



Geographic addressing and routing
for vehicular communications
<http://www.geonet-project.eu/>



ICT-2007.6.1: ICT for intelligent
vehicles and mobility services

GeoNet STREP N°216269

D7.1 GeoNet Experimentation Results

DATE	June 28 th 2010
CONTRACTUAL DATE OF DELIVERY TO THE EC	M24 – January 2010
ACTUAL DATE OF DELIVERY TO THE EC	M25 – February 2010
EDITOR, COMPANY	Thierry Ernst, INRIA
WORKPACKAGE	WP7 Experimental Validation
DOCUMENT CODE	GeoNet-D7.1-ExperimentationResults-v1.1
SECURITY	Public

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.0	2010-01-23	First internal draft	Draft	Internal
0,6	2010-02-18	First completely integrated version	Draft	Internal
0,7	2010-02-24	Version for review	Draft	Internal
0,8	2010-03-01	Comments from internal reviews integrated	Draft	Internal
1	2010-03-04	Final version	Final	EC
1.1	2010-06-25	Revised final version (minor editorial changes and cross-referencing GeoNet deliverables)	Final	EC

Name of the coordinating person: Arnaud de La Fortelle, INRIA
E-mail: Arnaud.de_La_Fortelle@inria.fr

Contents

1. Executive Summary	4
2. Structure of the Document.....	6
3. Evaluation Environment	7
3.1 Evaluation Methodology.....	7
3.2 Network Platform Configuration.....	8
3.2.1 Network Topology.....	8
3.2.2 Multicast Configuration.....	12
3.3 Vehicular Platform Description.....	14
3.4 Traffic Hazard Application.....	15
3.4.1 Client-Server Application.....	16
3.4.2 Traffic Hazard Reports.....	17
3.4.3 Monitoring.....	19
3.4.4 Implementation Viewpoint.....	20
3.5 Evaluation Parameters.....	20
3.5.1 C2CNet Layer Implementation.....	20
3.5.2 Network Configuration.....	21
3.5.3 Type of Communication Flow.....	21
3.5.4 Geographic Position and Movement.....	22
3.6 Evaluation Metrics	22
3.6.1 Round Trip Time.....	22
3.6.2 Throughput.....	22
3.6.3 Jitter.....	22
3.6.4 Packet Delivery Ratio.....	22
3.6.5 Hop-Count.....	23
3.7 Evaluation Tool: AnaVANET.....	23
4. Indoor Platform Evaluation.....	28
4.1 Test Scenarios.....	28
4.1.1 ICMPv6 Latency Evaluation Scenario.....	28
4.1.2 UDP Packet Loss Ratio and Throughput Evaluation Scenario	29
4.1.3 TCP Throughput Evaluation Scenario.....	29
4.2 Test Results and Analysis.....	30
4.2.1 ICMPv6 Evaluation.....	30
4.2.2 UDP Evaluation.....	33
4.2.3 TCP Evaluation	34
5. Vehicular Platform Evaluation.....	36
5.1 Test Scenarios.....	36
5.2 Test Results and Analysis.....	38
5.2.1 Distance Evaluation.....	38
5.2.2 Static Test.....	40
5.2.3 Dynamic Test (up to 30 Km/h).....	41
5.2.4 Dynamic Test (up to 100 Km/h).....	42
6. Development Tests.....	44
6.1 Plug-Test at NEC in Heidelberg.....	44
6.2 1st Plug-Test at INRIA Rocquencourt	45
6.3 Plug-Test at HITACHI in Sophia-Antipolis.....	45
6.4 Plug-Test at IMDEA in Madrid.....	47
6.5 2nd Plug-Test at INRIA in Rocquencourt.....	49
7. Demonstration.....	51
7.1 Demonstration at ITST 2009 in Lille.....	51
7.2 Final Demonstration at INRIA Rocquencourt.....	53

8. Conclusion.....	56
Annex A: Contributors.....	58
Annex B: References.....	59

1. Executive Summary

This document is a deliverable of Work Package 7 “Experimental Validation” of the GeoNet project. Its purpose is to report experiments performed during the lifetime of the GeoNet project and to analyse the performance of IPv6 GeoNetworking as implemented in [GeoNetD3.1] following the specification detailed in [GeoNetD2.2]. The GeoNet software implementation is not publicly available and the means for third parties to use the current GeoNet software implementation is reported in [GeoNetD6.1].

The most original part of the work performed during the GeoNet project is the development of the C2CNet layer, the “IPv6 over C2CNet” sub-module part of Module 3A “IP Forwarding” at the IPv6 layer and the C2C-IP SAP linking the IPv6 and C2CNet layers. Functional modules of the GeoNet architecture are shown Figure 1 but the reader is advised to report to [GeoNetD1.2] for a detailed description of the architecture or [GeoNetD8.3] for an overview.

Two versions of C2CNet have been implemented independently (HITACHI implementation and NEC implementation) following the GeoNet specification without seeing the source code from one another. The two implementations of the C2CNet layer have then been integrated by INRIA into an IPv6 stack extended with the “IPv6 over C2CNet” sub-module and SAP C2C-IP to transmit packets between the two layers.

The performance evaluation reported in the present document therefore focuses on these specific parts. Scenarios have been designed to evaluate the transmission of IPv6 packets over the virtual IPv6 C2CNet link, i.e. between a set of GeoNet OBUs and GeoNet RSUs located in a GeoNet domain (see [GeoNetD1.2] for definitions of C2CNet link, GeoNet domain, GeoNet OBU, GeoNet RSU, etc.). The evaluation is not performed on other parts such as mobility support, handovers and multicast because these features are not specific to the IPv6 GeoNetworking architecture proposed by GeoNet, although they are essential for its proper operation.

The evaluation environment is reported in Section 3. An evaluation tool, nicknamed AnaVANET (VANET analyser) has been designed in order to collect and then post-process data used in the analysis reported in this document.

The experimental evaluation has been performed in two phases: first indoor on the table, and second outdoor on vehicles and in realistic live conditions. The performance results obtained with the indoor and vehicular evaluation environment are detailed in Section 4 and 5 respectively. The work reported in these two sections basically complements the emulation work reported in [GeoNetD5.1].

Section 6 reports development tests conducted during plug-test meetings in order to debug, improve and test the proper integration and interoperability of the two C2CNet layer implementations, the IPv6 layer implementation and SAP C2C-IP linking the two layers. The most visible progress in improving the performance of the code were actually made at

such plug-test meetings. The work reported in this section basically complements the conformance tests reported in [GeoNetD4.1].

In addition to be an important achievement of the GeoNet project, the public demonstrations were flagship milestones that helped GeoNet partners to progress in the development and performance of the implementation. As such, Section 7 provides a technical description of the public demonstrations performed at the ITST conference (October 2009, Lille, France) and at INRIA premises (January 2010, Rocquencourt, France).

The reader is expected to be familiar with the GeoNet architecture and specification as detailed in [GeoNetD1.2] and [GeoNetD2.2] or [GeoNetD8.3]. All new terms related to IPv6 GeoNetworking used in this document are defined in the Annex of [GeoNetD1.2] and copied in [GeoNetD8.3].

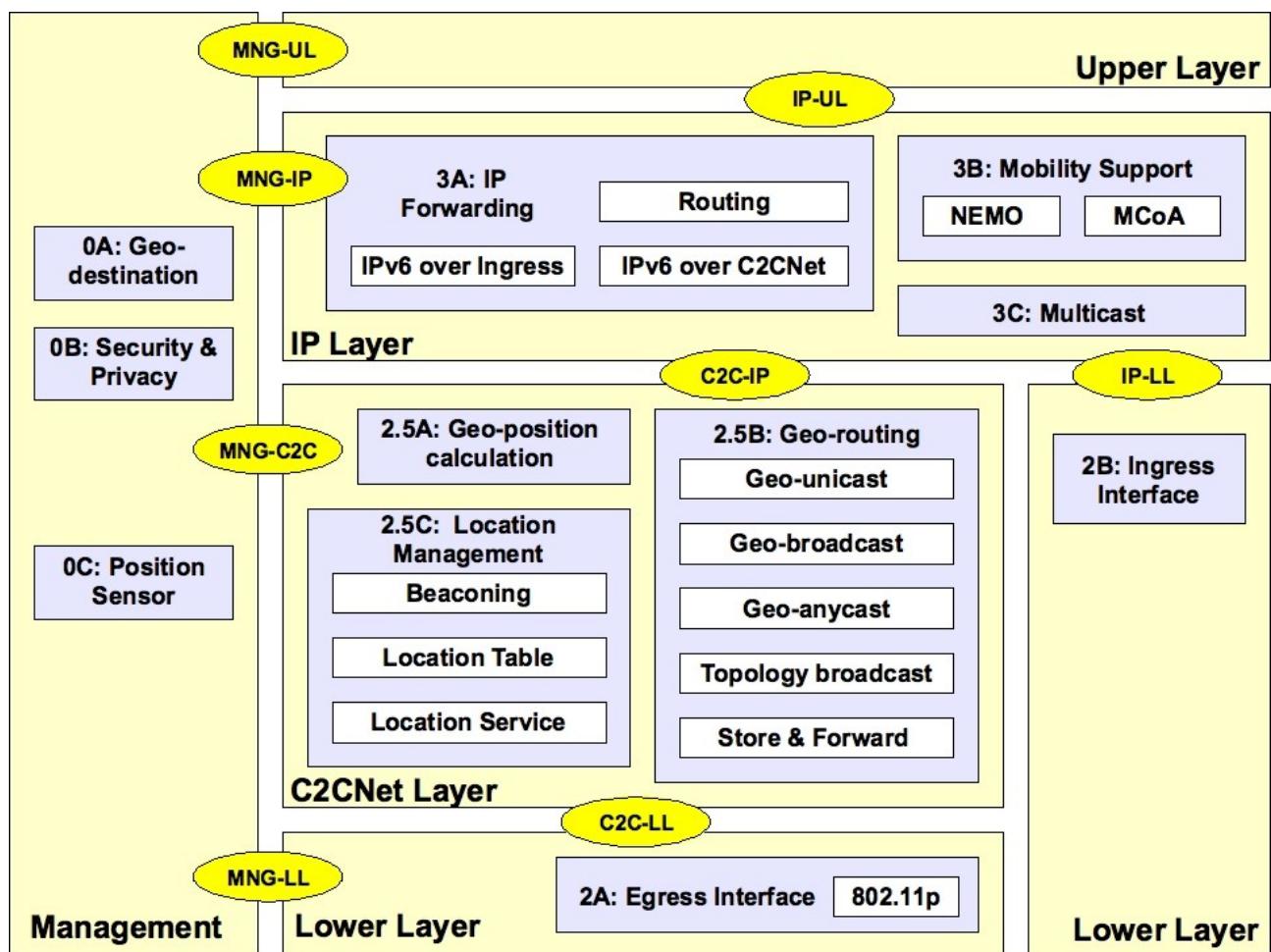


Figure 1: GeoNet Functional Modules [GeoNetD1.2]

2. Structure of the Document

The present document is structured as follows:

- Section 3 describes the methodology conducting the experimentation and metrics for evaluating the results;
- Section 4 describes the tests performed on the in-door testbed and the analysis of the results;
- Section 6 describes the tests performed on the vehicular testbed and the analysis of the results;
- Section 6 describes intermediate tests performed to debug, improve or validated the implementations;
- Section 7 details the two performed public demonstrations;
- Section 8 concludes this document;
- Annex A lists contributors of this document;
- Annex B lists all the references presented in this document.

3. Evaluation Environment

In this Section we present the experimental environment for the evaluation of the performance of IPv6 GeoNetworking. First, we describe our evaluation methodology, then the network configuration, the vehicular platform and parameters and metrics that were considered. Finally the AnaVANET tool which we have developed for such type of evaluation is described. The relation between test parameters and evaluation metrics is shown in Table 9.

3.1 Evaluation Methodology

IPv6 GeoNetworking is evaluated first indoor in confined conditions and then outdoor in real conditions.

The **indoor test environment** is designed to evaluate pure the performance of IPv6 GeoNetworking avoiding interferences due to unexpected radio perturbations and difficulties to trace the movements of the GeoNet OBUs. The test are actually performed on a table without any vehicle as shown in Figure 2. The GPS information is not from actual GPS device but from static position recorded in a configuration file. The advantage of this method is that the same test scenarios can be repeated several times with various parameters.



Figure 2: Indoor Test

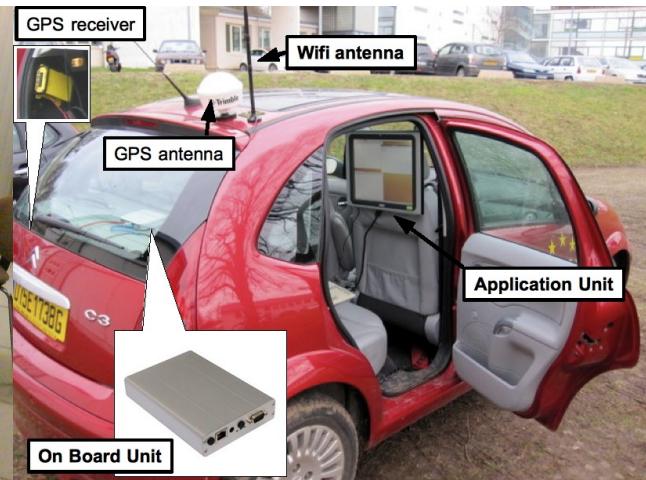


Figure 3: Outdoor Field Test Environment

To evaluate the performance in more realistic scenarios, we setup an **outdoor field test environment** with three vehicles equipped with a GeoNet OBU (IPv6 mobile router), an AU (IPv6 host), GPS receiver and wifi antenna as shown in Figure 3. The topology of the network dynamically changes during the test depending on the location of the vehicles. The performance of IPv6 GeoNetworking depends on the radio propagation which is influenced by obstacles. Network performance also depends on other factors such as the distance, movement of vehicles. We have therefore developed the AnaVANET evaluation

tool (described in Section 3.6) to perform the evaluation taking into account all of these factors.

3.2 Network Platform Configuration

The actual configuration of the platform varies slightly according to the experiment under consideration (for example, not all experiments are using multicast capabilities). However, all used capabilities are configured according to the description given in the following paragraphs.

3.2.1 Network Topology

IPv6 routers: GeoNet OBUs and GeoNet RSUs are Alix3d3 embedded PCs on which Ubuntu 9.0.4 is installed with a Linux 2.6.29.6 kernel. Each PC has one built-in Ethernet port (ingress interface) which is connected to the Ethernet hub connecting other PCs, and a mini-pci wireless card (Atheros AR5414 802.11 a/b/g Rev 01) used as wireless connection to other GeoNet OBUs or GeoNet RSUs. GeoNet OBUs and RSUs are running the complete IPv6 suite of protocols and the C2CNet layer as indicated in [GeoNetD1.2] and implemented in [GeoNetD3.1]. The configuration of GeoNet OBUs and GeoNet RSUs is summarised in Table 1.

Feature	Description
Model	Alix3d3
OS Features	Ubuntu 9.0.4, kernel version 2.6.29.6
Processor Features	Geode(TM) Integrated Processor by AMD PCSi586 CPU 498.128 MHz
Wireless card model	mini-pci wireless card (Atheros AR5413 802.11abg NIC)
Ethernet card model	VT6105M [Rhine-III]
Disk and memory features	4110MB SanDisk SDCFX3-004G247MiB System memory

Table 1: Configuration of GeoNet OBUs and RSUs

IPv6 hosts: Applications Units (AU) are conventional PCs (desktop or laptop) configured with Ubuntu 9.0.4. They are connected to either the GeoNet OBU or GeoNet RSU through the Ethernet hub. IPv6 hosts are running a conventional IPv6 suite of protocols. The configuration of GeoNet OBUs and GeoNet RSUs is summarised in Table 3.

Feature	Description
Model	PAC-1000GB-R20
OS Features	Ubuntu 9.10 , kernel version 2.6.31-17
Processor Features	Intel ® Core(TM)2 Quad CPU Q9650 CPU 2003 MHz
Ethernet card model	82566DM-2 Gigabit Network Connection
Disk and memory features	192GB TS192GSSD25S-M 3015 MB System Memory

Table 2: Configuration of IPv6 Hosts

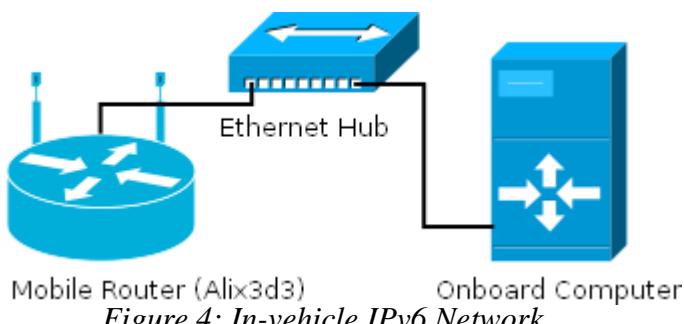


Figure 4: In-vehicle IPv6 Network

Wireless network configuration: Communications between GeoNet OBUs and GeoNet RSUs in the GeoNet domain is performed using the Madwifi driver (version [madwifi-trunk-r4099-20090929](#)) is used and mounted with the following configuration:

- Standard: IEEE 802.11g
- Wireless channel: 3
- Frequency: 2.422 GHZ
- Data rate: 6 Mbits/s

For the indoor tests we used the new EZ-Xtender indoor range extender antennas (all the description is provided in http://www.aerial.net/shop/product_info.php?cPath=35_37&products_id=510). The antennas used in vehicles are 2.4GHz 9dBi indoor OMNI antenna RP-SMA like (the description is provided in http://www.aerial.net/shop/product_info.php?cPath=35_37&products_id=172).

IPv6 address configuration: IPv6 prefixes are configured as indicated in Table 3.

