International Journal of Advanced Research and Development
ISSN: 2455-4030
Impact Factor: RJIF 5.24
www.advancedjournal.com
Volume 3; Issue 1; January 2018; Page No. 135-142

On-demand routing protocol, which utilizes greedy methods with backtracking technique for route request propagation during path establishment in mobile ad-hoc network (MANETs)

Pretty Goel
Research Scholar, Department of Computer Science, OPJS University, Churu, Rajasthan, India

Abstract
Mobile ad hoc networks are described by multi-hop wireless cell nodes that transfer data with each other without unified control or set up. Routing protocols for a Mobile Ad-hoc system can be named as proactive (table-driven) and reactive (on-demand), contingent upon how they respond to topology changes. For MANET, on-demand routing protocols have been considerably studied. Generally, An on-demand routing protocol just tries to find/maintain routes when needed throughout the whole system, which isn't an adaptable approach. A routing protocol is required to address 3issues in MANET: The route maintenance, the data forwarding and the route finding. On the other way, Geographic routing has become one of the greatest routing schemes in MANET mostly due to its adaptability. It isn't build up route a priory but route information packets in a greedy manner towards the destination. The standard approach in geographic routing is greedy forwarding, which flops if the packet encounters a void node (a node that has no neighbor that is nearer to the destination than itself), it is routed nearby the void using several techniques such as creating a planarized network chart and then using other hand rule to route around the void. In this paper, we proposed an on-demand routing protocol (ODGBk) which utilizes greedy approaches for path request propagation during path creation. At the point when route request encounters a void, it simply utilize backtracking technique to forward the route request around the void without needful the development of planarized network chart to round nearby the void. Our protocol performance are better as compare with the popular routing protocol AODV which has over 20000 citations. The Simulation outcomes display that proposed protocol has a lower control overhead, less hop count and higher packet-delivery ratio on average than AODV.

Keywords: MANET, routing in MANETs, geographic routing, topology routing, hybrid routing

1. Introduction
A Mobile Ad-hoc network (MANETs) is consists of stationary nodes or mobile routers that are connected automatically to each other without any stable infrastructure [1]. In MANETs nodes can self-organize dynamically connected arbitrarily for some temporary time that is why all node in MANET is freely to move [2]. MANET include in some zone such as business colleague sharing data, military relaying information, disaster recovery, rescue operations, monitoring animal habitats, earthquake, VANET [3]. In such network might seem to easily flood the whole network and no pre-existing infrastructure for communication. [4]. A MANETs routing protocol is required whenever a data packet should be transfer to an end point (destination) from source through the number of hubs. In past several routing protocols for mobile Ad-hoc network have been proposed [5]. In MANET each node is able to forward packets of data to other nodes. These protocols searching a path for data packet delivery and carry the data packet to the correct destination. We can characterized of routing protocol in several ways Maximum protocol are categorized based on the network structure and scheme of routing [6]. This is categorized into 3 special approaches as position based routing protocol (geographic routing protocol), Table Driven Protocols (Proactive or topology based routing protocol) and On-Demand Protocols (Reactive Protocols) [7]. Topology or Table Driven-based routing protocols utilize data about the links. That is, data about the paths is maintained and routes are managed based on the data of the connections that exist in the network and this is upon the current topology of the network can be categorized into 3 types, first is proactive, second is reactive and third is hybrid routing protocols [8, 9]. Proactive and classical routing protocols are similar like distance-vector routing this is constantly find routes and keep them in the routing tables [10] and reactive protocols find and maintain paths only if required, which outcomes in primary delays until the paths are manage [11] and hybrid routing protocols, which is a blend of reactive approach and proactive approach [12]. To remove the limitations of the topology-based protocols we present the Geographic or Position-based routing approaches in MANETs. All this routing protocols based on having one part of messages and that is the nodes' physical area information. The Geographical routing protocols imply that the hosts joining in the routing procedure should be alert of their geographic locations. The aim of Geographical location-based routing protocol is to reach a particular host and it is considered stateless which means that hops do not want to maintain network topology related information. Geographic routing does not need a route administration process, it conveys a low overhead compared to other routing schemes, like proactive, hybrid topology based routing and reactive routing protocols [13]. The only information they need is nodes' location. In general, nodes need destination location, neighboring nodes' locations, and their own location information. Routing is then accomplished using this area

information by forwarding data packets node by- node until the destination node is reached [14]. One of the main approaches used in position-based routing is Greedy forwarding GPSR [15] in which a node forwards packets to its next neighbor that is nearer to the Destination (end-point) node than itself. Main focus in this research is evaluating or designing an efficient low overhead routing protocol by using a novel backtracking scheme through which routing control information (i.e., route requests) propagates. This is done by combining features from topology based and geographic routing protocols.

