A Light Architecture for Opportunistic Vehicle-to-Infrastructure Communications

Farah El Ali, Bertrand Ducourthial

A Light Architecture for Opportunistic Vehicle-to-Infrastructure Communications

Farah El Ali
(1) Université de Technologie de Compiègne
(2) CNRS Heudiasyc UMR6599,
Centre de Recherche de Royallieu
B.P. 20529, Compiègne, France
elalifar@utc.fr

Bertrand Ducourthial
(1) Université de Technologie de Compiègne
(2) CNRS Heudiasyc UMR6599,
Centre de Recherche de Royallieu
B.P. 20529, Compiègne, France
ducourth@utc.fr

ABSTRACT

The development of the Intelligent Transportation Systems (ITS) highlights the need of connecting vehicles to the infrastructure. Indeed, many ITS applications rely on such connections to offer new on board services. The networking architecture allowing vehicle-to-infrastructure (V2I) communication is then a key challenge for new pervasive applications.

In this paper, we present an architecture designed for opportunistic vehicles to infrastructure communication. This light architecture allows to transfer data from the vehicles to the infrastructure through IPv4 or IPv6 connections using 3G networks or WiFi access points, depending on their availability. It relies on any VANET routing protocol like geocast or conditional based routing instead of traditional routing. We use conditional transmissions to benefit from its intrinsic discovery facilities, in order to find a gateway towards the infrastructure.

We describe the architecture, its implementation and our road testbeds, allowing to conclude on the interest of such an architecture that allows to exploit already installed networks.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; H.4.m [**Communications Applications**]: Miscellaneous

General Terms

Performance, Experimentation, Measurement, Design

Keywords

V2I, ad hoc communications

1. INTRODUCTION

Context. The Intelligent Transportation Systems are intended to improve the transportation in terms of safety, mobility, impact on the environment, productivity... It is expected that new on-board services will be available; they may contribute to fund the deployment of such a complex system, involving both the road-side (infrastructure) as well as vehicles themselves.

We can separate ITS applications under four families [12]. The first family concerns the *infrastructure oriented applications* such as freeway management, intermodal freight, emergency organization... The second family concerns the *vehicle oriented applications*. These applications give necessary information to the vehicles in order to adapt their behavior for road safety or to diagnostic internal problems for instance. The third family regards the *driver oriented applications*, such as traffic jam alert, upcoming danger warning and so one. Finally, the last family concerns the *passenger oriented applications* such as infotainment services, Internet access [9] or pervasive applications [21].

Hence a large set of ITS applications requires so-called *vehicle-to-infrastructure* (V2I) communications and connecting vehicles to the infrastructure has become a major subject of study. It is however a difficult challenge due to the dynamic nature of vehicular networks and the difficulty (and cost) of a large network access points (AP or gateway) deployment.

Works. Motivated by road safety and infrastructure management, large R&D initiatives were launched in the USA, in Europe and in Japan... Most of them include V2I communications. For instance, the *IntelliDrive* project (formally *Vehicle Infrastructure Integration*) develops V2I communications in order to increase security and limit congestion. The PReVENT project aimed to help the driver avoiding accidents or limit their impact; the sub-project WILLWARN used V2V and V2I communications. The goal of the GST project (Global System for Telematics) was about creating an open standard for on board services [4]. The MYCAREVENT project [27] studied the connexion of vehicles to Internet, where an IP connection is established by the mean of a gateway and that is on multiple communication networks. The Drive-Thru Internet project [3] has investigated the usability of the 802.11 hotspots for offering Internet access to the vehicles. The SAFESPOT project tends to develop a Safety Margin Assistant based among other things, on V2V and V2I communications. The CVIS project (Cooperative Vehicle Infrastructure Systems) also treats the road safety; it includes V2V and V2I communications [2].

From the network protocol point of view, the development of an adequate standard concerns the different international organizations. The IEEE develops the protocol stack

MobiWac'10, October 17–18, 2010, Bodrum, Turkey.

WAVE, including an extension of the 802.11 family protocols for the low layers as well as an alternative to IP in higher layers. The ISO develops the Calm standard for vehicular networks. The IETF works on extensions for IP (Mobile IP, IPv6, Nemo) and auto-configuration in MANET (Mobile Ad hoc NETwork) networks in the Autoconf working group. The car-to-car consortium (C2C-CC) develops and experiments specific protocols for vehicular networks. The ETSI is involved in the harmonization of ISO, IETF, IEEE and C2C standards (ETSI Technical Committee ITS).

Contribution. As we can see, the new ITS applications are leading to new network protocols for V2I communication. Several experiments have been done but the standardization process is not achieved. The integration and the interoperability of the different solutions lead to intense discussions. If the access to the Internet requires IP, its native use in the V2V communication is controversial. The end-to-end IP communication standardizes the network layer but increases the network overhead, and shows real problems for address auto-configuration [10].

Our work deals with the design, the implementation and the test of a light communication architecture for connecting vehicles to the infrastructure. It relies on multi-hops communications between vehicles until reaching an access network.