IPv6 prefix	Network	Connected Nodes
2001:660:3013:ca01::/64 (MNP)	IPv6 In-vehicle network	OBU1, AU1
2001:660:3013:ca06::/64 (MNP)	IPv6 In-vehicle network	OBU2, AU2
2001:660:3013:f006::/64	IPv6 C2CNet link	OBU1, RSU1
2001:660:3013:f005::/64	IPv6 C2CNet link	RSU2,OBU2
2001:660:3013:3::/64	Internet	Data Centre, RSU1, RSU2, Gateway

Table 3: IPv6 address configuration

Neighbor Discovery configuration: *radvd* is the software that enables advertising an IPv6 prefix in the GeoNet domain. Below is shown an example of the *radvd* configuration file for RSU1:

```
interface tun0
{
    AdvSendAdvert on;
    MaxRtrAdvInterval 3;
    MinRtrAdvInterval 1;
    AdvIntervalOpt on;
    prefix 2001:660:3013:F005::/64
    {
        AdvRouterAddr on;
        AdvOnLink on;
        AdvAutonomous on;
        AdvPreferredLifetime 60;
        AdvValidLifetime 120;
    };
};
```

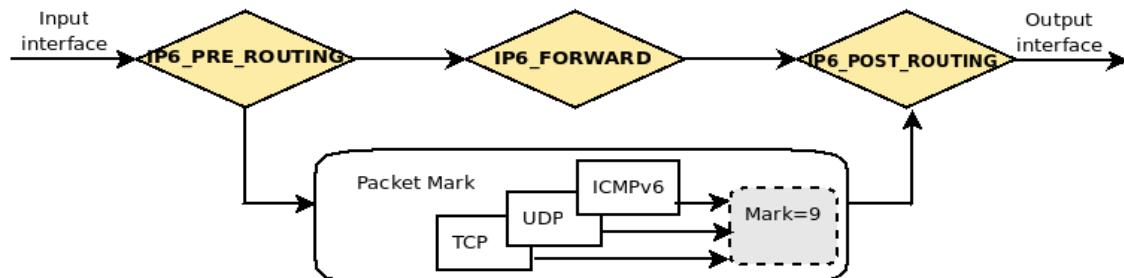


Figure 5: Nefilter

Netfilter configuration: The packet mangling table is used for actual changing of packet information. It is used to mark the packet before routing as illustrated in Figure 5.

Below is presented some commands to configure packet marking:

```
ip6tables -F
ip -6 rule add from $InVehicleNetwork fwmark 0x9 lookup 9 prio 301
ip -6 route add default from $InVehicleNetwork dev $TUNNEL table 9 metric 10 proto 16
ip -6 route add default dev $TUNNEL table 9 metric 10

ip6tables -t mangle -F PREROUTING
ip6tables -A PREROUTING -t mangle -j MARK -p icmpv6 -set- mark 9
ip6tables -A PREROUTING -t mangle -j MARK -p tcp --set-mark 9
```

Routing configuration: The packet should be routed correctly in the GeoNet domain. Every packet received by a GeoNet OBU on the ingress interface from the in-vehicle network should be forwarded to the C2CNet tunnelling interface (tun0). Table 4 shows a sample of the routing table of OBU1 using the `netstat -nr6` command:

Destination	Next Hop	Flag	Ref	Met	Use	If
::/0	::	U	0	10	0	tun0
::/0	::	!In	-1	1	35	lo
2001:660:3013:3::/64	2001:660:3013:f005::ca06	UG	1024	0	0	tun0
2001:660:3013:ca01::/64	::	U	256	0	0	eth0
2001:660:3013:f005::/64	::	U	256	0	1	tun0

Table 4: IPv6 Routing Table After Running the GeoNet Configuration

C2CNet layer configuration: Each GeoNet node belonging to the GeoNet domain has its own C2CNet Identifier. Table 5 shows the C2CNet identifiers attributed to GeoNet nodes.

OBUs/RSU	C2C ID
OBU1	CA01
OBU2	CA06
OBU3	CA03
RSU1	CA02
RSU2	CA05

Table 5: The C2CNet ID Table

In the indoor testbed the GeoNet OBUs are put close to one another. In this case, the three OBUs are in the same wireless range and each OBU can receive the beacons from the others. As a result, at the IP layer, the destination receives redundant packets from both the forwarder (the relaying node) and the sender. To resolve this problem, the

C2CNet layer configuration sets up some variables to filter the reception of the packets by their C2CNet ID as presented below for HITACHI's C2CNet layer implementation:

ENABLE_ID_FILTER	YES
C2CID_TO_FILTER	CA01
ENABLE_INROOM	YES
INROOM_DISTANCE	250

The **location table** is maintained by each GeoNet node at the C2CNet layer in order to save positioning information about the neighbours with which it exchanges beaconing frames. This table is refreshed every laps of time when new neighbours are detected and disappearing neighbours are purged after 5 seconds. The position information used was statically configured in each OBU because GPS positioning could not be used in the indoor-test. This is shown on Table 6.

OBUs	List of neighbours	Positions of neighbours (lat,long)
OBU1	OBU3	(48.4,10.000)
OBU2	OBU3	(48.4,10.005)
OBU3	OBU1;OBU2	(48.4,10.000); (48.4,10.003)

Table 6: C2CNet Location Table

GeoDestination configuration: IPv6 multicast addresses are used to specify the GeoBroadcast radius for transmission at the C2CNet layer. IPv6 multicast addresses are attributed statistically and embed a GeoDestination ID (see [GeoNetD1.2] approach 5) which is bound to a geographical area at the C2CNet layer, as indicated in Table 7:

	The multicast address	The Area radius (m)
GeoDestination ID 1	FF0E:0:0:0:0:0:1	500
GeoDestination ID 2	FF02:0:0:0:0:0:2	1000

Table 7: Binding Between IP Multicast Address and GeoDestination

3.2.2 Multicast Configuration

Multicast configuration: Multicast capabilities are configured on the 2.6.29 version of the kernel. MLD (Multicast Listener Discovery) [GeoNetD2.2] is used by routers to discover the presence of Multicast listeners. GeoNet IPv6 OBUs and GeoNet RSU have to run the multicast daemon which enables multicast forwarding and records the multicast groups on their ingress interfaces. All IPv6 hosts (AUs) that must send or receive IPv6 multicast packets must on their side be configured to join the multicast group used in GeoNet applications (I.e FF0E::2 for the road hazard application described in Section 3.4 and used in demonstrations). Multicast groups are configured statically by the application which triggers the joining process when it is started. At boot-up, the nodes join the default

multicast groups which are the *all-node multicast group* on each interface (FF02::1) and the solicited-node multicast group (FF02::interface-id) on each interface where Neighbor Discovery is performed. Joining is performed by means of MLDv2 Listener report of the multicast addresses is sent to the GeoNet OBUs or GeoNet RSU by in-vehicle IPv6 hosts or roadside IPv6 hosts, respectively. The multicast network topology (mostly used for the final demonstration described in Section 7.2) is illustrated on Figure 6.

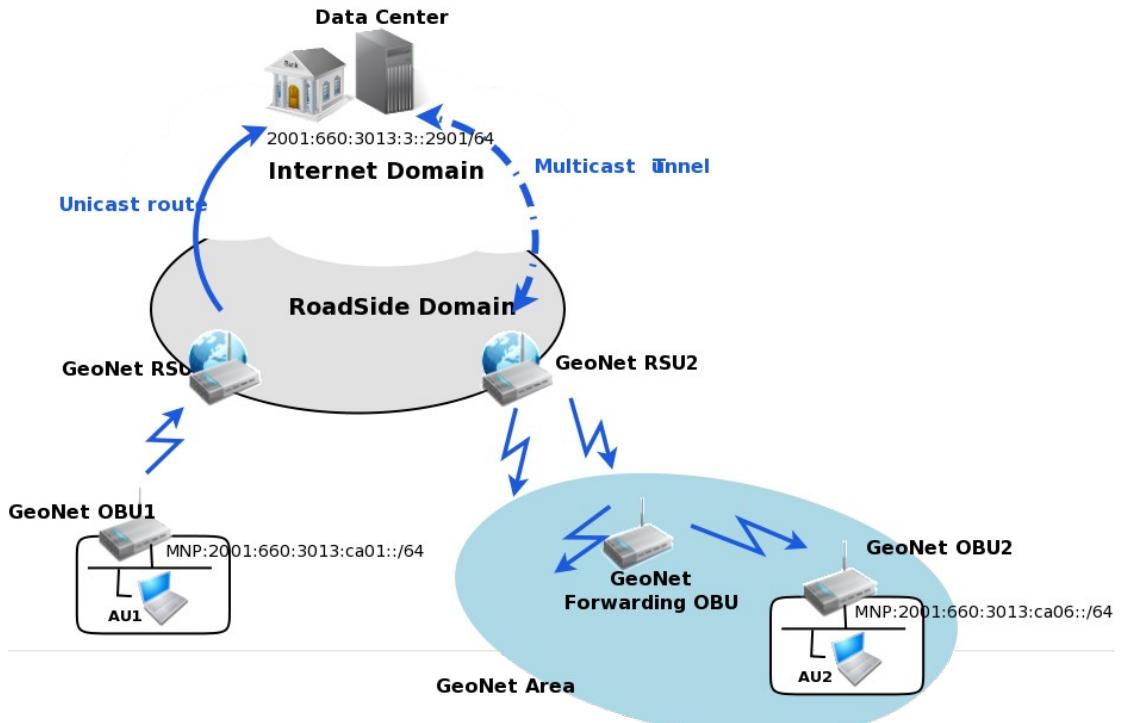


Figure 6: Multicast Network Topology Configuration

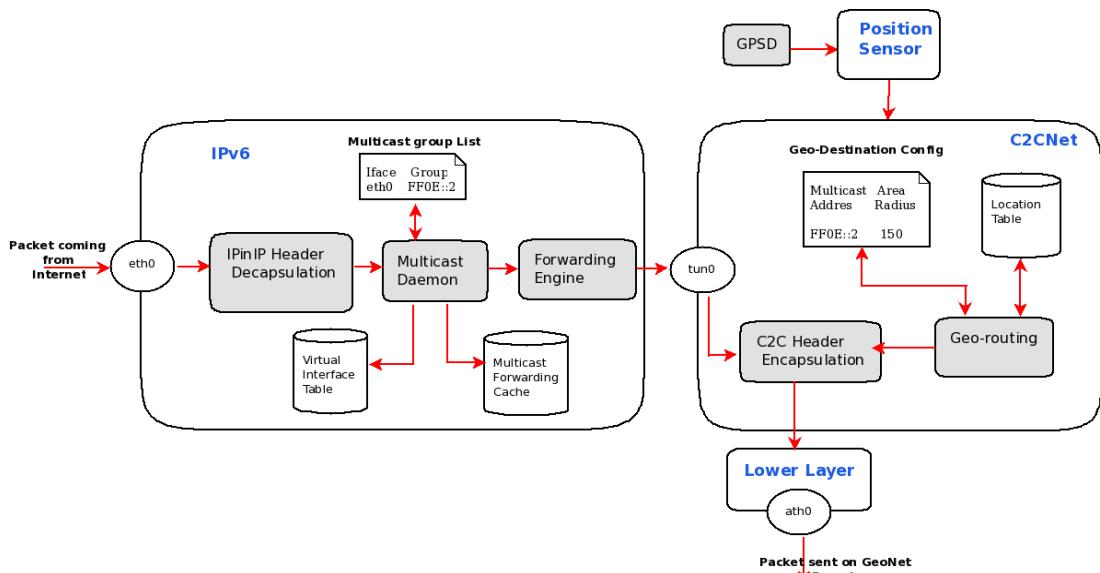


Figure 7: Configuration of the Multicast-Enabled RSU in the Vehicular Platform

Multicast configuration of the GeoNet RSUs: Figure 7 shows the configuration of multicast capabilities on GeoNet RSU2 (serving the GeoDestination). The multicast group list is set statically on RSU2 since there isn't any attached host to which groups shall be reported nor is multicast proxying enabled for the tests (so this is a simplified configuration of multicast compared with the GeoNet specification - see [GeoNetD2.2] Section 11.9 for more information about multicast proxying and Section 11.4.1. for information about the Virtual InterFaces table (VIF) and the Multicast Forwarding Cache (MFC) structures). The daemon updates the MFC and delivers the packets to the forwarding engine which forwards it on the tun0 interface in order to deliver packets to the C2CNet layer. An entry in the MFC of the GeoNet RSU is statistically configured with the multicast group address corresponding to the GeoDestination (ff0e::2, see Table 7), the multicast source (the Data Centre address), the incoming interface (C2CNet/tun0) and the outgoing interface (eth0).

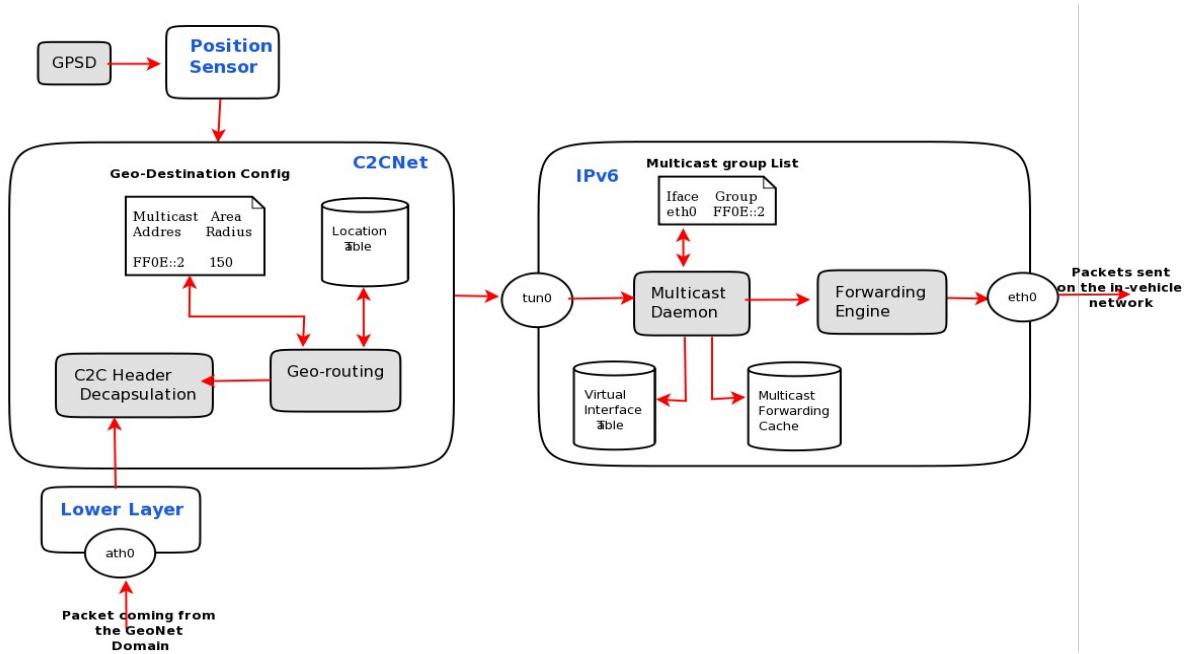


Figure 8: Receiver's Internal Multicast Configuration

Multicast configuration of the GeoNet OBUs: Figure 8 shows the configuration of multicast capabilities on GeoNet OBU2 (receiver in the GeoDestination). MLDv2 is running on MNPs (AU2) which report the groups to which application wants to subscribe. The GeoNet OBU queries MNPs by performing MLD router part on its ingress interface and maintains a list of multicast groups reported by its MNPs.

3.3 Vehicular Platform Description

INRIA owns a test fleet of four conventional vehicles (Citroën C3 shown on Figure 9) with a communication and perception architecture built for React and Com2React European projects. These vehicles are used for the IPv6 GeoNetworking tests performed on the outdoor evaluation platform (Section 5) and the final public demonstration (Section 8).

Each vehicle is equipped with communication capabilities introduced in Section 3.2 and specifically designed electrical cabling. Vehicles are also equipped with a CAN Gateway and perception capabilities not used in GeoNet experiments since the focus of the GeoNet experimentation is put on IPv6 GeoNetworking. INRIA also owns a fleet of Cycabs, driverless vehicles used in the CyberCars2 project that could also be used to extend this test platform in the near future, if appropriate.



Figure 9: INRIA's Fleet of Conventional Vehicles Used in the GeoNet Experiments and Demonstrations

3.4 Traffic Hazard Application

The GeoNet project is gathering partners with networking expertise only. It is not the purpose of GeoNet to develop at the application layer. However, for the purpose of the demonstration, a realistic application is needed. The first intention of GeoNet partners was to use applications developed by other European projects, particularly the project selected as the target platform, i.e. CVIS. The Enhanced Driver Awareness (EDA) application developed by CVIS was the most attractive one. This application requires support of the so-called "ITS facilities" which is provided by an additional layer between the transport and application layers. Its use with IPv6 GeoNetworking would require a few modifications so that the application sends to an IPv6 multicast group. However, as reported in [GeoNetD6.1], porting to CVIS could not be achieved from a technical development view point and would require quite some effort to integrate IPv6 GeoNetworking within the CVIS communication platform given the differences in the version of the used Linux kernel. It was thus decided to limit the cooperation between CVIS and GeoNet to the production of a specification that would push forward the integration of IPv6 GeoNetworking into the CALM architecture. As a result, we have decided to develop our own application, similar to EDA, but without the support of the ITS facilities.

Three hazards are reported: an accident, an approaching emergency vehicle, and a congestion. The former two are directly reported by a moving vehicle (Green) to the road

traffic centre while the latter is sent on demand by the traffic centre operator (in a real situation, the report would be sent based on fusion of information received at the traffic centre). There are no sensing capabilities in the reporting vehicle, so these reports are sent automatically as soon as the vehicle reaches a pre-defined location on the INRIA campus (based on GPS position).

Traffic hazards are sent periodically for a certain period of time (every 30 seconds) from the road traffic centre to a target geographic area (GeoDestination). IPv6 multicast packets are sent from the road traffic centre to a RSU serving the GeoDestination where packets are GeoBroadcast at the C2CNet layer. The size of the GeoDestination is about two times the radio range of the RSU. A static vehicle (Black) in the radio range of the RSU receives the packets and continue to GeoBroadcast them at the C2CNet layer. A third vehicle (Red) not in the radio range of the RSU is moving in and out of the GeoDestination. Traffic hazards are received and displayed when the vehicle is located in the GeoDestination.

