Distance based Greedy Routing in VANET

Ms. Prajakta V Ambekar[1] Prof. Nilesh R. Wankhade[2]
[1]P.G. Student [2]Head of Department
[1,2]Department of Computer Engineering
[1,2]Late. G. N. Sapkal COE

Abstract— Vehicular ad hoc networks (VANETs) have become viable and valuable for their wide variety of novel applications, such as safety of road, content sharing of multimedia, commerce on wheels, etc, using advanced wireless technologies. Distribution of multi-hop information in VANETs is constrained by the high mobility of vehicles and the frequent disconnections. Geographic routing do not require route maintenance and route construction phases hence they are widely adopted for VANETs. With connectivity awareness, they perform well in terms of reliable delivery. Flooding is use in some protocols to obtain destination position. Further in case of regions like sparse and void regions, upraise hop count by frequent use of the recovery strategy. Some geographic routing protocols adopt the minimum weighted algorithm depending on distance or connectivity to select intermediate intersections. However, the path with higher connectivity or the shortest path may include numerous intermediate intersections. Therefore, these protocols yield routing paths with higher hop count. In this paper, we propose a distance based greedy routing and hop greedy routing scheme that yields a routing path with the minimum number of intermediate intersection nodes while taking connectivity into consideration. Moreover, back-bone nodes are introduced that play a key role in providing connectivity status around an intersection. Also the back-bone nodes allow packet to be forwarded in the changed direction by tracking the movement of source as well as destination. Distance based greedy routing overcomes the limitations of a Hop Greedy Routing algorithms. In this paper we have considered design and implementation of routing algorithms that will provide maximum packet delivery ratio and minimum end to end delay.

Key words: Geographic Routing Protocols, Hop Greedy Routing, Multi- hop Information and VANET

I. INTRODUCTION

A. Detail Problem Definition

The Automotive industry is currently undergoing a phase of revolution. Today, a vehicle is not just a thermo mechanical machine with few electronic devices; rather, a late advancement in wireless communication technologies has brought a major transformation of vehicles from a simple moving engine to an intelligent system carrier. A wide spectrum of safety and entertainment services are being driven by a new class of communications that are broadly classified as vehicle-to-vehicle communication and vehicle-to- infrastructure communication. Currently, intelligent transportation system components provide a wide range of services such as freeway management, crash prevention and safety, driver assistance, and infotainment of drivers and/or passengers. Recent trends swing toward business advertisement, and marketing services and products on wheels. Hence, these applications appear to be very remunerative and promising in terms of commerce and research. The remarkable use of vehicular communications in safety and infotainment applications has resulted in the development of a new class of media access control and network layer protocols. The current domain of vehicular research includes safety message broadcast, routing, collision avoidance, congestion control, vehicular sensing, security, etc. Different terrains pose different challenges to vehicular routing. The issues in a city network would not be exactly the same as in a highway or in a delay torrent network. The outskirts may have scarce vehicular density, whereas downtown has to deal with vehicular congestion. The evening may have the highest vehicular traffic, and midnight may be seen with least traffic. It is difficult to predict the exact traffic density of a region. The structure of the road (i.e., road having no deviation or with deviation), number of intersections, number of lanes, length of the road (i.e., based on road ID), public transport availability, and behavior of driver have a great impact on the node density and network connectivity of a vehicular network.

B. Justification of Problem:

In a city network, intersections place a unique challenge to routing protocols. To decide the routing path a routing protocol has to key on some parameters. When the routing path is the shortest distance path, it may involve a very large number of changes of directions, resulting in more number of hop counts. If the connectivity is chosen as the parameter, the most connected road segment would be overcrowded by frequently routing data packets through the same path. As a consequence, the data packets experience longer queuing delays. A third approach suggested in the literature involves broadcasting request messages to fetch the destination position information and connectivity information. However, in a city, flooding is not advisable as multiple nodes would probe for destination position and connectivity information. As a result, every blind search (i.e., flooding) would disrupt all the on-going communications.

