



An Investigation on VANET Geographic Junction Based Multipath Routing Algorithm

Rohan Verma¹, Prof. M. M. S Rauthan², Varun Barthwal³

^{1,3} Dept of Information Technology, H.N.B Garhwal Central University, India

² Head Dept of Information Technology, H.N.B Garhwal Central University, India

Abstract— *In this paper we like to introduce a routing technique designed for VANET. The algorithm was named Geographic Junction-based Multipath Source Routing (GJMR). Its core features comprises the multiple routes headed for the destination, the junction-based logic and the adoption of geographic source routing mechanisms.*

Keywords— *VANET, GJMR, Geographic, Multipath routing, GPSR*

I. INTRODUCTION

This document is a template. Vehicular Ad Hoc Networks (VANET) has been proposed as a technology that will enable connectivity of users on-the-move and also implement Intelligent Transportation Systems (ITS). The latter is a technology that refers to smart transportation information collection and dissemination, such as cooperative traffic monitoring, collision prevention and real-time detour computation [1]. VANET resemble classical Mobile Ad-hoc Networks (or MANETs), but they also have some unique characteristics. The most important one is the geographically constrained topology. In VANET, nodes cannot freely move around an area or surface; their movements are restricted within the roads formed around obstacles such as buildings. These obstacles have also a great effect in the transmitted wireless signals, causing large distortions to them. A side-effect of this is the partial predictability of nodes' movements. Another interesting characteristic is the large scale of the network, as VANET may comprise hundreds or thousands of nodes that may be too close together or too far away. Finally, in VANET the nodes' power consumption is not a critical design parameter [2].

The vast majority of routing protocol for VANET so far, such as Greedy Perimeter Coordinator Routing [3], Geographic Source Routing [4] or Connectivity-Aware Routing [5], uses only one single route from the source to destination. GPSR [13] handles each packet separately. Some of its variations, like GPCR-MA [6] and [14], exploit the use of additional information, like electronic maps and traffic, in order to improve GPSR's performance. However, in [7] the authors investigate the advantages and disadvantages of multiple nodes disjoint routes in VANET. Some of their main conclusions were that: single-path and multipath have similar performance when source and destination are only a few (2-3) hops away, but for larger source-destination distances (4-5 hops) some difference is observed; route coupling plays a significant role; for long source-destination distances (5 hops or more) better performance may be achieved by using multiple node-disjoint paths, especially in the case of high data rates. In the authors propose a variation of the AOMDV protocol that exploits vehicles' speed in order to optimize the best route decision. Finally, in Fast Restoration On-demand Multipath Routing (FROMR) is proposed, which maintains partially link-disjoint multipath routes for link failure recovery.

In this work, we present our efforts to design a VANET routing protocol not based on MANET protocols variations, but based on the characteristics of urban environments from the very beginning. Due to size limitations, we will only present the basic idea and some preliminary results based on simulations of realistic vehicular environments. We call our protocol Geographic Junction-based Multipath Source Routing, or GJMR for short. GJMR is a geographic routing protocol, in the sense that it exploits the location of the nodes and also of the street junctions, known via digital street maps. It maintains concurrently two paths from the source to the destination as a series of junctions the packets should pass through, and not as a series of nodes-relays. GJMR injects the routing information inside each packet, according to the source routing paradigm, so that every node on the path is aware of the route the packets should follow.

The remaining of this letter is structured as follows: After briefly explaining the protocol's main characteristics in Section II, Section III describes the protocol's main procedures.

II. GJMR'S CHARACTERISTICS

As per the previous studies. We assume that each node is equipped with a GPS receiver, thus always knowing its position. Furthermore, each node has a digital map of the city streets (some commercial vehicles are already equipped with such systems). Hello packets (including position information) are used so that each node is aware of the number and position of its neighbours. Finally, we assume that the vehicles' density is relatively high, as in urban environments during rush hours.

GJMR is characterised as junction-based because it is a geographic or position-based routing protocol, where the junctions' positions are of much higher importance than the positions of the nodes themselves. Using the location information of the junctions instead of the nodes' has some attractive advantages: on one hand we avoid routing loops and local maxima/minima, while on the other hand nodes' movements have much less influence on routing decisions and paths are much more reliable in that sense. Packet forwarding is handled by the nodes within the junctions. All nodes in the junctions are potential forwarders and the selection is done randomly.

