Supporting electro-mobility in an urban environment with the Nomadic Agent concept

Christos S. Ioakimidis
ERA Chair 'Net-Zero Energy Efficiency on City Districts'
Research Institute for Energy, University of Mons
Mons, Belgium
christos.ioakeimidis@umons.ac.be

Philippe Moeyersoms
Industrial Engineering - Electronics & ICT
Karel de Grote University
Hoboken, Belgium
philippe.moeyersoms@gmail.com

Konstantinos N. Genikomsakis
DeustoTech Energy
Deusto Institute of Technology, University of Deusto
Bilbao, Spain
kostas.genikomsakis@deusto.es

Dimitrios Savvidis
DG Clima
European Commission
Brussels, Belgium
dsavvidis@hotmail.com

***Abstract*—The present work advances the use of the Nomadic Agent (NA) concept for supporting electro-mobility through the provision of location based information in an urban environment. In the envisaged system, each NA periodically broadcasts relevant data to other mobile nodes representing electric vehicles (EVs) inside its target area. At the same time, it is considered that multiple NAs are deployed to achieve full coverage throughout an urban area, establishing a chain among different NAs over a multi-hop communication network. Moreover, the NAs exchange information with EV charging posts that play the role of the aggregators in a smart grid, i.e. intermediate actors between the power system and the users. Hence, this communication platform can support value-added functionalities of EVs and their integration with the power system. As a first step towards the implementation of the envisaged system, this paper focuses on modeling the NA concept with the ns-3 discrete-event network simulator in order to assess its operation under realistic conditions. The results obtained from a small-scale simulation model, yet representative of the structural components of the urban web, indicate that the proposed approach has the potential to provide a suitable solution for the communications within the active area of the NA.**

Keywords—electro-mobility; nomadic agent; urban environment; vehicle-to-infrastructure

I. Introduction

Recent years have witnessed a growing interest in the so-called cooperative intelligent transportation systems (ITS), where vehicles communicate with each other (vehicle-to-vehicle, V2V) and with the infrastructure (vehicle-to-infrastructure, V2I), in order to improve the effectiveness [1], efficiency [2], and safety of transport networks [3]. In this context, ITS can benefit from the use of vehicular ad hoc networks (VANETs) in metropolitan areas by crowd-sourcing traffic information [1], while enabling real-time information exchange has the potential to improve existing alternative transportation services that aim to reduce the number of vehicles on the road and increase the average vehicle occupancy, such as vehicle sharing and pooling respectively.

Furthermore, the integration of electro-mobility, either as privately-owned electric vehicles (EVs) or in the form of sharing/pooling services based on EVs [4], with ITS enables smarter and more sustainable use of transport networks. On the one hand, the large scale deployment of EVs will add new loads to the electricity grid that may compromise its reliability during peak charging times. On the other hand, the future “smart grid” envisages that EVs communicate with electric utilities not only to determine the most beneficial time for charging based on grid capacity and energy prices, but also to provide supporting services to the grid through vehicle-to-grid (V2G) technology [5]-[7].

It becomes obvious that the seamless integration of EVs with the grid and ITS will add a new communication component to the road transportation networks, with requirements for real-time and two-way information flow between the connected entities. In this paradigm, the concept of the nomadic agent (NA) can be employed to provide location based services (LBS), given that information about traffic incidents (e.g. accidents, congestion and construction works) or availability of nearby EV charging stations is typically location sensitive. The author in [8] introduces the conceptual framework of using the NA approach in the context of mobility and V2V communication services in order to reduce the latency and bandwidth requirements of vehicle-to-roadside (V2R) communication. Specifically, an NA refers to an application/software process running on a mobile node (e.g. smart phone) that acts as an agent with the ability to keep localized information to a close proximity of its target area, while this information can be retrieved from the NA system upon user request. In contrast to the conventional definition of agents that move to gather or disseminate information in a different location, an NA switches host nodes in order to maintain its location inside its target area.

The present paper extends the role of NA to support electro-mobility in an urban environment and examines the data transfer in the envisaged system. Given that it depends on various factors, such as the number of nodes, their direction and speed, a simulation model is developed to assess the proposed approach under realistic conditions. To this end, this work employs a discrete-event network simulator to represent and closely monitor the mobile nodes and their communication within the target area of the NA, considering also the structural characteristics of the urban environment.

