Back-Bone Assisted HOP Greedy Routing for VANET

Naresh Kumar[1] D. Asha[2] Dr. V. S. K. Reddy[3]
[2]Assistant Professor [3]Principle
[1,2,3] Department of Electronics and Communication Engineering
[1,2,3] MRCET, Secunderabad, Telangana, India

Abstract— Using advanced wireless local area network technologies, vehicular ad hoc networks (VANETs) have become viable and valuable for their wide variety of novel applications, such as road safety, multimedia content sharing, commerce on wheels, etc., currently, geographic routing protocols are widely adopted for VANETs as they do not require route construction and route maintenance phases. Again, with connectivity awareness, they perform well in terms of reliable delivery. Further, in the case of sparse and void regions, frequent use of the recovery strategy elevates hop count. Some geographic routing protocols adopt the minimum weighted algorithm based on distance or connectivity to select intermediate intersections. However, the shortest path or the path with higher connectivity may include numerous intermediate intersections. As a result, these protocols yield routing paths with higher hop count. In this paper, we propose a hop greedy routing scheme that yields a routing path with the minimum number of intermediate intersection nodes while taking connectivity into consideration. Moreover, we introduce back-bone nodes that play a key role in providing connectivity status around an intersection. Apart from this, by tracking the movement of source as well as destination, the back-bone nodes enable a packet to be forwarded in the changed direction. Simulation results signify the benefits of the proposed routing strategy in terms of high packet delivery ratio and shorter end-to-end delay.

Key words: Greedy Routing, VANET, Back-Bone.

I. INTRODUCTION

Ad hoc networks also called infrastructure less networks are complex distributed systems consist of wireless links between the nodes and each node also works as a router to forwards the data on behalf of other nodes. The nodes are free to join or left the network without any restriction. Thus the networks have no permanent infrastructure. In ad hoc networks the nodes can be stationary or mobile. Therefore one can say that ad hoc networks basically have three forms, one is static ad hoc networks (SANET), mobile ad hoc networks (MANET) and the other one is Vehicular ad hoc networks (VANET).

VANET means vehicular Adhoc network and it is the technology which is used to move vehicles as joint in network to make a transportable network. Participating vehicles become a wireless connection or router through vanet and it allow the vehicles almost to connect 100 to 300 meters to each other and in order to create a wide range network, other vehicles are connected to each other so the mobile internet is made. It is supposed that the first networks that will incorporate this technology are fire and police mobiles to interact with one another for security reasons. Brilliant way to use Vehicular Networking is defined in VANET or Intelligent Vehicular Ad-Hoc Networking. Multiple ad-hoc networking technologies integrated in VANET such as, ZigBee, IRA, WiMAX IEEE, and Wi-Fi IEEE for convenient, effective, exact, simple and plain communication within automobiles on active mobility. Useful procedures like communication of media within automobiles can be allowed as well process to follow the automotive automobiles are also favored. Security measures are defined in vehicles by VANET, flowing communication within automobiles, edutainment and telemetric.

II. SYSTEM ANALYSIS

A. *Existing system*

Originally, many routing protocols were solely designed for mobile ad hoc networks and later enhanced to suit the VANET scenarios. Number of Existing Routing protocols like GPSR work well in city environments. However, these protocols encounter different problems that motivate us to design a new robust scheme.

1) *Demerits*
- Intersection Node Probing Problem
- Location Service Requirement Problem

B. *Proposed system*

In this section, we present a position-based connectivity aware back-bone-assisted hop greedy routing protocol for VANET's 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.

1) *Merits:*
- The zone wise partitioning of a city road network is an important design frame- work for the efficient functioning of the destination discovery procedure.
- The hop greedy algorithm finds the best possible path in terms of both hop count and connectivity.

C. *Routing Protocols*

An ad-hoc routing protocol is a convention, or standard, that controls how nodes decide which way to route packets between computing devices in a mobile ad hoc network .In ad-hoc networks, nodes are not familiar with the topology of their networks. Instead, they have to discover it. The basic idea is that a new node may announce its presence and should listen for announcements broadcast by its neighbors. Each node learns about nodes nearby and how to reach them, and may announce that it, too, can reach them.

Note that in a wider sense, ad hoc protocol can also be used literally, that is, to mean an improvised and often impromptu protocol established for a specific purpose.

III. SYSTEM DESIGN

People communicate. One way or another, they exchange some information among themselves all the times. In the past several decades, many electronic technologies have been invented to aid this process of exchanging information in an efficient and creative way. Among these is the creation of fixed telephone networks, the broadcasting of television and radio, the advent of computers, and the emergence of wireless sensation. Originally, these technologies existed and operated independently, serving their very own purposes. Not until recently that these technological wonders seem to converge, and it is a well-known fact that a computer communication network is a result of this convergence.

System modeling refers to an act of representing an actual system in a simply way. System modeling is extremely important in system design and development, since it gives an idea of how the system would perform if actually implemented.

Traditionally, there are two modeling approaches: analytical approach and simulation approach.

