TOWARDS A MICROSCOPIC TRAFFIC SIMULATION FRAMEWORK TO ASSESS VEHICLE-TO-VEHICLE NETWORKS

João F. B. Gonçalves, Rosaldo J. F. Rossetti, Edgar F. Esteves, Eugénio C. Oliveira
Department of Informatics Engineering (DEI)
Artificial Intelligence and Computer Science Lab (LIACC)
Faculty of Engineering, University of Porto (FEUP)
Rua Dr. Roberto Frias, 4200-465 Porto, Portugal
E-mail: {ee02123, rossetti, edgar.esteves, eco}@fe.up.pt

KEYWORDS

Intelligent transportation systems (ITS), service-oriented architectures (SOA), vehicular *ad-hoc* networks (VANET), vehicle-to-vehicle (V2V)

ABSTRACT

This paper presents the specification of a framework based on the concept of service-oriented architectures (SOA) to support the assessment of vehicular ad-hoc networks (VANET). A preliminary study of concepts related to SOA was carried out, as well as of those technologies that allow real-time data acquisition and dissemination within urban environments, and simulation tools to aid the simulation of the VANET. The requirements for our simulation framework were identified and a two-layered architecture was specified, which rely on the abstraction levels of services for vehicle-to-vehicle (V2V) communication. A prototypical application was implemented, which was used to demonstrate the feasibility of the approach presented through experimental results.

INTRODUCTION

The number of people living in urban areas has increased very quickly and unquestionably. Unfortunately, most people opt for using private car instead of public transport. This trend has turned infrastructures unable to cope with the ever increasing demand, which work most of the time in saturation regime. This problem motivates the scientific community that strives to devise different solutions. Some approaches suggest the physical improvement of infrastructure by means of increasing road capacity. Others try to enhance the control system responsiveness, with relative success. More recently, though, researchers have experimented promising innovative information technologies to aid driving tasks and traveller decision-making. The latter ones underlie the concept of intelligent transportation systems (ITS), which attempts at integrating contemporary and breakthrough computer and information technologies to better manage and control modern urban transportation systems. ITS are intended to turn future urban transport (FUT) into greener, cheaper, reliable, secure, and sustainable systems, both functionally, energetically and environmentally efficient as well.

The work reported in this paper is based on our study of the communication technologies underlying vehicle-to-vehicle interactions. High-class vehicles will very soon be equipped with short-range wireless communication interfaces, which will allow them to form an ad-hoc communication network. V2V networks of such a genre are being studied already and motivating much advances in ITS-based technology. The scientific community believes information disseminated through these networks will play an imperative role in future urban traffic network security and efficiency. This new application domain brings about major concerning issues, though. Thus, simulation tools will need to be adapted to support the assessment of V2V communication infrastructures and performance, allowing researchers and practitioners to define which architecture is most adaptable to urban traffic scenarios, for instance.

In this work, we aim at studying concepts related to SOA and ways in which such concepts can be adapted to the VANET. The work started by the specification and implementation of a simulation framework that will allow us to gain further insight into such motivating and challenging new arena. Such a framework is described in more details and some preliminary results are discussed.

This work is organised as follows. In the next section, we review some important concepts such as SOA, V2V communication networks and some simulation tools already available, in the light of concepts we are going to use in the proposed approach. In section three, the development of a simulation prototype is discussed and detailed, whereas in section for some experimental results are discussed and commented. Finally, we draw some conclusions in the last section, point up directions for further developments and opportunities for future works.

FUTURE URBAN TRANSPORT TECHNOLOGIES

ITS-based technologies have proven a great impact and influence in future urban transport scenarios. As the automotive industry starts marketing vehicles equipped with wireless communication capabilities, some technologies are being deemed to be as potentially applicable and beneficial.

Service-oriented Architectures

Service-oriented architectures are devised as software architectures whose main goal advocates that application functionalities must be made available in form of services. Thus, services from SOA's point of view are functions of a computational system that are made available to other systems. Such a novel approach might be well represented by the "find-bind-execute" paradigm, which is analogous to the Deming's cycle applied to services, involving planning, execution, monitoring and pro-active decision-making phases to improve systems' performance. The "find-bind-execute" paradigm allows a consumer of a service to ask for registering in a service provider that suits its criteria and requirements. If the service of interest is found, the provider sends the consumer a contract and the address in which the service can be found. Such a mechanism is depicted in Figure 1.

