Vehicular Testbeds - Model Validation before Large Scale Deployment

Mario Gerla, Jui-Ting Weng, Eugenio Giordano, Giovanni Pau
Department of Computer Science, Univeristy of California, Los Angeles
Email: {gerla,jtweng,giordano,gpau}@cs.ucla.edu

***Abstract*—Vehicular communications are becoming a reality due to the investments by stakeholders like car manufacturers and Public Transport Authorities. The building blocks of the "Vehicle Grid" (radios, Access Points, spectrum, standards, etc.) are coming into place enabling a broad gamut of applications that range from navigation safety, intelligent transport, entertainment and urban sensing. Vehicular protocols and applications, however, must be carefully tested before deployment in the urban grid and introduction to the users. This validation must be carried out progressively in simulation, emulation and small scale testbed environments. In this paper we discuss the important role of the vehicular testbed in validating models and protocols before deployment in large scale scenarios. We illustrate the concept using two case studies that were carried out in the UCLA open vehicular testbed.**

***Index Terms*—vehicular testbed, VANET, AODV, OLSR, virtualization, emulation, radio propagation**

I. INTRODUCTION

Vehicular communications have been receiving increasing attention over the last ten years as a viable mean to augment road safety and travel efficiency [1]–[4]. The field has attracted consistent investments from auto manufacturers and public transport authorities, further stimulating academic research. We have reached now a situation where the essential building blocks of vehicular networks (On Board Radios, Road Side Access Points, Reserved 5.9 Ghz spectrum and dedicated communication standards) are available thus opening interesting opportunities for a wealth of car-to-car applications [2].

Security-oriented applications are still the top priority for auto industry and transport authorities, and recent testbed experiments have proved the effectiveness of vehicular communications in preventing intersection crashes [6]. The availability of the technology is also stimulating interest in new applications beyond mere safety. For example, automatic and efficient traffic management (using "Intelligent Transport" techniques) can exploit vehicular communications to reduce traffic congestion and keep chemical pollution under control. One can envision a comprehensive urban traffic planning system that receives inputs from vehicles (e.g., route plans, destinations, sensor readings, positions, driver's preferences, etc), processes such information to generate an "urban routing" plan, and implements the plan through the careful control of traffic lights (say green waves) and through on board navigator instructions (with the assistance of Navigator Servers).

Another family of applications is enabled by vehicles acting as sensor platforms to collect information. On board sensors such as GPS, cameras, microphones, pollution sensors, humidity, temperature, etc can be used to build a distributed and enriched awareness of the vehicular environment, which, in turn, can boost the creation of "environment-aware" applications. Among such applications, vehicular surveillance can assist in crime investigation and suspicious activities monitoring, and; repositories on wheels can store location relevant commercial, entertainment and cultural information. Simulation captures the main characteristics of the VANET protocol and applications and allows them to interact with different environment settings. It provides an efficient way to study the feasibility and scalability of the application. In this paper, we define the Emulator to be a testing platform where the test nodes are implemented with real code (such as in a Cloud for instance), and; the wireless channels are simulated in software. Each node in the emulation runs real VANET protocol and application implementation. When packets are transmitted to another emulation node, it is handled by another layer of software to create artificial wireless characteristics. Therefore, the mobility pattern and wireless channel condition are provided through software. The testbed features exclusively real hardware and software components.

Today, applications such as urban traffic management or popular content distribution to vehicles are modeled and tested mainly via simulation. This required a leap of faith by the stakeholders (transport authority, providers, manufactures) to go ahead with large scale deployment. In fact, this is one of the main obstacles to VANET applications introduction. In our vision, the leap of faith gap must be bridged with a progression of validation steps that ease the acceptance of this technology. In this progression, the VANET testbed will play two critical roles. First, it will be used to validate the accuracy of the MODELS to be used in simulation and emulation experiments (for example, propagation, channel access, spectrum sensing, etc). This preliminary validation will make the models more credible. Secondly, the testbed will be used to validate the correctness of protocols and applications in a real (albeit small scale) scenario.

This research was sponsered by NSF project CNS-1111971: "Collaborative research: closing the loop between traffic/pollution sensing and vehicle route control using traffic lights and navigators."

