GRB: Greedy Routing Protocol with Backtracking for Mobile Ad Hoc Networks

Baban A. Mahmood
University of Kirkuk
Kirkuk, Iraq.
and
D. Manivannan
University of Kentucky, Lexington, Kentucky, USA.

Routing protocols for Mobile Ad Hoc Networks (MANETs) have been extensively studied for more than fifteen years. Position-based routing protocols route packets towards the destination using greedy forwarding (i.e., an intermediate node forwards packets to a neighbor that is closer to the destination than itself). Different position-based protocols use different strategies to pick the neighbor to forward the packet. If a node has no neighbor that is closer to the destination than itself, greedy forwarding fails. In this case, we say there is void (no neighboring nodes) in the direction of the destination. Different position-based routing protocols use different methods for dealing with voids. In this paper, we use a simple backtracking technique to deal with voids and design a position-based routing protocol called “Greedy Routing Protocol with Backtracking (GRB)”. We compare the performance of our protocol with the well known Greedy Perimeter Stateless Routing (GPSR) routing and the Ad-Hoc On-demand Distance Vector (AODV) routing protocol as well as the Dynamic Source Routing (DSR) protocol. Our protocol needs much less routing-control packets than those needed by DSR, AODV, and GPSR. Simulation results also show that our protocol has a higher packet-delivery ratio, lower end-to-end delay, and less hop count on average than AODV.

Keywords: MANETs, Routing in MANETs, Geographic routing.

---

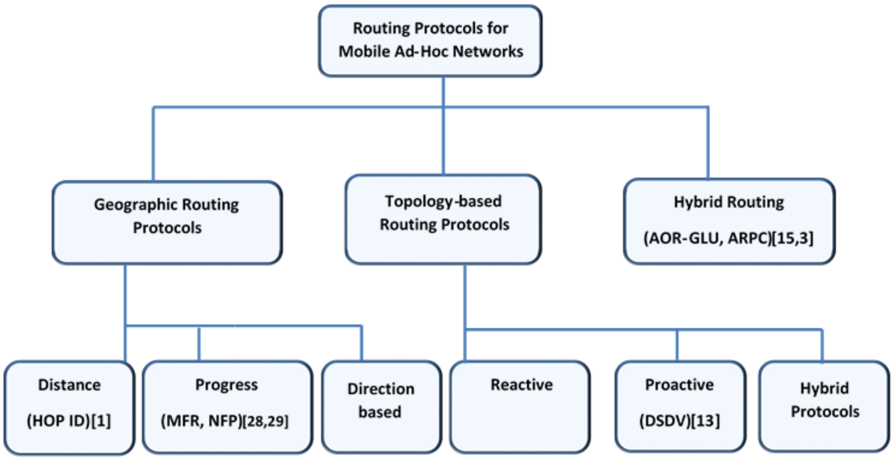
1. INTRODUCTION

A Mobile Ad-hoc Network (MANET) consists of a set of nodes each of which is capable of being both a host and a router. The nodes form a network among themselves without the use of any fixed infrastructure, and communicate with each other by cooperatively forwarding packets on behalf of others. Mobile ad-hoc networks have applications in areas such as military, disaster rescue operations, monitoring animal habitats, etc. where establishing communication infrastructure is not feasible (Li and Singhal [2005], Shen and Zhao [2013], Talooki et al. [2013], (Kavitha et al. [2013], Jain et al. [2013]). Routing protocols designed for mobile ad hoc networks need to be scalable, robust, and have low routing overhead. Routing protocols designed for MANETs can be broadly classified as geographic routing protocols (or position-based routing protocols) and topology-based routing protocols. Figure 1 shows a brief classification of routing protocols for MANETs. In geographic routing protocols, nodes do not maintain information related to network topology (i.e., they are topology independent). They only depend on the location information of nodes to make forwarding decisions. Generally (Lin and Kus [2006]), nodes need their own location, their neighbors' location, and the location of the destination node to which the packet needs to be forwarded. Using this location information, routing is accomplished by forwarding packets hop-by-hop until the destination node is reached (Flury and Wattenhofer [2008]). Greedy forwarding (Karp and Kung [2000]), is one of the main strategies

---

Authors' addresses: Baban A. Mahmood, Computer Science Department, University of Kirkuk, Baghdad St., Kirkuk, Iraq Email: Baban@netlab.uky.edu
D. Manivannan, Department of Computer Science, University of Kentucky, Lexington, KY 40506. Email: mani@cs.uky.edu.

used in geographic routing protocols. Under Greedy forwarding, an intermediate node on the route forwards packets to the next neighbor node that is closer to the destination than itself.



Routing Protocols for MANETs.

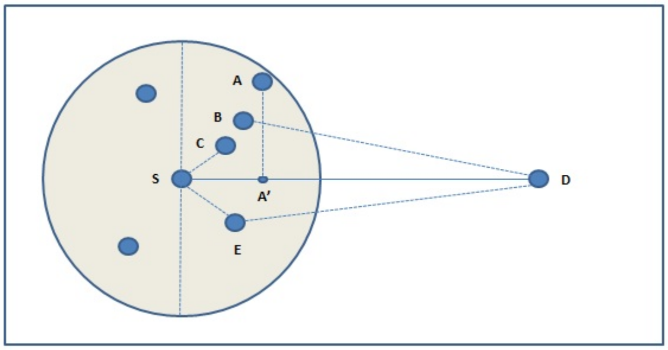
Different protocols use different greedy forwarding strategies which can be defined in terms of distance, progress, and/or direction towards destination nodes.

Distance Strategy. Euclidean distance is mainly used in greedy forwarding to minimize the distance a packet traverses from the source node to the destination node.

Progress Strategy. This strategy tries to maximize the distance between the source node S and the projection A' of a neighboring node A (chosen for forwarding) onto the straight line joining S and the destination node D as shown in Figure 2.

Direction-based Strategy. This strategy is also called compass routing (Mauve et al. [2001]). It minimizes the spatial distance that packets travel by using deviation as its criteria to forward packets. The deviation is the angle between the straight line joining the source node and the next hop and the straight line joining the source node and the destination.

Topology-based routing protocols depend on current topology of the network. Topology-based routing is also known as table-based routing. Topology-based routing can be classified in to proactive routing protocols, reactive routing (on-demand) protocols, and hybrid routing protocols (Li and Singhal [2005]), (Hus and Lei [2009]), (V.Giruka and Singhal [2007]). In proactive protocols, like (DSDV -C.Perkins and Hagwat [1994]), nodes use pre-established table-based routes (Basagni et al. [1998]). Therefore, routes are deemed reliable and nodes do not wait for route discovery which cuts off latency. However, overhead incurred for route construction and maintenance can degrade performance, limit scalability, and the routing table will consume lot of memory as the network size grows. Reactive Routing Protocols are also called on-demand routing protocols wherein senders find and maintain route to a destination only when they need it. Reactive routing needs less memory and storage capacity than proactive protocols. However, in network areas where nodes can move more unpredictably and frequently, path discovery may fail since the path can be long and links may break due to node mobility or when facing other obstacles (Li and Singhal [2005]). The delay caused by route discovery for each data traffic can increase latency. On the other hand, geographic routing protocols require only the location information of nodes for routing. They do not require a node to establish a route to the destination before transmitting packets. Unlike on-demand routing protocols, they do not depend on flooding route request messages to discover routes. This feature helps geographic routing protocols to reduce the extra overhead imposed by topology constraints for route discovery (Zhao et al. [2007], Flury and



Greedy Routing Strategies

Wattenhofer [2008]). A node only needs to know the position of its neighbors and the position of the destination to forward packets. Therefore, geographic routing protocols generally are more scalable than topology based routing protocols (Lemmon et al. [2009; Shobana and Karthik [2013; Cadger et al. [2013]). In spite of the benefits mentioned above, geographic routing protocols have the following limitations: Greedy forwarding, the primary packet forwarding strategy used by geographic routing protocols, may fail in low density networks, networks with non-uniformly distributed nodes, and/or networks where obstacles can be present. Moreover, Location Service is required to obtain location information of destination nodes which may result in high overhead. The non-hierarchical address structure used in ad-hoc networks requires more control overhead to update node location(Lemmon et al. [2009]). The rest of the paper is organized as follows. In Sect. 2, we discuss the related work and paper objectives. In Sect. 3, we present our protocol. In Sect. 4, we present the performance evaluation results of our protocol. In Sect. 5 we give a brief discussion of our protocol. Sect. 6 concludes the paper.

