Emulation platform design for multimedia applications over vehicular networks

Sergi Reñé Vicente[a]

[a] *Department of Telematics Engineering, Universitat Politècnica de Catalunya*
C/Jordi Girona 1 i 3, Campus Nord UPC 08034 Barcelona,Spain

Abstract

Safety applications seems that will be decisive for a successful introduction to the automotive market for the vehicular networks. However, another kind of applications could be very helpful in order to reach the maximum number of equipped vehicles after market introduction, because can attract a greater number of users and facilitate a vehicular infrastructure investment because vehicular communication must provide business opportunities for Internet service providers to generate revenue. One of these kind of applications is live video streaming over vehicular networks. Video streaming is an attractive feature to many applications, such as emergency live video transmission, video on demand services, road-side video advertisement broadcasting and inter-vehicle video conversation. Test and evaluate implementations in a real testbed environment could be very costly and difficult in this kind of networks. Simulations are still commonly used as a first step in any development for vehicular networks research. Therefore, to test this kind of applications an emulation platform for multimedia applications over vehicular networks is presented in this article. We've studied the performance of video streaming services in a infrastructure environment over a highways taking special account in the losses that produces handovers during the communication caused by the network mobility.

Keywords:
vehicular network, emulation, video streaming

1. Introduction

Mobility has changed the way people communicate. Nowadays, as Internet becomes more global, demands for mobility are not restricted to single terminals. Road and vehicle circulation systems are one of the most important infrastructures and are supporting the humans daily life. Intelligent Transportation Systems (ITS) aim to optimize the social costs of road systems and enhance their security as well as drivers comfort by allowing such services as fleet management, navigation, billing, multimedia applications, etc. Vehicular Ad hoc Networks (VANET) are becoming a reality mainly focused on navigation safety applications, but vehicular networks are not only useful for safety applications. Another kind of applications are also very important for the succesful deployment of vehicular networks. Infotainment services, combining information with entertainment, as can be Internet access, multiplayer games, multimedia applications, videoconference, etc., could be an impulse not only for users, but also for network operators to do an infrastructure investment necessary for the vehicular communications because infotainment applications could be an interesting business opportunity.

Applications for vehicular networks are grouped into safety and non-safety applications. The communication requirements for this two kind of applications are different. For example, non-safety applications establish sessions with peer entities, and safety applications data usually disseminate data in geographical areas, using *in-network* processing. The fundamentally different information dissemination strategy of safety applications results in unique protocol mechanisms for geographically-based data forwarding, congestion control, and reliable data transfer with strong cross-layer dependencies. These mechanisms are not part of TCP/IP protocol. However, safety and non-safety applications must be integrated into a single system, for a successful market introduction of a vehicular communication system.

Vehicular networks are mainly impulsed in Europe by Car2Car Communicacion Consortium [1]. The C2C-C Consortium is an industry consortium of car manufacturers and electronics suppliers that focuses on the definition of an European standard for vehicular communication protocols. The consortium defines a C2C-C protocol stack that offers specialized functionalities and interfaces to (primarily) safety-oriented applications and relies as a communication technology on a modified version of IEEE 802.11. This protocol stack is placed beside a traditional TCP/IP stack, exclusively based on IPv6, which is used mainly for non-safety applications or potentially by any application that is not subject to strict delivery requirements, including Internet-based applications. In this part are included the infotaimnent applications, including multimeda applications.

To provide a global mobility to vehicles from the Internet, the C2C-C architecture use an IP Mobility solution. As vehicles connecting to the Internet using access points change their attachment points while driving, C2C considers Mobile IP [2] support to provide session continuity and global reachability. In fact, C2C-C protocol stack doesn't include Mobile IP but include Network Mobility (NEMO) [3]. When considering that passenger devices can be plugged into car communication equipment, therefore turning a vehicle into an entire moving network, NEMO principles have clear benefits in the discussed scenario (i.e. passenger devices shielded from mobility, centralized mobility management).

Automobiles could benefit from better bandwidth, delay, and probably cheaper communications by forming vehicular ad-hoc networks. VANET can also be used to reach the infrastructure from a vehicular node that is not able to communicate with the infrastructure directly because is not within the coverage area of the access points. NEMO Basic Support alone does not provide connectivity over multi-hop, intermittent access to the infrastructure. For this purpose, NEMO BS has to inter-operate with a VANET routing protocol and this is still an open topic.

As far as the wireless technology is concerned, a variant of Wireless LAN IEEE 802.11 called IEEE 802.11p [4] is currently considered as best candidate for a basic safety-oriented system, especially in terms of propagation behavior and overall complexity. Frequency allocation around 5.9 GHz in a protected frequency band dedicated to road safety . Due to the limited available bandwidth compared with the large scale of VANETs, a high deployment complexity of multi-channel operation schemes and a lack of congestion control algorithms in IEEE 802.11 MAC suitable for scenarios with high vehicle density, the use of wireless resources has to be controlled in a distributed way, so that messages with high priority and hard real-time constraints can be delivered immediately. As a consequence, the allocated frequency bands will most likely be reserved exclusively for safety applications. Other types of data traffic, such as for comfort and infotainment, may rely either on different frequency bands or on alternative wireless technologies. In particular, one or more variants of the IEEE 802.11a/b/g standard family could be installed in extended systems with minimum additional complexity.

Vehicular networks based on short-range communication involve several entities and different network domains, as depicted in figure 2. Vehicles are equipped with devices termed On-Board Units (OBU), which implement the communication protocols and algorithms. Units of different cars can communicate with each other or with fixed stations installed along roads termed Road Side Units (RSU). OBUs and RSUs implement the same protocol functionalities and form a self-organizing network, here referred to as the Ad-hoc Domain. These units differ from each other with respect to the networks they are attached to: OBUs offer an interface to the set of driver and passenger devices present in a car, which are called Application Units (AUs). The mobile network, composed of AUs, defines a domain that is usually termed In-Vehicle Domain. RSUs can either be isolated or attached to a larger structured network. In the first case, their function is to distribute static information (e.g. dangerous curve, construction site ahead) or simply to extend the OBUs communication range by acting as forwarding entities. In the second, RSUs distribute information towards or from a remote entity (e.g. control center). They can also connect the vehicular network to an infrastructure network, which is generally referred to as Infrastructure Domain.

Multimedia data, specially video, if feasible, is not only useful for entertainment, it also will help enhance navigation

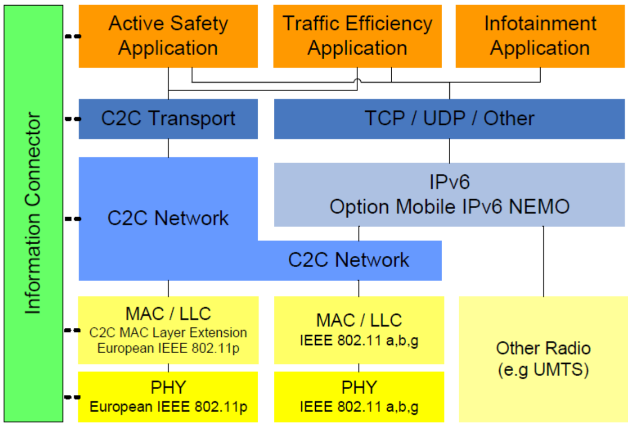

Figure 1: Protocol architecture defined by the Car-to-Car Communication Consortium (C2C-CC) [1]

safety. For example, videos clips of an accident or dangerous situation ahead will provide drivers warning advertisements with precise information. This will allow them to make a more informed decision (whether to proceed or turn back) based on personal priorities and/or on vehicle capabilities. Suppose that a critical traffic/safety situation occurs on a highway, as could be major traffic congestion, natural disaster, fire, etc. In such cases, video streaming could be triggered on one or more lead cars and propagated to vehicles following several km behind - to visually inform the drivers of the problem and allow them to decide if they should turn around. Besides private vehicles, also first responders and rescue operations can greatly benefit from more prompt and precise situation awareness delivered by such video streams.