C. Need of Proposed System

In this, we choose hop count as the metric to find the routing paths. The hop greedy routing protocol exploits the transmission range and avoids intersections that are used to change the direction of the routing path. It is assured that the selected intersections have enough connectivity. Proactively the sender decides the routing path hence it is not possible to predict the actual connectivity value without probing the whole network. To compute the connectivity parameter we adopt an indirect method for each intersection. We found that connectivity depends on the number of lanes. Hence we obtain the connectivity parameter based on the count of lanes. Packet congestion will occur as the path with the highest connectivity may be used by multiple source destination (srcdst) pairs. Hence, we specify a connectivity threshold, and paths having connectivity parameter beyond this threshold are assigned the same connectivity status. Generally, the multi-constrained optimal path finding problems are known to be NP-hard problems. Hence, we develop an approximation algorithm to choose a path based on both hop count and connectivity. Apart from the routing algorithm, we also introduce a back-bone system in which some specialized nodes perform functions such as tracking the movement of end nodes, detecting void regions on segments of road, storing packets when there is unavailability of forwarding nodes, and selection of the most suitable intersection node as the forwarding node. As the routing algorithm selects a path using destination position, we use a unicast request-reply-based destination probing mechanism. Hence To implement this approach, we divide the city into number of zones that are outlined by the multilane road structures. Some dense intersections (identified as the meeting point of multiple road segments) on the boundary of the zones are chosen as the boundary intersections. As the location of each boundary intersection is known, the unicast request messages can be easily sent to each boundary intersection which is initiated by the source. The back-bone nodes stationed at boundary intersections then take the responsibility to distribute the request messages within the respective zones. The fact that unicast packets do not provide burst traffic and are shielded by request to send/clear to send (RTS/CTS) handshake is the basic motivation to adopt unicast to carry out all control packet forwarding. Once the destination node receives the request message, it searches a suitable path to the source and sends the reply. By receiving the reply message, the source forwards data on the same routing path computed by the hop greedy routing algorithm. Finally, the routing protocol includes an update mechanism that takes care of interzone movement of end nodes.

II. RELATED WORK

A. Study of Existing techniques

V. Naumov, et.al. Propose a "Connectivity-aware routing (CAR) in vehicular ad hoc networks" a position-based routing scheme called Connectivity-Aware Routing (CAR) designed specifically for inter-vehicle communication in a city and/or highway environment. A distinguishing property of CAR is the ability to not only locate positions of destinations but also to find connected paths between source and destination pairs. These paths are auto-adjusted on the fly, without a new discovery process.

C. Lochert, et al., propose "A routing strategy for vehicular ad hoc networks in city environments," This paper present a Routing of data in a vehicular ad hoc network, it is a challenging task due to the high dynamics of such a network. It was shown for the case of highway traffic that position-based routing approaches can very well deal with the high mobility of network nodes. However baseline position-based muting has difficulties to handle two dimensional scenarios with obstacles (buildings) and voids as it is the case for city scenarios.

B. Karp, et al., propose "GPSR: Greedy perimeter stateless routing for wireless networks," a novel routing protocol for wireless datagram networks that uses the positions of routers and a packet's destination to make packet forwarding decisions. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region.

H. Menouar, et al., propose "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks". They present how this MOPR concept can be applied to position-based routing protocols, and how it improves their performances. Based on simulation results they compare MOPR with the position-based routing protocol GPSR and another movement-based routing protocol called MORA.

C. C. Hung, et al., proposes "Mobility pattern aware routing for heterogeneous vehicular networks". This paper proposes a new Heterogeneous Vehicular Network (HVN) architecture and a mobility pattern aware routing for HVN. HVN integrates Wireless Metropolitan Area Network (WMAN) with VANET technology and reserves advantages of better coverage in WMAN and high data rate in VANET. Vehicles in HVN can communicate with each other and access Internet ubiquitously.

J. Zhao and G. Cao, propose "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks", several vehicle-assisted data delivery (VADD) protocols to forward the packet to the best road with the lowest data-delivery delay. Experimental results show that the proposed VADD protocols outperform existing solutions in terms of packet-delivery ratio, data packet delay, and protocol overhead.

Y. Ding, C. Wang, and L. Xiao, propose, "A static-node assisted adaptive routing protocol in vehicular networks", which is a static node assisted adaptive data dissemination protocol for vehicular networks. With the assistance of static nodes at intersections, a packet is forwarded to the static node when there are no vehicles available to deliver the packets along the optimal path. The static node is able to store the packet and transmit it when the optimal delivery path becomes available.