2. RELATED WORK

In this section, we discuss the basic idea behind several of the recently proposed geographic (position-based) routing protocols and on demand routing protocols bringing out their week and strong points.

2.1 Position-based Routing Protocols

GPSR (Karp and Kung [2000]), a well known geographic routing protocol proposed by Karp and Kung, uses greedy forwarding as the default forwarding strategy. When a packet confronts a void (i.e., when greedy forwarding fails), they planarize the local topology graph either by constructing the Relative Neighborhood Graph (RNG) or Gabriel Graph (GG) of the graph and use those graphs to route around the void. However, constructing the graph could be time consuming. Zhao et al. (Zhao et al. [2007]), proposed a routing protocol called HIR. This protocol selects specific nodes as landmarks and builds a multidimensional coordinate system based on which they find the Hop ID distance between each pair of nodes in the network. When a packet meets with a void, the protocol switches to a landmark-guided, detour routing which tries to forward the packet to the landmark nearest to the destination. This protocol does not scale well due to the central packet flooding technique used by a designated node to select LANDMARKs. Nodes proportionally exchange high amount of control information to select LANDMARKs. However, since this happens only to select LANDMARKs, normal data packet forwarding process does not produce additional overhead to the protocol. On the other hand, the level of stability of LANDMARK nodes plays a significant role in the control information overhead.

Li and Singhal (Li and Singhal [2005]), proposed ARPC in which the routing process is divided into several parts. First, Location-based Clustering Protocol in which several physical locations are assumed to be known in the network area which are called anchors and they have coordinates.

Second, Inter-cell Routing Protocol which lets every node maintain a dynamic routing table that contains routes to its neighboring cells. Third, Intra-cell Routing Protocol which is an on-demand routing technique that is performed inside the same cell. Fourth, Data Packet Routing which deals with how the data packets are routed from a source to a destination node. ARPC is less scalable since nodes use routing tables which can become large as the network size increases. Moreover, the announcement packets sent by agent nodes to announce their existence to the nodes in the cell they reside can cause large overhead as the node density increases.

Lin and Kus (Lin and Kus [2006]) proposed LGR which is a location-fault-tolerant geographic routing protocol that uses both traditional geographic routing and position-based clustering technique. Routing is performed using global geographic routing and local gradient routing. A cluster head (CH) broadcasts messages to all the nodes in its own cluster as well as all nodes in its neighboring clusters so that each of these nodes can establish a routing path to the CH. This process obviously affects the scalability. If nodes are highly mobile, frequent CH election occurs which results in more message broadcasting by newly elected CHs to announce their existence which overwhelms nodes with control packets exchanged between nodes and hence limits scalability. Yau et al. (Yau et al. [2006]) proposed a Location-based Directional Route Discovery protocol (LDRD). In LDRD, the route discovery process is confined to designated areas within which the route discovery is performed using directional route requests. The authors presented an adaptable location service protocol named Node Location Service based on Location Coordinates (NLS-LC). They assume that nodes are either stationary or moving at different velocities and they have knowledge about the range of the nodes' moving speeds. Nodes maintain location table whose size increases depending on the number of neighbors surrounding each node. More nodes means more overhead and vice versa, moreover, when a node needs to send a message, it needs to discover a route by broadcasting a route request message within a predefined area which limits scalability. When requesting a route, more complexity incurred for calculating respondable area and destination area which requires the node to know the speed of the destination node as well as the predefined delay of the route request. Even though it expands the respondable area and increases the destination's radius for better success in routing, it still does not guarantee packet delivery even if a route exists. This results in higher latency and is less robust.

Zhou et al. presented Geo-DFR (B.Zhou et al. [2008]) which incorporates directional forwarding (DRF) in routing (Y.Lee et al. [2005]). Routing is done mainly using greedy forwarding. However, in case of dead ends, DFR is used. Geo-DFR improves DFR to solve the dead end problem so that perimeter face routing is avoided. The authors use Fisheye State Routing protocol (FSR) (Pei et al. [2000]) which is the protocol that is "hosting" Geo-DFR. Scalability of Geo-DFR is affected by different factors. First, maintaining three tables in each node increases overheard especially when a node has many neighbors. But this limitation is local, since the number of records in two of these tables depends on the density of the neighboring nodes. Li et al. (Li et al. [2012]) proposed localized load-aware geographic routing using the concept of cost-to-progress ratio in greedy routing (CPR-Routing). The main idea behind this protocol is to combine the greedy forwarding technique and localized cost-to-progress ratio (CPR) (Stojmenovic [2006]). The load awareness used in this protocol tends to minimize load and maximize progress geographically towards the destination; however, this is difficult to achieve, so it tries to balance the two factors in making routing decisions. A drawback of this approach is the complexity involved in the calculation for selecting appropriate neighbor to forward a packet which could result in higher end-to-end delay.

Macintosh et al. proposed LANDY (Macintosh et al. [2012]) which uses locomotion (movement) and velocity of each node to predict the future location of each of these nodes so that data packets are forwarded efficiently towards the destination nodes. LANDY uses only local broadcasting to build a Locomotion Table (LT). When forwarding fails, instead of broadcasting, a recovery mode is invoked from the point of failure, hence allowing the protocol scale better. Overhead involved in this protocol is higher than that of the protocols discussed so far since it uses more control information to build tables at each node. That includes different samples of each node's location information exchanged periodically between nodes and building planar graphs to be used as alternatives to the normal forwarding mode.

2.2 On-demand Routing Protocols

We compare our proposed protocol with two of the well-known on-demand routing protocols that are discussed bellow. Under AODV (Perkins and Royer [199]), when a node needs to establish a route to a destination, it broadcasts a route request to all its neighbor nodes. A node receiving the route request node replies to the source node, if it has a route to the destination; otherwise it rebroadcasts the route request to all its neighbors. This process continues until the a route to the destination is found. This protocol is robust because broadcasting route request guarantees finding a route to the destination if there is one; however, as number of nodes increase, the number of redundant rebroadcasting of route requests increases. This means this protocol is not scalable.

Under DSR (Johnson and Maltz [1996]), when a node S needs to find a route to a node D, it broadcasts a Route Request packet (RREQ). On receiving RREQ, an intermediate node adds its address to the source route if it has no route to the destination D and if its address is not in the source route; then it rebroadcasts the updated RREQ packet. When a target node receives a RREQ packet, it puts the accumulated route record in a Route Reply (RREP) packet and sends it back to the sender. When it receives the RREP, the sender caches the route for subsequent routing. This protocol consumes lot of memory since each node stores large number of end-to-end routes. When the network grows, longer routes can cause frequent memory overflow, hence data packets can be dropped making the protocol less scalable and less robust.