The main characteristic of our architecture is its lightness. It does not rely on IP for vehicle-to-vehicle communication in the aim of circumvent the address assignment problem and to adapt to any VANET routing protocol. It handles both IPv4 and IPv6 access networks and can use either WiFi hot spots or 3G cellular networks, depending on their availability. We show that such a light architecture is sufficient for collecting data produced by on-board sensors until a server in the infrastructure.

We present the issue of Internet access from vehicles in Section 2. In Section 3, we describe our architecture. Sections 4 to 6 introduce its components. The architecture relies on the Airplug middleware [12]. We use the conditional transmissions [14] as routing protocol. Section 7 reports road experiments. These road testbeds show that our communication architecture is efficient and very suitable for opportunistic communications. We conclude in Section 8.

2. THE ISSUE OF INTERNET ACCESS FROM VEHICLES

In this section, we summarize the main proposed solution to access the infrastructure from vehicles.

WAVE. The IEEE extended its protocol family 802.11 by adding the 802.11p, being inspired for that by the ASTM E213-03 standard, which in turn is based on the 802.11a standard. This protocol modifies the physical and MAC layers in order to adapt to VANET constrains, conform to the DSRC (Dedicated Short Range Communication). In particular, there is no more "association" in order to be able to send messages in dynamic environments. IEEE has also defined WAVE[1] (Wireless Access in Vehicular Environment) or the 1609 protocols family [6]. WAVE specifies a complete protocol stack (1609.0 to 1609.4), relying on 802.11p for the low layers. The 1609.3 standard includes the WSMP protocol (WAVE short Messages Protocol) for inter-vehicle communication, presented as an alternative to IPv6 [5]. In this protocol, messages are routed with an application class identifier (ACID) and an application context mark (ACM) to replace the IP address and the port number [7]. This would ease the communications in dynamic environments.

CALM. IEEE developments are linked to the ISO, specifically the "Technical Committee 204 Intelligent Transport Systems, working group 16, Wide Area Communication" in charge of the medium and short range communication, that works on the Calm[2] (Continuous Air-Interface for Long and Medium range telecommunications) standard [1]. Calm goal is to offer continuous communication in a transparent way to users via a variety of communication networks, such as 802.11, 802.11p, 802.15, 802.16e, 802.20, cellular networks 2G, 3G, 4G and other specific national ITS systems protocols. Calm integrates the IEEE and IETF propositions. Vertical handovers would be mainly handled by IP while horizontal handovers would be left to be handled by the lower layers.

Mobile IP and Nemo. IETF has been working for several years on mobile networks, ad hoc networks, and recently vehicular networks. The vision is a complete deployment of IP, giving each vehicle an IPv6 address.

To deal with the mobility, the Mobile IPv6 protocol is based on the update of a temporary address, called the "care-of address". The mobile node has then 2 addresses, a permanent one related to the original network of the node, and a temporary one related to the visited network. When several on board IP addresses are used, Mobile IPv6 would be inefficient. *Nemo Basic Support* protocol[3] [11], which relies on Mobile IPv6, deals with that issue, while *Nemo Extended Support* [24, 25, 23] studies multi-domiciliation and routing optimizations, without being based on Mobile IPv6.

The Geonet project aims at integrating IP with the geocast routing protocol proposed by the C2C consortium [18, 19]. Geocast routing protocols rely on GPS positions to route messages from vehicle to vehicle.

IP address assignment. The IP address assignment is treated in the "Ad hoc Network Auto Configuration Working Group". The multi-hop ad hoc nature of the vehicle network does not allow the use of address auto configuration protocols like those in RFC 4861 and 4862 [22, 26]. Till now there is no standard for IP addressing to vehicles [10], nor much published papers about that subject. We can brief two of these works. In [16], VANET topology is supposed to be composed of small linear independent convoys. Leaders are chosen among vehicles; they act as DHCP servers. This "distributed DHCP" solution guaranties the address uniqueness in each small convoy, but two distant vehicles can however have the same address. The solution proposed in [8] is based on the C2C-CC architecture and the SLAAC (Stateless Address Auto-configuration) technique, that relies on the NDP (Neighbor Discovery Protocol) signaling to verify the IPv6 address uniqueness (supposing that each node in the LAN can communicate with all others). GeoSAC extends SLAAC to geographical distributed networks by using the geographic routing protocol of the C2C-CC, which al-

[1]The DSRC term 2 to different concepts, from the frequencies range to the kind of applications. The IEEE introduced the term WAVE to clarify the use of the term DSRC [6].

[2]Since 2007, Calm stands for Communication Architecture for Land Mobile (previously, Continuous Air-Interface for Long and Medium range telecommunication).

[3]Nemo stands for *Network Mobility*.

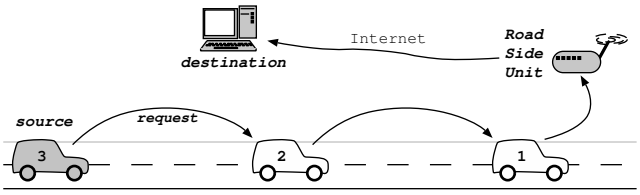

Figure 1: First stage: sending a request to the infrastructure.