II. Background on the Operation of the NA

The role of NA is to gather and disseminate information in a specific area, thus it migrates between host nodes (e.g. vehicles) to maintain its position in the desired location. Once an NA moves out of its active area with its current host (Fig. 1), either it switches to another node to maintain the information inside the active area or it is terminated resulting in loss of information. For each NA i, the point $pi(x,y)$ defines its target location, while its active area is centered at $pi(x,y)$ with a radius of R^i_a. The holding area of NA i lies within its active area, is also centered at $pi(x,y)$, and its role is to help in identifying the time instance to initiate the search process for migrating the NA to another host. Given that it takes time to switch between hosts, once the NA detects that its current host has moved out of the holding area, it begins to search for candidate vehicles inside the holding area to migrate to.

The provision of V2V communication services using the NA approach is based on the following scenario: a vehicle A (e.g. an EV) wants to receive information (pull) while it is en-route, such as traffic incidents (e.g. accidents, congestion and construction works) or availability of nearby EV charging stations. Given that an NA holder broadcasts its information list through the V2V communication periodically, the vehicle A first attempts to identify the existence of an NA holder in the area and obtain access to its information list. If an NA holder is found, the vehicle A retrieves the information list to find if the required information is available by the NA holder. If the information is recent enough, the vehicle A tries to access and get the relevant information from the NA holder. If no NA holder is found, it is suggested that the vehicle A connects to the internet directly to retrieve the information, followed by the creation of a new NA (either by vehicle A itself or by selecting the vehicle closest to the center of the area) with updated information. If an NA holder is found, but the information is outdated or not available in the list of the NA holder, the vehicle A once again tries to retrieve it through the infrastructure based communication, and update the information kept on the NA via the V2V communication.

Extending the use of this system as a way for V2I communications, a number of NAs can be deployed at the same time to achieve full coverage of a given area of interest, establishing thus a chain among different NAs over a multi-hop communication network. In addition to the periodical data broadcast inside each target area using V2V communications, the NA also sends relevant information through the shortest available path to the aggregator of the smart grid. The latter acts as intermediary between the power system and users [5], realizing thus a V2I communication to support value-added functionalities of EVs and their integration with the power system. In this case, the EV charging posts play the role of the aggregators, each one incorporating the necessary intelligence to decide and forward the proper information or instructions to the corresponding target area. Based on this architecture, the required intelligence is assembled in the aggregators and not in the roadside units (RSU) that appear so far in the literature [6], [8], [9]. The deployment of RSUs in urban environments is a challenging task that requires the consideration of several factors, such as mobility patterns, operating and propagation conditions, budget restrictions, as well as network design and operation. Removing the need for RSUs and shifting the coordination tasks to the EV charging stations/aggregators provides potentially significant cost savings in capital and operational expenditures.

III. Development of Simulation Model for the NA

A. Simulation Environment

The present work employs the discrete-event network simulator for Internet systems ns-3 [10], which is the third of a series of open-source network simulators that follows the architectural design concepts of the Georgia Tech Network Simulator (GTNetS) [11]. Ns-3 is mainly developed in the C++ language; but it comes with an optional Python scripting interface which can be used for parts of the simulation.

B. Simulation Model

This paper builds upon the VANET-highway example in [12], which is an extension of the model introduced in [13]. The original model is based on two classes, namely Vehicle and Highway. A Vehicle represents a mobile node (containing information such as acceleration, velocity and position) with wireless communication capabilities. The Highway class is the core of the model and uses the Model and LaneChange objects attached to Vehicles to move vehicles based on the Intelligent Driver Model (IDM) and Minimizing Overall Braking Induced by Lane Change (MOBIL) [14].

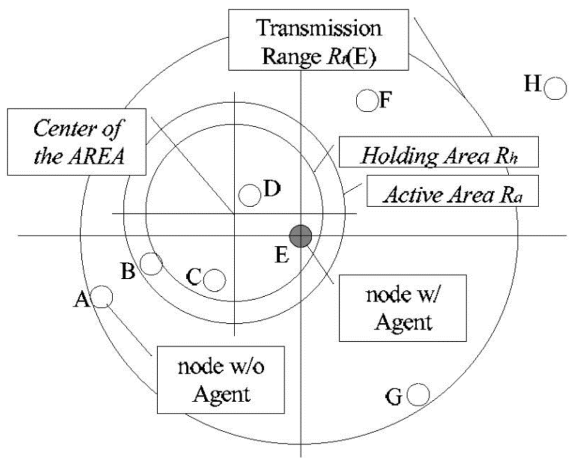


Fig. 1. Nomadic Agent and its active area (retrieved from [8]).