3. PAPER OBJECTIVE AND DESIGN OF OUR PROTOCOL

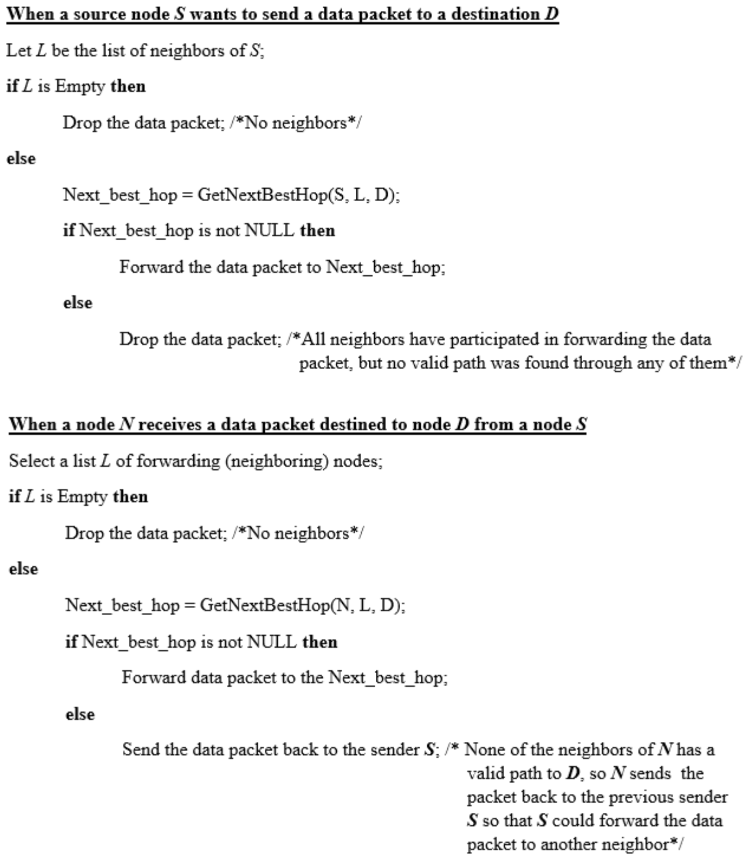
In this section, we first present the objective of the paper and basic idea behind our protocol, and then present detailed description of the protocol.

3.1 Paper Objective

From the geographic routing protocols that we discussed above, we observe that the more robust the protocol, the less scalable it is. Furthermore, many of the existing geographic routing protocols either use more control information (to make it more Robust) which may result in redundant messages, contention, and collision, or use less control information (to make it more scalable) which may lead to less packet delivery ratio. We address this issue in this paper and propose Greedy Routing Protocol with Backtracking (GRB), a novel and simple position-based routing protocol which allows each node forward data packets to its ***best neighbor*** (method used for best neighbor selection is explained in Section 3.2) possible until the destination is reached. Unlike GPSR, GRB uses less computation to determine the next hop on the route and it performs as well as or better than GPSR, AODV, and DSR.

3.2 Basic Idea Behind Our Protocol

GRB routes data packets either in forwarding mode (greedy mode or simple forwarding) or in backtracking mode. When a sender/intermediate node S wants to send/forward a packet to a destination D, it picks the *best neighbor* $N1$ and sends the packet to $N1$. The best neighbor $N1$ is determined as follows: It picks the neighbor that is closest to the destination than any other neighbor; note that this neighbor may not be closer to the destination than S itself because S may be facing a void. If the packet backtracks from this node to S, it picks the one that is closest to the destination among the remaining neighbors, and this process continues until all neighbors have been tried; if it cannot forward the packet through any of its neighbors, it sends the packet back to the node from which it received the packet. Every node on the path uses the same strategy to forward packets. Note that if the node picked is closer to the destination than S, then the forwarding is implicitly greedy; otherwise, forwarding is around the void. In this protocol,



```
When a source node S wants to send a data packet to a destination D
Let L be the list of neighbors of S;
if L is Empty then
    Drop the data packet; /*No neighbors*/
else
    Next_best_hop = GetNextBestHop(S, L, D);
    if Next_best_hop is not NULL then
        Forward the data packet to Next_best_hop;
    else
        Drop the data packet; /*All neighbors have participated in forwarding the data
                                packet, but no valid path was found through any of them*/

When a node N receives a data packet destined to node D from a node S
Select a list L of forwarding (neighboring) nodes;
if L is Empty then
    Drop the data packet; /*No neighbors*/
else
    Next_best_hop = GetNextBestHop(N, L, D);
    if Next_best_hop is not NULL then
        Forward data packet to the Next_best_hop;
    else
        Send the data packet back to the sender S; /* None of the neighbors of N has a
                                                     valid path to D, so N sends  the
                                                     packet back to the previous sender
                                                     S so that S could forward the data
                                                     packet to another neighbor*/
```

GRB data forwarding (Sending/Receiving Data Packets).

a source node drops data packets if it has no neighbors, it tried all the neighbors to forward the packet and failed, or the number of times the packet backtracked reached a predetermined threshold.

Formal description of the protocol for data forwarding and finding the next best hop is given in Figures 3 and 4.

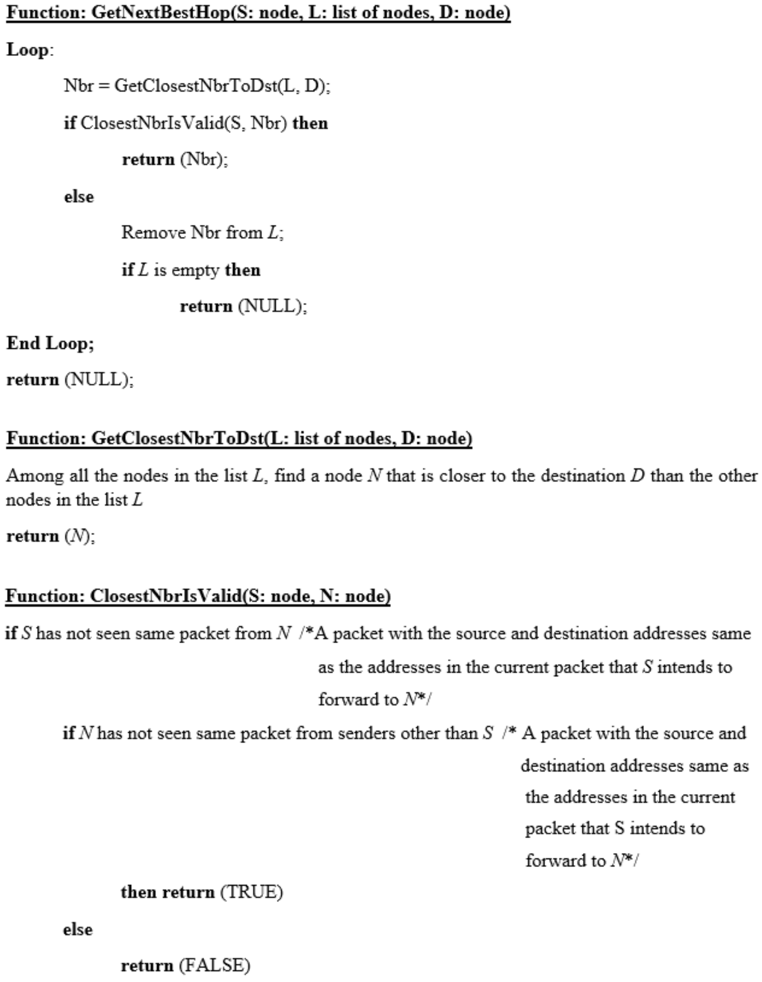
3.3 Assumptions

We assume that all nodes have the same transmission range (i.e., all links are bidirectional). We also assume that each node is equipped with a GPS and each node can get the location of the destination node through an available Location Service. In the following subsections we describe our protocol in detail.