B. Analysis of existing systems/ technologies:

The VANET has witnessed several endeavors toward the development of suitable routing solutions. Originally, many routing protocols were solely designed for mobile ad hoc networks and later enhanced to suit the VANET scenarios. Later on, few novel protocols were developed for adverse VANET environments. Currently, researchers are working on a more concrete version of routing protocols with a higher performance index. However, noteworthy pioneering works such as greedy perimeter stateless routing (GPSR), greedy perimeter coordinator routing (GPCR), geographic source routing (GSR),vehicle assisted data delivery (VADD),anchor-based street- and traffic-aware routing (A-STAR), connectivity-aware routing (CAR),greedy traffic-aware routing (GyTAR), road-based using vehicular traffic (RBVT), static-node-assisted adaptive data dissemination in vehicular networks (SADV), etc. have laid the foundation for routing in VANETs.

C. Comparison of existing systems with proposed system

The position-based routing protocol GPSR relies on the location service to acquire the position information of the destination. Basically, it uses two strategies like greedy forwarding and perimeter routing, to send packets from source to destination. In greedy forwarding, a neighbor node is chosen as the forwarding node if it has the shortest Euclidian distance to the destination among all neighbors. On the other hand, if no neighbor is witnessed closer to the destination than the sender itself, then perimeter routing is applied. In GPCR, packets are forwarded or transferred by applying a restricted greedy forwarding procedure. During the selection of a forwarding node, a junction node termed as the coordinator node is preferred over a non-junction node. Note that the coordinator node is not necessarily the closest node to the destination. However, the recovery strategy in GPCR remains the same as GPSR. The A-STAR features the better use of city bus route information to identify anchor paths. The idea behind such arrangement is that more packets can be delivered to their destinations successfully using paths having more connectivity. Geographic routing uses a static street map and location information about each node. The sender computes a sequence of intersections of road segments using Dijkstra's shortest path algorithm to reach to the destination. The sequence of intersections of road segments is placed in the data packet header. The improved GyTAR is an intersection based geographical routing protocol that finds a sequence of intersections between source and destination considering parameters such as the remaining distance to the destination and the variation in vehicular traffic. The data transferring between the intersections in GyTAR adopts either an improved greedy forwarding mechanism or a carry and forward mechanism, depending upon the presence of the forwarding node.

III. METHODOLOGY

A. Concept

In this section, we present a position-based connectivity aware back-bone-assisted hop greedy (BAHG) routing protocol for VANETs city environments. The proposed routing protocol finds a routing path consisting of the minimum of intermediate intersections. The protocol is designed considering certain features in a city map, such as road segments, intersections etc. To maintain connectivity at the intersections and to detect void regions, we rely on a group of nodes called back-bone nodes. Overall, the goal of the hop greedy routing algorithm is to reduce the hop count, which ultimately reduces the end-to-end delay. In addition, the protocol also ensures successful delivery of data packets to the destinations.

1) Assumptions

It is assumed that each vehicle is aware of its position through the GPS. Moreover, it is assumed that GPS errors are minimized by various standard procedures like augmentation, precise monitoring, time- keeping, and carrier phase tracking. In addition to that, these nodes are equipped with a digital map and a navigation system.

2) Boundary Intersection Selection and Zone Formation

This section explains how a city map is divided into several zones and how some of the intersections are chosen to be the boundary intersections that are located on the outline of a zone. In city maps similar to those shown in Fig.3.4, it is observed that major roads intersect each other and many polygons are formed out of it. By major roads term, we mean roads having more than two lanes. The polygonal areas are called as zones that are the building blocks of a city map. These zones contribute to major roads with the adjacent zones. One such zone formed by four major roads is shown in Fig.3.4. Many minor roads are running inside a zone. By minor roads we mean roads having less than or equal to two lanes. The city map shown in Fig.3.4 is divided into 20 zones. Apart from the corner intersections, major roads may also meet with a cluster of minor roads on the zone border, creating wider intersections. Wider intersections at the corner as well as on the zone border are termed as the boundary intersections. Basically, the boundary intersections will act as the entry points for the packets sent to a zone. In our system, major roads, intersections, and minor roads are assigned unique IDs.