However, the limitations of this model, e.g. a Highway can only run east-west and there is no possibility to create intersections, render it unsuitable for simulating the conditions in urban environments. To increase the capabilities of the simulator, several improvements are introduced in [14], including the enhancement of the Highway class to point in any direction, as well as the storage of most configuration parameters in an XML document to facilitate the reuse and reconfiguration of a complex simulation after its initial set-up.

1) HighwayProject Class

The purpose of the HighwayProject class is to process the XML document that contains the set-up of the Highways, connect the various classes and perform the simulation. Based on the configuration parameters in the XML document, a routing map is created for each Highway instance to other Highway instances using the Dijkstra's algorithm [15], so that each one can direct a vehicle left, right or straight forward. With the Highways created and mapped, the HighwayProject class creates instances of WifiConfiguration and VehicleGenerator [14]. The output of this class consists of two trace files for logging vehicle location and network traffic respectively. The vehicle trace file contains information, such as simulation time, vehicle ID, velocity, x and y position. At each time step of the simulation, one record is created for each existing vehicle. The network trace file contains information about packet reception; however this is not logged for every vehicle, but only for the NA at that time. Other types of messages can also be added by using the provided callbacks in the class, e.g. log of receipt errors.

2) Highway Class

The Highway class is responsible for managing the Vehicle instances it contains at the current simulation time step, as it is a container of Vehicles rather than a class that generates Vehicle instances on its own. Each Highway is defined by the following parameters [14]: Start location, Length, Number of lanes, Lane width, and Direction. The Highways can be connected by stating the order in the XML file. Each lane has an ID, which can be referred to by other lanes to state their way of connection. Specifically, when a lane A is directly followed by another lane B, the follow-up (B) is referred to as a frontHighway connection for A, and vice versa, i.e. lane A as a backHighway for lane B. The present work considers the case of a simple set-up consisting of 3 Highways, each one with 2 lanes and a length of 100 m: 2 highways are straight connected, parallel to the x-axis, and the third highway goes to the left, as illustrated in Fig. 2. After the definition of the simulator's parameters, such as total runtime and time step, the Highways are declared and configured. For instance, the parameters leftTurnSpeed and rightTurnSpeed are employed to create a more realistic approach of vehicle behavior: when a Vehicle approaches a left or right turn, its (desired) speed is reduced linearly to either the left- or rightTurnSpeed value respectively, and it will try to regain its normal speed after the turn.

3) Vehicle Movement

In each step of the simulation, the HighwayProject calls a step function on all Highways to update the vehicles' positions. A Vehicle's acceleration is based on the Vehicle's distance to the Vehicle instances in front and behind it, as calculated in the Highway class. The latter also attempts lane change calculations every tenth step, triggering the LaneChange objects that are associated with single Vehicles. When these calculations are performed, the system looks into the routing map to make sure that the Vehicle does not attempt to change lane to a non-desirable one, e.g. a lane that would lead to a left turn if the vehicle needs to go straight forward.

4) WifiConfiguration Class

The role of the WifiConfiguration class is to hold the Wi-Fi parameters that are not likely to change during the simulation, so that multiple VehicleGenerators can share the same Wi-Fi configuration. The IEEE 802.11g standard is employed in the proposed model as a representative example of a widely spread wireless communication standard for current smartphones using mobile hotspots.

5) VehicleGenerator Class

Given that the HighwayProject class is responsible for the generation of Vehicles through VehicleGenerator instances and the Highway class is a container of Vehicles, a Highway instance is passed in the constructor of the VehicleGenerator that is used by the latter to determine how often it should evaluate. The VehicleGenerator randomly increments a counter in each evaluation and attempts to inject a new Vehicle when a predefined limit is exceeded. If the lane to inject the new Vehicle is reserved by another vehicle, then the VehicleGenerator waits until the latter has passed at least a predefined minimum distance from the start of the Highway. This minimum gap along with other relevant parameters, such as the maximum and minimum values of the Vehicle's desired speed, as well as the destination map to determine its route, can be configured in the VehicleGenerator class.

6) Vehicles and Models

The Vehicles are the primary entities of the simulator and each one is characterized by its dimensions (length and width), velocity, acceleration, direction, and the lane in which it is currently travelling. Each Vehicle has also a Model associated with it; this Model class contains the IDM implementation that calculates the acceleration of the vehicle, taking into consideration the desired velocity, maximum acceleration and braking, as well as the minimum gap to maintain.