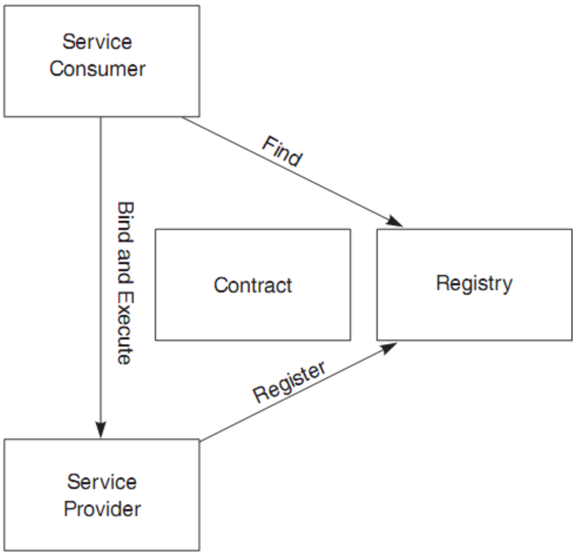

Figure 1: Find-Bind-Execute Paradigm
(adapted from McGovern et al. 2003)

According to McGovern et al. (2003), the entities that support such a service infrastructure are:

- *Service consumer*, which might be basically an application, a service, or another type of software requiring a service. This is the entity who initiates the process, looking for a service capable of supplying its requirements. The consumer executes the service by sending a request formatted in accordance to the contract.
- *Service provider*, which is the addressable entity that accepts and executes the service. This entity might be a mainframe, a component or another type of software that executes a requested service. Service providers publish their contract in the service registry for other potentially interested consumers to access them.
- *Service registry*, which is a sort of network directory that "knows" the available services. This entity accepts and stores contracts of providers and provides consumers with these service options.
- *Service contract*, which is sort of a specification of the way in which a service consumer must interact with the provider. This entity rules the protocol for service request and respective answers from the services. A contract may require a set of pre- and post-conditions representing the state of a service to be deemed acceptable to execute certain functions. Contracts may also contain information concerning the quality of services, as well as certain conditions to which consumers must comply.

Vehicle-to-vehicle networks

Vehicle-to-vehicle communication networks are, as its designation suggests, networks formed by several vehicles equipped with wireless communication devices that can communicate with each other. In V2V networks, each vehicle analyses, within a certain radius, the other vehicles that are in range, and can inform its position, velocity, direction and other characteristics. This kind of communication has been one of the fields of interest in telecommunications that grew very quickly lately. Thus, vehicles with such capabilities can form a special type of mobile ad-hoc network with particular applications, known as Vehicular Ad-hoc Networks (VANET). VANET are a special type of Mobile Ad-hoc Networks (MANET) that supports communications between vehicles. According to Mello and Endler (2006), the VANET inherit some characteristics from the MANET, but also improve the former with new characteristics, which differentiate it from other ad-hoc mobile networks. These characteristics include high mobility, open network with dynamic topology, limited connectivity, potential to achieve larger scale, all nodes are providers, forwarders and consumers of data and the wireless transmission can suffer from much noise and interferences.

These characteristics make the VANET sufficiently different from other networks and significantly affect their properties. For example, in (Chen et al. 2001) is demonstrated that the movement of vehicles has a significant effect on the latency of messages delivery. Most applications that can be implemented on a MANET require a certain type of data dissemination. Studies argue, however, this must be implemented through specific routing protocols, as long as VANET are concerned, basically due to their particular characteristics, such as in (Blum et al. 2004). Nonetheless, there are other studies that suggest it is possible to use the available MANET protocols. Some of those protocols are designed to support dedicated short range communications (DSRC), which is already implemented in USA, such as VITP and PAVAN. Due to its topology, the MANET already have a large set of protocols.

There are several ways to classify routing protocols for such networks, some of which are listed below:

- According to the range, they can be either *unicast* or *multicast*.
- According to the route discovery, they can be either pro-active, reactive or hybrid.
- According to the search algorithm they are based on, they can be either Distance Vector, Link State, based on geographic information, or Zone based.

The *unicast* protocols are those that transmit information from one transmitter to one receiver. In contrast, multicast protocols are those in which information is sent to a previously created group of nodes.