Manuscript received January 26, 2012; revised March 1, 2012; accepted March 25, 2012.

doi:10.4304/jcm.7.6.451-457

To illustrate this progressive validation, consider a traffic management application that reroutes vehicles dynamically based on measured congestion. This application may require car-to-car and car-to-infrastructure communications; GPS positioning and; car-to-car license plate reading, and; driver compliance, say. The feasibility of these operations must first be checked in a small scale testbed. The basic protocols are tested for correctness. Approximate performance models for the key operations (eg, beacon delays, license plate read accuracy, propagation properties, etc) are then derived and experimentally validated. At this stage, even behavioral models (eg, driver reaction times, or compliance with navigator instructions) are identified and validated V over a small but representative population. The next step is the implementation of the protocol code in the Emulator. Using small scale testbed experiments (up to 20 vehicles) as test cases, we validate the correctness of emulator. Once the emulator is validated, it scales up in size and generates credible results for up to say thousands of vehicles. Recall that in the UCLA emulator each vehicle runs real code on a virtual machine in a Cloud Server, say. The emulator however will not easily scale to millions of vehicles - the cost and overhead would be prohibitive. This is where simulation comes to help. In fact, with the proper abstractions (say, replacing discrete events like cars with continuous flows). The key issue here is that we can first VALIDATE the scalable simulator with a 1000 node emulator. This validation will make the one million vehicle simulator much more credible than if it had been developed merely from abstract, untested models (of traffic, drivers, communications, etc). The progressive experimental validation that leverages the Testbed will mitigate the leap of faith anxiety (of the stakeholders) and will pave the way to deployment through a sequence of satisfactory simulation, emulation and small scale testbed experiments.

Following this vision, in this paper, we illustrate the dual role of the VANET testbed in two case studies: validation of urban propagation models, and; validation of a protocol comparison methodology based on the parallel use of virtual machine . For these case studies, we will use the Campus testbed architecture, called C-VeT [5], deployed at UCLA. C-Vet is based on the integration of a vehicular ad hoc component and a flexible wireless mesh infrastructure - MobiMESH [7]. C-VeT differs from traditional Campus testbeds in that it relies on a dedicated Mesh infrastructure and on a fleet of Campus Facilities Vehicles endlessly circulating on Campus as they respond to service requests. These two features guarantee a well connected VANET topology that can adequately support a broad range of experiments.

The paper is organized as follows: in Section II we report on related vehicular testbed activities. Section III describes the main building blocks of the C-VeT architecture, whereas Section IV describes two C-VeT case studies. Section V concludes the paper.

II. Related Work

C-VeT is one of the few testbeds that provides both Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) connectivity. Most of the existing vehicular testbeds lack either V2V or V2I consistent connectivity. For example, CarTel at MIT [8] features a fleet of taxi cabs equipped with wireless interfaces that gather information and exploit open access points around the city to upload the data to a central data server. DOME at UMass [9] uses the same concept as CarTel but adopts Campus buses that form a sparse network with prescheduled mobility patterns. DOME buses connect to the internet via dedicated access points. These access points however are not connected by a wireless mesh. Moreover, the vehicles roam a large area and cannot maintain continuous peer to peer connectivity.

Orbit [10] is a project that combines an indoor radio grid emulator and an outdoor field trial network. It is available for use either via remote or on site access. As for mobility support, the outdoor testbed is grounded (ie, stationary), while the indoor emulator only supports virtual mobility. Of relevance to our case study to be introduced in Sect IV, D. Rastogi et al. present a comparison between AODV and OLSR, performed on the Orbit indoor testbed. Their results indicate that AODV performs better than OLSR in terms of stability. One limitation of Orbit is the rather simplistic radio propagation model that does not account for buildings and obstacles.

The Wireless Signal Propagation Emulator developed by CMU accurately models wireless signal propagation in a physical space [12]. The emulator takes the signals generated by wireless network cards through the antenna port, transform them by a set of FPGAs, and feed the signals back to the wireless card. The FPGAs transforms the signal base on data collected through small scale experiments and effects that occurs in real physical space, such as attenuation, multi-path fading,.... The emulator, however, has limitations in generating wireless condition between nodes with arbitrary motion patterns.

The CS Dept of the University of Uppsala has recently opened to the community the Ad hoc Protocol Evaluation Testbed (APE Testbed) [13]. APE is an encapsulated execution environment with tools for post test-run data analysis. It is like a small Linux package with ad hoc configuration and network traffic analysis tools. The package can be installed in either Windows or Linux environment to perform ad hoc experiments and display the result with GUI. Incidentally, Lundgred et al. used APE to evaluate the performance of AODV and OLSR with up to 37 nodes deployed along indoor hallways and the athletic field. Their results show that AODV outperforms OLSR in high mobility. To the best of our knowledge, APE has not been extended to VANET applications.

III. C-VeT Framework

C-VeT is an open platform that support Vehicular Network and Urban Sensing research and related applications. Following the pioneer work by Larry Peterson and Tom Anderson with Planet Lab [14], C-VeT is an "always on", fully virtualized and web-accessible test facility. It is a combined of varies testbed nodes, and a emulation platform which can directly inject testbed node configurations. The UCLA campus, with its 10 acres of urban development, reproduces many of the propagation and spectrum interference challenges typical of a small city. In particular the C-VeT architecture provides:

- A fully virtualized platform that runs both Linux based and Windows based operating system with full insulation among the guest virtual machines, and enables the users to re-design low level protocols such as, for instance, MAC protocols. This flexibility is critical in network centric experiments.
- A Campus Wide Mesh network developed using OPEN WRT and optimized for the integration and support of the Vehicular network. It provides backup connectivity in case of V2V connectivity disruption (quite common in small scale testbeds). It enables opportunistic, interactive as well as delay tolerant experiments that exploit the Infrastructure,
- 30 Facility Management vehicles equipped with the C-VeT hardware/software, providing an always-on platform to run experiments, collect traces and measurements. The facility Management vehicles perform both routine maintenance trips and on-demand interventions in response to emergencies resulting in a varied mobility pattern that well approximates real city traffic. The Facility vehicles can be complemented or replaced by private vehicles and drivers if experiments with specific motion patterns and driver participation are required.
- a large scale, virtualized emulator that will allow users to debug their algorithms and protocols on the same hardware as the actual C-VeT nodes but with an emulated network component developed with the Qualnet hybrid simulation mode.
- a robust web interface that manages users and deploys the experiments in a streamlined fashion. The Web server will provide the front-end for user-friendly services and tools enabling users to focus on research rather than testbed implementation. For example: services to set up the experiments and gather the data; APIs to low lever interfaces for hardware component virtualization; virtual MadWiFi layer for the support of Virtual Machines.
- an organized live database of mobility traces, sensed environmental data, road traffic information, Vehicle CanBus statistics, MAC layer statistics (through MAD WiFi) and Physical Layer statistics taken using a variety of radios (Cognitive Radios, MIMO etc). This data collection is made available to the research community in collaboration with existing trace collection programs and archives such as CRAWDAD [15]

The testbed was designed with a top-down approach. The system consists of a small number of relatively simple building blocks: *the C-VeT mobile node, the C-VeT mesh node, the C-VeT-Census platform, the Web based Control Center, and the Emulation platform*. We describe the first two components in more detail below.

Fig. 1. C-Vet Mobile Node

The C-VeT mobile node: (Figure 1) is an industrial strength Cappucino PC powered by an Intel Dual Core Duo processor at 2.5GhZ, 2GB of RAM, 320GB Disk. Hard drive and internal parts are rugged to sustain physical stress (i.e. large temperature fluctuations, vibrations, etc). The PC has 3 Wireless Interfaces: IEEE802.11a/b/g/n based on the Atheros AR9160 chipset; IEEE802.11p interface based on a Daimler-Benz customized chipset; a standard Bluetooth interface mostly for internal communications. WiMAX and 3G radio interfaces can also be configured

The C-VeT mesh node: is based on MobiMESH hardware. The C-VeT mesh nodes feature Open WRT OS and Atheros Chipset with MadWiFi support, thus easing up the integration with mobile nodes. The fixed infrastructure will be installed on UCLA roof tops aiming at full campus coverage and integration with the existing campus WiFi infrastructure. The mesh allows opportunistic Internet access from vehicles and provides a control channel to the vehicles. The Mesh network can be configured via web; e.g., customized routes can be set up by the network operator to perform particular experiments. This C-VeT integrated approach with infrastructure and vehicles broadens the experimental scenarios. The initial phase covers the south Campus creating an initial backbone of 6 mesh points.

IV. C-VET CASE STUDIES

A. Corner model Validation

In urban VANET scenarios, the prohibitive costs (and the privacy implications) of real field experiments with thousands of instrumented vehicles forces researchers and developers to fall back to simulation models and tools. Simulation fidelity is important in order to get meaningful results. In particular, radio propagation fidelity is essential for credible evaluation of vehicular network protocols and applications in urban scenarios. Ray tracing is the ideal technique to obtain accurate models, but it is too time consuming. In contrast, the propagation models implemented in most simulators are overly simplified (eg, two-ray model). To overcome this limitation we have imple-

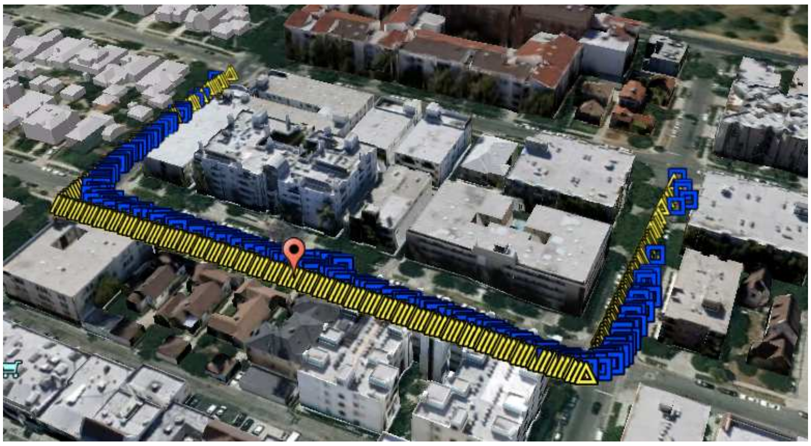


Fig. 2. Corner Evaluation

mented in the QualNET simulator a propagation model, CORNER, that accounts for "buildings and corners" by exploiting a set of analytic models reported in [16]. The path loss prediction formulae were extensively validated in [16]. However, for our applications, we had to validate the models not only for received power accuracy but also for the resulting topology connectivity. This had to be done within the context of the QualNET implementation [17].

Our "model validation" experiments were carried out using two cars equipped with a laptop with linux OS, a GPS receiver and a IEEE802.11b/g wireless card. The wireless card features an Atheros chipset allowing the use of the open source driver MadWiFi [18]. We performed experiments on our Campus Testbed to assess the connectivity around corners, using both fixed and mobile nodes. An important advantage of Campus testing is the ability to monitor (and avoid) interference from other APs, since the latter can cause incorrect power readings . The frequency of transmission was set to 10 packets per second. One car was revolving around a block and the other was fixed, in the middle of the block. The fixed car periodically sends out broadcast packets with geographic coordinates and the GPS timestamp. Figures 2, plotted on Google Earth [19], shows the set of locations where packets were received both for the field test (blue squares) and in simulation (yellow triangles). The simulated connectivity is remarkably similar to the real one. In addition we can see that in the simulation the number of received packets is much higher than in the field experiments. This is a consequence of the surrounding environmental interference that exists in the experiments but cannot be reproduced in simulation. Following testbed validation, CORNER has been uploaded in the QualNet simulator and the Virtual Emulator.

B. Parallel, Virtualized Routing Experiments

In this second case study we develop a methodology to compare routing protocols in a mobile VANET environment. The most straightforward approach is to run separate, sequential experiments with the two routing protocols A and B. However, the resulting comparison will generally be inconsistent due to the dynamic nature of VANET. By the time the one experiment terminates, the external conditions (interference, surrounding vehicle motion patterns, and radio propagation properties) may have changed and become unrepeatable.

Simulation provides a repetitive, consistent evaluation. However, simulators use oversimplified and often unrealistic models of physical characteristics (eg mobility/traffic patterns, propagation, and external interference). Thus, accuracy suffers. A third solution is to collect traffic/mobility traces during the first experiments and then run simulations based on those traces. However, the external interference, a very important component in urban scenarios, is extremely difficult to capture in a trace.

C-VeT overcomes these obstacles by exploiting the virtualized router implementation and running multiple applications (eg, routing algorithms) in parallel on different virtual machines. Namely, each router runs multiple virtual machines and each virtual machine runs one experimental configuration. The proposed software structure is shown in Figure 3.

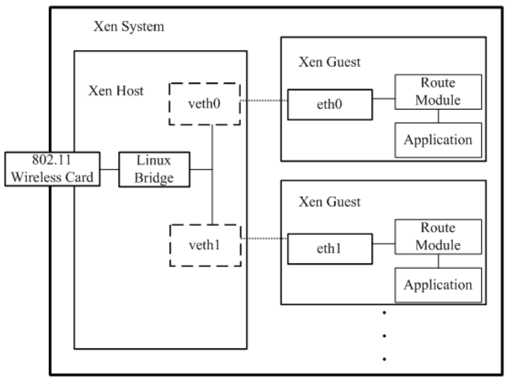


Fig. 3. Software Platform

We use Xen [20] and Gentoo [21] to set up parallel experiments. Each node runs the Linux Gentoo distribution (kernel version 2.6.21) patched with Xen, an open source industry standard virtualization environment that allows

TABLE I
EXPERIMENT SUMMARY

Experiment Name	AODV	OLSR	Interference	Mobile Number
Round 1	√			1
Round 2		√		1
Round 3	√		√	1
Round 4		√	√	1
Round 5	√	√		1
Round 6	√	√	√	1
Round 7	√	√		2
Round 8	√	√	√	2

several virtual machines (Xen guests) to share hardware. Virtual network interfaces are connected to the physical wireless card in the host operating system (Xen host) via a Linux virtual bridge that handles all incoming and outgoing wireless traffic.

The experimental campaign consists of comparing two well known routing protocols, AODV and OLSR. We ran eight rounds of 20-minute experiments with AODV and OLSR activated individually or concurrently. External interference (created by external nodes) was turned on or off on a run by run basis. Table I reports the experiment configurations.

Figure 4 shows the top of the UCLA building around which the vehicles were driven. There are four fixed nodes at the corners. Half of the experiments used two interference nodes indicated by the wave icons in Figure 4. The interference nodes generated layer 2 bursts of random length (avg = 50 pkts) at random intervals (avg = 30 sec).

Fig. 4. Experiment top view

The performance metric of interest is packet hop count, namely the number of hops that a packet takes to reach the destination. Since four nodes form a complete circle, there are two possible ways for a source node to reach destination. In general, the better protocol reacts to topology changes faster and finds a shorter path (ie, smaller hop count). Figures 5 show the packet hop count distribution for experiment rounds 1 through 4. Suppose we pick the result from AODV round 1 and OLSR in round 3, we see OLSR perform better route selection than AODV. However, if we pick the result from AODV in round 2 and OLSR in round 4, we get the opposite result. Generally, in order to provide consistent comparison result, experiments are performed multiple times and hope the special cases are evened out. However, each rounds in our experiment lasted twenty minutes, and the result still varies. When real mobile obstacles and wireless interference are involved, even if the mobility is simple, it could take too long for each experiment rounds to cover the real situation.

Using C-VeT, Figures 6 and 7 show packet hop count distribution for parallel experiment rounds 5 and 6 respectively. The careful reader will note that in Round 5 the hop count is much lower than in Round 6. This is due to the presence of external interference in Round 6. If we ran OLSR alone in Round 5, and AODV alone in Round 6, we would have reached the WRONG conclusion, assuming that OLSR performs better than AODV. Round 7 and 8 are parallel experiment with two vehicles circling the building, and traffic are sending from one vehicle to another and similar comparison results are shown in Figures 8 and 9. By running the two protocols SIMULTANOUSLY we can correctly conclude that AODV is superior to OLSR. Disclaimer: AODV performs better in this particular configuration, with few nodes and a very dynamic topology. However, in larger, static configurations, OLSR was shown to outperform AODV. Thus, we are not claiming that AODV ALWAYS outperforms OLSR. Besides, the purpose of this case study is to prove the consistency of the comparison methodology; it is not to prove the superiority of one scheme over the other in all possible conditions.

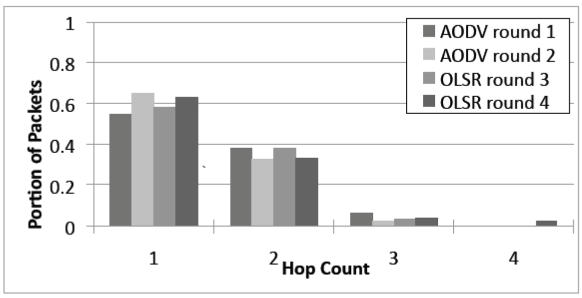


Fig. 5. Parallel experiment round 5

V. CONCLUSIONS AND FUTURE PLANS

In this paper we have argued that the Vehicular Testbed experiments represents an essential intermediate step toward large scale, urban deployment of VANET protocols and applications. Two important roles of the Testbed were identified: validation of the models that are to be used in simulator and emulator experiments, and; validation of the correctness of protocols and methodologies before they are deployed in the real world. We have illustrated these two functions in the UCLA C-VeT Campus Testbed.

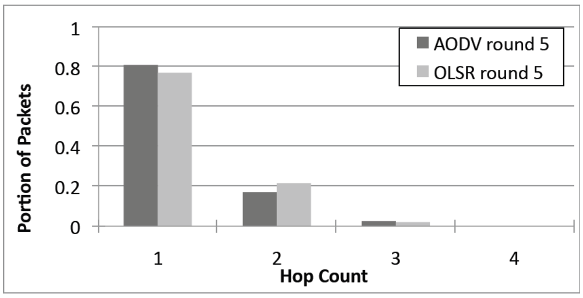


Fig. 6. Parallel experiment round 5

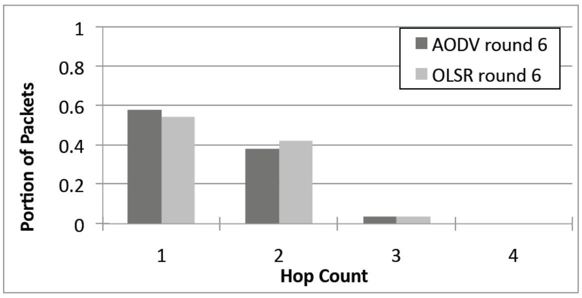


Fig. 7. Parallel experiment round 6

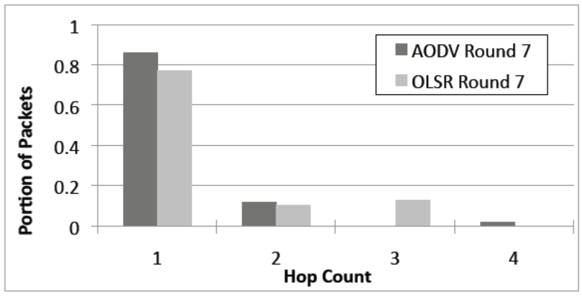


Fig. 8. Parallel experiment round 7

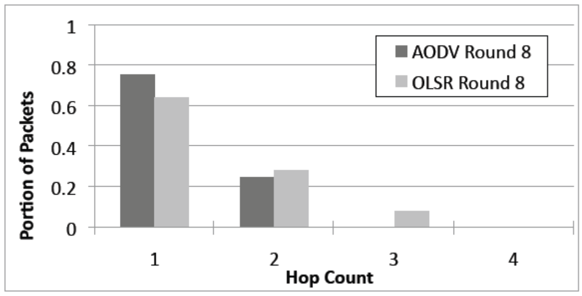


Fig. 9. Parallel experiment round 8

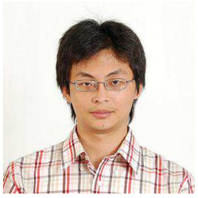
First, the C-VeT testbed validates the corner model so it can be applied to large scale scenario which can only be experimented through emulation or simulation. Next, we use C-VeT to perform parallel experiment to compare different network configuration under the same mobility and external interferences.

In the future we plan to validate other critical models with the C-VeT testbed. In particular, spectrum sensing performance with cognitive radios; MIMO channel acquisition performance; energy models associated with different MIMO channel strategies, etc. For processing efficiency, these models must be abstracted and approximated before implementation in simulators and emulators and use in medium and large scale experiments. Regarding protocol validation, we plan to implement and test several protocols in our controlled C-VeT environment, for example epidemic dissemination, content centric routing, secure network coding, etc. After Campus validation, these protocols will be run in a real traffic scenario (eg Westwood) with private cars. In fact, one of the first experiments on our list is the virtual, parallel comparison of AODV and OLSR in the Westwood area adjacent to UCLA.

REFERENCES

[1] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch and D. Vollmer, *"Position-aware Ad Hoc Wireless Networks for Inter-vehicle Communications"*, Mobihoc 2001.
[2] DSRC, "http://grouper.ieee.org/groups/scc32/dsrc/".
[3] eSafty, "http://europa.eu.int/information_society/programmes/esafety".
[4] CarTALK2000, "http://www.cartalk2000.net".
[5] CVeT Testbed "http://www.vehicularlab.org/testbed.php".
[6] F. Bai, H. Krishnan *"Reliability Analysis of DSRC Wireless Communication for Vehicle Safety Applications"*, IEEE Intelligent Transportation Systems Conference (ITSC), 2006.
[7] Antonio Capone , Stefano Napoli , Alberto Pollastro *"MobiMESH: An Experimental Platform for Wireless MESH Networks with Mobility Support"* Proc. of ACM QShine 2006
[8] CarTel "http://cartel.csil.mit.edu/".
[9] DOME "http://prisms.cs.umass.edu/dome/".
[10] ORBIT: Open-Access Research Testbed for Next-Generation Wireless Networks, "http://www.orbit-lab.org/".
[11] Devashish Rastogi, Sachin Ganu, Yanyong Zhang, Wade Trappe, and Charles Graff, *"A comparative study of AODV and OLSR on the ORBIT testbed"*, Milcom 2007.
[12] CMU wireless emulator "http://www.cs.cmu.edu/∼emulator/".
[13] APE testbed "http://apetestbed.sourceforge.net/".
[14] PlanetLab "http://www.planet-lab.org/".
[15] CRAWDAD "http://crawdad.cs.dartmouth.edu/".
[16] E. Giordano, R. Frank, G. Pau, and M. Gerla, *"Corner: a realistic urban propagation model for VANET"*, in The Seventh International Conference on Wireless On-demand Network Systems and Services (WONS), 2010.
[17] CORNER: a Realistic Urban Propagation Model for QualNet "http://www.vehicularlab.org/archives/vergilius/cornerQualnetManual.pdf".
[18] MadWifi project "http://madwifi-project.org/".
[19] Google Earth "http://earth.google.com/".
[20] Xen "http://www.xen.org/".
[21] Gentoo "http://www.gentoo.org/".

Mario Gerla Dr. Mario Gerla is a Professor in the Computer Science Dept at UCLA. He holds an Engineering degree from Politecnico di Milano, Italy and the Ph.D. degree from UCLA. He became IEEE Fellow in 2002. At UCLA, he was part of the team that developed the early ARPANET protocols under the guidance of Prof. Leonard Kleinrock. He joined the UCLA Faculty in 1976. At UCLA he has designed network protocols including ad hoc wireless clustering, multicast (ODMRP and CODECast) and Internet transport (TCP Westwood). He has lead the ONR MINUTEMAN project, designing the next generation scalable airborne Internet for tactical and homeland defense scenarios. He is now leading several advanced wireless network projects under Industry and Government funding. His team is developing a Vehicular Testbed for safe navigation, content distribution, urban sensing and intelligent transport. Parallel research activities are wireless medical monitoring using smart phones and cognitive radios in urban environments. He has served as a Technical Program Committee member of many international conferences, and is active in the organization of conferences and workshops, including MedHocNet and WONS. He is on the Steering Committee of MobiHoc. He has chaired and cochaired several conferences including Mobicom in 2006 and most recently WoWMoM and IWCMC in 2011. He is on the IEEE TON Scientific Advisory Board.



Jui-Ting Weng received the B.S. and M.S. degrees in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 2003 and 2005, respectively. He is a PhD candidate in Computer Science at UCLA. His research interests include vehicular ad hoc network, wireless communication, wireless mesh network and mobile computing in ad hoc networks.

Eugenio Giordano was born in 1982 in Reggio Emilia, Italy. He received both Laurea and Laurea Magistrale degrees in telecommunications engineering from the University of Bologna, respectively in 2003 and 2006. He has been awarded with the Microsoft Research European PhD Scholarship in 2007. He received both Master of Science and Doctor of Philosophy degrees in Computer Science from the University of California Los Angeles respectively in 2009 and 2011. His research interests include the design, performance and evaluation of ad hoc wireless networks with a focus on inter-vehicular applications.

Giovanni Pau received the Laurea Doctorate degree in computer science and the Ph.D. degree in computer engineering from the University of Bologna, Italy, in 1998 and 2003, respectively. He joined the Computer Science Department at the University of California, Los Angeles (UCLA) as a Research Scientist. His research interests include wireless ad hoc networks, urban sensing, and distributed mobile computing. He is the architect of the UCLA Campus Vehicular Testbed, an experimental facility to support urban sensing for pollution monitoring and Intelligent Transportation Systems. He authored numerous research papers on vehicular networks, wireless multimedia, and peer-to-peer systems.