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

D2.2 Final GeoNet Specification

DATE	January 20th, 2010
CONTRACTUAL DATE OF DELIVERY TO THE EC	M22 – December 2009
ACTUAL DATE OF DELIVERY TO THE EC	M23 – January 2010
EDITOR, COMPANY	Andras Kovacs, BROADBIT
WORKPACKAGE	WP2 Specifications
DOCUMENT CODE	GeoNet-D.2.2-v1.1
SECURITY	Public

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.1	06/29/09	Andras Kovacs: First draft of D.2.2, editorial changes and addition of lower layer SAP description	Draft	Internal
0.2	11/03/09	Andras Kovacs: combining of new chapter contributions and general update	Draft	Internal
0.5	12/03/09	Andras Kovacs: drafting of GeoDestination Manager added	Draft	Internal
0.7	12/08/09	Carlos Bernardos: Added multicast to geo-area mapping section and security section	Draft	Internal
0.8	12/21/09	Hamid Manouar: general review and revisions	Draft	Internal
0.9	12/23/09	Andras Kovacs: general revisions	Draft	Internal
1.01	12/31/2009	Andras Kovacs: added MNPP chapter and editorial changes	Draft	Internal
1.02	1/7/2010	Andras Kovacs: added chapter on limitations and future work, editorial work	Draft	Internal
1.03	1/15/2010	Thierry Ernst: updated chapter on IPv6 Layer	Draft	Internal
1.1	1/20/2010	Andras Kovacs: editorial changes and revision of GeoDestination handling	Released doc.	Public

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

Contents

1. Executive summary

The GeoNet project aims at providing a standard solution for IPv6 geonetworking to all Intelligent Transportation Systems (ITS), relying on the IPv6 standard and on Car-to-Car Communication Consortium's (C2C-CC) geonetworking. Since the GeoNet Consortium is by no mean a standardisation body, it has been decided to elaborate a detailed, reference specification that would then be pushed to standardisation bodies (IETF, ETSI, ISO) by individual GeoNet members, many of them being active in these bodies.

This document is describing the final GeoNet specifications. It is the basis for GeoNet system implementation. Note that although this specification is public, GeoNet does not intend to provide a free solution: the two independent implementations will remain property of Consortium members. Moreover it relies on C2C-CC geonetworking solution that remains C2C-CC property and that is not fully publicly specified yet.

These specifications are based on the GeoNet architecture document [GeoNetD1.2] that describes the choices made for combining IPv6 networking and C2C-CC geonetworking capabilities into a single protocol stack for ITS and are in compliance with the network architecture as defined in ETSI TS 102 636-4. Of particular interest for the reader is the Section 1.1 of D1.2 *Links with related standardisation activities*. Also, among usual difficulties in such a convergence effort between several areas, is the precise meaning of the terms used. For this purpose, please refer to Appendix D of D1.2 *Terminology* whenever a new term appears or in case of doubt in the interpretation of some term.

The first chapters of this document specify details of service access points (SAPs), which are bounding GeoNet functional modules within the overall communication architecture. Then the functional specification of each functional module is described – their roles and inter-relations are explained in GeoNet architecture deliverable D1.2. Finally, packet formats of GeoNet layer specific message data are specified.

2. Structure of the Document

The present document is structured as follows:

- Section 3 gives an overview of the GeoNet specification.
- Sections 4-12 specify individual GeoNet modules and SAPs
- Section 13 describes the binary encoding of C2CNet layer packet headers
- Section 14 concludes this specification document by describing its limitations and planned future work
- Annex A describes IPv6 options for encoding of GeoDestination information
- Annex B gives an example of a possible SQL implementation of the MNG-UL SAP, which is provided by module 0A: GeoDestination
- Annex C discusses Service Metrics, Scalability, and Congestion Control for the GeoNet system
- Annex D lists contributors of this document
- Annex E lists the references

3. GeoNet Specification Overview

3.1. Purpose

This document comprises of module specifications and Service Access Point (SAP) specifications. The purpose of a module specification is to describe the externally testable functionalities that a specific software module shall perform. The purpose of an SAP specification is to describe the access to services that the GeoNet protocol stack offers. The specification covers the externally available SAPs at the networking endpoints of GeoNet protocol stack corresponding to OSI networking reference model.

Additionally, an overview of internal structure is provided to get an information about all dependencies. The interfaces between internal modules are specified internally and are not publicly available.

3.2. Modules and SAPs in GeoNet architecture

Detailed GeoNet architecture is described in the GeoNet Architecture document [GeoNetD1.2]. An overview of GeoNet main functional modules and interfaces is provided in Figure 1. This architecture and the role of individual modules is explained in the GeoNet Architecture document.

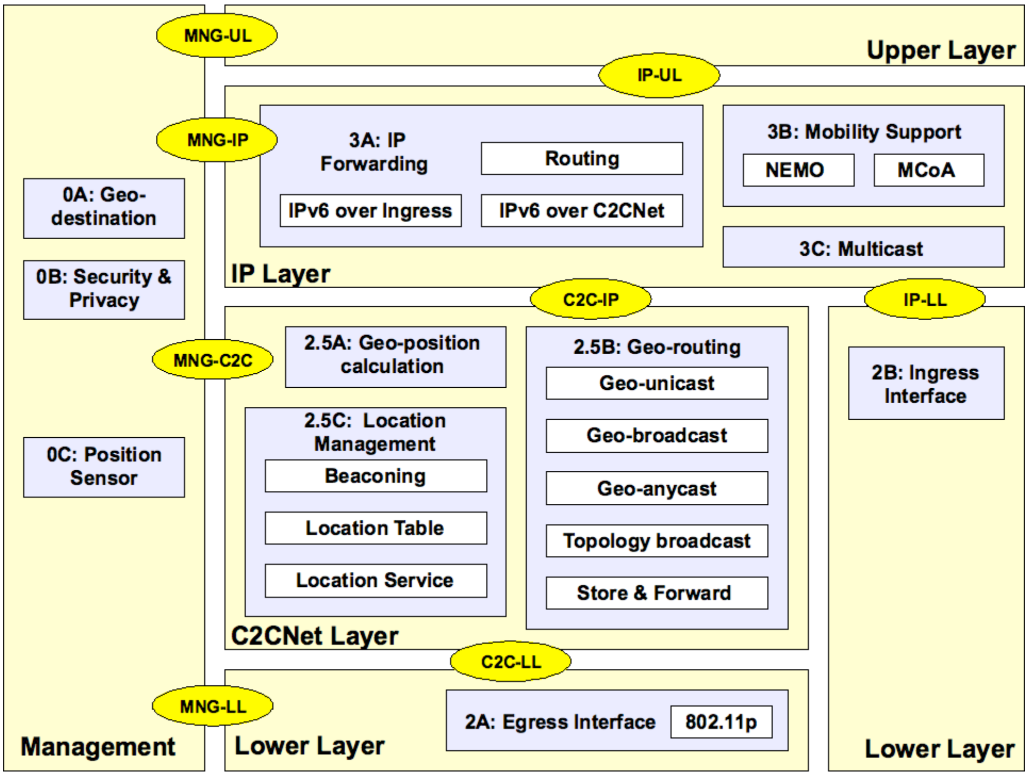


Figure 1: GeoNet main functional modules and SAPs

Each module is specified in a standalone section; its title matches the module's name. The arrangement of SAP specifications within this document is the following:

- **SAP C2C-IP** is described in section 9.2, titled 'SAP C2C-IP: Transmission of packets between IP and C2CNet layers'. This SAP is an internal one to the GeoNet system, therefore its specification is an implementation guideline without data field mapping details.

- **SAP C2C-LL** is described within section 3, which is titled 'Module 2A: Egress Interface'

- **SAP IP-UL** is using standard IPv6 socket mechanism and is therefore not detailed in this document.

- **SAP IP-LL** is using standard Ethernet link mechanism and is therefore not detailed in this document.

- **SAP MNG-IP** is not specified currently. The role and functionality of this future SAP is described in the section titled 'Module 0B: Security and privacy'.

- **SAP MNG-C2C** is described in section 12.3, titled 'Module 0C: Position Sensor'. The role of this SAP may be extended in the future by the functionality described in the section titled 'Module 0A: Geo-destination'.

- **SAP MNG-UL** is not specified currently. The role and functionality of this future SAP is described in the section titled 'Module 0A: Geo-destination'. Furthermore, Annex C describes an SQL based possibility for this SAP, which may be adopted into a future specification.

- **SAP MNG-LL** is not specified currently as a standalone SAP. The above referenced SAP C2C-LL specification presently includes the link management functionality, which would be eventually moved into the dedicated SAP MNG-LL.

The larger context of GeoNet architecture is shown in Figure 2 below. This figure highlights the scope of GeoNet project. While only IPv6 networking is currently specified and implemented on top of the geonetworking protocol, this specification may be extended in the future by a transport layer directly on top of geonetworking protocol.

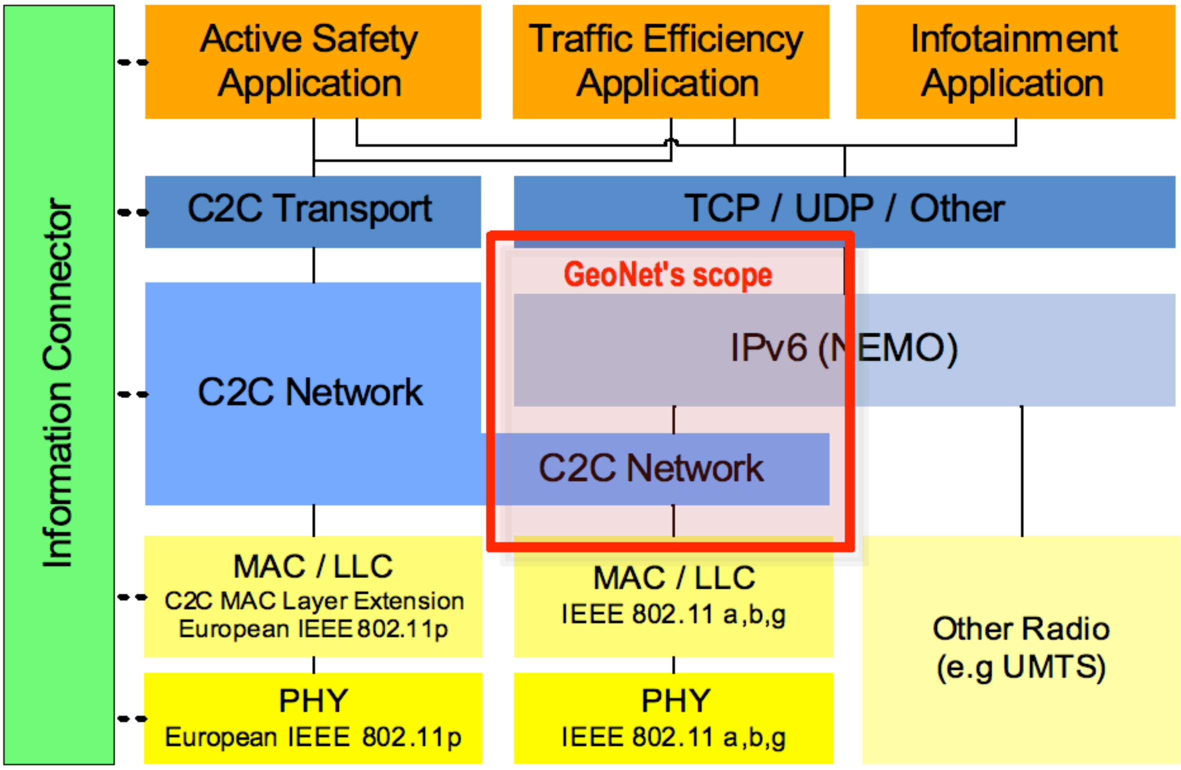


Figure 2: The scope of GeoNet within the overall ITS architecture from Car-2-Car Communications Consortium's Manifesto

All GeoNet modules may use functions and services of the Operating System.

3.3. Conventions used in this document

The following conventions apply to the present document:

- Whenever the terms AR (Access Router) or MR (Mobile Router) are used, they actually respectively refer to the terms GeoNet RSU and GeoNet OBU as defined in [GeoNetD1.2].

- The term AU (Application Unit) is used quite loosely and refers to any IPv6 node located on a link attached to the ingress interface of the GeoNet OBU (MR) or GeoNet RSU (AR). In the case of a GeoNet OBU (MR), IPv6 nodes are also referred to as “MNN” (Mobile Network Node) as indicated in [GeoNetD1.2] (section 5.2).

- Nodes attached to MR's ingress interface are referred to as Mobile Network Nodes (MNNs).

- Nodes attached to the AR's ingress interface are referred to as Correspondent Nodes (CNs) for convenience and similarity with MNNs.

4. Module 2A: Egress Interface

4.1. Overview

The SAP to data Link Layer (C2C-LL SAP) provides the service of GeoNet protocol stack to the LLC or MAC sub-layers, as shown in Figure 2. The hardware is hidden to GeoNet modules and may vary between different hardware vendors. Only the C2CNet layer may directly access the C2C-LL SAP.

The C2C-LL SAP is provided by the Egress Interface module. This module offers two basic services:

- Platform independent access to the PHY/MAC layer of ETSI ITS-G5 / IEEE802.11 and IEEE802.11p compliant devices using LLC UI frame format services. It includes the capabilities for packet sending/receiving and link status/statistics information.

- Vendor specific access to ETSI ITS-G5 / IEEE802.11p compliant hardware. This SAP is not specified by GeoNet consortium and may be subject of special license agreement.

Note: This part of PHY/MAC/LLC adaptation module is developed outside the GeoNet project and is background know-how owned by these vendors.

The C2C-LL SAP specified in this document supports both the exchange of data packets and Lower Layer management, including adjustment of channel settings and collection of channel statistics. If and when an SAP between between Management Layer and Lower Layer (MNG-LL SAP) is specified in the future, the Lower Layer management aspect would be migrated to that MNG-LL SAP.

4.2. Description of Module 2A: Egress Interface

The Egress Interface module is configurable with relation to the supported 802.11 hardware interface and the ITS protocol assigned to this interface. Multiple ITS protocols per hardware interface are permitted.

It is allowed to run multiple 802.11 hardware interfaces in parallel. Therefore this module implements addressing functionality of packets to these multiple hardware interfaces.

There are multiple sessions[1] allowed, i.e. several modules could open a session and use it in parallel to other modules.

[1] A session in the scope of this document means all operations to open, use and (later) close a raw Ethernet socket connection, initiated by a GeoNet module as a service user and maintained by the Egress Interface Module as a service provider.

The Egress Interface module supports both internal (local) management commands and data communication. Local management commands are used to configure 802.11 hardware interfaces. Packets designated for data communication are directed to the corresponding 802.11 hardware interface.

No packets are stored. All incoming packets are processed immediately.

The module will provide a stub that may be used by hardware vendors to implement a vendor specific communication protocol to their own 802.11 hardware.

4.3. Algorithm description

Routing information is delivered by C2CNet layer. Depending on this routing information, the Egress Interface module is capable to direct packets to the selected hardware. Internal packet of GeoRouting module are encapsulated into packets specific to the selected type of the C2C-LL SAP.

Packets received from MAC or LLC sub-layer are delivered to C2CNet layer.

The C2C-LL SAP supports setting and retrieving of management information from IEEE802.11 and European IEEE802.11p devices.

At GeoNet protocol stack start-up the Egress Interface module checks the available 802.11 hardware interfaces and initialises them corresponding to to provided configuration settings. If no configuration settings for initialisation are provided in the module configuration, the Egress Interface module does nothing than waiting for new sessions.

Once the Egress Interface module is initialised, it starts to operate in working mode:

- GeoNet internal packets are received from C2CNet layer. After receiving a packet, the local destination 802.11 hardware interface is read from C2C-LL SAP packet header as well as the ITS protocol to be used for air transmission. The Egress Interface module extracts the payload and compiles (if needed) the packet required by the selected ITS protocol. Afterwards the the packet is immediately sent to the local 802.11 hardware interface.

- External packets received by C2C-LL SAP interface from data link layer are directed to C2CNet layer.

Note: In first implementation step only the C2C-CC protocol and only one 802.11 hardware interface is supported. Subsequent extensions will support multiple 802.11 hardware interfaces and multiple ITS protocols.

Figure 3 shows further details of Egress Interface module structure in GeoNet context.

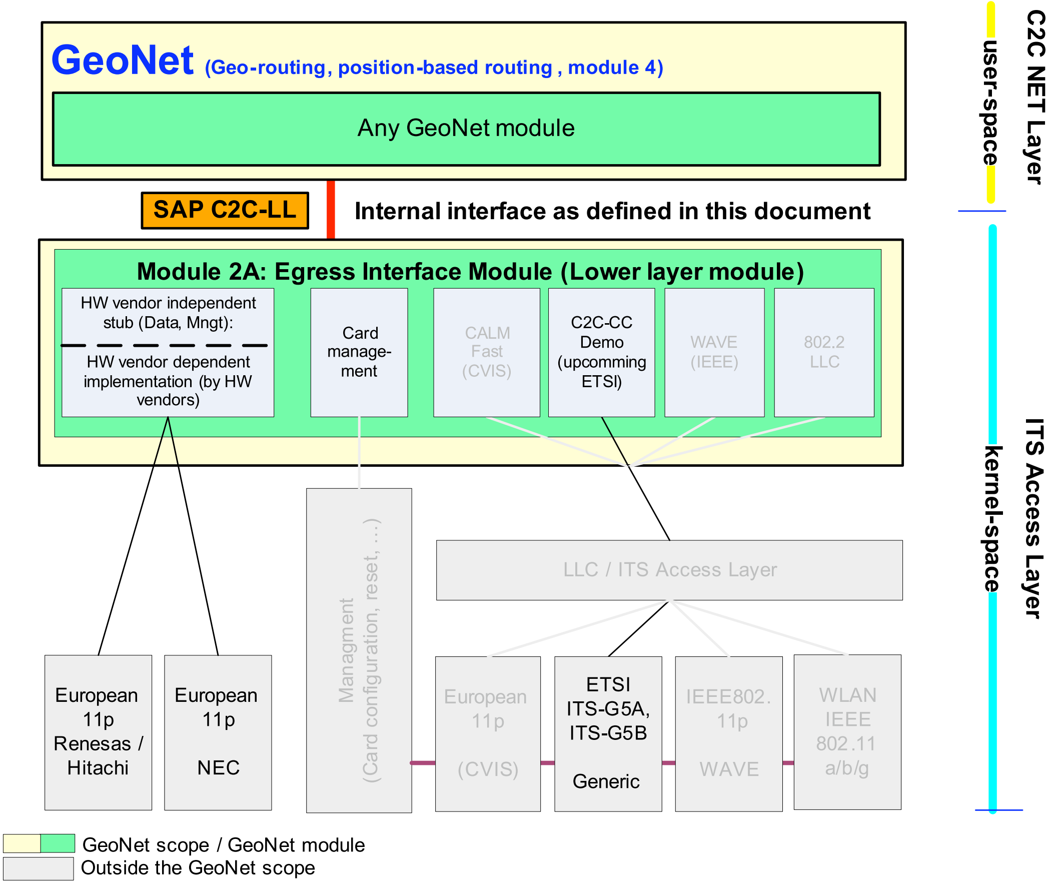

GeoNet (Geo-routing, position-based routing , module 4)
Any GeoNet module
SAP C2C-LL
Internal interface as defined in this document
Module 2A: Egress Interface Module (Lower layer module)
HW vendor independent stub (Data, Mngt):
HW vendor dependent implementation (by HW vendors)
Card manage-ment
CALM Fast (CVIS)
C2C-CC Demo (upcomming ETSI)
WAVE (IEEE)
802.2 LLC
Managment (Card configuration, reset, ...)
LLC / ITS Access Layer
European 11p Renesas / Hitachi
European 11p NEC
European 11p (CVIS)
ETSI ITS-G5A, ITS-G5B Generic
IEEE802. 11p WAVE
WLAN IEEE 802.11 a/b/g
C2C NET Layer
user-space
ITS Access Layer
kernel-space
GeoNet scope / GeoNet module
Outside the GeoNet scope

Figure 3: Harmonization of GeoNet modules via common adaptation layer that is provided in the Egress Interface module

4.4. Implementation

Configuration information is stored in a configuration file divided by keyword and corresponding entries. This configuration is read at module start containing the number and type of present HW modules, the port number of listening Ethernet raw socket and the supported ITS protocol.

The served interface to the C2CNet layer is implemented as an Ethernet raw socket capable to support multiple sessions. Depending on the type, the packets from GeoNet will be transformed to either management settings or data packets. Packets coming from HW are transformed and directed to C2CNet layer. Egress Interface module supports basically two types of lower layer transformation; conversion of C2CNet packet format specified in this document to/from

- C2C Demo packet format
- CVIS (CALM Fast) packet format.

The WAVE 802.11p and the standard IEEE802.2 packet format will be implemented on request of other GeoNet partners.

In order to support non-standardised, vendor specific cards with proprietary software interfaces, a special software module – a so called stub – shall be provided. In this stub the internal interface to GeoNet modules is implemented. The interface to the specific hardware may be implemented by hardware vendors themselves. In this way the implementation of thin and very effective vendor specific interfaces is enabled.

Limitations of preliminary implementation:

- Only one HW module is supported

- Only packets with Ethertype 0x0707 (C2C Demo 2008) are supported

4.5. Format of packets exchanged over C2C-LL SAP

Packets sent over C2C-LL SAP are of the following types:

Type	Description
0	Data field contains payload to be sent and the necessary header.
1	Initialization: The given interface is initialized to the default settings.
2	Set the Mac ID.
3	Set the observation time for the channel load.
4	Request channel information . The answer will be a packet of type 34 (get-MACID)

Packets sent from the Lower Layer are of the following types:

Type	Description
32	The packet contains received data and information about the sender and target MAC-ID. That packet will be sent after initialization, setting a new MAC-ID and as response to a request channel information)
33	Information about the current channel load. This packet will be sent depending on the time set by the observation time.
34	Contains a received (from MAC/PHY) data packet.