In what refers to route discovery, the pro-active algorithms are those that periodically update the network routes. These particular algorithms have, at every moment, the knowledge of the network topology, trying to get the optimal route for the time that is necessary to send information. The exchange of control packets and the update of routing tables are continuous and all nodes are known in advance. To the contrary, there are reactive algorithms (also known as on-demand ones) that only discover the route to a destination when they need to send information. In these algorithms, the route discovery is done on-demand, eliminating the permanent routes. Hybrid algorithms combine the formers' good characteristics using, for instance, pro-activity in the node neighbourhood and discovering the route to distant nodes at the exact moment it is need only.

Now, as for the type of algorithms on which the routing protocols are based, Distance Vector algorithms require that all nodes in the network exchange their distance vectors periodically, meaning each node knows the cost information for each destination; each router (node) maintains a table (a vector) containing the cheapest known route to each destination. These tables are updated by exchanging messages with neighbours and comparing the received table with their own table. In the case of a better route is found, the router updates its table and stores the source of this information too. In Link State algorithms, each node monitors the state of the link with its neighbours and disseminates this information. Each node knows the status of all links, and determines the complete topology of the network as well as the locally shortest path. On the other hand, geographic information based algorithms are able to estimate the location of the nodes and the information is sent in which direction. In turn, in zone based (source routing) algorithms, one route is only created when the source node demands it, after which creation the route is maintained by a maintenance protocol and there is no need for periodic updates.

In the Figure 2, the protocols previously mentioned are related, as well as are their classification according to the characteristics mentioned above.

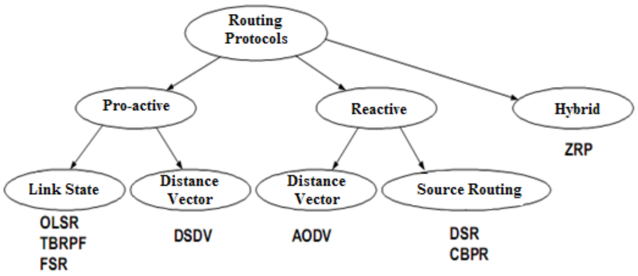

Figure 2: MANET Routing Protocols Classification

Simulation tools for V2V networks

The development of applications and protocols associated to VANET can be studied through simulation, especially when a real traffic network in urban environments, which must involve a large number of nodes, is the subject of study. Through simulation models, the performance of V2V networks, as well as other characteristics can be assessed and improved. Indeed, conducting experiments and studies on such a large scale within the real scenario has proven extremely difficult and expensive. Thus, simulation has become an indispensable and even an imperative tool.

Basically, the simulation of V2V networks requires two different components, namely a communication networks simulator, capable of simulating the properties of a wireless network, and a vehicular traffic simulator, able to monitor and represent the kinematic aspects of mobility throughout the VANET nodes. Recent studies (Choffnes and Bustamante 2005) suggested that the vehicular mobility model is very important to obtain significant results and should be well integrated with the wireless communication networks model. Other authors further suggest that the use of an inappropriate model, such as the popular "random waypoint model" (which can work very well for some Mobile ad-hoc networks, but very likely is not an appropriate representation of mobility in wireless vehicular networks) can lead to erroneous results (Chooffnes and Bustamante 2005; Saha and Johnson 2004).

Nevertheless, traffic simulators have been subjected to enormous developments and greatly improved in recent years to include communication between vehicles. Several ways to achieve such an advanced feature have been proposed and actually implemented. The greatest trend in most studies moves towards the creation of simulators that include traffic and wireless communication simulation models in one single simulation tool. Some examples of such tools include GrooveNet and Divert.

However, other studies prefer to use two independent simulators, combining stand-alone traffic and communication networks simulators, which are interconnected by an application that ensures the exchange of information between them. Among traffic simulators used for this purpose, one can mention the popular SUMO, as well as VISSIM and CARISMA. Wireless communication networks simulators include GloMoSim, QualNet and the NS2.

ARCHITECTURAL PROTOTYPING

The Layered SOA Structure

Currently, the services that both drivers and travellers in general can use on-board in journey time demand a great deal of hardware and software. Each new feature must be implemented in a new device to be embedded in the car. Such an approach has proven very expensive, with little flexibility, which contradicts the increasing trend of services made available on-demand, for instance. On the other hand, services made available as software to be executed in an onboard unit (OBU) based on an embedded computer with considerable processing and memory, as well as communication capabilities seem to be very promising and present great potentials and advantages. According to such not-so-much futuristic scenario, we intend to specify and implement an extensible architecture based on OBU computers and featured with communication capabilities to the level of services.

