Title	Prototyping telematic services in a wireless vehicular mesh network environment
Authors(s)	Olariu, Cristian, Hava, Adriana, Szczechowiak, Piotr, Ghamri-Doudane, Yacine
Publication date	2012-04-01
Publication information	Olariu, Cristian, Adriana Hava, Piotr Szczechowiak, and Yacine Ghamri-Doudane. "Prototyping Telematic Services in a Wireless Vehicular Mesh Network Environment." IEEE, 2012.
Conference details	2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Paris, France, 1 April 2012
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7521
Publisher's statement	© © 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/WCNCW.2012.6215525

Prototyping Telematic Services in a Wireless Vehicular Mesh Network Environment

Cristian Olariu*,
Adriana Hava*
*University College Dublin,
School of Computer Science and Information,
Belfield, Dublin 4, Ireland
cristian.olariu@ucd.ie,
adriana.hava@ucdconnect.ie

Piotr Szczechowiak‡,
Yacine Ghamri-Doudane**§
‡Telcordia Poland, ul. Umultowska 85, 61-614 Poznań, Poland
§ENSIIE, 1 Square de la résistance, 91025 Evry CEDEX, France
**Université Paris-Est, LIGM Lab, 75420 Champs sur Marne
pszczech@telcordia.com
ghamri@ensiie.fr

***Abstract*—Next generation telematic services are expected to play a key role in future automotive applications. In order to achieve strong integration between the services and the underlying network infrastructure there is a need for both simulation and emulation of the entire system. This paper presents a combined simulation and emulation approach for telematic services prototyping in an emulated wireless vehicular mesh networking environment. The ns-3 wireless mesh model, SUMO vehicular mobility model and different telematic services are integrated to demonstrate high scalability and flexibility of the proposed approach.**

I. Introduction

With the advances in car automation there is a growing demand for networked telematic services to facilitate navigation, improve car safety and provide infotainment data to mobile users. The dynamic nature of vehicular networks creates also an opportunity to provide location-based services. These next-generation telematic services are expected to be an integral part of tomorrows end-user communication patterns, ensuring continuity of services usually available to end-users in home and other public places.

The current offer of telematic services for automotive users is very limited, but it is expected to expand significantly in the near future. Users driving around metropolitan areas will be able to receive safety alerts and traffic updates from local authorities while downloading infotainment data (e.g. news, music, video) from local and remote servers. The information contained in the announcements will differ based on the location of users and their preferences. This vision of telematic services offering over a wireless mesh network is presented in Figure 1.

An essential point that would contribute to the success of telematic services in such environment lies in a strong service integration and good coupling between the services and the underlying network infrastructure. This integration can be achieved by appropriate design of the services and extensive simulation and emulation of the entire system. In this paper we present the evaluation results of a system, which enables an integrated telematic services offering in a metropolitan area using a wireless mesh network as the backhaul. The main goal

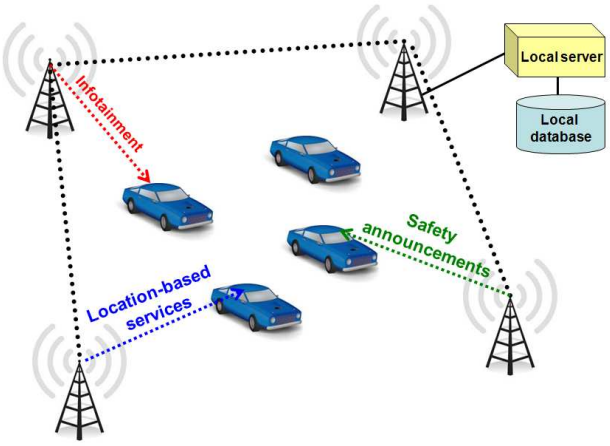

Figure 1. The vision of telematic services offering over a wireless mesh network

of this work is to design a joint emulation/simulation framework, which will serve as a platform for prototyping different service distribution protocols, mobility support mechanisms and QoS solutions. The emulation aspect of the framework is the most important element of the system, which will enable testing in close to real conditions.