4.5.1. Packet containing data to be sent:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0        |   channel     |    tx power   |    data rate  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  priority     |   reserved    |   reserved    |    reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     destination MAC ID
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |    reserved   |    reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        source MAC ID
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |    reserved   |      reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| payload ... (max 65512 bytes)                                 |
```

Figure 4: Packet containing data to be sent

channel	Channel in which to send the payload
tx-power	Transmit-power to use for sending
data rate	Data rate to use (not possible to change on some interfaces, then ignored)
Priority	The priority of the packet (ignored if there is no difference)
destination MAC	Mac-address of the destination (broadcast address may be used in most situations)
source MAC ID	The MAC address to use for transmit
Payload	Message sent

4.5.2. Initialisation packet:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      1        |default channel|    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   | reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          name of the interface ...(max 108 bytes)             |
```

Figure 5: Initialisation packet

channel	Channel in which to send the payload
tx-power	Transmit-power to use for sending
data rate	Data rate to use (not possible to change on some interfaces, then ignored)
Priority	The priority of the packet (ignored if there is no difference)
destination MAC	Mac-address of the destination (broadcast address may be used in most situations)
source MAC ID	The MAC address to use for transmit
Payload	Message sent

4.5.3. Packet for setting the MAC ID:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      2        |    channel    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 local MAC ID to be set
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |  reserved     |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: setting the MAC-ID

channel	Channel to set the MAC id to use for
local MAC	Mac address to use

4.5.4. Packet for setting channel observation time:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      3        |    channel    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 observation time to use in ms                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: setting channel observation time

channel	Channel to set the observation time for channel load
observation time	Time used for observation

4.5.5. Packet for requesting channel information:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      4        |    channel    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: requesting channel information

channel	Channel to request information about

4.5.6. Packet for receiving information about the current MAC-ID:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      32       |    channel    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         MAC ID used
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Information about the current MAC-ID

channel	The channel that uses the MAC – ID
MAC-ID	The MAC ID used in this channel