3.4 Data Structures Used in the Protocol

Each node maintains the following two tables.

Neighbor Table.. Each node sends a $HELLO$ packet to all its neighbors in each time interval T. This $HELLO$ packet includes the node's id as well as its position. To minimize collision of $HELLO$ packets due to concurrent transmissions, we jitter each $HELLO$ packet transmission interval by R milliseconds between two successive transmissions of $HELLO$ packets so that each node transmits $HELLO$ packets at a random time chosen in the interval $[T - R, T + R]$. When a node receives a $HELLO$ packet, it creates in its Neighbor Table an entry containing neighbor



```
Function: GetNextBestHop(S: node, L: list of nodes, D: node)
Loop:
    Nbr = GetClosestNbrToDst(L, D);
    if ClosestNbrIsValid(S, Nbr) then
        return (Nbr);
    else
        Remove Nbr from L;
        if L is empty then
            return (NULL);
End Loop;
return (NULL);

Function: GetClosestNbrToDst(L: list of nodes, D: node)
Among all the nodes in the list L, find a node N that is closer to the destination D than the other
nodes in the list L
return (N);

Function: ClosestNbrIsValid(S: node, N: node)
if S has not seen same packet from N  /*A packet with the source and destination addresses same
                                        as the addresses in the current packet that S intends to
                                        forward to N*/
    if N has not seen same packet from senders other than S  /* A packet with the source and
                                                                destination addresses same as
                                                                the addresses in the current
                                                                packet that S intends to
                                                                forward to N*/
        then return (TRUE)
    else
        return (FALSE)
```

GRB data forwarding (Functions).

identifier (NbrID), neighbor position, and lifetime if that neighbor is not in the table; however, the lifetime is updated if already there is an entry corresponding to that neighbor. If a node does not receive $HELLO$ packets for a time longer than $2T$ form a neighbor node, it assumes the neighbor has moved and removes the associated entry from the table.

Seen Table.. This table helps picking ***best neighbor*** for forwarding packets to the destination. For that purpose, when a node receives a data packet, it stores the information about the packet in its Seen Table. As shown in Table I, each record of this table contains five fields namely, neighbor ID (NbrID), source address (Src), destination address (Dst), flag (Flag), and lifetime (Lifetime). $NbrID$ is the address of the neighboring node that has sent the packet, forwarded the packets, or the node from which the packet has backtracked. Src contains the address of the source node that generated the data packet. $Flag$ indicates whether the received packet is a new packet (i.e., forwarding mode) or it has backtracked from a neighboring node (i.e., backtracking mode). The flag is set to $FALSE$ when the packet is in forwarding mode and set to $TRUE$ when it has backtracked. The lifetime field specifies the lifetime of the associated record in the Seen Table. When a node receives a data packet, it creates an entry in its Seen Table if the packet is new. However, if it has received a data packet with the same source and destination addresses from the same neighboring node, then it updates the lifetime of the associated record. On the other hand, when the lifetime expires, the associated record with that lifetime is removed from the table.

3.5 Sending and Forwarding Packets

Each node can send, forward, and/or receive data packets. When a node has data packets to send and the destination node is not one of its neighbors, it picks the best neighbor as described in Sect. 3.2 and forwards the packet to that neighbor. Since our protocol does not enforce the next-hop N to be closer to the destination than the sender S, N is either closer to D than S (i.e., Greedy mode), or farther to D than S. However, the next-hop N must be closer to D than any other neighbor that has not seen a packet to the same source-destination pair according to their Seen Table. Before forwarding the packet to N, the source or intermediate node S does the following:

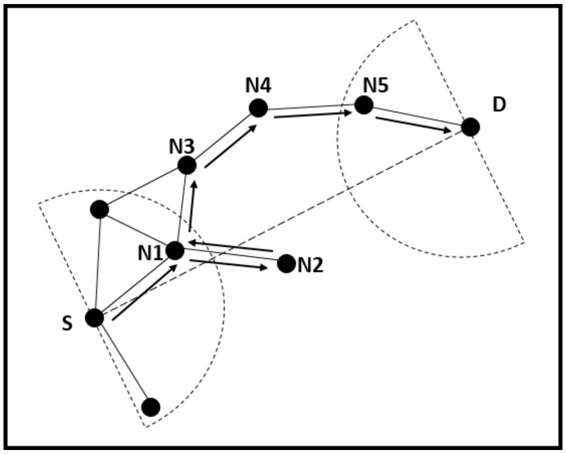
S Looks up its Seen Table for N.. If S has a record for N with the same source and destination addresses as that in the packet, then it considers N as invalid next hop for that packet and picks another neighboring node as the next hop. This means that S has received this packet from N which is either a new packet (i.e., flag is FALSE) or a backtrack packet (i.e., flag is TURE). Therefore, it cannot forward the packet to that node because that results in a loop. For example, in Figure 5, if the node $N3$ receives a data packet from node $N1$, it creates an entry in its Seen Table as shown in Table I. This entry tells $N3$ that $N1$ is invalid next hop because it has received the packet from $N1$ and as a result, the Seen Table prevents loop between $N3$ and $N1$. However, the Seen Table of $N1$ does not have $N3$ as a neighbor node so it can forward data packets to $N3$.

NbrID	**Src**	**Dst**	**Flag**	**Lifetime**
N1	S	D	False	T

Table I: Seen Table at Node $N3$ in Figure 5

S verifies with N if it is a valid next hop.. If N is not in the Seen Table of S, then S sends N a verification packet, with same source-destination pair in the header as in the data packet's header, asking N to check whether it has seen data packets from the same source-destination pair from any of its other neighbors. When N receives the verification packet, it checks its Seen Table for an entry that has the same source and destination addresses as that in the verification packet, with a Flag set to $False$, but with NbrID different from the ID of S. If such an entry is found, it means that N has seen a packet for the same source-destination pair and it sends a reply back to S indicating that it is invalid next hop. However, if such entry is found but Flag set to $True$, it means a neighbor $N1$ of node N has sent back the data packet to N after $N1$ failed to forward the packet. In this case, maybe there are neighbors of N other than $N1$ that have not tried to forward the packet yet, therefore N is not considered as invalid next hop and as a result, N sends a reply back to S indicating that it is a valid next hop for that data packet. After receiving the reply from N, if S finds N is a valid next hop, it forwards the packet to N. Otherwise, it picks another neighbor as a new candidate for next hop and checks if it is a valid next hop and so on. For example, in Figure 5, when $N1$ needs to send a data packet to $N3$, it sends a verification packet to $N3$. $N3$ checks its Seen Table for an entry with NbrID set to any ID other than $N1$, same Src and Dst values as those in the verification packet, and Flag set to $False$. However, $N3$ does not have such entry in its Seen Table (refer to Table I), hence $N3$ sends a positive reply to $N1$, and $N1$ forwards the packet to $N3$. If a node finds all its neighbors are invalid next hops, then the packet is sent back to the node from which it was received.

Packet backtracks.. A packet backtracks from the current node to its sender in the following two cases:



Data forwarding Example.

(1) All the neighbors of the current (intermediate) node have seen that packet. This means none of the neighbors could forward the packet.
(2) The current node has no neighbors other than the sender. For example, in Figure 5, $N2$ has no neighbors other than $N1$ which sent the packet to it. Therefore, the packet backtracks to $N1$ and $N1$ inserts a new entry to its Seen Table as shown in Table II. The Flag of the new entry (i.e., second row) is set to $True$ which means that from the perspective of $N1$, $N2$ is considered invalid next hop for that packet. Hence, when $N1$ tries to pick next hop for the same destination next time, it will not pick $N2$ as long as the Lifetime of the associated entry (i.e., second row in Table II) in the Seen Table of $N1$ has not expired.