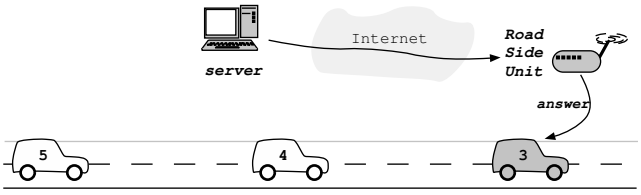

Figure 2: Second stage: fetching the answer.

lows to offer a limited zone of broadcast in order to make the address configuration easier.

3. SCENARIO AND ARCHITECTURE

In this section we present the scenario we consider and the architecture we propose. We then discuss about its use and its advantage.

Considered scenario. As we can see, maintaining communications between vehicles and Internet while using Road Side Unit (RSU) – that is, wireless network equipments along the roads – is not simple. To circumvent this problem, we adopt a strategy based on opportunistic communications in the aim to describe a simple yet powerful architecture . At any time, a vehicle can send requests toward Internet; the request can reach Internet using several hops in the vehicular network (Figure 1). The vehicle then fetches the answer when it is close to an access point, or when the network dynamic is low (Figure 2), allowing it to ask for the answer and then to receive the caching data using the same temporary connection. Note that this architecture can be used with any road scenario (not necessarily a convoy). Indeed, the relay nodes are determined by the conditional transmissions, that are able to adapt to any scenario.

In this paper we focus on the design, implementation and test of the first stage of that scenario, where cars send data to the infrastructure (Figure 1). Besides the general scenario sketched above, the target applications are those that collect data produced by embedded sensors and calculators in vehicles, like positioning, speed, adherence, luminosity, security equipment self-diagnostic, etc. Such information can then be used by the infrastructure-side to manage a truck freight, to determine the traffic conditions, to anticipate in case of danger, to offer new value-added services...

We experimented our architecture by computing the mean speed on a road, using seven equipped cars. Such an application requires to aggregate data in the vehicular network to optimize communications. However such algorithms are out of the scope of this paper, focusing on V2I communications.

Overall architecture. In order to reach a server on the Internet, a classical HTTP connection over TCP/IP is used. Such a connection is done by the vehicle sending packets to the Road Side Unit, which is called *gateway vehicle* (Figure 3). The gateway vehicle is not necessarily the one which has produced the information. Vehicle-vehicle communications do not rely on IP.

An embedded application wanting to send data (APP application on Figure 3) to a web server, contacts its local gateway (GTW), which is a program running on the same vehicle. If this gateway has detected an Internet access using embedded 3G device (if available on this car) or a near WiFi hot spot for instance, it sends immediately the data on the Internet. If not, sending depends on the priority of the data. If the priority is low, the gateway waits for a certain delay, hoping to find soon a WiFi access (or to reach a 3G covered zone). By the way, no message is sent in the VANET, so the bandwidth is preserved. If the priority is high (or the waiting delay is over), the gateway then forwards the data to near vehicles (our experiments show that it is always shorter to forward the packet in the VANET). If one of these cars has an Internet access, its GTW application sends the message to the infrastructure. If not, the message is forwarded from car to car until it reaches an Internet gateway, except if a terminal condition is true (such as the maximal delay or number of hops reached).

Opportunistic communication. We note that it is possible that a gateway is not found in a reasonable delay. In this case, the message will not be sent. Therefore, in some unfavorable cases, the message will not reach the server. Meanwhile, for applications collecting data produced by embedded sensors, a message that is not up-to-date has no interest, and it is more interesting to send a newer and more up to date message (containing data produced more recently by sensors and embedded calculators). In the case of the above scenario (Figures 1 and 2), if the vehicle does not fetch any answer when accessing directly to the Internet, it will be able to resend the request. The reception may then be delayed until the next hot spot. Note that the routing in this architecture is based on conditional transmissions. This routing is described in Section 5.

It is possible to increase the delivery rate to the server by sending several time the request (duplicate packets). On the server side, it is easy to withdraw duplicate requests. The balance between the case where the message does not

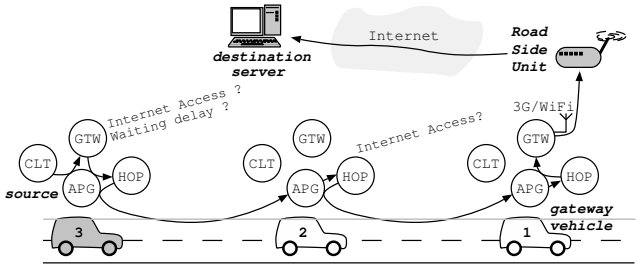

Figure 3: Architecture for connecting vehicles to the infrastructure. APG: airplug program that manages the intra- and inter-vehicle communications. APP: application producing data (eg. by scanning embedded sensors). HOP: VANET routing agent (conditional transmissions). GTW: gateway program, probing Internet access (using 3G if available in the car, or near WiFi hot spot).

reach the server and the case where it reaches it many times depends on some parameters (number of try, frequency of sending...). Placing those parameters is related to the importance we give overloading resources compared to loosing data, knowing that they could or not be sent frequently. Our experiments (Section 7) give indications for determining the adequate values for these parameters.