4.5.7. Packet for receiving information about channel load:

```
  0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      33       |    channel    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  reserved     |   reserved    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               channel load  (measurement unit tbd.)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Information about observed channel load

channel	The channel whose load is being reported
Channel load	Channel utilization measurement value, for example percentage of carrier sensing time

4.5.8. Received data packet:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      34       |    channel    |   RSSI/RCPI   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   reserved    | reserved      |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         destination MAC ID
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        source MAC ID
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| payload... (max 65512 bytes )                                 |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Received data packet

Channel	channel in which the message was received
RSSI/RCPI	The value of the RSSI or RCPI (if provided by the MAC, if not it is left 0)
destination MAC	Mac-address of the destination (own mac address or broadcast)
source MAC ID	The MAC address to use for transmit (Mac address of the source)
Payload	Message received

5. Module 2B: Ingress Interface

There is no individual specification provided by GeoNet since state-of-the-art LLC implementations are used.

6. Module 2.5A: Geo-Position calculation

6.1. Overview

Module “2.5A: Geo-position calculation” describes the calculation of position information for Location Table, C2CNet header and broadcast geo-area representation. This module is also responsible for geo-area relevance check.

The functionalities described in this chapter may be called and utilised by any other module.

6.2. Position processing before saving sensor data

There is no filtering of GNSS sensor output in this specification.

The purpose of future sensor output processing, that will be added here is to filter out GNSS-positioning jitters before saving position data acquired through the position sensor into the Location Table. Beacons and routing algorithms then work with more reliable position information.

6.3. Position description for unicast packets

Address format in source and destination messages conforms to address format of C2CNet layer messages. Destination position is represented by Latitude and Longitude. Source position is represented by Latitude, Longitude, Altitude, Heading, Speed, and accuracy estimates of each parameter. The required resolution of the position data:

* Latitude: 1/8 microdegree

* Longitude: 1/8 microdegree

* Altitude: 1 meter

* Heading: 0.005493247 degrees

* Speed: 0.01 meters per second

6.4. Distance calculation between two nodes

Distance calculation takes the latitude/longitude value pairs of any two nodes, and converts this data into a distance value expressed in meters. The calculation is based on the Haversine formula [Haversine], and should be performed to double precision type. If the two nodes coordinates are (lat1, long1) and (lat2, long2), their distance **d** is calculated through following method:

R = Earth's radius *Note: mean radius = 6371000 m*

$\Delta lat = lat2 - lat1$

$\Delta long = long2 - long1$

$a = \sin^2(\Delta lat/2) + \cos(lat1) \times \cos(lat2) \times \sin^2(\Delta long/2)$

$c = 2 \times \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$ *Note: atan2(y,x) calculates the arctangent of y/x*

$\mathbf{d} = R \times c$

6.5. Vehicle centered cartesian coordinate conversion

If the local implementation of Location Table uses vehicle centred cartesian coordinates, a transformation of other nodes' latitude/longitude values into this coordinate system may be needed. The calculation is based on the Haversine formula [Haversine]. The first step is to calculate distance d and bearing θ of the other vehicle. Distance calculation formula has been given in previous paragraph. If own coordinates are (lat1, long1) and the other vehicle's coordinates are (lat2, long2), the bearing **θ** is calculated through following formula:

$\Delta lat = lat2 - lat1$

$\Delta long = long2 - long1$

$\boldsymbol{\theta} = \operatorname{atan2}(\sin(\Delta long) \times \cos(lat2) , \cos(lat1) \times \sin(lat2) - \sin(lat1) \times \cos(lat2) \times \cos(\Delta long))$

Note: this angle is radian measure clockwise from North

Assuming that in the local cartesian coordinates **x** points to the East and **y** points to the North, their definitions are:

$\mathbf{x} = d \times \sin(\theta)$

$\mathbf{y} = d \times \cos(\theta)$

7. Module 2.5B: Geo-routing

7.1. Overview

In vehicular networks, data are geo-routed from a source to a destination. GeoRouting protocol layer uses geographic position and movement information of vehicles to route data. Depending on the destination type, several GeoRouting schemes may be used. For example, to reach a node in the network for which we already know the exact geographical location (geo-location), we need just to forward the data towards the geo-location of that destination node. If the destination we want to reach all nodes belonging to specific geographical area (geo-area), then we should forward the data in an efficient way to disseminate all nodes in this geo-area.

A geo-area could have different shapes: circular, rectangular, eclipse, etc. In GeoNet project, we limit the geo-area shapes to just circle, which can be represented means of a centre point (Latitude, Longitude) and a radius.

For the sake of simplicity we consider in GeoNet project only sender-based georouting schemes. But, receiver-based georouting schemes are not excluded and could be considered in future works.

In GeoNet project we consider four georouting schemes: GeoUnicast, GeoAnycast, GeoBroadcast, and TopoBroadcast. When receiving an IPv6 packet through C2C-IP SAP, module "2.5B: Geo-routing" selects the appropriate georouting scheme to be used to forward that packet. This selection is based on the destination information, as provided from the SAP. Module "2.5B: Geo-routing" may consult the management module "0A: Geo-destination" in order to understand the intended destination area. The four schemes/ protocols supported by GeoNet system are described in detail in the following sections.

Note: the Next Header field must be set to 0x44[2] value when IPv6 layer is used over the C2CNet layer.

7.1.1. GeoUnicast

GeoUnicast refers to the routing protocol which, based on position and movement information of involved nodes, routes data from a source node to a destination node for which the exact geographical location is known. This corresponds to point-to-point scenario.

GeoUnicast protocols use a forwarding mechanism to route packets through intermediate nodes till reaching the destination location. As forwarding mechanism, the Greedy Perimeter Stateless Routing (GPSR) protocol [Karp2000] is adopted in GeoNet project.

[2] This value has been chosen randomly. C2C-CC should assign an appropriate number for it.

GPSR makes greedy forwarding decisions using only information about immediate neighbours. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region.

The greedy forwarding consists on selecting as next forwarder the neighbour which is the closest to the destination. In a future extension of GeoNet specification, we may consider more than distance in the selection of the next forwarder. For example, movement direction and speed of vehicles may be used [Menouar2007].

7.1.2. GeoAnycast

GeoAnycast refers to the routing protocol which, based on position and movement information of involved nodes, routes data from a source node to any node located within a specific geographical area. This corresponds to Geoanycast scenario.

Compared to GeoUnicast, GeoAnycast has only one difference which belongs in the destination nature. GeoAnycast, as GeoUnicast, targets one destination node, but not defined as destination in advance. In fact, the destination in GeoAnycast is the first node reached in a specific geographical area.

Within GeoNet, a adapted version of GeoUnicast protocol is used for GeoAnycast. The main adaptation belongs in the fact that each node, when receiving a GeoAnycast, first checks either it is located within the destination geo-area or not. If it is, then it considers itself as destination, otherwise, if it is the next forwarder, it forwards the packet towards the destination area.

7.1.3. GeoBroadcast

GeoBroadcast refers to the routing protocol which, based on position and movement information of involved nodes, delivers data from a source node to all nodes located within a specific geographical area. This corresponds to GeoBroadcast scenario.

In GeoBroadcast, the source node may be located inside or outside of the targeted geo-area. If the source node belongs to the destination geo-area, then the broadcast packet should be just broadcast in this area. If the source node does not belong to the destination geo-area, then the packet should be forwarded until reaching a node which belongs to the destination geo-area, which takes care on broadcasting the packet to all nodes located within this destination geo-area.

Within GeoNet project, GeoAnycast is used to reach the first node which belongs to the broadcast geo-area, and then, a simple broadcast mechanism is used to deliver the packet to all nodes located in the destination geo-area. A basic flooding mechanism is used, where each node located in the broadcast geo-area forward the packet once.

More advanced broadcast mechanisms could be defined in the future, such as MHVB (multi- hop vehicular broadcast) [Mariyasagayam07a, Mariyasagayam07b].

7.1.4. TopoBroadcast

TopoBroadcast (topology broadcast) refers to the routing protocol which, based on network topology information, routes data from a source node to all nodes located up to a specific distance in terms of hops. This corresponds to point-to-multipoint scenario.

In GeoNet project, a basic flooding mechanism is used to disseminate all nodes up to a desired hop distance. The Hop Limit in the common network Header is used to limit the flooding to the distance (in terms of hops) we want to broadcast.

7.2. Algorithm description

Let us suppose we have a source node S which receives data from upper layer to be delivered to a specific destination. This destination may be:

- a specific vehicle, then GeoUnicast is used,
- any vehicle located in a specific geographical area, then GeoAnycast is used,
- all vehicles located in a specific geographical area, then GeoBroadcast is used,
- or all vehicle located at a certain distance in terms of hops, then TopoBroadcast is used.

7.2.1. GeoUnicast

Sender part

1. S generates a GeoUnicast packet P (See Section 9. : C2CNet packets format)
2. if there is a fresh entry in Location_table which says that D is a direct neighbour, then send P to D (D should be within the communication range)
3. else, if the position of D is already provided by the upper layer or known from the location management table, then Dest_GeoLocation = geographical location of D
4. else, ask the Location_management (Location Service) to provide the geo-location of D (Dest_GeoLocation = geographical location of D).
5. if there are neighbours available around, then:
   - for each neighbour i, calculate Dist(i, D) which corresponds to the distance from the location of i to Dest_GeoLocation.

- send P to the neighbour i which has shortest distance Dist(i, D), where Dist(i, D) < Dist(S, D)

6. else, put P in the store and forward buffer

Receiver part

Node j receives through a forwarder F GeoUnicast packet P which has been initialy sent from a source node S. The packet P is destined to node D.

j proceeds as follows:

1. if security check is failed, then j drops P.

2. j checks either the same packet P has not been processed previously by checking the Source node ID and Timestamps. If the packet P has been already processed, then it is dropped.

3. j updates its location table by updating/adding the two entries that correspond to S and F respectively.

4. if j is the destination D, then it delivers the payload of P to upper layer.

5. else, if there is a fresh entry in Location_table which says that D is a direct neighbour, then send P to D (D should be within the communication range). The position vector in the common header of P is updated with the j's position information before retransmitting P.

6. else, if there are neighbours available around, then:

   - for each neighbour i calculate Dist(i, D) which corresponds to the distance between the location of the neighbour i and Dest_GeoLocation.

   - send P to the neighbour i which has shortest distance Dist(i, D), where Dist(i, D) < Dist(j, D). The position vector in the common header of P is updated with j's position information before retransmitting P.

7. else, put P in the store and forward buffer

7.2.2. GeoAnycast

In following we consider only circle shape as destination geo-area. The destination geo-area is a circle of a centre D and a radius R.

Sender part

1. S generates a GeoAnycast packet P (See Section 9.)

2. if S is located within the destination geo-area, then do nothing (S is already the destination)

3. else, the packet should be forwarded towards the destination location (Destination Latitude, Destination Longitude). Therefore, S proceeds as follows.

4. if there are neighbours available around, then:

   - for each neighbour i, calculate Dist(i, D) which corresponds to the distance from the geo-location of i to the geo-location of the destination area centre (Destination Latitude, Destination Longitude).

   - send P to the neighbour i which corresponds to shortest distance Dist(i, D), where Dist(i, D) < Dist(S, D).

5. else, put P in the store and forward buffer

Receiver part

Node j receives through a forwarder F GeoAnycast packet P which has been initially sent from a source node S. The packet P is destined to any node located within a geo-area of circle shape with D as centre and R as radius.

j proceeds as follows:

1. if security check is failed, then j drops P.

2. j checks either the same packet P has not been processed previously by checking the Source node ID and Timestamps. If the packet P has been already processed, then it is dropped.

3. j updates its location table by updating/adding the two entries that correspond to S and F respectively.

4. if j belongs to the destination geo-area, then it delivers the payload of P to upper layer (j is the Destination.

5. else, if there is a fresh entry in the Location_table which corresponds to a direct neighbour which belongs to the destination geo-area, then forward P to this neighbour. The position vector in the common header of P is updated with j's position information before retransmitting P.

6. else, if there are neighbours available around, then:

- for each neighbour i, calculate Dist(i, D) which corresponds to the distance from the location of i to the location of the destination area centre (Destination Latitude, Destination Longitude).

- send P to the neighbour i which corresponds to shortest distance Dist(i, D), where Dist(i, D) < Dist(j, D). The position vector in the common header of P is updated with j's position information before retransmitting P.

7. else, put P in the store and forward buffer

7.2.3. GeoBroadcast

In following we consider only circle shape as destination geo-areas. The destination geo-area is a circle of a centre D and a radius R.

When an IPv6 multicast packet is received through the C2C-IP SAP of the sending node, the destination geo-area centre D shall be set to the current position of this originating node. The packet's IPv6 multicast address is statically matched to a corresponding configuration directive, which describes the assigned radius R.[3]

Example of 500m and 1000m range assignments to IPv6 multicast addresses:

IPv6 multicast address	GeoBroadcast range (m)
ff1e:0:0:0:0:0:0:1	500
ff1e:0:0:0:0:0:0:2	1000

Note, that this simple geo-area assignment method may be extended in the future, when some of the possible IPv6 GeoDestination encoding mechanisms described in Annex A are taken into use.

Sender part

1. S generates a GeoBroadcast packet P (See Section 9) and then sets the GeoBroadcast destination geo-area. The distance in the extended header (Distance) is set to the area circle radius and the latitude and longitude fields are set to the area centre coordinates.

2. if Dist(S, D) ≤ R (i.e. S belongs to the destination geo-area), then S sends out the packet in a broadcast mode.

[3] The expected future extension to this specification is that when an IPv6 multicast packet is received through the C2C-IP SAP of the sending node, the format of the GeoDestination information provided through the SAP C2C-IP differs according to the encoding approaches discussed in the Annex A, and may require to enquire additional information from module "0A: Geo-destination" through SAP MNG-C2C.

3. else (i.e. S does not belong to the destination geo-area), the packet should be forwarded towards D. In this situation S proceeds as follows:

   - if there are neighbours available around, then for each neighbour i calculate Dist(i, D), and then send the packet P to the neighbour i which corresponds to shortest distance Dist(i, D), where Dist(i, D) < Dist(S, D).

   - else put P in the store and forward buffer

Receiver part

Node j receives through a forwarder F a GeoBroadcast packet P which has been initially sent from a source node S. The packet P is destined to all nodes located within a geo-area of circle shape with D as centre and R as radius.

j proceeds as follows:

1. if security check is failed, then P is dropped.

2. j checks either the same packet P has not been processed previously by checking the matching of Source node ID and Timestamp value. If the packet P has been already processed, then it is dropped.

3. j updates its location table by updating/adding the two entries that correspond to S and F respectively.

4. if j does not belong to the destination geo-area, it shall drop any packet which comes from a forwarder which is located inside the destination geo-area. So if Dist(F, D) ≤ R and Dist(j,D)>R, then P is dropped.

5. else, if Dist(j, D) ≤ R then j is considered as destination. Therefore j delivers the payload of P to upper layer, and then sends out the packet in a broadcast mode.

6. else, if Dist(F, D) > R, the packet should be forwarded towards D. In this situation S proceeds as follows:

   - if there are neighbours available around, then for each neighbour i calculate Dist(i, D), and then send the packet P to the neighbour i which corresponds to shortest Dist(i, D), where Dist(i, D) < Dist(F, D). The position vector in the common header of P is updated with the j's position information before retransmitting P.

   - else put P in the store and forward buffer

7.2.4. TopoBroadcast

Sender part

1. S generates a TopoBroadcast packet P (See Section 9) and then set the hop limit field in the common network header to the desired broadcast distance in terms of hops.

2. S broadcasts P.

Receiver part

Node j receives through a forwarder F a TopoBroadcast packet P which has been initially sent from a source node S. The packet P is destined to all nodes located up to N hops.

j proceeds as follows:

1. if security check is failed, then j drops the packet P.

2. j checks that same packet P has not been processed previously by checking the matching of Source node ID and Timestamp value. If the packet P has been already processed, then it is dropped.

3. j updates its location table by updating/adding the two entries that correspond to S and F respectively.

4. j delivers the payload of P to upper layer.

5. j decreases by 1 the Hop Limit value in the common header of P.

6. if the new Hop Limit in the common header of P is zero (Hop Limit=0), then P is dropped.

7. If the new Hop Limit in the common header of P is different from zero, then j updates the position vector in the common header by putting its own information, and then retransmits the packet P in a broadcast mode.

7.3. Store and Forward Buffer

7.3.1. Overview

Store and Forward Buffer receives either unicast or broadcast messages for holding until a suitable forwarder candidate appears. The reason for storing messages in this buffer is that there are no neighbours which would bring the message closer to its destination. This may change either by appearance of a new neighbour on communication range, or by the change of the direction of an existing neighbour since message deposition by over a threshold angle – this latter event being defined as 'change in the directionality'. Therefore new neighbour events and changes in the directionality of existing neighbours form triggers for re-assessing whether there is a suitable forwarding candidate for cached messages. These triggers initiate passing of buffered messages to the respective sub-module, i.e. the georouting scheme, which has originally deposited the message. Location table is periodically queried for re-assessing neighbour data. In case the buffer gets full, it appropriately selects messages with less criticality/priority for removal.

7.3.2. Algorithm description

1. Store and Forward Buffer is storing deposited GeoRouting packets until a trigger event defined in step 2. Messages are discarded from buffer upon expiration of validity timer.

2. Data of Location Table is periodically queried for the evaluation of new neighbour events and change of directionality events. A new neighbour event is defined as the appearance of a new direct neighbour in the location table. A change of directionality event is defined as a larger than threshold angle difference between current and initial relative direction of any direct neighbour. Initial relative direction of a neighbour node is saved when it is registered into the location table, and this initial value is updated to current value at each directionality event by that node.

3. All packets in the buffer are sequentially delivered to originating sub-module. It is the responsibility of those sub-modules to evaluate per message whether it is forwarded to wireless interface or sent back to buffer via step 1.

7.3.3. Description of SAPs

Input SAP: receives message pushing with following parameters:

* originating sub-module

* message datagram

* expiration timestamp

Each buffer entry stores above parameters.