As shown on Figure 6, multicast capabilities are set up between the Data Centre and the vehicles within the area monitored by the data centre. GeoNet RSU1 - which provides Internet connectivity to the GeoNet OBUs - GeoBroadcast the road traffic hazard notification to that area. Multicast packets sent from Internet peer nodes (e.g. Data Centre) to the RSU are indeed encapsulated into unicast packets because not all the routers are multicast-enabled on the whole path between the Internet peer nodes and the RSU. The GeoDestination radius was set to 150m. RSU1 was the centre of this area. As explained in Section 5.1, the wireless range is about 130 meters in normal urban conditions including obstacles. The GeoNet forwarding vehicle was placed about 100 meters from RSU1 and the GeoNet receiving vehicle was placed to about 140 meters from RSU1 before moving inside and outside the GeoDestination area.

The multicast group (FF0E::2) is statically configured in the application.

3.4.1 Client-Server Application

The application consists of two types of entities, a server (IPv6 road traffic centre in the Internet) and clients (IPv6 host in moving vehicles):

- The server aggregates road traffic hazard information reported by a client vehicle and propagates them to other client vehicles using a UDP multicast socket.
- Client: As a sender, the client software automatically detects road traffic hazards and reports them to the server using the generic UDP unicast socket. As a receiver, the client software gets road traffic hazard messages from the server through IPv6 multicast and displays them to vehicle passengers.

In order to evaluate the efficiency of GeoBroadcast, the software has a mechanism to display vehicles' position, the GeoDestination and the description of the road traffic hazard information on the map.

The client detects a particular traffic hazard and reports the description of the event to the server. The description consists of an event's name, type and its geographic position. The client-side user interface consists of three types of windows:

- 'Basic information' window: shows the vehicle's name and its current position;
- 'Status' window (*sender window*): shows the status of the traffic event detection;
- 'Message form centre' window (*receiver window*): shows the road traffic hazard information notified by the server.

3.4.2 Traffic Hazard Reports

The application supports two modes of propagation of the road traffic hazards.

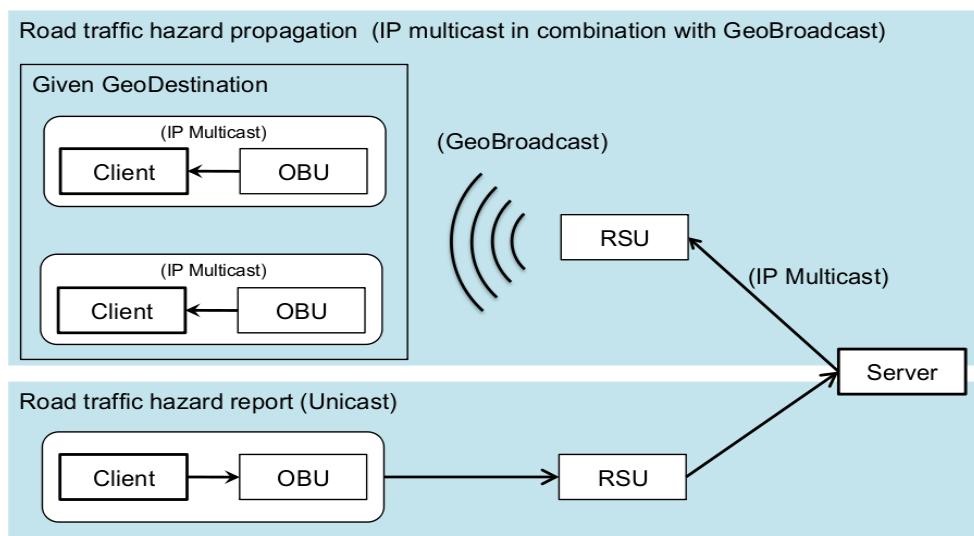


Figure 10: Automatic Reporting (From Vehicle to Server Then From Server to Vehicles)

Propagation of road traffic hazards detected by the client: a client vehicle detects/reports the road traffic hazards to the server. After that, the server propagates it to client vehicles in the target geographic area (GeoDestination). The road traffic hazard detection mechanism itself is out of focus of GeoNet, so perception features on the vehicles are not used and the emission of the road traffic hazard is triggered by the vehicle entering a specific geographic area. The position of the hazard is thus pre-configured and automatically reported by the application when the vehicle enters the pre-defined geographic area. The client periodically calculates the distance from its current position to the hazard's pre-configured position. When the client reaches a radius of 15m from the hazard's position, the client displays an event detection message in the *sender window* and immediately reports to the server the description of the hazard and repeats it periodically. The client's position is provided by the *gpsd* software module from the GeoNet

OBU. After going out from the pre-configured distance, the client stops sending the report and erases the road traffic hazard event from the *sender window*. The report is sent to the server using a generic UDP unicast socket. The GeoNet functionality is not used for this report mechanism. For demonstration purposes, we have defined two traffic hazard events, 'Traffic accident' and 'Approaching Emergency Vehicle'. Figure 4 shows the traffic hazard detection message in the client-side user interface. Figure 10 shows the operation of this mode (automatic mode from vehicle) and Figure 11 shows the road traffic hazard event has it is displayed in the reporting vehicle.



Figure 11: Traffic Event Detection in the Sender Window

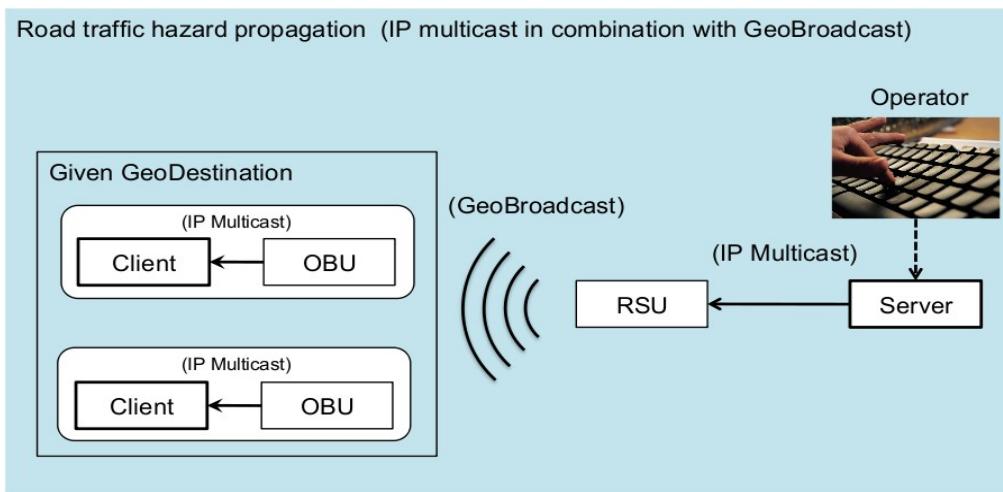


Figure 12: Manual Mode (From Server to Vehicles)

Propagation of road traffic hazards from the server: in addition to the hazards report received by clients, an operator of the server, who takes the role of the operator of an ITS road traffic centre, can manually generate road traffic hazards. The server-side user interface is basically only a button to generate one type of road traffic hazard notification:

report about congestion occurring in a specific geographic area. The server sends all these road traffic hazard to a pre-defined geographic area (GeoDestination) using multicast UDP socket whereas unicast UDP socket is used for receiving hazard reports from clients. After the reception of clients' report/operation of the server's button, the server sends messages to clients periodically for a pre-configured duration (30 seconds). Figure 12 shows the operation of this mode (manual mode from server).



Figure 13: Client-Side User Interface

Reception of road traffic hazard reports: Whenever the client receives a road traffic hazard report from the server, it shows the event's description to the receiver window. 5 seconds after the reception of the message, the receiver window is automatically cleared. The message is received using generic UDP multicast socket. The client sends the generic unicast packet to the server and it receives the generic IP multicast packet from OBU. Figure 13 shows the road traffic hazard event has it is displayed in the receiver window.

3.4.3 Monitoring

As shown on the screen shot on Figure 14, all the vehicle positions, the boundary of the GeoDestination area, and the position of the traffic event reported by the client are shown on a dynamic map. For the demonstration, we use Google Earth with the netlink mechanism of KML. The server has a local http server that generates KML files containing the position information. The position information, which is demonstration purpose only, is only accessed by the server locally.

In order to show that the road traffic hazard reports are only delivered to clients in a particular geographic area, the client application provides the functionality to show the border of the GeoDestination area and the vehicle's current position in the map. The application has a local http server that provides vehicle's current position and information about the GeoDestination area.



Figure 14: Screen Shot of the Map View

3.4.4 Implementation Viewpoint

The application is implemented using Java SE6 in combination with JavaFX 1.2. JavaFX is only used to make the graphical user interface.

The application itself does not have GeoNet functionalities (a real deployment of such application would require the application to be GeoNet-aware but this is not supported yet in GeoNet so the application was statistically configured).

3.5 Evaluation Parameters

3.5.1 C2CNet Layer Implementation

The GeoNet project has produced two independent implementations of C2CNet layer (HITACHI implementation and NEC implementation), and one of the IPv6 over C2CNet module (Module 2A) and the C2C-IP SAP (see [GeoNetD1.2]). C2CNet developers implemented it independently without seeing the source code each other from same GeoNet specification. The interoperability between the two proves that there is a common

and correct understanding of the specification. Both implementations of the C2CNet layer have been utilised in the experimental performance evaluation.

3.5.2 Network Configuration

The performance depends on the network configuration, particularly the number of hops that packets are transmitted through between their source and destination. Thus we can get the best network performance of IPv6 GeoNetworking when the GeoNet OBUs are directly connected (single hop). In contrast, multi-hop configuration adds transmission delay and processing delay. The evaluation considers both **single hop** and **multi-hop** cases.

3.5.3 Type of Communication Flow

UDP, TCP and ICMPv6 are used to measure the network performance of IPv6 over C2CNet between two communication end-nodes (AU to AU):

- **UDP** is a unidirectional transmission flow from the source to the destination communication end-node. The traffic is generated by *iperf*. The tool can configure sending packet size and sending rate. Metrics under consideration are packet loss rate, throughput and jitter.
- **TCP** is a connection based communication flow established between the two communication end nodes. The traffic is also generated by *iperf*. The tool can configure TCP window size and maximum segment size. The metric under consideration is the throughput.
- **ICMPv6** is a bi-directional communication flow between the two end-nodes. The traffic is generated by *ping6*. The tool can configure packet size and sending interval. Metrics under consideration are RTT and packet loss rate.

Type of Communication	Configurable parameter	Evaluation Metric
UDP	Packet size, Sending rate	Packet loss rate, throughput, Jitter
TCP	TCP window size, Maximum segment size	Throughput
ICMPv6	Packet size, send interval	RTT, Packet loss rate

Table 8: Evaluation Parameters and Metrics for Communication Flow

3.5.4 Geographic Position and Movement

Network performance also depends on other live factors, for example, the distance, movement of vehicles, speed, obstacles and so on. These are taken into account using AnaVANET evaluation environment (Section 3.6).

3.6 Evaluation Metrics

3.6.1 Round Trip Time

The round trip time is measured using ICMPv6. An IPv6 AU sends the ICMPv6 echo request to the other IPv6 AU. The AU receives the request and replies with an ICMPv6 echo reply. On the sender side, the time between sent request and received reply is measured. *ping6* shows RTT as a standard output.

3.6.2 Throughput

Throughput is measured in UDP and TCP. Both are taken with the *iperf* tool. In UDP, *iperf* is executed in both the sender and the receiver, then the sender sends UDP to the receiver with fixed rate. The sender is not able to see the result because the communication is unidirectional from the sender to the receiver. The throughput is shown on the receiver side.

On the other hand, the sending rate is automatically adjusted with TCP's congestion control mechanism. The sender manages the sending rate depending on the acknowledgement messages received from the receiver. The throughput appears in both the sender and receiver.

3.6.3 Jitter

Jitter is a measure of the variability over time of the packet latency across a network. A network with constant latency has no variation (or jitter). Packet jitter is expressed as an average of the deviation from the network mean latency. The value is displayed in *iperf* using UDP. Our AnaVANET (Section 3.6) computes it.

3.6.4 Packet Delivery Ratio

The packet delivery ratio is the percentage of packets arrived at the receiver divided by packet sent by the sender. *iperf* shows it in the receiver side. AnaVANET described in Section 3.3 calculates the packet delivery ratio on the link between GeoNet OBUs as well as between an AU to the other AU.

Configurable parameters	Elements		
Implementation	HITACHI		NEC
Test environment	Indoor	Outdoor (Geographic position and movement)	
		Distance	Speed Obstacle
Network configuration	Single hop		Multi-hop
Communication flow type	UDP	TCP	ICMPv6
	Packet size, Sending bandwidth	TCP window size, Max segment size	Packet size, send interval



Evaluation Metric	Packet loss rate, throughput, Jitter, Hop count	Throughput	RTT, Packet loss rate, Hop count
-------------------	---	------------	----------------------------------

Table 9: Evaluation Parameters and Evaluation metrics

3.6.5 Hop-Count

Network performance depends on the number of hops in the GeoNet domain. Thus we can get most basic network performance of IPv6 over C2CNet when the OBUs are directly connected (single hop). In contrast, multi-hop configuration adds transmission delay and processing delay.

3.7 Evaluation Tool: AnaVANET

AnaVANET is a tool developed internally at INRIA to analyse vehicular networks. It has originally been used to evaluate OLSR-based ad-hoc vehicular networks [Santa2009]. For the purpose of evaluating the performance of IPv6 GeoNetworking, AnaVANET is extended in order to analyse IPv6 packets transmitted with a C2CNet header in the GeoNet domain.

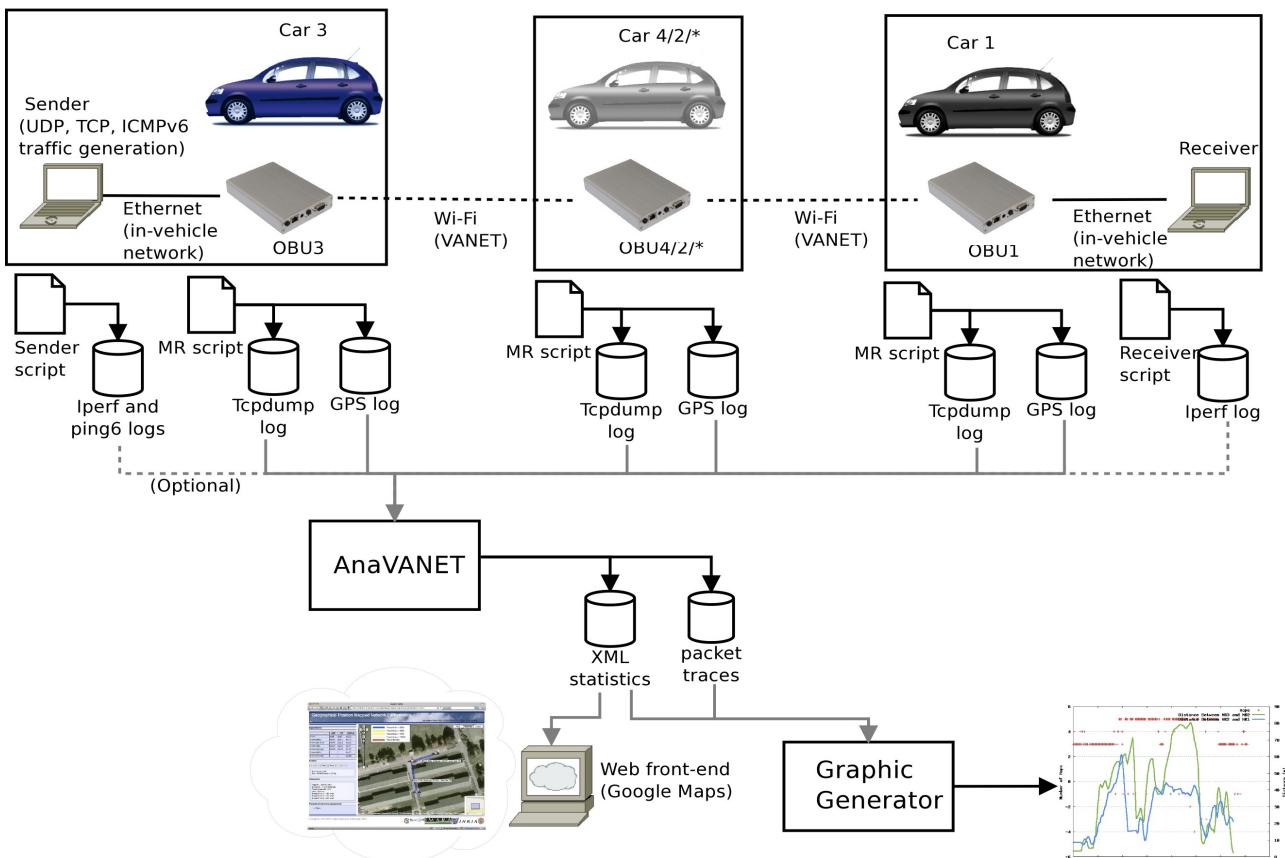


Figure 15: AnaVANET: Overview of Packet Processing and Analysis

Figure 15 provides an overview of the experimental evaluation process carried out in the tests. The Sender (AU) is in charge of generating data traffic, and both the sender and the receiver save a high level log, according to the application used to generate network traffic. All GeoNet OBUs record information about forwarded data packets by means of the tcpdump software, and log the vehicle position continuously. All this data is post-processed by the AnaVANET software and then analysed. A Java application traces all the data packets transmitted from the sender node. This way, it is possible to detect packet losses and calculate statistics for each link and end-to-end, and merge all these per-hop information with transport level statistics of the traffic generator. As a result, AnaVANET outputs an XML file with statistics of one-second periods, and a packet trace file with the path followed by each data packet.

The data processing system is composed of four levels as shown on Figure 16:

- The **network set of applications** through which we can change different variants of performance. In these tests, we use *iperf*, *tcpdump* and the *gpsgetter*.