2. Proposed System

In this sector, firstly we present the objective of research and basic idea of our protocol after that we present a complete description of the proposed protocol.

2.1 Objective

In the routing protocols that we discussed above, we observe that robust protocols are often less scalable. We address this problem in this paper and propose a On Demand Routing Protocol using greedy forwarding with Backtracking Technique (ODGBk), a novel and simple routing protocol that combines features of topology based or position based (RP) routing protocols. ODGBk allows every node to forward route requests to its best possible until the destination is reached. ODGBk tries to minimize the transmission of redundant route requests by letting only one node to forward route request message (RREQ) at a time. Performance evaluation shows that ODGBk significantly outperforms AODV.

2.2 Basic Idea

To send data packets, routes should be created between source and destination nodes. ODGBk sends route requests (RREQ) either in forwarding mode or in backtracking mode. When a sender node A wants to send a packet to a destination B, it picks the best neighbor P1 and sends a route request (RREQ) packet to P1.The best neighbor P1 is determined as follows: Apicks the neighbor that is closer to the destination than all of the other neighbors. Note that this neighbor may not be closer to the destination than A itself because A may be facing avoid. If the RREQ packet backtracks from P1 to A, A 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 RREQ packet through any of its neighbors, it sends the RREQ packet back to the node from which it received the RREQ packet. Every node on the path uses the same strategy to forward RREQ packets. Note that if the node picked is closer to the destination than A, then the forwarding is implicitly greedy; otherwise, the RREQ packet is forwarded around the void. Once the route is built between the source node and the destination node, data packets are transmitted from the source node to the destination node via that route. In this protocol, a source node drops data packets if it has Po neighbors, if it tried to forward the RREQ packet through all the neighbors and failed, or if the number of times the RREQ packet backtrack reached a predetermined threshold.

2.3 Data Structures

Nodes that participate in path searching process maintain a seen table during the route discovery and maintain a route table during the data forwarding process. Each node also maintains a neighbor table.

2.3.1 Neighbor Table

Each node maintains identity and location in formation of its neighboring nodes in its neighbor table. Each node sends a HELLO packet to all its neighbors in each time interval M. This HELLO packet in cludes the node's id as well as its position. To minimize collision of HELLO packets due to concurrent transmissions, every HELLO packet transmission interval by Cmilli seconds between two successive transmissions of HELLO packets so that each node transmits HELLO packets at a random time chosen in the interval [M - C, M + C]. When a node receives a HELLO packet, it creates in its Neighbor Table an entry containing neighbor identifier, neighbor position, and lifetime of the neighbor associated with that hello packet. The lifetime of an entry in this table is updated whenever the node receives any packet (RREQ, data, route reply packet, etc.) from the neighbor associated with that entry.

Table 1: Routing Table at node P1 in Fig 2

Sequence-Number	Destination-Address	Source	Lifetime	Next-hop	Activated
Seq N	A	False	M	A	False

Table 2: Routing Table at node P5 in Fig 2

Sequence-Number	Destination-Address	Source	Lifetime	Next-hop	Activated
Seq N	A	False	M	P4	True
Seq N	B	False	M	B	True

2.3.2 Seen Table

This table helps picking best neighbor for forwarding RREQ packets to the destination. For that purpose, when a node receives a RREQ packet, it stores the information about the packet in its Seen Table. As shown in Table 3, each record of this table contains five fields namely, neighbor ID, source address, destination address, flag, and lifetime. Neighbor ID is the address of the neighboring node that has sent the RREQ packet, forwarded the RREQ packets, or the node from which the RREQ packet has backtracked. Source address contains the address of the source node that generated the RREQ packet. Flag indicates whether the received RREQ 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 RREQ 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 of the neighbor associated with the RREQ packet. When the lifetime expires, the associated record with that lifetime is removed from the table.

2.3.3 Routing Table

This table has five fields namely, Destination-Address is the address to which the data packet, RREQ, or RREP is forwarded. Lifetime is the age of the associated record in the routing table. Activated is a Boolean field, which indicates whether the associated entry is active or not; if this field is set to True, then the related record can be used for forwarding either control or data packets. Source field indicates a source node that initiates a route request or a node that initiates a route reply (i.e., either the destination or an intermediate node that has an active route to the destination node).Sequence-Number is an integer field associated with each RREQ.

2.4 Best Next-Hop Selection and Verification

For selecting the next-hop, P to forward the RREQ packet, the source or an intermediate node A does the following. It picks the neighbor P that is closer to the destination than any of the other neighbors that have not been considered for the next-hop selection and does the following verifications.