Advantages. First, such an opportunistic communication scheme limits the number of network equipments along the road as well as the need for 3G connections inside the cars.

Second, vehicle to vehicle communication do not need IP protocols. Only the gateway vehicle (the one sending data on the Internet) will have a TCP/IP connection. By the way, temporary addresses given by DHCP-like servers on the Road Side Unit are sufficient. Indeed, there is no need to manage handovers, which are problematic in such dynamic networks. This scheme fits well with the WAVE protocol stack and the IEEE 802.11p protocol, in which the association step is not a requirement.

Third, any routing protocol specific to VANET can be applied, such as geocast or content-based [13]. We use here the *conditional transmissions* [14] in order to replace addresses by conditions, that determine whether a received message should be transmitted or not to the upper layer and/or nearby vehicles (HOP program in Figure 3). This routing technique allows to avoid the search for destination and to relay addresses in the network. It also has a native service discovery, that will be used to find gateways. Each car will detect the presence of a gateway to Internet through its gateway application (GTW). Conditions will be evaluated at the reception, avoiding any control messages for neighbors discovery (which can be heavy and useless in VANET) to be used. Our architecture then reduces the control in the network. If there is no message to transmit, there is no control messages. The only necessary messages are those required to discover the WiFi hot spot (if any). For instance, with the 802.11p protocol, this could be a WAVE Routing Advertisement, embedded into a WAVE service information element (WSIE), broadcast by the Road Side Unit (RSU).

Finally, a last advantage of this architecture concerns the privacy [17]. One of the things that holds back the development of certain ITS applications, is the driver's privacy preservation, knowing that GPS positions, speed, trajectory among others can be collected. Here, data is not necessarily produced by the gateway car, and without the help of the source, there is no way to distinguish between data coming from the gateway car itself or another vehicle (the source in Figure 3). The IP connection is established between the gateway car and the server and not between the sending car and the server; moreover it uses only a temporary address. Note that, it is still possible to authenticate the source, if it gives sufficient information on a voluntary basis (depending then on the applications).

Components. The architecture realization is described in the following sections. We use the communication middleware Airplug (APG program in Figure 3), dedicated to dynamic networks such as vehicle networks (Section 4). Airplug allows to develop applications in user space [12, 15]. APP refers to an application having data to send on the Internet server (eg. produced by sensors). To deal with the multi-hop communications in the vehicular networks, we use

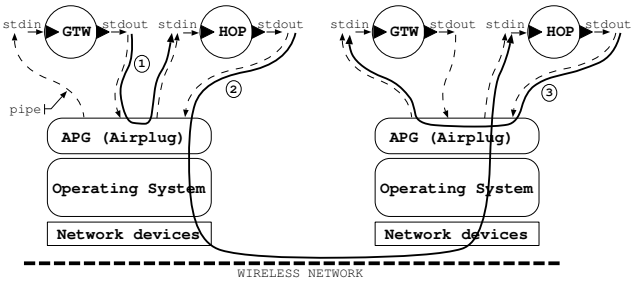

Figure 4: Airplug architecture. (1) and (3) are intra-vehicle communications, (2) is an inter-vehicle communication.

conditional transmissions (Section 5) developed as an Airplug application called HOP [14]. The gateway application called GTW handles sending data to the Internet when it has a WiFi access or an embedded 3G card, and if not, it forwards the message to the local HOP that will search for a gateway to the Internet (Section 6).

4. AIRPLUG MIDDLEWARE

Even if other implementations are foreseen, we have based the realization of our architecture on Airplug, that we describe in this section. Airplug is a light middleware for ad hoc networks [12, 15]. It is characterized by its robustness and its simplicity to organize exchanges of inter- and intra- vehicle messages, which is well adapted to dynamic networks.

The Airplug architecture relies on independent processes; the Airplug core itself (APG in Figure 3) is 0 in user mode for robustness and portability reasons. All communications rely on message passing. A message coming from a given application can be sent to many other applications, remotely or locally. However, by default an application A will only receive messages addressed to it and sent by a local application B (application on the same vehicle). For other receptions, the application must first subscribe to Airplug, indicating that it accepts messages from an application, either local or remote. This registering system (relative confidence locally, and limited confidence remotely) allows an application to control its receptions. It also increases the architecture's robustness by avoiding chained problems in case of bogged applications.

Messages use a specific addressing format, well adapted to dynamic networks. The destination of a message is composed by two fields: an area (local or air) and the name of the destination application. The zone can be intern (LCH for localhost) or external (AIR), which means composed by cars in the neighborhood, or both (ALL). But it can also be more specific (name or address of a nearby vehicle). Note that this addressing scheme is closed to the one in the WAVE Short Messages Protocol (WSMP).