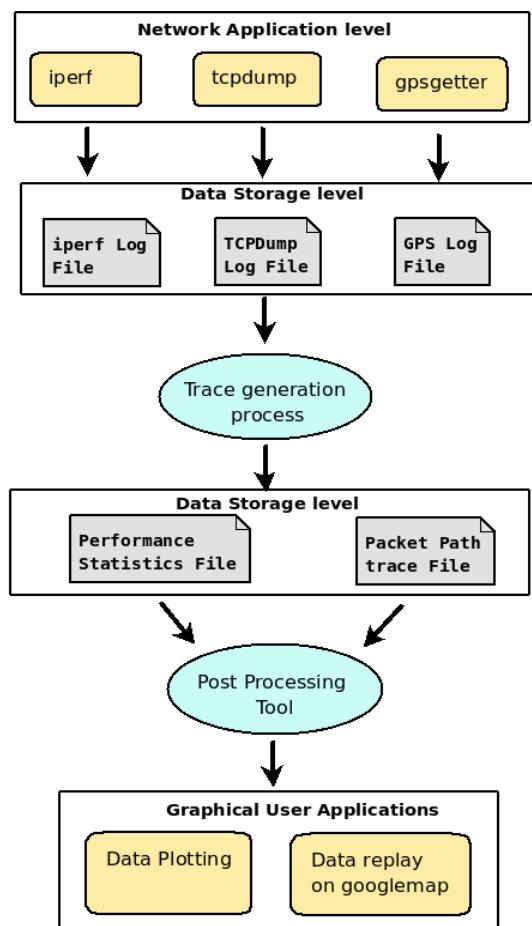


Figure 16:: Performance Data Processing in AnaVANET

XML and packet traces to process results and then creates several types of figures through the *gnuplot* utility.

GPS positioning: GPS positioning is of particular importance for GeoNet tests. The GPS daemon software (*gpsd* [version 2.37-7](#)) is used to get GPS positioning from the serial ports (ttyS0), and to process it. The following options were set at the BIOS level in order to enable *gpsd*:

Advanced Chipset Features
UART Port A [3F8/IRQ4]

Below, is presented a sample of the log file containing native GPS positioning:

```

TIME:00:00:11:959630352
TIME:00:00:11:987516143
$GPGGA,151608.00,4850.25790414,N,00206.08270980,E,1,06,2.4,139.747,M,47.280,M,,*
TIME:00:00:12:019653854
TIME:00:00:12:051613542
$GPVTG,123.8,T,,,000.06,N,000.11,K,A*4E

```

```

TIME:00:00:12:083010869
$GPGSA,A,3,23,17,13,20,31,32,,,,,,3.5,2.4,2.5*31
TIME:00:00:12:111996557
TIME:00:00:12:143692113
TIME:00:00:12:171519421
$GPRMC,151608,V,4850.257904,N,00206.082710,E,000.06,123.8,260110,4.0,W*4A

```

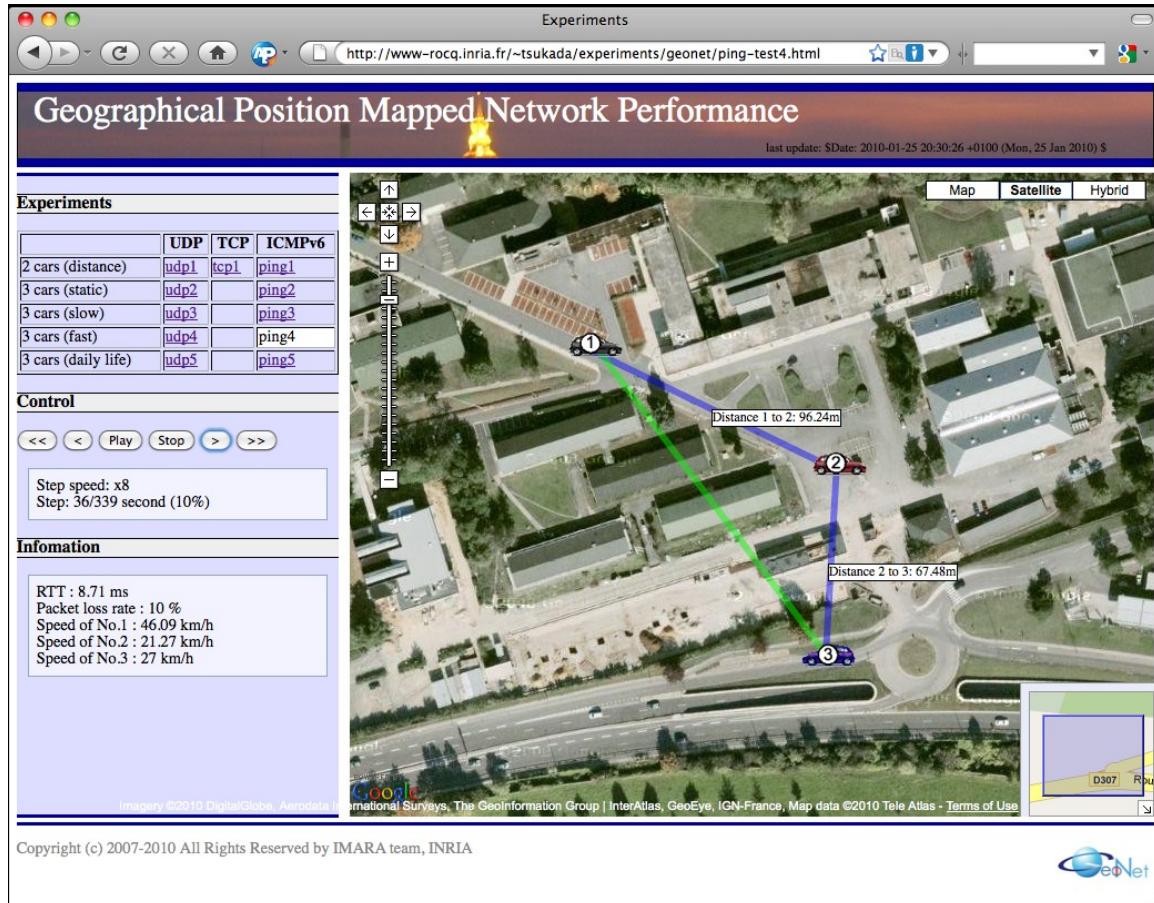


Figure 17: Screen Shot of the Web Site

Use of a network sniffer: *tcpdump* is used to record network communication traces between vehicles. These traces are processed later. Some pointers are retrieved through the log files of GeoNet OBUs in order to analyse the traces and generate graphics. The *tcpdump* result file is parsed in order to facilitate the analysis of the network traces by the post-processing tool. In case of a multi-hop test with three GeoNet OBUs, the file presents a set of entries as follows:

(*time, [Mac src1, Mac dest2]; [Mac src2, Mac dest3], IP src1, srcPort number, IP dest3, dest Port number, protocol identifier, packet size*)

The experiments carried out are available on the GeoNet web site¹ and can be replayed to see the momentary performance of the network during the tests. Figure 17 shows a screen

¹ <http://www.geonet-project.eu/demonstration/geonet/>

shot of the web site. All the experiments can be selected and main performance metrics can be monitored at any time. Users can play and stop at any arbitrary point of the test with the control buttons on the left side of the page. The player speed, one step forward and one step backward are also implemented. On the map, the position and movement of the vehicle are depicted with the speed of each vehicle and the distance between them. The transferred data size, bandwidth, packet loss rate, round-trip delay time and jitter, for each link and end to end are displayed. The network performance is visualised by watching the width of links and the colours used to draw them.

4. Indoor Platform Evaluation

In this section we evaluate the performance of IPv6 over C2CNet in the GeoNet domain using the indoor platform described in Section 3. Tests scenarios are described in Section 4.1 and results are presented and analysed in Section 4.2.

4.1 Test Scenarios

As shown on Figure 18, the scenarios of the tests are performed on GeoNet OBUs to evaluate the latency in the case of ICMPv6 traffic and the packet delivery ratio and the throughput when transmitting UDP packets. An IPv6 AU which runs a script that evaluates several performance parameters is attached to each GeoNet OBU. We evaluated single and multi-hop communications in the GeoNet domain. In the single hop tests, GeoNet OBU1 and GeoNet OBU2 are IPv6 routers directly reachable. In the multi-hop test, the GeoNet forwarding OBU is a GeoNet node that relays the message sent by IPv6 AUs without involving the IPv6 stack.

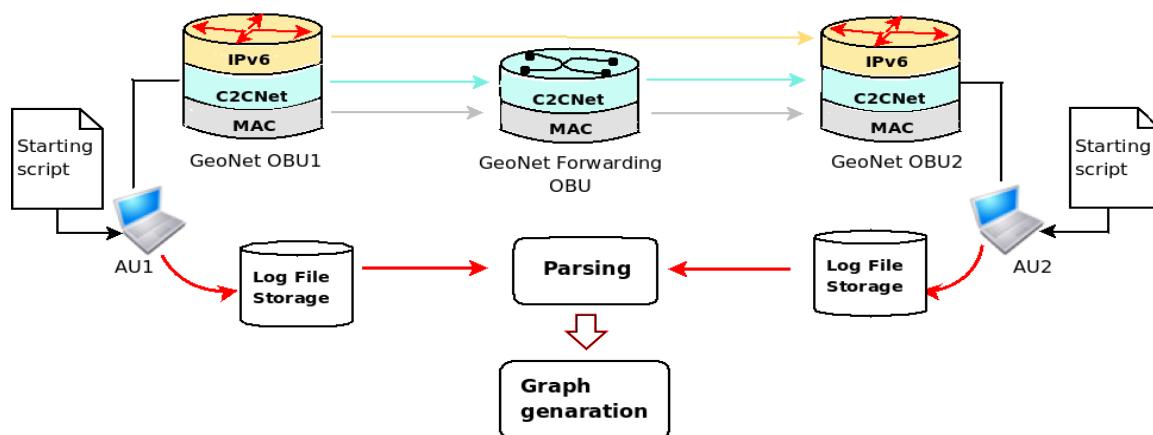


Figure 18: General Scenario of the Indoor Test

4.1.1 ICMPv6 Latency Evaluation Scenario

To evaluate the latency, we measured the Round Trip Time between the AUs attached to the OBUs. AU1 sends ICMPv6 Request every 0.1 second. The ICMPv6 packet is increased by 20 bytes. The packet size is varying from 20 bytes to 1500 bytes. Once the test is finished, the log file is parsed in order to get the maximum, the minimum and the average RTT as well as the packet loss for each packet size. In these tests, the log files are stored and parsed only on AU1. Table 10 presents a sample of the ICMPv6 log file parsed results. The first column shows the packet size which increases by 20 bytes. The last column presents the packet loss rate. The other columns show the minimum, the maximum and the average RTT (on this sample, the packet loss is always 0 because the packet size is small, however packet loss does occur with bigger packet sizes).

size	min	avg	max	mdev	packet loss
20	4.77	5.28	34.03	2.92	0
40	4.87	5.11	7.82	0.53	0
60	4.97	5.19	7.8	0.43	0
80	5.03	5.38	16.5	1.22	0
100	5.18	5.29	6.47	0.16	0
120	4.69	5.94	24.54	3.04	0
140	4.74	5.49	30.21	3.48	0
160	4.25	4.57	7.8	0.54	0
180	4.4	7.14	39.69	7.09	0
200	4.49	8.91	57.16	10.52	0

Table 10: Sample of the ICMPv6 Log File Parsed Results

4.1.2 UDP Packet Loss Ratio and Throughput Evaluation Scenario

In this test, we evaluate the packet loss ratio in a UDP communication. We set up a UDP client attached to OBU1 that generates UDP packets and sends them to a UDP server attached to OBU2. The UDP client and server saves the log file traces. After the tests, the log files of both the client and the server are parsed through pointers as the used port number and the packet loss results are plotted. The UDP packets are generated in AU1 and sent through C2CNet link to AU2. In this test the bandwidth is varying from 1 to 6 Mb/s. For each bandwidth value, the read-write buffer is increased from 20 bytes to 1900 bytes. Table 11 shows a sample of the log file obtained after parsing the native log files. The first column is the read-write buffer and the other columns show percentage of the obtained packet loss.

Buffer	1M	2M	3M	4M	5M	6M
100	22	2	1	1	1	1
300	82	37	22	15	10	3
500	96.3	66	41	28	23	21
700	100	95.5	61	43	44	33
900	99.82	99.86	79	58	62	48
1100	99.82	99.98	99.52	73	78	62
1300	100	99.9	99.98	88	94.6	77
1500	14	12	8	7	5	5
1700	13	13	10	7	8	6
1900	15	14	11	8	9	7

Table 11: Sample of the ICMPv6 Log File After Parsing

4.1.3 TCP Throughput Evaluation Scenario

In this test, the maximum throughput is measured using TCP traffic. We evaluate the throughput for three values of the TCP segment size: 400, 800 and 1200 bytes. For each value, the window size is increased from 200 to 1600 bytes. The TCP client is attached to OBU1 and the TCP server is connected to OBU2. In this case, only the log file obtained on the server side is parsed and analysed. Table 12 shows a sample of the formatted output file obtained in the multi-hop test. The throughput is measured in Kbits/s.

WindowSize	MAX400	MAX800	MAX1200
200	38.9	49.7	80.7
400	39.5	50.9	80.8
600	37.1	51	79.6
800	35.5	50.4	76.4
1000	38.6	50.3	77.1
1200	38.8	51.2	78.1
1400	37.1	77.5	80.1
1600	72.5	80.2	67.7

Table 12: Sample Output of the Multi-Hop Test

4.2 Test Results and Analysis

4.2.1 ICMPv6 Evaluation

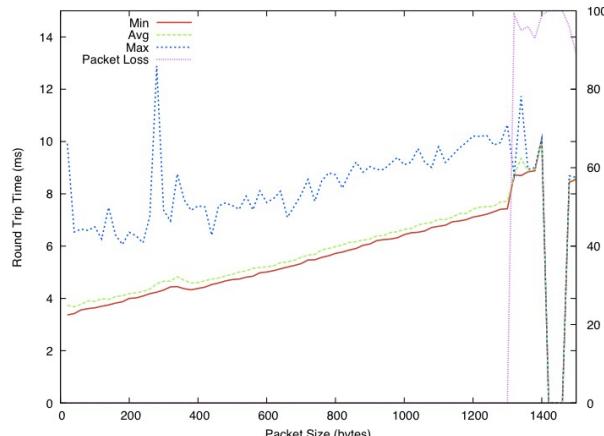


Figure 19: HITACHI C2CNet Layer With Next Hop Cache - Single Hop

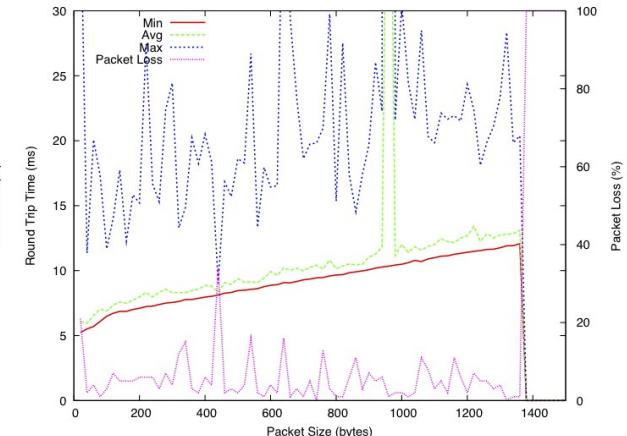


Figure 20: NEC C2CNet Layer Without Next Hop Cache - Single Hop

Figures 19 and 20 show single hop test with C2CNet layer implementation with and without next hop cache respectively. Both implementations mark increasing RTT as the packet size increases until the packet size is 1300 bytes which causes packet fragmentation. The packet loss rate is around 5 % on average when no next hop cache is being used, while there is no packet loss when the next hop cache is being used. The difference in the results comes from the fact that one implementation resolves the next hop for every packet via *netlink* as described in Section 3.3 of [GeoNetD3.1], while the other contains the next hop in the cache for the same IPv6 destination of the packet. This avoid heavy interaction between the *userland* software and the *kernel*. With the next hop cache there is a trade-off between reducing the latency of the next hop resolution and having higher possibility to loose the packet in case of a route change. In our tests the *next hop cache* doesn't present any disadvantage because the routing entries in the routing table have not been updated during the entire test. For the same reason, the RTT with the *next hop cache* is 2 ms shorter than the one without for a packet size from 20 bytes to 1300 bytes.

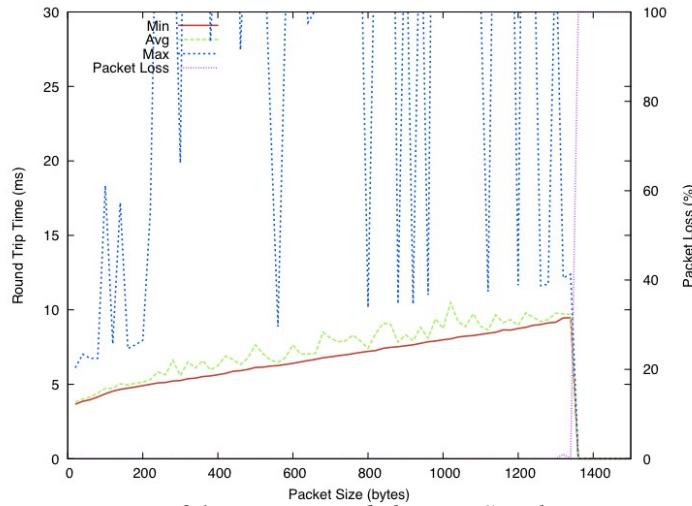


Figure 21: Interoperability on Single Hop

Figure 21 shows the results of the interoperability test where one GeoNet OBU runs HITACHI's C2CNet layer and the other one run NEC's. In this test, HITACHI C2CNet layer implementation doesn't resolve the next hop of the IPv6 destination frequently for outbound packets while NEC C2CNet layer implementation resolves the next hop for every inbound packet. The RTT is 4 ms with 20 bytes of packet size and 10 ms at 1300 bytes which is almost the average between HITACHI and NEC measurements displayed on Figures 19 and 20.