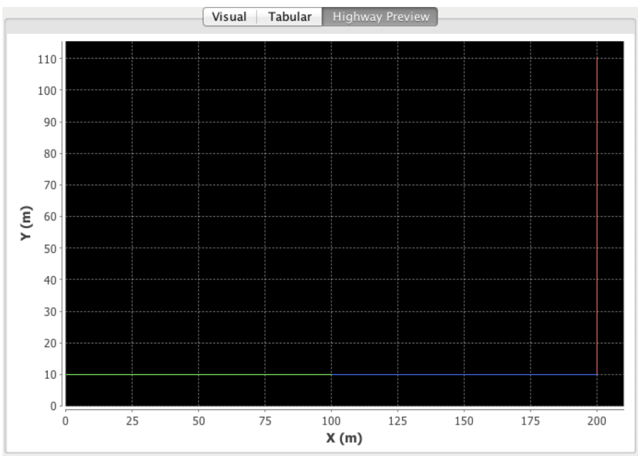


Fig. 2. The intended highway set-up.

7) Area Class

The objects detailed so far are based on existing implementations with the ns-3 simulator, assuming a model that injects vehicles at configurable or semi-random intervals, on a highway with multiple lanes that can be connected in a user-defined way. The highways can be placed anywhere and in any direction in a 2-dimensional plane. However, the main contribution of the present work in this respect is the implementation of the NA concept. To this end, the Area class is implemented in the proposed model in order to represent the active area for the NA. An instance of the Area class is defined by a center point and a radius, representing a circular area. The Area class also contains the method GetTypeID inherited from the Object class in ns-3, a method to determine whether a given point is inside the range of the area, and another method to calculate the distance from a given point to the center.

8) Nomadic Agent Holder and Migration Technique

To define whether a Vehicle is currently the NA holder, a new parameter is added to the Vehicle class: a Boolean variable called NA_holder states whether or not the Vehicle currently holds the NA functionality. Then, the Area class is employed to simulate the migration procedure in the NA concept: when a Vehicle is currently the NA holder, a new NA has to be found as soon as the current one exits the area. The method implemented here searches for the vehicle closest to the center of the area. This approach requires multiple loops: given that the simulator generates for each step the output for the vehicle trace file, the final situation for each vehicle needs to be determined. At this point, it is important to note that simply looping over the list of Vehicles would result in errors regarding the integration of the migration technique.

Let us consider the following example: Vehicle A is currently the NA, but exits the area. The simulator will loop this step through the entire list of vehicles and put their location and output in the trace file, including the value of the parameter NA_holder. In this case, a logical error occurs when the migration process is being applied to a selected Vehicle with an ID lower than that of Vehicle A: the simulator would generate wrong output, since the new NA's output is already written (incorrectly). Therefore, it is necessary to capture the output stream and properly process it before it is written to the vehicle trace file. To create a correct representation for the output file, the stream is written once the entire vehicle list is checked, updated and the migration techniques that might occur are performed, while adding for each vehicle the NA_holder value at the end.

To address the aforementioned issue, all the Vehicles are included in a total list, which is constructed in the step function and stored in the variable totalList. Next, the iteration over this total list starts and if necessary the migration technique is applied: The migration algorithm checks if the vehicle currently considered is the NA, and if it is outside the area. If these conditions are true, it follows that a new NA should be assigned. For this purpose, the algorithm loops again through the entire Vehicle list and checks for each Vehicle inside the area what its distance to the center of the area is. Next, the one with the shortest distance is picked and – if it is not the same Vehicle with the previous NA – the algorithm sets the old NA inactive (NA_holder value is set to false), and assigns the NA functionality to the new Vehicle. Given that all the Vehicles have been examined and thus the information in the instances is correctly updated, the final output can be generated by iterating once again through all the Vehicles in order to write for each one the corresponding data to the vehicle trace file; including the NA_holder parameter.