Bearing in mind such architecture was intended to promote services throughout V2V communication networks. We then opted for a layer-based structure. Thus, the proposed architecture is divided into two main levels, namely the network services level and the end-user services level. This approach is illustrated in Figure 3, in which the two different levels are identified. The first level in such a structure is responsible for network-related tasks, such as building network topology, as well as discovering and exchanging services among vehicles (the nodes of the communication network). The second level, on the other hand, implements the necessary basis underlying the management of the so called high-level services, to be made available to end-users.

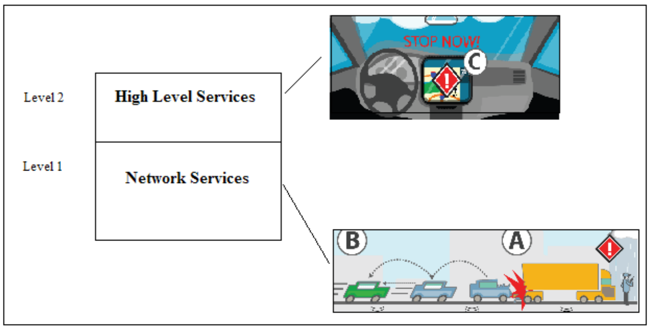

Figure 3: SOA Layered Architecture for V2V Networks

For the first level, and accounting for its functions within the proposed structure, we adopted for the solution suggested in (Halonen and Ojala 2006). The scenario illustrated in Figure 4 is a good representation of the adopted solution, as explained below.

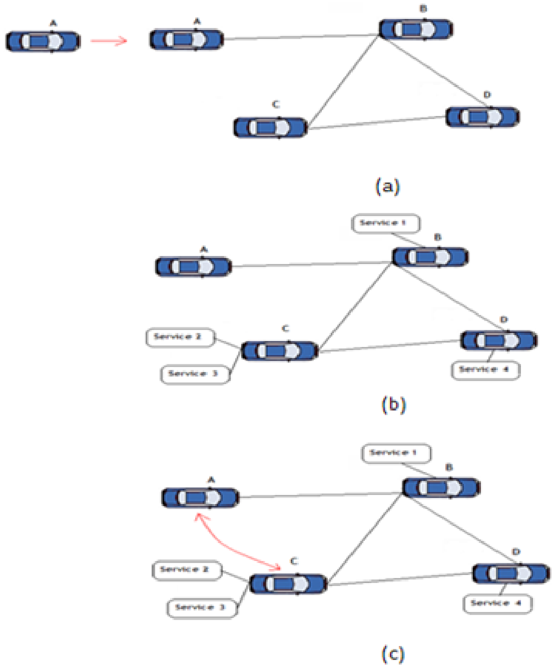

Figure 4: SOA-Based V2V Communication Interactions

Let us consider the communication network formed up by the vehicles as illustrated in Figure 4. In (a), the network is actually set up. All nodes (A to D) are vehicles equipped with mobile wireless V2V communication devices. Initially, the A node is not connected for there is no other vehicle within its range vicinity. As soon as it finds other vehicles within its range, then they get connected. All connected nodes thus form a VANET, allowing V2V communication. After all the necessary automatic configurations are set up, the VANET is ready for routing any messages sent by any node and addressed to any other node of the network.

After the VANET is then ready, the SOA-based features must be deployed too. All nodes that provide any service must publish it alongside a comprehensible description, so that other nodes will be able to discover the service and use it as needed. The service discovering capabilities of each node must be implemented in such a way that the node may be able to find the needed service throughout the VANET whichever configuration must apply. Thus, in (b), every provider node publishes its services, whereas any consumer will be able to find them whenever necessary.

An illustrative service use example is demonstrated in (c). The A node gets connected to the service interface of service whose ID is 3, hosted in node C. All information necessary to message exchange between consumer (A node) and provider (C node) is available in the description of service 3, needing A no further information about node C. As A and C nodes are not directly connected to each other, messages must be routed through the B node.

Two basic components are necessary for the example presented above to be possible, namely a routing protocol for the VANET and the implementation of SOA functionalities. One of most important SOA features is the service discovering mechanism. Two approaches are then possible for us to implement such a mechanism. First, we can integrate it within the VANET routing protocol or, alternatively, we can implement it on top of the network layer, as a separate functionality. In the current work we opted for the first solution, and integrated it in the VANET routing mechanism. Such a decision proved advantageous as both the routing tables and protocol routing techniques are used in the service discovering process. For this purpose, we chose the optimised like state routing protocol (OLSR) for it is a pro-active protocol, which facilitates the propagation of services description throughout a network of known topology. Among pro-active protocols, OLSR has the advantage of using multipoint relaying (MPR) and offers an open source application (OLSR *Daemon*) allowing it to be extended.