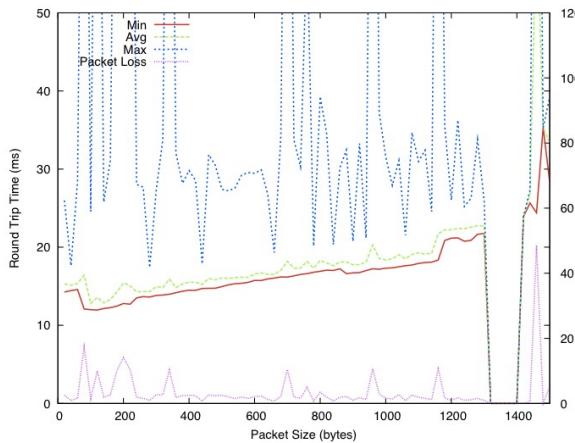


Figure 22: HITACHI C2CNet Layer With Next-Hop Cache - Multi-Hop

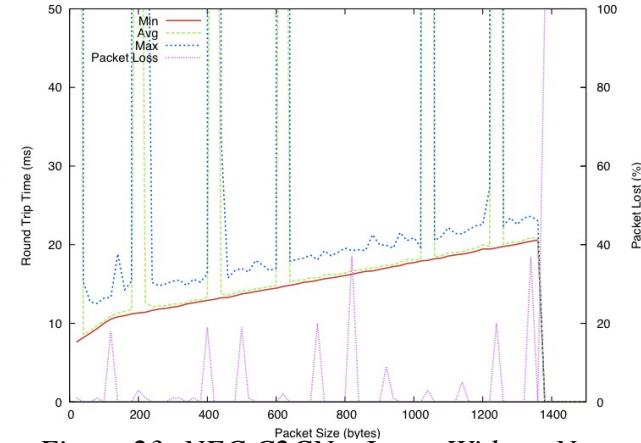


Figure 23: NEC C2CNet Layer Without Next-Hop Cache - Multi-Hop

Figures 22 and 23 show the RTT and packet loss rate in the multi-hop case with C2CNet layer implementations with or without *next hop cache*, respectively. The packet loss rate is under 10% during the test with both implementations. The RTT of both implementations has similar values from 10 ms to 20 ms during the test from 20 bytes to 1300 bytes of packet size. Buffering occurs with the implementation without *next hop cache*, which causes around 350 ms of delay. As the packets are buffered in the C2CNet layer, there is a limited packet loss while RTT is around 350 ms.

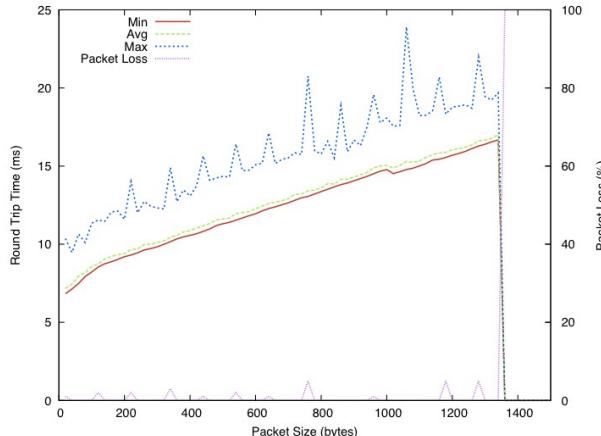


Figure 25: NEC-HITACHI-NEC Multi-Hop

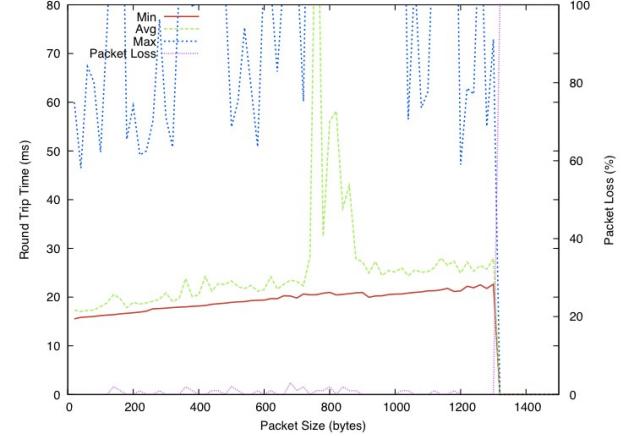


Figure 24: HITACHI-NEC-HITACHI Multi-Hop

Figures 25 and 24 shows the interoperability test between NEC and HITACHI implementations of the C2CNet layer. The test is performed with exactly the same configuration as in Figures 22 and 23 on the source GeoNet OBU and the receiver GeoNet OBU, but the intermediate GeoNet OBU is replaced by the other implementation. The result shows that the two implementations are fully interoperable.

Figure 23 shows the results of the configuration where only NEC implementation is used while Figure 25 shows the results of the configuration where HITACHI implementation is used in the middle node. They show almost the same RTT. However the packet loss rate and the packet buffering related delay are significantly reduced.

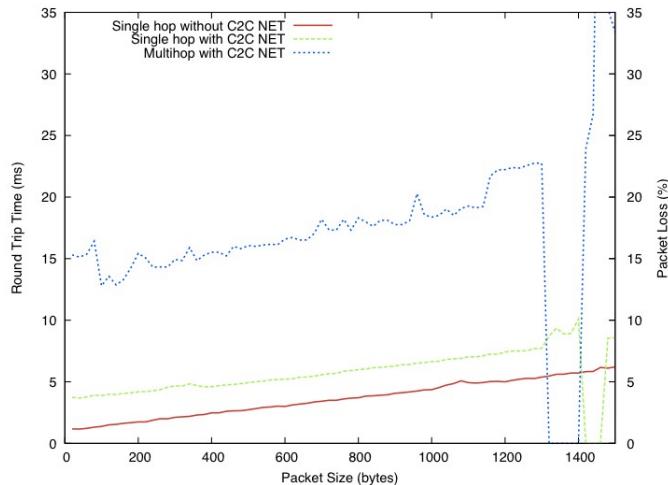


Figure 26: Overhead of IPv6 over C2CNet

Comparison of RTT is given in Figure 26. It shows the RTT on single hop without C2CNet (red line), single hop with C2CNet (green line) and multi-hop with C2CNet (blue line). In the single hop case, the RTT with C2CNet is 3 ms higher than one without C2CNet. In addition, packets with size exceeding 1300 bytes cannot be delivered with C2CNet because of the MTU, while the packet without C2CNet is delivered until 1500 bytes.

4.2.2 UDP Evaluation

UDP evaluation is performed with various packet sizes (100~1900 bytes) and sending rates (1M ~ 6M).

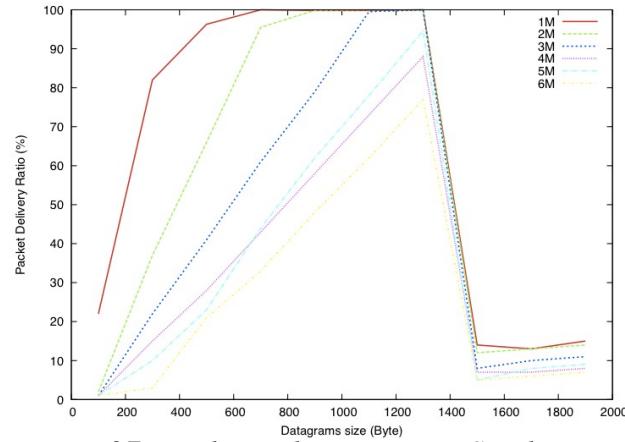


Figure 27: Packet Delivery Ratio - Single Hop

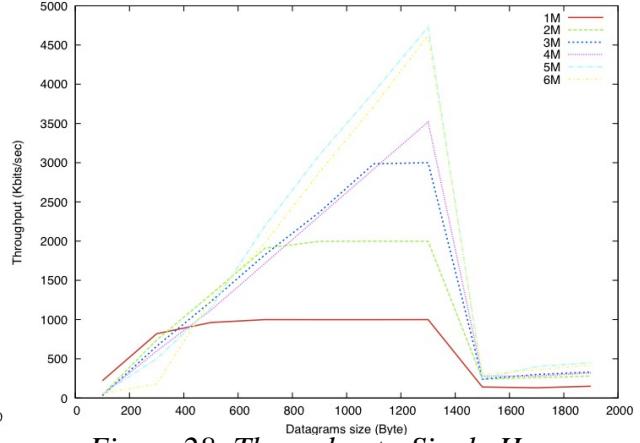


Figure 28: Throughput - Single Hop

Figure 27 shows the packet delivery ratio on single hop. Packet delivery ratio is low while packet size is small. There is no packet loss with a packet size between 700 bytes and 1300 bytes with 1M of sending rate, with a packet size between 900 bytes and 1300 bytes with 2M sending rate and with a packet size between 1100 bytes and 1300 bytes with 3M sending rate.

Figure 28 shows the throughput for the same tests as reported on Figure 27. The throughput is maximised with a 1300 bytes packet size for all sending rates. It shows that the most efficient configuration to send maximum data is realised with a 1300 bytes packet size and 5M sending rate. Maximum throughput is around 4500 Kbits/second.

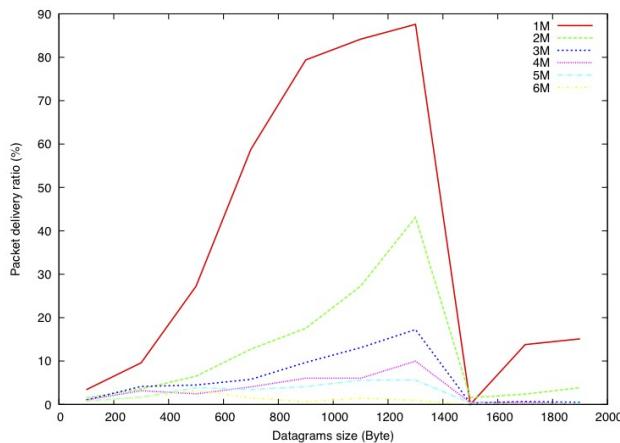


Figure 29: Packet Delivery Ratio - Multi-Hop

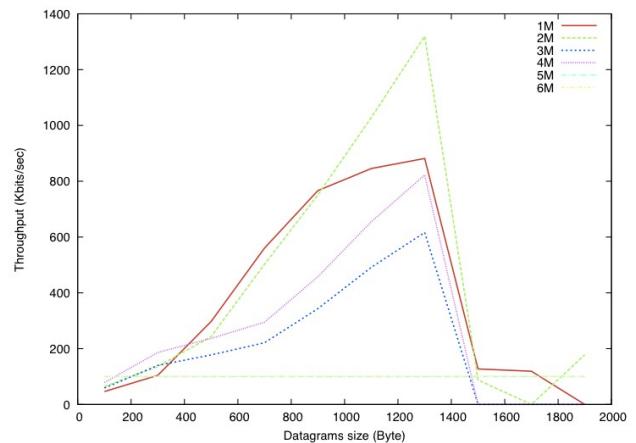


Figure 30: Throughput - Multi-Hop

Figure 29 shows the packet delivery ratio on the multi-hop path. No configuration allows to reach 100 % packet delivery ratio. 1300 bytes of packet size is the best configuration to obtain high packet delivery ratio. More than 90% of packets are dropped under a sending rate above 4M.

Figure 30 shows the throughput for the same tests as reported on Figure 29. The best throughput is obtained with a 1300 bytes packet size for all the sending rates. The best configuration on the multi-hop path is 1300 bytes of packet size and 2M of sending rate that is reached around 1400 Kbits/ second.

4.2.3 TCP Evaluation

We observed that TCP transmission doesn't work well with the current implementation of IPv6 over C2CNet. The throughput on the single hop path with HITACHI C2CNet implementation is shown in Figure 31 and the throughput on the multi-hop path on HITACHI is shown in Figure 32. Figure 31 shows that the throughput of TCP on the single hop path is under 200 Kbits/second which is extremely low compare to UDP tests reported in the previous section. Figure 32 also shows that the throughput of TCP on the multi-hop path is low (under 85 Kbits /second).

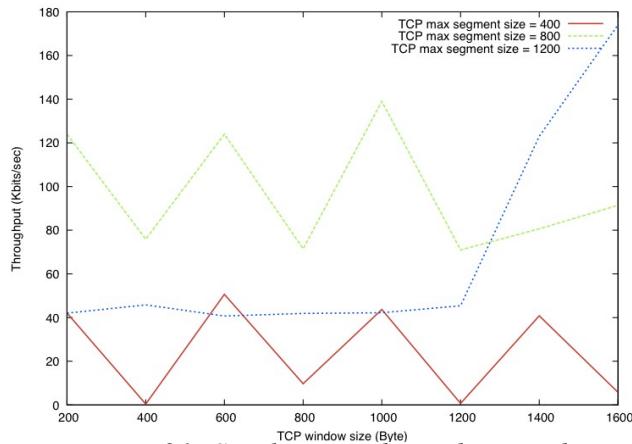


Figure 31: Single Hop Throughput with HITACHI

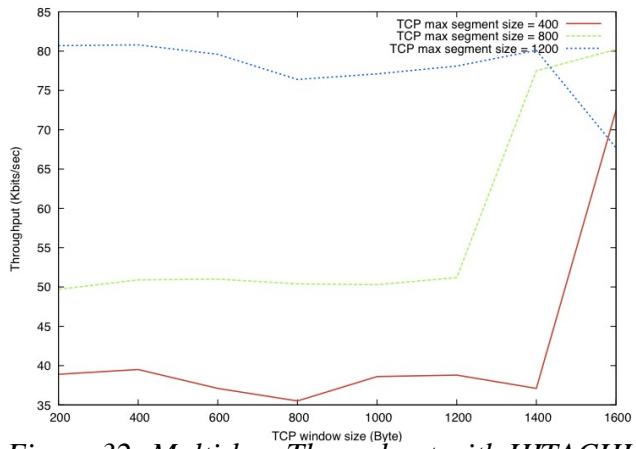


Figure 32: Multi-hop Throughput with HITACHI

We investigated this problem deeper with *tcpdump* logging. The results are shown in Figure 33. The log is taken from Scenario 2 of Section 5.1. In Scenario 2, we could check that the multi-hop configuration works in both unidirectional and bidirectional ways, meaning that UDP works with good quality (about 10 % packet loss), and ICMPv6 works without problems. According to the *tcpdump* result, the problem comes from the collision between the input and output traffic on the tun0 interface, which is the implemented interface between the C2CNet and IPv6 layers. In the TCP traffic case, some data simultaneously arrives in both ways (from C2CNet to IPv6 and from IPv6 to C2CNet). Some data is lost in the collision and the TCP connection is broken by these dropped packets.

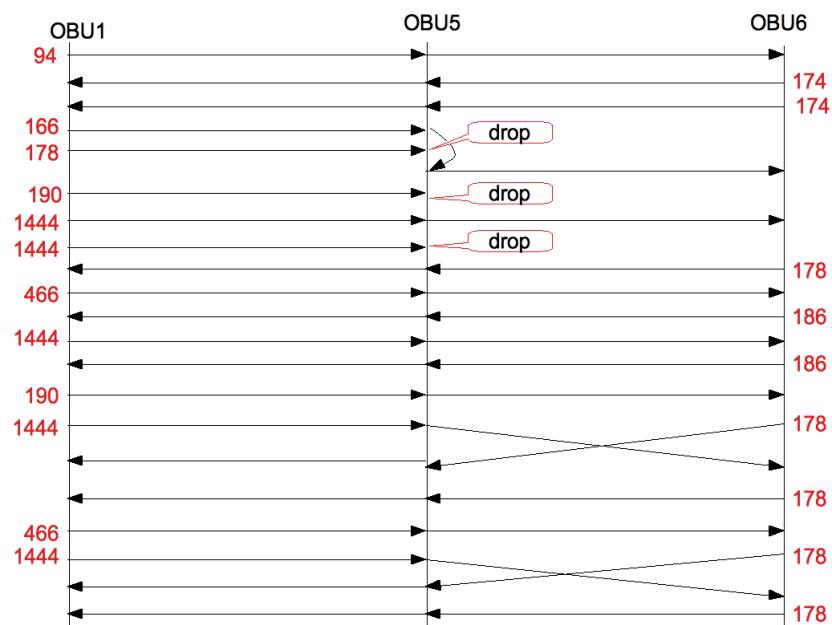


Figure 33: TCPDUMP Result

5. Vehicular Platform Evaluation

In this section we evaluate the performance of IPv6 over C2CNet in the GeoNet domain using the outdoor vehicular platform described in Section 3. The network configuration is almost the same as for the indoor experiments except that GPS is used to dynamically get vehicle position information and a network sniffer is used to retrieve packet information and paths. Tests scenarios are described in Section 5.1 and results are presented and analysed in Section 5.2.

5.1 Test Scenarios

A set of scenarios were considered taking into account several road parameters and constraints to render the scenario as realistic as possible. The main factors which determine these scenarios are:

- *Mobility*: Vehicle mobility is a key issue to cope with realistic VANET conditions. This way, we have considered static scenarios, to test the network operation in a controlled way, but also dynamic scenarios under common traffic situations. Of course, field operational tests should be conducted to confirm the experimental results taking into account proper handling of mobility, i.e. doppler shifting, fast fading, etc.
- *Environment*: Urban and interurban locations affect communication performance, because the signal propagation is hidden by buildings (among other elements), and the line of sight between vehicles is not always possible. In our tests, two environments are considered: a semi-urban one located at the INRIA-Rocquencourt installations, which contains a set of small buildings surrounded by streets, and a highway stretch, the French A-12, near INRIA-Rocquencourt. The test are performed in the 2.4 GHz radio band due to lack of available hardware. Also, the results are affected by the type of antenna. Field operational tests should thus be performed in the 5.9 GHz frequency band and with antenna diversity.
- *Number of vehicles*: The number of hops between the source and the destination vehicles affect the communication delay, as it was expected. In addition to the extra forwarding delay, the packet loss at MAC level also increases due to transmission interferences. Up to three vehicles are considered in the field trials, in order to check the increase of communication delay with the number of hops. Of course, field operational tests should be conducted to confirm the experimental results with hundreds of vehicles.

