R A P P O R T D E R E C H E R C H E

L R I

LINK TRANSPORT TIME AND INFRASTRUCTURE AIDED DELAY EFFICIENT ROUTING IN VANETs

DIEZ N / BOUKHATEM L / STOJMENOVIC I

Unité Mixte de Recherche 8623
CNRS-Université Paris Sud – LRI

03/2009

Rapport de Recherche N° 1513

CNRS – Université de Paris Sud
Centre d'Orsay
LABORATOIRE DE RECHERCHE EN INFORMATIQUE
Bâtiment 490
91405 ORSAY Cedex (France)

Link Transport Time and Infrastructure Aided Delay Efficient Routing in VANETs

Nicolas DIEZ
Lila BOUKHATEM
LRI, Univ. Paris-Sud, CNRS
Bât 490, 91405 Orsay, France
Email: $\{nicolas.diez|lila.boukhatem\}@lri.fr$

Ivan STOJMENOVIC
SITE, University of Ottawa, Canada
EECE, University of Birmingham, UK
Email: $stojmenovic@storm.ca$

***Abstract*—In this paper, we address the problem of routing in Vehicular Ad hoc NETworks (VANETs). We present a new metric, Link Transport Time (LTT) to estimate connectivity and channel load on a road segment (link) between two RSUs (road side units) located at neighboring intersections. Regular beacon and other messages include the time of last encounter with an RSU. This time is propagated and updated from vehicle to vehicle until a vehicle arrives at the next RSU, which is then able to calculate the time needed for a message to travel between two RSUs by either data muling or V2V communications. This metric is applied in LoP (LTT-over-Progress) a new infrastructure aided routing algorithm. In LoP routing from a source to a destination D with known location, all links leading closer to D and having acceptable LTT metric are identified. The selected link is the one with the minimal ratio of LTT over the progress in terms of distance to D. We show that LoP provides efficient use of the channel, and low transmission delays by the use of the connectivity and channel load estimation, compared to GSR algorithm.**

I. INTRODUCTION

Recently, there has been increasing interest in exploring new advances of wireless communication technology in transportation systems. A joint effort of the different stakeholders such as automobile manufacturers and suppliers, government and research organizations has been dedicated to further increase the road traffic safety and efficiency by the mean of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications [1], [2]. Therefore, more and more vehicles are being equipped with onboard electronic and communication devices such as GPS (Global Positioning System), digital maps and various communication interfaces. Other communication facilities are envisaged to be deployed along the roads enabling vehicles to exchange information with the road infrastructure for communication and collaboration purposes. Communications can then be supported by nearby vehicles using V2V links or by direct V2I communications. One example of service is to broadcast alert messages to vehicles in the vicinity of a dangerous location. Another example is to inform the drivers about real-time traffic conditions, or allow them to share entertainment files making their trip more pleasant.

This work was supported by grant from Région Ile-de-France, UK RSWRMA and Canada Strategic grants

VANETs are a form of Mobile Ad hoc NETworks (MANETs) [3] and represent a real candidate for sustainable deployment of ad hoc technology. However, VANETs have their own particularities which differentiate them from traditional Ad hoc communications. In VANETs, each vehicle moves on the road with a high but a limited speed and the roadway geometry constrains the vehicle movement leading to a quite predictable mobility pattern. Furthermore, the high vehicles velocity and density combined to the particular propagation city environment (reflection due to buildings and obstacles) induce a highly error-prone and noisy channel. As a consequence, many link disconnections may occur leading to frequent network topology changes.

Inversely, in other communication scenarios, the network can suffer from partitioning issues due to the lack of possible forwarding possibilities (node or cluster isolation). Finally, energy constraints are not an issue in VANETs since vehicles can provide continuous energy to their communication devices. All these VANETs specificities make it mandatory to develop adapted and robust routing protocols. Existing routing protocols for MANETs (AODV, OLSR and DSR [4], [5], etc.) are inadequate to meet this purpose. Most of them aim to find the minimum hop path from a source to a destination which may not be suitable for the highly dynamic VANETs environment.