Multiple paths have been proposed in order to distribute the traffic load to a wider area, such that better performance can be achieved. GJMR is a multipath protocol; however, multipath refers to paths comprising series of junctions and not series of nodes, as usual. Hence, GJMR's paths are not only link and node disjoint, but also they have a significant probability to be interference-disjoint too, since buildings prevent communication among nodes on either side of them; GJMR exploits this fact.

Finally, GJMR adopts the source routing concept. Each data packet carries forwarding information, injected in the packet by the source node. Unlike other source routing protocols, the injected information refers not to the nodes but to the junctions a packet should visit. Thus, we need no routing packets to maintain routes. Finally, more freedom is provided to forwarding nodes for more appropriate next hop selection, since they have better knowledge of their neighbourhood than the source node, given the high node mobility.

III. DESCRIPTION OF GJMR

A detailed description of our protocol. Similar to many previous works, we assume that the position of the destination node is a-priori known to source node via an appropriate mechanism (e.g. RLS – Reactive Location Services). Although such a mechanism affects the performance of the overall system, it is orthogonal to the design and performance of the routing protocol.

A. At the source node

In GJMR, the source node is responsible for determining the routes towards the destination. Taking into account its own position, the destination's position and the junctions in the area between and around these two nodes, the source node calculates two routes towards the destination¹. Each route is composed of a sequence of junctions. After calculating the two routes (the exact method will be discussed later), the source node, appends to the header of each data packet the coordinates of the junctions of the route the packet should follow. In this way, every forwarding node will be aware of the geographical area that they should search for the appropriate next hop. Then, the source node behaves as a forwarding node: it searches among its neighbours for one that resides in the first junction and sends the packet to it. Since GJMR uses two paths, the next data packet is routed via the other path. In this way, the traffic load is distributed between the two routes.

B. Forwarding packets

When an intermediate node receives a data packet, it looks at the packet's header to determine the location of the next junction on the path. Then, it searches for an available node at that junction, and forwards the packet to it. If no next junction is found, it means that the destination is supposed to be the next hop (and also its neighbour), so it just passes the packet to it. The selection of the next hop node inside the specified junction follows no specific rule; the choice is random. In case no node is found to reside in the junction, the packet is dropped. At this initial version of the algorithm, we have

Adopted no recovery mechanism. Nevertheless, in a crowded city centre this case is practically rare, as most vehicles are stuck in junctions, due to city traffic lights or signs. We intend to add a recovery mechanism in the future, to enhance the algorithm and permit it to work in networks with smaller density and in cases where the distance between two junctions is larger than the transmission range of the forwarding node. An idea would be to search for nodes that are in the middle of a road, or even to wait for some time for a node to advertise itself as residing inside a junction, and then forward the packet, or a carry-and-forward technique as the one proposed in [12].

C. Finding multiple disjoint routes

As mentioned earlier, the source node stores two routes towards the destination. The aim of the protocol is to store routes that are as interference-disjoint as possible, so that they do not interfere with one another. GJMR decisions on which routes to choose make use of the well-known Dijkstra's shortest path algorithm applied on graphs that model the city environment. The graph is based on the digital city map assumed to exist in every vehicle. The map's junctions correspond to the vertices of the graph. We also add two more vertices, which correspond to the source and destination node. The edges of the graph are defined as following: Supposing that two vehicles lie on two different junctions of the city, if they can communicate with each other, based on the capabilities of the physical layer, then the two vertices of the graph, which correspond to these junctions, are assumed to be connected by an edge. The same rule is used to define the edges between the source or destination vertex and the other vertices of the graph. The communication range can never be guaranteed, of course; however a good estimation of the achievable range at the physical layer could be calculated based on historical data of the node's neighbours and their distance from the node itself.).

The size of the map's area that we will transform into a graph, where the source is going to search for routes, is crucial. We use the rectangular area of the map defined by the two furthest points among the source node, the destination node, the junctions between them and the neighbouring junctions of the source and destination node. In this way we avoid

losing routes that overcome some peculiarities in the topology and provide the shortest path algorithm with more available routes, thus increasing the probability of finding interference-disjoint paths.