Output SAP: message is pushed back to originating sub-module

SAP to Location Table: query is specified on the Location Table side

7.3.4. Listing of parameters

Maximum number of buffered entries (implementation dependent value)

Periodicity of querying Location Table (0.5 s default value)

Threshold angle change for 'change of directionality' trigger (90 degree default value)

8. Module 2.5C: Location Management

8.1. Data store at each node

8.1.1. Location table

Each node maintains a location table including location related information for itself and a list of its neighbouring nodes.

The location related information of itself is updated locally by a positioning system, some sensors, or some functions dealing with location information.

Entries in the location table for neighbouring nodes may be added and updated in case any types of C2C packets are received which have the common C2CNetwork header.

The location information for each entry shall include at least all the position information in the common C2C header, it is called position vector, and its has the following data fields:

- MAC id
- C2CNet id
- Timestamp
- Position in latitude, longitude and altitude along with their accuracy
- Speed and header along with their accuracy

The data type and format of the location information correspond to those in the common C2CNet header.

Each entry may also include other information, and it is implementation specific.

Entries in location table are considered invalid after the timeout interval to avoid outdated location information.

8.1.2. Parameters

Location table entry expiration interval: 5s

8.2. C2CNet Beacon

8.2.1. Overview of beacon protocol

The beacon protocol is used to exchange information between nodes about the positions of their neighbours.

A beacon is sent periodically at distinct intervals. A beacon contains the position of the node and the node's identifier, etc. as detailed by C2C-CC in the common network header [EtsiC2CDemo2009].

A node that receives a beacon locally stores the information and assigns a lifetime. When the lifetime expires, the node assumes that the neighbour is no longer present and removes the information.

In the basic scenario, beacons are sent using single hop broadcast with the following data values in the common header:

- Protocol type: 1
- Protocol subtype: 0
- Hop limit: 1
- Length is set to zero

More advanced beacon scenarios (e.g. multi-hop beacons) will be specified in the future.

8.2.2. Source operation

In the basic operation, each node sends beacons at the frequency defined by the beacon interval.

Upon the arrival of the scheduled time, the sender will read its own information from the location table, assemble a beacon packet, and send out to the lower layer. Then the sending time for the next beacon will be scheduled.

If security is enabled, corresponding security header/trailer will be added before sending

Failed sending will lead to error report to the system.

In advanced operation, the beacon interval, the transmission power may be adapted according to channel load and/or other network related parameters. The beacon itself may also be re-scheduled if an application message has been sent recently. Detailed security features may also be included. These are to be specified in the future.

8.2.3. Receiver operation

When a node receives a beacon, it performs the following processing steps:

·If security check fails, the packet will be dropped.

·If the beacon comes from a node that is not included in the location table, a new entry will be created for this node, and location information from the common C2CNet header will be written into the entry. The MAC ID of the source node as given from C2C-LL SAP is also added to the created entry.

·If the beacon comes from a node that is already included in the location table, the location information will be updated with information from the common C2CNet header and from the C2C-LL SAP (Source MAC ID).

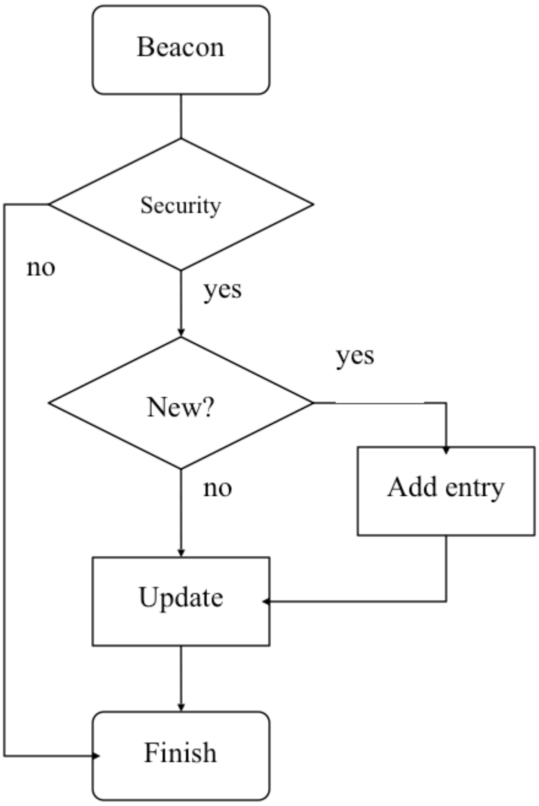


Figure 12: Receiver operation

.

8.2.4. Parameters

Beacon interval: 0.5s

Transmit power: 20 dBm

Location entry life time: 10s

8.3. Location service

8.3.1. Overview of location service

In order to send data from a source node to a destination node using position-based routing, the source node must determine the position of the destination node. This functionality is provided by the location service protocol.

In principle, the location allows a node to query the geographical position of another node when its node identity is known and to maintain a location table that contains the knowledge about other nodes positions.

Figure 13 represents a simplified sequence chart with four nodes. Node 1 broadcasts a location request packet to the neighbour node 2 and 3 in order to locate node 4. Node 3 re-broadcasts the requests to node 4 and node 4 issues a location reply back to node 1 in a unicast mode, and node 1 sends data packet afterwards.

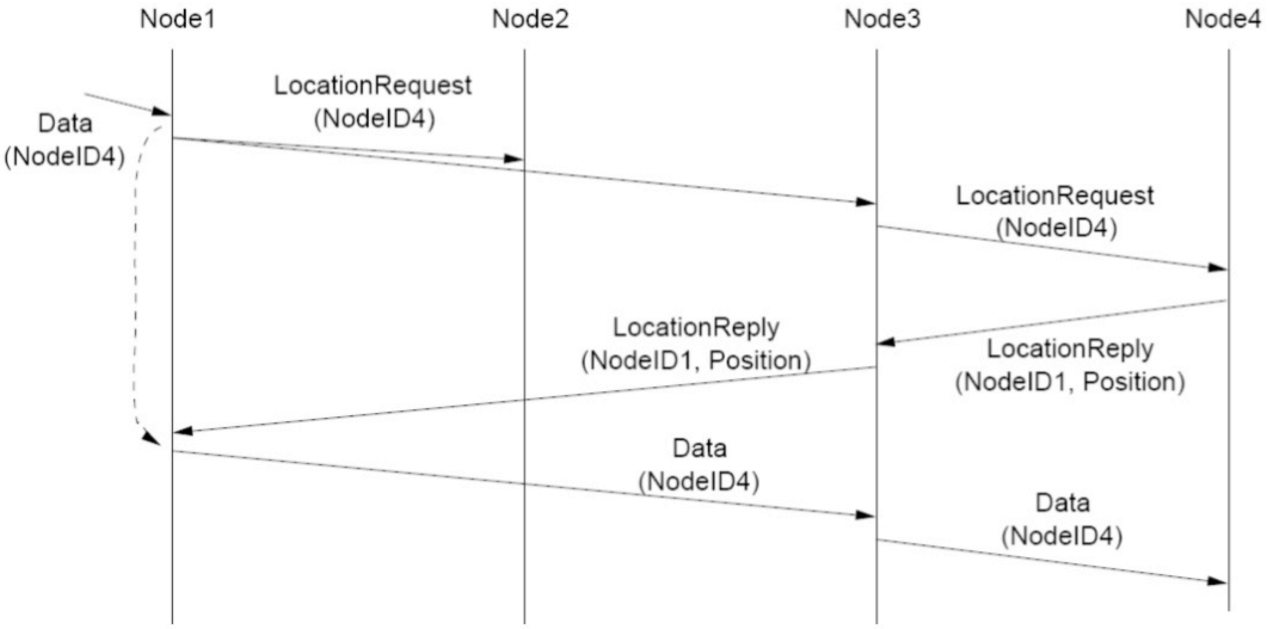


Figure 13: Location service

8.3.2. Source operation

The source node works as follows: it issues a location request packet in case a packet is to be forwarded to a node with unknown position. The data packet that has triggered the location request is buffered until the location is determined.

The location request packet is sent by broadcast to all neighbours.

If location reply is received, the source will add the location of node in the location table and send out the buffered data packets. If further location reply with older timestamps received from the same node, they are dropped.

If no location reply is received, a time-out triggers the initiation of another request cycle up to a pre-defined number of times. If still no location reply is received, a location unavailable message is sent to the corresponding application.

8.3.3. Sender operation

When a node receives a location request packet, it checks whether the packet is not destined for the node itself and it has not been received the message yet (using source identifier and timestamp). If both conditions are met and the Hop Limit in the packet is larger than zero, the node will reduce the Hop Limit in the packet and forward the packet with its own position vector set in the sender position vector in the packet.

Each time the sender receives a Location Request or Reply, it shall update in its location table the entries which correspond to the source and the sender nodes respectively.

Infinite packet looping and duplications due to flooding are avoided by keeping track of the processed location request messages at each node by means of a timestamp and source identifier.

If security is enabled, each node first checks the security data field and drops the packet if it does not comply with security requirements.

8.3.4. Receiver operation

When a node receives a location request packet, it checks whether the packet is destined for the node itself and it has not been received the same packet yet. If both conditions are met, it replies with a location reply message.

Each time the receiver receives a Location Request or Reply, it shall update in its location table the entries which correspond to the source and the sender nodes respectively.

If a node encounters a location request that has already been received (using source identifier and timestamp), the request is dropped. The location request packet is also discarded, if the hop limit is reached.

8.3.5. Parameters

Location request time out: 5s
Location request retry times: 5

9. Module 3A: IPv6 Forwarding

This module is responsible for IPv6 packet assembly and forwarding. It communicates with other layers through SAPs “C2C-IP”, “MNG-IP”, “IP-UL” and “IP-LL”.

Implemented in all IPv6 nodes, this module acquires necessary IP parameters for communications such as IPv6 addresses and prefix information. It performs common IPv6 functions such as IPv6 address configuration, IPv6 packet generation and packet forwarding and routing. It also enables IPv6 to run over different lower layer technologies, particularly C2CNet. As defined in [GeoNetD1.2] (section 6) and shown on Figure 1, module “3A: IP Forwarding” includes three sub-modules:

- **IPv6 over C2CNet:** Implemented in GeoNet nodes only, this sub-module enables GeoNet OBUs and GeoNet RSUs to support geonetworking and is in charge of delivering efficiently a packet to its destination over the C2CNet link. It acquires necessary IP parameters such as IPv6 address and performs IP next hop determination and IP address resolution over the C2CNet link in order to communicate with nearby GeoNet OBUs and GeoNet RSUs. It performs Neighbour Discovery [RFC4861] and Stateless Address Auto-configuration [RFC4862]. The behaviour of Neighbour Discovery has to be modified and varies whether it is implemented in a GeoNet RSU or a GeoNet OBU. Specifically, the behaviour of Neighbour Discovery must be extended in order to determine what is the IPv6 address space belonging to nearby vehicles (see Section 9.4.5 “Mobile Network Prefix Provisioning“). This sub-module is specific to the combination of IPv6 and geonetworking. It communicate internally with the routing sub-module and with module “2.5B Geo-routing” through SAP C2C-IP. For privacy reasons, this sub-module also interact with module “0B: Security & Privacy” to dynamically change the C2CNet ID used to configure IPv6 addresses on the C2CNet interface.

- **IPv6 over ingress:** this sub-module is not specific at all to IPv6 geonetworking but is a required component in some deployment cases (i.e. when the GeoNet OBU or GeoNet RSU has nodes directly attached to an ingress interface). This sub-module allows the GeoNet node to play on the ingress interface the role of a router as defined in [RFC4861]. It communicate internally with the routing sub-module and with module “2B: Ingress Interface” through SAP IP-LL.

- **Routing:** this sub-module transmits packets internally between ingress and egress interfaces and possibly to/from upper layers through SAP IP-UL. It receives routing tables updates through its interfaces.

Routing and **IPv6 over C2CNET** sub-modules must be supported in all implementations of GeoNet OBUs and RSUs, but they function differently in a GeoNet OBU and a GeoNet RSU. Support of **IPv6 over ingress** is needed only when GeoNet OBU and GeoNet RSU have nodes attached via an ingress interface.

9.1. IPv6 C2CNet link management

IPv6 operates over the C2CNet layer (on the C2CNet egress interface of GeoNet OBUs and GeoNet RSUs). Therefore, the C2CNet layer plays the role of a sub-IP layer. The physical wireless access link layer (i.e. IEEE 802.11p in the context of the GeoNet project, but possibly another one) is not visible to IPv6. IPv6 packets are sent down to the C2CNet layer and encapsulated using the C2CNet protocol. On the IP next hop, the C2CNet layer removes the C2CNet header and delivers the packet up to the IPv6 stack. Two IPv6 neighbours might be separated by more than one wireless hop, but this is transparent to IPv6.

For infrastructure-based communications (Roadside-based and Internet-based scenarios as defined in Section 4 in [GeoNetD1.2]), the IPv6 C2CNet link is defined as the broadcast area around the GeoNet RSU in which IPv6 multicast messages are transmitted (such as Router Advertisements, sent to *all-nodes multicast IPv6 address*). This area is defined by means of a geographic area. Thus, an IPv6 C2CNet link is composed of all the nodes within a defined geographical area, containing at least one RSU.

Different areas have different IPv6 prefixes assigned. The on-link/off-link destination determination is based on the comparison of the destination address and the on-link prefix list (or the routing table): if the prefix of the IPv6 destination address is on the on-link prefix list, the destination is assumed to be on-link. The destination is considered to be off-link otherwise.

It should be noted that the C2CNet link connectivity may exist between two nodes that are assumed to be off-link. In those cases (and also when there is not connectivity to the GeoNet RSU), it might be useful to use link-local addressing for that communication, since link-local addresses are assumed to be on-link and reachable through the GeoNet domain.

9.2. SAP C2C-IP: Transmission of packets between IP and C2CNet layers

9.2.1. From IP layer to C2CNet layer

The *GeoIPv6_send* function is defined to transmit the packet from the IP layer (module “3A: IP Forwarding”) to the C2CNet layer (module “2.5B Geo-routing”). As illustrated in Table 2 titled “Parameters of the GeoIPv6_send function”, this function provides three parameters: scope, destination and payload.

- **scope**: according to the destination type as described in Table 1, four scopes are needed: GeoUnicast, GeoAnycast, GeoBroadcast and TopoBroadcast. These correspond to IPv6 unicast, IPv6 anycast, and IPv6 multicast packets, respectively.

- **destination**: the format of this field varies according to the scope of the packet. Two possibilities could be considered. There can be multiple variations of the details of these following two possibilities.

  - First possibility: the IP layer provides a complete IP destination address (128 bits) to the C2CNet layer. The C2CNet layer shall extracts the required information from this IP address such as the C2CNet ID in the case of IPv6 unicast packet or the GeoDestination (position or distance) in the case of an IPv6 multicast or IPv6 anycast packets.

  - Second possibility: the IP layer provides only the useful information without IP prefix. In this case, like with IPv6 over Ethernet, the IP layer should ensure a proper mapping between the C2CNet ID (64 bits) and the IP address through Neighbour Discovery [RFC2461] in the case of IPv6 unicast, or details of the GeoDestination information (position or distance - various field size according to the shape of the GeoDestination) in the case of an IPv6 multicast or IPv6 anycast packets. At the time of writing, there is no requirement for other shape than the circle. For the circle, the C2CNet layer needs to provide latitude (32 bits), longitude (32 bits) and radius (16 bits).

- **payload:** the complete IPv6 packet (IPv6 header and its payload)

As illustrated in Table 1, four types of destinations are considered. There is a one-to-one mapping between the destination represented at the IPv6 layer and the C2CNet layer.

Destination	IPv6 layer	C2CNet layer
A node in a specific vehicle	unicast	GeoUnicast
Nodes in vehicles in area	multicast	GeoBroadcast
Nodes in vehicles *x* hops away	multicast	TopoBroadcast
A node in a certain vehicle in area	anycast	GeoAnycast

Table 1: Relation between destination types at the C2CNet and IP Layers

For IPv6 unicast packets / GeoUnicast, the destination corresponds to the IP next hop, not the final destination. The C2CNet layer uses directly this address for multi-hop routing. The IP next hop must first be determined by the IP layer and should then be provided to the C2CNet layer. The IP next hop is considered as the destination from a C2CNet viewpoint.

For IPv6 multicast / GeoBroadcast, the destination (i.e. GeoDestination) corresponds to the *positions* where the packet shall be GeoBroadcast. The actual parameters transmitted by the *GeoIPv6_send* function depend on the approach adopted to encode the GeoDestination at the IP layer. For Approaches A-C (see Annex A “IPv6 Encoding of GeoDestination” for description of Approaches A-E), all the parameters needed by the C2CNet layer must be provided explicitly. For Approach D, the GeoDestination information is embedded in the destination IPv6 multicast address and must be inferred by the C2CNet layer. For Approach E, only a GeoDestination ID must be transmitted through the SAP, using this GeoDestination ID the C2CNet layer shall retrieve the GeoDestination information through SAP MNG-C2C. The implemented GeoNet solution for IPv6 multicast / GeoBroadcast handling is a simplified version of Approach E.

For IPv6 multicast / TopoBroadcast, only the *hop-distance* must be transmitted.

For IPv6 anycast / GeoAnycast, the destination (i.e. GeoDestination) corresponds to the *positions* where the packet shall be GeoBroadcast, as for the IPv6 multicast / GeoBroadcast case.

Destination\Parameters	Scope	Destination	payload
A nodes in a specific vehicle	GeoUnicast	C2CNet ID of IP next hop	IPv6 packet
Nodes in vehicles in area	GeoBroadcast	Area ID, Radius	IPv6 packet
Nodes in vehicles x hops away	TopoBroadcast	Hop limit	IPv6 packet
A nodes in certain vehicle in area	GeoAnycast	Area ID, Radius	IPv6 packet

Table 2: Parameters of the GeoIPv6_send function

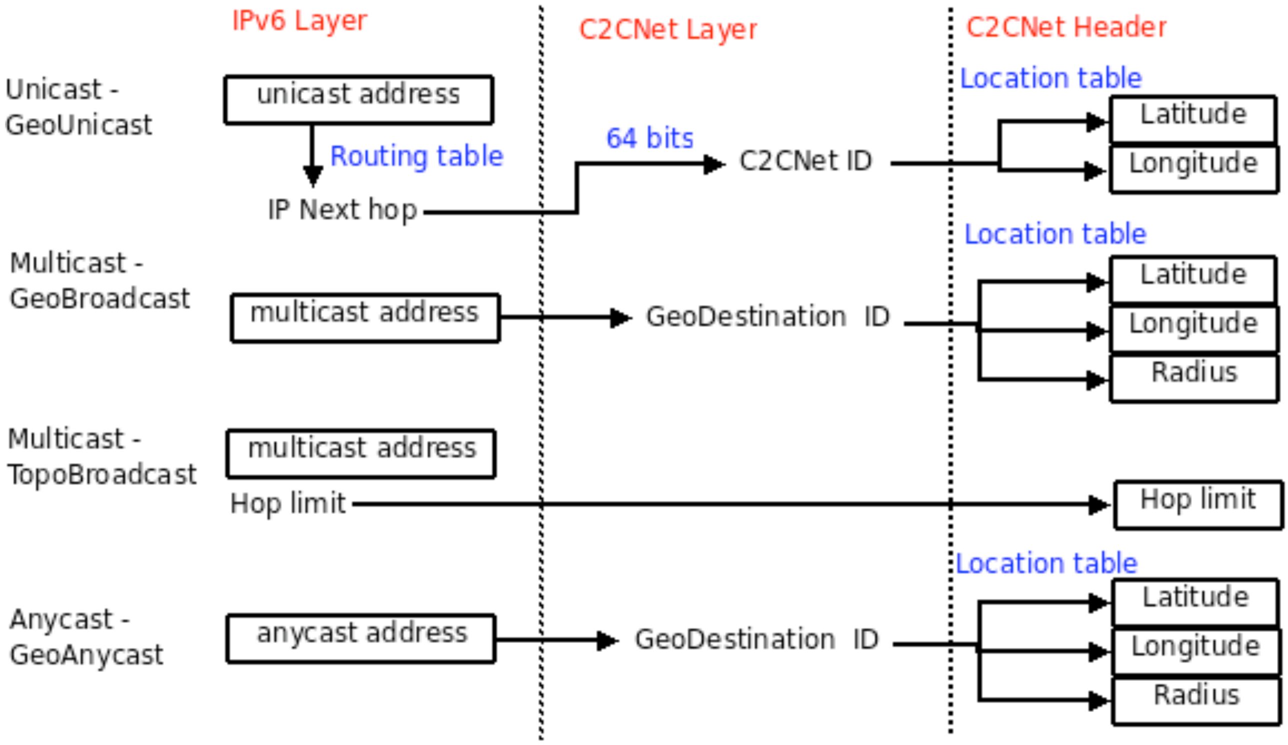


Figure 14: SAP C2C-IP between IPv6 and C2CNet Layer

.

9.2.2. From C2CNet layer to IP layer

Since this operation is performed at the IP next hop node, the plain IP packet is simply extracted from C2CNet layer and is provided to IP layer, without the need of passing any further information.

9.3. Sub-module: Routing

This sub-module is in charge of routing and interface management.

9.3.1. Interface management

The MR (GeoNet OBU) has at least one C2CNet egress interface. In addition, as it is using NEMO Basic Support for maintaining the Internet reachability (see module “3B: Mobility Support” in Section 10), it has an other one for the NEMO tunnel over the C2CNet interface. Additional egress interfaces may be available such as standard WLAN (IEEE802.11a/b/g) or 3G interfaces. Details are out of scope of the GeoNet specification and thus specific modules handling such configuration is not shown on Figure 1) though this possibility is guaranteed through the use of MCoA (see module “3B: Mobility Support” in Section 10). In addition, it has an ingress interface if it has attached nodes (see sub-module “IPv6 over ingress interface”)

Thus the routing table of an GeoNet OBU must be a.ble to maintain several types of interfaces as follows (see Figure 15):

- **C2CNet egress interface** is a tunnel interface (see (a) option on the figure): the packets are passed to the C2CNet layer and therefrom encapsulated with a C2CNet header. They are then passed to module “2A: Egress Interface” where they are encapsulated with an IEEE802.11p MAC header and actually emitted on the air. The C2CNet interface is thus recognised as a tunnel interface in the kernel.

- **Other egress interfaces** are ’normal’ interfaces (see (b) option on the figure): the packets are passed to the corresponding data link layer of the interface and therefrom encapsulated with MAC header of the link type and emitted on the link.

- **NEMO tunnel over C2CNet interface** is a tunnel interface (see (c) option on the figure): the packets are first encapsulated with an IPv6 header (source address: Care-of Address, destination address: HA address). Then the packets are encapsulated by C2CNet header. Finally they are passed to module “2A: Egress Interface” where they are encapsulated with an IEEE802.11p MAC header and actually emitted on the air.

- **NEMO tunnel over other egress interfaces** is a tunnel interface (see (d) option on the figure): the packet are first encapsulated with an IPv6 header (source address: Care-of Address, destination address: HA address). The packets are passed to the corresponding data link layer of the egress interface and therefrom encapsulated with MAC header of the link type and emitted on the link.

- **Ingress interface** is a ’normal’ interface (see lowest arrow on the figure): the packets are passed to module “2B: Ingress Interface” and therefrom encapsulated with MAC header of the link type and emitted on the link.

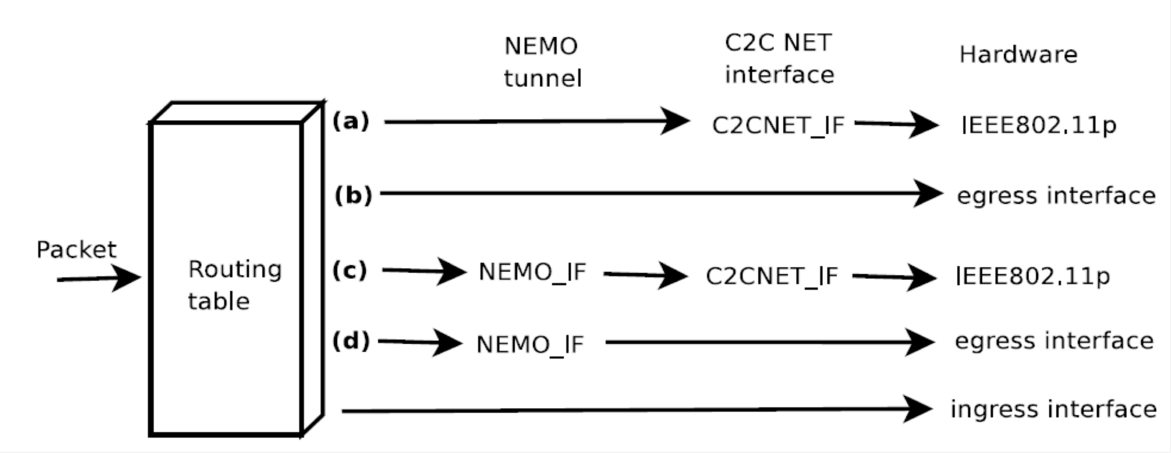


Figure 15: Routing on GeoNet OBU (MR)

The GeoNet RSU on the other hand is not supporting NEMO Basic Support and as such doesn't have NEMO tunnel interfaces. Interface management is thus simplified.

9.3.2. Routing table setup

To distribute packets to multiple paths simultaneously in the GeoNet OBU, policy routing is used. Classic routing mechanisms are not suitable, because of the ’longest match’ principle. We propose to introduce multiple routing tables using Route Policy Database (RPDB) to the system as shown in Figure 16 below. The RPDB allows to maintain several independent routing tables in the kernel. Each packet can then be routed according to one of these tables. The determination of which routing tables should be used in a particular case is up to the implementer. It is usual to route depending on the type of flow that is being routed, and the destination address.

A GeoNet OBU maintains at least five routing tables:

- **C2C_NET:** routing table for C2CNet native packets. This is used for Vehicular-based and Roadside-based communications through the C2CNet interface (scenarios VYi and RYi as indicated in [GeoNetD1.2] Section 4). This table is filled up with input from sub-module “IPv6 over C2CNet”.

- **Normal**: routing table for packet emitted over non-C2CNet egress interfaces. There are as many routing tables of this type as there are non-C2CNet egress interfaces on the GeoNet OBU. This table is filled up with input from sub-modules managing non-C2CNet egress interface (omitted from the GeoNet architecture design).

- **C2C & NEMO**: routing table for NEMO and C2CNet. This is used for Internet-based communications through the C2CNet interface (scenarios IYi as indicated in [GeoNetD1.2] Section 4). This table is filled up with input from module “3A: Mobility Support” and sub-module “IPv6 over C2CNet”.

- **NEMO**: routing table for Internet-based communications through egress interfaces other than C2CNet. There are as many routing tables of this type as there are non-C2CNet egress interfaces on the GeoNet OBU. This table is filled up with input from module "3A: Mobility Support"

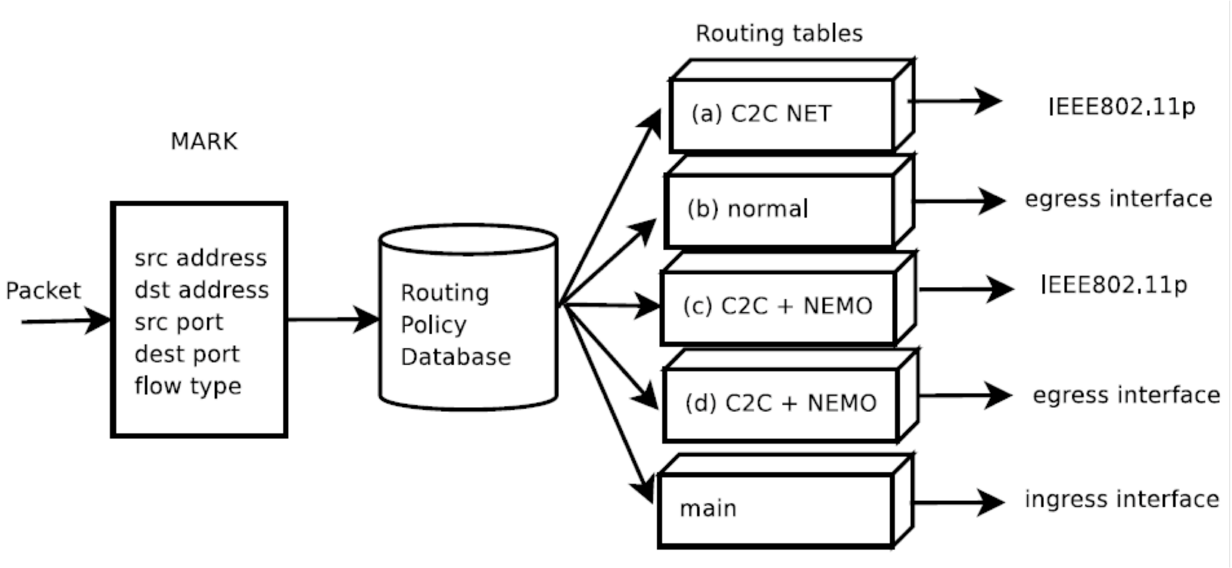


Figure 16: Policy routing on GeoNet OBU (MR)

Given module "3B: Mobility Support", all the packets between any two IPv6 communication endpoints involving a vehicle (MNNs and their CNs) must in principle be encapsulated and tunnelled between the GeoNet OBU and its HA. Strictly following [RFC3963] would require that packets sent between any two nearby vehicles or between a vehicle and the nearby roadside (respectively vehicle-based scenarios VYi and roadside-based scenarios RYi detailed in [GeoNetD1.2] Section 4) be sent to the HA too, which is not necessary, would cause performance degradation and would prevent any communication taking place at all in the situation of lack of Internet connectivity. The behaviour of [RFC3963] must therefore be overlooked in some parts so that a different routing decision is taken when the destination is reachable directly in the GeoNet domain. The corresponding routing entries are provided by the MNPP function of the "IPv6 over C2CNet" sub-module.