9) Packet Transfers

The network transfer is applied using packets transferred through callback statements: each *N* time steps, the Vehicle passed in the parameter list of the callback creates a packet and sends it to the broadcast address. The amount of time steps can be configured, although making it too small causes the model to flood (see next section). In the proposed model, only the non-NA Vehicles inside the Area broadcast their information when selected, as shown in the excerpt of C++ code in Fig. 3. The term selected is used in this context to denote that a relevant "message" counter reaches its configured value and the present Vehicle is allowed to send its information if it meets the required conditions. As for the packet receiving mechanism, only the NA is allowed to accept these packages according to the proposed concept, i.e. Vehicles in the Area broadcasting their information to the NA at that moment. Therefore the packet receiving mechanism checks if the Vehicle considered is currently the NA holder, and if so, it writes this information to the network trace file.

IV. Simulation Results

A. Vehicle Trace Information

The VehicleViewer [12] is a Java tool that reads the vehicle trace file created by the ns-3 model and displays the relevant information on a 2-dimensional plane. Using the VehicleViewer tool as a base, a few parameters are added to implement the NA functionality and demonstrate the migration system. Indicatively, the active area and the vehicle that currently holds the NA are marked by using annotations (i.e. shapes drawn on top of the plot of a figure). In this example, the active area is centered at (175, 30) with a radius of 35 m (Fig. 4), following the Highway set-up in Fig. 2.

```
if (msgCounter == 999 && area->insideArea(veh->
    GetPosition()) && !veh->isNA_holder()) {
        stringstream msg;
        msg << veh->GetVehicleId()
        << " x=" << veh->GetPosition().x
        << " y=" << veh->GetPosition().y
        << " v=" << veh->GetVelocity()
        << " d=" << veh->GetDirection()
        << " l=" << veh->GetLane();

        Ptr<Packet> packet = Create<Packet>((
    uint8_t*) msg.str().c_str(), msg.str().
    length());
        veh->SendTo(veh->GetBroadcastAddress(),
    packet);

    }
    msgCounter = (msgCounter + 1)%1000;
```

Fig. 3. Code excerpt of packet sending mechanism.

Once the simulation starts, the vehicles (represented as red square boxes) start coming from the left side, going towards the area. As a kick-start, the first node entering the active area is initially assigned the NA functionality to initiate the operation of the system. Fig. 5a depicts the first node approaching the area, while Fig. 5b shows that it is marked as soon as it enters the active area, indicating that the NA has been assigned to it. Then, the node continues to follow the configured highways, and when it leaves the area, the migration technique is initiated: the system searches for a vehicle inside the area that is closest to the center in order to assign the NA to a new node. As a typical example, Fig. 6 graphically represents the first migration of the NA. Specifically, the figure on the left shows that the current NA holder is about to move outside the boundaries of the active area, while the figure on the right shows that the NA process is terminated in that mobile node and migrates to the vehicle closest to the center, so that the NA functionality is kept within the area of interest. Moreover, Fig. 7 demonstrates the fact that the present work considers the movement of vehicles under real-world driving conditions through with the LaneChange functionality, where a vehicle (top left node in Fig. 7a) may accelerate in order to overtake another vehicle (Fig. 7b) and change lane (Fig. 7c).

Apart from the graphical representation, the VehicleViewer tool also contains a tab that shows the information of the present vehicles on that specific simulation time in tabular form (Fig. 8). For the purposes of this work, the original tool is modified to keep track of not only Vehicle ID, type ID, Location, Direction, Velocity, and Acceleration, but also the two Boolean parameters that indicate whether a given vehicle is inside the area and it holds the NA functionality respectively.

B. *Network Trace Information*

The network trace information is related to the packets sent by a vehicle inside the area and received by the NA. Specifically, it consists of the simulation time, (receiving) vehicle ID, a string indicating the message type, and the two parameters indicating the presence of the vehicle within the area and whether it is the NA holder, as shown in Table I.