Testing scenarios have been divided into urban and highway; mobility has been set to static, urban-like speed, and high speed; and a wide range of performance metrics have been used, such as bandwidth, RTT (Round-Trip delay Time), jitter and PDR (Packet

Delivery Ratio). The set of traffic types (UDP, TCP and ICMPv6) has been applied over each defined scenario.

The following features of the communication were checked before obtaining the performance results:

- The wireless range was about 435 meters when the vehicles are in a direct sight (obstacles such as buildings, car traffic and interferences were avoided)
- The wireless range was about 135 meters in normal conditions including urban obstacles and wireless interferences
- The maximum tested average vehicle speed is around 100 km/h in the highway.
- The maximum tested average vehicle speed is around 30 km/h in urban conditions

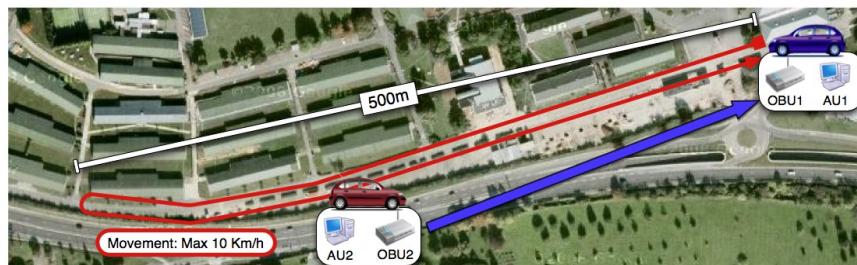


Figure 34: Scenario 1 - Maximum Range

Scenario 1 as illustrated on Figure 34 considers one parked vehicle and one moving vehicle. The test was performed both when vehicles were in normal urban conditions and isolated from obstacles and interferences. The aim of this scenario was to check the maximum distance the wireless range can reach.

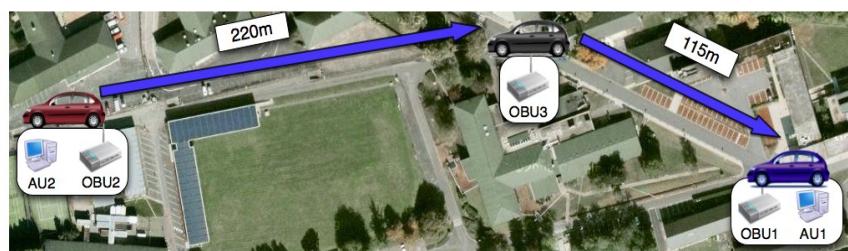


Figure 35: Scenario 2 - Static Urban

Scenario 2 as illustrated on Figure 35 considers communication between three parked vehicles, in a typical urban environment, where a building (or a set of them) hides the line of sight between the source and the destination vehicles. A multi-hop network is suited for handling this kind of situation.

Scenario 3 illustrated on Figure 36 considers moving vehicles in urban environment where communication is limited by urban obstacles such as buildings, trees and sources of interferences.

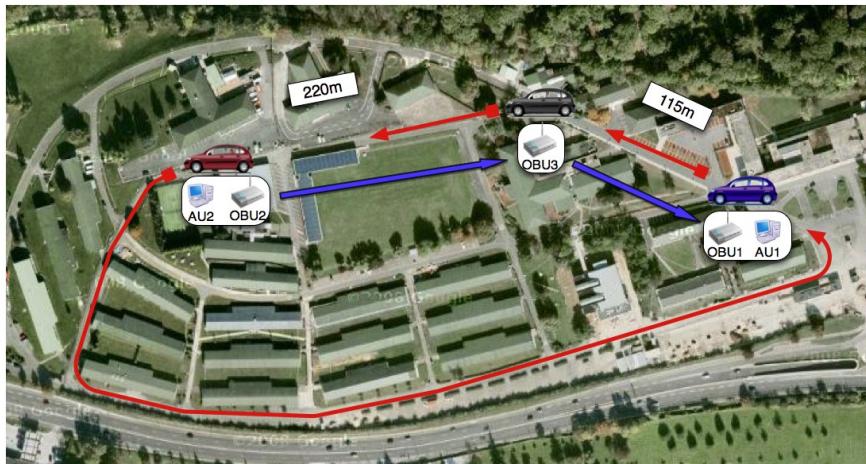


Figure 36: Scenario 3 – Dynamic Urban Environment

Scenario 4 as illustrated on Figure 37 considers three moving vehicles on the highway. Vehicle 2 acts as a forwarder of the message received from Vehicle 3 to Vehicle 1. The distance is dynamically varying due to road traffic conditions.

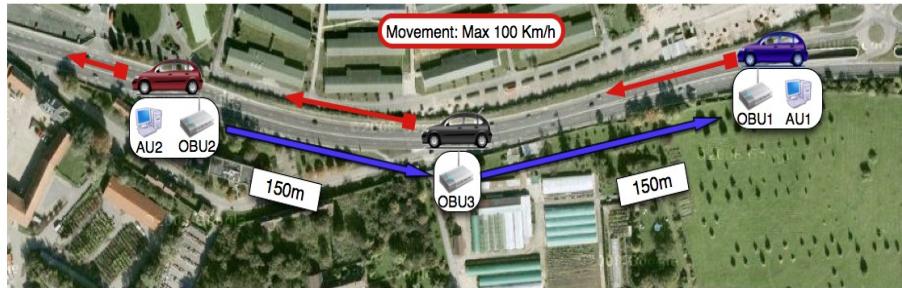


Figure 37: Scenario 4 – Three Vehicles on the Highway

5.2 Test Results and Analysis

5.2.1 Distance Evaluation

The evaluation of the distance has been performed with two cars considering Scenario 1. The sender vehicle gets away from the receiver vehicle (whose position is static), and then comes back when the communication is interrupted, and finally returns to the initial point. The speed of the sender was maintained under 10 Km/h to smoothly check the loss of connectivity.

Figure 38 shows the RTT with ICMPv6 transmission. The RTT is within 5 ms to 10 ms until 420 meters. After this point, no packets are delivered, until the sender vehicle comes back and reaches 100 meters of distance. Since periodical C2CNet beacon messages are lost when the distance is around 420 meters, the destination C2CNet ID is removed from the location table and the transmission ends at this point.

Throughput using TCP and considering the same scenario is given in Figure 39. The maximum throughput is around 1000 Kbits/sec when the vehicles are parked next to one another. When the distance is from 50 meters to 200 meters, the average throughput is around 500 Kbits/sec and the TCP communication is interrupted at 270 meters. The communication doesn't recover during the rest of the test, because the TCP session time out. It takes 50 seconds to come back to a distance of 100 meters where ICMPv6 recovered during previous test.

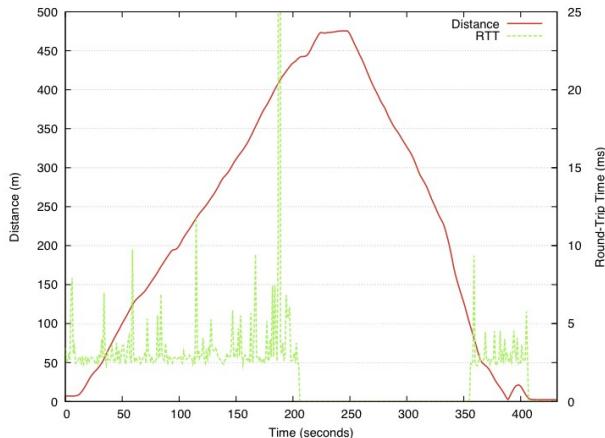


Figure 38: RTT on ICMPv6 with Distance

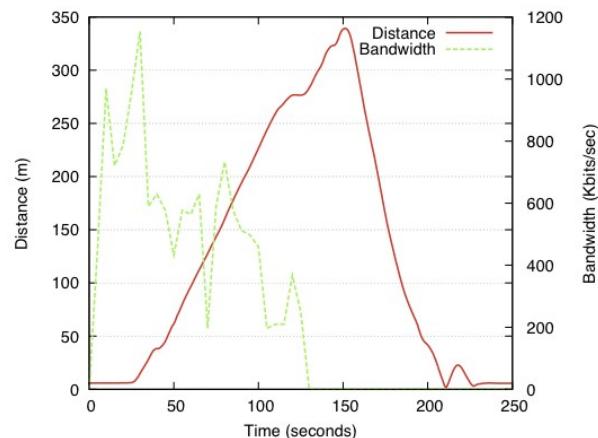


Figure 39: Throughput on TCP with Distance

The packet delivery ratio using UDP with Scenario 1 is shown in Figure 40. The packet delivery ratio is almost 100 % from beginning to 200 meters. From 200 meters, the packets are starting to be dropped and the packet transmission finally ends at a distance of 420 meters. The packets are not delivered until the vehicle comes back to a distance of 400 meters 50 seconds after the communication ends.

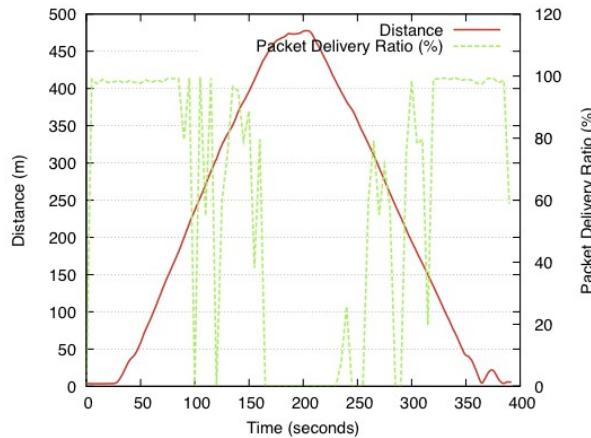


Figure 40: Packet Delivery Ratio on UDP with Distance

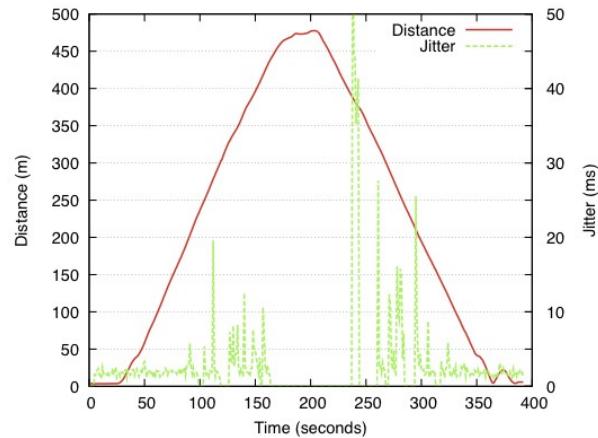


Figure 41: Jitter on UDP with Distance

The jitter of in the same test is illustrated on Figure 41. When the sender car leaves the receiver one, at a distance between 250 and 420 meters, the jitter is higher, due to layer two retransmissions caused by the increase of the distance. When the sender approaches

the receiver again, this effect, but higher, is again visible at distances between 400 and 200 meters. This is due to packet flood of buffered packet during the disconnected period.

5.2.2 Static Test

The results of Scenario 2 (three parked vehicles) are summarised in Table 13. The total distance between the sender and receiver vehicles was 330 meters (220 plus 110 meters). The average packet delivery ratio and throughput were 90.18% and 901.95 Kbits/sec. As expected, the packets were sometimes transmitted directly from sender to receiver. It was foreseeable because there are large obstacles (buildings) near the receiver and the sender. And the packet delivery ratio degrades to 21.88 % at the path change. The jitter reached up to 39.2 ms at during path change period while the average jitter is 2.98 ms. The packet delivery ratio was stable around 95%.

The RTT with ICMPv6 packet is also summarised in Table 13. The communication is unstable during 300 seconds. The RTT varies from 4.6 ms to over 5000 ms. This is due to the link failure. 600 ICMPv6 packets are sent during the test (2 packets in a second). The packets passed via four links when the communication goes over the multi-hop path. We see the packet loss in all the links. 31 packets are dropped on the first link, 18 packets on the second link, 65 packets on the third link and 10 packets on the last link. The total packet loss was 124 packets which represent 20 % loss. Also 4 packets went on a single hop path and 9 packets went on an asymmetric path. There was a stable period of 25 seconds during which packets always went through a multi-hop path. During this period, the average RTT was 5.81 ms as for the indoor tests analyzed in Section 4.3.

Test	Metric	Minimum	Average	Maximum	Standard deviation
UDP 3 vehicles	PDR (%)	21,88	90,18	98,13	14,99
	Bandwidth (Kbps)	274,56	901,95	998,4	151,31
	Jitter (ms)	1,25	2,89	39,2	5,27
ICMPv6 3 vehicles	RTT (ms) all 300 sec	4,6	477,43	5080	992,31
	RTT (ms) stable 25 sec	4,74	5,81	9,66	1,46

Table 13: Network Performance in Static Test

5.2.3 Dynamic Test (up to 30 Km/h)

Figure 42 shows the RTT and hop count with distance between OBU1-OBU2 and OBU2-OBU3. According to Scenario 3, three vehicles have been driven around a set of buildings, with the intention of blocking the direct link between Vehicles 3 and 1. The speed of the test where set between 15 km/h and 30 km/h. The two-hop and four-hop cases are corresponding respectively to bidirectional single hop and multi-hop paths (ICMP echo request and echo reply take the same path). Several three-hop routes have been monitored, due to, sometimes, the ICMP Echo Request packets taking a different route than the Echo Reply ones. At some period, no hop count is marked because the sender OBU doesn't send any packet due to lack of next hop C2CNet ID in the location table because it beacons were not received. In this case, no packet is emitted from the OBU. In the test, most of the packet transmission passed to single hop (two hop for returning); this is because the C2CNet layer tries to select each hop as distant as possible in order to minimise the hop-count. The mechanism works very well and the INRIA Rocquencourt (500 meters x 250 meters) campus is too small to observe multi-hop path.

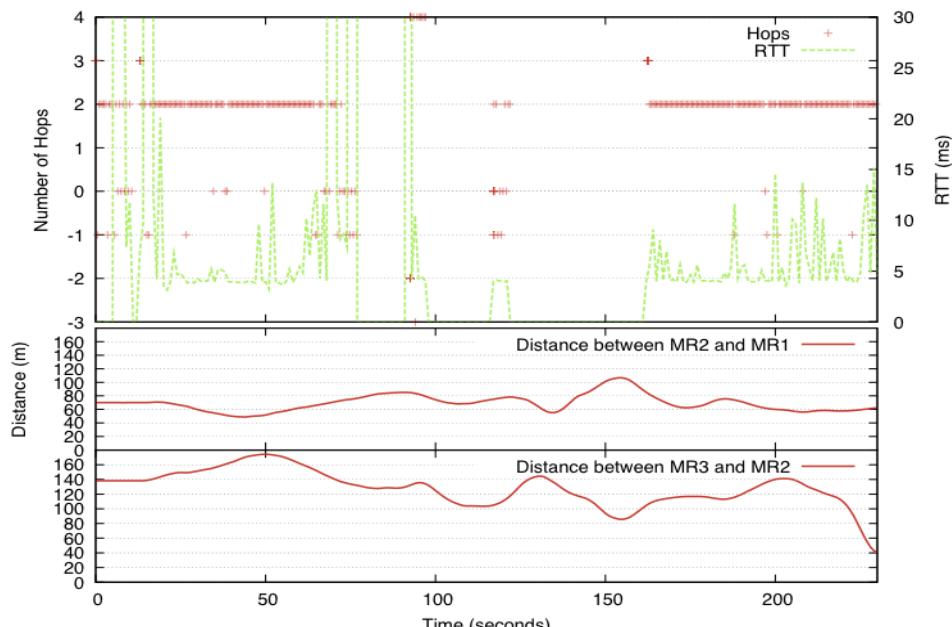


Figure 42: RTT and Hop Count with Distance (under 30km/h)

Figure 43 shows hop count, packet delivery ratio and jitter on dynamic tests under 30 km/h. The upper plot shows the number of hops used in the paths followed by UDP packets, whereas the lower graphs show the packet delivery ratio, computed end to end and per link. The packet delivery ratio is calculated per second, while the hop-count is plotted for each packet transmitted from the sender node. When no hops are drawn, the route to the destination vehicle is not available. Zero hops means that the packet was sent by the first GeoNet OBU but was not received by any other. Negative values represent those packets which did not arrive to the destination vehicle, but some hops were reached. As can be seen, a direct relation exists between packet delivery ratio and number of hops. When this last value is equal or lower than zero, the packet delivery ratio decreases. When

the vehicles are in the same street, some direct paths (one-hop) appear; however, when the distance between the sender and the receiver vehicles is large enough, the two-hop route is used. These different types of paths can be also seen if the per-link packet delivery ratio is observed. Whereas the direct link (OBU3-OBU1) gives intermediate packet delivery ratio values, the packet delivery ratio between consecutive vehicles is almost identical and near 100% when the two-hop link is used, due to the lower distance between nodes.