A promising approach has been introduced by the geographic (or location) based routing concept which relies on the idea that the source sends a message to the geographic location of the destination instead of using the network address. The forwarding decision is then based on both the position of the destination and the position of the node's immediate neighbors. This approach can be envisaged for VANET networks due to the vehicle embedded GPS facility. Several routing protocols using this concept have been proposed [6] and some of them have been adapted to the vehicular context [7] [8] [9] [10]. Major proposals are based on a source routing approach, only few use an adaptive routing we define as "intersection-based routing" where routing decision is taken at each intersection. Intersection-based routing approach is adapted to urban area vehicular networks where, at each intersection, local routing decisions can be optimized by taking into account the instantaneous vehicle connectivity and the link channel load toward each neighboring intersection.

In this paper, we propose a novel intersection-based geographical routing called LoP (LTT-over-Progress). In LoP, vehicle-to-vehicle communications are used to relay data packets from source to destination nodes. When reaching an intersection, a routing decision is made to select the best link (road segment between two intersections) toward a neighboring intersection through which the data packet should be relayed. The best link should be the one which i) ensures progress toward the destination node, ii) guarantees vehicle connectivity, and iii) shows the minimum channel utilization load. Such an approach will reduce the end-to-end packet delay by choosing the most reliable and less-heavy loaded channel routes toward the destination. This is achieved by defining a new metric called LTT (Link Transport Time) which reflects the channel utilization rate and the observed latency between two intersections. To estimate the LTT value, LoP relies on a simple road infrastructure which consists in RSUs (Road Side Units) placed at each intersection.

The rest of this paper is organized as follows. Section II introduces previous works on routing in VANETs and the motivations for new routing schemes. Section III presents our algorithm for delay efficient routing. In section IV, we argue on the efficiency of the new algorithm. Finally, section V concludes this paper.

II. Related work

Geographic-based routing protocols have been proposed to support efficient and scalable routing for Ad hoc and sensor networks. Karp and Kung [14] proposed the GPSR (Greedy Perimeter Stateless Routing) that uses greedy perimeter forwarding method to select the next relaying node. The geographic distance routing (GEDIR) [11] uses a greedy approach to achieve efficient and loop-free data delivery (in a collision free-environment). These protocols perform very well in static ad hoc configurations but they where not initially planned for topology constrained and high mobility environments.

Recently, routing in VANET has been extensively studied. In GSR [7] a geographic source routing approach is proposed. GSR extends the position-based routing in a vehicular environment by considering topology information. Once the destination location is discovered (through the RLS flooding system), the source uses a digital map of the roads to determine using the Dijkistra algorithm the set of intersections a packet will follow. A basic greedy forwarding is used to relay packets between the vehicles. The route chosen by GSR is based on the shortest distance in terms of intersection hops between source and destination. It does not reflect the density of vehicles on each road segment which can lead in some cases to routing holes.

CLA-S [12] is a connection-less routing protocol proposed for city environments which can adapt to network topology changes mainly caused by fast moving vehicles and the presence of large obstacles (buildings). This approach does not require a node to maintain its neighbor's locations or hop-by-hop routes. In CLA-S, the road topology is divided in cells in such a way that each vehicle in a cell can communicate with all the vehicles of the adjacent cells. To transmit a packet, a source node defines a *forwarding zone* around a geometric reference line between the source and the destination. This forwarding zone is the area where nodes are able to forward the message to the destination. In case of low node density environments, the destination node may not receive any data from the source node for a period of time. In this case, the destination sends a request to the source to increase the size of the forwarding zone. CLA-S shows good performance but is seems too restrictive in the cells construction, route selection (only routes around the reference line) and dependant on the forwarding zone adjustment phase.

The authors in [13] propose A-STAR, an Anchor-based Street and Traffic Aware Routing algorithm for urban environments. A-STAR uses the existence of bus lines to select high connectivity paths. Since busses pass regularly on some road segments, the vehicle connectivity is more stable there than in other city roads. The authors give a weight to each segment of road based on the number of bus lines on it. The weight value is inversely proportional to the number of bus lines. Then, a source routing decision can be taken using Dijkstra's algorithm to find the route with the least weight. The author propose to dynamically adjust roads weights using a periodic monitoring of the road traffic. In case of route failure, a new recovery process is executed to find a new anchor route. A-STAR shows interesting performance, but it did not really define a method to estimate the road traffic condition for the weight adjustment. Furthermore, the vehicle density does not reflect the real vehicle connectivity or the real channel load on the considered route.