Routing in MANET is based on the cooperation of participating nodes; all nodes must run the protocol in such a way they collectively achieve routing goals. Such a behavior is also desirable in VANET, as all nodes are expected to collaborate in services discovery as well. In practice, it means all nodes manage message routing even though they may not know its content. This way all service provider nodes may use the entire network and routing protocol.

A node intending to publish its services must generate and propagate SOA messages periodically, carrying the service information, according to certain transmission interval. Whenever a node receives a SOA message, it must resend it and, if it supports SOA services, process it too.

The format of a SOA message is presented in Figure 5. The format has two basic fields, namely the message length and the service description, which describes the service as a unique service (no other service will have the same description). This sort of message is carried within OSLR packages and propagated throughout the network.

Nonetheless, there is no specific format for the content of a service descriptor, which will basically depend on the type of service and the way it is implemented. XML files might be used for this purpose, though.

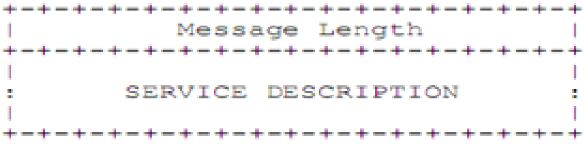

Figure 5: SOA Message Format

In the second layer of our proposed architecture, high-level services are defined. These services are made available to the end-users, which mean they can implement any kind of application the user might opt to use. In other words, they are the applications to which the traveller can access during the journey. Thus, every hardware and software components a vehicle must feature in order to provide such services are specified in this layer. As previously mentioned, we consider the vehicle will be equipped with an OBU.

Development Environment

The prototypical implementation herein presented is conceived on the basis of extending the MAS-T^{2}er Lab framework (*Laboratory for MAS-based Traffic and Transportation Engineering Research*). MAS-T^{2}er Lab is an integrated multi-agent system that applies a methodological approach based on the agent metaphor to devise, test and implement intelligent transportation solutions. A more detailed discussion on the main principles underlying such a framework, as well as a full description of its architecture and implementation can be found elsewhere (Rossetti et al. 2007; Ferreira et al. 2008).

For the present work, it is enough to mention that the MAS-T^{2}er Lab framework is conceived in terms of three main subsystems, which are multi-agent systems themselves, namely the *real world* (RW), the *virtual domain* (VD), and the *control strategies and management policies inductor* (CSMPI). The RW represents the real urban transportation system, in which physical entities such as vehicles, control systems, intelligent transport solutions and travellers in general cohabitate and interact. These components are replicated within the VD and modelled by means of agents. Software agents in the VD are expected to emulate the individual behaviour of their counterparts in the RW. Finally, the CSMPI if formed up by expert agents, both humans and virtual ones, which observe the synthetic population within the VD and are able to directly interfere upon their behaviour so as to test different control strategies and management policies.

The VD subsystem relies on a multi-paradigm traffic simulator and is the core of the MAS-T^{2}er Lab framework. The high-level architecture devised for its microscopic traffic model, depicted in Figure 6, is structured in layers where the most important component is undoubtedly the simulator engine controller (SEC).

The SEC component is responsible for managing all applications connected to the VD's simulation environment and their interaction with the urban network data structure. It also controls the simulation process, ensuring the necessary synchronisation of parts that can be run as distributed processes. All VD components and their interactions are detailed presented elsewhere (Ferreira 2008).

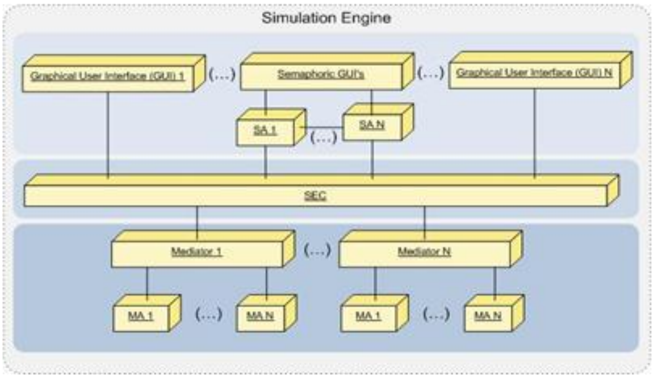

Figure 6: High-Level Architecture of VD Subsystem

In this work, the SEC component was extended with additional features especially designed to support modelling and simulating V2V communication networks, as well as their application in urban scenarios. The extended version of the SEC component is presented as an UML component diagram, as depicted in Figure 7, in which the V2V interface is emphasised in the red ellipse.