Analytical Approach: The general concept of analytical modeling approach is to first come up with a way to describe a system mathematically with the help of applied mathematical tools such as queuing and probability theories, and then apply numerical methods to gain insight from the developed mathematical model. When the system is simple and relatively small, analytical modeling would be preferable (over simulation). In this case, the model tends to be mathematically tractable. The numerical solutions to this model in effect require lightweight computational efforts.

If properly employed, analytical modeling can be cost-effective and can provide an abstract view of the components interacting with one another in the system. However, if many simplifying assumptions on the system are made during the modeling process, analytical models may not give an accurate representation of the real system.

Simulation Approach: Simulation is a process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system. Simulation is widely-used in system modeling for applications ranging from engineering research, business analysis, manufacturing planning, and biological science experimentation, just to name a few. Compared to analytical modeling, simulation usually requires less abstraction in the model (i.e., fewer simplifying assumptions) since almost every possible detail of the specifications of the system can be put into the simulation model to best describe the actual system. When the system is rather large and complex, a straightforward mathematical formulation may not be feasible. In this case, the simulation approach is usually preferred to the analytical approach.

In common with analytical modeling, simulation modeling may leave out some details, since too many details may result in an unmanageable simulation and substantial computation effort. It is important to carefully consider a measure under consideration and not to include irrelevant detail into the simulation.

In this section, we present a position-based connectivity aware back-bone-assisted hop greedy (BAHG) routing protocol for VANET's 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. Basically, we adopt a request-reply scheme to obtain destination position, which is then used to compute the routing path.

To avoid the impact of mobility on routing decisions, an update procedure is specifically designed to supervise the movement of source as well as destination. Overall, the objective 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.

A. Zone Formation and Boundary Intersection Selection

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 below,it is observed that major roads intersect each other, and many polygons are formed out of it. By "major roads," we mean roads having more than two lanes. The polygonal areas are termed as zones that are the building blocks of a city map. These zones share major roads with the adjacent zones. One such zone formed by four major roads is shown below.

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 below is divided into 20 zones. At the corner of each zone, wider intersections are witnessed. As major roads meet there, it is highly probable that at least one node will be present at that intersection. 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, intersections, major roads, and minor roads are assigned unique IDs.

B. Back-Bone Nodes and Connectivity Preservation

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. Similarly, if the source or destination changes its original position, the ongoing communication may get disrupted. Apart from this, the high mobility of vehicles may create temporary void regions on a road segment.

As a result, 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 called as back-bone nodes. Based on the specific action they perform, they are classified into back-bone nodes at intersection and back-bone nodes at road segments.

C. Back-bone setup

Back-bone nodes of this kind are of three types, namely, stable, primary, and secondary back bones. 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. Initially, a random node declares itself as the primary back bone. Then, the primary backbone node selects a secondary back-bone node comparing the average vehicle speed, the position, and the moving direction of all its neighbors.

When the current primary back-bone node leaves the intersection region, it notifies the secondary back bone to become the new primary back bone. This notification also informs vehicles at or around the intersection about the new primary back bone.

1) Packet forwarding

When there is a need to choose a forwarding node from an intersection, a back-bone node is always preferred. This is because back-bone nodes can maintain the communication history and store packet in the absence of a forwarder at the intersection. A forwarding node checks its neighbor 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 prefers 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.

2) Message queuing and retrieval

The stable back-bone nodes take the responsibility of packet buffering. In the absence 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. While a packet is being routed along the selected path, either destination or source may change its position and moves to a new road segment.To allow back-bone nodes to keep track of the movements, both source and destination inform about their identity in their beacons. Whenever source or destination changes direction, the back-bone node updates the corresponding entry in its communication history. When a packet is being forwarded, the back-bone nodes provide the updated information.

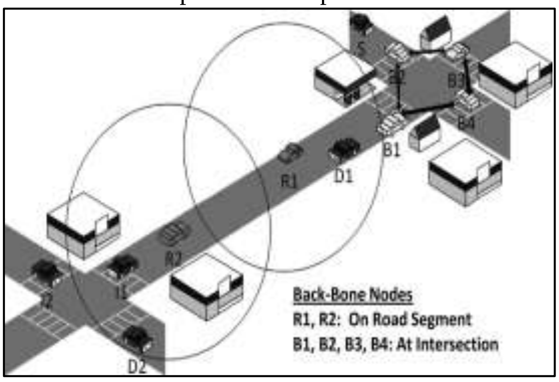


Figure.1 Road Segments and Intersection

This enables a packet to be forwarded in the new direction. In below figure nodes B1, B2, B3, and B4 represent the back-bone nodes that take care of the activities at intersections.

3) Back-bone nodes at road segment

If any part of a road segment longer than the transmission range is devoid of nodes, it can be noticed by the nodes present at the periphery of the void region. Nodes closest to the void region from both directions declare themselves as back-bone nodes. These backbone 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 neighboring back-bone nodes stationed at intersections. For all such transactions among the back-bone nodes, a piggybacked beacon message is used. On being aware of an unconnected road segment, the back-bone node at the intersection prohibits packets from being forwarded to the identified road segment. In this case, the packet is forwarded by selecting a new route. In above fig, nodes R1 and R2 are back-bone nodes of this type.