2.5 A looks up its Seen Table for P

If A has an entry for P with the same source and destination addresses as that in the RREQ packet, then it considers P as an invalid next-hop for that packet and picks another neighboring node as the next hop. This means that A has received this request from P which is either a new request (i.e., flag is FALSE) or a backtrack request (i.e., flag is TRUE). Therefore, it cannot forward the RREQ packet to that node because that would result in a loop. For example, in Fig 2, if the node P3 receives a RREQ from node P1, it creates an entry in its Seen Table as shown in Table 3. This entry tells P3 that P1 is an invalid next hop because it has received the RREQ fromP1 and as a result, the Seen Table prevents loop to occur between P3 and P1. However, the Seen Table of P1does not have P3 as a neighbor node so it can forward RREQ packets to P3.

2.6 A verifies with P if it is a valid next hop

If P is not in the Seen Table of A, then A sends P a verification packet, with the same source-destination pair in the header as in the RREQ packet's header, asking P to check whether it has seen RREQ packets with the same source-destination pair from any of its other neighbors. When P receives the verification packet, it checks its Seen Table for an entry that has the same source and destination address esas those in the verification packet, with a Flag set to False, but with a neighbor id different from the ID of A. If such an entry is found, it means that P has seen a RREQ packet for the same source-destination pair and it sends a reply back to A indicating that it is an invalid next hop. However, if such an entry is found but the Flag is set to true, it means a neighbor P1 of node P has sent the RREQ packet back to P afterP1 failed to forward the RREQ packet. In this case, there may be neighbors of P other than P1 that were not checked to forward the RREQ packet yet, therefore P is not considered as an invalid next hop and as a result, P sends a reply back to A indicating that it is a valid next hop for that RREQ packet. After receiving the reply from P, if A finds P is a valid next hop, it forwards the RREQ packet to P otherwise, it picks another neighbor as a new candidate for next hop and checks if it is a valid next hop and soon. For example, in Fig 2, when P1 needs to send a RREQ packet to P3, it sends a verification packet to P3. P3checks it is Seen Table for an entry with the neighbor ID set to any ID other than P1, with same source and destination values as those in the verification packet, and Flag is set to False. Since P3 does not have such an entry in its Seen Table (refer to Table 3), it sends a positive reply (i.e., P is a valid next hop) to P1, and P1 forwards the RREQ packet to P3. This verification process is necessary to prevent loops. For example, when node P8 receives the RREQ back from node P9, after it has verified with P2, P8 send sa verification packet to P1. P1 finds that it has seen are quest (i.e., an entry) with a neighbor ID set to A which is different from P2, with same source and destination values as those in the verification packet, and a Flag set to False (see Table 4). Therefore, P1 sends a negative response to P8indicating that P1 is an invalid next hop for that request. Hence, P8 sends the request back to P2, which in turn sends the request back to P1. Generally, if a node finds all its neighbors are invalid next hops, then the RREQ packet is sent back to the node from which it was received.

2.7 RREQ packet backtracking

A route request packet backtracks from the current node to its sender in the following two cases:

- The current node has no neighbors other than the sender. For example, in Fig 2, P9 has no neighbors other than P8, which sent the request packet to it. Therefore, the packet backtracks to P8 and P8 inserts a new entry into its Seen Table as shown in Table 6. The Flag of the new entry (i.e., second row) is set to true which means that from the perspective of P8, P9 is considered an invalid next hop for that RREQ packet. Hence, when P8 tries to pick the next hop for the same destination next time, it will not pick P9 if the Lifetime of the associated entry (i.e., second row in Table 6) in the Seen Table of P8 has not expired.
- All the neighbors of the current (intermediate) node have seen that packet. This means none of the neighbors could send the data packet.

2.8 Path searching and Maintenance

In various of the pre-existing AODV on demand routing protocols, when a source node need to discover a path to a destination so source node will start broadcasting, means it send the data packet with unique ID (Route Request, RREQ) to their neighbors and neighbors also broadcasting for same data packet to their neighbors (DSR, AODV). This method results in flooding broadcasts and incurs lots of overhead. In order to minimize such overlapped broadcasts, ODGBk uncast RREQ packets as follows: when a node needs to forward a RREQ packet, it picks the best neighbor P and forward the RREQ data packets to that neighbor. When P receives this message, it looks up its routing table to see whether it has a path to the destination node. If so, it forwards a (RREP) that is route reply message back from their neighbors to the source node. Otherwise, it send the RREQ to the next best neighbor node. Since our protocol does not enforce the next-hop P to be closer to the destination than the sender A, P is either closer to B than A (i.e., Greedy mode), or farther to B than A. However, the next-hop P must be closer to B than any other

neighbor that has not seen a RREQ packet to the same source-destination pair.