Current wireless mesh networking technologies can offer high bandwidth communication links for mobile users without requiring many interconnections into a wired infrastructure. The deployments of such networks can be also seen as a low cost alternative to cellular systems especially in dense urban environments. Mesh networks can be easily deployed by different stake holders (local authorities, network operators, ISPs etc.) to deliver not only Internet access but also other services such as safety announcements and traffic updates. One of the main advantages of wireless mesh networks is the ability to manage the services in a centralized or distributed manner, depending on the nature of the application. This gives a great flexibility in designing the service architecture and the mesh network topology. The joint emulation/simulation framework

presented in this paper will serve as the main tool in the design and prototyping process of such a system.

The rest of the paper is organized as follows: Section II presents the related work. Section III describes the general approach to joint emulation/simulation of distributed systems. Section IV discusses our emulation setup. In Section V we present the evaluation results of three telematic services tested in our system. Finally we conclude in Section VI with a brief description of future work.

II. Related Work

The majority of research work in wireless networking is performed through simulations. Emulation is not that frequently used due to larger complexity and a difficult development environment. In the literature we can find papers which investigate the accuracy of different simulation/emulation programs.

The work presented in [1] is an experimental evaluation of the wireless model implemented in ns-2, which compares the results of a real system, an emulated network and a simulator. It is proven that a good parametrization of the simulation model can give a good approximation of a real network behaviour. Emulation takes also into consideration the delays introduced by the operating system and the hardware itself. Therefore system emulation will provide more accurate results than simulation. In our work the wireless mesh model is implemented using the ns-3 simulator while the mobile clients and their servers are emulated using real computers.

Judd *et al.* [2] proposed a hardware-based emulator, which mimics the characteristics of wireless links and assures realistic and repeatable experiments. The framework described in [2] provides real-time modelling of the wireless communication channel between all pairs of nodes that are attached to the emulator. In order to reduce the effects of the environment the authors use cables attached to the antenna port of the wireless cards instead of wireless links.

The work described in [3] extends the study presented in [2]. It enables the analysis of higher layers performance in real networks and facilitates the development and evaluation of different wireless networking protocols.

Zhou *et al.* in [4] present an efficient emulation framework called TWINE, which combines emulation with repeatability of simulations and physical components, in order to achieve scalability for wireless network systems. It emulates the wireless channel, the MAC layer and the physical layer but only supports the 802.11b WLAN standard, which limits its applicability. The three papers presented above focus mainly on designing an emulator which imitates wireless links characteristics. In our emulation setup the main goal is on prototyping different telematic services and performance evaluation of the wireless mesh and the networking protocols working on top of it.

MobiNet, presented in [5], is a scalable emulation infrastructure for wireless networks. It consists of core nodes, used to emulate hop-by-hop network characteristics, and edge nodes, which generate the network load and forward the packets to the core nodes. The presented framework enables mobile, ad-hoc and wireless emulation and allows researchers to experiment with a variety of MAC protocols and routing protocols, which are not usually available in a test-bed environment.

In [6] Staub *et al.* presents VirtualMesh, which is an emulation for wireless mesh networks using the OMNeT++ simulator. It combines computer simulation, network emulation and virtualization into a single platform for wireless mesh network experiments. This framework is able to virtualize an entire test-bed by emulating the wireless network and running the wireless nodes as virtual hosts. Real network traffic is captured at given nodes using a virtual interface and then redirected to the OMNet++ simulator. Staub's *et al.* work is closely related to ours but their analysis does not show how the system behaves in the presence of mobile nodes. Our work investigates how node mobility influences protocols performance and the entire network behaviour.

The joint simulation/emulation framework presented in this paper relies on the state-of-the-art implementation performed in the ns-3 simulator. This approach allows us to use advanced functionality, which is not available in the emulation systems, which are based on OMNet++ or the ns-2 simulator. In addition, our system can simulate different mobility patterns, which can realistically emulate car movements in metropolitan areas. The presented framework is a perfect simulation/emulation tool for prototyping different telematic services and network protocols in a wireless network environment.