Routing at a GeoNet RSU is much simpler as all packets would in principle be routed directly through the C2CNet interface. Of course, much more complex GeoNet RSU configurations may exist, but it is not in the scope of the GeoNet project.

9.4. Sub-module: IPv6 over C2CNET

This sub-module enables GeoNet OBUs and GeoNet RSUs to support geonetworking and is in charge of delivering efficiently a packet to its destination over the C2CNet link.

9.4.1. IPv6 C2CNet interface autoconfiguration

Each C2CNet egress interface of a GeoNet OBU (MR) or GeoNet RSU (AR) must be configured with two different IPv6 addresses: a link-local address and a global address. All GeoNet nodes attached to the same IPv6 C2CNet link should be reachable using both addresses, while the global address must be used when trying to reach other nodes no directly attached to the C2CNet link (i.e. nodes on the global Internet as well as nodes attached to the GeoNet OBUs and GeoNet RSUs). If an GeoNet OBU knows – by using any mechanism (not defined in this section) – that the other communication endpoint is reachable through the IPv6 C2CNet link (on-link destination), it may consider using the link-local address, to avoid traversing the backbone.

The mechanism used to configure the IPv6 C2CNet egress interfaces of GeoNet OBUs in an automatic way is based on the IPv6 Stateless Address Autoconfiguration protocol specified in [RFC4862]. This protocol basically enables a host to generate its own addresses using a combination of locally available information (interface identifier part of the address) and information advertised by routers (prefixes that identify the subnets associated with a link). The concept used in GeoNet is the same: GeoNet RSUs advertise prefix information by sending Router Advertisements (that can be unsolicited or sent in response to a Router Solicitation), and the GeoNet OBUs use that prefix information, together with their C2CNet interface identifiers, to generate a valid global IPv6 address to be assigned to its C2CNet egress interfaces. A link-local address is also generated on the C2CNet egress interface, using the same C2CNet interface identifier and the link-local IPv6 prefix (fe80://64).

A prefix length of 64 bits is used and thus the length of the C2CNet interface identifier is 64 bits. Consequently, the C2CNet ID is used as the C2CNet interface identifier. It should be noted that at a given time an GeoNet OBU may have more than one C2CNet ID, and therefore may generate more than one different IPv6 address. This can be used to change periodically the IPv6 address used by a node and make privacy attacks harder to perform. The valid C2CNet IDs are communicated to module “3A: IP Forwarding” by module ”0B: Security & Privacy” through the MNG-IP SAP.

C2CNet IDs are assumed to be globally unique in the GeoNet domain (this uniqueness is ensured by C2CNet mechanisms, and it is out of the scope of this specification). Therefore, the use of the Duplicate Address Detection (DAD) mechanism defined in [RFC4862] is disabled.

Router Advertisement messages sent by GeoNet RSUs are not only used in the IPv6 address autoconfiguration process, but also to discover GeoNet RSUs and the GeoNet OBUs reachable on the IPv6 C2CNet link (see Section 9.4.5 “Mobile Network Prefix Provisioning”). The basic operation is the following: each GeoNet OBU keeps track of the locally reachable routers so they can be used as the next-hop for a off-link destination. More complex mechanisms, such as the ones specified in [RFC4191] (“Default Router Preferences and More-Specific Routes”) are not considered in the basic specification of GeoNet.

The treatment of Neighbour Discovery Router Advertisement and Router Solicitation is discussed in Section 9.4.2, titled “Treatment of Neighbour Discovery IPv6 multicast packets”.

9.4.2. Treatment of Neighbour Discovery IPv6 multicast packets

IPv6 multicast packets produced by Neighbour Discovery, i.e. the *all-nodes multicast address* and the *all-routers multicast address* must be mapped to a GeoDestination when transmitted to the C2CNet layer through SAP C2C-IP (see Section 9.2).

Since Router Advertisement are typical IPv6 multicast packets, they would be encapsulated at the C2CNet layer and forwarded as GeoBroadcast. The IP layer must thus pass the necessary information to the C2CNet layer so that it could determines the GeoDestination of Router Advertisement.

The *all-nodes multicast address* is of particular relevance since it is the destination address used when sending Router Advertisements. In the context of GeoNet, Router Advertisements shall be GeoBroadcast within a well delimited geographical area. The limits of this area are administratively set at the GeoNet RSUs serving that area, through the IPv6_MULTICAST_GEOAREA configuration variable (the format of this variable is the same defined in the C2CNet header to specify the destination area).

Analogously, the mapping of the *all-routers multicast address* it also important, since it is the destination address used when sending Router Solicitations. In the context of GeoNet, Router Solicitations shall be GeoBroadcast within a well delimited geographical area. The same area limits are used for Router Advertisements and Router Solicitations. GeoNet OBUs learn the boundaries of this geographical area from the C2CNet frames containing unsolicited IPv6 Router Advertisements and maintain and update this information in a new conceptual data field stored by GeoNet OBUs (IPV6_MULTICAST_GEOAREA). If no unsolicited IPv6 RA has been received, unsolicited RS can be sent using TopoBroadcast (limited in number of hops, with a default value of 5 hops).

Until an improved mapping of IPv6 multicast addresses is designed, it is expected that every reserved IPv6 link-scoped multicast address is mapped to the same reserved GeoDestinaton ID, representing the whole geographical area, and therefore meaning that every IPv6 link-scoped multicast packet is actually GeoBroadcast to the entire geographic area. For this specific case of Neighbour Discovery, Approach E for encoding the GeoDestination at the IP layer is adopted (see Annex A "IPv6 Encoding of GeoDestination"). Special GeoDestination ID must be defined (and reserved) for each of the reserved IPv6 multicast addresses and mapped to a GeoDestination ID. The *all-nodes multicast address* (FF02::1) and the *all-routers multicast address* (FF02::2) are mapped to GeoDestination ID 1 and 2, respectively. These special reserved GeoDestination IDs are mapped in turn to GeoDestinations recorded in a table located in module 0A "Geo-destination" and reachable by the C2CNet layer through SAP MNG-IP".

9.4.3. GeoNet OBU operation and state machine

A GeoNet OBU is a router implementing the MR operation as defined in [RFC3963]. On the egress interfaces, the GeoNet OBU behaves as a host from the viewpoint of IPv6 neighbour discovery and IPv6 stateless autoconfiguration. Next, we describe the detailed behaviour of a GeoNet OBU in terms of IPv6 operations.

Since an GeoNet OBU behaves as a host on its C2CNet egress interface, it needs to maintain some information. The required information is described in Section 5.1 of [RFC4861], and comprises the following[4]: a Neighbour Cache, a Destination Cache, a Prefix List and a Default Router List. Since the GeoNet OBU is not a pure host, but a router it makes sense to implement these conceptual data structures as a single longest-match routing table. Actually, the most sensible approach would be to reuse the IPv6 stack of a router, enabling the egress interfaces as a host interface (this is at least possible with Linux kernel).

The **state machine and algorithms** that an GeoNet OBU has to implement on its C2CNet egress interface are as specified in [RFC4861] (for a host) and [RFC4862], except for a few changes. We summarise these differences below:

- The interface identifier used to generate the link-local address – and the global addresses, when the GeoNet OBU has connectivity to a GeoNet RSU – is obtained from the C2CNet ID of the GeoNet OBU. Therefore, there always exists a direct mapping between the IPv6 address and the link-layer address seen by the IPv6 stack. This allows to perform address resolution without sending any actual message on the IPv6 C2CNet link Therefore, it is avoided to send Neighbour Solicitation and Neighbour Advertisement messages over the C2CNet egress interface (that is, within the GeoNet domain, where sending multicast messages is costly). Therefore, there cannot be any entry in the Neighbour Cache of the GeoNet OBU (referring to a neighbour reachable through the egress interface) in the INCOMPLETE state.

- Neighbour Solicitation messages for reachability check purposes should be sent only to unicast addresses. Since address resolution is performed without signalling, it should not be necessary to send any multicast Neighbour Solicitation message.

- Duplicate Address Detection (DAD) is disabled, since the link-layer addresses used by the IPv6 stack are the C2C IDs, which are assumed to be unique.

9.4.4. GeoNet RSU operation and state machine

A GeoNet RSU is a router behaving as an IPv6 Access Router (AR). The GeoNet RSU is configured to send unsolicited Router Advertisements (RAs). The periodicity of these messages has to be experimentally analysed, based on real trials. For the first phase, RAs should be sent every 10 seconds.

The GeoNet RSU should maintain the same information that an IPv6 router as described in [RFC 4861]. The C2CNet interface of an GeoNet RSU must be configured as follows (see section 6.2 of [RFC4861]):

[4] For further details about the content of these conceptual data structures, please refer to [RFC4861].

- Every Router Advertisement sent must include at least one IPv6 prefix with the L-bit (on-link flag) set and the A-bit (autonomous address-configuration flag) set in the Prefix Information Option (defined in Section 4.6.2 of [RFC4861]).

- The Valid Lifetime and Preferred Lifetime values contained in the Prefix Information Option of sent Router Advertisement messages should take into account the expected dynamic nature of C2CNet Networks and the cost associated to send multicast messages over these links. Recommended values for initial phase: 100s. Experimental trials should be conducted to tune the recommended values.

The **state machine and algorithms** that a GeoNet RSU has to implement for its C2CNet interface management are the same that as specified in [RFC4861] (for a router), except for a few changes. The differences are:

- The interface identifier used to generate the link-local address is obtained from the C2CNet ID of the GeoNet RSU. Therefore, there always exists a direct mapping between the IPv6 address and the link-layer address seen by the IPv6 stack. This allows to perform address resolution without sending any actual message on the C2CNet link. Therefore, it is avoided to send Neighbour Solicitation and Neighbour Advertisement messages over the C2CNet egress interface (that is, within the GeoNet domain, where sending multicast messages is costly). Therefore, there cannot be any entry in the Neighbour Cache of the GeoNet RSU (referring to a neighbour reachable through the egress interface) in the INCOMPLETE state.

- The IPv6 global address of the C2CNet interface is confiigured based on the C2CNet

- Neighbour Solicitation messages for reachability check purposes should be sent only to unicast addresses. Since address resolution is performed without signalling, there should not be necessary to send any multicast Neighbour Solicitation message.

- Duplicate Address Detection (DAD) is disabled, since the link-layer addresses used by the IPv6 stack are the C2C IDs, which are assumed to be unique.

9.4.5. Mobile Network Prefix Provisioning (MNPP)

IPv6 nodes attached to GeoNet OBU and GeoNet RSUs must be able to communicate with one another over the C2CNet link (Vehicle-based scenarios and Roadside-based scenarios are detailed in [GeoNetD1.2] Section 4).

In GeoNet, NEMO Basic Support [RFC3963] is used to maintain Internet reachability at a permanent IPv6 address for all nodes attached to the GeoNet OBU (see specification of module “3B: Mobility Support”). The IPv6 prefix announced in the in-vehicle network is called the Mobile Network Prefix (MNP). This prefix is topologically meaningful in a remote attachment point (the home network). In the meantime, the C2CNet egress interface is configured with a link-local and a global address known as the CoA using another prefix (e.g. a prefix announced by the GeoNet RSU).

Without any specific mechanisms, GeoNet OBUs would only know what is the link-local address or CoA of neighbour vehicles, but not their associated MNPs. In order to allow direct Vehicle-based communication (V2V) or Roadside-based communications (V2I) over the C2CNet link MNPs belonging to distinct GeoNet OBUs must be exchanged over the C2CNet link. As a result, the C2CNet IPv6 over ingress sub-module must communicate to the routing sub-module new routes that supercede the normal routes as otherwise packets would be sent through the normal route added by NEMO, i.e. in a NEMO tunnel to the Home Agent which would cause routing inefficiencies.

In this section, we specify MNPP (Mobile Network Prefix Provisioning), a mechanism part of sub-module “IPv6 over C2CNet” in module “3A: IP Forwarding”. MNPP is a based on an extension of Neighbour Discovery Protocol (NDP) [RFC4861] to distribute the MNP following either a proactive method or a reactive method.

MNPP extends the router advertisement (RA) and router solicitation (RS) messages defined in NDP to announce the MNP assigned to a vehicle (the GeoNet OBU and its attached nodes) to other vehicles or the roadside infrastructure on the same IPv6 C2CNet link.

In order to distribute the MNP, MNPP Advertisement messages are sent through the C2CNet interface of the GeoNet OBU. MNPP Solicitation messages are used to request GeoNet OBUs to quickly send MNPP Advertisement messages. This could be done via either of the two provided methods:

- **Proactive method**: the GeoNet OBU periodically sends the unsolicited MNPP Advertisement messages including the MNP being used in its in-vehicle network. Upon reception of the MNPP Advertisement message, other GeoNet OBUs and GeoNet RSUs obtain the MNP of the GeoNEt OBU. It is similar to the case of sending unsolicited RA messages defined in NDP.

- **Reactive method**: the GeoNet OBU upon reception of the MNPP Solicitation messages immediately sends the solicited MNPP Advertisement messages including the MNP being used in its in-vehicle network. The MNPP Solicitation messages SHOULD be used to prompt GeoNet OBUs to generate the MNPP Advertisement messages quickly. It is similar to the case of requesting solicited RA messages defined in NDP.

MNPP defines two new ICMPv6 messages:

- **Mobile Network Prefix Provisioning (MNPP) Solicitation Message:** A solicitation message sent by the GeoNet OBUs and GeoNet RSUs prompting the GeoNet OBUs to quickly issue an advertisement message containing the MNP information quickly.

- **Mobile Network Prefix Provisioning (MNPP) Advertisement Message**: An advertisement message sent by the GeoNet OBU to announce its MNP information. This message is periodically sent, or in response to a solicitation message.

Receivers must silently ignore any option and continue processing if they do not recognise it. Upon the reception of a valid MNP, the information is provided to sub-module “Routing” where new routes are added in the C2C_NET routing table.

9.4.5.1. Mobile Network Prefix Provisioning Solicitation Message

This is a message sent by the GeoNet OBUs and GeoNet RSUs on the IPv6 C2CNet link to prompt GeoNet OBUs to quickly generate a MNPP Advertisement message. MNPP Advertisement messages have the following ICMPv6 structure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type         |     Code      |             Checksum          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Reserved                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Options                              |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- IP Fields
  - Source Address
    - An IPv6 address assigned to the sending C2CNet interface.
  - Destination Address
    - The *all-routers multicast address* or a specific IPv6 address.
  - Hop Limit
    - 255

- ICMP Fields
  - Type
    - 201
  - Code
    - 0
  - Checksum
    - The ICMP Checksum
  - Reserved
    - This field is unused. It MUST be initialised to zero by the sender and MUST be ignored by the receiver.

- Valid Options
  - Source link-layer address
    - The link-layer address of the sender is included.

9.4.5.2. The Mobile Network Prefix Provisioning Advertisement

This message is sent on the IPv6 C2CNet link by GeoNet OBUs either periodically (proactive method), or in response to MNPP Solicitation messages (reactive method). The purpose is to announce the MNP served by the GeoNet OBU. MNPP Advertisement messages have the following ICMPv6 structure:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type         |     Code      |             Checksum          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Reserved                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Options                              |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- IP Fields
  - Source Address
    - An IPv6 address assigned to the sending C2CNet interface.
  - Destination Address
    - The *all-routers multicast address* or a specific IPv6 address.
  - Hop Limit
    - 255

- ICMP Fields
  - Type
    - 201
  - Code
    - 0
  - Checksum
    - The ICMP Checksum
  - Reserved
    - This field is unused. It MUST be initialised to zero by the sender and MUST be ignored by the receiver.

- Valid Options
  - Source link-layer address
    - The link-layer address of the sender is included.
  - In-vehicle MTU
    - The MTU used in the in-vehicle network is included.
  - Mobile Network Prefix Option
    - The MNP is included to indicate which prefix is being used in the in-vehicle network.

9.4.5.3. Security Issues in MNPP

MNPP is used to provide the MNP information being used in the in-vehicle network served by the GeoNet OBU. In addition, the MNPP Advertisement message includes the C2CNet ID. Accordingly, the two messages used for MNPP have to be protected:

- Message Authentication: Both messages used for MNPP have to be authenticated. Without being authenticated, the receiver cannot determine if the sender is a valid node or not. The message authentication can be provided by the use of the certificate. Moreover, the receiver should check that the sender is the real owner of the link-layer address presented in the message. From the perspective of the attacker, the fake link-layer address would be used to redirect traffic to him or other nodes. To solve this issue, Cryptographically Generated Address (CGA) could be applied to validate the address ownership.

- Message Authorisation: The sender of the MNPP Advertisement message has to prove it is the legitimate owner of the MNP contained in the message. Otherwise, the message could be used to divert traffic to the wrong GeoNet OBU. From the perspective of the attacker, the fake or false MNP would be used to redirect traffic to him or other nodes. This redirect attack leads to further attacks such as traffic interception, DoS attack, etc. To solve this issue, Resource Public Key Infrastructure (RPKI) certificate could be applied to avoid such attacks. In addition, a new ExtKeyUsageSyntax field of the certificate can be used to specify that this node having the certificate allows to send MNPP messages.

- Message Encryption: In order to provide message confidentiality, both messages should be encrypted. In particular, the MNPP Advertisement message has to be encrypted because this message includes enough information for prospective attackers to track the sender of the MNPP Advertisement message, therefore exposing the sender to location privacy threats. Moreover, without the message encryption, attackers can capture all information being exchanged with other nodes and use if for different types of attacks.

9.5. Sub-module: IPv6 over ingress interface

This sub-module enables IPv6 over the lower layer technology provided by module “2B: Ingress Interface” through SAP IP-LL and allows to attach other IPv6 nodes behind the GeoNet OBU or the GeoNet RSU. The IPv6 operation on the ingress interfaces is not affected by the combination of IPv6 and C2CNet layers. There are as many instances of this sub-module as there are instances of physical ingress interfaces implemented in a given node.

From an IPv6 operation viewpoint, all IPv6 nodes, either GeoNet OBU, GeoNet RSU, GeoNet-aware or legacy IPv6 node perform this sub-module similarly with no modification of the IP layer. As such they run Neighbour Discovery and Stateless Address Autoconfiguration as specified in [RFC4861] and [RFC4862].

GeoNet OBUs and RSUs play on this interface the role of a router as defined in [RFC 4861]. They may perform Neighbour Discovery [RFC4861] and Stateless Address Auto-configuration [RFC4862] in order to provide IP parameters such as the IPv6 prefix necessary to the attached nodes to acquire an address and to determine the default router. Alternatively, these settings may also be provided statically.

On the vehicle side, the in-vehicle network must be configured with a global IPv6 prefix. Since the GeoNet OBU serving the in-vehicle network is using NEMO Basic Support (see module “3B: Mobility Support”) to maintain both the global reachability from and to the Internet using a global IPv6 prefix referred to as the Mobile Network Prefix (MNP) and permanent connectivity to the Internet, all IPv6 nodes in the in-vehicle network must configure and address based on the MNP advertised by the GeoNet OBU.

10. Module 3B: Mobility Support

This module is responsible for IPv6 mobility support. It communicate with Mobile 3A "IP Forwarding". Implemented in GeoNet OBU nodes only, it maintains Internet connectivity and provides session continuity to the GeoNet OBU nodes (MR) and in-vehicle IPv6 nodes (MNNs) attached to the ingress interface of the GeoNet OBU. As defined in [GeoNetD1.2] (section 6) and shown on Figure 1, module "3B: Mobility Support" includes two sub-modules:

1. **NEMO** (NEtwork MObility): Required for ubiquitous Internet connectivity, this sub-module is in charge of maintaining globally reachable IPv6 addresses for all nodes in the vehicle and to maintain Internet connectivity at the GeoNet OBU through the C2CNet egress interface when GeoNet RSUs or other nearby GeoNet OBUs are able to provide Internet access over the GeoNet domain. It takes care of the set-up and management of a MR-HA tunnel between the GeoNet OBU (MR) and the home agent (HA) . It also take care of the encapsulation / decapsulation of packets sent to or received from the HA. It conforms to [RFC3963] (NEMO Basic Support). Module "3A: IPv6 forwarding" is informed about the availability of MR-HA tunnels.

2. **MCoA**: (Multiple Care-of Address Registration): Required for media-diversity, this sub-module is in charge of maintaining several MRHA tunnels between the GeoNet OBU and the HA. It is necessary when the GeoNet OBU is equipped with multiple wireless technologies (i.e. multiple egress interfaces). The ability to use them simultaneously while managing network mobility is provided by [RFC5648].

10.1. NEMO: NEtwork MObility Support

To support communication in mobile environment, NEMO Basic Support (or NEMO for short – see [RFC4885] for all NEMO-related terminology) has been standardised at the IETF in the former NEMO Working Group (which work is now taken over by the MeXT Working Group). NEMO Basic Support is one essential key feature for vehicular communications as on the one hand it allows all IPv6 nodes deployed in a in-vehicle network to be reachable at a permanent address, and on the other hand to maintain Internet connectivity and open sessions over subsequent point of attachment to the network. The references [ETSI-TS-102-636-1], [ISO-21210] and [Ernst2009] provide further explanation on the use of NEMO protocol for ITS networking.

NEMO Basic Support [RFC3963] is a protocol for supporting network mobility, i.e. in-vehicle IPv6 networks by opposition to single IPv6 hosts. Since a vehicle may have IPv6 nodes attached to the GeoNet OBU, network mobility support is essential. Note that NEMO Basic Support also has the ability to support single host mobility (this is considered in GeoNet since the support of module "2B: Ingress Interface" is optional on the GeoNet OBU – see [GeoNetD1.2] Section 6).

To support network mobility, a router referred to as the Mobile Router (MR – i.e. GeoNet OBU in the context of GeoNet) manages mobility on behalf of all in-vehicle network nodes (MNN). MNNs can benefit from this feature without any specific support, which means that any node equipped with an IPv6 stack can be attached in the in-vehicle network and engage into Internet-based communications.

Internet reachability is ensured by keeping in the in-vehicle network a permanent IPv6 prefix, referred to as the **Mobile Network Prefix** (**MNP**). The MNP belongs to a **home network**, located somewhere in the Internet (the car manufacturer headquarters, road authority, or ITS services operators). It could be assigned by different means which are out of concerns of the GeoNet work.

At each subsequent point of attachment to the Internet, the MR must conform to IPv6 addressing requirements and configure a IPv6 link-local address and then an IPv6 global address on each of his interface based on the IPv6 prefix announced on the link the interface is attached to (as indeed indicated in specification of module “3A: IP Forwarding” of the present document). In principle, the MR is configuring a transient IPv6 address on its egress interface at each subsequent point of attachment.

In order to be reachable at a permanent address, an address configured on the MNP must be kept configured on the MR and on all its attached nodes. This is where the operation of NEMO Basic Support takes place. The MR is sending a message (“Binding Update” registration) to a server named the **Home Agent** (**HA**) located on the home network. This message contains the transient address configured on the egress interface, therefrom instructing the HA to redirect all packets addressed to an address part of the MNP to the transient address, named **CoA** (**Care-of Address**) in the context of NEMO.

As a result of the Binding Update registration, the HA and the MR establish a NEMO IP-in-IP tunnel (also referred to as MRHA tunnel) in which all packets between a MNN and their correspondent in the Internet (CNs) are encapsulated. This tunnel has to be updated each time a new CoA (with global reachability) is configured on the egress interface.

In the context of GeoNet, this CoA is configured on the IPv6 C2CNet interface using the procedures specified in Section 9, titled “Module 3A: IP Forwarding” (more precisely sub-module “IPv6 over C2CNet”). This configuration is performed with the prefix announced by the GeoNet RSU providing access to the Internet over the IPv6 C2CNet link. There is a strong interaction between modules 3A and 3B for maintaining the routing table.

Figure 17 shows an IPv6 communication example in GeoNet. In this example, an IPv6 node (an MNN, also known as Application Unit or AU) attached to a GeoNet OBU is communicating with a node in the Internet (a Correspondent Node, CN), that is an off-link destination. Packets sent by the MNN are handled by the GeoNet OBU (which implements module “3B: Mobility Support” and more specifically NEMO Basic Support protocol for maintaining Internet connectivity of the vehicle). An MNN about to send a packet chooses the IPv6 source address among its available addresses in case it has multiple available IPv6 addresses (e.g. configured from the MNP). The GeoNet OBU then decides how to forward packets received from the MNN based on the source and destination addresses of the packet IP header.

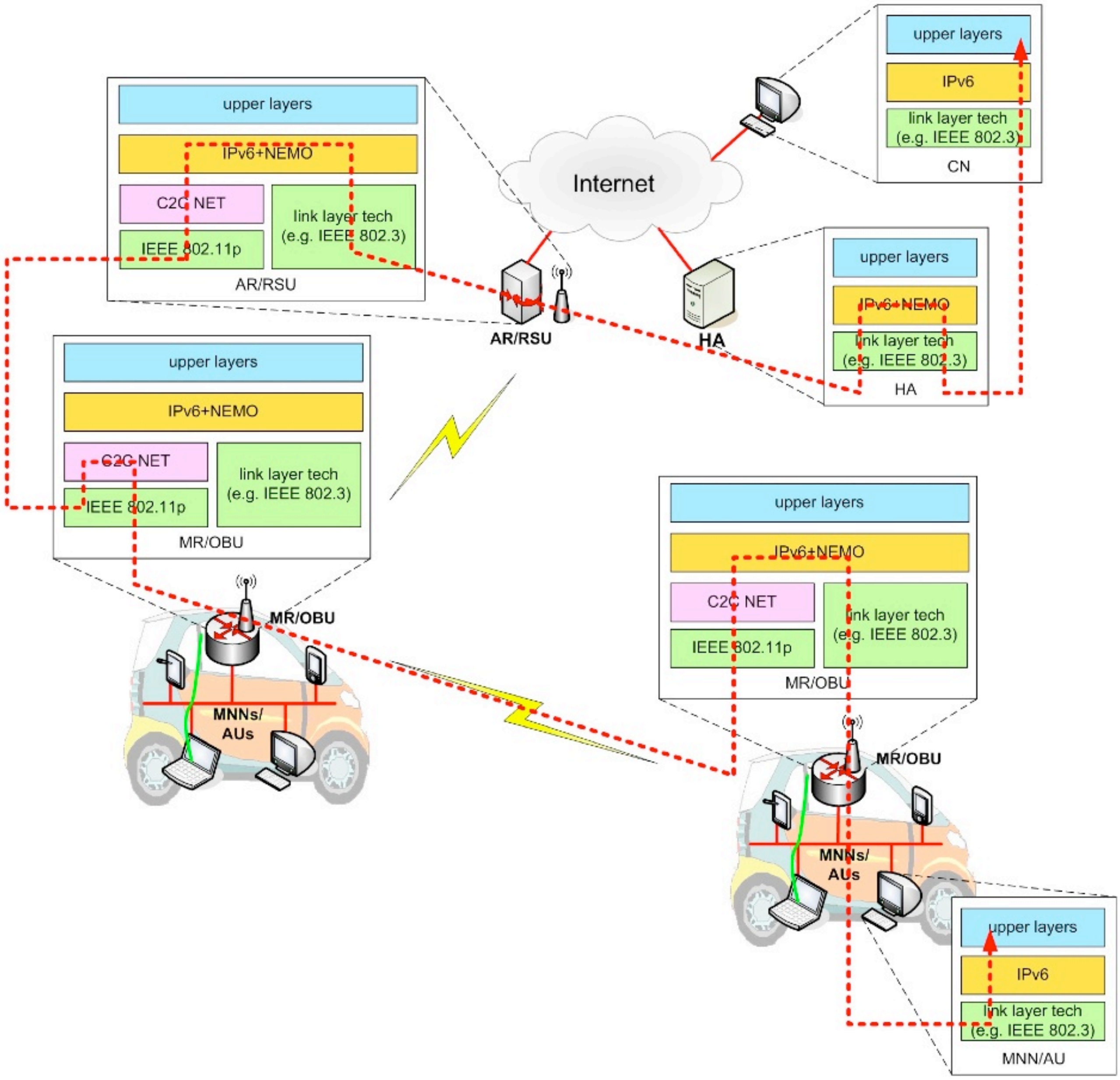


Figure 17: IPv6 Geonetworking overview

.

For a destination CN located in the Internet as illustrated on Figure 17, the GeoNet OBU transmits packets to its Home Agent (HA) using a bi-directional IPv6-in-IPv6 tunnel (see module “3B: IP Forwarding”). In order to reach the HA, the GeoNet OBU has to forward its packets to its IPv6 next-hop towards the HA. This IPv6 next-hop is the GeoNet RSU (AR on the figure) to which the GeoNet OBU is attached, that is – as can be seen in the figure – two wireless hops away. The C2CNet layer takes care of delivering these packets to the GeoNet RSU, providing to IPv6 a single-hop link view (that is, the GeoNet RSU and the GeoNet OBU IPv6 neighbours). From the viewpoint of the IPv6 stack of the GeoNet OBU, the C2CNet layer is a sub-IP layer and therefore, the only thing that the GeoNet OBU needs to know to send an IPv6 packet is the sub-IP layer address of the next-hop, which is the C2CNet ID of the GeoNet RSU. Then the IPv6 stack gives the C2CNet layer the IPv6 packet and the C2CNet destination address, and the C2CNet layer forwards the packet to the GeoNet RSU, using the C2CNet position based routing.

The packet reaches the GeoNet RSU where it is transmitted from the C2CNet layer to the IPv6 layer. From there on, the packet is routed using conventional mechanism up to the HA. The HA delivers the received packet (after removing the outer IPv6 header of the tunnel) to its final destination (the CN).

Some issues still remain in NEMO Basic Support. In particular, sub-optimal routing via the HA, an issue known as “Route Optimisation”. Sub-optimal routing is caused by the packets being forced to pass by the HA. This leads to performance degradation due to increased delay and is undesirable for some applications. Packet Encapsulation of additional 40 bytes header increases packets overhead and may result into packet fragmentation. This in turn results into an increased processing delay for every packets being encapsulated and decapsulated in both the GeoNet OBU and the HA. Bottlenecks in the HA are a severe issue because significant traffic to and from MNNs is aggregated in the HA when it supports several GeoNet OBUs acting as gateways for several MNNs. This may cause congestion at the HA that would lead to additional packet delays, or even packet losses. This issue is subject to bringing further enhancements to this specification although it is not peculiar to IPv6 geonetworking and thus out of scope of the GeoNet project.

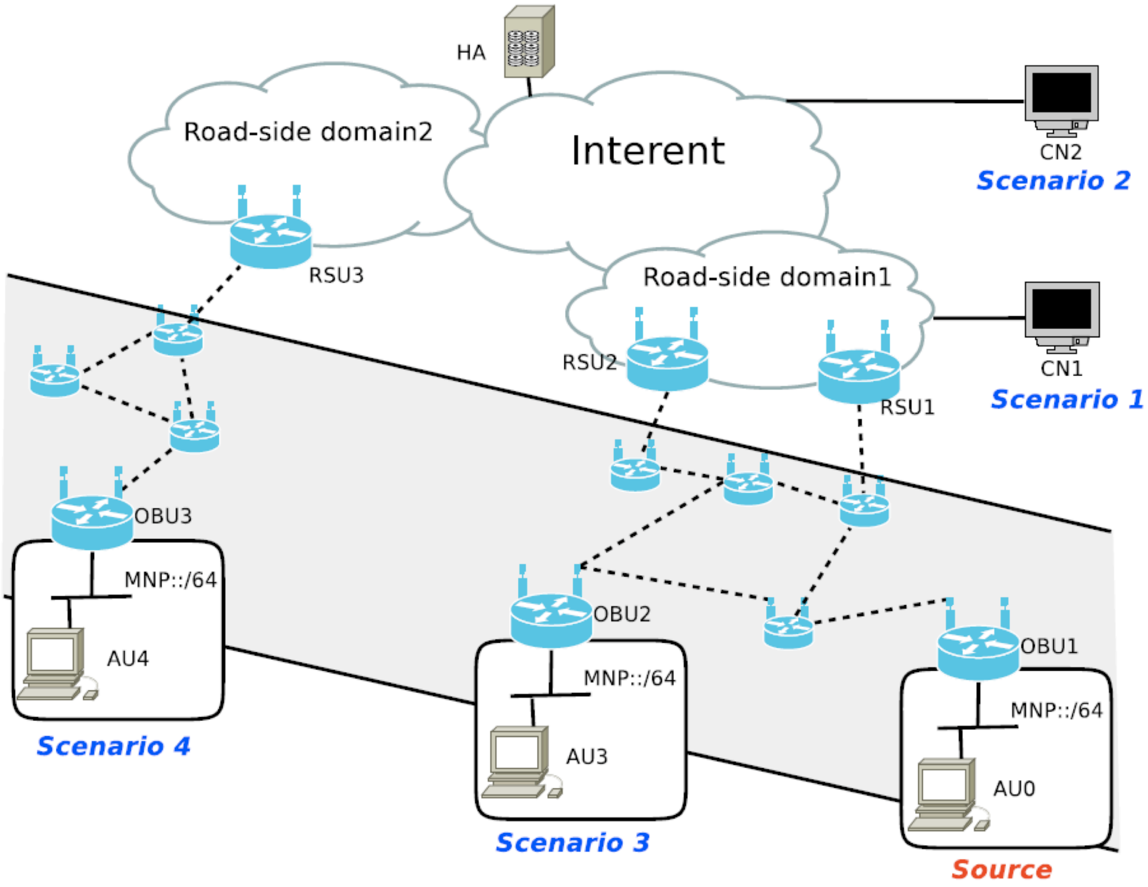


Figure 18: Route Optimisation scenarios

In the context of GeoNet, route optimisation scenarios are divided into four scenarios as illustrated in Figure 18. On this figure, the OBU and MNN under consideration is indicated as the “source”. The classification, first, depends on whether the correspondent node (CN) is located in the Internet (Internet-based communication scenarios as depicted in

[GeoNetD1.2] Section 4) or is attached to a GeoNet OBU (Vehicle-based communication scenarios) or a GeoNet RSU (Roadside-based communication scenarios). Issues relevant to IPv6 geonetworking are the latter two, and is depicted as Scenario 3 when the source OBU and the OBU the CN is attached to are on the same IPv6 C2CNet link. This issue is indeed solved by the MNPP function specified within module 3A “IP Forwarding” (“IPv6 over C2CNet” sub-module).

10.2. MCoA: Multiple Care-of Address Registration

NEMO Basic Support configures a tunnel between the HA address and the GeoNet OBU using the CoA address. Multiple Care-of Addresses Registration (MCoA) [RFC5648] is thus proposed as an extension of both Mobile IPv6 and NEMO Basic Support to establish multiple tunnels between GeoNet OBU and HA when the GeoNet OBU has multiple egress interfaces. Each tunnel is distinguished by its Binding Identification number (BID). The multiple CoAs and the BID are registered with the HA. In other words, NEMO Basic Support only realises interface switching while MCoA supports simultaneous use of multiple interfaces. A GeoNet OBU can register multiple CoAs at once by sending a single BU to the HA (this is defined as bulk registration).

Effective support of non-C2CNet egress interfaces and the selection criteria of the appropriate egress interface to be used for sending out a given packet is out of scope of the GeoNet project and is not detailed further.

11. Module 3C: Multicast

This module is responsible for IPv6 multicasting. Its main purpose is to gather group membership subscription in order to establish a packet delivery tree and to perform multicast routing. In this section, we describe the IPv6 multicast group membership, the IPv6 multicast routing fabric and the IPv6 multicast addressing to put in place in order to deploy IPv6 multicast in a GeoNet domain.

In what follows below, we are using the term MR to mean “IPv6 Multicast Router”. In the context of GeoNet, all GeoNet OBU and RSUs implement module “3C: Multicast” and are de-facto IPv6 Multicast Routers if they have more than one interface (as explained in [GeoNet1.2] Section 5, GeoNet OBUs and RSUs must at least implement a C2CNet egress interface).

11.1. Design of IPv6 multicast in GeoNet

The one-to-many packet delivery requires an efficient use of the network infrastructure and bandwidth. Originally, to deliver packets from sources to a set of destinations, multiple copies of the same message are sent or the entire network is flooded. IP multicast is used to efficiently propagate data packets to a set of recipients. The principle of IP multicast is that only one copy of a given packet is transmitted on any given link, and only to the condition that there is are known destinations reachable through this link. Senders are called **multicast sources** and are identified by the source address of the packet they send to the multicast group. Multicast receivers are known as **listeners**. Senders have no prior knowledge about the multicast receivers addresses nor about the number of receivers.

Two main multicast functions must be performed:

- a **group membership management** function covers the subscription of the multicast receivers to a special group identified by a multicast address. Hosts instruct their routers telling them which multicast groups they are interested in. Otherwise, multicast packets for all multicast groups announced in a specific network domain (network administered by the same network authority) would be received by all nodes located in the network domain. The protocol used to manage the group membership on an IPv6 link is the **Multicast Listener Discovery** protocol known as **MLD**. Section 11.3, titled “IPv6 multicast group management with Multicast Listener Discovery (MLD)” defines the use of MLDv2 in the context of GeoNet.

- a **multicast packet delivery** function is ensured by a **multicast distribution tree** built by a dedicated **multicast routing protocol.** DVMRP [RFC 1075] and MOSPF [RFC 2328] are two multicast protocols that are no longer used because they are neither scalable nor adaptable to the Internet. Protocol Independent Multicast (PIM) [RFC 4601] is the most used multicast routing protocol in both IPv4 and IPv6 networks. MLD Forwarding Proxy [RFC 4605] is a simplified approach for propagating group information in the network used to avoid the deployment of PIM multicast routers in a given network domain. Two operational solutions for deploying multicast in the GeoNet domain are presented in the following sub-sections. The first one relies on **static multicast routing** and the second one is using **MLD Forwarding Proxy**. Both methods have been tested; static multicast routing is the method used in the experiments and more details will be provided in other GeoNet deliverables, particularly [GeoNet7.1]. These methods are described in Sections 11.4 and 11.5, titled “Static multicast packet delivery” and “MLD Forwarding Proxy”.

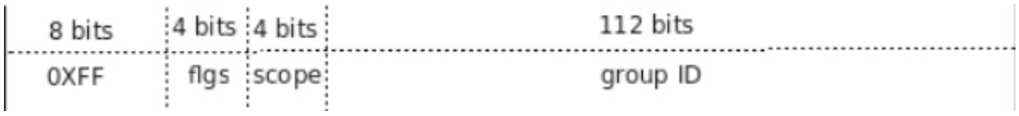
To address the set of the multicast receivers interested in receiving multicast traffic, a **multicast destination address** that identifies the target group is required. The format is defined in Section 11.2, titled “Multicast Addressing in GeoNet”.

In the context of GeoNet where IP multicast is used to send packets to a geographical area in the GeoNet domain, we need to consider how the geographic information defining the target geographical destination area (GeoDestination) is encoded at the IP layer. Four approaches are discussed in Annex A. At the time of writing, we think it is not reasonable to take a decision on which solution is best because the right choice depends on parameters not specific to IPv6 geonetworking.

In this section, we thus specify a solution for multicast group management and routing that can be deployed immediately, with no modification to the legacy IPv6 multicast routing fabric and to IPv6-related standards. It has to be noted that whatever Approach is adopted to encode the GeoDestination at the IP layer, the multicast group management and routing mechanisms described in Section 11.3 “IPv6 Multicast Group Management with Multicast Listener Discovery” and 11.2 “Multicast addressing in GeoNet” below are still necessary and work transparently.

Deployment of IPv6 multicast in the GeoNet domain is illustrated with a basic Vehicle-based communication scenario. Take notice here that from an IP multicast view point, there is no difference with Roadside-based nor Internet-based communication scenarios (refer to [GeoNetD1.2] Section 4 for a description of communication scenarios).

11.2. Multicast addressing in GeoNet



8 bits	4 bits	4 bits	112 bits
0XFF	flgs	scope	group ID

Figure 19: Format of an IPv6 multicast address

As defined in the IPv6 addressing architecture [RFC4291], a specific IPv6 format is attributed to the multicast addresses. The multicast address consists of a FF00::/8 prefix, a set of flags, a scope and a group identifier as pictured in the figure below. The flags are used to specify whether the multicast address is permanent (assigned by IANA) or temporary. The scope field limits the range of the multicast packets.

- *flgs* is a set of four flags which for the solution have the value 0001, i.e., non-permanently-assigned multicast addresses.
- *scope* is a 4-bit multicast scope value used to limit the propagation of the multicast packet.
- *group ID* identifies the multicast group.

IP-based applications are supported in the GeoNet architecture by encapsulating IPv6 packets within geonetworking packets at the C2CNet layer. From the IPv6 layer perspective other GeoNet OBUs and RSUs appear as directly connected over a virtual C2CNet link i.e., they appear to be reachable with one single hop. This is shown in figure 20.

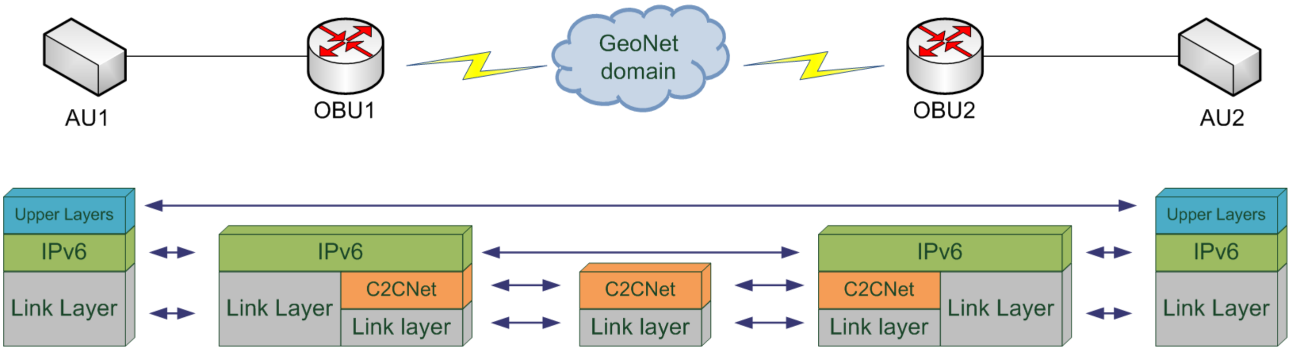


Figure 20: IP-based application communication

In this scenario, an IP-based application installed in an GeoNet-aware IPv6 node (AU) that wants to send information to a geographical area (GeoDestination) need to specify somehow this GeoDestination to the GeoNet OBU using the IP protocol stack. This is necessary in order to transfer the information from the GeoNet-aware nodes (AU) to the GeoNet OBU which is the node able to perform geonetworking functions.

Due to the fact that a geographic address can address several GeoNet OBUs and RSUs, it is needed to enable IP multicast to also support a geographical scope. This would enable an AU to send IP multicast traffic to a multicast group that is scoped by a geographic destination area. The usage of IP multicast and geographic routing limits the forwarding to the multicast group receivers in a geographic area.

The challenges here are to enhance IP multicast to support the geographical scope and design the mechanism that allows IPv6 communication end-points to address nodes within a certain geographic area. For demonstrating the concept of IPv6 geonetworking, we have adopted for GeoNet a simplified version of Approach E which doesn't require specific encoding of the GeoDestination at the IP layer. At this point in time, any of the approaches reported in the Annex A may be preferred for full deployment of ITS services based on IPv6 geonetworking.

In the simplest case, IPv6 multicast packets must be GeoBroadcast at the C2CNet layer within a specific radius around the originating node or a target node. In this case the IPv6 multicast address is statically matched to a corresponding configuration directive, which map a GeoDestination ID to a GeoDestination. This simplest case has been implemented in GeoNet. Table 3 below shows the IPv6 multicast addresses assigned for a particular GeoDestination ID corresponding to a radius around the target GeoDestination. In this specific case, the centre of the area where the packet shall be GeoBroadcast is centred on the GeoNet OBU and is retrieved at the C2CNet layer and thus unknown to the IP layer.

IPv6 multicast address	GeoDestination ID	GeoDestination
ff1e:0:0:0:0:0:0:1	1	500m range
ff1e:0:0:0:0:0:0:2	2	1000m range

Table 3: Example of IP multicast address mapping

Multicast addresses required for IPv6 geonetworking and defined in the context the GeoNet project will of course be of the temporary type though an operational development would required allocation by IANA.

11.3. IPv6 multicast group management with Multicast Listener Discovery (MLD)

The protocol used to manage the group membership on an IPv6 link is the **Multicast Listener Discovery** protocol known as **MLD**. It is based on IGMPv3 used in IPv4. It specifies separate behaviours for multicast address listeners (multicast hosts or routers that listen to multicast packets) and multicast routers.

In the context of GeoNet, we focus on the operation of the MLDv2 protocol [RFC 3810]. Compared to MLDv1, MLDv2 includes the source filtering mechanism which enables hosts and routers to specify which multicast sources should be allowed or blocked for a specific multicast group. As a consequence, it defines new MLD message formats. MLD routers perform the router part of the protocol on each of their attached links whereas multicast listeners perform the protocol over all interfaces on which multicast reception is supported. Note that the use of MLDv2 also simplifies the development work performed by the GeoNet partners as new Linux kernel versions are natively MLDv2 enabled.

The listener part consists of reporting to the multicast group the listeners that wants to join and eventually the sources from which it wants to receive multicast packets. This is done by sending the Multicast Listener Report messages.

The router part consists of sending the Multicast Listener Queries on the interfaces where multicast is supported and of intercepting the multicast Listener Report and maintaining a state for every reported multicast group.

MLDv2 defines two types of messages:

- Multicast Listener Queries are MLD messages sent from the local router to the all node link local address (FF02::1) on all the interfaces where MLD is supported. There are three types:
    - General Query: The router sends periodically General Multicast Queries to learn multicast address listener information from an attached link.
    - Multicast Address Specific Query: The router sends this kind of query to learn if a particular multicast address has any listeners
    - Multicast Address and Source Specific Query: The multicast router Sends these queries to learn if any of the sources from the specified list for the particular multicast address has any listeners.
- Multicast Listener Reports are MLD messages that are sent from hosts to the all MLDv2 router address FF02::16 in response to the MLD queries. There are three types:
    - Current State Record: Sent by a host in response to a Multicast Address and Source Specific Query to specify the INCLUDE or EXCLUDE mode for every multicast group in which the host is interested.
    - Filter mode change report: The host sends this report to the MLD router if the filtering mode (INCLUDE, EXCLUDE) changes
    - Source List change record: If the list of the sources changes, the host sends this report to inform the router

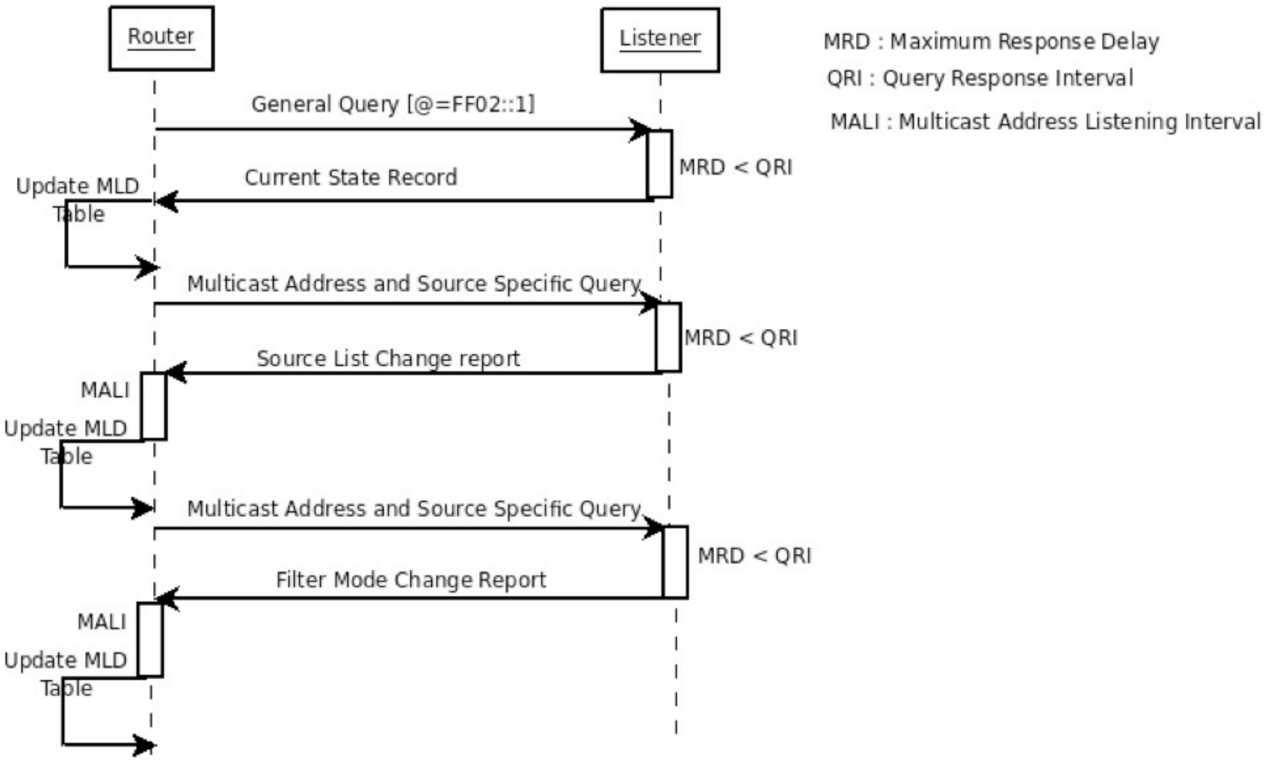

Router
Listener
MRD : Maximum Response Delay
QRI : Query Response Interval
MALI : Multicast Address Listening Interval
General Query [@=FF02::1]
Current State Record
MRD < QRI
Update MLD Table
Multicast Address and Source Specific Query
Source List Change report
MRD < QRI
MALI
Update MLD Table
Multicast Address and Source Specific Query
Filter Mode Change Report
MRD < QRI
MALI
Update MLD Table

Figure 21: Flowchart of MLDv2 functionality