Table 3: Seen Table at node P3 in Fig 2

Destination Address	Lifetime	Flag	Source Address	Neighbor ID
B	M	False	A	P1

Table 4: Seen Table at node P1 in Fig 2

Destination Address	Lifetime	Flag	Source Address	Neighbor ID
B	M1	False	A	P2
B	M2	False	A	P2

2.9 Reverse Path Setup

When node P1 in Figure 1receives, a RREQ from source node A, it creates an entry in its routing table as shown in Table 1, and forwards the RREQ up to next hop P2. However, since P2 has no neighbors other than P1, P2 sends the RREQ back to P1. Even though it receives a RREQ from P2, but P1 does not create a reverse path for that request because it is backtrack control packet. The reverse path entries are maintained for a lifetime long enough for the RREQ to reach the destination and produce a reply to the source node.

Table 5

Destination Address	Lifetime	Flag	Source Address	Neighbor ID
B	M1	False	A	P2
B	M2	True	A	P9

2.10 Route Setup

When node P5 receives the RREP packet from node B, it inserts a forwarding entry to the destination B as shown in Table 2. Once the RREP arrives at the source A, it can start data transmission. It is worthy to point out that nodes which send back RREQ to previous nodes are not considered part of the route. This feature reduces the number of hops the data packets will travel through after the route is built between source node and destination node. The dashed lines in Fig 2 is the route created through the route setup process and it does not include node P2 since that node sent the RREQ back to P1.

2.11 Route Maintenance

When an established route breaks due to a node's movement on the path, route discovery is re-initiated by the source node if it still needs to send data packets. If the source node moves during that active session, it simply re-initiates the route discovery process. When either an intermediate node or the destination node moves, the source node is notified through a special control packet sent by the node at which the link broke. As a response to that, the source node stops sending data packets through that broken route and re-initiates a new route discovery. Like AODV, when a new RREQ is built, a new sequence number is assigned to the new RREQ so that other nodes can recognize this is a new request.

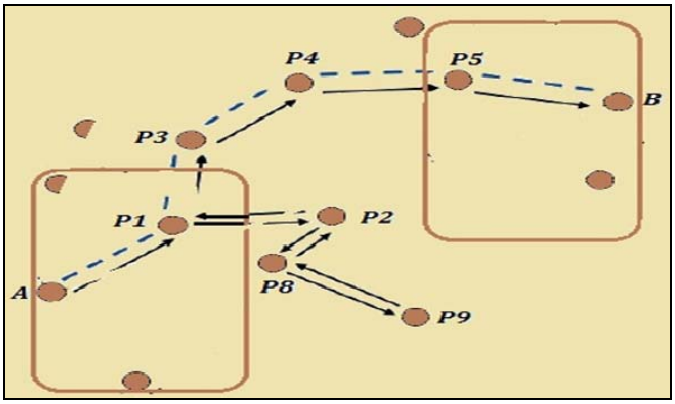

Fig 1: Route Setup

3. Performance Analysis

In this section, we present the performance evaluation results of ODGBK compared to AODV. We first describe the simulation environment and then discuss the simulation results. We simulated ODGBK and AODV on variety of network topologies.

3.1 Performance Metrics

We used the following three metrics to evaluate performance in three different scenarios namely, mobility, node density, and network diameter. In this experiment, we varied the number of nodes simulated from 100 to 500. We used a set of 40 CBR random traffic flows in the simulation. Each CBR flows ends packets at speed of 2.5 Kbps and uses 128-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 senders ends a packet every 0.20 second. Node mobility is set using random Waypoint 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 predetermined amount of time and then moves to a new randomly chosen destination at a randomly chosen speed. In our simulation, the speed randomly chosen lies between 0 and25 m/s. In order to study how mobility affects the performance of the routing protocols, we selected pause times of 0, 20, 40, 60, 80, and 100 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.

Table 6: Simulation Topologies

Network Area	Nodes	Packets Sent	CBR Flows
2000 X500	100	9170	40
2000 X2000	100	9170	40
3000 X3000	200	9170	40
2500 X2500	100 to 500	9170	40

3.2 Mobility

In this scenario, we evaluated our protocol with respect to the three metrics as node mobility changes from 0 to 25 m/s. We selected CBR flows randomly, hence it is not known whether there is a valid path between the source node and the

destination node for each flow. Higher number of packets imposes higher demand on routing protocols as higher traffic is generated between sources, destination pairs. ODGBk 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 RREQ packets, which makes ODGBk, adapt locally to location changes, hence it tolerates mobility better than AODV. Therefore, ODGBk delivers slightly higher number of data packets than AODV for most of the different pause times as shown in Fig3.However, as shown in Fig 4, the routing protocol overhead for ODGBk is much lower than that of AODV. AODV is are active routing protocol that broadcasts route request packets when nodes need to build routes. This feature of AODV generates increased routing traffic. When a route breaks, AODV uses broadcast to set up a new route. ODGBk does not use broadcast, so it incurs less control overhead. Hence, this feature reduces the control overhead compared to that generated by AODV.