III. System Architecture

The concept of emulation can be described as a process, in which any computer platform or a program mimics the behaviour of a different computer platform or a program. It enables users to test their implementations before deploying them into the real environment. In the emulation process it is possible to attach a real network device to a simulated environment. Emulation is used when a part of the system is not available and needs to be *imitated* in terms of functionality. Software emulation is the most common approach, which provides emulation of software or hardware parts of a system.

Emulation is very important especially when performing research studies in the area of wireless networks. The importance of emulation in computer networks has been recently highlighted through the addition of emulation capabilities to the well known network simulators such as ns-2, ns-3 and Qualnet.

The emulated components of a computer network have to provide the same functionality as their real counterparts. The interfaces between different modules must be implemented in an appropriate way, to allow seamless operation. The emulation needs to be done in such a way that any real component will not be able to notice the differences in functionality.

Figure 2 presents the general concept of a combined simulation/emulation approach for distributed computing systems. The part of the network which does not have any connection point with the real components, is called the *simulation core*. A typical emulated component is the Network Interface Card

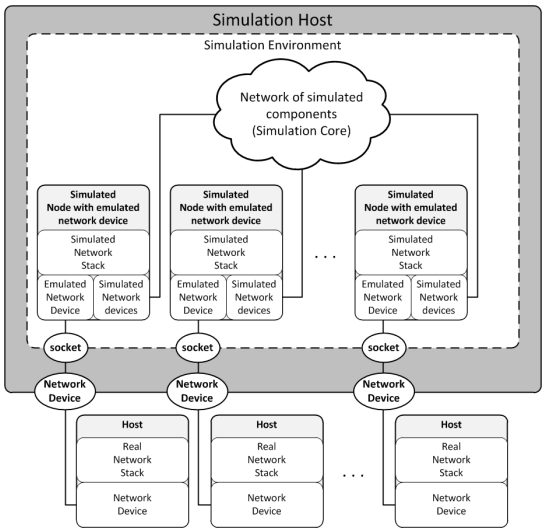

Figure 2. The diagram of the network emulation concept

(NIC) installed on a particular node. We note that usually the network device is emulated, i.e. a software component that directly links with a real network device of the simulation host (real computer hosting the simulation). The rest of the network protocol stack is provided by the simulation environment. In this manner the simulated node becomes a *virtual host*.

The interaction between the simulation environment and the machine, which hosts the simulation (simulation host) is done through sockets. The simulation host connects these sockets to the actual networking devices. The network devices on the simulation host grant entry points into the simulation for real hosts. This enables real traffic to flow between real hosts while being backhauled through a complex simulated topology.

Typically the simulation host needs to provide as many real network devices as emulated devices are in the simulation. In this way, real packets from real hosts can traverse the simulation core and reach nodes inside the simulation core or other real hosts. Thus, a large part of the network can be simulated while some real hosts can be virtualized.

The necessity to provide a real network device for each emulated node could become a challenge in case of large systems. However many interfaces can be set in a *promiscuous* mode, in which packets are not interpreted by the network stack of the real host but are directly forwarded to the sockets. In this way, one real interface can be shared by multiple emulated nodes which will be directly connected to the sockets. The combined traffic streams from the emulated nodes can be split into individual streams by a *network splitter*. For packet switched networks, a network switch can play the role of a splitter. One problem that might appear in such a system is when the combined data streams of emulated nodes reach the data rate limit of the real network interface card. This problem needs to be resolved at the design phase of the emulation system.

IV. Framework Description

This section presents the main components of the emulation setup. The first part describes the main telematic services, which will be evaluated through emulation. The second part describes the simulation environment used in our research experiments. The last part presents the architecture of the emulated system.

A. Telematic Services description

Future telematic services for automotive users aim at improving safety on the roads through proactive, interactive, and regular announcements. They ease the access to points of interest by combining location-based services with the navigation system. They can also provide interfaces for local businesses to announce special events and offers to passing by cars. Dedicated services might be deployed to improve the driving experience around metropolitan areas through constant location-based traffic updates. On top of the above mentioned services entertainment and infotainment services can be delivered to the on-board unit in each car. In general we can classify telematic services into three broad groups:

- Safety applications;
- Location Based Services;
- Infotainment services.