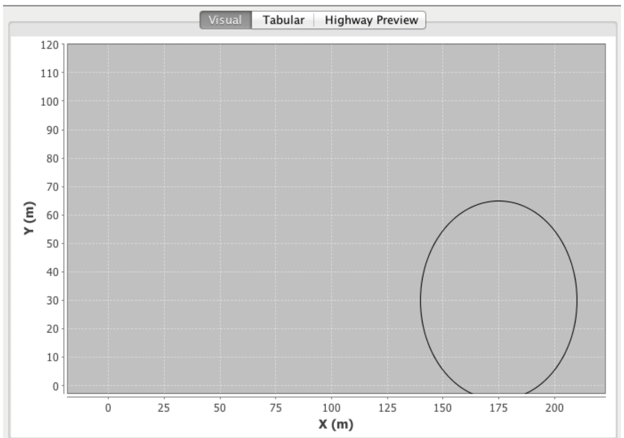
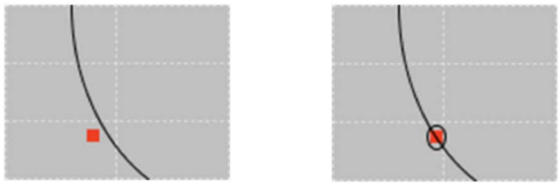
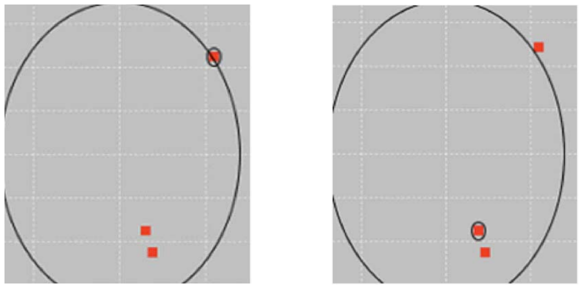


Fig. 4. VehicleViewer showing the active area.



(a) (b)

Fig. 5. First node kick-starting the migration technique: (a) Mobile node out of the active area, and (b) Activation of NA holder.



(a) (b)

Fig. 6. First migration of the NA: (a) NA holder exiting the active area, and (b) Assignment of NA to a new node.

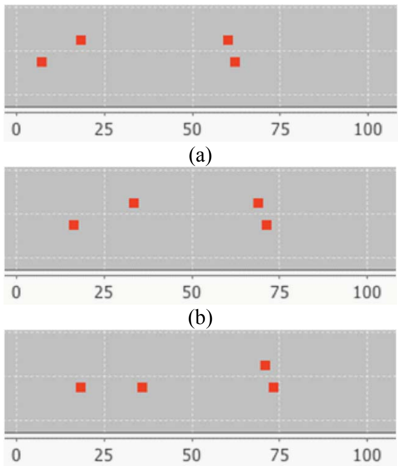
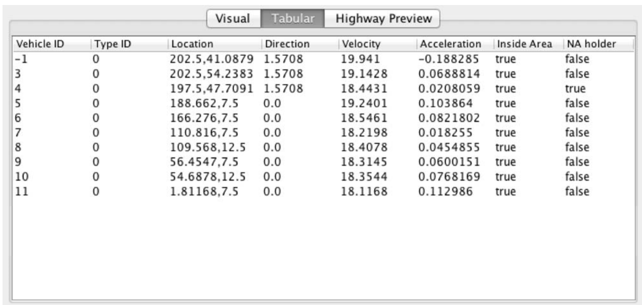


Fig. 7. Simulation of real-world driving situations: (a)-(b) A vehicle overtaking another, and (c) A vehicle changing lane.



Visual | Tabular | Highway Preview

Vehicle ID	Type ID	Location	Direction	Velocity	Acceleration	Inside Area	NA holder
-1	0	202.5,41.0879	1.5708	19.941	-0.188285	true	false
3	0	202.5,54.2383	1.5708	19.1428	0.0688814	true	false
4	0	197.5,47.7091	1.5708	18.4431	0.0208059	true	true
5	0	188.662,7.5	0.0	19.2401	0.103864	true	false
6	0	166.276,7.5	0.0	18.5461	0.0821802	true	false
7	0	110.816,7.5	0.0	18.2198	0.018255	true	false
8	0	109.568,12.5	0.0	18.4078	0.0454855	true	false
9	0	56.4547,7.5	0.0	18.3145	0.0600151	true	false
10	0	54.6878,12.5	0.0	18.3544	0.0768169	true	false
11	0	1.81168,7.5	0.0	18.1168	0.112986	true	false

Fig. 8. Vehicle trace information in VehicleViewer tool.

The packet delivery latency, which is equal to the interval between the time of sending the packet and the time of delivery at the NA holder, can be calculated by combining the network trace information with the callback of the packet sending function. For the purposes of this work, two simulations were performed with run length of 300 s, but different time steps, i.e. 0.1 and 0.08 s, in order to assess the system performance in each case. Figs. 9 and 10 illustrate the corresponding packet delivery latency at different points in time during each simulation, indicating also that the average value is 0.127 and 0.128 ms respectively. Given that the packet sending mechanism in a regular time interval is based on the message counter of the code listing in Fig. 3, altering the time step size to a lower value generates more messages during a simulation run. Therefore, the latency measurements with smaller time step provide a more reliable sample to calculate the average latency. However, the observed difference between the two cases in terms of average packet delivery latency is small, supporting the validity of the results obtained in both cases. As a side observation, it is noted that further reducing the time step caused the simulation to flood, an error generated by the Wifi channel helper class from the ns-3 simulator that was employed.