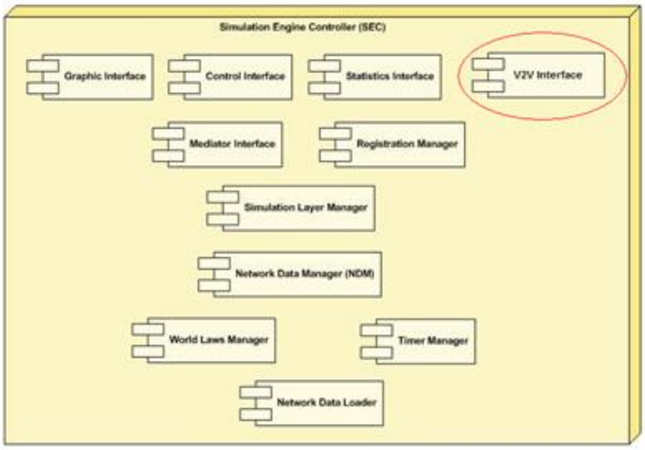

Figure 7: UML Diagram Illustrating the Physical Architecture of the SEC Component

The V2V interface is the SEC communication GUI that allows, whenever it is demanded, information on the vehicle-to-vehicle communication simulation process to be sent. At the current version of our prototype, such information includes global positioning of vehicles, their communication radius and the other vehicles within their range, as well as all messages that are sent and received by different vehicles of the synthetic population of the VD subsystem. The V2V interface also manages interaction with the end-users, periodically sending structured information on the simulation process and network status to the GUI modules attached to the VD simulation engine controller. Through this interface module, end-users can also interact with the model in simulation time, changing parameter values and other simulation variables.

The prototype has been implemented in the C++ object-oriented programming language, with the Qt framework that allows easy GUI development. All vehicle references are stored in a dynamic structure, in which each car can be identified unequivocally by their IP address, for instance. In this structure we also keep the global position of each vehicle, as well as other characteristics such as the communication range and messages log. As mentioned before, end-users can interact with the model in simulation time through dialog boxes, through which a list of all cars currently using the network is rendered. From this list it is possible to access each vehicle’s current status. A circle is drawn for the selected vehicle by double clicking its ID, which represents its communication range. Through a similar mechanism, the user can send messages on behalf of any vehicle travelling throughout the network. They can be sent either through unicast or multicast protocols, with vehicles recursively relaying the messages to their destinations. Addressees are identified through their global positioning and vehicles receiving sent messages are also emphasised in the graphical interface (a circle representing their communication range is drawn as well). Users can distinguish between different messages sent in different ways, namely through the messages log in the vehicles list or through differently coloured circles surrounding cars in the graphical visualisation of the network.

Figure 8 depicts a screenshot of the simulator GUI, with the vehicles list and the network visualisation windows. Different coloured circles represent different messages propagation throughout the network. In addition, the simulation environment also features graph-based representation of some performance measures, which can be dynamically updated in simulation time.

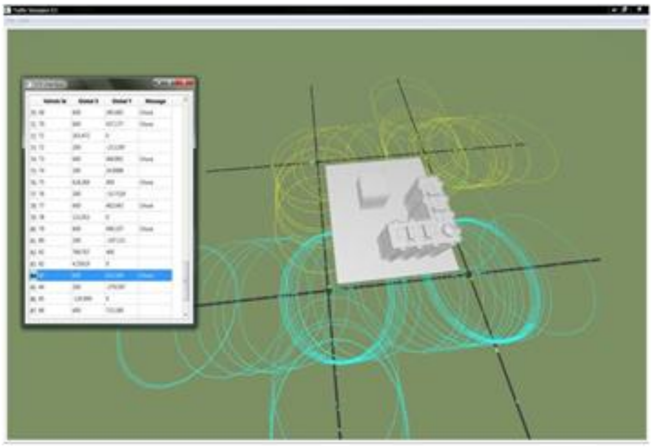

Figure 8: GUI for the V2V Communication Extension

An external tools interface is also available, allowing third-party tools to interact with the Simulator in run-time. Such an interface was implemented through socket-based streams via TCP. Any external application can then interact with our simulation model in run time and interact with the vehicles travelling throughout the network according to a protocol designed for this specific purpose. For instance, different graphical interfaces can be attached to the simulator, rendering alternative visualisations or graphs to ease performance assessment. Also, other tools can be implemented to improve the services available via the V2V communication network.