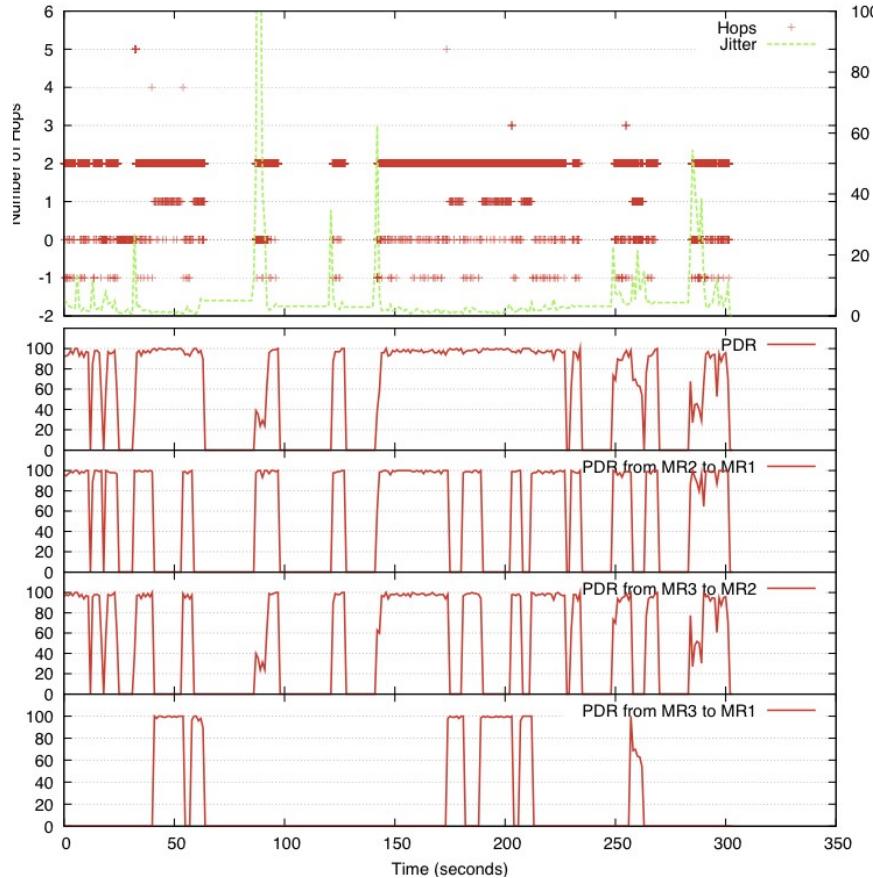


Figure 43: Hops, Packet Delivery Ratio and Jitter on Dynamic Test

5.2.4 Dynamic Test (up to 100 Km/h)

The dynamic tests performed over highway conditions follow Scenario 4. The speed of the cars was around 100 km/h, but the distance between vehicles was variable, due to the rest of traffic on the road. Moreover, communication problems in this test are not only due to buildings, but also to surrounding vehicles. Figures 44 and 45 show hop count and RTT with distance between OBU3-OBU2 and OBU2-OBU1.

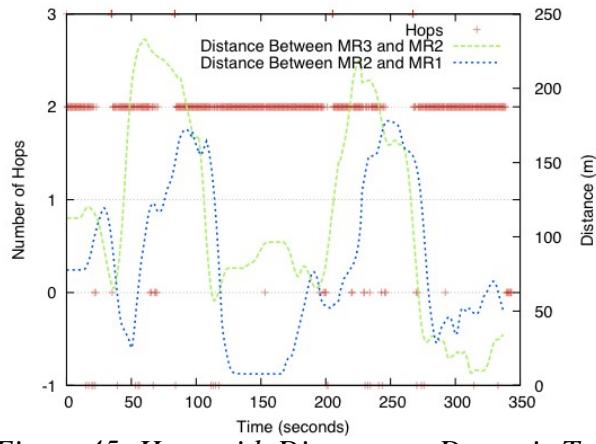


Figure 45: Hops with Distance on Dynamic Test

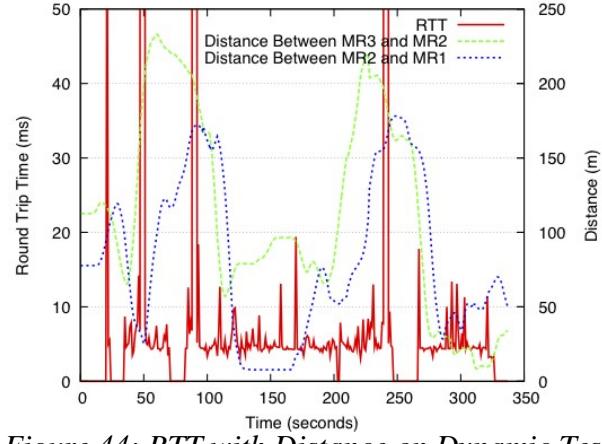


Figure 44: RTT with Distance on Dynamic Test

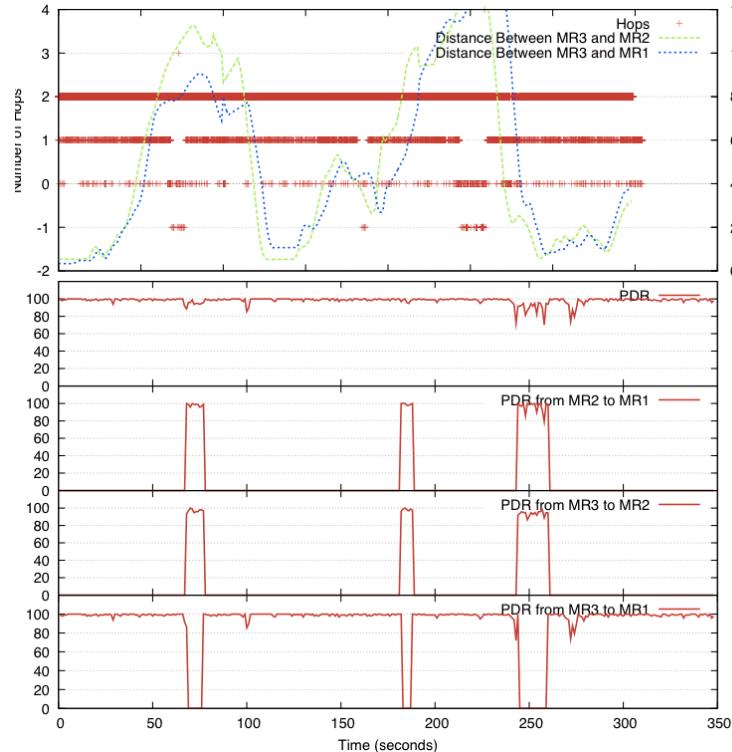


Figure 46: Hops and Packet Delivery Ratio with Distance on Dynamic Test

6. Development Tests

In this section we report the history of the debugging and improvement of the GeoNet development. The main milestones were plug-tests meeting where GeoNet partners had specific goals and were successful to make the significant progress. More tests have of course been performed by individual GeoNet partners in-between these plug-tests meeting, but with much less visible achievements than during plug-tests gathering the partners working in a single test platform and helping each other.

6.1 Plug-Test at NEC in Heidelberg

This was the first developers' plug-test where we could use the actual implementation together. Most of the time was spent to check interoperability among IPv6, C2CNet and Lower Layers and among NEC, HITACHI, Lesswire and INRIA's implementation. First, we agreed to use common wifi configuration which is Channel: 149 (5.745), ESSID: GeoNet, Data Rate: 6. Second, connection test between C2CNet and Lower Layer was successfully performed with both of NEC and HITACHI implementation. For test between Position Sensor and C2CNet implementation, we got two problems. 1) Position Sensor uses "/dev/gpspos" as the default GPS device. But, USB devices, e.g. those used by HITACHI and NEC are attached to device "/dev/ttyUSB0". 2) NEC laptop has some problems with *localhost* address. Because of those two issues, when the Position Sensor is launched the GPS device and localhost IP address should be set as follow:

```
./PositionSensor /tmp/GeoNetPos gpsd /dev/ttyUSB0 127.0.0.1 2947
```



Figure 47: Development Test 18 September
~ 19 September

Next, the beacons between NEC and HITACHI were exchanged with Ethernet cable but not wifi. They were not able to communicate with one another because the wifi card shipped with the laptop created a new interface `ath0.x` for each new communication, which make the two boxes sending and listening on different interfaces. With the test with

Ethernet, we confirmed the beacons are interoperable between the two C2CNet implementations. We didn't have time to confirm if C2C-IP SAP (virtual interface tun0) works.

6.2 1st Plug-Test at INRIA Rocquencourt



Figure 48: Development Test 30 September
~ 1st October

Before the plug-test we worked on the same laptop for both C2CNet implementation and test. During the meeting we started using the hardware equipped in INRIA vehicles as the GeoNet OBU as that hardware will be used in the forthcoming demonstration and experimental evaluation. The hardware has an AMD architecture and we got problem with treatment of network byte order and host byte order. We've fixed it so that we can use the AMD hardware. IP-C2C SAP start working from IPv6 to C2CNet, but not the opposite way. We tried to set organise more meeting before ITST 2009 in Lille (22nd of October) to make a demonstration of C2CNet implementations there.

6.3 Plug-Test at HITACHI in Sophia-Antipolis

At this point, neither HITACHI and NEC implementations of C2CNet layer worked correctly

With HITACHI implementation, the C2C-IP SAN can not write on tun0, thus received IPv6 packets could not be delivered to the IP layer. The problem was that C2C-IP SAP was dealing with Short types (2 bytes) and HITACHI C2CNet was dealing with char types (1 byte). The function which writes on tun0 is sensitive to such a detail. This problem has been solved, and now the HITACHI implementation, including the C2C-SAP is working as it should be.



Figure 49: Development Test 8th October ~ 9th October

C2C-IP SAP has not been tested because it was not possible to launch the NEC's C2CNet on one of the two INRIA Boxes. Whenever NEC software was launched on one of the two INRIA boxes the Kernel crashed. A possible reason is that the *madwifi* driver crashes the system (possible solution is to update the kernel version). It was decided to continue the test with HITACHI's implementation, until NEC provides an improved version of the C2CNet implementation.

The performance improved after ICMPv6 echo request and echo reply worked. When the Camera (IPv6 host acting as AU) sends the data to the browser of the AU with HTTP, the frame size can be up to 1500 bytes. But when C2CNet adds the C2CNet headers and the C2C-LL SAP header, the total size of the flame became over 1500 bytes and exceeded MTU of wifi (ath0). Thus the MTU of the IP-C2C SAP should be set-up to a value under 1350 bytes. So, we have limited the MTU of tun0 to 1350 bytes in order to allow IPv6 over C2CNet to work properly.

Not only the video quality which was not very good, but also the HITACHI C2CNet implementation was crashing very often. After some investigation, it was understood that the C2C-IP SAP was making the kernel buffer getting full after a period of time, which made the C2CNet software to crash. The C2CNet Layer was not processing the IP packet fast enough. We decided to go to the ITST workshop with this configuration, even though the video quality was not very good, and even though the C2CNet Layer crashed often.

To overcome the fact that the C2CNet crashed that often, it was decided to make a bash script which relaunches the software whenever it crashes. After many tests and investigations, INRIA proposed to make some changes in the C2C-IP SAP. It was proposed to have the C2CNet Layer checking the IPv6 Routing Table to get the IP address of the next hop only when an IPv6 packet with a new destination address is received from IP layer. Which means, the C2CNet will access to the IPv6 routing table much lesser. A new version of HITACHI's C2CNet Layer implementation has been provided, which includes a version of the modified C2C-IP SAP. Single hop communication was starting to work properly, but multi-hop communication still had performance issues. We prepared a launch script to start the GeoNet OBU features.

6.4 Plug-Test at IMDEA in Madrid

Date: 24th November ~ 25th November

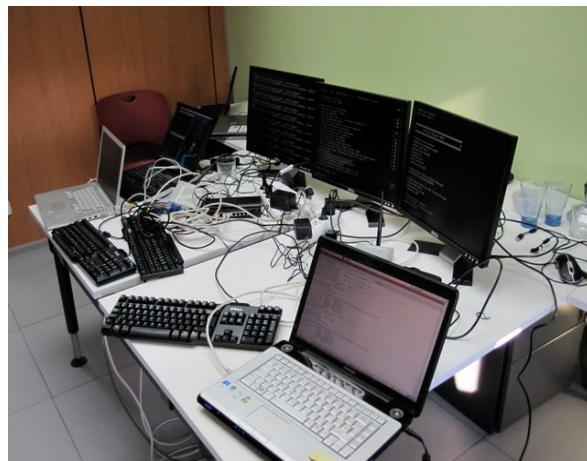


Figure 50: Meeting at IMDEA in Madrid

Plug test Meeting purpose: Validation and debugging of the indoor major tests, with the following test objectives:

- Test unicast and multicast communication in both single and multi hops
- Validate interoperability between NEC's and HITACHI's implementations
- Evaluate and fix the performance issues

Table 14 summarises the set of tasks which was scheduled for the meeting and their final status. The plug test meeting was in general successful. About 70% of the test agenda was achieved. Single hop communication was almost validated for both NEC and HITACHI implementations. There were some issues regarding multi-hop and interoperability which were caused by a wrong set of the IPv6 packet. Some performance issues were still not solved.

Task	Description	Problems	Status
<i>Unicast communication using single and multihop tests</i>	Debug of NEC's implementation in single hop	beacons were being lost unexpectedly and route was not being re-discovered	fixed
	Debug of NEC's implementation in multi-hop test	issues with Location Request being sent out	fixed

	Debug of Hitachi's implementation in multi-hop test	- Performance problems: Long Round Trip Time (1-3 sec) -indoor range flag should not be used, but instead other way of suppressing packets -Daemon crash and restart	fixed
<i>Interoperability tests</i>	Debug of Single hop interoperability test : Tested for packet length of 64, 1308 bytes	Missing bytes from the IPv6 packet after decapsulation of the C2C header due to different understandings of length field in the common header	fixed
	Debug of Multi-hop interoperability test: Hitachi -NEC-Hitachi	Initially long Round Trip Time is experienced (6-10 seconds) - Some packets are lost	fixed
<i>Multicast communication using single and multi-hop tests.</i>	Debug of NEC's implementation in single hop	multicast IPv6 packets are sometimes received at the eth0 interface but not sent to tun0	Not fixed
	Debug of NEC's implementation in multi hop	Same problems as single hop tests	Not fixed
	Debug of HITACHI's implementation in single hop		OK
	Debug of HITACHI's implementation in multiple hop		OK
<i>Interoperability testing</i>	Debug interoperability test in single hop, packet size= 6 - 1300 bytes	multicast IPv6 packets are sometimes received at the eth0 interface but not sent to tun0	Not fixed
	Debug interoperability test in multi-hop, packet size= 6 - 1300 bytes	Same as single hop	OK
<i>Performance issues</i>	Test of several types of traffic: ICMPv6, TCP & UDP	for multi-hop scenarios we were experiencing heavy losses (70-80%) and long round trip times.	Not fixed

Table 14: Plug-Test Agenda at IMDEA

6.5 2nd Plug-Test at INRIA in Rocquencourt

Date: 13th January ~ 15th January



Figure 51: 2nd at INRIA in Rocquencourt

Meeting purpose: Preparation of GeoNet final demonstration with the following objectives:

- Agreement and adjustment of the demo scenario
- Setting up and configuration of the demo on vehicles

Table 15 summarises the scheduled tasks, the problems we met and the final status. As an outcome of this meeting, the demonstration scenario was preliminarily set and agreed but some extensions were still required. The wireless environment and the vehicle configuration were checked. GeoNet main features such as receiving messages in a dedicated GeoDestination area and C2CNet packet forwarding were validated.

Task	Description	Problems	Status
<i>Testing a preliminary scenario with one RSU and one OBU</i>	Check antenna	Antennas were not compatible with the 149 (5.4 GHz) and the 153 (5.7 GHz) frequency channel so range was very short	fixed
	Test of the Position Sensor module	- Problems due to using different GPS data speed rates - Conflict in accessing GPS serial port device between two applications	fixed

<i>Test of multicast using GeoBroadcast</i>	Check exchange of beacons		OK
	Check multicast using GeoBroadcast for both implementations	With NEC's implementation, sometimes the daemon is not able to forward the multicast message received in tun0	Not fixed
	Test interoperability when using GeoBroadcast		OK
<i>Test of the agreed scenario</i>	Test the scenario with 1 RSU, 2 OBUs and one server: one OBU reports the road hazard message to the server through RSU which broadcast it to a dedicated area	With NEC's implementation, sometimes the daemon is not able to forward the multicast message received in tun0	Not fixed
	Check receiving the GeoBroadcast message when the vehicle is inside and outside the area		OK
	Check GeoNet application functioning	Some use cases have to be adjusted and some modification in the GUI are required	OK

Table 15: Plug-Test Agenda at INRIA Rocquencourt

7. Demonstration

In this section we report the two demonstrations GeoNet partners performed within the framework of WP8 “Dissemination and Use”. The first demonstration was performed during the GeoNet workshop co-located within the ITST conference. The second one was organised during a standalone GeoNet workshop.

7.1 Demonstration at ITST 2009 in Lille

GeoNet organised its first workshop at the ITST conference in Lille in October 2009. We showed the IPv6 over C2CNet communication with IPv6 video camera. It was a single hop communication because the live camera video stream was of much better quality in that case. The visitors came to see the demonstration after the GeoNet workshop plenary presentation took place, so all visitors already knew the concept of GeoNet.

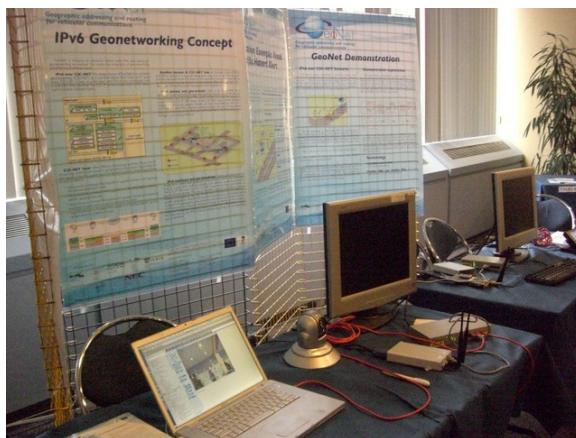


Figure 52: Development Test: 22nd October
(GeoNet Workshop at ITST)

The purpose of the table-top GeoNet demonstration was to demonstrate that GeoNet has existing implementation and to show how the interaction between IPv6 and non-IP vehicle-to-vehicle GeoNetworking is made possible. There were two GeoNet OBUs (mobile routers) running an IPv6 stack and two OBUs not running an IPv6 stack. Each OBU is supposed to represent a vehicle. All of them are running the GeoNetworking protocol (C2CNet layer). To facilitate the understanding of this new concept, an IPv6 video streaming was sent from an IPv6 camera attached to the first IPv6 OBU and was received by an IPv6 host attached to the second IPv6 OBU. The non-IPv6 OBUs acted as intermediate non-IP hops in between the two IPv6 OBUs.