The inter-applications communications are done in the simplest and more robust way possible: by using the standard inputs and outputs. This guaranties a complete independence from the programming language used to develop applications. As Airplug also manages the network interfaces, applications access the network in the same way they do to communicate with other local applications, simply by writing on their standard output.

With Airplug, the development of new communication protocols is done in user mode, in a process that will receive the data to send on its standard input and will handle transmitting them to Airplug via its standard output. Many protocols can be implemented this way, such as routing or transport protocols. The prototyping of new protocols is made easier, as for the cross-layering solutions. Airplug can avoid the protocol stack of the operating system by using raw sockets. Figure 4 details relations between the gateway application GTW and the HOP protocol in our architecture: GTW sends locally towards the local instance HOP (1), that will transmit to the remote HOP instance (2), that will transmit to the remote instance GTW (3).

5. CONDITIONAL TRANSMISSIONS

Conditional transmissions is a kind of routing where logical conditions replace addresses [14]. A message is sent by the module responsible of conditional transmissions with two conditions namely, CUP and CFW (Figure 5). When receiving a message, if CUP is true, the message is transmitted to the upper layer. If CFW is true, the message is forwarded to nearby cars. By dynamically evaluating conditions at receptions, the protocol accommodates better to the dynamic than other protocols relying on addresses (including geographical ones).

All sorts of logical conditions can be used (including conditions testing eventual IP or geographical addresses). But the most interesting conditions deal with distance, duration, trajectory correlation (allowing to determining whether the receiving car follows the sender or not).

Conditional transmissions were implemented as an Airplug compatible application called HOP [15], which has been studied in Network Simulator and tested on the road. For needs of our architecture, we have completed this application to make it accept particular messages that will inform about certain keywords to be considered true while evaluating conditions (these messages will not be accepted unless they come from local applications to the vehicle). This way, the GTW application (present on each vehicle) sends periodically such messages to HOP, to warn about the presence of 3G networks (keyword `3G`) or WiFi hot spots (keyword `3G`).

When a GTW application cannot send the message on the Internet (lack of 3G device or WiFi access point) and cannot wait (due to the message's priority) till it gets near a WiFi hot spot, it forwards the message to HOP with two appropriate conditions (CUP and CFW). The initiator HOP sends then GTW's message along with the two given conditions and some additional information necessary for the condition's evaluation (Figure 5). The CUP condition "$\texttt{wifi} \vee \texttt{3G}$" allows the message's transmission to GTW applications that will actually have an Internet access point. The CFW condition "$\neg\texttt{wifi} \wedge \neg\texttt{3G} \wedge \texttt{dst} < 2000 \wedge \texttt{dur} < 180$" allows for example to forward the message if there is no Internet access found and if the covered distance is less than 2 km range (`dst` stands for distance from the sender) and if the delay is less than 3 min (`dur` stands for duration since the first sending). In this case, the additional information in the message is the date and the source vehicle's position at the moment of the first emission (obtained via the embedded GPS), ensuring that every potential relay will be able to calculate its distance to the sender and the age of the received message.

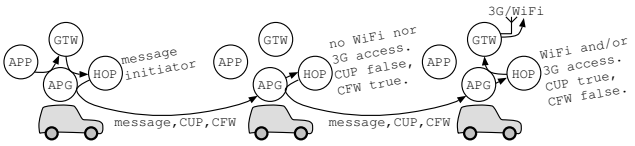

Figure 5: Conditional transmissions (here implemented by the HOP program). A message is sent with the conditions CUP and CFW. When CUP is true, the message is given to the upper layer (here the GTW application). When CFW is true, the message is resent in the neighborhood.

A timestamp forbids any processing of a message that was received before.

While the conditional transmission have not been designed for this purpose, it is important to notice that when using with such conditions, they offer an intrinsic service discovery. It is not necessary to add any pre-processing to search for an Internet access as well as a route towards this gateway. Moreover, it is possible to limit the area covered by the messages by refining the conditions. For instance, conditions related to the trajectory will restrict the area to the vehicles preceding or following the source vehicle (see [14]).

6. GATEWAY

In complement to new functionality added to HOP, a new Airplug application has been developed for the needs of our architecture. This gateway application, called GTW, is in charge of establishing a one hop connection between the vehicular network and the Internet network.

GTW checks periodically the availability of external networks. The networking interfaces that will be used constitutes a subgroup of the interfaces detected, according to some manual or automatic settings. It is actually possible to restrict the choice to the 3G, WiFi access points, or LAN (for the tests in the lab, see next section), to IPv4 or IPv6. GTW informs periodically HOP of the available networks via an intra-vehicle communication by indicating to it the keywords to be evaluated as true when examining the conditions associated to the received messages. This means that if a message is received by HOP with the keyword `WiFi` in a condition, while the local GTW program announced to HOP the presence of a WiFi hot spot, then HOP will replace this keyword by true in the condition.