EXPERIMENTAL FRAMEWORK

Scenarios Set-up

The experimental framework carried out was based on a simple network, as depicted in Figure 8. Despite its simplicity, results showed promising potentials of the proposed approach. The network was coded in the XML format supported in the simulation engine of MAS-T^{2}er Lab, as defined in (Ferreira 2008).

Three different traffic flow configurations were set up, which allowed us to test the system behaviour under different flow regimes. The effects of these flow regimes on the V2V communication performance were analysed then.

First scenario: a low traffic flow regime was implemented in this scenario, in which source nodes were set to yield 100 to 250 vehicle/hour traffic flows, representing a free-flow regime.

Second scenario: this scenario is intended to represent an average traffic flow regime. In this scenario, source nodes were set to yield 250 to 850 vehicle/hour traffic flows, allowing us to analysis V2V performance under average conditions.

Third scenario: in the last scenario, source nodes were set to yield 850 to 1300 vehicles/hour traffic flows, representing traffic under saturation conditions. Such a scenario is quite common in must urban areas, during peak hours, for instance, or during the occurrence of some incidents strangulating road capacity, such as accidents.

Each simulation run was set to last for 10 minutes’ time, corresponding to 30 minutes in real time. The time scale can be easily modified, though. To test the communication interaction between vehicles, a car approaching an intersection sends a message (this is done with a 2 minutes’ frequency). Intersections are selected sequentially, following a clockwise order. Allowing cars to send messages at intersections is required as an attempt at maximizing the neighbouring cars within the range of the sender’s wireless range. Indeed, as intersections are spots of converging traffic streams, then it is very likely the number of cars at intersections, especially in the second and third scenarios, will be considerable high.

Results Discussion

Preliminary results are plotted in the graph of Figure 9. The graph shows how much of the network is covered by the V2V communication mechanism over time. The results are very interesting and demonstrate the ability of the implemented prototype to cope with both communication and vehicular traffic simulation. As we expected, the results of different scenario set-ups suggest different behaviours for the system under varying traffic conditions.

In the low traffic flow scenario, message dissemination through the V2V infrastructure is quite poor, basically due to the discontinued coverage of the network. Only a small part of the network, *circa* 27%, can be affected by the message propagation. In such circumstances, it is possible that even none of the vehicles will receive the message sent.

Scenarios in which traffic flow follows an average flow regime, such as in the second experimental set-up above, a larger part of the network can be easily covered. In our case, about two thirds of the network was covered in the most appropriate circumstances, meaning the increasing number of cars propitiated a better coverage and improved communication.

Only in the third scenario it was possible to achieve a full coverage of the network, which is quite expectable too. Indeed, in nearly saturated traffic conditions, network links tend to work in their full capacity, meaning the density is very high and vehicle headways tend to the average vehicle unit. In these circumstances, it is most probable that neighbouring vehicles will be within the range of other vehicles' wireless sensors, improving the connectivity of the network.

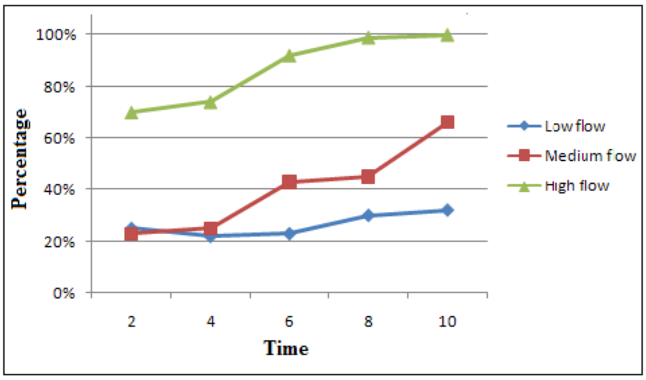

Figure 9: Comparison of the Three Different Scenarios, Relating Simulation Time and the Percentage of Network Covered by the V2V Communication

In all simulation scenarios, nonetheless, coverage increases as time evolves, which can be associated to the small network used and to the semaphore control at all junctions. Indeed, controlled intersections tend to group vehicles together at red lights, especially if traffic flow is higher than the saturation flows of each phase of the control plan. In such a situation, intersections will never be empty improving the connectivity of the V2V communication network.

