

Investigating a Junction-Based Multipath Source Routing Algorithm for VANETs

Pavlos Sermpezis, Georgios Koltsidas, and Fotini-Niovi Pavlidou, *Senior Member, IEEE*

Abstract—In this letter we introduce a new routing technique designed exclusively for VANETs and present some initial performance results. The algorithm was named **Junction-based Multipath Source Routing (JMSR)**. Its main characteristics comprise the multiple routes towards the destination, the junction-centric logic and the adoption of source routing mechanisms.

Index Terms—VANETs, routing, multipath, geographic, junction-based.

I. INTRODUCTION AND RELATED WORK

VEHICULAR Ad Hoc Networks (VANETs) have 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]. VANETs 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 VANETs, 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 VANETs may comprise hundreds or thousands of nodes that may be too close together or too far away. Finally, in VANETs the nodes' *power consumption* is not a critical design parameter [2].

The vast majority of routing protocol for VANETs so far, such as Greedy Perimeter Coordinator Routing [3], Geographic Source Routing [4] or Connectivity-Aware Routing [5], use only one single route from the source to destination. GPSR [6] handles each packet separately. Some of its variations, like GPCR-MA [7] and [8], exploit the use of additional information, like electronic maps and traffic, in order to improve GPSR's performance. However, in [9] the authors investigate the advantages and disadvantages of multiple node-disjoint routes in VANETs. 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 (6 hops or more) better performance may be achieved by using multiple node-disjoint paths, especially in the case of high data rates. In [10], the authors propose a variation of the AOMDV [11] protocol that exploits vehicles' speed in order to optimize the best route decision. Finally, in [12] 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 Junction-based Multipath Source Routing, or JMSR for short. JMSR 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. JMSR 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. Our performance evaluation documents most of the conclusions of [9] described earlier.

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. The performance evaluation is demonstrated and commented on in Section IV. The ending section summarises our findings.

II. JMSR'S CHARACTERISTICS

Similar to 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 neighbors. Finally, we assume that the vehicles' density is relatively high, as in urban environments during rush hours.

JMSR 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

Manuscript received September 10, 2012. The associate editor coordinating the review of this letter and approving it for publication was C. Sacchi.

P. Sermpezis is with EURECOM, France (e-mail: sermpezi@eurecom.fr).

G. Koltsidas and F.-N. Pavlidou are with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece (e-mail: {fractgkb, niopi}@auth.gr).

Digital Object Identifier 10.1109/LCOMM.2013.011513.122045

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. JMSR is a *multipath* protocol, however, multipath refers to paths comprising series of junctions and not series of nodes, as usual. Hence, JMSR'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; JMSR exploits this fact.

Finally, JMSR 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 JMSR

In this section we provide 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 JMSR, 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 nodes 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 JMSR 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 center 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 [15].

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. JMSR 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

¹It has been shown that in MANETs more than two routes are only slightly beneficial [13]. However, depending on the application, one could find more than two routes [14].

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_{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. PERFORMANCE EVALUATION

We used NS-2 [16] to evaluate the performance of JMSR. We modelled the city environment as a rectangular area with horizontal and vertical streets. The dimensions of the modelled area is 1500m×500m and the node density is 800 nodes/km². The distance between two neighbouring parallel streets is 100m (corresponding to a medium size city block, derived from real-world cities). Spaces between the streets are assumed to be buildings that don't allow the propagation of radio waves through them. We followed the IDM_LC (Intelligent Driver Model with Lane Changing) mobility model [17], included in the VanetMobiSim framework. It uses smart intersection management, lane changes management and speed regulation depending on traffic conditions. The node's communication range is 350m, and IEEE802.11p protocol is used with 6Mbps rate at 5.9GHz. Hence a transmission of a node can be received by nodes located 3 junctions away, on average. Each simulation lasts for 500 seconds, all traffic sessions start at random times near the beginning of the simulation run and CBR/UDP traffic flows of 512 bytes data packets are used.