TABLE I. NETWORK TRACE INFORMATION

Simulation time (nanoseconds)	Vehicle ID	Message type	Inside area	NA holder
37120132018	22	Vehicle Receive	1	1
48240120130	30	Vehicle Receive	1	1
52000128018	34	Vehicle Receive	1	1
78160124137	56	Vehicle Receive	1	1
85600128017	62	Vehicle Receive	1	1
126720132018	96	Vehicle Receive	1	1
137920132017	106	Vehicle Receive	1	1
152960132022	118	Vehicle Receive	1	1
171680124149	134	Vehicle Receive	1	1
175360124144	136	Vehicle Receive	1	1
186640132021	146	Vehicle Receive	1	1
212720132029	168	Vehicle Receive	1	1
216400132023	170	Vehicle Receive	1	1
238880124158	190	Vehicle Receive	1	1
242560124127	192	Vehicle Receive	1	1
261280132025	208	Vehicle Receive	1	1
272560124137	218	Vehicle Receive	1	1
294880132020	236	Vehicle Receive	1	1

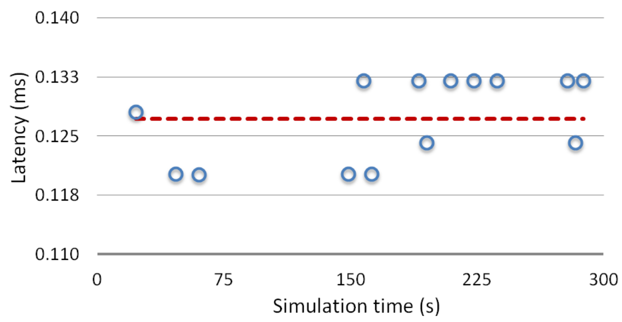


Fig. 9. Packet delivery latency with time step of 0.1 s.

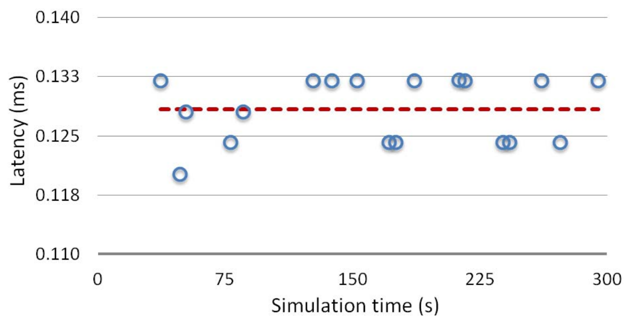


Fig. 10. Packet delivery latency with time step of 0.08 s.

V. DISCUSSION

The results presented in the previous section demonstrate the successful implementation of the migration technique under the simulated conditions, since there is always a new NA assigned when the previous NA holder leaves the active area, considering that the latter is occupied with vehicles. Furthermore, the network packets are consistently delivered to the NA with a reasonable latency, regardless of which vehicle is hosting it. In this regard, the results confirm the stability of the simulated system in an urban environment.

The highway set-up used in the proposed model is a small-scale example; yet representative of the conditions met in urban areas. In particular, the present work examines the mobile nodes and their communication within the active area of the NA holder, considering the structural characteristics of the urban environment as well as driving conditions that are usually met in real-world situations. Hence, it provides an enabling framework to assess the potential of the proposed concept on a realistic basis. The base model also supports the development of simulations that combine the creation of intersections with the functionality of traffic lights through the TrafficLightGenerator class, allowing for a more detailed representation of urban traffic conditions. Moreover, the current implementation considers the case of sudden death (termination) of the NA functionality, when the NA holder exits the active area and no suitable follow-up can be found. Under these circumstances, this approach results in information loss and a new NA holder must be assigned once the active area is populated again with vehicles. As an alternative, the so-called "multi-hop recovery" may be employed in order to save the information within the NA, even if the NA holder moves out of its active area. In this case, if the NA is unable to return back directly, it tries to migrate temporarily to another vehicle near the active area as an intermediate step towards its return inside the active area. However, this approach not only requires the development of recovery algorithms that take into account the moving direction of surrounding vehicles for the selection of the next NA holder, but may also result in delays until the NA finally migrates inside the active area.