A link quality estimation approach was developed in MURU [9] and GVGRID [10] which are two source routing protocols. They define a new metric for the quality of a network path and more specially the probability of disconnection. MURU aims to minimize the probability of path breakage by exploiting mobility information of vehicles. A new metric called expected disconnection degree (EDD) is defined to select the most robust path from source to destination. In GVGRID, the stable routes are selected by choosing a route by vehicles which are likely to move at similar speed and toward similar directions using information on road topology and vehicles positions. GVGRID uses a simple line of sigh channel propagation model without fading assumptions.

In summary, most the above mentioned schemes are source routing protocols since route selection to link source to destination is achieved once before data transmission. The best route is found using several criteria such as: distance, link stability, vehicles direction, etc. The limitation of such an approach is that even thought in some cases the route can be optimal when it is built, this may change during data transmission phase due to high vehicles mobility and routes instability. To address this problem, we propose in this paper an intersection-based routing approach which dynamically adapts routing by taking decision at each intersection based on more accurate and up-to-date local information of route segments status.

Besides, several works use the vehicle density as a parameter for route selection. However, this density factor does not reflect the exact distribution of vehicles. A road segment with a high density may show a lack of connectivity and thus a routing hole if vehicles are not uniformly distributed but concentred on a part of the road. In addition, the vehicle density does not neither reflect the channel utilization rate on the considered road segment. A road with a high vehicle density may show a very slight data traffic channel load and vice versa.

Consequently, we present in this work LoP, a novel intersection-based routing which handles vehicle connectivity and real time channel traffic conditions to ensure the best possible delivery delay for data packets. Our algorithm adaptively selects the route path by choosing at each intersection the next road segment which ensure both connectivity and the minimum latency between pairs of intersections. This approach has also the advantage to achieve an efficient data traffic balancing over the network.

III. LoP PROTOCOL

A. Basic assumptions

LoP is a routing protocol suitable for vehicular urban environments in which a simple communication infrastructure is installed. The infrastructure consists of Road Side Units (RSUs) placed at each intersection. The RSUs can communicate with the vehicles within their coverage range and have knowledge of their local road topology (each RSU knows its neighboring RSUs). The RSUs are not directly connected by a communication link, but can use vehicle-to-vehicle communications to exchange information. The role of the infrastructure is to support the vehicle network in making routing decisions using LTT metric.

The urban area is represented by a undirected graph $G(V, E)$ where V is the set of intersections (junctions) and E the set of links (road segments). No restriction is made on the number of lines on each road segment. RSUs and all vehicles are assumed to be equipped with GPS devices, necessary for their geographical localization. We also assume that vehicles are installed with a digital map which describes the road topology.

Their periodic "hello" messages to neighbors will be augmented with time of last encounter of RSU at previous intersection.

B. LTT estimation

The key feature of LoP is the estimation of LTT, the time spent by a packet to be transmitted from a source RSU (RSU_i) to one of its neighboring RSUs (RSU_j) using vehicle-to-vehicle communications. This expected delay on each road segment is estimated by recent historical data on vehicles that just traversed the road segment. These vehicles will carry the time of last encounter of RSU at previous intersection.

LTT time reflects the current channel traffic load and the vehicle connectivity on the road segment between RSU_i and RSU_j. The aim of this approach is to easily provide an estimation of the real-time latency between two adjacent intersections. This latency will then reflect several parameters: vehicles density, vehicles distribution, and channel traffic load. Vehicles density and distribution represent the network connectivity and the channel traffic load is related to the data delivery ratio on the concerned road segment. Using this latency metric, the routing protocol can choose at each intersection the road segment which ensures the minimum latency toward the next intersection.

To estimate the LTT value of each road segment, every RSU of the network needs to broadcast periodically an *LTT-Scan* information to vehicles in its transmission range.