The router maintains a state for every multicast address on every attached link. In MLDv2, the multicast address state consists in addition to timers of a list of sources and a filtering mode. Figure 21 is a simplified sequence diagram used to explain the operation of the MLDv2 Protocol.

11.4. Static multicast packet delivery

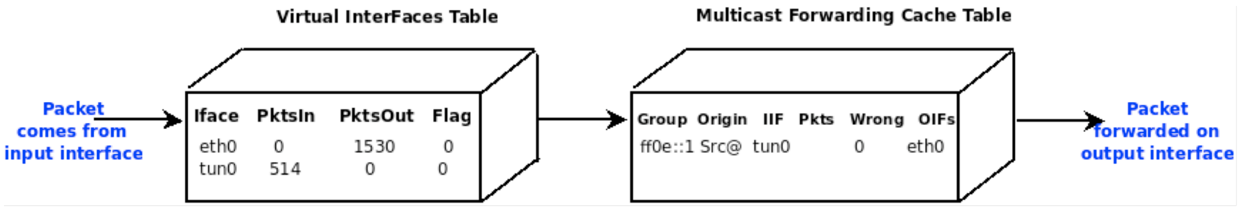
In this section, we explain a static approach for routing multicast packets in the GeoNet domain.

11.4.1. Operational interfaces & routing table setup

The multicast routing is based on three structures as shown in Figure 22:

- A Virtual InterFaces (VIF) table in which are saved the multicast installed interfaces as an abstraction for physical and tunnel interfaces
- A Multicast Forwarding Cache (MFC) table is a table which contains statistics about the sent and received packets for a multicast group and an origin address. It is used to give a state of the multicast forwarding table. It is composed of a set of (S,G) entries with a unique incoming interface and a set of outgoing interfaces
- A multicast routing table base which contains the multicast routes and is an instance of the routing table

In Figure 8 tun0 and eth0 are two multicast installed interfaces in the VIF table. The MFC table contains one entry bound to the group ff0e::1 (taken as an example). In this example eth0 is a physical input interface of multicast packets and tun0 is a tunnelling output interface. They are added statically to the VIF table.


Packet comes from input interface
Packet forwarded on output interface

Virtual InterFaces Table

Iface	PktsIn	PktsOut	Flag
eth0	0	1530	0
tun0	514	0	0

Multicast Forwarding Cache Table

Group	Origin	IIF	Pkts	Wrong	OIFs
ff0e::1	Src@	tun0		0	eth0

Figure 22: The operational Interfaces and Routing structures for multicast packet delivery

The MFC table uses the installed interfaces to set up an entry containing the multicast group bound to the input and the output interface.

11.4.2. Interaction with other modules

The general diagram below explains the configuration of multicast in the GeoNet domain. It shows the packet journey from an IPv6 multicast sender located in an in-vehicle network and attached to GeoNet OBU1 to an IPv6 multicast receiver located in another in-vehicle network and attached to GeoNet OBU2.

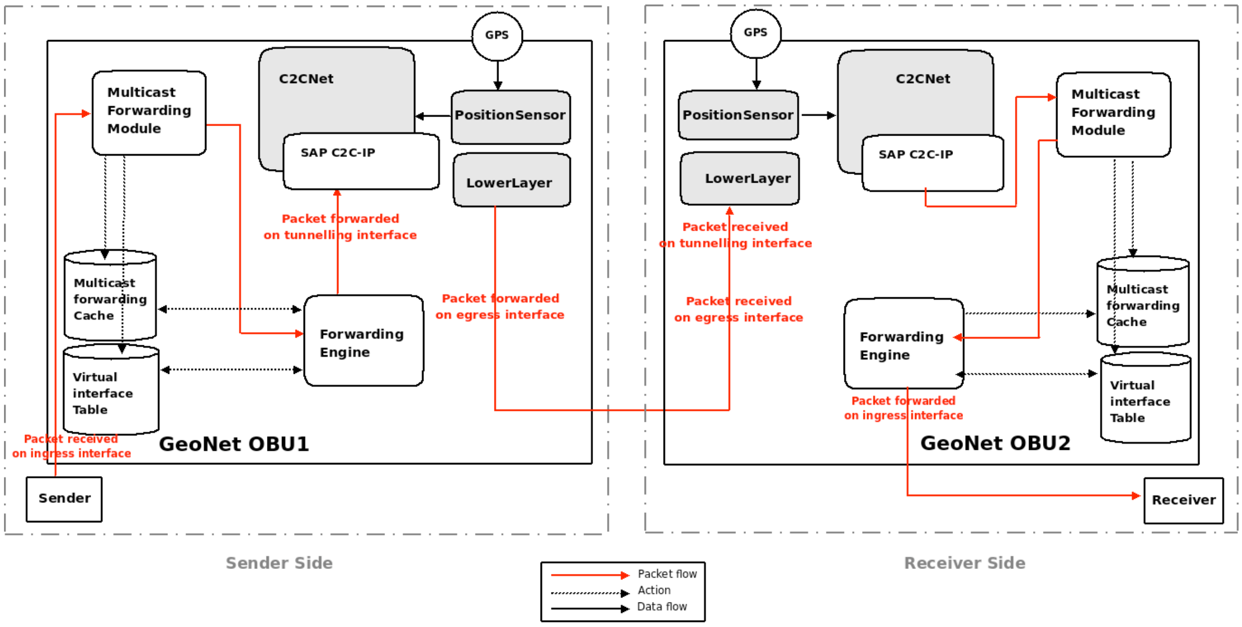

GPS
Multicast Forwarding Module
C2CNet
PositionSensor
SAP C2C-IP
LowerLayer
Packet forwarded on tunnelling interface
Packet forwarded on egress interface
Multicast forwarding Cache
Forwarding Engine
Virtual interface Table
Packet received on ingress interface
GeoNet OBU1
Sender
Sender Side
GPS
PositionSensor
C2CNet
Multicast Forwarding Module
SAP C2C-IP
LowerLayer
Packet received on tunnelling interface
Packet received on egress interface
Forwarding Engine
Multicast forwarding Cache
Virtual interface Table
Packet forwarded on ingress interface
GeoNet OBU2
Receiver
Receiver Side
Packet flow
Action
Data flow

Figure 23: Design of multicast interaction with other GeoNet modules

11.4.3. IPv6 multicast operations performed in the GeoNet domain

Referring to figure 24, we explain the operations performed on the following entities: Multicast Routers (GeoNet OBU1 and GeoNet OBU2), the Multicast Receiver and the Multicast Source (GeoNet-aware nodes or legacy IPv6 nodes).

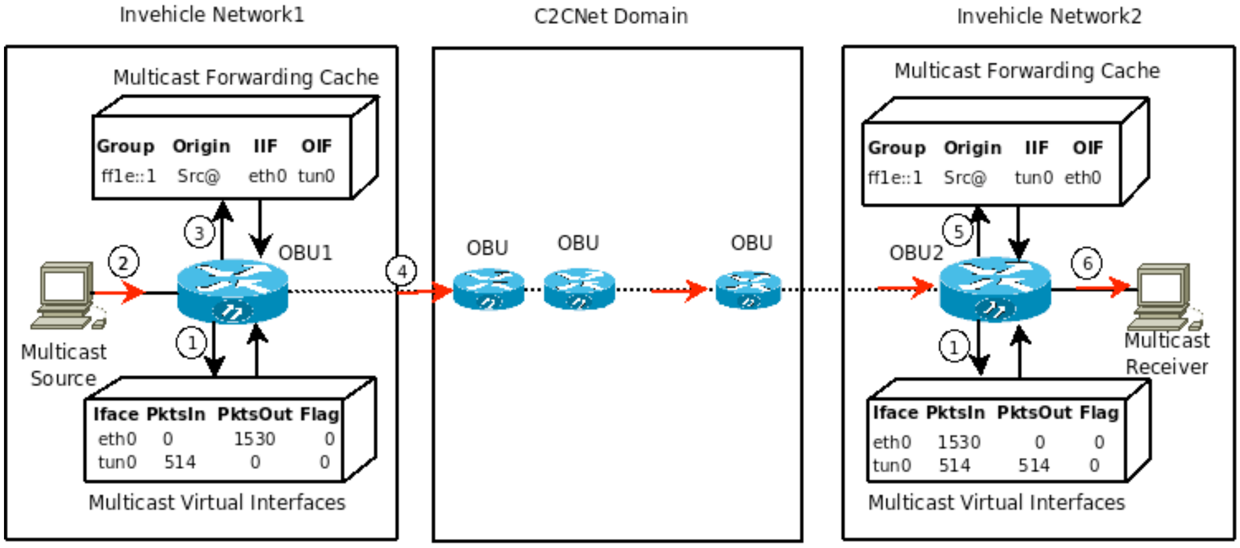


Figure 24: The static multicast delivery mechanism in GeoNet

The multicast source is attached to OBU1, one of the *multicast* receivers is attached to OBU2. eth0 is the interface that connects the mobile router OBU1 to the MNN. tun0 is the virtual interface created on the C2CNet layer to send packets on the C2CNet link.

Vehicle-based and Roadside-based communication scenarios are defined in [GeoNetD1.2] Section 4 are following this set of operations. For Internet-based communication scenarios (e.g. when IPv6 multicast packets reporting traffic hazard are sent from a road traffic center to all the vehicles in a GeoDestination – this is the scenario detailed in [GeoNetD1.2] Section 4.5 and correspond to the 3rd action i.e. IM1 type of scenario), the packet must first be routed in the Internet infrastructure until it reaches a GeoNet RSU where it is GeoBroadcast at the C2CNet layer and IPv6 multicast in the GeoNet domain. In order to achieve this, the multicast source must first generate the IPv6 multicast packet and then tunnel it to the GeoNet RSU (IP-in-IP unicast) where it is decapsulated. This is a simple scenario, and more complex scenarios could be handled.

Operations performed on the sending side (In-vehicle Network 1):

1. The multicast router creates statically a multicast VIF interfaces in which are put the multicast installed interfaces (operation 1 in the above figure).

2. Once the multicast router OBU1 receives the multicast packets from its attached multicast source (operation 2), it verifies the installed multicast interfaces from the Virtual InterFaces table and adds the source address of the packets to the Multicast Forwarding Cache entry.

3. If the output interface is valid and the hop limit of the packet is greater than 1, the multicast packets are sent down to the C2CNet egress interface (tun0), they are encapsulated by the C2CNet layer and finally are sent out onto the wireless link.

Operations performed on the receiving side (In-vehicle Network 2):

1. When the multicast packets are received by multicast router OBU2 the C2CNet layer decapsulates them and sends them up to the IP layer (tun0).

2. Multicast router OBU2 verifies if the output interface exists in the Virtual InterFaces table and adds the source address of the packets to the Multicast Forwarding Cache entry

3. If the output interface is valid and the hop limit of the packet is greater than 1, it forwards the packets on its ingress interface eth0

11.5. MLD Forwarding Proxy

MLD Forwarding Proxy [RFC 4605] is a simplified approach for multicast routing used to avoid the deployment of PIM multicast routers in a given network domain. Each router in the network domain is a MLD Forwarding Proxy. Only the border router connected to the multicast infrastructure is performing PIM. This approach relies on a spanning tree which connects the routers of the same network domain to the border router considered as the root of the tree.

11.5.1. Overview of MLD Forwarding Proxy

As any MLD router, each proxy performs the group management part in its downstream interfaces to learn the multicast groups having listeners on its subnet. It maintains a database of the membership information in each of its downstream interfaces. The membership database is a set of records of the following form:

{ multicast address, group timer, filter-mode, source-element list }

Once the Proxy receives an MLD Listener Report on its downstream interface, it updates its Membership database and send a Multicast Listener Report to the Proxy directly linked to it. As a result, the root of the tree will receive a Multicast Listener Report of all the multicast groups in the same network domain. Furthermore, the root can forward the multicast packets sent to the groups which there are receivers belonging to.

Figure 25 explains the MLD Forwarding Proxy topology.

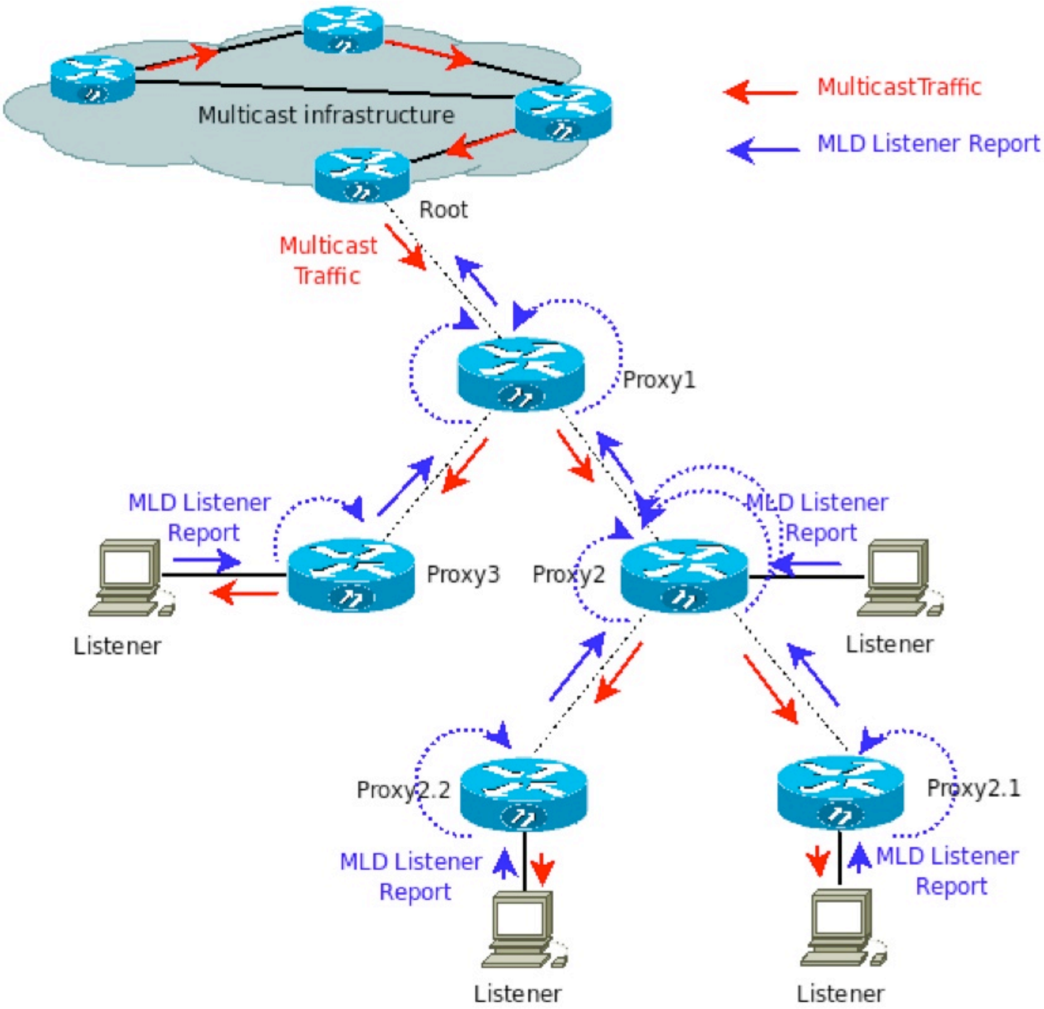


Figure 25: MLD Forwarding Proxy topology

11.5.2. MLD Forwarding Proxy in the GeoNet domain

In GeoNet, MLDv2 is used to discover the presence of multicast receivers within the GeoNet domain. To ensure the packet delivery from the multicast source to the receivers in the GeoNet domain, all GeoNet OBUs and GeoNet RSUs are performing MLD forwarding proxy.

MLD Forwarding Proxy provides a simple mechanism of propagating the information about the multicast group in the local network domain by informing the directly attached router about them. This is done by means of replicating the Multicast Listener Report received on the ingress interface to the egress (i.e. C2CNet) interface. This provides a dynamic knowledge about the groups for the other GeoNet OBUs and GeoNet RSUs. In GeoNet, each GeoNet OBU that has listeners in the in-vehicle network can inform the other routers belonging to the same GeoNet domain about the groups that its MNNs are interested in thanks to its MLD Forwarding Proxy capabilities. It can expect then receiving multicast traffic from one of the surrounding GeoNet OBUs and GeoNet RSUs that have a multicast source sending to the previously reported group.

11.5.3. Operational Interfaces

The behaviour of a router acting as an MLD Forwarding Proxy on its downstream interface is different from its behaviour on its upstream interface.

Take notice here that the definition of the downstream and the upstream interfaces of the MLD Forwarding Proxy depends on where the multicast packet comes from (output interface) and where they should be forwarded (input interface):

- The downstream interface: In this interface, the multicast router performs the Router part: it sends the Multicast queries to learn the presence of multicast listeners and receives Multicast Listener Report.

- The upstream interface : In this interface, the multicast router performs the listener part: it sends the Multicast Listener Report about the groups for which it has subscribers from its MNNs and receives multicast traffic. This interface could be the C2CNet tunnel, which is the interface used in the case of a multicast communication in the same GeoNet Domain.

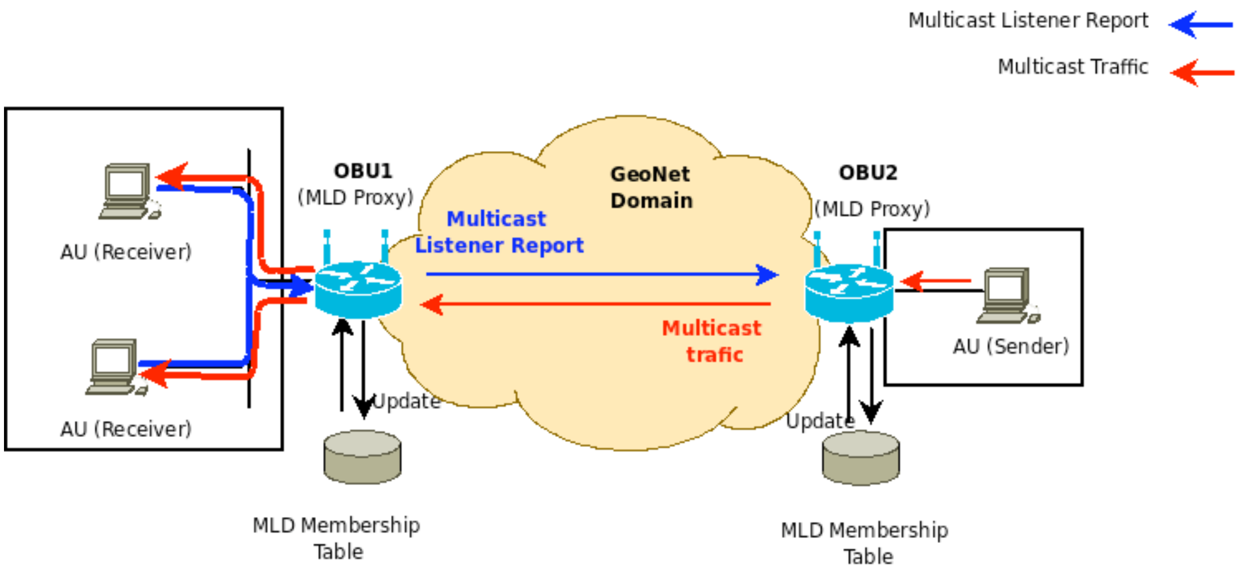


Figure 26: The operational interfaces of the MLD Forwarding Proxy

Figure 26 shows the example of two MLD Forwarding Proxies: one is OBU1 to which are attached multicast receivers and the other is OBU2 to which is attached the multicast source.

11.5.4. Main Operation of the MLD Forwarding Proxy

Referring to figure 27, we explain the operations performed on the following entities: Multicast Routers (GeoNet OBU1 and GeoNet OBU2), the Multicast Receiver and the Multicast Source (GeoNet-aware nodes or legacy IPv6 nodes).

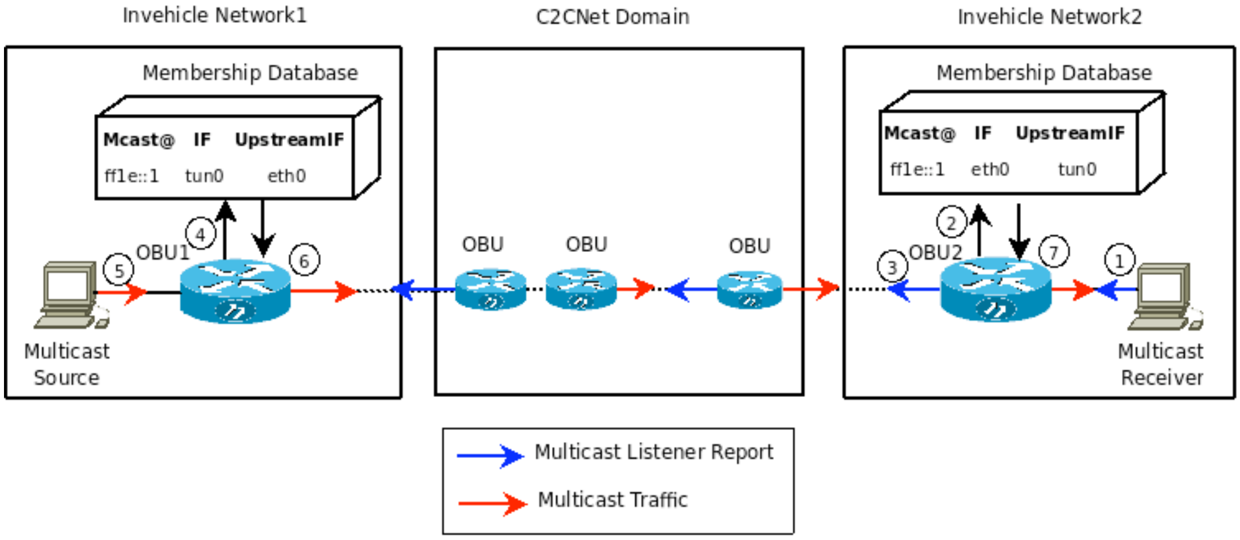


Figure 27: The MLD Forwarding Proxy Functionality in GeoNet

Operations performed on the receiving side (in-vehicle network 2):

1. The Multicast Receiver in the in-vehicle network 2 sends a Multicast Listener Report to OBU2 (operation 1 in figure 27)

2. OBU2 updates its Membership database by adding the multicast group address and the interfaces according to the report it receives (eth0 in figure 27). It will then forward the packet on its upstream interface (tun0) from which it expects receiving multicast data (operation 2 in figure 27)

3. When it receives the multicast data, it checks the Membership Database to determine where to forward the multicast packet and forwards the packets to the right interface (operation 7 in figure 27)

Operations performed on the sending side (in-vehicle network 1):

1. Once it receives the Multicast Listener Report sent by OBU2 on its C2CNet interface (tun0) OBU1 updates its Membership Database by adding the multicast group address and the tun0 interface as ingress interface (operation 4 in figure 27)

2. Once it receives the multicast traffic from its attached Sender (operation 5 in figure 27) on eth0 which is considered as its upstream interface in this case, it forwards it on the C2CNet link (operation 6 in figure 27).

12. Management Layer

12.1. Module 0A: Geo-destination

This module facilitates cross-layer information sharing between C2CNet layer, IPv6 layer, and Application layer. The main purposes of this module are:

- To coordinate the mapping between a GeoDestination IDs and the corresponding information defining the boundary of the geographical area.
- To share neighbour location awareness information with the Application layer, so that an application may select the required counter-parties for its communication. This information sharing is provided through the SAP MNG-UL. This provides applications with sufficient information for selection of relevant GeoDestination in their transactions.

This module is specified as an add-on to the rest of GeoNet modules; it means that other GeoNet modules are able to perform their basic functionality without the implementation of module “0A: Geo-destination”, and furthermore this module can be seamlessly added into an existing GeoNet implementation. The information management of this module and its management SAPs (MNG-C2C, MNG-UL) can be possibly implemented through an SQL database mechanism.

12.1.1. Coordination of GeoDestination id mapping

The requirement for this functionality depends on the selected mechanism for IPv6 GeoDestination encoding. Applications configure into the 'GeoDestination ID Table' needed GeoDestination IDs along with corresponding information that defines the boundary of the geographical area. The type of field to be recorded in the table may vary according to the approach used for GeoDestination encoding. The mapping procedure matches GeoDestination information by GeoDestination IDs.

Whether the same 'GeoDestination ID Table' is shared by several applications, or whether each application would have its own 'GeoDestination ID Table' shall be determined in the future.

12.1.2. Collection of location awareness information

Neighbour information from incoming beacons is passed from C2CNet layer to module “0A: Geo-destination”, which saves this information into its 'Location Table' database. This procedure is illustrated on the following figure. A maintenance procedure ensures that old entries are periodically removed.

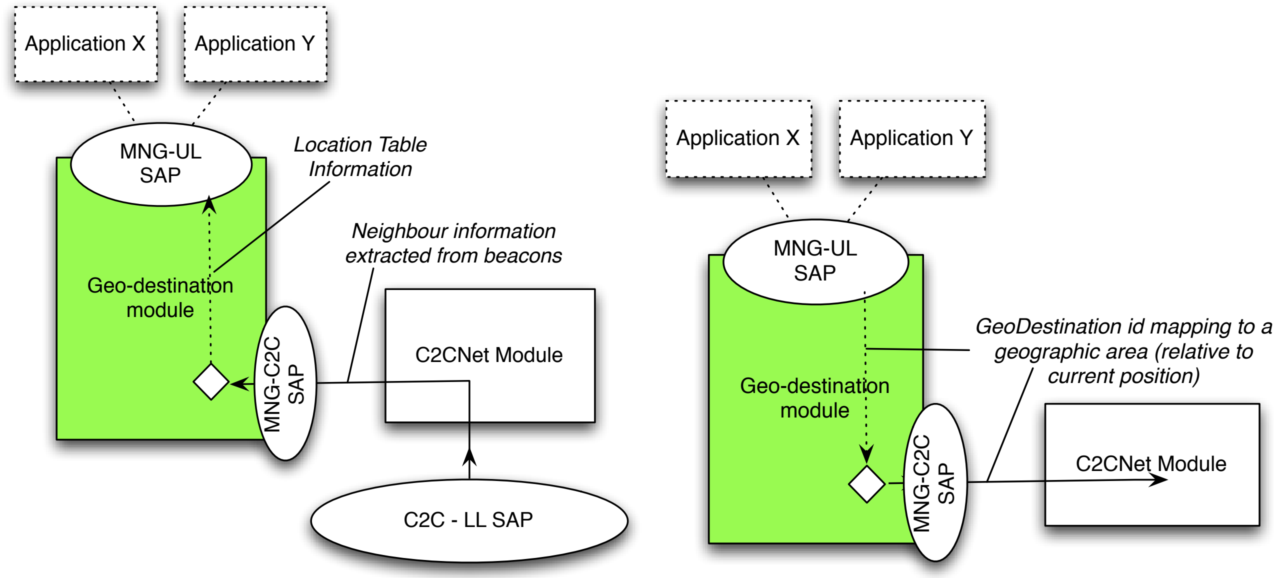


Figure 28: Geo-destination module data flows

12.2. Module 0B Security and privacy module

The security and privacy module is in charge of tackling the security and privacy concerns analysed in detail in [GeoNetD1.2]. It provides C2CNet IDs to the C2CNet layer (module 2.5A “Geo-routing”) and the IP layer (module 3A “IP Forwarding”) through SAP MNG-C2C and SAP MNG-IP respectively.

We analyse in the following sub-sections requirement by requirement how the GeoNet specification tackles – as a whole – the security requirements identified and described in [GeoNetD1.2].

12.2.1. Location Privacy

The module allows a GeoNet OBU to generate a set of IPv6 addresses, by making use of different C2CNet ids (pseudonyms). Each GeoNet OBU is provided with a set of C2CNet ids, and therefore several IPv6 addresses can be generated by each GeoNet OBU. Then, GeoNet OBUs may change the address they are using periodically to make tracking attacks harder to perform.

The communication between the Location Privacy module and the related modules is managed by SAP MNG-C2C and SAP MNG-IP, respectively. The communication is unidirectional as Location Privacy module sends to these interfaces the messages regarding the current C2CNet ID to be used. Module 2.5B 'Geo-Routing' and Module 3A 'IP Forwarding' listen to the interface waiting for a new C2CNet ID to be issued. When this happens they update their information coherently.

The current C2CNet ID is updated with a time driven policy. Location Privacy module chooses among the available C2CNet IDs with a well defined algorithm that is chosen by parameter. The module supports different ways of managing the list. The default one is

Round Robin where all the available C2CNet ID are used sequentially. The frequency of the updates can be set as parameter.

12.2.2. Revealing geographic location from the IPv6 address used as communication identifiers

A severe attack that could be enabled in a GeoNet domain (or more generally, by the integration of IPv6 and geonetworking in a common architecture) is the fact that the location of a node may be revealed (or be guessed) from any IPv6 enabled node (i.e. Potentially any node in the Internet). Therefore, this deserves special attention and the GeoNet specification has to ensure that location information cannot be directly obtained from the IPv6 address.

In GeoNet, IPv6 prefixes used by GeoNet OBUs and GeoNet RSU do not have any geographic location information embedded (they are normal IPv6 prefixes). The interface identifier of the IPv6 addresses does not directly contain any geographic location information, since it is generated from the C2CNet id.

The only way of resolving the location of a node based on its IPv6 address is to access the location service (note that the C2CNet id – which is used as the key to ask to the location service – can be easily obtained from the IPv6 address). However, only authorised nodes can access to the location service deployed in the GeoNet domain, and therefore this attack could only performed by nodes belonging to the GeoNet domain. This type of attack (i.e. disclosing geographical information of nodes belonging to the same GeoNet domain) is also possible in a GeoNet domain and has nothing to do with integrating IPv6. Therefore, solutions not specifically designed for GeoNet domain could also be adopted by GeoNet to protect against this types of attacks.

12.2.3. Secure binding between the IPv6 address and the C2CNet id

Another security aspect that has to be tackled is related to the binding between the IPv6 address and the sub-IP layer address. Since in the GeoNet architecture the C2CNet plays the role of the sub-IP layer, a secure binding between the addressing scheme used in each layer is required.

GeoNet IPv6 design uses a very secure binding, since there is a one-to-one mapping between the IPv6 interface identifier and the C2CNet id. This guarantees that it is not possible to bind an IPv6 address with a false C2CNet id (i.e. a malicious node with C2CNet id *M* cannot try to make a node B to bind an IPv6 address Pref01::A/64 with the C2CNet id *M)*.

12.2.4. IPv6 address spoofing

In addition to the previously described attack, another concern that needs to be address is the following. A malicious node may try to spoof the IPv6 address of a legitimate node (for example to steal a node's identity or to perform a Denial-of Service attack).
The protection against IPv6 address spoofing in GeoNet is provided by the fact that there is a secure binding between the IPv6 address and the C2CNet id, and that it is assumed that C2CNet address spoofing is not possible. The latter needs to be guaranteed by a mechanism out of the scope of GeoNet (C2CNet id spoofing has to be avoided in any C2C-CC based architecture, even if IPv6 is not supported).

12.3. Module 0C: Position Sensor

This module delivers geographic position data of a GPS module to the GeoNet protocol stack through SAP MNG-IP and C2C-IP. In the first phase the functionality is mainly provided by the Open Source tool gpsd (http://gpsd.berlios.de/). The position sensor module provides independence of the method of gathering position information.

12.3.1. Algorithm description

No algorithms are used in the Position Sensor module. It forwards the GPS data to GeoNet in a defined GeoNet specific order and format. The module does not care if the specified accuracy is reached by the used devices.

The module is initialised during start-up of GeoNet from a global initialisation routine. The GeoNet module that needs geographic positioning data should listen to a later defined port. Every time when the positioning data changes the positioning sensor module sends a message to the socket.

12.3.2. Description of module interfaces

Input-Interface:

None specified. For every kind of device or application a different kind of input is used.

Output-Interface:

The following message structure will be sent from the positioning sensor module to C2CNet layer:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     type      |   validity    |    reserved   |   reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        time stamp                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        latitude                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      longitude                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            speed              |         heading               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          altitude             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Parameters:

Name	Description
type	Index of what service is used. (currently only 0 for GPS)
validity	0 if the data is old and not accurate or the GPS device has no data. Not 0, if there is valid data in the message.
time stamp	UTC time in seconds, when the GPS data was calculated, NOT the time this message was generated.
latitude	the latitude of the global position in 1/8 microdegree
longitude	the longitude of the global position in 1/8 microdegree
speed	current speed in 0.01 meters per second
heading	current curse in 0.005493247 degrees
altitude	the altitude (meter over mean sea level)

13. C2CNet Packet formats

This chapter defines the encoding of geonetworking protocol data, with the exception of C2C packets' common header, which is defined by the Car-2-Car Communications Consortium (C2C-CC).

The byte order of all field values is big endian.

13.1. Packet type definition

Each C2CNet packet has the field packet type and subtype in the common header, which are used to identify the packet.

Type	Value	Description
C2C_ANY	0	unspecified
C2C_BEACON	1	Beacon
C2C_UNICAST	2	GeoUnicast
C2C_GEOANYCAST	3	GeoAnycast
C2C_GEOBCAST	4	GeoBroadcast
C2C_TSB	5	Topologically-scoped broadcast
C2C_LS	6	Location service

Table 4: C2C Packet types

Type	Value	Description
C2C_LS_REQUEST	0	Location service request
C2C_LS_REPLY	1	Location service reply
C2C_GEOBCAST_CIRCLE	0	GeoBroadcast circle area
C2C_GEOBCAST_RECT	1	GeoBroadcast rectangular area
C2C_GEOANYCAST_CIRCLE	0	GeoAnycast circle area
C2C_GEOANYCAST_RECT	1	GeoAnycast rectangular area

Table 5: C2C Packet subtypes

13.2. Packet format description

13.2.1. Beacon (from C2C-CC common network header)

GeoNet re-uses the network common header as defined in the C2C-CC Demo. 2008. Since this C2C-CC common header has not yet fully publicly published by the C2C-CC consortium, GeoNet will just refers to it without disclosing it. In all the following definitions, the C2C Common Header refers to C2C-CC common header as defined for the C2C-CC 2008 demo.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Common Header                         |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 29: C2C Beacon Packet

The C2C Common Header includes a position vector which contains position information of the sender node.

13.2.2. Location service

The location request packet contains the common header and dedicated extended header. The location request messages are sending using topologically-scoped broadcast mechanism. It sets both the source node position vector and sender node position vector from its local position data. There is a hop limit field in the common header which is used to scope the spread of the message and is decreased by each forwarder node.