The three different scenario set-ups were inspired in a typical daily flow profile of an urban network, in which the first scenario might represent a free flow regime, e.g. at high night and dawn, the second scenario might represent the average situation of off-peak hours during the day, whereas the third one might represent the morning and afternoon peak hours. Nonetheless, the first scenario might also be associated to rural areas, whereas the third scenarios might well be representative of a bottleneck caused by an incident, such as accident, for instance.

In either case, results corroborate the idea that V2V communication presents a great potential and is a promising technology of future urban transport, whereas implementing SOA beneath such communication infrastructure can contribute a great deal for a better transport service and quality of life in urban areas.

CONCLUSIONS

In this work, a V2V architecture based on the concept of services was specified. A layered architecture based on different levels of abstractions of services providing SOA in V2V networks was specified, as well as was a prototype implemented for testing and evaluating the approach proposed. The proposed architecture is structured into two main levels, namely the network services level and the consumer/service higher level. The former is responsible for building the network topology and for discovering and exchanging services among vehicles. The latter consists of the architecture to support service distribution, coined high-level services, to serve consumers. For the first level, we have implemented a SOA-based feature that allows service discovering within the routing protocol of VANET networks. This resulted in the specification of a dynamic and adaptable routing structure compliant with the abstract nature inherent in any SOA-based applications. The SOA features are rather decentralised in the first level, meaning there is no registry or central repository for services.

For the second level, on the other hand, we presented a modular and easily extensible architecture that allows services to be made available in vehicles. We have specified the vehicular architecture that underlies the implementation of on-board services, as well as the adequate means to request services from exogenous sources and other vehicles too.

Also, we have conceived and implemented a prototype of the proposed framework and extended the MAS-T^{2}er Lab original structure to support V2V communications within its microscopic simulation model. Besides V2V communications, such an extension allowed us to integrate external applications within the framework's traffic simulation engine.

In general terms, the developed prototype and experimental results demonstrated the feasibility of the proposed approach. The simulation environment resulted from the improvements implemented is an important asset for testing and experimenting new generation intelligent transportation systems, which will strongly rely on V2V communication capabilities. Further developments will include the implementation of more complex scenarios and adequate tools to support the assessment of a whole urban network. The introduction of other concepts, such as multi-agent systems are equally envisaged and will certainly contribute to the improvement of the way the decision-making mechanisms of drivers are affected under the influence of such a novel informational paradigm. The information visualization is another concern being addressed at the moment. Service utilisation and dissemination in large networks sometimes represents a tricky piece of information to visualise. Concepts of visual interactive modelling and simulation will be used to address such issues.

REFERENCES

Blum, J.J., Eskandarian, A., Hoffman, L.J., "Challenges of inter-vehicle ad hoc network", *IEEE Transactions on Intelligent Transportation Systems*, 2004. p.347-351.

Chen Z. D., Kung H., Vlah D., "Ad hoc relay wireless networks over moving vehicles on highways". In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing. 2001. p.247-250.

Choffnes D.R., Bustamante F.E., "An Integrated Mobility and Traffic Model for Vehicular Wireless Networks", 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET), 2005. p.65-78.

Ferreira P., "Specification and Implementation of an Artificial Transport System." Master's Dissertation. FEUP, Porto, 2008.

Ferreira P., Esteves E., Rossetti R., Oliveira E., "A Cooperative Simulation Framework for Traffic and Transportation Engineering," in CDVE2008: The 5th International Conference on Cooperative, Mallorca, Spain, 2008. p.89-97.

Halonen T., Ojaja T., "Cross-Layer Design for Providing Service Oriented Architecture in a Mobile Ad hoc Network." In Proceedings of the 5th international conference on Mobile and ubiquitous multimedia. Paper 11. 2006.

McGovern J. et al., "Java Web Services Architecture", 1st Ed, Elsevier Science and Technology Books, 2003.

Mello, H., Endler, M., "Identificação de Região de Congestionamento através de Comunicação Inter-veicular." In 24th Brazilian Simposyum on Computer Networks, 2006.

Rossetti, R.J.F., Oliveira, E.C., Bazzan, A.L.C., "Towards a specification of a framework for sustainable transportation analysis." In 13th Portuguese Conference on Artificial Intelligence, Guimarães, Portugal, 2007.

Saha A., Johnson D., "Modelling Mobility for Vehicular Ad-hoc Networks", ACM International Workshop on Vehicular Ad Hoc Networks (VANET), 2004. p.91-92.