LTT-Scan packet contains two main information: the identifier of the RSU source RSU_i, and the current transmission time T_t. T_t time will then reflect the time of last encounter with RSU_i. If a vehicle A passing by RSU_i receives the *LTT-Scan* message, it can either carry the T_t information or transmit it greedily to the vehicles ahead that are progressively closer to the RSU destination. Note that this T_t information can be either sent in a specific *LTT* message or carried by periodic "hello" packets. In the following, we consider the solution which uses dedicated *LTT* packets. The generated *LTT* packet contains: the identifier of the RSU source RSU_i, the T_t time, and the identifier of the RSU destination RSU_j.

The RSU destination can be one of the neighboring RSUs of RSU_i. Each vehicle receiving the *LTT-Scan* packet discovers the destination RSU_j toward which it should send the *LTT* packet using the RSU source identifier (recorded in the *LTT-Scan* packet), its own position, and the digital map information. Thus depending on the road segment to which they belong, the forwarding vehicles (which received the *LTT-Scan* packet) independently select their respective destination RSU_j.

Figure 1 shows a scenario where vehicles A, B and C receive the *LTT-Scan* message of RSU_i. To prevent from multiple retransmissions of the same

T_t information, when a vehicle receives a LTT message with a T_t information, it first checks that this value is more recent than its previous buffered one, in which case a new LTT packet is generated and the new T_t is stored. Otherwise, the message is ignored.

In the scenario of figure 1, vehicle A upon receiving the ***LTT-Scan*** message, generates a LTT packet and sends it greedily to the destination RSU_j. At each hop, the selected forwarding node is the neighbor which is the closest to RSU_j. If such a vehicle does not exist, the carry-and-forward strategy is adopted to handle network partitioning. Vehicle A will then buffer the ***LTT*** packet and wait until there exists a valid next hop vehicle toward the destination (RSU_j). Upon receiving the ***LTT*** packet, the forwarding vehicle will update its previous recorded T_t value if the received value is fresher (more recent). This greedy forwarding procedure is repeated from vehicle to vehicle until the ***LTT*** packet reaches the destination RSU_j.

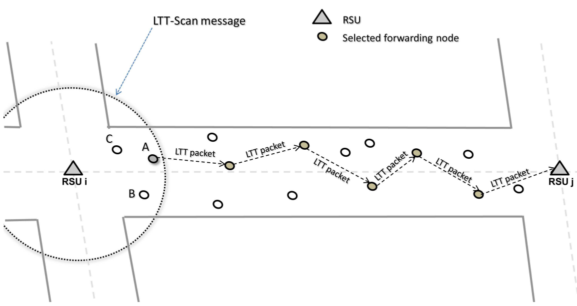

Figure 1. LTT-Scan and LTT packets propagation

Let T_r be the reception time of the ***LTT*** packet at the destination RSU_j. RSU_j can then simply derive the Link Transport Time $LTT_{i,j}$ between RSU_i and RSU_j as follows:

$$LTT_{i,j} = T_r - T_t \qquad (1)$$

At this step, note that this simple LTT estimation is possible only if RSU_i and RSU_j are time synchronized. This synchronization issue can be simply resolved using to the GPS facility which provides the same global time reference.

Another encountered issue is asymmetry. Once RSU_j gets the $LTT_{i,j}$ value, it has the V2V latency between RSU_i and RSU_j which is needed at RSU_i side for routing decisions. This information can be communicated in several ways: by using wired/wireless direct connection between RSUs, or by piggybacking $LTT_{i,j}$ value in data and/or hello packets transmitted in opposite direction (from RSU_j to RSU_i), or by considering symmetric time delay computation. In this work, we opted for the piggybacking solution.

The LTT estimation process being executed periodically, each RSU holds an estimation of current link transmission time toward every neighboring RSU. To help vehicles in their routing decision (see section III-C2), each RSU periodically broadcasts an ***LTT-Info*** packet which informs all vehicles in its vicinity about the current estimated LTT values of the different next road segments. Using these LTT values, vehicles can by themselves select the best road segment toward their respective destination nodes. Let n be the number of road segments connecting a given RSU_i to its RSU_1, RSU_2, ... RSU_n neighbors. The ***LTT-Info*** packet will then consists in pairs of values: $\{\{LTT_{i,1}, RSU_1\}, \{LTT_{i,2}, RSU_2\},$ $\{LTT_{i,3}, RSU_3\}, ..., \{LTT_{i,n}, RSU_n\}\}$.