Fig 2 shows a comparison of the number of hops of routes through which both ODGBk and AODV successfully deliver data packets. ODGBk tries to find the route greedily such that the length of the route is shorter. This makes ODGBk builds routes that have less hops on average than AODV.

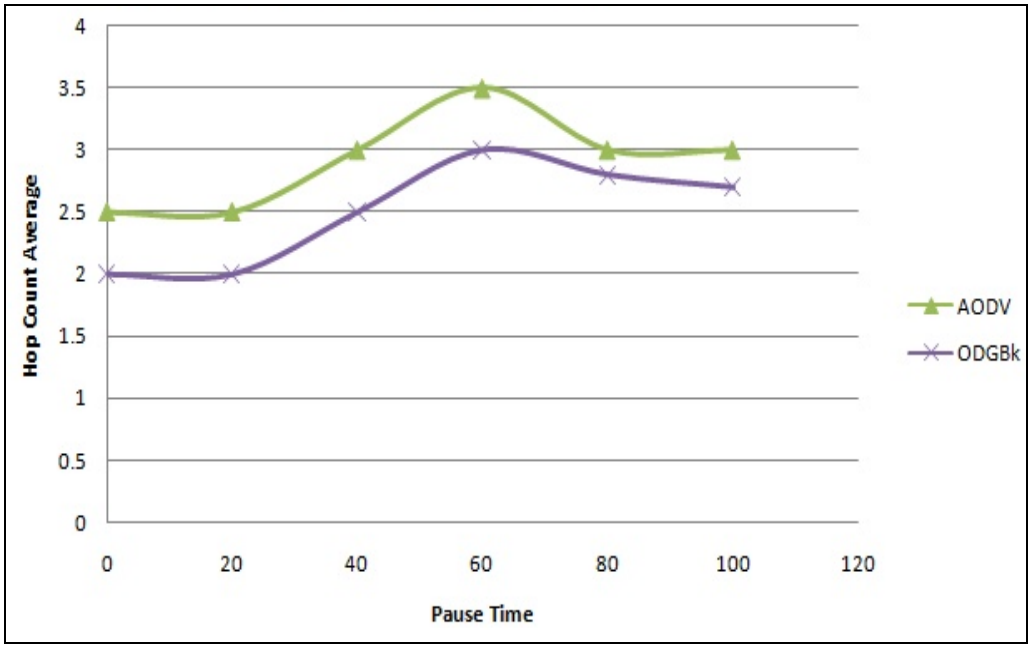

Fig 2: Hop Counting as Mobility Changes (60 Nodes, network area (1500m x 300m)), Comparison ODGBK compared with AODV

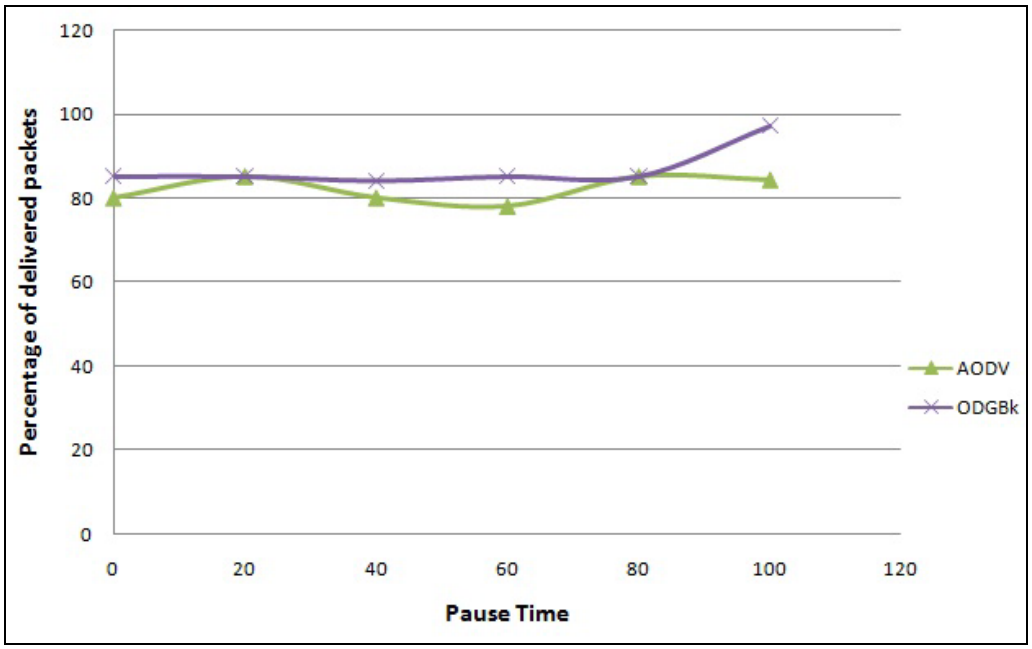

Fig 3: Packet Delivery Ratio as Mobility Changes (60 Nodes, network area (2500m x 500m)), comparison between ODGBk and AODV.