NbrID	**Src**	**Dst**	**Flag**	**Lifetime**
S	S	D	False	T1
N2	S	D	True	T2

Table II: Seen Table at Node $N1$ in Figure 5

Packet is dropped.. A packet is dropped by a node in the following cases:
(1) All the neighbors have been identified as invalid next hops or the node has no neighbors.
(2) The number of times the packet backtracks reaches a predefined threshold.

4. PERFORMANCE ANALYSIS

In this section, we present the performance evaluation results of GRB compared to AODV (Perkins and Royer [199]), DSR (Johnson and Maltz [1996]), and GPSR (Karp and Kung [2000]). We first describe the simulation environment and then discuss the simulation results. We simulated GRB, AODV, and DSR on a variety of network topologies. Then we compared the results of GRB with the results provided in GPSR (Karp and Kung [2000]).

4.1 Simulation Environment

We used GloMoSim (glo []), a network-simulation tool for studying the performance of routing protocols for ad-hoc networks, for evaluating the performance of GRB. We chose IEEE 802.11 and IP as the MAC and network layer protocols, respectively. All nodes have a fixed transmission range of 250 m. We used the implementation of AODV and DSR that comes with the GloMoSim 2.0.3 package to compare their performance with GRB. We ran several simulations on two different sets of traffic flows. The simulations run in different terrain areas are shown in Table III; each simulation lasted for 900 seconds of simulated time. The nodes were distributed uniformly at random in the network area. We used the following four metrics to evaluate performance:

(1) Packet Delivery Ratio: Measures the success rate of delivered data packets.
(2) End-To-End Delay: Average time a packet takes to reach the destination node.
(3) Hop Count: The number of hops a packet traverses to reach the destination.
(4) Node Density: Number of nodes in the area.
(5) Network Diameter: Studying the effect of different network areas.

In this experiment, we varied the number of nodes simulated from 50 to 300. Two sets of random traffic flows have been used in the simulation. The first set is 30 CBR (Constant Bit Rate) flows in which different senders generate data packets to be forwarded to destinations. Each CBR flow sends packets at speed of 2Kbps and uses 64-byte packets. Depending on the start time and end time of each sender in each flow, different number of packets are sent by different CBR flows. However, in each and every flow, each sender sends a packet every 0.25 second. Node mobility is set using random Way-point (Johnson and Maltz [1996]) model. Under this model, each node travels from a location to a random destination at a random speed, the speed being uniformly distributed in a predefined range. After a node reaches its destination, it pauses for a predetermined amount of time and then moves to a new destination at a different randomly chosen speed. In our simulation, the speed randomly chosen lies between 0 and 20 meters/second. In order to study how mobility affects the performance of the routing protocols, we selected pause times of 0, 20, 30, 40, 60, 80, 100, and 120 seconds. When the pause time is 0 seconds, every node moves continuously. As the pause time increases, the network approaches the characteristics of a fixed network. The second set consists of 20 CBR flows which has 20 sender nodes generating packets at a speed and size same as that in the first set.

Nodes	**Network Area**	**CBR Flows**	**Packets Sent**
{50,75,100,125,150,175,200,225,250,300}	1500m X 1500m	30	8780
50	1500m X 300m	30	8780
112	2250m X 450m	30	8780
200	3000m X 600m	30	8780
50	1500m X 300m	20	5168

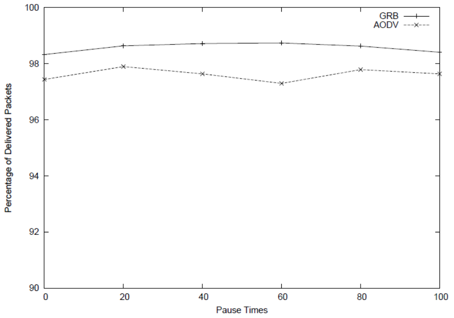
Table III: Topology used for Simulation

4.2 Packet Delivery Ratio

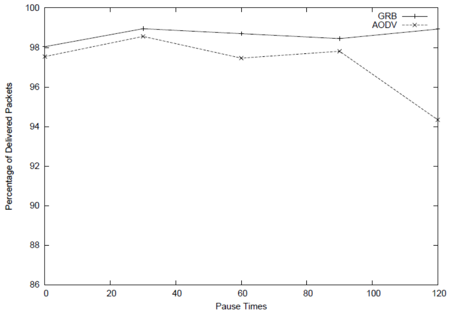
The overall average packet delivery ratio for DSR, AODV, and GRB are 55.52%, 97.38% and 98.60%, respectively. We select CBR flows randomly, hence it is not known whether there is a valid path between the source node and the destination node in each flow. Higher number of packets (refer to Table III) imposes higher demand on routing protocols as higher traffic is generated between source and destination pairs. GRB finds next hops locally with the most up to date location information of the nodes involved in the forwarding process. It simply picks next hops based on Seen Tables to forward data packets which results in few control packets. This makes GRB adapt locally to location changes, hence it tolerates mobility better than AODV and DSR. Therefore, GRB delivers higher number of data packets than DSR and AODV for different pause times as shown in Figures 6, 7, 8, and 9.

4.3 End-To-End Delay

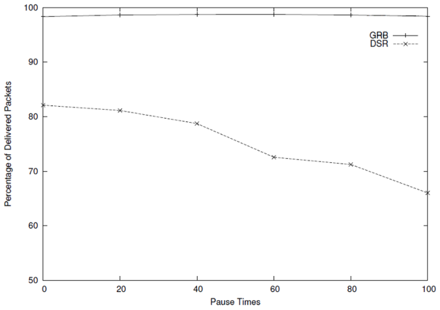
As shown in Figures 10 and 11, the overall average end-to-end delay for AODV and GRB are 22.17 milliseconds and 14.98 milliseconds, respectively. For each CBR flow, we take the average end-to-end delay of all the packets received by the destination node in that flow. Then, we take



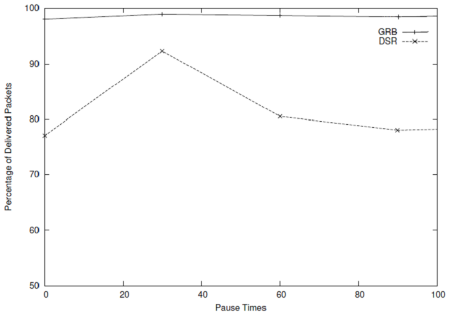
Packet Delivery Ratio (50 Nodes, 30-CBR Flows, network area (1500m x 300m)), GRB compared with AODV.



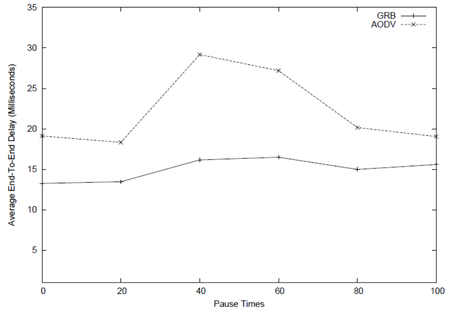
Packet Delivery Ratio (50 Nodes, 20-CBR Flows, network area (1500m x 300m)), GRB compared with AODV.



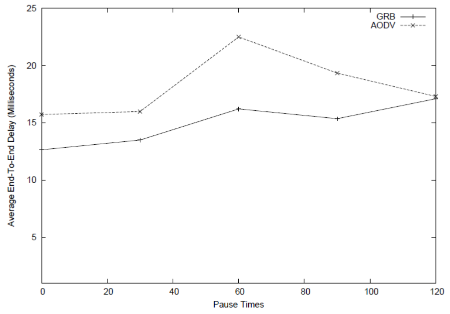
Packet Delivery Ratio (50 Nodes, 30-CBR Flows, network area (1500m x 300m)), GRB compared with DSR.