C. Routing protocol

The LoP protocol includes two essential phases: 1) dynamically selecting a route by choosing at each intersection the best road segment in terms of estimated latency to the next RSU, 2) efficiently forwarding packets using V2V communications through each road segment of the selected route.

Suppose a source vehicle S wants to send data packets to a destination vehicle D. With pre-existing location registration and a lookup services [16], [17], we assume that vehicle S is able to get the geographic location of vehicle D. Vehicle S can then select one of the two RSUs at each side of the road segment to which it belongs. The selected destination RSU is the one that allows the highest progression to the destination vehicle D. S can then send its packets toward the destination RSU using the greedy forwarding approach we describe here after.

1) Forwarding between two intersections: Forwarding through road segments is achieved using a simple greedy forwarding approach as it was used to propagate the LTT message. Recall that in this approach each node makes a decision to which neighbor to forward the message based only on the location of itself, its neighboring nodes, and the intended destination. In our case, the intended destination is the selected destination RSU. Since every vehicle periodically beacons its location to all its neighbors, the vehicle holding the data packet can select the forwarding vehicle as the one of its neighbors which allows the best progress toward the destination RSU. Forwarding in this scheme follows successively closer geographic vehicles until the destination RSU is reached. Here again, since network partitioning can occur (no possible forwarding vehicle), a carry-and-forward approach is used.

Figure 2 shows an example of the greedy approach used to forward packets on the road segment until the destination RSU is reached.

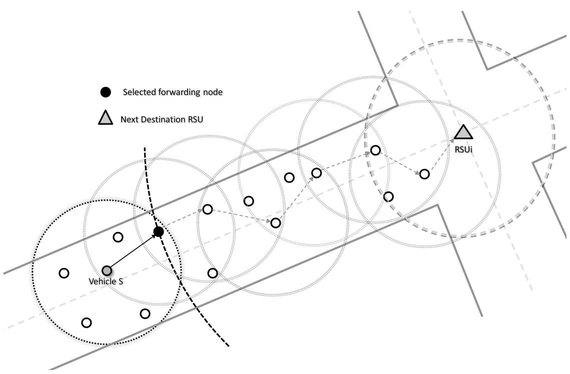

Figure 2. The gready forwarding between two RSUs

2) Routing decision at intersections: Suppose a vehicle A is the current forwarder of a data packet which final destination is vehicle D. When A reaches an intersection it receives the current RSU_i beacons containing the *LTT-Info* packet. This means that it should execute the intersection routing process to select the best road segment (and thus the next RSU) toward D. We assume that RSU_i is connected to n neighboring RSUs through n road segments.

The routing decision is taken by vehicle A using a Cost-over-Progress (CoP) function based on LTT metric and the achieved progress toward the final destination D. The CoP function of a road segment between RSU_i and a neighboring RSU_j is defined as follows:

$$CoP_{i,j} = \frac{LTT_{i,j}}{d - d_i} \quad \begin{array}{l} LTT_{i,j} < LTTthreshold \\ d - d_i > 0 \end{array} \tag{2}$$

where $LTT_{i,j}$ the link transport time between RSU_i and RSU_j, d (respectively d_i) the distance separating RSU_i (respectively the neighboring RSU_j) from the destination node D, and $LTTthreshold$ represents the maximum sustainable value before declaring a lack of connectivity between the two RSUs. Equation (2) estimates the cost of all road segments leading closer to D and having acceptable LTT metric.

Finally, the selected next destination RSU is the one which minimizes the cost function. Formally, the CoP_i function is defined as:

$$CoP_i = \min_{j=1,n} CoP_{i,j} \tag{3}$$

CoP_i function gives to vehicle A, at intersection i, the identity of RSU_j that should be used as next RSU destination. Then, vehicle A executes the greedy forwarding previously detailed in section III-C1 using RSU_j as next RSU destination. The two routing phases are repeated until final destination D is reached.