3) Back-Bone Nodes and Connectivity

Connectivity is the key requirement for any routing protocol for reliable and fast delivery of packets. This section describes mechanisms to ensure connectivity of a routing path. A routing path involves many intermediate intersections at which the packet direction is changed. Selection of a wrong intermediate intersection may result in the dropping of packets. Correspondingly, if the source or destination changes its original position, the ongoing communication may get disrupted. Apart from this, the high speed of vehicles may create temporary void regions on a road segment. Therefore, routing paths passing through such road segments are seriously impaired. In our approach, we allow some of the nodes to take care of the foregoing connectivity issues. Such nodes are known as back-bone nodes. Based on their specific action they perform, they are categorized as back-bone nodes at intersection and back-bone nodes at road segments.

a) Back-bone nodes at intersection

These nodes are used to maintain connectivity at an intersection. It is necessity of a back-bone node to declare its presence as soon as it enters the intersection region. For this, the periodic beacons cannot be used because the beacon interval might be larger than the duration of stay of a node at an intersection. To overcome this issue, back-bone nodes use positional beacons.

Back-bone setup: Back-bone nodes of this kind are of three types, namely, primary, secondary and stable back bone. A stable back-bone node is selected from the stream of vehicles waiting at the intersection during red traffic signal. Among the waiting vehicles, the vehicle closest to the intersection declares itself as the stable back bone. However, primary and secondary back bones are selected from the fleet of vehicles crossing the intersection when the signal turns green. The primary back bone is the one located at the intersection, whereas the secondary back bone is outside the intersection. At initial, a random node declares itself as the primary back bone. Then, the primary back-bone node selects a secondary back-bone node comparing the average node speed, the position, and the moving direction of all its neighbours. When the current primary back-bone node leaves the intersection region, it informs the secondary back bone to become the new primary back bone. This notification also notifies vehicles at or around the intersection about the new primary back bone.

Packet forwarding: When there is a need to choose a forwarding node from an intersection of a road segment, a back-bone node is given a preference. This is because back-bone nodes can maintain the communication history and store packet in the absence of a forwarder at the intersection of road segment. A forwarding node checks its neighbour list to probe the available back-bone nodes. It compares the packet forwarding time with the staying time of each back-bone node. If the forwarding node is moving, it chooses stable back-bone nodes as the forwarder. Otherwise, it prefers the moving back bones (i.e., primary and secondary). The primary back bone has higher priority over the secondary back bone. Among the stable back bones, the back bone closest to the intersection has the highest priority.

Message queuing and retrieval: The stable back-bone nodes take the responsibility of packet buffering. In the unavailability of a suitable forwarding node, the packet is stored in a stable back-bone node. On availability of a forwarding node in the desired direction, packet is retrieved and forwarded. The stable back-bone nodes maintain the database of all communications with a timestamp. They store source and destination addresses along with the time of arrival of packets. If a similar packet arrives with a new timestamp, the previous database information is updated.

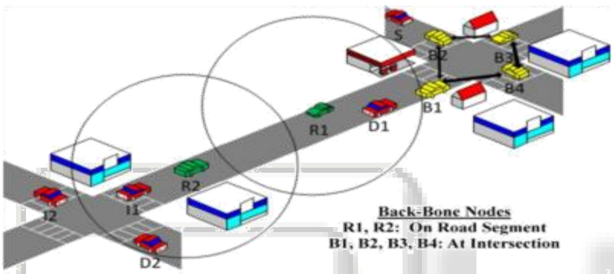

Fig. 3.1: Back-bone nodes engaged in void region detection and forwarding packets at intersections

While a packet is being routed along the selected path, either destination vehicle or source vehicle may change its location and moves to the movements, both source and destination inform about their new road segment, to allow back-bone nodes to keep track of identity in their beacons. Whenever source node or destination changes direction of road segment, the back-bone node modifies or updates the corresponding entry in its communication history. When a packet is being routed, the back-bone nodes provide the updated information. This allows a packet to be forwarded in the new direction. In Fig. 3.1, nodes B1, B2, B3, and B4 represent the back-bone nodes that take care of the activities at intersections.

b) Back-bone nodes at road segment

If any part of a road segment longer than the transmission range is devoid of nodes, this can be noticed by the nodes present at the periphery of the void region. Nodes nearer to the void region from both directions declare themselves as back-bone nodes. These back- bone nodes are termed as void-guard back-bone nodes.

The purpose of a void-guard back-bone node is to inform the presence of a void region to the neighbouring back-bone nodes stationed at intersections. For all such transactions among the back-bone nodes, a piggy- backed beacon message is used. On being aware of an unconnected road segment, the back-bone node at the intersection restricts packets from being forwarded to the identified road segment. In this case, the packet is forwarded by selecting a new route. In Fig, nodes R1 and R2 are back-bone nodes of this type.