In this letter we present the results for two performance metrics: The *Packet Delivery Ratio (PDR)*, defined as the ratio of the data packets delivered to the destinations to the total data packets generated by the CBR sources, and *Average end-to-end delay*, which is the average time that the received packets

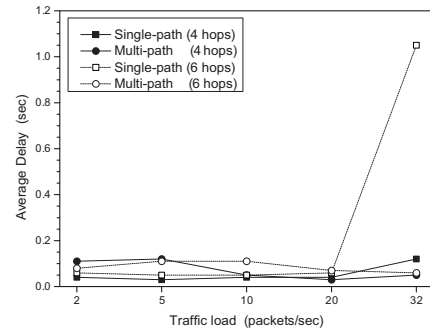


Fig. 1. Simulation set 1: average delay versus traffic load.

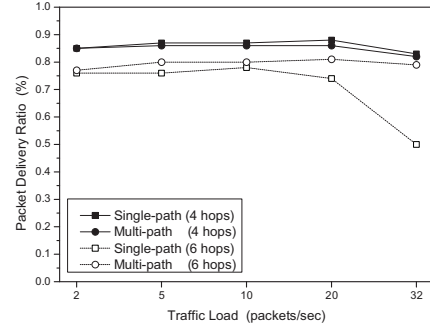


Fig. 2. Simulation set 1: packet delivery ratio versus traffic load.

needed to reach the destinations. We ran two simulation sets, varying the traffic load generated by the sources. For single-path routing, we implemented a version of our multipath protocol, in which all the packets are forwarded only through the shortest available path. In the second simulation set, we have also included AOMDV in the simulations for comparison. In the following we depict the preliminary simulation results, where JMSR refers to the multipath version.

In the first simulation set we study the performance of JMSR for only one source-destination pair, so we can better focus on the effect of the increasing traffic load. Two source-destination distances were simulated, 4 and 6 hops. In both cases, traffic load is varied from 2 to 32 packets per second.

The average end-to-end delay is depicted in Fig. 1. For 4 hops distance, single-path performs better than multipath except for the very high traffic load. The behaviour is the same for the 6 hops distance too. Multipath seems to be beneficial only for very high traffic load. Regarding PDR, presented in Fig. 2, in the case of 4 hops distance, no particular difference is observed. However, when source-destination distance is 6 hops, multipath performs clearly better than single-path, and the difference increases as traffic load increases. For the highest traffic load, multipath delivers almost 50% more packets than single-path, while achieving lower delay. Hence, this basic simulation reveals that multipath becomes beneficial only in high traffic load cases, where traffic is split among the paths, avoiding congestion on one single route which would delay packet delivery and risk packet loss.

In the second simulation set we investigate the protocol's performance under more realistic situations. We have set 4 source-destination pairs to exchange traffic at a rate of 2, 5 and 10 packets/second. We also included AOMDV in the

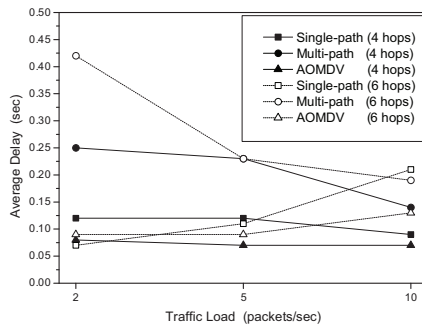


Fig. 3. Simulation set 2: average delay versus traffic load.

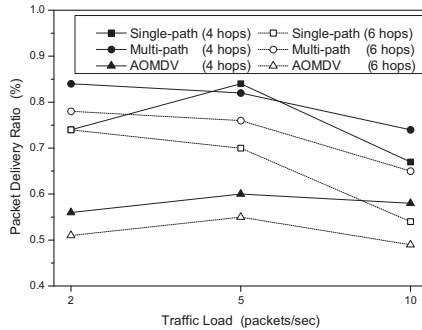


Fig. 4. Simulation set 2: packet delivery ratio versus traffic load.

simulations for comparison with another multipath protocol, although not designed for VANETs. Two cases were assumed where the source-destination pair distance is 4 and 6 hops.