Figure 3 depicts an example scenario in which RSU_a helps the forwarding vehicle F to choose the next forwarding vehicle by selecting the best next route segment. In this example, RSU_a broadcasts in its periodic *LTT-Info* packet the current LTT values toward all its neighboring $RSUs$. The direct shortest path between source and destination (S and D) should pass through RSU_a and RSU_c. However, as we can see in the example, the road segment ($RSU_a - RSU_c$) shows a routing hole and thus a high LTT value. Vehicle F will then choose RSU_b as next destination RSU , since it shows the minimum LTT delay while progressing to destination D. The road segments ($RSU_a - RSU_d$)

and ($RSU_a - RSU_e$) show respectively a high LTT latency and non acceptable stand back from destination D.

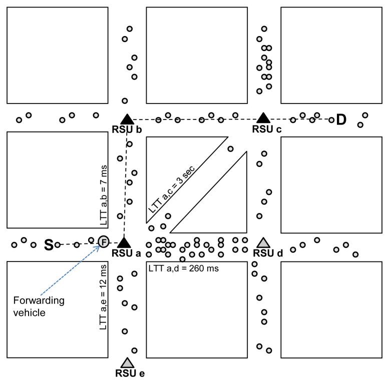

Figure 3. Routing decision in LoP

The LoP routing scheme is illustrated by the following algorithm.

Notations:
M: a data packet
D : final destination of packet M
RSU_i: current RSU
$Neighboring(RSU_i)$: set of n neighboring RSUs of RSU_i
$dist(x, y)$: gives distance between x and y

$d = dist(RSU_i, D)$
for $j \in Neighboring(RSU_i)$ **do**
  if $(LTT_{i,j} < LTTthreshold)$ AND $(d - dist(j, D) > 0)$ **then**
    $CoP_{i,j} = \frac{LTT_{i,j}}{d-dist(RSU_j,D)}$
  else
    $CoP_{i,j} = 0$
  end if
  if $Cop_{min} > CoP_{i,j}$ **then**
    $Cop_{min} = CoP_{i,j}$
    $NextRSU = j$
  end if
end for
if $Cop_{min} = 0$ **then**
  Failure
else
  Send packet M using greedy forwarding to $NextRSU$
end if

The CoP_i function may fail in deriving a result if either no progress is possible or LTT is greater than $LTTthreshold$ for all the links outgoing from RSU_i. In this case, a simple repair mechanism can be adopted by keeping the packets at the RSU_i side until a new refreshed LTT value allows the cost function to derive a solution. To avoid a long packets buffering at RSU, we define a Maximum Buffering Time (MBT) parameter which increases the $LTTthreshold$ value to accept higher LTTs.

IV. SIMULATION RESULTS

We implement LoP protocol using Opnet Modeler [18] and compare it with GSR [7] routing scheme. Recall that GSR uses a basic greedy forwarding scheme between vehicles on a route chosen based on the shortest distance in terms of intersection hops between source and destination.Both protocols implement the store-and-forward mechanism when no possible forwarding nodes exist.

The road layout consists of 12 intersections and 17 road segments totalizing 30 kilometers. To be aware of realistic vehicle movements, we used the VanetMobiSim [19] tool to generate the vehicles movement with a total of 105 generated vehicles and a mean vehicle density of 5 vehicles per kilometer. Vehicles move with a random speed between 10 and 50 kilometers per second. The simulation time is 2000 seconds and each experiment is repeated 15 times with different seeds of the random generator.

The simulated vehicle nodes and RSUs are equipped with an IEEE802.11 transceiver with a radio range of 250m. Several data flows are generated to send packets at a constant bit rate (a packet is sent each 100ms). The packet size is between 500 and 1500 bytes. Source and destination of packets are chosen randomly.

To evaluate the performance of LoP routing protocol and to compare it with the GSR scheme, we used two main metrics: end-to-end packet transmission delay from source to destination, and packets delivery ratio.

Figure 4 shows the end-to-end delay as a function of the Euclidean distance between source and destination. We can see that LoP shows a lower end-to-end delay than GSR. The main reason for that is that the adaptive path selection in LoP reduces the inaccurate route density selection and forward packets on the optimal local path. For GSR, since channel load and route connectivity are not considered, more network partitioning and packet buffering occur on the shortest path when forwarding the packets; leading to an increased delay.