B. Algorithm

1) Distance Based Greedy Routing Algorithm

The Distance Based Greedy Routing Algorithm ensures the minimum distance from the source to the destination while considering connectivity. This algorithm is transformed to a single source minimum weight algorithm. The street map is converted into a graph G. All intersections as well as the source and the destination of the data packet are considered as the vertices. For two vertices u and v, there is an edge (u, v) in G if u and v are connected through a road segment.

Each edge is associated with a weight that is the sum of two parameters: hop count and delta count. Let R denote the transmission range. For an edge (u, v) of length L, the hop count is given by L/R + 1 if a direct line-of-sight path exists between u and v. However, for a curved path, the hop count is equal to the possible number of small line-of-sight paths. The delta count parameter represents the degree of disconnection of an edge. It is evident that the higher the number of lanes, the higher the connectivity. We specify a threshold value for the number of lanes, and an edge having a number of lanes higher than the threshold is considered as a connected edge. Otherwise, it is necessary to determine the disconnection level. Let hc and dc denote the hop count and delta count of a road segment. To determine the weight, the delta count can be normalized as dc = σ hc. Note that $\sigma > 1$ as the delay of a disconnected edge is larger than the delay of a connected edge. We derive the value of in the next section.

S denotes the source vertex from which a minimum weight path is sought to the destination vertex denoted as D. The pseudo code of the main algorithm is shown in Fig. 3.3. All notations used in the algorithm are summarized in Table with Figure 3.2. The algorithm starts by initializing the weight of the pseudo-code of the main algorithm is shown in Figure 3.3. S denotes the source vertex from which a minimum weight path is sought to the destination vertex denoted as D. All notations used in the algorithm are summarized in TABLE with Figure 3.2. The algorithm starts by initializing the weight of all vertices in V [G] using the initialization routine shown in Figure 3.3.Initially, the weight estimate of each vertex is set to ∞, except the source vertex whose weight estimate is set to 0. The predecessor of each vertex denoted as Prev[v] is also initialized. In Figure 3.3, line 2 shows initializing the set of visited vertices as an empty set and in line 3, the min-priority queue is populated with the vertices in V [G]. The vertices are keyed by their weight which is the sum of their hop-count and delta-count estimates. Starting with the source vertex, in each step vertex having smallest weight is extracted from the min-priority queue and the weight estimates and predecessors of its neighbors are updated by executing H-RELAX procedure whose pseudo-code is given in Figure 3.3.

Suppose a vertex u executes H-RELAX to relax an edge (u, v). Suppose vertex k represents the predecessor of u. If k and v are in the line-of-sight or they share the common road ID (i.e. they are located on a same major road), then k will be chosen as the predecessor of v. Otherwise, u will be chosen as the predecessor. The potential predecessor is stored in variable Tp rev. This step aims at minimizing the number of intermediate intersections along a path. In other words, the algorithm seeks to find a straightest possible path between source and destination. It is observed that selecting more number of intermediate intersections results in a longer path. The actual predecessor is decided only after the new weight of v is computed and is compared against the stored value. Line 8 in Figure 3.3 estimates the weight for v considering TPrev as its immediate predecessor in the path from S to v. For the edge (TPrev, v), the hop count is obtained as described above, whereas the delta-count is computed using the pseudo-code given in Figure 3.3.

If the estimated weight of v is less than the stored weight, then TPrev becomes the predecessor of v and its weight components are updated. In this way, the H- RELAX procedure processes all the neighbors of the vertex having minimum weight. Note that the functions INSERT and DECREASE-KEY are implicit in line 3 and line 8 of Figure.3.3 respectively.