Packet Delivery Ratio (50 Nodes, 20-CBR Flows, network area (1500m x 300m)), GRB compared with DSR.



Average End-To-End Delay (50 Nodes, 30-CBR Flows, network area (1500m x 300m)), GRB compared with AODV.

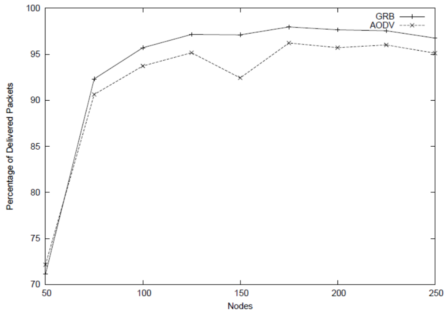


Average End-To-End Delay (50 Nodes, 20-CBR Flows, network area (1500m x 300m)), GRB compared with AODV.

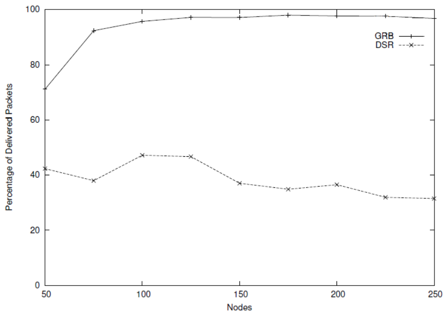
the average delay of all the CBR flows. Because of its simplicity, GRB takes less time to deliver data packets in most of the scenarios. As shown in Figures 10 and 11, packets take more than 18 milliseconds on average to reach their destinations under AODV; however GRB delivers packets in less than 16 milliseconds. We can see GRB delivers packets much faster when network size and area is moderately small (50 nodes, (1500m x 300m) area). That is because most of the packets find greedy paths which take less calculation time and the decision is taken quickly based on the neighbors' location information and their status regarding whether or not they are valid next hops. However, under AODV, data packets should wait for the route to be set up. Moreover, routes discovered under AODV may not be shorter than those discovered under GRB, because GRB uses greedy approach. As a result, AODV results in higher average end-to-end delay.

4.4 Node Density

Since our protocol uses only information about neighbors in forwarding decision, as node density increases, GRB keeps delivering higher fraction of data packets than AODV and DSR as shown in Figures 12 and 13. That is because both AODV and DSR depend on end-to-end route to forward data packets and that route is affected by mobility of the nodes and the size of the network. Hence due to mobility, more frequent link breaks occur leading to more route repair and setup and as a result, packets are lost more frequently. However, since the average end-to-end delay is taken only for packets that are delivered to their destinations and because data packets follow existing routes which decreases waiting time, AODV routes data packets in slightly less time than GRB when number of nodes grows to more than 200 as shown in Figure 15. However, GRB is faster in smaller networks (less density) because less computation required by nodes to make forwarding decisions since nodes have less neighbors. Average hop count is another parameter that we measure in this simulation to show that our protocol routes data packets with less number of hops as node density increases. For this metric, only the successfully delivered data packets are counted in the simulation results for both GRB and AODV. As shown in Figure 14, in smaller networks (i.e., less than 150 nodes), AODV uses less number of hops to forward data packets than GRB because there are more voids in sparse networks. This makes GRB data packets to go around voids through either next best hop or backtracking technique which makes GRB packets traverse more hops than AODV. However, as number of nodes increases, number of voids decreases and data packets move through greedy paths, hence GRB uses less number of hops than AODV in dense networks. It is worth to mentioning that under GRB, average hop count is also reduced because next hop is chosen greedily.

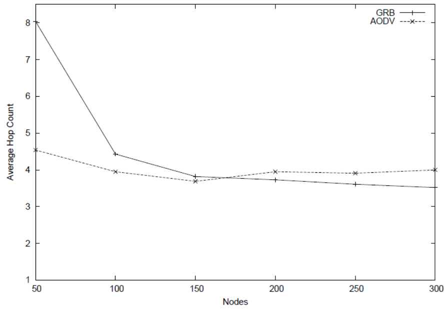


Packet Delivery Ratio as Number of Nodes increases (Network Area (1500x1500)), GRB compared with AODV.

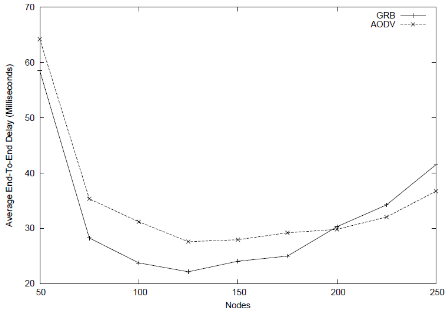


Packet Delivery Ratio as Number of Nodes increases (Network Area (1500x1500)), GRB compared with DSR.

4.4.1 *Network Diameter.* Figures 16 and 17 present packet delivery ratio results and Figures 20 and 21 present average hop count results for 112-nodes and 200-nodes networks with same CBR traffic and same node density for both networks. In these simulations, the terrain area within which nodes move are (2250x450) meters and (3000x600) meters successively. In these simulations, we evaluate the effect of changing network diameter on success rate and hop count for both GRB and AODV. The probability of route breaking increases as the routes grow longer. GRB's percent delivery ratio is higher than AODV and DSR at all pause times on larger networks because our protocol uses only local topology information, hence no penalty for GRB as the path increases from source nodes to destination nodes. Moreover, GRB recovers from loss

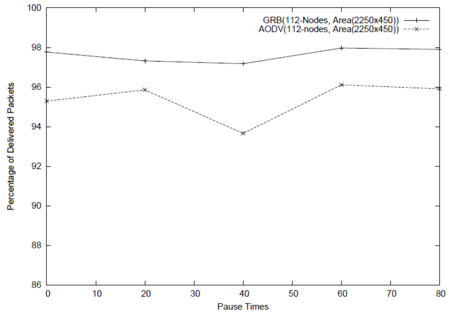


Average Hop Count as Number of Nodes increases (Network Area (1500x1500)), GRB compared with AODV.

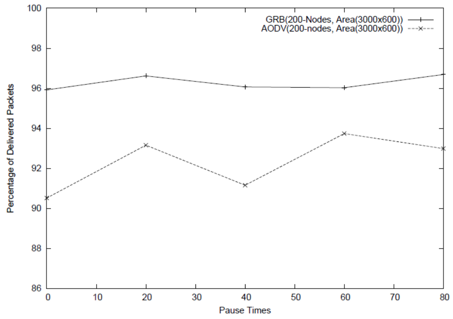


Average End-To-End as Number of Nodes increases (Network Area (1500x1500)), GRB compared with AODV.

of neighbor (next hop) instantaneously by simply finding another candidate next hop which will take over the forwarding process. However, AODV's percent delivery ratio decreases considerably as the network diameter increases because it needs to maintain longer end-to-end routes. DSR incurs higher traffic overhead in wider networks since it needs to maintain longer end-to-end source routes, hence its success rate decreases accordingly and it is much lower than that of GRB as shown in Figures 18 and 19. For the hop count metric, we calculate the average of all the received packets by all the destination nodes in all the flows for both GRB and AODV routing protocols. In small areas, AODV traverses less paths in higher mobility rates (i.e., lower pause times); however, GRB uses less hop counts when node mobility decreases (i.e., higher pause times) because the seen table entries will be more accurate as nodes remain for longer times in there destinations before moving to another destination. This gives GRB more chances to direct data packets through valid paths. In wider networks (i.e., larger diameter), GRB uses less hop counts than AODV for all the pause times (i.e., for low and high mobility rates) because in such networks, AODV suffers from more route breaking occurred due to longer end-to-end paths from source to destination nodes. Since same node density is used for both networks, it does not cost GRB any additional calculation since forwarding decisions are made locally, so it remains using less hops than AODV.