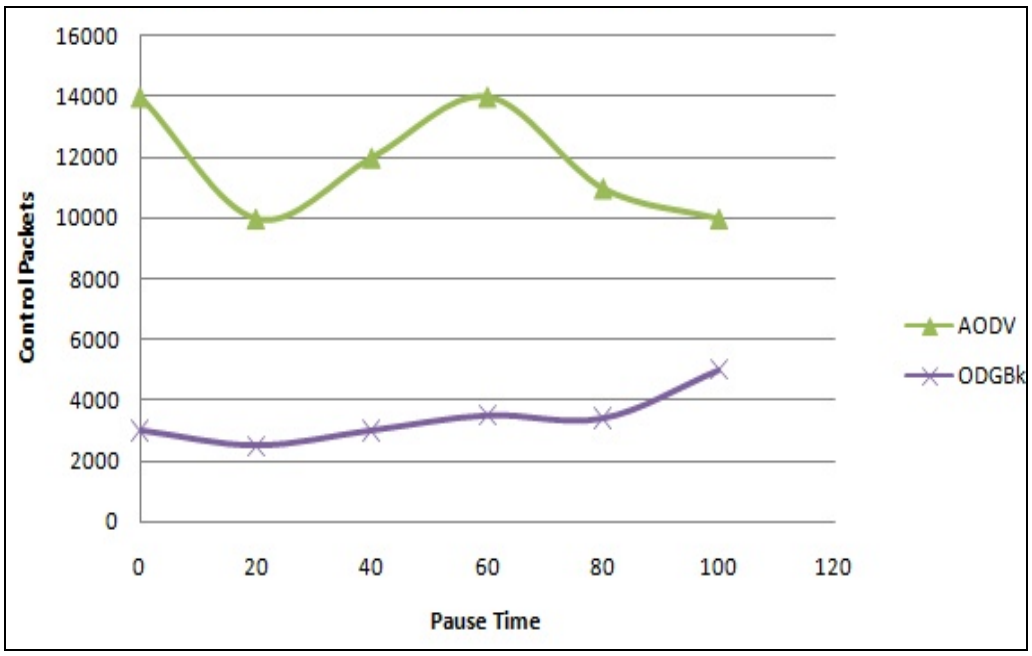

Fig 4: Packets Control as mobility changes 60 nodes network area 2500 x 500, comparison between ODGBk and AODV

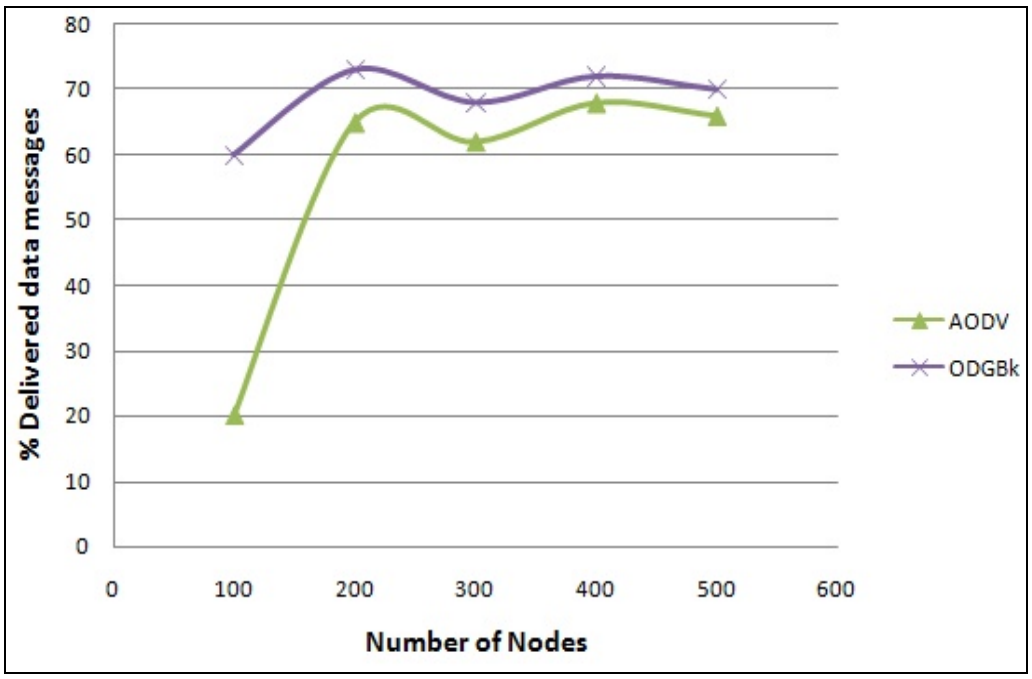

Fig 5: Data messages delivery ratio as node density increases, Area of network 2500 x 2500, comparison between ODGBk and AODV