The first group of applications is very important, because it delivers safety related information to drivers (road conditions, accidents etc.). These announcements are usually not that frequent but require guaranteed data delivery. This class of services might include also emergency VoIP calls in case of critical situations. The second group includes short periodical announcements and advertising information based on users location and their preferences. Traffic update information are also broadcasted depending on the location of the users. The last group of services has lower priority than the previous applications, but delivers popular services like music and video streaming. In order to evaluate the performance of telematic services over a wireless mesh network we decided to test one application from each group: emergency VoIP calls, video streaming, and location-based announcements.

B. Simulation model description

In order to simulate the wireless mesh network environment we used the ns-3 simulator. This tool is a discreet-event network simulator, commonly used for research purposes. Our configuration is a grid topology consisting of nine mesh nodes separated by a distance of 100 meters. The gateway is selected as the top-left node of the grid.

In our setup, the clients are represented by cars and are spread across the mesh network. The most important characteristic is that the clients have different mobility patterns and are roaming inside the mesh network along different routes. The Nakagami fast fading loss model was used as the propagation model. This is the most popular model for urban scenarios, which are the main target in this work.

Each mesh node is equipped with two 802.11g interfaces for communication between the mesh nodes and one 802.11a

interface for communication with the cars. The choice of 802.11g for the communication inside the mesh network is justified by its large communication range, which is larger than the 802.11a's one. The increased number of non-overlapping channels that are available in 802.11a protocol justifies our selection of the wireless communication standard for the car-client interface. The 802.11g channels were allocated using a spread channel policy, where two non-overlapping 20MHz frequency channels are assigned to different mesh interfaces. Two interfaces are also used to improve network capacity and reliability.

The 802.11s standard implemented in ns-3 is used as the mesh network routing protocol. This employs the HWMP (Hybrid Wireless Mesh Protocol) [7] as the MAC-layer routing protocol. The 802.11s standard is an extension of 802.11 for arbitrary multi-hop topologies, where each mesh node operates as a link-layer router and cooperates with all the other mesh nodes in the process of frame forwarding. HWMP is based on the AODV protocol [8], but it works at the MAC layer for efficient path selection. HWMP can work in two modes: proactive and reactive. In the reactive mode the path discovery starts when the source wants to transmit data to an unknown destination. In the proactive mode a single mesh node is configured as a root and if the route to a destination is unknown the data is sent to the root node, which is responsible for sending it to the destination node. In all experiments we used the proactive mode of HWMP to limit the delay in the route discovery process.

We have also developed a method which uses three different tools to generate mobility patterns for the cars: SUMO (Simulation of Urban MObility) – an open-source road traffic simulation package designed to handle large road networks, OpeenStreetMap – used to obtain the XML file containing the map information from a specified area on the map, and TraNS (Traffic and Network Simulation Environment), which links SUMO and the ns-2 network simulator. TraNS allows importing an XML file and to adjust different parameters such as number of cars, delay of cars arrival and simulation time. TraNS is capable of generating mobility traces for ns-2, which can be imported into ns-3 using the Ns2MobilityHelper class.

C. Emulation system description

This subsection describes which computing and networking equipment is needed to build an emulated Wireless Mesh Network environment. In our research experiments we used three different computers. Each computer hosted one important part of the end-to-end telematic service delivery system. One computer hosted the server-side (one or more server instances can be hosted on a single computer by using virtualization software - e.g. VirtualBox), the second computer hosted the simulated network topology used to enable emulation, and the third computer hosted user applications.

