M. Ahmed, R. Cocco, and D. Urbano, *A Comparative Study of Location Aided Routing Protocols for MANET*, Proc. of 4th IEEE IFIP Wireless Days 2011 conference, Niagara Falls, Canada

[5] V. Hnatyshin, and H. Asenov, *Design and Implementation of an OPNET model for simulating GeoAODV MANET routing protocol*, Proc. of the OPNETWORK 2010 International Conference, Session: Wireless Ad Hoc and Wireless Personal Area Networks, Washington DC, August 2010

[6] H. Asenov, and V. Hnatyshin, *GPS-Enhanced AODV routing*, Proc. of the 2009 International Conference on Wireless Networks (ICWN'09), Las Vegas, Nevada, USA (July 13-16, 2009)

[7] Y. Ko and N. H. Vaidya, *Location-aided routing (LAR) in mobile ad hoc networks*, Wireless Networks, 6(4), July 2000, pp. 307-321

[8] Y. Ko and N. H. Vaidya, *Flooding-based geocasting protocols for mobile ad hoc networks*, Mobile Networks and Applications, 7(6), Dec. 2002, pp. 471-480

[9] A. Husain, B. Kumar, A. Doegar, *A Study of Location-Aided Routing (LAR) Protocol for Vehicular Ad Hoc Networks in Highway Scenario*, International Journal of Engineering and Information Technology, 2(2), 2010, pp. 118-124

[10] K.M.E. Defrawy and G. Tsudik, *ALARM: Anonymous Location-Aided Routing in Suspicious MANETs*, Proc. of the IEEE International Conference on Network Protocols, 2007, pp. 304-313

[11] D. Deb, S. B. Roy, N. Chaki, *LACBER: A new Location-Aided routing protocol for GPS scarce MANET*, International Journal of Wireless & Mobile Networks (IJWMN), 1(1), August 2009

[12] F. De Rango, A. Iera, A. Molinaro, S. Marano, *A modified location-aided routing protocol for the reduction of control overhead in ad-hoc wireless networks*, Proc. of the 10th International Conference on Telecommunications, 2003, pp. 1033 – 1037

[13] Y. Xue, B. Li, *A Location-aided Power-aware Routing Protocol in Mobile Ad Hoc Networks*, Proc. of the IEEE Global Telecommunications Conference, San Antonio, TX, November 2001, pp. 2837 – 2841

[14] Y. Wang, L. Dong, T. Liang, X. Yang, X., D. Zhang, *Cluster based location-aided routing protocol for large scale mobile ad hoc networks*, IEICE Transactions, 2009, E92-D(5), pp. 1103–1124

[15] OPNET Modeler ver. 16.0. OPNET Technologies, Inc®, www.opnet.com last visited 2/1/13

[16] Adarshpal S. Sethi and Vasil Y. Hnatyshin, *The Practical OPNET User Guide for Computer Network Simulation*, 527 pages, Chapman and Hall/CRC (August 24, 2012), ISBN-10: 1439812055, ISBN-13: 978-1439812051

[17] H. Cheng, J, Cao, H. Chen and H. Zhang, *GrLS: Group-Based Location Service in Mobile Ad Hoc Networks*, IEEE Transactions on Vehicular Technology, Vol. 57, No. 6, November 2008

[18] Gajurel, S., Heiferling, M., *A Distributed Location Service for MANET Using Swarm Intelligence*, Mobile WiMAX Symposium, 2009. MWS '09. IEEE, pp. 220 – 225

[19] Yongming Xie, Guojun Wang, Jie Wu, *Fuzzy Location Service for Mobile Ad Hoc Networks*, Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on, pp. 289 – 296

[20] S. M. Das , H. Pucha and Y. C. Hu *Performance comparison of scalable location services for geographic ad hoc routing*, Proc. INFOCOM 2005, vol. 2, pp.1228 -1239

[21] K. Wolfgang , F. Holger and W. Jorg *Hierarchical location service for mobile ad hoc networks*, ACM SIGMOBILE Mobile Comput. Commun. Rev. 2004, vol. 8, no. 4, pp. 47 – 58

[22] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in MobiCom ’00: Proceedings of the 6th annual international conference on Mobile computing and networking. New York, NY, USA: ACM Press, August 2000, pp. 243–254.

[23] OPNET Modeler 16.1 Documentation, OPNET Technologies, Inc., 2012.