3.3 Node Density

In this scenario, we varied the number of nodes from 100 to 500 in a network area of (2500m x 2500m). We studied the effect of increasing number of nodes on the three metrics. Since our protocol uses only information about neighbors in forwarding decision, as node density increases, ODGBK delivers higher fraction of data packets than AODV as shown in Fig 5. Average hop count is another parameter that we measured 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 ODGBk and AODV. ODGBK uses fewer hop counts than AODV in most cases as shown in Fig 7. Since there are more voids in sparse networks, the difference in average hop count between the two algorithms is smaller than dense networks. As number of nodes increases, number of voids decreases and ODGBK forwards RREQ packets through greedy paths, hence ODGBK uses less number of hops than AODV in dense networks. On the other hand, control packets produced by our protocol are less than those produced by AODV. As the network becomes denser, number of route control packets issued by ODGBK remains close to that in sparse networks. This gentle rise is because of ODGBK's single hop propagation mechanism of

RREQ packets, which is very efficient in controlling routing overhead. As shown in Fig6, ODGBK has relatively constant control overhead as number of nodes increases from 20 to 500.

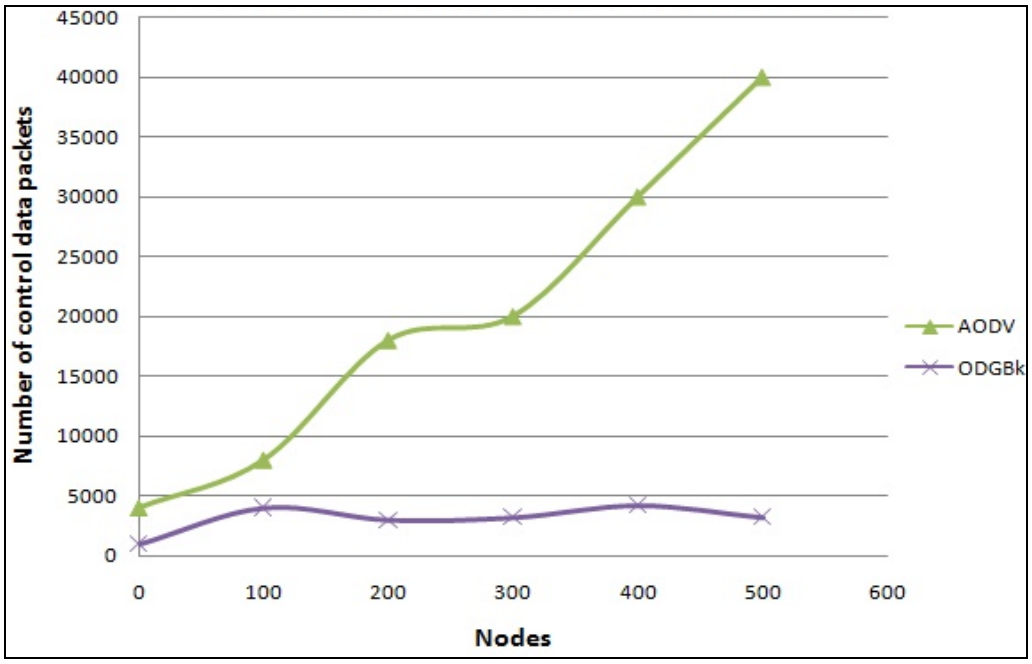

Fig 6: Control data packets as node density increases, network area 2500 x 2500, comparison between ODG Bknd AODV