In the previous paragraph a set of video applications are presented. However, this paper is focused on multimedia applications over vehicular networks focused on video on demand services in an infrastructure scenario, where a video server is placed in the infrastructure domain and vehicular nodes access to this server during a travel.

Vehicular applications developers needs a framework to test their applications. One way to test the applications can be testing the applications in a real environments. This testing scenario obviously is the one that fits better to the real conditions but it has drawbacks about the difficult and the cost to prepare a fleet of cars to do the test. Another way can be using network simulators (e.g. ns-2, opnet, etc) but this simulators doesn't work with real applications and the simulation must be using a model of the traffic that the application generates.

In vehicular networks, packets may be corrupted and lost due to channel errors and collisions. These type of packet losses tend to be random and diverse locally and thus can be countered efficiently with a local recovery strategy. However in a highway scenario where a video on demand service is offered, the main packet losses cause is the many handovers during the whole communication.

Our contribution is the design of a test framework based on an emulation platform combined with a network and mobility

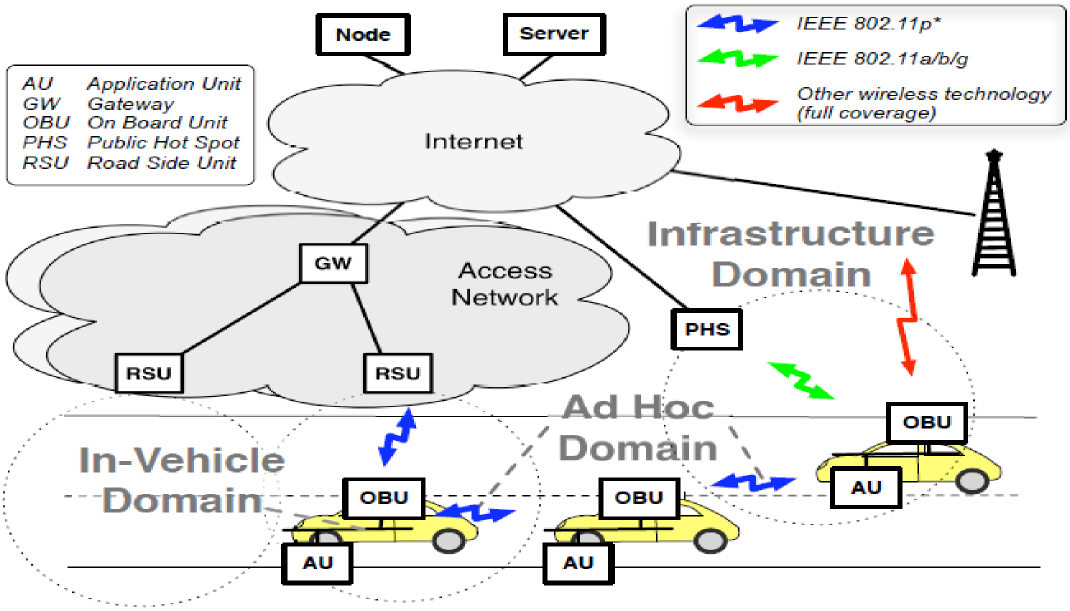

Figure 2: System architecture, currently assumed by the Car-to-Car Communication Consortium (C2C-CC) [1]

simulators that provides the network mobility simulation to a real application installed in a virtual machine and we've analyzed the behaviour of video streaming in a highway infrastructure scenario, and how affect packet losses when a movie clip is played.

The reminder of this article is organized as follows. Section 2 describes some background about Mobile IP and related work on vehicular networks simulators. In section 3 the proposed emulation platform is presented. The reference scenario used for the simulations is presented in section 4. The simulations results about the video service applications are analyzed in the section 5. Section 6 concludes the paper.

2. Background and Related Work

2.1. Mobile IP and Fast Handovers

Mobility is an important issue to take into account for vehicular networks. Vehicular nodes needs to be reach from the wired network or the Internet to access to infotaiment services (e.g., web services or video on demand). It's evident that cars, through its wireless interfaces, can not be attached to the network only within a static point due to its mobile nature.

Mobility can be handled at different layers of the TCP/IP stack. These different options are:

- Physical/Link layer: This is the faster and seamless method. However is limited to one single technology (e.g.,2G and 2.5G cellular networks, or IEEE 802.11b), and is limited to a single domain. For these reasons this method must be discarded because doesn't offers a global mobility solution.
- Network layer: It is the only common layer at the TCP/IP protocol stack, therefor only one solution is required, but is tricky to do it optimally.
- Transport layer: It could be interesting to provide mobility at this layer, but the main drawback is that it would require different solutions per each transport protocol (e.g.,mSCTP).
- Application layer: Some applications already do, but again every application would require to provide mobility support(e.g.,SIP).

Therefore, to provide a global mobility solution to vehicular nodes Network Layer mobility must be implemented. There exists a set of protocols into the IP protocol standardized in order to allow mobility. IP Mobility protocols, such as Mobile IPv4 [5], Mobile IPv6 [6] and Network Mobility [3], allow Mobile Nodes (MNs) to remain reachable while moving around in the Internet. Each MN is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a MN is also associated with a Care-of-Address (CoA), which provides information about the MN's current location. Packets destined to the MN's home address are transparently routed to its CoA. For this purpose, the protocol provides for registering the CoA with a Home Agent (HA) in the home network. The HA intercepts packets destined for the MN and forwards them to the CoA through a tunnel. After arriving at the end of the tunnel, each packet is then delivered to the MN. Usually, a Foreign Agent (FA) in the foreign network would be the endpoint of the tunnel. In this case, the CoA of the MN is the IP address of the FA. Upon receiving tunneled datagrams, the FA decapsulates them and delivers the inner datagram to the MN.

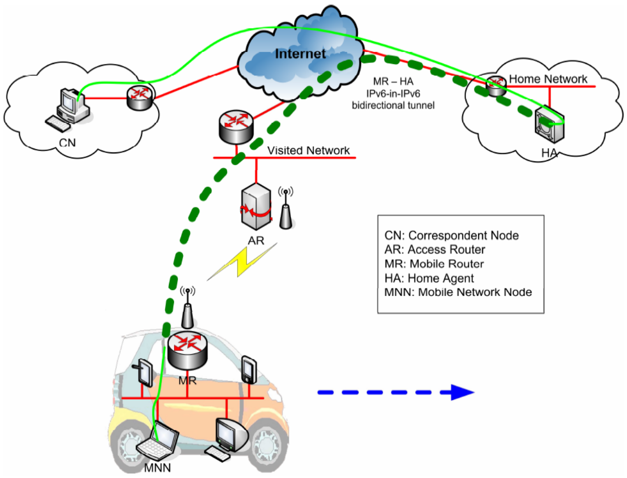

Figure 3: IP mobility example

There are two possible modes of communication between the MN and a Correspondent Node (CN). The first mode, bidirectional tunneling, does not require that the CN supports Mobile IP. Packets from the CN are routed to the HA and then tunneled to the mobile node. Packets to the CN are "reverse tunneled" from the MN to the HA and then routed normally from the home network to the CN. Such routing is called "triangular routing". The second mode, "route optimization", requires the MN to register its current binding at the CN. In this case, packets from the CN can be routed directly to the CoA of the MN and the triangular routing via the HA is avoided.

Since L3 protocols are decoupled from lower, technology-specific layers, Mobile IP is just as suitable for mobility across homogeneous media as for mobility across heterogeneous media. For example, Mobile IP should facilitate node movement from one wireless LAN cell to another, as well as node movement from a wireless LAN cell to an UMTS cell, provided as can be the MN is equipped with the appropriate link layer devices.

Handovers between subnets served by different FAs (L3 handovers) require a change of the CoA and a succeeding registration of the new CoA with the HA. This process takes some time to complete as the Registration Request propagates through the network. During this period of time the MN is not able to send or receive IP packets. The latency involved in Layer 3 handovers can be above the threshold required for the support of delay-sensitive or real-time services.

To achieve seamless handovers during a communication exists some methods that involves cross-layer messages to avoid blackouts during the time a MN detach to one access point and attach to another access point. These seamless handovers are achieved predicting a handover listening for a near access point and preparing the exchange between APs before the MN is detached from the previous AP. FMIPv6 [7] implements this fast handovers technique in a standard for Mobile IPv6

In general, since Fast Handover buffers all incoming packets after a layer 2 trigger, no packets are lost during the handover procedure. However, if an MN moves to another APs area before establishing a forwarding tunnel, some packets may be lost. Specifically, in Fast Handover, a tunnel is established after some messages exchanges as can be RtSolPr/PrRtAdv and the HI/HACK message exchanges.

2.2. *Related Work*

Data delivery through vehicular networks could be complex by the fact that vehicular networks are highly mobile and could be frequently disconnected. Existing data delivery schemes either pose too much control or no control at all on mobility, and hence not suitable for vehicular networks. Most research in vehicular networks relies on simulations for evaluation.

To evaluate vehicular protocols and services, the first option is to perform an outdoor experiment but it is costly and have several drawbacks. For this purpose software simulations can play a vital role in imitating real world scenarios.

Vehicular networks relies on and is related to two other simulations for its smooth functioning, namely traffic simulation and network simulation. Network simulators are used to evaluate network protocols and application in a variety of conditions. The traffic simulators are used for transportation and traffic engineering. Traffic simulators are essential because one of the most important parameters in simulating ad-hoc networks is the node mobility. It is important to use a realistic mobility model so that results from the simulation correctly reflect the real-world performance of a VANET. For example, a vehicle node is typically constrained to streets which are separated by building, trees or other objects. Such obstructions often increase the average distance between nodes as compared to an open-field environment.

In traffic research four classes of traffic flow models are distinguished according to the level of detail of the simulation: macroscopic, microscopic, sub-microscopic and mesoscopic models. In macroscopic models traffic flow is the basic entity. Microscopic models simulate the movement of every single vehicle on the street, mostly assuming that the behaviour of the vehicle depends on both, the vehicle's physical abilities to move and the driver's controlling behaviour. Mesoscopic simulations are located at the boundary between microscopic and macroscopic simulations. Herein, vehicle movement is mostly simulated using queue approaches and single vehicles are moved between such queues. Sub-microscopic models regard single vehicles like microscopic but extend them by dividing them into further substructures, which describe the engine's rotation speed in relation to the vehicle's speed or the driver's preferred gear switching actions, for instance. This allows more detailed computations compared to simple microscopic simulations. However, sub-microscopic models require longer computation times. This restrains the size of the networks to be simulated. Therefore the majority of the traffic simulators used for vehicular networks purposes are built using microscopic models because it is the model that fits better to the vehicular networks requirements.

To satisfy the need of vehicular networks, a solution is required to use network and traffic simulators together. There exists severals solutions into what can be called VANET simulator.

Traffic/network simulator	Approach	Microscopic mobility	Emulation	Fast handovers	Flexible	Well-accepted
MoVE	Federated	Yes	No	No	Yes	Yes
TraNS	Federated	Yes	No	No	Yes	Yes
NCTUns	Integrated	Yes	Yes	No	No	No
VanetMobiSim	Federated	Yes	No	No	No	No

Table 1: VANET simulators comparison

Two of the solutions presented above are built on top of an open source micro-traffic simulator called SUMO [8] and Network Simulator version 2 (ns-2) [9]. One of this solution is MOVE (MObility model generator for VEhicular networks) [10] a tool designed to facilitate users to rapidly generate realistic mobility models for VANET simulations. The output of MOVE is a mobility trace file that contains information of realistic vehicle movements which can be immediately used by popular simulation tools such as ns-2 or qualnet, using the same program configuring properly the network simulator, or exporting the mobility traces to be used later by the network simulator. In addition, MOVE provides set of Graphical User Interfaces that allows the user to quickly generate realistic traffic simulation scenarios without the hassle of writing traffic simulation scripts as well as learning about the internal details of the simulator. The another solution that use SUMO and ns-2 is TraNS, Traffic and Network Simulation Environment [11]. This tool also link the traffic simulator SUMO and network simulator ns-2. Moreover provides an specific interface for interlinking road traffic and networking simulators, called TraCI that performs a full-blown evaluation of VANET applications that influence vehicles mobility, i.e., safety and traffic efficiency applications, using a TCP based client/server architecture to provide access to a running road traffic simulation. One of the advantages of these tools is that they use widely extended in the research community network simulators.

Another option as a VANET simulator is VanetMobiSim [12]. VanetMobiSim a is an extension for the CANU Mobility Simulation Environment (CanuMobiSim), a flexible framework for user mobility modeling. CanuMobiSim is JAVA-based and can generate movement traces in different formats, supporting different simulation/emulation tools for mobile networks (NS2, GloMoSim, QualNet, NET). CanuMobiSim originally includes parsers for maps in the Geographical Data Files (GDF) standard and provides implementations of several random mobility models as well as models from physics and vehicular dynamics. The VanetMobiSim extension focuses on vehicular mobility, and features new realistic automotive motion models at both macroscopic and microscopic levels. At macroscopic level, VanetMobiSim can import maps or randomly generate them. Also, it adds support for multi-lane roads, separate directional flows, differentiated speed constraints and traffic signs at intersections. At microscopic level, VanetMobiSim implements new mobility models, providing realistic car-to-car and car-to-infrastructure interaction. According to these models, vehicles regulate their speed depending on nearby cars, overtake each other and act according to traffic signs in presence of intersections.

The last solution presented is NCTUns [13]. In contrast with the other solutions presented previously, it is a integrated solution. It is a network simulator that has a map editor to perform traffic simulations. However the mobility component is highly integrated with the network simulator. This makes it hard to use realistic road traffic simulators, such as, those developed within the Intelligent Transportation Systems (ITS) community. NCTUns its quite less flexible than other solutions and, as network simulator, it is not easy to add new protocols as, for example, with the widely accepted for the research community ns-2.

NCTUns solution, as well as the others solutions doesn't support fast handovers for Mobile IP. Therefore they are unuseful for the scenario where seamless handovers are vital to test multimedia applications or another QoS aware data traffic. In the table 1 a comparison between the different solutions is presented.

3. Emulation Platform Implementation

The emulation platform is built using different software projects, some of them introduced in chapter 2.2, and mixing them to get an enhanced platform. The novelty of our platform is to use live network applications and traffic mobility simluation, and the possibility to test real applications in a live network adding the automotive traffic mobility transparently to the tested application. The platform also supports mobility management, including smooth handovers using fast handovers techniques. This adds to the emulation platform the ability to perform better than any other solution presented in a vehicular networks using a infrastructure domain, given that the solutions presented in section 2.2 are more focused in VANET using ad hoc protocols but aren't focused in vehicular networks interacting with an infrastructured network. NCTUns is the only solution that are more focused on simulate vehicular networks with intrastructure, however does not include fast handovers, like the rest of solutions.

The emulation platform is divided in three logical modules based on functionality. These modules are: Application virtualization, network emulation and traffic mobility.

The application virtualization is mainly formed by UML machines [14]. This virtual machines are able to execute real software to perform a realistic testbed.

The network emulation is mainly formed by Network Simulator ns-2 [9] in emulation mode (nse) using some addons to enhance the functionality of this wellknown network simulator and add the possibility to connect the simulator to the application virtualization. One of this modification rises on an optimization to work with wireless devices in real-time network emulation or, for example, add the support to IEEE 802.11p vehicular technology.

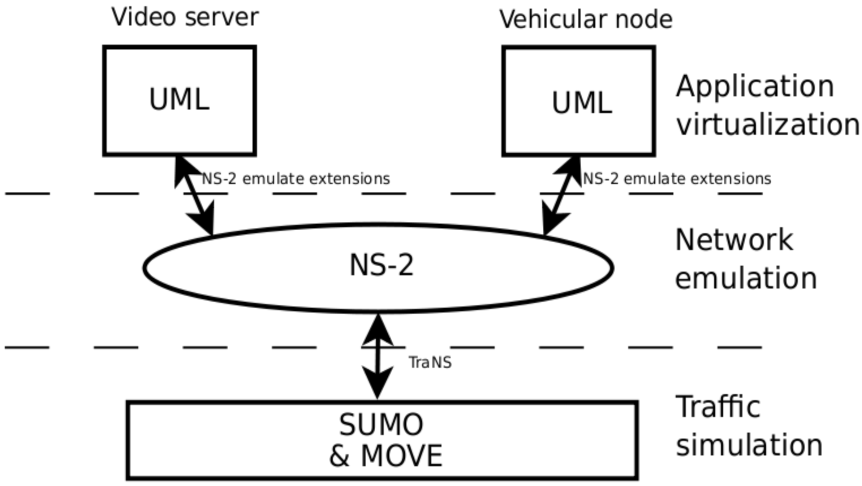

Figure 4: Emulation Platform

The traffic simulation is the responsible part to build a pattern mobility where the nodes acts as a mobile vehicles simulating the mobility across the roads. This mobility patterns can be built creating maps using a manhattan grid, for example, or importing real maps. Besides creating the maps, it is needed to create the nodes movement, configuring the vehicles introducing rates, max speed, etc.

The next paragraphs explains more detailed the logical modules of the emulation platform.

3.1. Application Virtualization

The live network in the emulation platform consist in a virtual network among a set of User-Mode-Linux (UML) virtual machines running in a host machine. UML is an extended Linux kernel executed as an user-space process on a Linux host. It can be assigned physical resources separated from the host machine and connected to a network from the host machine and connected to a network trough various transport mechanisms. On of the possible transport types for UML is the TAP virtual Ethernet device. In the next section regarding network simulator the transport mechanism to connect between the virtual machines and ns-2 is presented in more detail.

UMLs virtual machines emulate the nodes in our testbed. These can be both, mobile nodes (cars) or fixed nodes (usually servers). The applications run inside the virtual machines and do not notice that they communicate through a simulated network, so the applications run as they run in a real-system and the network is simulated as a vehicular network transparently to the tested applications.

The main advantages to work with virtual machines is the cost saving of infrastructure (it's just needed an unique real system -host machine-), a very important cost saving in a so complex networks as can be vehicular networks with a lot of nodes communicating in a test scenario, with a very difficulties to control the movement of all the nodes at the same time. Another advantage is the management simplification, because there is just a unique point to control, in contrast to manage and setting up a lot of disperse nodes in the network.

However the complete transparency using virtualization is so difficult. Virtualization introduce an additional process level that supposes an overhead due to the resources consume. To reach a performance similar to a real network there is needed very powerful host machines. Due to this reason, in the presented test scenario in the section 4 using the emulation platform for multimedia applications over vehicular networks there will be just two nodes virtualized as a UML virtual machine, the video streaming server and the vehicular video client. The another nodes involved in the reference test scenario will be emulated by the network simulator. However, in a different testing scenarios could be used more virtual machines, but the number of virtual machines used in the tests will affect directly to the resource requirements or the real-time performance.

Live555 [15] is used to test multimedia applications in the testbed. Live555 is an open source (LGPL) C++ library for multimedia streaming. It supports open standards such as RTP/RTCP, RTSP, SIP for streaming, and can also manage video and audio formats such as MPEG, H.263+ and JPEG video. It is designed in such a way that it can be easily extended to support more formats. Using Live555 libraries a video streaming server is configured in one side of the communication and VLC media player [?] or MPlayer [?] with live555 libraries in the vehicular node.

3.2. Network Emulation

To emulate the network in the platform is used the widely accepted Network Simulator version 2 (ns-2) using the emulation feature, providing the ability to introduce the simulator into a live network and simulate a desired network between real applications in real-time. It is actively used for wired and wireless network simulations.

To integrate the network simulator with emulation feature into our “live” vehicular network formed by UML virtual machines there is need an extension to connect the ns-2 and the UML machines using TAP interfaces. To enhance the network simulator for a better performance to simulate vehicular networks another extensions are needed. In the next sections these addons for ns-2 are detailed:

3.2.1. NS-2 Emulation Extensions

The NS-2 Emulation Extensions [16] is part of the contributed code of ns-2. This means that it is maintained by users and that has not been incorporated into the ns distributions. This extension enables ns-2 to emulate wireless networks using UML virtual machines and implementing an interface between ns-2 and UML using TAP interfaces, and improves the emulation of wireless networks in ns-2. This is why the real-time requirements of a network emulation introduce an inaccurate timing behavior of the simulator scheduler. These timing errors have a negative impact on the performance of network protocols in ns-2. Even more, they lead to false simulation results in the IEEE 802.11 protocol implementation. The ns-2 Emulation Extensions increase the accuracy of its virtual clock and the exactness of the real-time simulation. So, the utilization of the NS-2 Emulation Extensions permits a more accurate simulation of the wireless networks, and the emulation of real applications into this wireless networks in a real-time using the UML virtual machines. The union between the network simulator and the virtual machines are performed by the Tap Agents and the Network Objects.

Tap Agents. The union point between the network simulator and the UML virtual machines are the Tap agents. The TAP is a Virtual Ethernet network device. TAP virtual interface is created in the host machine. Userland application can write Ethernet frame to /dev/tapX and kernel will receive this frame from tapX interface. In the same time every frame that kernel writes to tapX interface can be read by userland application from /dev/tapX device.

Network Objects. Network objects provide access to a live network for the network simulator using the emulation feature. There are several forms of network objects, depending on the protocol layer specified for access to the underlying network, in addition to the facilities provided by the host operating system. Use of some network objects requires special access privileges where noted. Generally, network objects provide an entry point into the live network at a particular protocol layer (e.g. link, raw IP, UDP, etc) and with a particular access mode (read-only, write-only, or read-write). Some network objects provide specialized facilities such as filtering or promiscuous access (i.e. the pcap/bpf network object) or group membership (i.e. UDP/IP multicast). For the union between the Tap Agents related the UML virtual machines and the ns-2 a network object called Network/Raw.

3.2.2. Mobile IP in NS-2

NS-2 supports Mobile IP for wired and wireless networks. Mobile IP module for NS-2 it was developed by Sun Microsystems and is based on NS wired model. It includes all Mobile IP entities like HA, FA and MN. The HA and FA entities are realized as base stations. They have registering agent to send beacons to the mobile node to detect mobility, encapsulates and decapsulates data and replies to solicitations from MN. The MN have registering agents, which receives and responds to beacons and send out solicitations to HA or FAs. The MN stores the address advertised from the BS within its service coverage and registers them in a list. When no beacon message is received, the list entry expires itself and is removed. This indicates MN to perform a handover as it leaves the service range of the old BS. Afterwards MN chooses a BS from the list as its new FA. If the list is empty the MN sends an Agent Solicitation Message. Any BS that receives this solicitation sends an advertisement message to allow the MH to register with it.

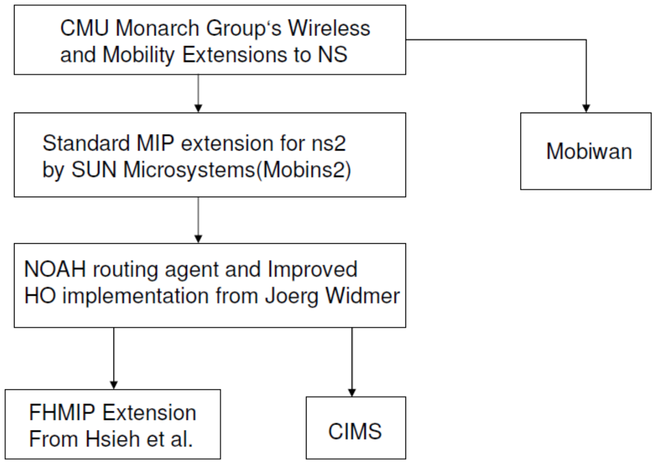

Figure 6: ns-2 implementation of Mobile IP

The handover starts when MH sends Registration request message. The new BS in the case FA receives and forwards the request to the HA. The HA then updates the MNs CoA in its address binding, and it installs encapsulator to tunnel all future IP packets to the MN via its new BS. Afterwards the HA sends back a Registration Reply message to the BS and in terms informs the MN. If the handover process is successful the MH will register the new CoA. During the handover the packets are dropped until the new connection is established even though the MN could still communicate with its CN via the old BS. This is result of received beacon message from a new BS, i.e. MN sends registration request and uses this new BS as its new FA.

Problems with NS-2 Mobile IP model. The implementation of Mobile IP in ns-2 works properly in a scenario where smooth handovers are not required. However, the main problem about the NS-2 Mobile IP model to work with the emulation platform for multimedia applications is that doesn't support fast handovers. This is very important in a vehicular scenario, overall in a highway scenario as the scenario it will be shown in the section 4. In this scenario Fast Handovers for Mobile IP are required and the implementation included in ns-2 is not enough.

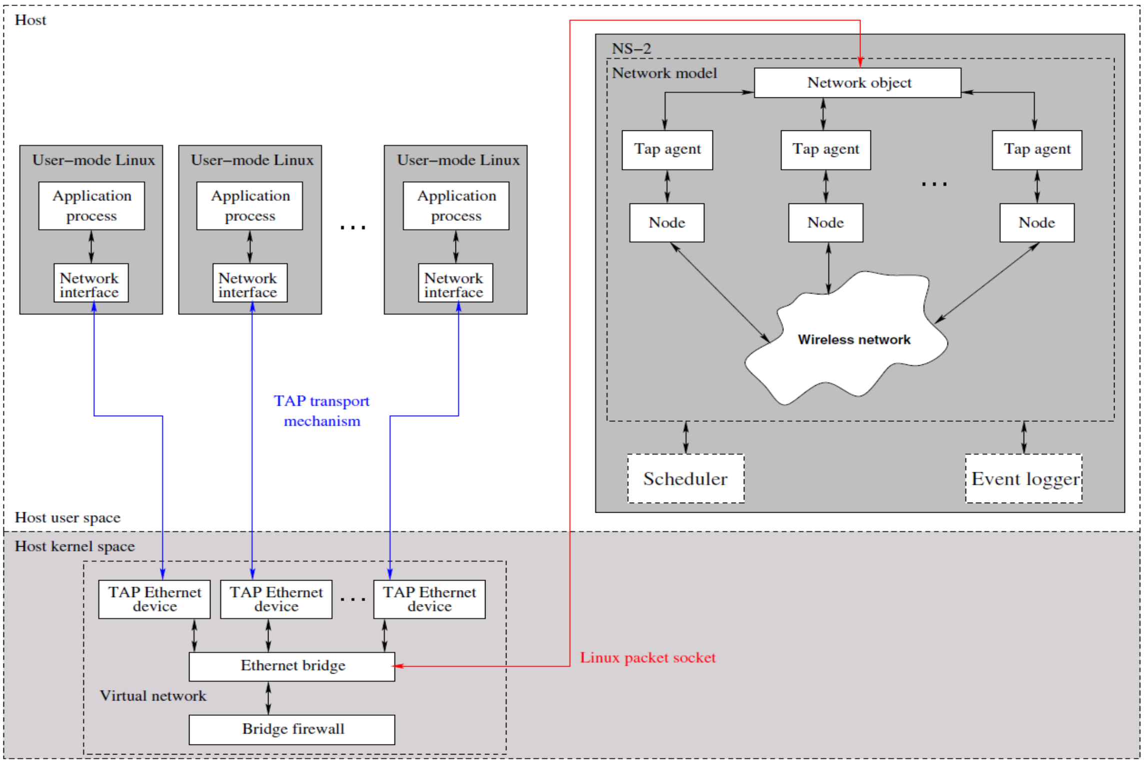

Figure 5: ns-2 emulation extensions

To provide Fast-handover for Mobile IP support to ns-2 an extension developed by Robert Hsieh [17] is added to ns-2, that implements fast handovers [7] and Hierarchical Mobility Management [18].

3.2.3. Routing

The five different ad-hoc routing protocols currently implemented for mobile networking in ns are DSDV [19], DSR [20], AODV [21], TORA [22] and PUMA [23]. However there is a incompatibility with ns-2 implementation of this routing protocols and the implementation of Mobile IP. Just DSDV can work in a mobile environment with infrastructure and mobility. However DSDV is not a routing protocol that works properly in a vehicular scenario because is a proactive protocol and the tradeoff between convergence time or traffic overhead causes that a reactive protocol as can be AODV could be a better routing protocol to work in a vehicular scenario.

However the actual implementation of AODV or DSR doesn't work with Mobile IP, overcoat with Mobile IP with fast handovers. For this reasons, the decision is that it could be interesting don't use adhoc routing protocol. This means that NOAH [24] routing protocol proportioned by the same Robert Hsieh's Fast-handover for Mobile IP patch is used. This is an implementation that emulates the behaviour of a mobile node without using adhoc routing, so the mobiles nodes only connects with the base stations. This behaviour is the desired to work with mobility ip, and for this basic scenario without trying to enhance or test more complex scenarios with multi-hop could be very useful.

3.2.4. IEEE 802.11 extensions

A team from Mercedes-Benz Research and Development North America and from University of Karlsruhe have collaborated to develop a completely new 802.11 Mac and Phy model, called Mac802_11Ext and WirelessPhyExt, respectively [25]. The new model permits configure a lot of new parameters of the Mac and Phy layer that is not possible to configure in the current ns-2 implementation, providing a higher level of simulation accuracy. Using this extensions is possible to use IEEE 802.11p access technology in the network simulator ns-2, using a configuration file that provides the 802.11p parameters.

3.3. Traffic Mobility

The last part of the emulation platform is the traffic mobility module. This modules is the responsible part of create the node movements as a vehicles following the different itineraries defined by the road maps and the different configurable parameters, e.g. max speed limits, road lanes, crossroads, speed and acceleration of the cars, etc.

The traffic mobility module is formed by different open-source projects. The projects are SUMO [26], TraNS [11] and MOVE [10]. SUMO is the core of the traffic mobility module and is used too in the other two applications TraNS and MOVE when works individually. TraNS and MOVE, are extensions or programs that use SUMO and permits the connexion with the network simulator (ns-2). In the emulation project this last two tools are used just to perform a concrete functions. MOVE is used as a graphical user interface to easily construct maps and

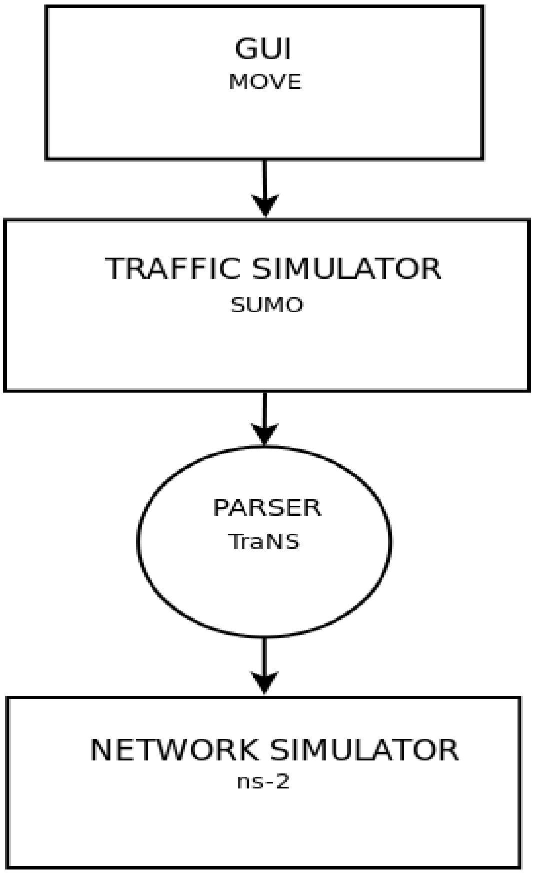

Figure 7: Traffic mobility emulation platform arquitecture

routes for SUMO. TraNS is used as a parser to be able to communicate the traffic simulator module with the network simulator module.

3.3.1. SUMO

The Simulation of Urban MObility (SUMO) is an open source, highly portable, microscopic road traffic simulation package designed to handle large road networks. It is developed by the Institute of Transportation Systems at the German Aerospace Center and is licensed under the GPL. Its development started in the year 2000. The major reason for the development of an open source, microscopic road traffic simulation was to support the traffic research community with a tool into which own algorithms can be implemented and evaluated with, without the need to regard all the artifacts needed to obtain a complete traffic simulation, such as implementing and/or setting up methods for dealing with road networks, demand, and traffic controls.

The main application was to implement and evaluate traffic management methods, but additionally is used for short-term traffic forecast during large events with many participants, and is used for evaluating traffic surveillance using GSM networks. Since 2002, SUMO is also in use at other institutions. Here, the major interest seems to be the evaluation of vehicle-to-vehicle and vehicle-to-infrastructure communication.

SUMO contains parsers for various topologies and routes assignment may also be imported from various sources. However, at that time, SUMO is not able to output traces straightforwardly usable by network simulators. To do this some extensions is needed. In the emulation platform TraNS tool is used. This tools is presented in section 3.3.2.

SUMO uses an own road network description to work. This networks description is built in xml files, built by either converting an existing map using or generating geometrically simple, abstract road maps, using a SUMO tools called NETCONVERT and NETGEN.

NETGEN allows to create networks in a very comfortable way. For some small-sized tests of rerouting strategies, tls-signals etc., this is probably the best solution to get a network one can run some simulations at. The clear naming of the streets also eases defining own routes. But networks generated using NETGEN are of course useless as soon as you want to simulate a real-world network.

Using NETCONVERT one can import road networks from several sources, among them VISUM [27], shape files, and OSM databases [28], setting up real-world networks.

After having generated a network still needs some kind of description about the vehicles. SUMO and GUISIM, a graphical user interface for SUMO, need routes as input for vehicle movement. There are several ways to generate routes for SUMO:

- Using trip definitions: As described above, each trip consists at least of the starting and the ending edge and the departure time.
- Using flow definitions: This is mostly the same approach as using trip definitions, but one may join vehicles having the same departure and arrival edge using this method.
- Using flow definitions and turning ratios: One may also leave out the destination edges for flows and use turning ratios at junctions instead.
- Using OD-matrices: OD-matrices have to be converted to trips first, then from trips to routes.
- By hand: You can of course generate route files by hand.
- Using random routes: This is fast way to fill the simulation with life, but nothing that has something to do with reality.
- By importing available routes.

By now, the SUMO-package contains four applications for processing routes. DUAROUTER is responsible for importing routes or their definitions from other simulation packages and for computing routes using the shortest-path algorithm by Dijkstra. Additionally, in combination with the simulation, the DUAROUTER can compute the dynamic user assignment formulated by C. Gawron. JTRROUTER may be used if you want to model traffic statistically, using flows and turning percentages at junctions. OD2TRIPS helps you to convert OD-matrices (origin/destination-matrices) into trips. The DFROUTER computes routes from given observation point measures.

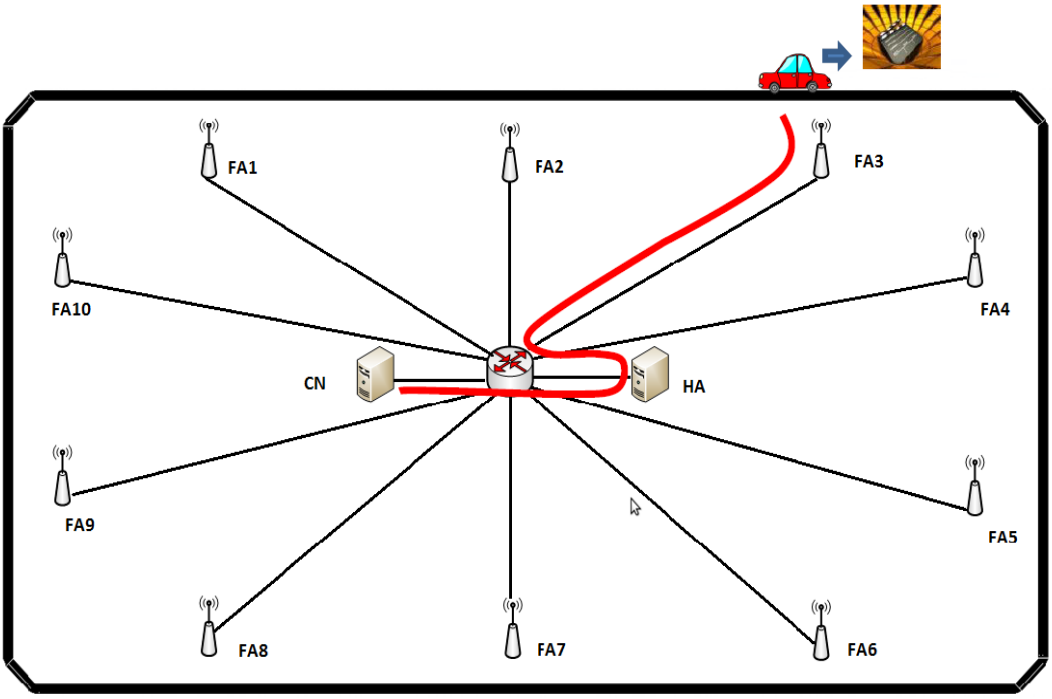

Figure 8: Reference scenario

3.3.2. TraNS

TraNS is a tool that works linking the network simultor ns-2 and traffic simulator SUMO. sing an interface called Interpreter, traces extracted from SUMO are transmitted to ns-2, and conversely, instructions from ns-2 are sent to SUMO for traffic tunning. Accordingly, interactions between the vehicular traffic and network may be implemented. Thus, the network simulator can use realistic mobility models and influence the behavior of the traffic simulator based on the communication between vehicles.

TraNS has two distinct modes of operation, each addressing a specific need. The first mode, termed network-centric, can be used to evaluate, for realistic node mobility, VANET communication protocols that do not influence in real-time the mobility of nodes. One example is user content exchange or distribution (e.g. music or travel information). The second mode, termed application-centric, can be used to evaluate VANET applications that influence node mobility in real-time, and thus during the traffic simulation runtime. Safety applications (e.g., abrupt braking, collision avoidance, etc.) are such examples.

As the test scenario that will be presented in this article is a scenario that do not influence in real-time the mobility nodes, TraNS will be used in the emulation platform in the first mode. However it can be used in the emulation platform in both modes if it is needed.

3.3.3. MOVE

The Mobility Model Generator for Vehicular Networks (MOVE) is a simple parser for the SUMO and enhances SUMO's complex configuration with a nice and efficient GUI. MOVE also contains a parser to generate traces usable by network simulators such as ns-2 or QualNet, but for the emulation platform is not used this platform, because it is a closed parser that just permits perform easy scenarios in network simulator. For this purpose TraNS parser is used.

MOVE consists of two main components: the Map Editor and the Vehicle Movement Editor. The Map Editor is used to create the road topology, and can be manually created by users, generated automatically, or imported from existing real world maps such as publicly available TIGER (Topologically Integrated GEographic Encoding and Referencing) database from U.S. Census Bureau [29]. The Vehicle Movement Editor allows user to specify the trips of vehicles and the route that each vehicle will take for one particular trip. The output of MOVE is a mobility trace, generated based on the information users input in the Map Editor and the Vehicle Movement Editor, which can be immediately used by a simulation tool such as ns-2 to simulate realistic vehicle movements. However, in the emulation platform this output won't be used. The SUMO map and routes created from MOVE will be used from TraNS to create the mobility traces.

4. Reference Scenario

Video streaming over vehicular networks could be applicable in reality. The car engine can provide enough power for intensive data computation and communication. Vehicles can also equip large On-board storage. Thus the node in vehicular

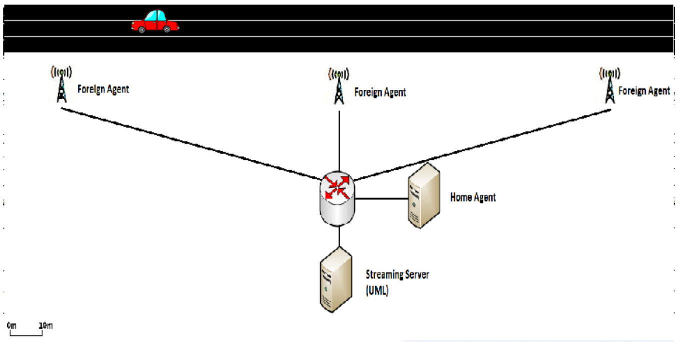

Figure 9: Reference scenario

networks is powerful to forward continuous video data to other vehicles or roadside receivers. Furthermore, the IEEE 802.11g standard can support up to 54Mbps transmission rate, or the vehicular specific IEEE 802.11p standard support up to 27Mbps. Even between high speed driving vehicles within a highway, it is reasonable to expect a 1Mbps data rate [30]. As to the transmission data rate required by compressed video there is enough bandwidth to support video streaming between vehicles.

Using the emulation platform described in this article, a testbed to analyze the capability to support a video on demand service over a highway vehicular scenario is built. The test scenario designed for this purpose is an infrastructure scenario where a set of base stations are deployed over a highway in an overlapped manner, therefore there isn't coverage blackouts in the road. All the base stations are connected to a central router and that is connected to a video streaming server. This streaming server is emulated by an UML machine, as is detailed in section 3.1.

The routing between the video server and the car node is performed always as a single hope between the vehicular nodes and the base stations, therefore no ad-hoc routing is used in this testbed. The goal of the simulations is to make a study of a video streaming service over a highway with a lot of handovers between base stations and analyze how to work a Mobile IP solution for a seamless video decoding. Another reason to avoid ad hoc routing in this testbed is because nowadays there is not a close topic how to manage mobile ip with ad hoc routing protocols. There exists some solutions, as can be [31], but as the goal of this article is not to work with this cases, it is not treated here.

The topology of the test scenario is a square shaped highway with 1000 x 500 m size as depicted in figure 8. The cars round over the highway as many time it is configured in the simulation, and with this topology an infinite simulation could be performed into the closed circuit. The base stations are deployed separared by 300m each one. For the simulations the IEEE 802.11p access technology is used.

5. Simulation Results

To check the possibility of the deployment of a video on demand service over a highway, a set of simulations are performed using the emulation platform. The main problem that could limit the deployment of a video service is the losses that can occurs during the handovers. The cars high speeds in the roads and the amount of handovers must be analyzed to deploy a video service.

5.1. *Losses*

To analyze the losses a CBR UDP traffic without any FEC or ARQ method used in the communication is sent from the server to the vehicular node, simulating a video streaming of a CBR class video. To simulate this CBR stream and check the losses Iperf tool [32] is used. The next graphs shows the losses obtained. The first graph, figure 10, show the losses for four different speeds and its evolution if the CBR data rate is increased. The second graph, figure 11, shows the losses for four different data rates, in function of vehicle speeds.

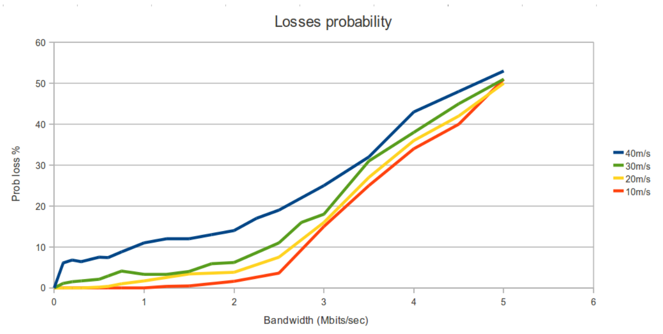

Figure 10: Losses probability for different speeds depending on the source rate

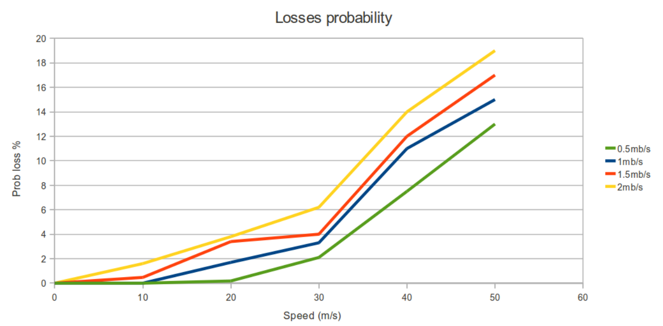

Figure 11: Losses probability for different source rates depending on the speed

Analyzing the figure 10, it could be deduced that for 40 m/s there will be problems reproducing a video stream without any special technique due to the number of losses that are achieved. However, for the other three speeds, 30, 20 and 10 m/s video streaming could be suitable. For 30 m/s a video bitrate greater than 500 Kbps could perform some troubles, however with this bitrate it is possible to reproduce a video with an interesting quality. For the last two speeds, 20 and 10 m/s, it could be seen that the problems will start between 1 and 2 Mbps. This

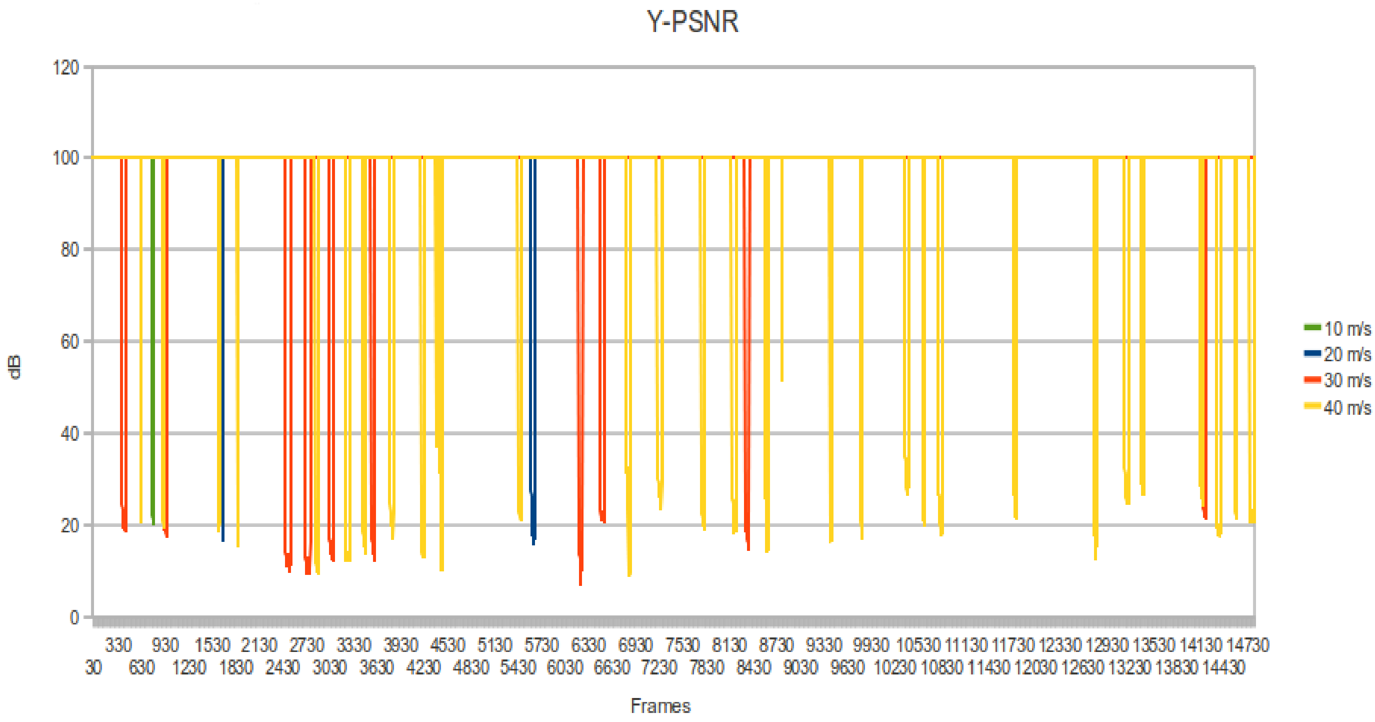

Figure 12: PSNR

represents that for urban mobility a video service it could be deployed with a number of losses that can support an enough video rate to assure a high video quality and, for highway mobility also could be supported a video service, but with a lower video bitrate and a poorer video quality.

5.2. PSNR

The objective of this test is to investigate how is affected a video clip streamed in a vehicular network by the handovers followed during the communication, measuring the quality of the video received compared with the original video. The video used in the simulations is a 352x288 MPEG-II video coded at 500 Kbps. To recover the gaps that are lost during the communication, a resilient decoder is used. When this decoder is not able to recover the lost frame, the previous frame is represented. In the figure 12 the PSNR for different speeds is represented. The PSNR when the video received is the same as the original video is represented as 100 dB. The figure shows the losses of video quality during the handovers. For 40 m/s a lot of gaps during the reproducing are performed. Therefore, this speed are unfeasible to play a video during a travel. For a slower speeds, as can be 30, 20 or 10 m/s, it could be observed that the falls of the quality during the video reproduction are reduced drastically, so it could be feasible to play a video during a travel over a highway going at those speeds.

6. Conclusions and further work

In this article an emulation platform for multimedia applications over vehicular networks is presented. The emulation platform provides a testing enviromnment that permits testing real applications using traffic mobility simulations, providing a fast and cheapert testing environment than outdoor experiments. In the article a set of simulations about live video streaming are presented. In this set of experiments we could experiment how the handovers during the reproduction of a video clip during a travel over a highway limits the overall quality that can be achieved for the decoded video. The losses grows as the video bitrate increments or the vehicle speed, decreasing the video quality percieved by the client, represented with the PSNR of the video decoded. The article permits emulate real video streaming applications to control the vehicle speed and the video bitrate limits for a affordable service deployment. Further research will extend the emulations over vehicular networks to TCP applications with using differnt TCP flavours and video streaming using P2P applications.

7. References

[1] C2C-CC: CAR 2 CAR Communication Consortium, `http://www.car-to-car.org/`.

[2] C. Perkins, Mobile ip, International Journal of Communication Systems 11 (1) (1998) 3–20.

[3] V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert, Network Mobility (NEMO) Basic Support Protocol, RFC 3963 (Proposed Standard) (Jan. 2005).
URL `http://www.ietf.org/rfc/rfc3963.txt`

[4] S. Eichler, Performance evaluation of the IEEE 802.11 p WAVE communication standard, in: 2007 IEEE 66th Vehicular Technology Conference, 2007. VTC-2007 Fall, 2007, pp. 2199–2203.
[5] C. Perkins, IP Mobility Support for IPv4, RFC 3344 (Proposed Standard), updated by RFC 4721 (Aug. 2002).
URL http://www.ietf.org/rfc/rfc3344.txt
[6] D. Johnson, C. Perkins, J. Arkkscreeno, Mobility Support in IPv6, RFC 3775 (Proposed Standard) (Jun. 2004).
URL http://www.ietf.org/rfc/rfc3775.txt
[7] R. Koodli, Mobile IPv6 Fast Handovers, RFC 5568 (Proposed Standard) (Jul. 2009).
URL http://www.ietf.org/rfc/rfc5568.txt
[8] SUMO - Simulation of Urban MObility, http://sumo.sourceforge.net.
[9] Network Simulator, version 2, http://www.isi.edu/nsnam/ns.
[10] F. Kristianto, K. Zhi, H. Mo, K. Lan, Rapid Generation of Realistic Mobility Models for VANET.
[11] M. Piórkowski, M. Raya, A. Lugo, P. Papadimitratos, M. Grossglauser, J. Hubaux, TraNS: realistic joint traffic and network simulator for VANETs, ACM SIGMOBILE Mobile Computing and Communications Review 12 (1) (2008) 31–33.
[12] VanetMobiSim, http://vanet.eurecom.fr/.
[13] S. Wang, C. Chou, NCTUns tool for wireless vehicular communication network researches, Simulation Modelling Practice and Theory 17 (7) (2009) 1211–1226.
[14] J. Dike, et al., The user-mode linux kernel homepage, URL: http://user-mode-linux.sourceforge.net.
[15] R. Finlayson, Live555 streaming media (2009).
[16] D. Mahrenholz, S. Ivanov, Real-Time Network Emulation with ns-2, in: In Proc. of DS-RT, 2004, pp. 29–36.
[17] Source code for ns-extension on HMIPv6 with Fast-Handover, http://mobqos.ee.unsw.edu.au/ robert/nsinstall.php.
[18] H. Soliman, C. Castelluccia, K. ElMalki, L. Bellier, Hierarchical Mobile IPv6 (HMIPv6) Mobility Management, RFC 5380 (Proposed Standard) (Oct. 2008).
URL http://www.ietf.org/rfc/rfc5380.txt
[19] P. Bhagwat, C. Perkins, Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers, in: Proceedings of ACM SIGCOMM, Vol. 94, 1994, pp. 234–244.
[20] D. Johnson, Y. Hu, D. Maltz, The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, RFC 4728 (Experimental) (Feb. 2007).
URL http://www.ietf.org/rfc/rfc4728.txt
[21] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561 (Experimental) (Jul. 2003).
URL http://www.ietf.org/rfc/rfc3561.txt
[22] Temporally-Ordered Routing Algorithm, http://wiki.uni.lu/secan-lab/Temporally-Ordered+Routing+Algorithm.html.
[23] Vaishampayan, R. and Garcia-Luna-Aceves, J.J., Vol. , 2004, pp. 304–313. doi:.
[24] NO Ad-Hoc Routing Agent (NOAH), http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/.
[25] Overhaul of IEEE 802.11 Modeling and Simulation in NS-2, http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php.
[26] D. Krajzewicz, G. Hertkorn, C. Rossel, P. Wagner, SUMO (Simulation of Urban MObility); An open-source traffic simulation, in: 4th Middle East Symposium on Simulation and Modelling (MESM2002), sn, 2002, pp. 183–187.
[27] PTV AG: VISUM, http://www.ptvag.com/software/transportation-planning-traffic-engineering/software-system-solutions/visum/.
[28] OpenStreetMap, http://www.openstreetmap.org.
[29] Tiger Map Service, http://www.census.gov/geo/www/tiger/.
[30] J. Singh, N. Bambos, B. Srinivasan, D. Clawin, Wireless LAN performance under varied stress conditions in vehicular traffic scenarios, in: 2002 IEEE 56th Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall, Vol. 2, 2002.
[31] C. J. Bernardos, I. Soto, M. Calderón, F. Boavida, A. Azcorra, VARON: Vehicular Ad hoc Route Optimisation for NEMO, Computer Communications 30 (8) (2007) 1765–1784.
[32] Iperf, http://sourceforge.net/projects/iperf/.