GTW is also the primary interface for applications willing to send data to the Internet. When an application (APP in Figure 3) wants to send data to a given Internet server, it transmits them to the local instance of GTW (located in the same vehicle), with a priority. If this instance has a connection to the Internet, it sends the data immediately. Else, if the priority is low, it waits hoping to find itself a connection. In the opposite case (priority is high) or when the waiting delay has expired, it forwards the message to the local instance of HOP (located in the same vehicle) that will be in charge of finding a gateway by the mean of the service discovery included in the conditional transmissions.

The GTW application is developed in Tcl/Tk (the initial choice of Tcl/Tk for the Airplug applications is explained by the fact that it is easier to adapt them to Network Simulator later [15]). We did not find any functional IPv6 implementation in Tcl. So, to circumvent this problem, we developed

Figure 6: Experimental platform composed of mini Dell under Linux, external WiFi cards, external antennas and GPS. The Dell on the right has a 3G card (USB connectors).

a Tcl/Tk application launcher in C called launchtk, allowing to add functionality (written in C) to the Tcl scripts thanks to the libtcl. We then developed a C function `send_IPv6` to be used from the Tcl scripts. The GTW application is launched via launchtk, launched itself by Airplug as any other ones [12].

7. EXPERIMENTAL VALIDATION

Testbed. To validate our developments and therefore the architecture we designed, we tested it on real situation. For this purpose, we used PCs (Dell mini-9 Model DP118) under Ubuntu (v8.04 Hardy Heron), see Figure 6. All PCs are equipped with a GPS receiver (Globalsat BU-353 USB) as well as an external WiFi card with USB connectors (Alfa AWUS036EH), allowing to connect an antenna on the roof of the vehicles (D-LINK ANT24-0700, 2.4 GHz, 7 dBi indoor, omni-directional). One PC (gateway car) is equipped with a 3G card (HUAWEI E510) of the SFR operator. The Internet server is an Apache web server in the laboratory, on which we added a specific web page in PHP that saves the data in a file.

The architecture has been validated with experiments in the laboratory as well as with two kind of road experiments (Figure 7). For the tests in the laboratory, the mobility of the vehicles has been emulated, by using logs of real GPS positions, obtained on the road. In the first road experiment, four cars have been used. In the second one, seven cars have been used. The road testbed relied on the 3G network, that only supported IPv4. Tests using IPv6 as well as tests using WiFi hot spots have been done in the laboratory.

In our scenario, the APP application is in charge of collecting and gathering car's speeds in the convoy by means of a distributed algorithm we do not detail here. The data to be sent to the server is then the mean of the cars' speeds in the convoy. Only one car (the one that computed the speeds mean at the end of the distributed algorithm of collect) has to send data to the server. In our scenario, such a car does not benefit from a direct Internet access. Hence, its local GTW application sends the data to the local HOP, that looks for an Internet access in the vehicular network. When the HOP message arrives to the car equipped with an access to the infrastructure, this gateway vehicle sends data to the Internet using an HTTP connection over TCP/IP.

In our experiments, in order to have results depending on the number of hops, we ensures that every message was relayed by each vehicle of the convoy, by giving appropriate conditions to HOP. As a consequence, some layer 2 collisions occurred in the convoy, that affected the performances, but keeping them closer to what could be expected in real VANET networks (non dedicated network). Communication in VANET are done with 802.11 in broadcast mode at 2 Mbit/sec.

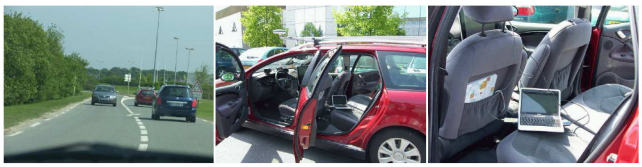

Figure 7: Road experiments used experimental vehicles of the lab as well as standard cars only equipped with the mini Dell PCs.

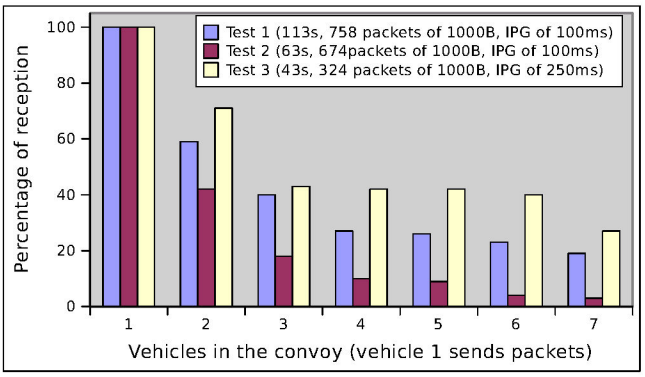

Figure 8: Reception rate per hop in the convoy.

Measures of the delays between cars have been done on the road, by using a combination of GPS time and the hardware clock [15] (frequency of our GPSs is only 1 Hz). Measures of the delay from the gateway vehicle until the web server have been done during the lab testbed, by synchronizing the PC of the gateway vehicle with the web server using Network Time Protocol (NTP). When used in a LAN, NTP can achieve accuracy of about 200 μsec.