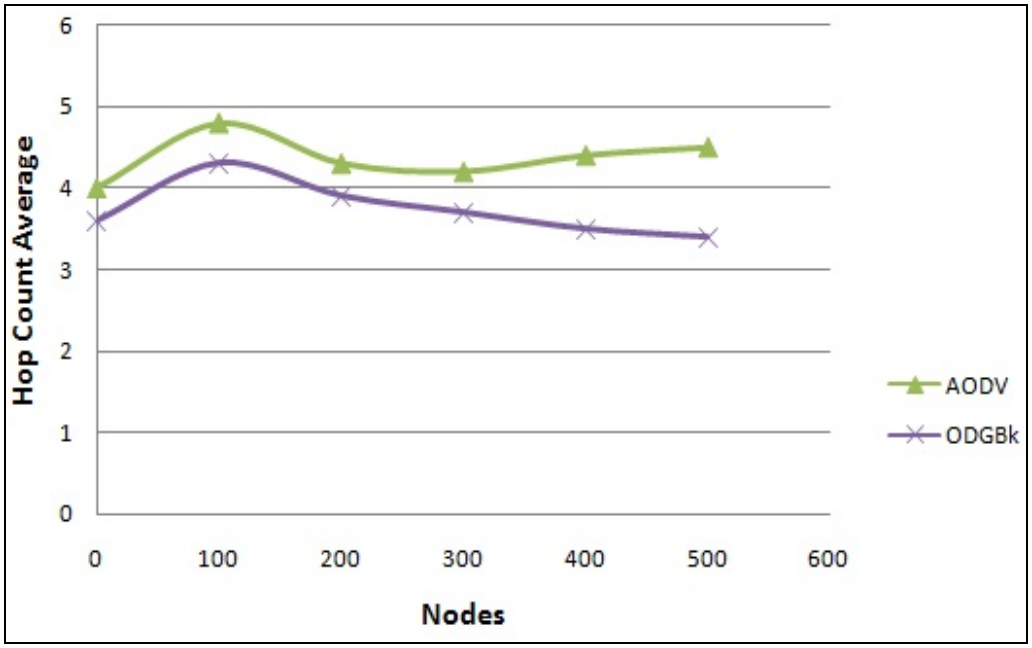

Fig 7: Hope Number as Node density increases, network area 2500 x2500, comparison between ODGBk and AODV

4. Conclusion

We proposed ODGBk, a simple low-overhead hybrid on demand routing protocol that combines features of geographic protocols and topology based protocols. ODGBk consistently and successfully delivers high percentage of data packets at lower routing control overhead. We compared our result of ODGBk with the AODV routing protocol in graph. Our performance evaluation present that ODGBK performs better than AODV in most scenarios. Unlike AODV, ODGBk does not need to flood route request packets; it simply picks the best next hop to forward the RREQ packets.

5. References

1. Ilyas M. The Handbook of Ad Hoc Wireless Networks, CRC Press, 2003.
2. Gaurav Sachan DK. Sharma, KarishmaTyagi, Abhimanyu Prasad. Enhanced Energy Aware Geographic Routing Protocol in MANET: A Review, International Journal of Modern Engineering Research, 2013.
3. AtekehMaghsoudlou, Marc St-Hilaire, Thomas Kunz.

Department of Systems and Computer Engineering Carleton University, Ottawa, ON, Canada, A Survey on Geographic Routing Protocols for Mobile Ad hoc Networks, Carleton University, Systems and Computer Engineering, Technical Report, 2011.

4. Pei G, Gerla M, Chen TW. Fisheye state routing: a routing scheme for ad hoc wireless networks, in Communications, 2000. ICC 2000. 2000 IEEE International Conference on, 2000s.
5. Sanjeev Sharma, Sanjay Singh. A Survey Of Routing Protocols And Geographic Routing Protocol Using Gps In Manet, Journal of Global Research in Computer Science, 2012.
6. Watanabe M, Higaki H. No-Beacon GEDIR: Location-Based Ad-Hoc Routing with Less Communication Overhead. Proc. the International Conference on Information Technology, 2007.
7. Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-case optimal and averagecase efficient geometric ad-hoc routing. In Proceedings of the 4th ACM International Symposium on Mobile Computing and Networking, 2003.
8. HL, Singhal M. An Anchor-based Routing Protocol with Cell ID Management System for Ad-Hoc Networks, in Proceedings of International Conference on Computer Communications and Networks, 2005.
9. Giruka VC, Singhal M. A self-healing on-demand geographic path routing protocol for mobile ad-hoc networks, Ad Hoc Netw, 2007.
10. Perkins C, Royer E. Ad-hoc on-demand Distance Vector Routing, Proc. 2nd IEEE Wksp. Mobile Comp. Sys. App., 1999.
11. Johnson D, Maltz D. Mobile Computing, Chap. 5-Dynamic Source Routing, Kluwer Academic Publishers, 1996, pp. 153–81.
12. [12] Z. Haas and M. Pearlman, The Performance of Query Control Schemes for the Zone Routing Protocol, ACM/IEEE Trans. Net, 2001.
13. Lin J, Kuo GS. A novel location-fault-tolerant geographic routing scheme for wireless ad hoc networks, in 2006 IEEE 63rd Vehicular Technology Conference, 2006.
14. Flury R, Wattenhofer R. Randomized 3d geographic routing, in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, 2008.
15. Karp B, Kung HT. Gpsr: Greedy perimeter stateless routing for wireless networks, in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking.