Performance Evaluation of VANETs in Different Real Map Scenarios

Ryuji Ono, Elis Kulla and Evjola Spaho

Abstract With the advancement of Vehicular technologies, Vehicular Adhoc Networks will soon be a reality in our daily lives. However in order to fully take advantage of the new applications, we need to consider different parameters, while implementing the network. Road patterns, types of data, vehicle density and so on are some parameters that can affect the performance. In this paper we conduct simulations for VANETs in the Japanese city of Soja, Okayama Prefecture and one of the busiest parts of Tokyo metropolitan area, Shibuya. In order to create the road map we used real data from Open Source Map (OSM). Then, we generated mobility traces from eWorld into SUMO-compatible format. The data traffic is generated and evaluated by NS3 and its related tools. We investigate the effect of road patterns, types of traffic data and vehicle density.

1 Introduction

Vehicular Ad-hoc Networks (VANETs) are a special type of Ad-hoc networks and are an important component of the Intelligent Transportation Systems (ITS). They can been utilized to guarantee road safety, to avoid potential accidents by creating

Ryuji Ono
Okayama University of Science, Department of Information ad Computer Engineering,
1-1 Ridai-cho, Kita-ku, 700-0005 Okayama, Japan
e-mail: t16jm02or@ous.jp

Elis Kulla
Okayama University of Science, Department of Information ad Computer Engineering,
1-1 Ridai-cho, Kita-ku, 700-0005 Okayama, Japan
e-mail: kulla@ice.ous.ac.jp

Evjola Spaho
Polytechnic University of Tirana, Department of Electronics and Communications,
Mother Teresa Square, No. 4, Tirana, Albania
e-mail: evjolaspaho@hotmail.com

L. Barolli et al. (eds.), *Advances on Broad-Band Wireless Computing, Communication and Applications*, Lecture Notes on Data Engineering and Communications Technologies 2, DOI 10.1007/978-3-319-49106-6_63

new forms of inter vehicle communications and applications. Due to the high cost of deploying and implementing VANET systems in a real environment, most of research is concentrated on simulations. In the recent years, a lot of simulators for VANETs have been developed [1]. For example, the IMPORTANT framework has been one of the first attempt to understand the dependence between vehicular traffic and communication performance [2], [3]. The authors analyzed the impact of the node mobility on the duration of communication paths. In [4], the authors present a simulator written in Java, which can generate mobility traces in several formats. There are also other powerful traffic simulators, like TranSim [5], which makes use of a cellular automaton for simulating the interaction of vehicles. Cellular Automaton based VEhicular NETwork (CAVENET) [7] is a lightweight simulator which can be used to understand the properties of the mobility models of vehicular traffic and their impact on the performance of VANETs. SUMO is another powerful traffic simulator, intended for traffic planning and road design optimization. There is an attempt to interface SUMO with NS2 [6]. Since VANETs are a specific class of ad-hoc networks, the commonly used ad-hoc routing protocols initially implemented for MANETs have been tested and evaluated for VANET environments. VANETs share some common characteristics with MANETs. They are both characterized by the movement and self organization of the nodes. We consider the possibility of using ad-hoc and MANET protocols for VANET scenarios. In other previous work [9], the evaluated the performance of MANET routing protocols, using as network simulator NS2 and CAVENET vehicular mobility model.

In this paper we conduct simulations for VANETs in the Japanese city of Soja, Okayama Prefecture and one of the busiest parts of Tokyo metropolitan area, Shibuya. In order to create the road map we used real data from Open Source Map (OSM)[10]. Then, we generated mobility traces from eWorld [11] into SUMO-compatible [12] format. The data traffic injected in the network is generated and evaluated by NS3 [13] and its related tools. We analyze and compare three routing protocols: Ad-hoc On-demand Distance Vector (AODV) [14], Optimized Link State Routing (OLSR) [15] and Destination-Sequenced Distance-Vector [**?**].

This paper is organized as follows. In Section 2, the three routing protocols are summarized. The simulation system design and implementation is presented in Section 3. In Section 4, we show the simulation results. Finally, the conclusions and future work are presented in Section 5.

2 Routing Protocols

2.1 Optimized Link-State Routing (OLSR) Protocol

The OLSR protocol [11] is a pro-active routing protocol, which builds up a route for data transmission by maintaining a routing table inside every node of the network. The routing table is computed upon the knowledge of topology information, which is exchanged by means of Topology Control (TC) packets. OLSR makes use of HELLO messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its Multi Point Relays (MPR) based on the one hop node which offer the best routes to the two hop nodes. By using this MPR-based flooding mechanism, the amount of control traffic can be reduced. Each node has also an MPR selector set which enumerates nodes that have selected it as an MPR node. OLSR uses TC messages along with MPR forwarding to disseminate neighbor information throughout the network. Host Network Address (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes.

2.2 Ad-hoc On-Demand Distance Vector (AODV) Routing Protocol

AODV is an on-demand routing protocol. It performs Route Discovery using control messages: Route Request (RREQ) and Route Reply (RREP). In AODV, routes are set up by flooding the network with RREQ packets. As a RREQ traverses the network, the traversed nodes store information about the source, the destination, and the node from which they received the RREQ. The later information is used to set up the reverse path back to the source. When the RREQ reaches a node, that knows a route to the destination or the destination itself, the node responds to the source with a RREP packet which is routed through the reverse path set up by the RREQ. This sets the forward route from the source to the destination. To avoid overburdening the nodes with information about routes which are no longer (if ever) used, nodes discard this information after a timeout. When either destination or intermediate node moves, a Route Error (RERR) is sent to the affected source nodes. When source node receives the RERR, it can reinitiate route discovery if the route is still needed. Neighborhood information is obtained by periodically broadcasting Hello packets [14]. For the maintenance of the routes, two methods can be used: a) ACK messages in MAC level or b) HELLO messages in network layer.

2.3 Dynamic Source Distance Vector

Destination Sequenced Distance Vector Routing (DSDV) [16] uses distance vectors to continuously maintain routes throughout a network. Unlike RIP, DSDV uses per-node sequence numbers to provide a total ordering on route information age in order to prevent loops. In DSDV, each node maintains a route to each other node.

3 Simulation System Design and Implementation

In order to generate our scenarios and conduct the simulations in this paper, we used different systems that can be interconnected with each other. We used eWorld, OSM, SUMO and NS3.

3.1 Simulation Environment

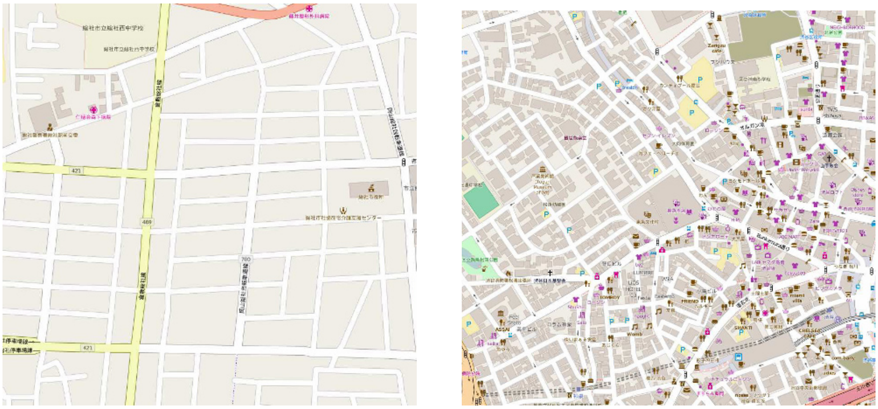
First, we use eWorld in order to get real from data from JOSM, a widely used open source map of the whole world. After getting the map data for the two parts of Japan, where we want to conduct simulations, we load it into eWorld and make certain adjustment to roads and streets. Then we add vehicle movement. eWorld allows users to setup starting and ending points of vehicle movements and the number of vehicles generated in a given time. You can also define different type of cars, with different maximum moving speed and physical length. All this data is exported to SUMO, where it is edited and compiled in order to get NS3 mobility data format.

After defining the mobility pattern, the network simulation model is based on NS-3. The NS3 simulator is developed and distributed completely in the C++ programming language, because it better facilitated the inclusion of C-based implementation code. The NS3 architecture is similar to Linux computers, with internal interface and application interfaces such as network interfaces, device drivers and sockets. The goals of NS3 are set very high: to create a new network simulator aligned with modern research needs and develop it in an open source community. Users of NS3 are free to write their simulation scripts as either C++ main() programs or Python programs. The NS3s low-level API is oriented towards the power-user but more accessible helper APIs are overlaid on top of the low-level API.

In order to achieve scalability of a very large number of simulated network elements, the NS3 simulation tools also support distributed simulation. The NS3 support standardized output formats for trace data, such as the pcap format used by network packet analyzing, tools such as tcpdump, and a standardized input format such as importing mobility trace files from different simulators.

The NS3 simulator has models for all network elements that comprise a computer network. For example, network devices represent the physical device that connects a node to the communication channel. This might be a simple Ethernet network interface card, or a more complex wireless IEEE 802.11 device. In our simulations we used IEEE 802.11p standard and TwoRayGroundPropagationLossModel.

IEEE 802.11p: Is an approved amendment to the IEEE 802.11 standard to add wireless access in vehicular environments (WAVE). It defines enhancements to 802.11 required to support Intelligent Transportation Systems (ITS) applications. The 802.11p standard is based on the 802.11 architecture, but version p is aimed at communications between vehicles and between them and fixed infrastructure. This new technology uses the 5.9 GHz band in various propagation environments: vehicle, open, urban, and so on. This standard defines the WAVE as the signaling



(a) Soja City Map (b) Shibuya Map

Fig. 1 JOSM Maps of Soja city and Shibuya.

technique and interface functions that are controlled by the physical layer (MAC) devices where the physical layer properties change rapidly and where the exchanges of information have a short duration. The purpose of this standard is to provide a set of specifications to ensure interoperability between wireless devices trying to communicate in rapidly changing environments and in particular time periods. TwoRayGroundPropagationLossModel: It considers the direct path and a ground reflection path. The received power at distance t is calculated with the following equation:

3.2 Simulation Settings

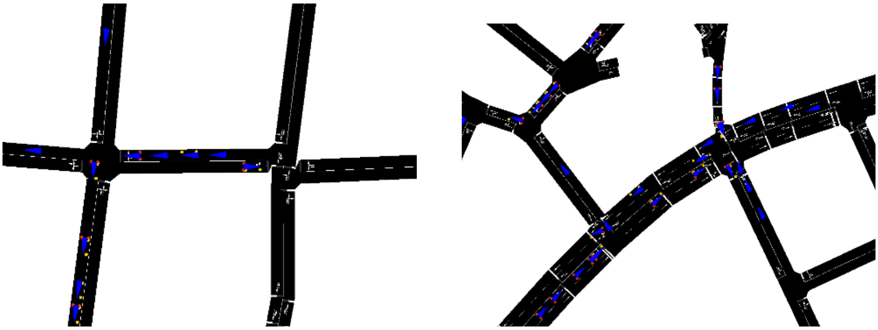
In order to analyze VANETs and the behavior of the routing protocols, we setup different scenarios for different maps (Soja and Shibuya), different number of vehicles in the simulation area and different routing protocols: OLSR, AODV and DSDV.

Different maps are used in order to create two environments where the density of the roads varies, considering Soja Town near our University and the traffic-packed region of Shibuya in Tokyo. The maps can be seen in Fig. 1. Data from the maps in then converted to e World and SUMO format, where we are able to run the vehicles in different patterns. Converted maps and a detailed road view of Soja town and Shibuya from SUMO Simulator can be seen in Fig. 2 and Fig. 3, respectively.

In order to analyze the behavior of VANETs, we setup three different densities in each map. In Fig. 2, we show three starting points and three ending points of car movements. For each pair of start-end points, vehicles with maximum moving speed of 60kmh, are generated in the area. They find their way to the end point in a random fashion, which is not the scope of this paper. Number of generated vehicles is: 50, 100 and 150. For a summary of parameter settings see Table 1.

(a) Soja City Map (b) Shibuya Map

Fig. 2 Maps of Soja city and Shibuya from SUMO Simulator.



(a) Soja City Map (b) Shibuya Map

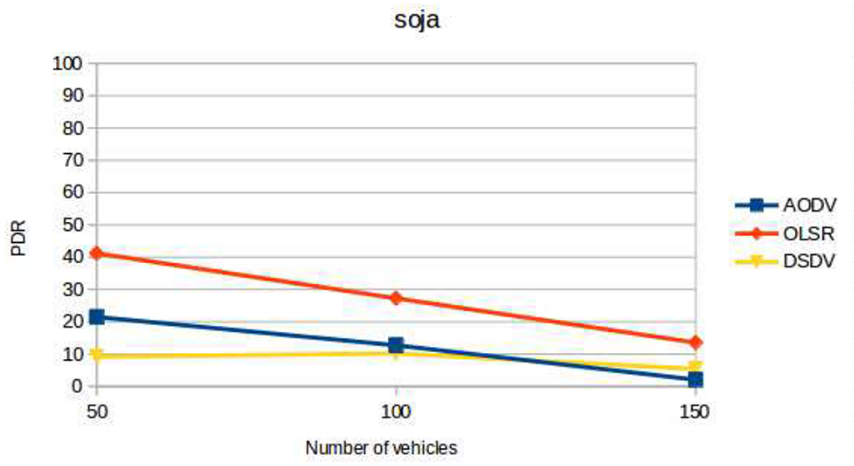
Fig. 3 Detailed road view of Soja town and Shibuya from SUMO Simulator.

Table 1 Simulation Parameters.

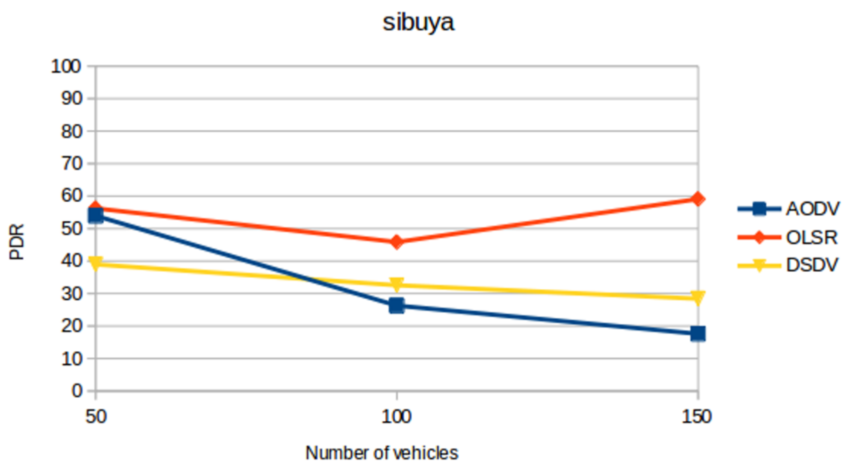
Parameter	Values
Area Size	$1000m \times 1000m$
Communication Distance	$250m$
Simulation Time	$300s$
Number of Vehicles	$50, 100, 150$
Maximum Moving Speed	$60km/h$
Routing Protocol	AODV, OLSR, DSDV

4 Simulation Results

In order to evaluate the performance of each scenario we used Packet Delivery Ratio (PDR) as a metric. In each of the start-end pairs of vehicle generation points, instead of choosing source and destination randomly among the moving vehicles, we put static nodes and sent three flows of CBR data transported over UDP. The reason we



(a) Soja



(b) Shibuya

Fig. 4 PDR Results for different number of vehicles and different routing protocols.

setup static nodes is because, when simulation starts, many random functions will decide the movement of vehicles. But, we wanted that at least the distance between source and destination that a packet should traverse in the network was fixed for all cases. If source and destination nodes were moving randomly, the performance would be affected considerably by the random functions of the simulator rather than the behavior of routing protocols. The results are shown in Fig. 4 as a graph and Table 2 and Table 3 as average values.

From results we can see that all protocols in all scenarios show a better performance when the vehicles move in Shibuya, compared to the case of Soja. An interesting finding concerns the performance decrease in both maps, when the number of vehicles increase. We expected an increase in performance, because by increasing the number of nodes there would be more routes available. But this was not the case in most of the scenarios, because the increase in number of vehicles created traffic and the vehicles were grouped in certain locations in the map. This was also affected by the difference is the road pattern difference between Soja and Shibuya. Shibuya has more roads and connections so the vehicles are more spread in the are and we see a PDR of $5-10\%$ in Soja and $30-40\%$ in Shibuya.

Table 2 Simulation Results (PDR Soja).

	AODV	OLSR	DSDV
50	21.48	41.19	9.16
100	12.71	27.26	10.20
150	2.05	13.53	5.39

Moreover, an increase in the number of routes and a high moving speed of vehicles, creates difficult and sometimes wrong route decisions especially from AODV, which is an on-demand routing protocol. In fact, OLSR shows an improvement for 150 vehicles in Shibuya, because OLSR MPR-based flooding mechanism makes the network respond faster to dynamic route changes.

Table 3 Simulation Results (PDR Shibuya).

	AODV	OLSR	DSDV
50	53.94	56.20	38.95
100	26.28	45.85	32.54
150	17.59	59.07	28.39

5 Conclusion and Future Works

In this paper we conducted simulations for VANETs in two Japanese cities with different road patterns. We analyze and compare three routing protocols: AODV, OLSR and DSDV. As evaluation metric, we used PDR and compared data for different scenarios. From the simulation results we found the following:

- When the number of vehicles increased, the PDR performance decreased in general, because of increased dynamism or routes.

- OLSR was able to handle dynamic routes better, because of the MPR-based flooding mechanism.
- In suburban areas like Soja, the lack of streets and connections creates traffic congestion, and brings performance decrease.

In our future works, we would like to investigate the behaviour of VANETs and VANET-specific routing protocols and VANET-specific applications. Moreover, we would like to test VANET in delay-tolerant applications.

References

1. J. Harri, F. Filali and C. Bonnet, Mobility Models for Vehicular Ad Hoc Networks: A Survey and Taxonomy, IEEE Communications Surveys & Tutorials, Vol. 11, No. 4, pp. 19-41, 2009.
2. F. Bai, N. Sadagopan, and A. Helmy, IMPORTANT: A Framework to Systematically Analyze the Impact of Mobility on Performance of Routing Protocols for Ad-hoc Networks, In Proc. of IEEE INFOCOM-2003, pp. 825-835, March-April 2003.
3. N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, PATHS: Analysis of Path Duration Statistics and Their Impact on Reactive MANET Routing Protocols, In Proc. of the 4-th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc-2003), pp. 245- 256, 2003.
4. M. Fiore, J. Harri, F. Filali, and C. Bonnet, Vehicular Mobility Simulation for VANETs, Proc. of the 40-th Annual Simulation Symposium (ANSS-2007), pp. 301-309, 2007.
5. L. Smith, R. Beckan, R. Anson, K. Nagel, and M. Williams, TRANSIMS: Transportation Analysis and Simulation System, In Proc. of the 5-th National Transportation Planning Methods Applications Conference, LA-UR 95-1664, April 1995.
6. M. Piorkowski, M. Raya, A. L. Lugo, M. Grossglauser, and J. P. Hubaux, Joint Traffic and Network Simulator for VANETs, In Proc. of Mobile and Information Communication Systems Conference (MICS-2006), October 2006, Available on line at: http://www.mics.ch/.
7. G. De Marco, M. Tadauchi and L. Barolli, Description and Analysis of a Toolbox for Vehicular Networks Simulation, In Proc. of IEEE ICPADS/PMAC-2WN-2007, Vol. 2, pp. 1-6, 2007.
8. K. Nagel and M. Schreckenberg, A Cellular Automaton Model for Freeway Traffic, Journal of Physics I France, Vol. 2, pp. 2221-2229, 1992.
9. E. Spaho, Gj. Mino, L. Barolli and F. Xhafa, Goodput and PDR Analysis of AODV, OLSR and DYMO Protocols for Vehicular Networks using CAVENET, International Journal of Grid and Utility Computing (IJGUC), Vol. 2, No. 2, pp. 130-138, 2011.
10. Open Street Map. Online at https://www.openstreetmap.org/.
11. eWorld framework. Online at http://eworld.sourceforge.net/.
12. SUMO User Documentation. Online at http://www.sumo.dlr.de/userdoc/.
13. Network Simulator version 3. Online at https://www.nsnam.org/.
14. C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On- Demand Distance Vector (AODV) Routing", IETF RFC 3561 (Experimental), 2003.
15. T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", IETF RFC 3626, October 2003.
16. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", In Proc. of ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, pp. 234-244, 1994.