The same behavior can be observed in figure 5 when considering the distance traveled by a packet which is the sum of all route segments lengths traveled by the packet. We can see that to travel the same distance between source and destination, LoP shows the lowest end-to-end delay thanks to its route selection criteria.

Figure 6 plots the packet delivery ratio as a function of the Euclidean communication distance (between source and destination). LoP outperforms GSR in terms of packet delivery since LoP when using the LTT metric selects at intersections the less traffic loaded routes while avoiding routing holes. The GSR protocol performs poorly because the geographic shortest path still suffer from high traffic loads and frequent network disconnections. Note that even though we added the store-and-forward mechanism to GSR, it shows a lower performance than LoP.

We note that both protocols achieve for high source-destination distances a poor delivery ratio. This is mainly due to a low mean vehicle density over the network leading to a high packet loss after a packet buffering expiration (fixed here to 20 seconds). Figure 7 shows the variation of the

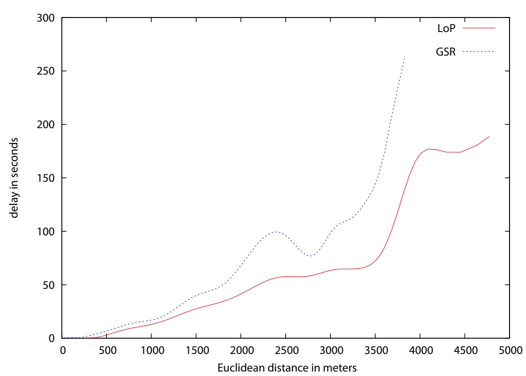

Figure 4. End-to-end delay as a function of the Euclidean distance

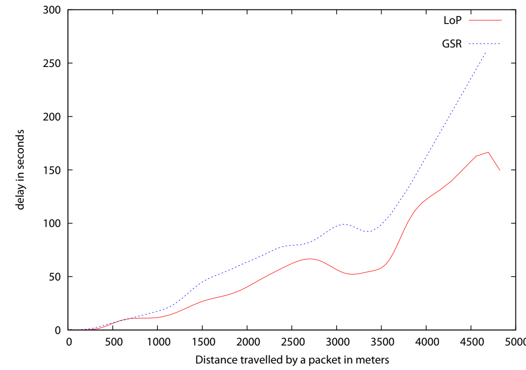

Figure 5. End-to-end delay as a function of the traveled distance

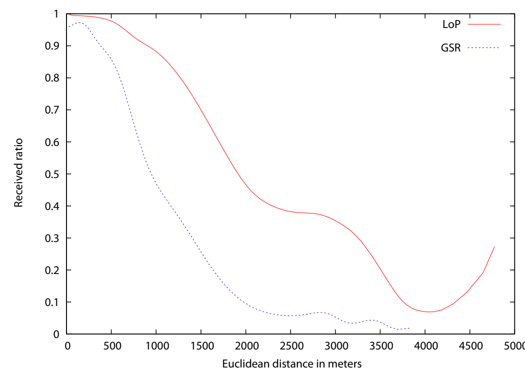

Figure 6. Data delivery ratio vs. the Euclidean distance

LTT value recorded on a given road segment as a function of the vehicular density on this segment. We can observe that LTT increases with the vehicle density. A higher density means at least a higher number of exchanged Hello messages; therefore the channel load increases and so does LTT. Finally, we analyzed the important LTT increase we can observe around a density of 16. The study shows that this behavior corresponds to the transition of a high data traffic through this road segment; therefore packets suffered from high contention and great number of retransmissions.