H – INITIALIZE (G, s)

- Step 1: for each vertex $v \in V[G]$
- Step 2: do $Prev[v] \leftarrow Nil$
- Step 3: $W_t[v] \leftarrow \infty$
- Step 4: $W_t[s] \leftarrow 0$

Initialization procedure

DELTA – COUNT (v, hc)

- Step 1: $f \leftarrow$ LANE (v)
- Step 2: if $f > th$ then return 0
- Step 3: else return $\sigma * hc$

Procedure to evaluate connectivity (delta count)

H- RELAX (u, v)

- k← Prev[u]
- LS←LOS(k, v)
- If k! = Nil AND LS= true
- Then *T*_ Prev← k
- else *T*_ Prev← u
- hc← HOP- COUNT(*T*_ Prev, v)
- dc← DELTA- COUNT (v, hc)
- *T*_ Wt[v] = Wt[*T*_ Prev]+hc +dc
- if Wt[v] > *T*_ Wt[v]
- then Wt[v] ← *T*_ Wt[v]
- Prev[v] ← *T*_ Prev

Relax Procedure

H- DIJKSTRA (G, s)

H- INITIALIZE (G, s)

- VS← Φ
- Q← V[G]
- While Q! = Φ
- do u = EXTRACT- MIN(Q)
- VS= VS U {u}
- for each vertex v € *Adj*[u]
- do H- RELAX (u, v)

Fig. 3.3: Hop Greedy Routing Algorithm

Notation	Definition
V[G]	Set of vertices of graph *G*
E[G]	Set of edges in graph *G*
Q	Min-Priority Queue formed using *V[G]*
b,c	Any two vertices in *V[G]*
Prev[b]	Predecessor of vertex *b*
Wt[b]	Weight of vertex *b*
T_Prev	Temporary vertex
T_Wt[b]	Temporary weight of a vertex *b*
VS	Set of vertices already visited
Adj[b] (*Adj*:Adjacency Matrix of Graph *G*)	Set of all vertices *c* such that there is an edge *(b, c)* in *E[G]*
LOS*(b, c)*	Returns true if vertex *b* and vertex *c* are in line-of-sight or they share a common Road ID
HOP-COUNT (*b*, c)	Returns number of hops in the edge formed by vertex *b* and vertex *c*.
LANE (*c*)	Returns number of lanes connected at the vertex *c*.
th	Threshold for number of lanes connected at a vertex.
σ	Ratio of delta-count and hop-count
hc	hop-count of a road segment.
dc	delta-count of a road segment.

Fig. 3.2: Summary of Notation

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the Distance based greedy routing protocol. Distance based greedy routing protocol is compared with Hop greedy routing protocols.

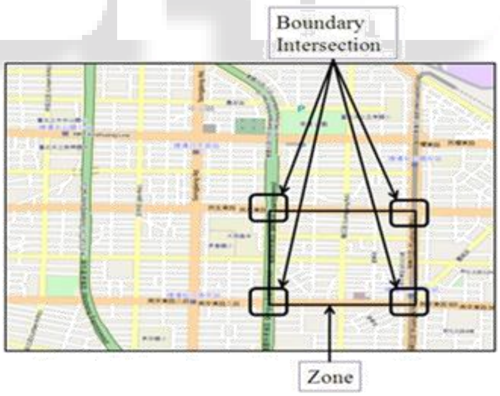

Fig. 3.4: Map extracted from Open Street Map Database (city scenario)

A. *Simulation Environment*

The simulation scenario is shown in Fig. 3.4. It represents an area of 3 km × 3 km of New York City. We designed the scenario in java and generated vehicle movements. A total of 100 vehicles are generated by using java. Once generated, they start moving along the specified routes while taking turns at the intersections.

B. *Routing Metrics*

- Packet delivery ratio: It is the ratio of the total number of packets received at the destination to the total number of packets generated by the source.
- End-to-end delay: This is the delay elapsed

between the packet generation at the source and successful reception at the destination.