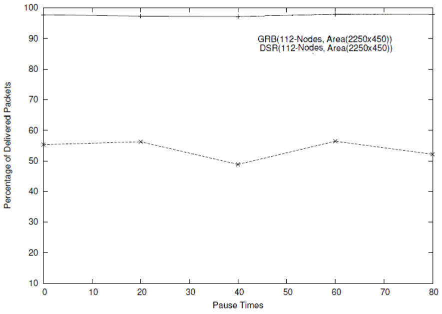
Packet Delivery Ratio of Network Area (2250x450), 112 nodes, 30-CBR Flows, GRB compared with AODV.



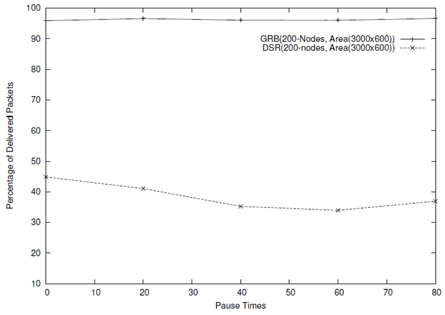
Packet Delivery Ratio of Network Area (3000x600), 200 nodes, 30-CBR Flows, GRB compared with AODV.

4.5 GRB Vs. GPSR

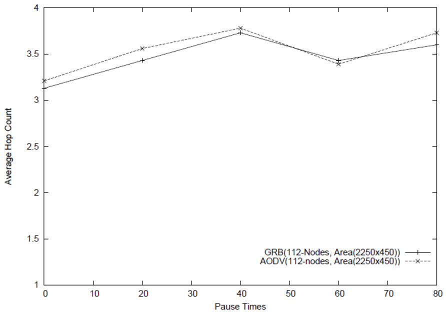
Even though we didn't simulate GPSR, we used the results published in GPSR (Karp and Kung [2000]) and the results we obtained for GRB to compare the performance of the two protocols. As stated in (Karp and Kung [2000]), GPSR counts only those packets for which a path exists



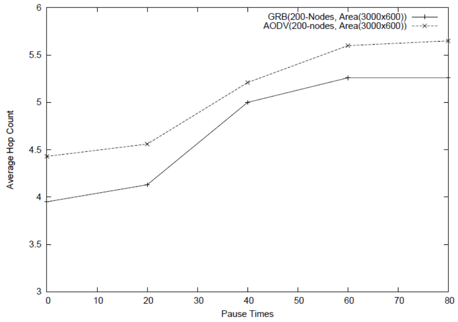
Packet Delivery Ratio of Network Area (2250x450), 112 nodes, 30-CBR Flows, GRB compared with DSR.



Packet Delivery Ratio of Network Area (3000x600), 200 nodes, 30-CBR Flows, GRB compared with DSR.



Average Hop Count of Network Area (2250x450), 112 nodes, 30-CBR Flows, GRB compared with AODV.



Average Hop Count of Network Area (3000x600), 200 nodes, 30-CBR Flows, GRB compared with AODV.

to the destination. We used the same input settings as those used for GPSR to compare the success rate. The settings are: 50 nodes, 30 CBR flows, pause times (0, 30, 60, and 120) seconds, area (1500x300) meters, and node density ($1node/9000m^2$). From the results presented in GPSR (Karp and Kung [2000]), successful packet delivery rate of GPSR ranges from 97.40% to 99.10% for pause time 0 second, while GRB successfully delivers 98.93% of the total packets sent. When pause time is 30 seconds, GPSR achieves success rates between 98.20% to 99.70%; while, GRB achieves a success rate of 99.33%. When pause time is 60 seconds, GPSR delivers from 98.70% to 99.40% of data packets sent, while GRB delivers 99.04% of all packets. Finally, when pause time is 120 seconds, GPSR's packet delivery rate ranges from 98.60% to 99.40% and GRB's packet delivery rate is 99.28%. In addition to that, we used another set of settings and accordingly, we show the corresponding results in Table IV. For 112 nodes, pause times 0 and 60 seconds, area (2250x450) meters, we see that GRB outperforms GPSR. We noticed that the performance of our protocol in successfully delivering data packets is very close to that of GPSR in some cases and higher than GPSR in other cases. However, when greedy routing fails due to a void in the direction of the destination, GPSR has to planarize the local network graph and use it to route around voids. Planarizing the graph results in computation overhead. GRB neither requires such planarizarion of local graph nor it switches from greedy mode to an alternative mode when packet faces void, instead a node simply selects the *best next hop* without imposing the condition that the next hop be closer to the destination than itself. GRB depends on the Seen Table to determine the next hop. Hence, GRB needs less control information which makes it to be fast and robust. So, we can see the difference between GRB and GPSR in delivering data packets ranges from 0.0017% to 0.0038% keeping in mind that GRB outperforms GPSR in some scenarios. When compared to the simplicity of GRB and the less control overhead GRB needs, this difference is negligible. It is worthy to recall that we select CBR flows randomly without knowing whether flows can deliver data packets. However, under GPSR, according to the authors: "Only packets for which a path exists to the destination are included in the graph".

Nodes	Network Area	Pause Time (s)	Delivery Ratio (GRB)	Delivery Ratio (GPSR)
50	1500m X 300m	0	98.98	99.10
50	1500m X 300m	60	99.07	99.40
112	2250m X 450m	0	98.00	97.50
112	2250m X 450m	60	98.54	98.25
200	3000m X 600m	0	97.02	97.45
200	3000m X 600m	60	96.84	97.65

Table IV: Input Settings and Corresponding Results for both GRB and GPSR

5. DISCUSSION

We note from the detailed description of the protocol Sect. 3 that data packets are forwarded greedily. When greedy forwarding fails, GRB picks the next best hop based on simple heuristics without incurring large computation overhead, unlike GPRS. A packet can come back (backtrack) to its sender/forwarder if the next hop picked for forwarding packet could not use any of its neighbors to forward the packet. If a packet backtracks from a neighbor selected as next hop, that next hop is recorded in the Seen Table; this prevents a node from selecting the same node as next hop when it has to forward succeeding packets to the same destination. Benefits of the Seen Table: (i) It prevents loops; (ii) after the first packet establishes the route to the destination, all the packets for the same source-destination pair follow the same route; (iii) it helps in decreasing hop count and latency.

6. CONCLUSION

In this paper, we presented GRB, a simple low-overhead position-based routing protocol which consistently and successfully delivers high percentage of data packets. We compared the performance of GRB with well known position-based protocol GPSR, the on-demand routing protocol AODV, and the Dynamic Source Routing (DSR) protocol. Our performance evaluation shows that GRB performs as good as GPSR (with low overhead) and better than AODV and DSR under most scenarios. Unlike GPSR, GRB does not need to construct planar graphs to route arround voids; it simply picks the best next hop to forward the data packets, hence GRB is simple. On the other hand, it achieves comparable packet delivery ratio to GPSR and AODV, hence it is robust.

Acknowledgement

A preliminary version of this paper appears in the Proceedings of the 12^{th} IEEE International Conference on Mobile Ad hoc and Sensor Systems as a poster paper.

References

GloMoSim: Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim.

Basagni, S., Chlamtac, I., Syrotiuk, V., and Woodward, B. 1998. A Distance Rouging Effect Algorithm for Mobility (DREAM). In *Proceedings of 4th annual ACM/IEEE International Conference on Mobile Computing and Networking.*

B.Zhou, Lee, Y., and Gerla, M. 2008. Direction Assisted Geographic Routing for Mobile Ad Hoc Networks. In *Proceedings of IEEE Military Communications Conference.*

CADGER, F., CURRAN, K., SANTOS, J., AND MOFFETT, S. 2013. A Survey of Geographical Routing in Wireless Ad-Hoc Networks. *IEEE Communications Surveys and Tutorials, Second Quarter 2013, Vol. 15, No. 2., pp. 621-653. 15,* 2, 621–653.

C.PERKINS AND HAGWAT, P. 1994. Highly Dynamic Destination Sequenced Distance Vector (DSDV) Routing for Mobile Computer. *ACM SIGCOMM Computer Communication Review 24,* 4, 234–244.

FLURY, R. AND WATTENHOFER, R. 2008. Randomized 3D Geographic Routing. In *Proceedings of the IEEE 27th Conference on Computer Communications.*

HUS, C. AND LEI, C. 2009. A Geographic Scheme with Location Update for Ad Hoc Routing. In *Proceedings of IEEE 4th International Conference on Systems and Networks Communications.*

JAIN, S., SHASTRI, A., AND CHAURASIA, B. 2013. Analysis and Feasibility of Reactive Routing Protocols with Malicious Nodes in MANETs. In *Proceedings of IEEE International Conference on Communication Systems and Network Technologies.*

JOHNSON, D. AND MALTZ, D. 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. *Mobile Computing 353*, 153–181.

KARP, B. AND KUNG, H. 2000. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the ACM 6th annual International Conference on Mobile Computing and Networking.*

KAVITHA, K., SELVAKUMAR, K., NITHYA, T., AND SATHYABAMA, S. 2013. Zone Based Multicast Routing Protocol for Mobile Ad-Hoc Networks. In *Proceedings of IEEE International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics, and Telecommunication System.*

LEMMON, C., LUI, S., AND LEE, I. 2009. Geographic Forwarding and Routing for Ad-Hoc Wireless Networks: A Survey. In *Proceedings of IEEE 5th International Joint Conference on INC, IMS and IDC.*

LI, H. AND SINGHAL, M. 2005. An Anchor-Based Routing Protocol with Cell ID Management System for Ad Hoc Networks. In *Proceedings of IEEE 14th International Conference on Computer Communications and Networks.*

LI, X., MITTON, N., NAYAK, A., AND STOJMENOVIC, I. 2012. Localized load-aware Geographic Routing in Wireless Ad Hoc Networks. In *Proceedings of IEEE International Conference on Communications (ICC).*

LIN, J. AND KUS, G. 2006. A Novel Location-fault-tolerant Geographic Routing Scheme for Wireless Ad Hoc Networks. In *Proceedings of IEEE Vehicular Technology Conference.*

MACINTOSH, A., GHAVAMI, M., SIYAU, M., AND LING, S. 2012. Local Area Network Dynamic (LANDY) Routing Protocol: A Position Based Routing Protocol for MANET. In *Proceedings of 18th European Wireless Conference.*

MAUVE, M., WIDMER, J., AND HARTENSTEIN, H. 2001. A Survey on Position Based Routing in Mobile Ad-hoc Networks. *IEEE Network Magazine 15,* 6 (November), 30–39.

PEI, G., GERLA, M., AND CHEN, T. 2000. Fisheye State Routing: A Routing Scheme for Ad hoc Wireless Networks. *IEEE Communications 1*, 70–74.

PERKINS, C. AND ROYER, E. 199. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of IEEE Mobile Computing Systems and Applications.*

SHEN, H. AND ZHAO, L. 2013. ALERT: An Anonymous Location-Based Efficient Routing Protocol in MANETs. *IEEE Transactions on Mobile Computing 12,* 6 (June), 1079–1093.

SHOBANA, M. AND KARTHIK, S. 2013. A Performance Analysis and Comparison of Various Routing Protocols in MANET. In *Proceedings of IEEE International Conference on Pattern Recognition, Informatics and Mobile Engineering.*

STOJMENOVIC, I. 2006. Localized Network Layer Protocols in Wireless Sensor Networks based on Optimizing Cost Over Progress Ratio. *IEEE Network 20,* 1, 21–27.

TALOOKI, V., MARQUES, H., AND RODRIGUEZ, J. 2013. Energy efficient dynamic MANET on-demand (E2DYMO) Routing Protocol. In *Proceedings of IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Network.*

V.Giruka and Singhal, M. 2007. A Self-healing On-demand Geographic Path Routing Protocol for Mobile Ad-hoc Networks. *Ad Hoc Networks 5,* 7 (September), 1113–1128.

Yau, S. S., Gao, W., and Huang, D. 2006. A Location-based Directional Route Discovery Protocol in Mobile Ad-hoc Network. In *Proceedings of IEEE Global Telecommunications Conference.*

Y.Lee, Chen, J., Zhou, B., and Gerla, M. 2005. DFR: Direction Forwarding for High Mobile, Large Scale Mobile Ad-Hoc Networks. In *Proceedings of the 4th annual Mediterranean Ad Hoc Networks.*

Zhao, Y., Chen, Y., Li, B., and Zhang, Q. 2007. Hop ID: A Virtual Coordinate-Based Routing for Sparse Mobile Ad Hoc Networks. *IEEE Transaction on Mobile Computing 6,* 9 (September), 1075–1089.

Dr. Baban A. Mahmood is currently a Lecturer at University of Kikuk, Kikuk, Iraq. He received his Ph.D in Computer Science from University of Kentucky, Lexington, Kentucky, USA. He received a B.Sc, degree in Computer and Software engineering from University of Al-Mustansryah, Baghdad, Iraq, in 2003 and an M.Sc. degree in Computer Science from University of Sulaimaniya, Kurdistan, Iraq, in 2009. He also received an M.Sc., degree in Computer Science from University of Kentucky, Lexington, Kentucky, USA, in 2015. He published his research work in the area of routing protocols in mobile ad hoc networks.

Dr. D. Manivannan is currently an associate professor of Computer Science at University of Kentucky, Lexington, Kentucky, USA. He received an M.Sc degree in mathematics from University of Madras, Madras, India. He received M.S and PhD degrees in computer and information science from The Ohio State University, Ohio, in 1993 and 1997 respectively. He published his research work in the following areas: fault-tolerance and synchronization in distributed systems, routing in wormhole networks, routing in ad hoc networks and vehicular ad hoc networks, channel allocation in cellular networks, wireless personal area networks and sensor networks. Dr. Manivannan has published more than 60 articles in refereed International Journals (a vast majority of which were published by IEEE, ACM, Elsevier, and Springer) and Proceedings of International Conferences.
He served as an Associate Editor of IEEE Transactions on Parallel and Distributed Systems and IEEE Communications Magazine. He served as Program co-chair of three International Conferences in the areas of reliable distributed systems and wireless networks and served as program committee member for over 40 International Conferences. He is on the Editorial Board of Information Sciences journal and Wireless Personal Communications journal. He served as reviewer for more than 30 International Journals published by ACM, IEEE, Elsevier, Springer, Oxford University Press, Taylor and Francis and others. He also served on several proposal review panels of US National Science Foundation and as external tenure reviewer for other Universities. Dr. Manivannan's research has been funded by grants from the US National Science Foundation and the US Department of Treasury.
Dr. Manivannan is a recipient of the Faculty Early Career Development award (CAREER award) from US National Science Foundation. He is a senior member of the IEEE and ACM.