4) BAHG Position Update

Before receiving the reply message, the source may change its position. Some back-bone nodes must be aware of the direction of the source movement. When a forwarder chosen among the back-bone nodes learns about such changes, it forwards the reply message toward the new direction. Ultimately, the source is able to receive the reply message. Likewise, the destination may change its position before receiving the data packet, and its movements are tracked by the back-bone nodes. The destination may move substantially far from its original position.

In such cases, the hop count will be elevated if the packet is forwarded using the updates received from the back-bone nodes. Thus, a fresh reply message is forwarded to the source if the destination changes its zone. On receiving this reply message, the source can compute a better path to the destination. This can marginalize the hop count, irrespective of the destination movement.

IV. SOFTWARE REQUIREMENTS

A. Hardware:

- Single PC
- 20 Gb Hard disc space
- 1Gb RAM
- Mouse : Logitech

B. Software:

- Linux OS (Ubuntu 10.04)
- NS 2.34

C. Language:

- TCL (front end)

V. TESTING & DEBUGGING

Testing and debugging a program is one of the most tedious parts of computer programming. The testing and debugging phase of a project can easily take more time than it took to write the application. Testing includes both checking that the code runs at all, that it runs correctly under all circumstances, and that it runs the same way it did before you made changes.

Tcl's error diagnostics make it easy to track down coding errors; the modular nature of Tcl code makes it easy to do unit testing of functions, and the tcl test package makes it easy to write integrated regression test suites.

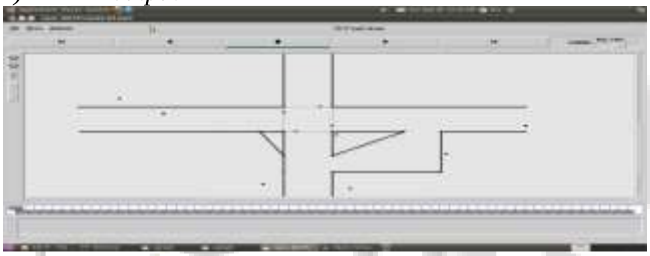
A. Debugging Code

The first step to debugging a Tcl script is to examine the Tcl error output closely. Tcl provides verbose error information that leads you to the exact line where a coding error occurs. Tcl error messages consist of a set of lines. The first line will describe the immediate cause of the error (Incorrect number of arguments, invalid argument, undefined variable, etc). The rest of the message describes more details about where the error occurs.

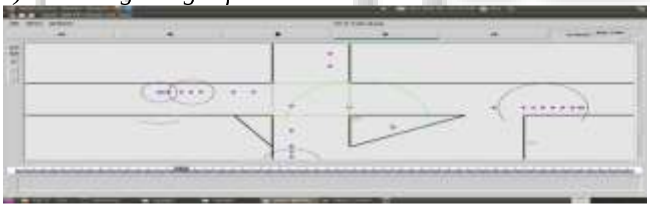
B. Screenshots

1) The output is generated as NAM and X-graph

2) NAM output



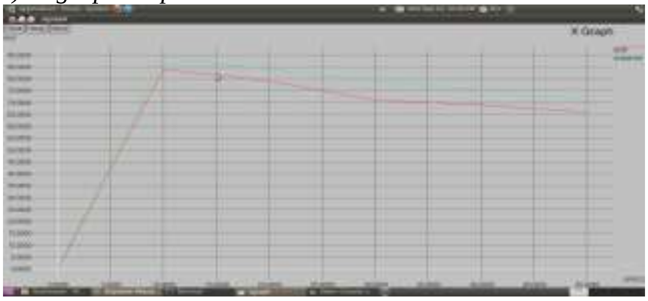
3) Showing Ranges for Vehicles and RSU



4) Data transmission from one vehicle to another vehicle via RSU

5) Xgraph Output 1

6) Xgraph output 2



VI. CONCLUSION

The VANET has witnessed several endeavors toward the development of suitable routing solutions. Multi-hop information dissemination in VANETs is constrained by the high mobility of vehicles and the frequent disconnections.

In this paper, we propose a hop greedy routing scheme that yields a routing path with the minimum number of intermediate intersection nodes while taking connectivity into consideration. Moreover, we introduce back-bone nodes that play a key role in providing connectivity status around an intersection. Apart from this, by tracking the movement of source as well as destination, the back-bone nodes enable a packet to be forwarded in the changed direction.

REFERENCES

[1] X. Jiang and T. Camp, “An efficient location server for an ad hoc net- works” Colorado School Mines, Golden, CO, Tech. Rep., MCS-03-06, May 2003.

[2] 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.

[3] J. Bernsern and D. Manivannan, “Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification” Pervasive Mob. Comput., vol. 5, no. 1, pp. 1–18, Feb. 2009.

[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] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks” in Proc. ACM MOBICOM, 2000, pp. 243–254.

[6] G. Liu, B. S. Lee, B. C. Seet, C. H. Foh, K. J. Wong, and K. K. Lee,“A routing strategy for metropolis vehicular communications” in Proc. ICOIN, LNCS, Aug. 2004, pp. 134–143.