Results. Our experiments indicated that our architecture is fully operational and allows to transmit data from vehicles to an Internet server, using several hops in the VANET, using a WiFi or 3G connection, with IPv4 or IPv6, depending on availability. The architecture adapts itself to the networks encountered.

Several experiments have been done. Note that it is difficult to perform similar real experiments on the road (duration, traffic, signal propagation...). When appropriate, we then study long road experiments instead of computing means of several road experiments, realized in very different conditions.

We notice that an inter-packet-gap (IPG) less than 100ms leads to too many collisions and very low performances in the convoy, when using 802.11 in broadcast at 2 Mbit/sec, which is close to results of [20]. These results depend on the modulation (in this case DQPSK). Note however that an IPG of 100ms is sufficient for many applications willing to send reports from in-car data to the infrastructure.

Figure 8 shows the number of messages received on each car during three tests. As we can see, results vary, depending on external conditions on the road. Nevertheless, a trend

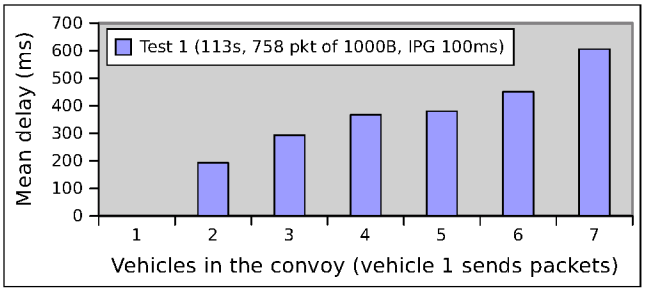

Figure 9: Delay per hop in the convoy.

emerges: one third to one half packets is lost per hop, in real conditions on the road. For comparison, during a similar experiment in the lab, we obtain a loss rate of 19 % for one hop. It is important to note that these are raw measures, without any retransmission techniques, such as those based on acknowledgment, as implemented in standard protocols (layer 2 unicast, transport protocol in layer 4, applicative retransmissions...).

Figure 9 shows the mean delay per vehicle during a long experiment (113s, 758 packets of 1000B with an Inter-Packet-Gap of 100ms). This delay is measured at the application level. It largely depends on the medium congestion and the real propagation conditions on the road (which depends on inter-vehicle distances, trucks and so one). The mean per hop is 1 100ms with a standard deviation of 64. For comparison, we obtain an applicative delay of 58ms for one hop in the lab.

These results confirm that a message with high priority should be forwarded to other vehicles instead of waiting until the vehicle encounters an Internet access point.

For the vehicle-to-infrastructure communication, the TCP/IP protocol was used. The mean delay we observed is equal to 264 ms with a standard deviation of 12.5.

Finally, for a data to be sent, several identical packets may be emitted to ensure a better success of transmission to the server on the infrastructure. Note that ensuring a probability of success of 100 % may lead to several retransmissions in a dynamic network, which could be a bad strategy for sending data produced regularly by some on-board applications. For a given probability of success, more packets lead to less Internet gateways. Figure 10 shows the maximal distance from the sender to the gateway in the vehicular network, for four different probabilities of success, assuming a wireless communication range of 400 m. For example, if we have a gateway each 1000 m, we have to send 3 packets to reach the destination with 70% of success, 4 packets to reach the destination with 80% of success, 5 packets for 90% of success, and 6 packets for 95% of success.

8. CONCLUSIONS

In the context of Intelligent Transportation Systems, many applications require V2I communications. In this paper, we proposed a light communication architecture for connecting vehicles to Internet. It is able to adapt to encountered networks (IPv4 and IPv6, WiFi hot spots and cellular networks). This architecture supports specific VANET routing protocol. We used conditional transmissions [14] for its ability to discover gateways, whatever is the road scenario.

It has been implemented with the Airplug middleware

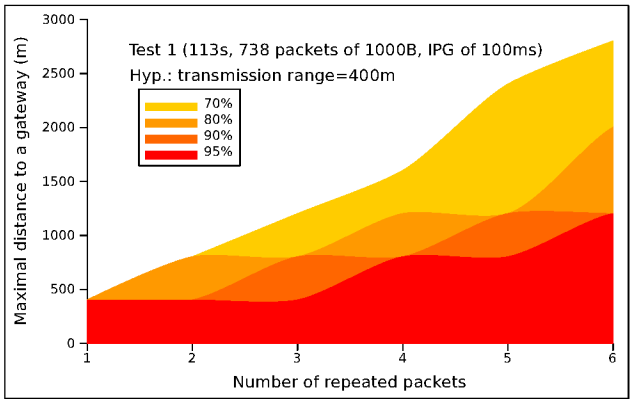

Figure 10: Admissible distance to an AP.

that allowed on-the-road tests using on-the-shelves material. The lab and road testbeds allowed to validate the architecture design and to evaluate its performances. While basic technologies have been used (eg. 802.11 in broadcast at 2 Mbits/sec), measures showed that both delay and packets loss rate are compatible with target applications, mainly collecting data produced by embedded sensors.