Moreover, the simulation model developed within the frame of the present work can be considered as the building block for the provision of V2I communication services based on the NA approach in order to support electro-mobility in an urban environment. Going a step further to achieve full coverage of a given urban area requires the deployment of multiple NAs, each one being responsible for providing location based information within its active area. The simulation of multiple areas allows for the assessment of the scalability of the proposed approach; however it is out of the scope of this work.

VI. CONCLUSION

The present paper focuses on the realistic representation of the mobile nodes and their communication within the active area of the NA holder, considering the structural characteristics of the urban environment. The simulation results obtained from the ns-3 model that was developed for this purpose provide a first indication that the NA concept offers a suitable communication solution for the application under study. To fully assess the potential scalability of the envisaged system requires the consideration of multiple active areas in order to cover the urban web of a city. Therefore, directions for future work include addressing of migration issues related to overlapping of different active areas, modeling and simulation of the communication not only between the NA and the aggregators of the smart grid (e.g. EV charging posts) but also between the aggregators, as well as further improvements of communication modeling to include distortions, propagation losses and delays.

REFERENCES

[1] I. Leontiadis, G. Marfia, D. MacK, G. Pau, C. Mascolo, and M. Gerla, "On the effectiveness of an opportunistic traffic management system for vehicular networks," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1537-1548, December 2011.

[2] M. Alsabaan, K. Naik, and T. Khalifa, "Optimization of fuel cost and emissions using V2V communications," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1449-1461, September 2013.

[3] J.-B. Tomas-Gabarron, E. Egea-Lopez, and J. Garcia-Haro, "Vehicular trajectory optimization for cooperative collision avoidance at high speeds," IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1930–1941, Dec. 2013.

[4] K.N. Genikomsakis, C.S. Ioakimidis, B. Bocquier, D. Savvidis, and D. Simic, "Electromobility and carsharing/carpooling services at the University of Deusto: A preliminary exploratory survey," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, The Hague, 2013, pp. 1935-1940.

[5] N. Matta, R. Rahim-Amoud, L. Merghem-Boulahia, and A. Jrad, "A cooperative aggregation-based architecture for vehicle-to-grid communications," in *Proc. Global Inform. Infrastructure Symp.*, Da Nang, 2011.

[6] Y. Han, Y. Chen, F. Han, and K.J.R. Liu, "An optimal dynamic pricing and schedule approach in V2G," in *Proc. 4th Asia-Pacific Signal and Inform. Processing Assoc. Annu. Summit and Conf.*, Hollywood, CA, 2012.

[7] X. Cheng, X. Hu, L. Yang, I. Husain, K. Inoue, P. Krein, R. Lefevre, Y. Li, H. Nishi, J.G. Taiber, F.-Y. Wang, Y. Zha, W. Gao, and Z. Li, "Electrified vehicles and the smart grid: The ITS perspective," IEEE Trans. Intell. Transp. Syst., vol. 15, no. 4, pp. 1388-1404, August 2014.

[8] T. Yashiro, "A new paradigm of V2V communication services using Nomadic Agent," in *Proc. 3rd Annu. Int. Conf. Mobile and Ubiquitous Syst.: Networking and Services*, San Jose, CA, 2006.

[9] J. Gozalvez, M. Sepulcre, and R. Bauza, "IEEE 802.11p vehicle to infrastructure communications in urban environments," IEEE Commun. Mag., vol. 50, no. 5, pp. 176-183, May 2012.

[10] *The ns-3 network simulator: A discrete-event network simulator* [Online]. Available: http://www.nsnam.org

[11] G.F. Riley, "Simulation of large scale networks ii: Large-scale network simulations with GTNetS," in *Proc. of 35th Conf. Winter Simulation: Driving Innovation*, 2003.

[12] B. Dupont. *ns-3-highway-mobility* [Online]. Available: http://code.google.com/p/ns-3-highway-mobility/

[13] H. Arbabi and M.C. Weigle, "Highway mobility and vehicular ad-hoc networks in ns-3," in Proc. Winter Simulation Conf., Baltimore, MD, 2010, pp. 2991-3003.

[14] B. Dupont, "Improvements in VANET Simulator in ns-3," M.S. thesis, Dept. Comput. Sci., Old Dominion Univ., Norfolk, VA, 2011.

[15] E.W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269-271, December 1959.