C. Result Analysis

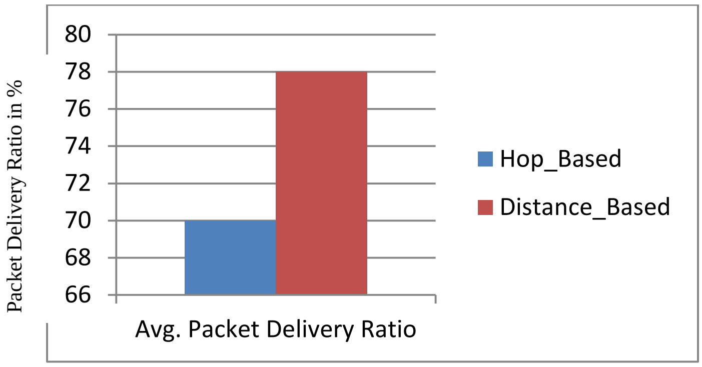

Fig. 3.5: Packet Delivery Ratio

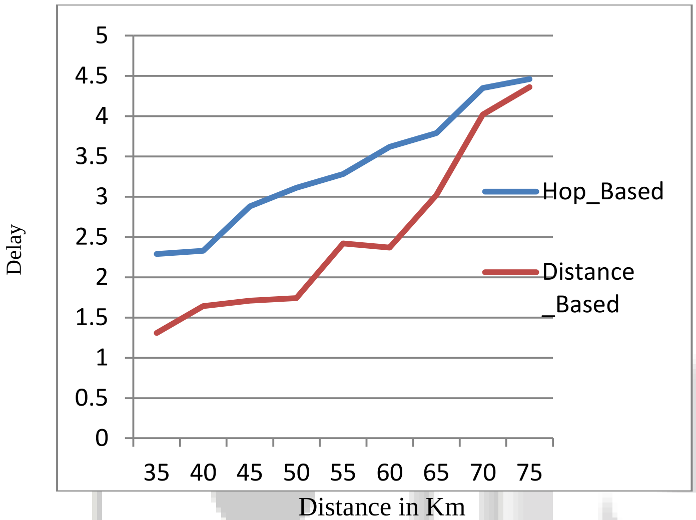

Fig. 3.6: End-to-end delay

V. CONCLUSION

We propose a distance based greedy routing protocol that aims to reduce the end-to-end delay by yielding a routing path that includes the minimum number of intermediate intersections. The zone wise partitioning of a city road network is an important design framework for the efficient functioning of the destination discovery procedure. The greedy algorithm based on distance finds the best possible path in terms of distance, hop count and connectivity. To address connectivity issues such as void regions and unavailability of forwarders, the system of back-bone node is used. Moreover, by implementing unicast messages, the distance based greedy routing scheme eliminates packet loss and congestion noticed in contemporary routing protocols that use broadcast request messages.

REFERENCES

[1] Pratap Kumar Sahu, et. al, "BAHG: Back-Bone-Assisted Hop Greedy routing for VANET's City Environments", IEEE Transactions on intelligent transportation system, VOL. 14, NO. 1, MARCH 2013
[2] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive connectivity aware routing for vehicular ad hoc networks in city scenarios" Mob. Netw. Appl., vol. 15, no. 1, pp. 36–60, Feb. 2010.
[3] V. Naumov, et.al, "Connectivity-aware routing (CAR) in vehicular ad hoc networks" in Proc. IEEE INFOCOMM, 2007, pp. 1919–1927.
[4] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments" in Proc. IEEE Intell. Veh. Symp. 2003, pp. 156–161.
[5] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks" in Proc IEEE VTC, 2007, pp. 2101–2105.
[6] C. C. Hung, et.al, "Mobility pattern aware routing for heterogeneous vehicular networks," in Proc. IEEE WCNC, 2008, pp. 2200–2205.
[7] K. C. Lee, J. Häerri, U. Lee, and M. Gerla, "Enhanced perimeter routing for geographic forwarding protocols in urban vehicular Scenarios," in Proc. IEEE GlobeCom Workshops, 2007, pp. 1–10.
[8] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," IEEE Trans. Veh. Technol., vol. 57, no. 3, pp. 1910–1922, May 2008.
[9] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in Proc. 2nd IEEE Workshop Mob. Comput. Syst. Appl., 1999, pp. 90–100.
[10] H. Menouar, M. Lenardi, and F. Filali, "Improving proactive routing in VANETs with the MOPR movement prediction framework," in Proc. ITST, 2007, pp. 1–6.
[11] Y. Ding, et.al, "A static-node assisted adaptive routing protocol in vehicular networks," in Proc. ACM VANET, 2007, pp. 59–68.
[12] M. Jerbi, S. M. Senouci, T. Rasheed, and Y. Ghamri-Doudane, "Towards efficient geographic routing in urban vehicular networks". IEEE Trans. Veh. Technol., vol. 58, no. 9, pp. 5048–5059, Nov. 2009.
[13] U. Lee, J. Lee, J. S. Park, and M. Gerla, "FleaNet: A virtual market place on vehicular networks," IEEE Trans. Veh. Technol., vol. 59, no. 1, pp. 344–355, Jan. 2010.
[14] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," IEEE Trans. Veh. Technol., vol. 58, no. 7, pp. 3609–3626, Sep. 2008.