V. CONCLUSION

In this paper, we presented an adaptive routing algorithm called LoP (LTT-over-Progress) for urban

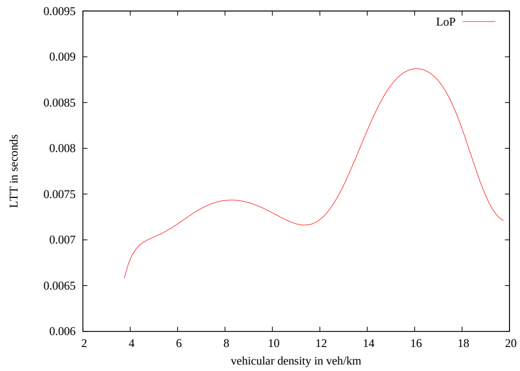

Figure 7. LTT variation on a road segment

area Vehicular networks equipped with a simple communication infrastructure. LoP targets at selecting road segments with the best transmission quality while avoiding network disconnections to provide the best delivery delays and reception rates. A new metric Link Transport Time (LTT) is introduced to estimate the real-time latency of a road segment and thus helps in choosing the best routes.

Our simulation results show that LoP achieves better performance than GSR in terms of higher data delivery ratio, and lower end-to-end delay. In our future work, we aim to extend the protocol using a DFS (Depth First Search) approach to further improve the LoP behavior against route failures.

REFERENCES

[1] "Car2car Consortium," http://www.car-to-car.org, last retrieved feb. 2009.

[2] "e-safety forum," http://ec.europa.eu/information_society/activities/esafety/forum/index_en.htm, last retrieved feb. 2009.

[3] "Manet working group," http://www.ietf.org/html.charters/manet-charter.html, last retrieved Sep. 2008.

[4] C. E. Perkins and E. M. Royer,"Ad hoc On-Demand Distance Vector Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, USA, February 1999, pp. 90–100.

[5] P. Jacquet, P. Muhletaler, A. Qayyum, A. Laouiti, T. Clausen, L. Viennot,"Optimized Link State Routing Protocol," in *IEEE INMIC*, Pakistan, Dec. 2001.

[6] I. Stojmenovic, "Position-Based Routing in Ad Hoc Networks," in *IEEE Communications Magazine*, Vol. 40, Jul. 2002, pp. 128–134.

[7] C. Lochert, M. Mauve, H. Fussler, and H. Hartenstein, "Geographic routing in city scenarios," in *ACM SIGMOBILE Mobile Computer Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[8] Qing Yang, A. Lim, Li Shuang, Jian Fang, and P. Agrawal, "ACAR: Adaptive Connectivity Aware Routing Protocol for Vehicular Ad Hoc Networks", in*Proceedings of the IEEE Internatinal Conference on Computer Communications and Networks*," Aug. 2008, pp. 1–6.

[9] Z. Mo, H. Zhu, K. Makki, and N. Pissinou, "Muru: A multi-hop routing protocol for urban vehicular ad hoc networks," *IEEE Mobiquitous*, pp. 1–8, 2006.

[10] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, "Gvgrid: A qos routing protocol for vehicular ad hoc networks," in *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*, June 2006, pp. 130–139.

[11] I. Stojmenovic, X. Lin, "GEDIR: Loop-free location based routing in wireless networks", in *IASTED Conference on Parallel and Distributed Computing and Systems*, Nov. 1997, pp. 1025–1028.

[12] A. Ho, Y. Ho, and K. Hua, "A connectionless approach to mobile ad hoc networks in street environments," in *Proceedings of the IEEE Intelligent Vehicles Symposium, 2005*, Las Vegas, USA, 2005, pp. 575–582.

[13] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-star: A mobile ad hoc routing strategy for metropolis vehicular communications," in *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, Athens, Greece, 2004, pp. 989–999.

[14] B. Karp and H. T. Kung, "Greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual International Conference on Mobile Computing and Networking (Mobicom2000)*, Boston, USA, 2000, pp. 243–254.

[15] T. Little and A. Argawal, "An information propagation scheme for vanets," in *Proceedings of the 8th International IEEE conference on Intelligent Transportation Systems*, Sep. 2005, pp. 155–160.

[16] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad-hoc Routing," in *ACM MobiCom00*, 2000.

[17] M. Grossglauser and M. Vetterli, "Locating nodes with ease: Last encounter routing in ad hoc networks through mobility diffusion," in *IEEE INFOCOM03*, Oct. 2003.

[18] "Opnet Modeler," http://www.opnet.com, last retrieved Feb. 2009.

[19] J. Harri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: Generating realistic mobility patterns for VANETs," in *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, ACM Press, 2006, pp. 96–97.