Once the graph is derived, each edge is assigned with cost $c_1 = c_{init} = 1$. This is the initial graph G_1 . On G_1 , starting from each vertex that is source node's neighbour, we use a shortest path algorithm (e.g. Dijkstra's algorithm) and determine the minimum cost route. To this route, we add the source node itself. So, we get a first route R_1 . Next, we formulate a new graph G_2 by removing R_1 from graph G_1 . This is a common practice in multipath route discovery on graphs [14]. There is another modification on the initial graph that we make. Since the second route should be as interference-disjoint as possible from the first one, we assign new costs on the graph's edges, whose one endpoint is a neighbour of a vertex belonging to R_1 . On these edges we add an extra cost $c = C_{add}$, so that the algorithm avoids the corresponding vertices when calculating the second route. However, we do not completely remove them, because there are cases where no other route would be available otherwise or cases where these vertices should be used, since the alternative is even more costly. Hence, the new costs of the graph's edges will be $c_2 = C_{init} + C_{add}$. When C_{add} is small, the two routes will be highly correlated but the total number of hops will be small. If C_{add} is high, the two routes will be independent; however, the total number of hops on the second route will be much higher than those on the first route. So, there is a compromise that should be made. After changing the edges' costs, we are ready to calculate the shortest path on the new graph G_2 , which will provide us with the second route R_2 .

Up to this step, we have found as many couples of routes as the number of the neighbouring vertices of the source node's vertex. We choose the best among the couples, which is the one that has the smallest total cost.

Finally, the update of the routes takes place each time we receive an update regarding the position of the destination node. Let us mention here that this is also an advantage over the traditional perspective of node based protocols. The latter need to recalculate the routes every time an update for a neighbouring node is received. On the contrary, JMSR is junction-based, so the update needs to be made only when the position of the destination node is changed, irrespective of the positions of the intermediate nodes.

IV. CONCLUSIONS

In this paper, we introduced a VANET geographic junction based multipath routing protocol, where other paths are disjoint. The general conclusion from our investigation is that junction based multipath is beneficial for VANET, in case the source-destination distances are middle or higher (At least 5 hops away or more), or traffic loads are medium to dense, conditions that real-world VANET will probably face. As a future work, we intend to add a recovery Technique to overcome or wrap the cases where no suitable forwarding nodes have been found. Moreover, we are aiming towards the hybrid algorithm that uses unidirectional path for short distances and for the low weight traffic for the multipath uses light.

REFERENCES

- [1] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, pp. 12–22, 2007.
- [2] I. Broustis and M. Faloutsos, "Routing in vehicular networks: Feasibility, modelling and security," *International Journal of Vehicular Technology*, vol. 8, Mar. 2008.
- [3] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, pp. 69–72, Jan. 2005.
- [4] C. Lochert, H. Hartenstein, J. Tian, H. Fuessler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proc. IEEE Intelligent Vehicles Symposium (IV2003)*, Jun. 2003, pp. 156–161.
- [5] V. Naumov and T. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *Proc. 26th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AK, USA, May 2007, pp. 1919–1927.
- [6] F. Granelli, G. Boato, D. Kliazovich, and G. Vernazza, "Enhanced gpsr routing in multi-hop vehicular communications through movement awareness," *IEEE Communications Letters*, vol. 11, pp. 781–783, 2007.
- [7] X. Huang and Y. Fang, "Performance study of node-disjoint multipath routing in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1942–1950, May 2009.
- [8] C.-S. Wu, S.-C. Hu, and C.-S. Hsu, "Design of fast restoration multipath routing in VANET," in *Proc. International Computer Symposium (ICS2011)*, Tainan, Taiwan, Dec. 2010.
- [9] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, Apr. 2006.
- [10] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125–145, 1974.
- [11] P. Sermpezis, G. Koltsidas, and F.N. Pavlidou, "Investigating a Junction-based Multipath Source Routing algorithm for VANET," *communication letters, IEEE*, mar, 2013.
- [12] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, Apr. 2006.
- [13] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. ACM/IEEE International Conference on Mobile Computing and Networking*, Boston, MA, USA, Aug. 2000.
- [14] S. Sun, J. Hu, X. Luo, and Q. Wang, "Improved gpsr in inter-vehicle communication," in *Proc. International Conference on Communications and Mobile Computing*, Shenzhen, China, Apr. 2010, pp. 259–265.