These results are impacted by the dynamic of the network. Obviously, the more the network is dynamic, the less the performances are high. Measuring such impact is not trivial, it requires to have a pertinent metric of the network dynamic. Our tests have been done with a convoy of mobile vehicles where the vehicles did not change place inside the convoy. However, the connections were dynamic due to experimental conditions.

Our future works deal with new transport protocols in order to adapt to such opportunistic communications in dynamic networks.

Acknowledgments. We would like to thank Anthony Buisset, Thierry Ernst and Manabu Tsukada for their help, their advice or their expertise.

9. REFERENCES

[1] http://www.calm.hu/.
[2] http://www.cvisproject.org/.
[3] http://www.drive-thru-internet.org/.
[4] http://www.gstforum.org/.
[5] Rita/its, ieee 1609 - family of standards for wireless access in vehicular environments (wave). http://www.standards.its.dot.gov/fact_sheet.asp?f=80.
[6] Amendment to standard [for] information technology - telecommunications and information exchange between systems - local and metropolitan networks - specific requirements - part ii: Wireless lan medium access control (mac) and physical layer (phy) specifications: Wireless access in vehicular environments. http://www.ieee802.org/secmail/doc00268.doc, 05 2004.
[7] Ieee, trial use standard for wireless access in vehicular environments (wave). Architecture, March 2007.
[8] R. Baldessari, C. Bernardos, and M. Calderon. *GeoSAC*-scalabe address autoconfiguration for VANET using geographic networkin concepts. Cannes, France, 2008.

[9] http://www.car-2-car.org/fileadmin/downloads/C2C-CC_manifesto_v1.1.pdf. August 2007.

[10] M. Calderon, H. Moustafa, C. Bernardos, and R. Baldessari. *IP Address autoconfiguration in vehicular networks*, chapter 9 in *Vehicular Networks, Techn., Standards and App.* CRC Press (Taylor & Francis Group), Auerbach, 2009.

[11] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network mobility (nemo) basic support protocol. Technical report, Network Working Group, January 2005.

[12] B. Ducourthial. About efficiency in wireless communication frameworks on vehicular networks. In *Proceeding of the ACM WIN-ITS workshop colocated with IEEE ACM QShine'07*, 2007.

[13] B. Ducourthial and Y. Khaled. *Routing in Vehicular Networks: User Perspective*, chapter Vehicular Networks: Techniques, Standards and Applications, H. Moustafa and Y. Zhang. CRC Press (Taylor & Francis Group), Auerbach, Mars 2009.

[14] B. Ducourthial, Y. Khaled, and M. Shawky. Conditional transmissions: performances study of a new communication strategy in vanet. IEEE TVT, Volume 56, Number 6, pages 3348 – 3357, November 2007.

[15] B. Ducourthial and S. Khalfallah. A platform for road experiments. Proc. of the 69th IEEE VTC2009-Spring.

[16] M. Fazio, P. Palazzi, S. Das, and M. Gerla. Facilitating real-time applications in vanets through fast address auto-configuration. Proc. of IEEE CCNC/NIME, January 2007.

[17] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar. Support of anonymity in vanets, putting pseudonymity into practice. Proc. IEEE WCNC, March 2007.

[18] Geographic addressing and routing for vehicular communications. http://www.geonet-project.eu/.

[19] Geographic addressing and routing for vehicular communications. Geonet leaflet, 07 2008.

[20] Y. Khaled, B. Ducourthial, and M. Shawky. Ieee 802.11 performances for inter-vehicle communication networks. Proc. of the VTC 2005-Spring.

[21] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla. First experience with cartorrent in a real vehicular ad hoc network testbed. In ACM, editor, *VANET MOVE'07*, Anchorage, Alaska, May 2007.

[22] T. Narten, E. Norddmark, W. Simpson, and H. Soliman. Neighbor discovery for ip version 6 (ipv6). Technical report, Network Working Group, September 2007.

[23] C. Ng, T. Ernst, E. Paik, and M. Bagnulo. Analysis of multihoming in network mobility support. Technical report, Network Working Group, February 2007.

[24] C. Ng, P. Thubert, M. Watari, and F. Zhao. Network mobility route optimization problem statement. Technical report, Network Working Group, July 2007.

[25] C. Ng, F. Zhao, M. Watari, and P. Thubert. "network mobility route optimization solution space analysis. Technical report, Network Working Group, July 2007.

[26] S. Thomson, T. Narten, and T. Jinmei. Ipv6 stateless address autoconfiguration. Technical report, Network Working Group, September 2007.

[27] E. Weiss, G. Gehlen, S. Lukas, and C. Rokitansky. Mycarevent - vehicular communication gateway for car maintenance and remote diagnosis. Proceedings of the IEEE International Conference on Computers and Communications, June 2006, pp. 318-323., `http://www.mycarevent.com`.