Fig. 3 depicts the results for the average end-to-end delay. When source and destination are 4 hops apart, AOMDV has the lowest delay and single-path has lower delay than multipath. When their distance increases to 6 hops, then for low traffic load, single-path achieves again lower delay values. But, for high traffic load, multipath achieves lower delay than single-path. AOMDV has again the lowest delay.

Apart from delay, PDR is also of crucial importance, and the results are presented in Fig. 4. For small source-destination distances, multipath achieves almost always the highest values, followed by single-path and AOMDV in decreasing order. For longer source-destination distances, multipath achieves again the highest values. Hence, in any case, multipath achieves the highest PDR values among the three simulated protocols. By taking also into account the previous graph too, we may conclude that multipath is beneficial for high traffic loads and when sources and destinations are far apart.

The results are explained by the fact that multiple paths are congested later in time than single paths, however, since usually the second path is longer than the shortest one, additional delay is introduced. Therefore, it is not until the single route is highly congested that the additional delay in the multipath case becomes lower than the delay of the single-path case. In addition, since congestion means higher packet losses, multipath has much better performance in this metric.

V. CONCLUSIONS

In this work, we introduced a novel VANET geographic multipath routing protocol, where the alternative paths are as disjoint as possible. The general conclusion from our preliminary performance evaluation is that multipath is beneficial for VANETs, in case the source-destination distances are medium or long (6 hops away or more), or traffic loads are medium to high, conditions that real-world VANETs will probably face. As a future work, we intend to add a recovery mechanism to cover the cases where no appropriate forwarding nodes have been found. Moreover, we aim to investigate the potential of a hybrid algorithm that uses single path for short distances and lightweight traffic and multipath otherwise.

REFERENCES

- [1] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Veh. Technol. Mag.*, vol. 2, pp. 12–22, 2007.
- [2] I. Broustis and M. Faloutsos, "Routing in vehicular networks: feasibility, modelling and security," *International J. Veh. Technol.*, vol. 8, Mar. 2008.
- [3] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Commun. Rev.*, 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. 2003 IEEE Intelligent Vehicles Symposium*, pp. 156–161.
- [5] V. Naumov and T. Gross, "Connectivity-aware routing (CAR) in vehicular ad-hoc networks," in *Proc. 2007 IEEE International Conference on Computer Communications*, pp. 1919–1927.
- [6] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 2000 ACM/IEEE International Conference on Mobile Computing and Networking*.
- [7] F. Granelli, G. Boato, D. Kliazovich, and G. Vernazza, "Enhanced QPSR routing in multi-hop vehicular communications through movement awareness," *IEEE Commun. Lett.*, vol. 11, pp. 781–783, 2007.
- [8] S. Sun, J. Hu, X. Luo, and Q. Wang, "Improved QPSR in inter-vehicle communication," in *Proc. 2010 International Conference on Communications and Mobile Computing*, pp. 259–265.
- [9] X. Huang and Y. Fang, "Performance study of node-disjoint multipath routing in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1942–1950, May 2009.
- [10] W.-I. Lee, S. I. Chowdhury, G.-Y. Kee, W.-S. Baek, and J.-Y. Pyun, "Velocity aware multipath distance vector routing protocol for high mobility over vehicular ad-hoc networks," in *Proc. 2011 IEEE/FTRA International Conference on Multimedia and Ubiquitous Engineering*.
- [11] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proc. 2001 International Conference on Network Protocols*, pp. 14–23.
- [12] C.-S. Wu, S.-C. Hu, and C.-S. Hsu, "Design of fast restoration multipath routing in vanets," in *Proc. 2010 International Computer Symposium*.
- [13] A. Nasipuri, R. Castaneda, and S. R. Das, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 6, no. 4, pp. 339–349, 2001.
- [14] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125–145, 1974.
- [15] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," in *Proc. 2006 IEEE International Conference on Computer Communications*.
- [16] "Network Simulator 2." Available: <http://www.isi.edu/nsnam/ns>.
- [17] J. Harri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in *Proc. 2006 ACM International Workshop on Vehicular Ad Hoc Networks*.