This scenario corresponds to the scenario depicted on Figure 53 where two vehicles are driving around a roadside unit (RSU). Vehicle A is within RSU's communication range. Vehicle B is out of the communication range of the RSU but inside the communication range of vehicle A. Vehicle B is thus reachable from the RSU via vehicle A, i.e. via a multi-hop path. On the roadside, an IPv6 node (AU1) is attached to a RSU functioning as an

IPv6 access router (GeoNet RSU). In vehicle B, an IPv6 node (AU2) is attached to OBU2 functioning as an IPv6 mobile router (GeoNet OBU) serving an in-vehicle network. In vehicle A, OBU1 (C2C OBU) does not serve any in-vehicle network:

- RSU, OBU2, AU1 & AU2 are IPv6 nodes, unlike OBU1.
- RSU, OBU1 & OBU2 are equipped with wireless communication interfaces (ath0).

In the ITST demonstration, AU1 is an IPv6 camera and AU2 a laptop receiving and displaying the IPv6 video streaming from the IPv6 camera. IPv6 video streaming packets from the camera are encapsulated by RSU into C2CNet layer packets and forwarded hop-by-hop up to OBU2, which decapsulates the packets and delivers them to the IPv6 laptop.

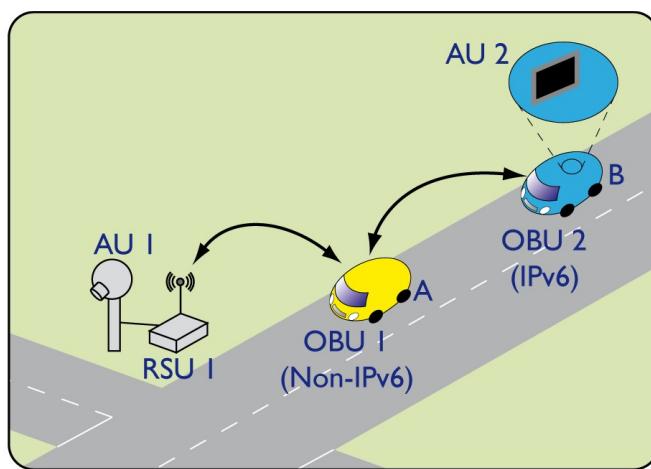


Figure 53: Scenario Demonstrated at ITST

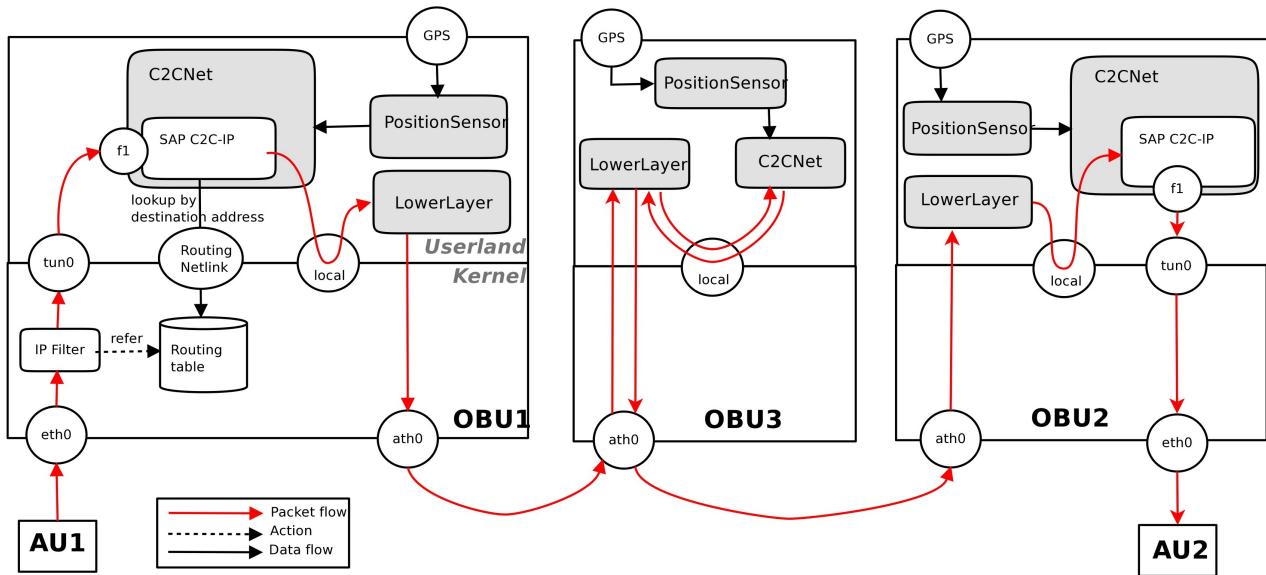


Figure 54: IPv6 Packet Forwarding over C2CNet

Figure 54 illustrates the implementation of IPv6 GeoNetworking. On RSU, the C2CNet layer gets IPv6 unicast packets from AU1 through tun0. By checking the IPv6 packet destination address, it looks up the routing table via Routing Netlink to obtain the IP next hop. From the IP Next Hop, C2CNet gets the C2CNet ID of OBU2, which corresponds to the last 64 bits in the IP Next Hop address. Once the C2CNet ID of OBU2 is obtained, C2CNet generates a new GeoUnicast packet and sends it to OBU2 with the IPv6 packet attached as payload. The packet is transmitted to the nearest OBU to OBU2, i.e. OBU1 and therefrom retransmitted up to OBU2. Once at OBU2, the GeoUnicast packet is decoded and its payload (IPv6 packet) is delivered to AU2 through tun0.

Around 150 persons visited the demonstration and many visitors were interested in the hardware of our GeoNet OBU which is a PC engine over ALIX system boards. We opened the board to show the inside. One of the most frequently asked question was about the IEEE 802.11p technology. We were asked if we use 11p and if we have a plan to use it.

The agenda, presentation material, the four posters designed on purpose and pictures can be found on <http://www.geonet-project.eu/?p=146>.

7.2 Final Demonstration at INRIA Rocquencourt

GeoNet organised its final workshop on the 29th January 2010 – the day of official project completion – at INRIA Rocquencourt, France. It was attended by 60 engineers, scientists and managers in the field of New of Information and Communication for Intelligent Transport Systems (ITS).

In the afternoon, a live demonstration of IPv6 GeoNetworking was performed both outside (vehicles) and inside (road traffic centre). A plenary presentation of the demonstration was given in the lecture hall prior to the live demonstration.

The purpose of the demonstration was to show GeoNet achievements. The demonstration of the combination of IPv6 and GeoNetworking focused on the following features:

- Multicast transmission at the IPv6 layer and GeoBroadcast transmission at the C2CNet layer to a set of destinations in particular destination area (GeoDestination);
- Internet-based communication (from Internet to GeoDestination);
- Multi-hop GeoRouting at the C2CNet layer (from RSU to OBU to OBU).

It was decided to perform realistic scenarios and to demonstrate use cases that would particularly benefit from IPv6 GeoNetworking. IPv6 GeoNetworking was thus illustrated through a **Traffic Hazard Notification** application developed on purpose by INRIA (see Section 3.4). This type of application, which targets road safety shall benefit the most from the combination of IPv6 and GeoNetworking and thus is the very appropriate to

demonstrate the usefulness of IPv6 GeoNetworking, which is not limited to infotainment types of applications. This scenario clearly shows that some applications ranged into the category “road safety” can benefit from IPv6 support in order to extend the dissemination range.

The demonstrated scenario is illustrated on Figure 55:

- The Green vehicle is reporting traffic hazard through RSU2 to the Road Traffic Centre (IP unicast / UDP);
- The house represents the Road Traffic Centre reporting traffic hazard (IP multicast / UDP);
- RSU1 is serving the target GeoDestination, it receives IP multicast packets and GeoBroadcast them at the C2CNet layer;
- The blue circle is the radio range of the GeoNet OBU (IP access router) .
- The yellow circle is the GeoNetworking range served by GeoNet OBU through multi-hop GeoRouting (GeoNet domain and GeoDestination of the traffic hazard);
- The Black vehicle (GeoNet OBU) is a static vehicle in radio range of the GeoNet OBU able to relay traffic hazard further (GeoBroadcast);

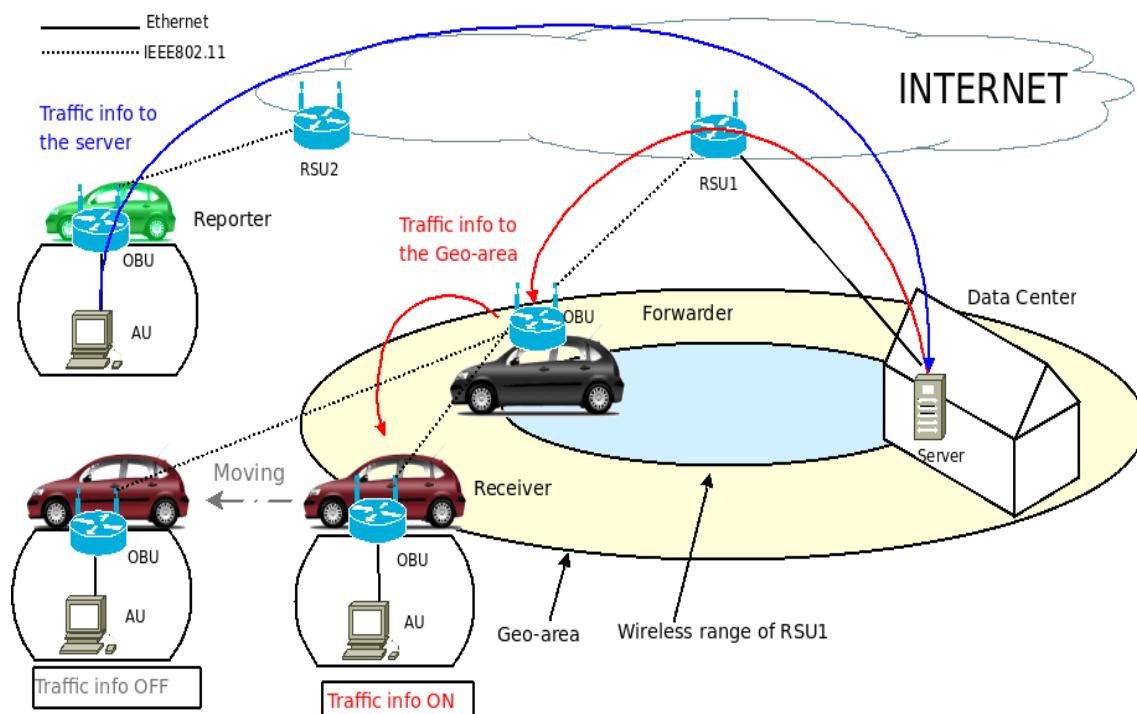


Figure 55: Scenario demonstrated at GeoNet final workshop: Network topology

- The Red vehicle displays traffic hazard as long as it is located within the yellow circle.

OBU/RSU	C2CNet layer implementation
OBU Green Car	NEC
OBU Red Car	Hitachi
OBU Green Car	Hitachi
RSU1	NEC
RSU2	NEC

Table 16: C2CNet Implementation Used on Each OBU/RSU

Vehicles used in the demonstration are configured as described in Section 3.2 with the C2CNet layer implementation indicated in Table 16. The network topology with multicast capabilities corresponds to Figure 6.

Workshop attendees were allowed to participate to the live demonstration as passengers into the green vehicle (reporting hazard) and the red vehicle (displaying hazard). Passengers in the red vehicle could directly observe that traffic hazard are effectively displayed as long as the hazard reports are received when the vehicle is in the target GeoDestination area. The demonstration ran for an hour and a half until the closure of the workshop.

The agenda, presentation material and pictures of the demonstration can be found on the GeoNet web site (slides: <http://www.geonet-project.eu/?p=223> agenda <http://www.geonet-project.eu/?p=160>).

8. Conclusion

Both experimental indoor testbed and outdoor testbed were set up to investigate the network performance on IPv6 over C2CNet. The indoor test environment was designed to evaluate the pure performance of IPv6 over C2CNet avoiding interferences due to unexpected radio perturbations. The network performance was measured with UDP, TCP and ICMPv6 traffic using *iperf* and *ping6*. In the outdoor testbed, the AnaVANET evaluation tool was developed to enable hop-by-hop performance measurement and position trace of the vehicles.

Indoor tests shows that the two implementations of C2CNet provided by HITACHI and NEC are perfectly interoperable. The experimental results show that IPv6 over C2CNet doesn't cause too much delay (less than 4ms with a single hop) and is feasible for vehicular communications. However, it also shows that the next hop IPv6 address resolution on the C2C-IP SAP causes delay overhead. The proposed *next hop IPv6 address cache* reduces the overhead but there is a trade-off between reducing the latency of the next hop resolution and packet loss probability in the case of path change.

Considering the outdoor test, we saw that IPv6 over C2CNet works according to the specification in various driving scenarios. The communication is stable even when the vehicle speed is around 100 km/h and when the relative speed between vehicles is high. The radio range is much larger than expected. The maximum distance of communication range is around 450 meters and it is not interrupted by the buildings on INRIA campus (all of them have only one floor). This call for more field tests in urban environments.

Our outdoor tests show that the performance of the current implementation under multi-hop scenarios must be improved. Currently, and under the conditions of our limited vehicular scenario, hardware and current implementation, a number of 3 hops and distance around 1500m between the vehicle and the roadside seems to be a limit. This calls for more scientific work to determine the appropriate radius of the GeoNet domain (IPv6 C2CNet link) for GeoBroadcasting IPv6 Router Advertisements.

TCP doesn't work well with the current implementation because of internal packet loss due to collision of input and output at the level of the C2C-IP SAP (UDP works because it is one way communication and ICMPv6 works in both ways because packet interval is high). To avoid the collision at the level of the C2C-IP SAP, the input and output between C2CNet layer and IPv6 layer should be maintained properly. We are considering two technical approaches to solve this problem. Firstly, a CSMA/CD (Carrier Sense Multiple Access with Collision Detection) like mechanism could be used. Secondly, two actual interfaces could be used between C2CNet and IPv6 (i.e. *tun0* and *tun1*). We will continue to investigate these approaches.

We focused on the evaluation of network performance using only one of the two C2CNet layer implementations in the outdoor test, but we plan to compare the two implementations. Also the scenarios are limited to Vehicle-to-Vehicle scenarios (vehicle-

based communication as described in [GeoNetD1.2]) in the reported evaluation but we also intend to continue the evaluation with Vehicle-to-Infrastructure scenarios (Roadside-based and/or Internet-based, referring to [GeoNetD1.2]).

The combination of IPv6 multicast and GeoBroadcast was demonstrated at the final GeoNet workshop, however we could not evaluate the performance with such a scenario. One of the reasons is that a sufficiently high number of receivers is necessary to properly evaluate multicast but experimental evaluation is limited in the number of vehicles (4 in our case). Scalability of the GeoRouting algorithm has been studied using the NCTUns emulator as reported in [GeoNetD5.1], but IPv6 is not supported by the tool so evaluation of IPv6 GeoNetworking with multicast capabilities by means of simulation or emulation is thus left for future work.

GeoNet OBUs comprise NEMO and MCoA functionalities (see the Module 3B “Mobility Support” in [GeoNetD1.2]). However in the experiments reported in this document GeoNet OBUs were equipped with only a single egress interface. The number of interfaces wouldn't impact the performance of IPv6 over C2CNet; however GeoNet OBUs equipped with multiple interfaces would allow to study new aspects not peculiar to GeoNet such as the criteria for switching between multiple paths and flow distribution.

In the evaluation, we only tested UDP, TCP, and ICMPv6 with fixed sending rate. The performance of an actual traffic hazard application such as the one used in the final GeoNet workshop is still not evaluated. In the future, the performance of IPv6 GeoNetworking should be evaluated under more realistic scenario such as this one.

To conclude, while conformance of the implementations (C2CNet layer and IPv6 over C2CNet sub-module) with the GeoNet specification [GeoNetD2.2] has been reported in [GeoNetD4.1], the live experimentation and the performance evaluation reported in the present document demonstrate the feasibility of IPv6 GeoNetworking.

Annex A: Contributors

The following people have contributed to the experiments reported in this document, by alphabetical order:

Thierry Ernst – INRIA
Maria Goleva – NEC
Ines Ben Jemaa – INRIA
Andras Kovacs - Broadbit
Hamid Menouar – Hitachi Europe
Satoru Noguchi - INRIA
Carsten Schulze – Lesswire
Manabu Tsukada - INRIA
Philippe Zhang - INRIA
Wenhui Zhang - NEC

Annex B: References

- [GeoNetD1.2] GeoNet. "Final GeoNet Architecture Design". GeoNet Deliverable D1.2, January 2010.
- [GeoNetD2.2] GeoNet. "Final GeoNet Specification". GeoNet Deliverable D2.2, January 2010.
- [GeoNetD3.1] GeoNet. "GeoNet Development Results". GeoNet Deliverable D3.1, January 2010.
- [GeoNetD4.1] GeoNet. "GeoNet Conformance Test Plan and Results". GeoNet Deliverable D4.1 version 1.1, June 2010.
- [GeoNetD5.1] GeoNet. "GeoNet Emulation Environment Results". GeoNet Deliverable D5.1 version 1.2, June 2010.
- [GeoNetD6.1] GeoNet. "GeoNet Porting and Integration Guidelines". GeoNet Deliverable D6.1, February 2010.
- [GeoNetD8.3] GeoNet. "GeoNet Handbook". GeoNet Deliverable D8.3, February 2010.
- [Santa2009] José Santa, Manabu Tsukada, Thierry Ernst, Olivier Mehani and F. Gómez-Skarmeta. "Assessment of VANET multi-hop routing over an experimental platform", Int. J. Internet Protocol Technology, Vol. 4, No. 3, 2009 Inderscience Publishers, 2009.