To enable connectivity between each of the three computers, we used a network switch. All four above presented components, were interconnected via Ethernet cables as depicted in Figure 3. The network interface card on the PC1 was

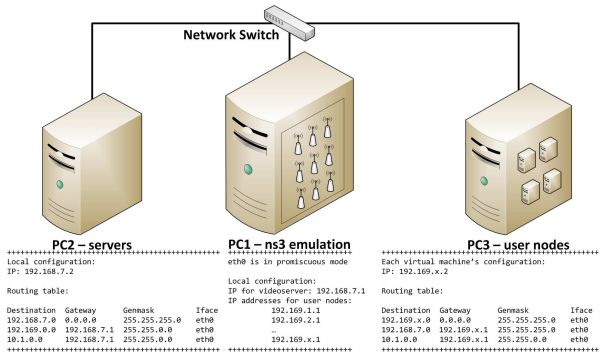

Figure 3. Emulation Setup

set to the promiscuous mode. Figure 3 presents also the IP routing tables for all the computers in the emulation setup. For example purpose, let us consider one instance of a server which supplies multiple clients with services. The server will be associated with a sub-network IP address of *192.168.7.0/24* while the clients will belong to multiple subnets defined by the IP address of *192.169.0.0/16*. For the proper function of the entire emulation it is essential to setup the routing tables on each computer.

V. Performance Evaluation

This section gives details about the emulation/simulation scenario used to obtain the performance evaluation results by running different telematic services through a simulated wireless mesh network.

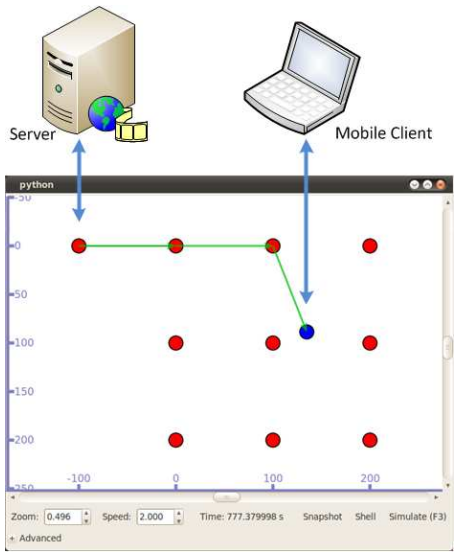

Figure 4. Visualization of the simulation and the emulated system components

Figure 4 presents the mapping between the real nodes and the virtual mesh nodes simulated inside the emulation. In our simulation scenario a mobile client roams in an area of a

3x3 mesh grid (as depicted in Figure 4). For the purpose of performance evaluation of telematic services we use the mobility model, which is already implemented in ns-3. The wireless interface of the mobile node is wireless connected to an Access Point of one of the mesh nodes. A hand-over mechanism takes care of the connectivity issues related to the users' mobility. On the mobile node, a point-to-point interface is set in emulation mode. The emulated interface is connected to the simulation host's Ethernet device and through IP layer settings it is linked to the client represented by a laptop machine (the blue dot in Figure 4).

At the top-left corner of the grid there is the mesh gateway which is connected through a wired link to the simulated node. This node has an emulated point-to-point device connected to a real machine, which hosts different server applications. Table I presents the main emulation setup parameters used in our experiments.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Car Speed	30 Km/h
Simulation Time	200 seconds
Routing Protocol	HWMP
Distance between the mesh nodes	100 m
No. of mesh interfaces	2
Mesh topology	grid 3x3
Car to mesh communication	802.11a
Mesh to mesh communication	802.11g
Mobility model	Random Walk Mobility Model
Radio propagation model	Nakagami fast fading loss model

The *iperf* application was used on the client and server side to generate the traffic patterns, which correspond to different telematic services. In order to model the *location-based announcements* we chose a UDP application sending packets with a payload of 200 Bytes and a data-rate of 1.6 kbps. From Figure 5 it can be seen that the jitter is maintained at a very low level which is sufficient for this type of service. The few spikes in the packet loss plot are due to the mobility of the car, which triggers hand-overs. Based on these results it is clear that the location-based announcement service can be supported in this kind of a network if a low percentage of packet loss is allowed. In order to improve the quality of service there is a need to design more efficient mobility support protocols.

The emergency VoIP calls were modelled as an UDP application sending packet of 100 Bytes (payload size) with a data-rate of 40 kbps. These values were chosen as a mean of the values used by current available voice codecs (e.g. AMR, G.729, G.711, etc.).

As depicted in Figure 6, the jitter is contained in the recommended limits. The tolerance for VoIP jitter is in the range of 20-30ms [9]. The losses occur in short bursts but more often than the ones that appear for the announcement services. This is due to the fact that packets are sent more often for the VoIP application. These results show that even in the case of time-sensitive traffic such as VoIP, the emulated environment

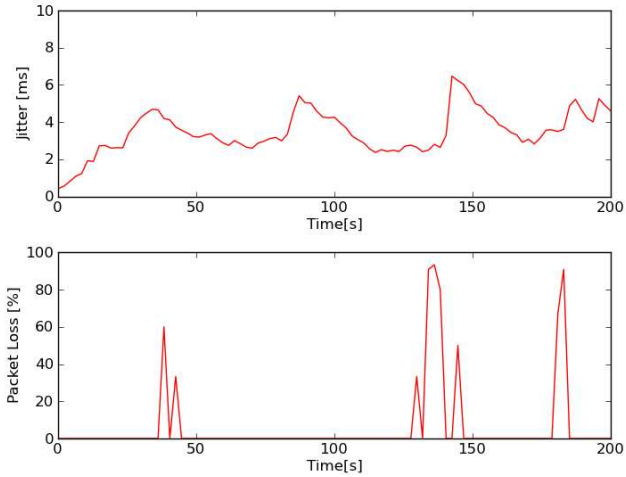

Figure 5. Location based announcements

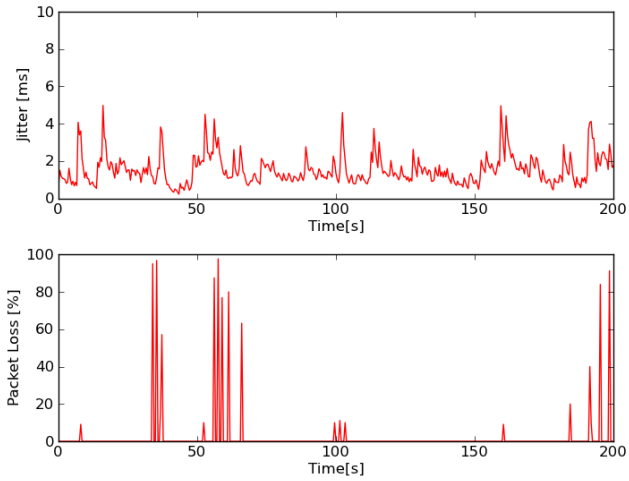

Figure 6. Emergency VoIP call performance in a wireless mesh network

performs as expected. The simulated wireless mesh network topology was consistent in supporting VoIP throughout a long period of time, validating this framework as reliable for real-time sensitive traffic.

In order to model the third group of services we used video traffic, which was generated through UDP packets of 1500 Bytes. This value represents the Maximum Transmission Unit (MTU) over a wireless medium. As the on-board-unit display is of small resolution, we considered fair to send video packets at a medium quality data-rate of 750 kbps.

The jitter for the video traffic (presented in Figure 7) is higher than for the previous services, but still in the acceptable range. The packet losses occur from the same reasons presented for the VoIP service, that is hand-overs due to user mobility. Video streaming is more demanding in terms of occupied bandwidth than a VoIP application but offers better quality of experience in similar network conditions. Our emulation environment shows that degradation of the quality

metrics are present but even in this case the delivery of the services is reliable and performs in a similar way to real conditions.

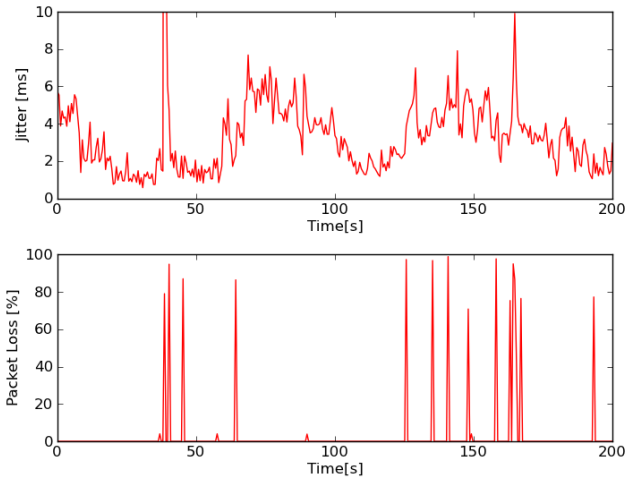

Figure 7. Video streaming performance in a wireless mesh network

VI. Conclusion

In this work we have presented an emulation framework which uses the latest advancements in the state-of-the-art of network simulators. This framework enables extensive evaluation of new services and networking protocols, which is difficult to perform in real tesbeds. The framework is easy to implement and has a low deployment cost. We have also described all the steps on how to build such emulation systems.

We have tested this framework in the context of telematic services and assessed its performance for several types of applications. The emulated network topology was a wireless mesh network with vehicular mobility support. The ns-3 network simulator was used to create a wireless vehicular mesh networking environment interfacing with real computers hosting the telematic services. The services were classified into three major categories: safety applications, location based services, and infotainment services. Each type of service was modelled with *iperf* and validated through tests using real network traffic.

With this work we have developed a joint simulation/emulation framework, which is a perfect platform for prototyping different telematic services and testing how they influence the underlying network infrastructure. This will allow strong integration of services with the network and fine-tuning of different network protocols and their parameters.

In the future work we plan to test different mobility models, which will emulate realistic car movements in metropolitan areas. We will also replace the synthetic traffic generated by *iperf* with real traffic patterns from our telematic services. We will perform large scale simulations with large number of mobile clients and mesh node deployments, which take into account the characteristics of a metropolitan area. Finally in order to validate our results we plan to compare them with the measurements acquired from a real world testbed.

Acknowledgement

This work is supported by the European Union through the Marie Curie IAPP program under the grant agreement no. 230684: ”CarMesh: Ubiquitous Wireless Mesh Networks for Next-Generation Personal Digital Automotive Services”.

References

[1] S. Ivanov, A. Herms, and G. Lukas, “Experimental validation of the ns-2 wireless model using simulation, emulation, and real network,” *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference*, pp. 1 –12, 26 2007-march 2 2007.

[2] G. Judd and P. Steenkiste, “Repeatable and realistic wireless experimentation through physical emulation,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 63–68, January 2004. [Online]. Available: http://doi.acm.org/10.1145/972374.972386

[3] ——, “Using emulation to understand and improve wireless networks and applications,” in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI’05. Berkeley, CA, USA: USENIX Association, 2005, pp. 203–216. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251218

[4] J. Zhou, Z. Ji, and R. Bagrodia, “Twine: A hybrid emulation testbed for wireless networks and applications,” in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –13.

[5] P. Mahadevan, A. Rodriguez, D. Becker, and A. Vahdat, “Mobinet: a scalable emulation infrastructure for ad hoc and wireless networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 10, pp. 26–37, April 2006. [Online]. Available: http://doi.acm.org/10.1145/1137975.1137979

[6] T. Staub, R. Gantenbein, and T. Braun, “Virtualmesh: an emulation framework for wireless mesh networks in omnet++,” in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools ’09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 64:1–64:8. [Online]. Available: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5563

[7] J. Camp and E. Knightly, “The IEEE 802.11 s extended service set mesh networking standard,” *Communications Magazine, IEEE*, vol. 46, no. 8, pp. 120–126, 2008.

[8] C. Perkins, E. Belding-Royer, S. Das *et al.*, “RFC 3561-ad hoc on-demand distance vector (AODV) routing,” *IETF, July*, 2003.

[9] ITU-T G.114, “One-way Transmission Time,” http://www.itu.int/itu-t/recommendations/index.aspx?ser=G.