The following data values are set in the common header:

- Protocol type: 6
- Protocol subtype: 0
- Hop limit: 5
- Length is set to C2C packet length

The Request ID is the C2CNet ID of the node, whose location is being requested.

If location request is forwarded by an intermediate node, the intermediate node will update the sender position vector in the packet.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Common Header                         |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                  Source node position vector                  :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Request ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 30: Location request

The location reply message contains the current position of the node as well as the time stamp. A location reply packet is issued by a node receiving a location request packet which has the same identifier as the look-up ID contained in the location request. The location reply message is sent using a GeoUnicast packet with the source node set to be the replying node and destination node set to be the requesting node.

The following data values are set in the common header:

- Protocol type: 6
- Protocol subtype: 1
- Hop limit: 5
- Length is set to C2C packet length

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Common Header                         |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                  Source node position vector                  :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Destination node ID                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Latitude                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Longitude                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 31: Location reply

13.2.3. GeoUnicast packet

The following data values are set in the common header:

-**Protocol type**	: 2	(GeoUnicast)
-**Protocol subtype**	: 0	
-**Hop limit**	: 255	(set to maximum)
-**Length**	: is set to C2C packet length	

The C2CNet ID field of common header has the value of selected next forwarder's C2CNet ID.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Common header                          :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
|
:                 Source node position vector                   :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Destination node ID                    |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Latitude                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Longitude                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            Payload                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 32: GeoUnicast packet

13.2.4. GeoAnycast packet

The following data values are set in the common header:

-**Protocol type** : 3 (GeoAnycast)
-**Protocol subtype** : 0 (circle GeoAnycast area)
-**Hop limit** : 255 (set to maximum)
-**Length** : is set to C2C packet length

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Common header                          :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |
|
:                    Source node position vector                :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved           |             Radius            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Latitude                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Longitude                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            Payload                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 33: GeoAnycast packet

13.2.5. GeoBroadcast packet

The following data values are set in the common header:

-**Protocol type**	: 4	(GeoBroadcast)
-**Protocol subtype**	: 0	(circle GeoBroadcast area)
-**Hop limit**	: 255	(set to maximum)
-**Length**	: is set to C2C packet length	

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Common header                          :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Source node position vector                :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Reserved          |             Radius            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Latitude                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination node Longitude                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            Payload                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 34: GeoBroadcast packet

13.2.6. TopoBroadcast packet

The following data values are set in the common header:

-**Protocol type**	: 5 (TopoBroadcast)
-**Protocol subtype**	: 0
-**Hop limit**	: set to broadcast distance in terms of hop
-**Length**	: is set to C2C packet length

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                        Common header                          :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                    Source node position vector                :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            Payload                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 35: TopoBroadcast packet

13.3. Field Name Definitions

Common Header
The C2C-CC common header, which includes the Position Vector of the last forwarder (node from which the packet comes).

Position vector
includes: C2CNet id, Timestamp, position in Latitude, Longitude and Altitude along with their accuracy, and Speed and Heading along with their accuracy.

Source ID
The identifier of the node generated the corresponding packet.

Source TS
The time stamp of the packet, which corresponds to the time in milliseconds when the corresponding packet has been generated by the source node.

Source Latitude
The latitude of the source node when the corresponding packet has been generated.

Source Longitude

The longitude of the source node when the corresponding packet has been generated.

Destination ID
The identifier of the destination node.

Destination Latitude
The longitude of the destination. In case of GeoUnicast it corresponds to the longitude of the destination node. In case of GeoAnycast or GeoBroadcast it corresponds to the longitude of the center of the destination geo-area, which is defined in GeoNet project as circle.

Destination Longitude
The latitude of the destination. In case of GeoUnicast it corresponds to the latitude of the destination node. In case of GeoAnycast or GeoBroadcast it corresponds to the latitude of the center of the destination geo-area, which is defined in GeoNet project as circle.

Payload
Packet payload (data)

When subtype is set to 1 (circle) in the packet common header, the Radius corresponds to the radius of circle which limits the broadcast/anycast geographic area.

14. Limitations and Future Work

This chapter describes the limitations encountered in the course of GeoNet specification work, which have an impact on reaching some design goals set forth in the GeoNet Architecture document. For each identified limitation, a corresponding description of future work presents the roadmap for reaching the established architectural design goals.

While the 'GeoNet Adaptation Layer' chapter specifies initialisation and selection of a specific radio channel for the multi-channel operation support of the C2CNet layer, the current specification of C2CNet layer is designed to make use of just a single wireless channel. The main reason is that the ETSI standardisation of the 5.9 GHz profile has not been completed yet. The presently specified GeoNet system may support multi-channel operation only by running multiple instances of the C2CNet layer; with each instance configured to a different data link layer and upper layer. A suitable transport layer solution, such as the SCTP multi-homing mechanism, may bind these instances together. A future extension of GeoNet specification should define multi-channel operation of the C2CNet layer.

In the current GeoNet specification the C2C-IP SAP is the sole service access point for passing data to upper layers from the C2CNet layer. While the definition of a direct SAP between the C2CNet layer and a geonetworked transport layer is out of GeoNet project's scope, the definition of such additional SAP should be added in a future extension of GeoNet specification. Such extension will allow the use of both IPv6 based applications and non-IP applications that directly utilise geonetworking capabilities.

The current GeoNet specification does not support the handling of IP priorities, meaning the higher / lower priority IP data cannot be distinguished in the C2CNet layer. The main reason is that such priority handling can be meaningfully specified only after multi-channel C2CNet operation and corresponding congestion control mechanisms have been defined. With the presently specified GeoNet system priority handling can be done at the IPv6 layer using 'Flow ID' field. A future extension of GeoNet specification should define the mapping between the IPv6 'Flow ID' and the 'Priority' field in the common network header of C2CNet packets, as well as corresponding priority differentiation at the C2CNet layer.

While Annex C informatively describes a possible design of the MNG-UL SAP and a possible design of C2C-MNG SAP is outlined in the 'Collection of location awareness information' chapter, there is no exact specification presently for the C2C-MNG and MNG-UL SAPs. The main reason is that MNG-UL SAP has been judged to be outside the core GeoNet project scope. Consequently, there are two ways to provide applications with neighbour awareness information under the presently specified GeoNet system:

A: Through an application layer beaconing mechanism, which duplicates beacon information over the air interface. The drawback in this case is the additional air interface data overhead.

B: Through an external management module, which extracts incoming beacon information from data link layer socket before passing it to the C2C-LL SAP, and implements the MNG-UL SAP outlined in Annex C. The drawback is a complicated solution architecture.

A future extension of GeoNet specification should specify the C2C-MNG and MNG-UL SAPs for providing neighbour awareness information to the upper layers.

In order to have a precise application layer GeoDestination area decision mechanism for a particular transaction, the application layer neighbour location awareness described in above paragraph is required.

Furthermore, the most optimal approach should be selected among the area encoding mechanisms described in the 'Enabling geographical scope in IPv6 multicast'. Since each of the five described mechanisms has particular advantages and disadvantages, choosing the one or two selected approaches requires further analysis and simulation, and is left to a future version of GeoNet specification.

The current GeoNet specification discusses congestion control requirements and approaches, but does not specify an actual congestion control mechanism for the GeoNet system. The reason is partly the overly complex nature of congestion control issues, and partly the fact that an effective congestion control mechanism would take into account the specific radio channel definitions and priority assignment scheme, which therefore needs to be defined first. The currently specified GeoNet system therefore cannot give hard guarantees on latency or reliable delivery of important messages. A future extension of GeoNet specification should turn the discussion presented in Annex D into an effective link management mechanism for achieving wireless congestion control of the GeoNet system.

The presently specified C2CNet layer routing and location query service mechanisms are simple yet robust means of geonetworked data packet routing. These operationally straightforward algorithms are suitable for the first iteration of the GeoNet system, but can be eventually further optimised in terms of transmitted data overhead and latency. In particular, movement predictive routing approaches are known to be a step towards such optimisation. A future extension of GeoNet specification should therefore describe more sophisticated C2CNet layer routing and location query services.

The 'Application Support Library' describes the role application layer geonetworking support, but does not specify a specific support mechanism. Therefore receiver side filtering of received data relevance and the aggregation of similar data are out of the current GeoNet system's scope. A future extension of GeoNet specification should therefore describe such application layer support functionality.

It has been envisioned in the GeoNet architecture, that upper layer processing would be supported by application-layer support libraries, which perform following common tasks:

- Receiver side message filtering, for example to distinguish vehicles driving into an intersection – i.e. towards a destination coordinate – from vehicles driving out of it
- Information aggregation for combining GeoBroadcasts with same information content – i.e. same type of sensor information – originating from different sources

While the concept of such receiver side filtering and data aggregation is a valid one and should be part of the overall system, these specifications have not been developed in GeoNet. The reasons for limiting the scope on just the networking layers are the following:

- It was out-of-scope to work on application-level issues. It has been decided that GeoNet architecture will be more clean, understandable, and presentable if it works out just the networking layer solution.
- At the outset of GeoNet work, the Application workgroup of ETSI ITS has been in the initiating phase, and it was not clear whether the idea of such application support libraries would be suitable.

Development of envisioned application support libraries that work on top of GeoNet project results should be part of future work.

It is anticipated that a number of future GeoNet system implementations would be based on the present specification document - besides the two independent implementations that are produced within the GeoNet project. The compliance of these future implementations to the present specification can be ensured through the conformance test suite described in [GeoNetD4.1]. These future implementations of the present specification will furthermore find [GeoNetD5.1] and [GeoNetD7.1] results useful for performance benchmarking and simulations.

Annexes

ANNEX A: IPv6 Encoding of GeoDestination

In order to combine IPv6 and geonetworking and provide geographically scoped IP multicast services in vehicular networks, we need to consider how the geographic information defining the target geographical destination area (GeoDestination) is encoded at the IP layer. This topic is relatively new (an analysis of previous work on this is provided in [GeoNetD1.2] Annex C). GeoNet partners have identified four more or less complicated approaches, also detailed in conferences papers [Bernardos2009, Choi2008, Khaled2009a, Khaled2009b]. At the time of writing, we think it is not reasonable to take a decision on which solution is best because the right choice depends on parameters not specific to IPv6 geonetworking. One approach will have to be picked up by the ITS community and will require significant standardisation effort at the ETSI, ISO and/or IETF. Our intention at this stage is to feed the ITS community with the various approaches. As for what GeoNet is concerned, the most important is to demonstrate the concept of IPv6 geonetworking so a simplest approach (no encoding) was adopted for experimental purposes.

When the AU wants to send IPv6 multicast traffic to a specific geographic area, the geographic information needs to be encoded somehow into the IPv6 packet until it reaches the GeoNet OBU. Five approaches have been considered:

- Approach A: Encoding geographic information into IPv6 Tunnel destination address (an IP multicast address)
- Approach B: Encoding geographic information into an IPv6 Extension Header
- Approach C: Encoding geographic information into IPv6 Extension Headers of an IPv6 Tunnel from AU to OBU
- Approach D: Encoding geographic information directly into the multicast group-ID
- Approach E: Group ID identifies a GeoDestination known at the C2CNet layer

These approaches are not strictly exclusive; an actual implementation may allow simultaneous use of several approaches.

The study reported in this Annex considered the following design goals:

- To be as IP standard as possible. In order to guarantee inter-vehicle operability, the solution should not require strong modifications of the IP stack of the involved components.
- To introduce the minimum overhead in the radio interface. Due to the limited throughput constraints of the radio interface the solution overhead should be as reduced as possible.

- Not to require modifications to legacy IPv6 nodes (MNNs and CNs). The solution must allow these devices to communicate without performing any modifications in their protocol stack.

Approach A: Encoding geographic information into IPv6 Tunnel destination address (an IP multicast address)

The first approach is to encode the GeoDestination into the IPv6 destination field of an IPv6 multicast encapsulated into another IPv6 multicast packets and originated at the AU as seen in figure 36.

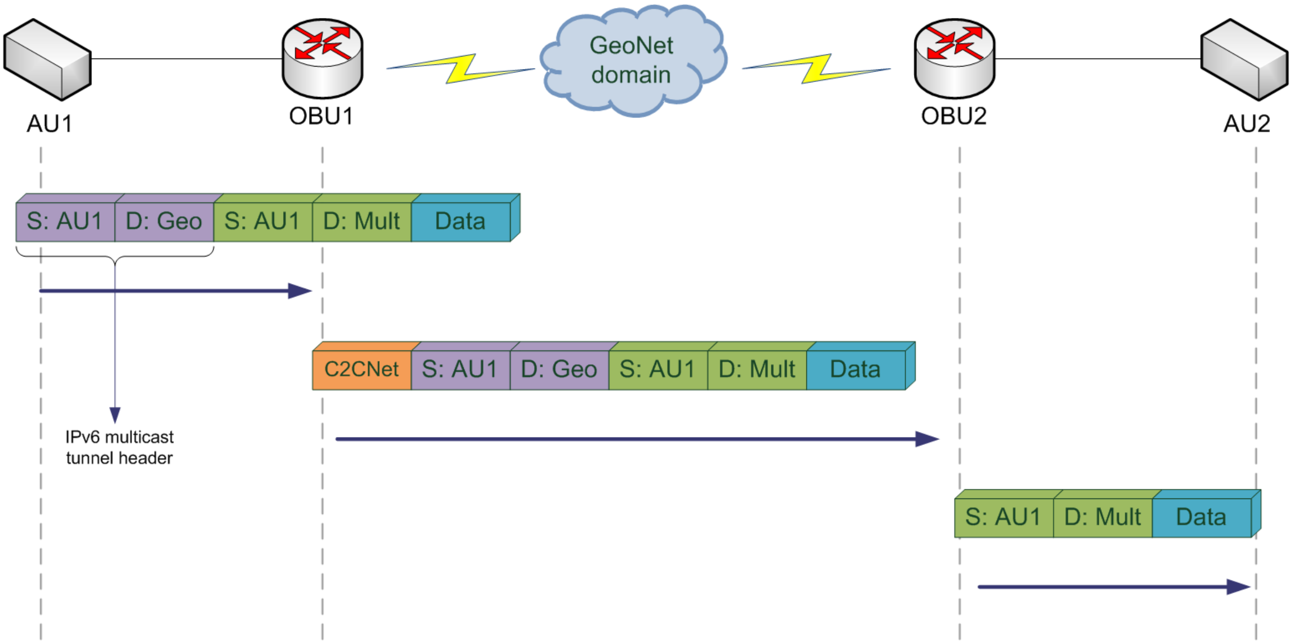


Figure 36: Approach A

The MNN (AU1) generate a standards IPv6 multicast packet with the destination multicast address set to the corresponding application group (*Mult* on the figure) and encapsulates it within another IPv6 multicast packet whose destination address contains (embeds) the GeoDestination (*Geo* on the figure). When the MR the MNN is attached to (OBU1) receives the packet, it checks the IPv6 destination address field and recognises that the address corresponds to a GeoDestination. Therefore it translates the GeoDestination for using it at the C2CNet layer.

Then the packet is forwarded at the C2CNet layer until it reaches the GeoDestination. MRs in this area (e.g. OBU2) transmit the packet to the upper layer (the IPv6 layer) which checks if the IPv6 multicast address of the outer header corresponds to a subscribed IPv6 multicast group. This means that the relation between the GeoDestination and the IPv6 multicast address must be univocal. A relative definition of a GeoDestination, e.g., 200 meters around the vehicle, is not allowed because although it works at geographic routing layer, at IP layer, destination nodes cannot be subscribed to a relative address.

In order to solve this problem, the GeoDestination at the IPv6 layer has been limited to predefined squares of fixed length. In order to provide granularity, squares of different sizes have been defined. How to binary encode the geographic information and square identification is explained below.

When a destination MR (OBU) receives the packet at the IPv6 layer, it decapsulate the packet and extracts the IPv6 multicast packet which it forwards it to the corresponding MNNs (AUs) subscribed to the IPv6 destination address of the inner packet.

Benefits:

- At MRs the IPv6 behaviour is the standard one, they forward to the lower layer according to upper layer destination address (as MAC multicast does with IP multicast), i.e., MRs map IPv6 destination to C2CNet layer destination directly.

Drawbacks:

- Destinations must subscribe to the group defined by the IP multicast address which is a location dependent address. Therefore relative definition of geographic areas is not allowed.

- Destination geographic areas are limited to a set of predefined squares of fixed length. Although some granularity is provided by defining squares of different size.

- The GeoDestination is transported twice, once at the C2CNet layer and once in the IPv6 tunnel. This causes overhead on the radio interface.

Binary encoding of geographic coordinates

For binary encoding of geographic coordinates, Hain [Hain2008] divides the world in squares of different size in function of the divisor value used, i.e., the greater the divisor value the smaller the size of the squares. In our scenario addressing world coordinates is not necessary so the region to address has been reduced to the European continent. Defining Europe latitude and longitude as 35ºN to 70ºN and 25ºW to 40ºE respectively, an “square” of 35 degrees of latitude and 65 of longitude is defined.

The following table shows the approximate length of the arc resulting of dividing 35 degrees of latitude by a factor; this factor is expressed in bits to measure its binary length. The eight resulting values are defined in order to provide eight different square sizes.

Bits	Arc length (km)
9	7.610
10	3.805
11	1.902
12	0.951
13	0.476
14	0.238
15	0.119
16	0.059

The following table shows the approximate length of the arch resulting of dividing 65 degrees of longitude by a factor at three values of latitude, 40º (corresponding to Spain latitude) 50º (Germany) and 60º (Sweden). These additional values are presented to give a view of the reduction of longitude arc length when latitude is closer to the pole.

Bits	**Arc length (km)**		
	40º	50º	60º
9	9.084	10.826	12.239
10	4.542	5.413	6.119
11	2.271	2.706	3.060
12	1.135	1.353	1.530
13	0.568	0.677	0.765
14	0.284	0.338	0.382
15	0.142	0.169	0.191
16	0.071	0.085	0.096

As seen in these tables, the more precise squares (the smaller ones) are obtained using the greater divisors, i.e., with a higher number of bits.

Therefore, for binary encoding the geographic areas for the solution of Section 3.1.1, the total number of bits needed is 35 which correspond to 32 bits for identifying the lower left corner of the square (16 bits for latitude, 16 bits for longitude) and 3 bits for identifying the square size.

According to IPv6 Addressing Architecture [RFC4291], IPv6 multicast addresses have the following structure:

```
|   8    |  4 |  4 |                  112 bits                   |
+------ -+----+----+---------------------------------------------+
|11111111|flgs|scop|                  group ID                   |
+--------+----+----+---------------------------------------------+
```

- *flgs* is a set of four flags which for the solution have the value 0001, i.e., non-permanently-assigned multicast addresses.
- *scop* is a 4-bit multicast scope value used to limit the scope of the multicast group. The value of this field is set to VEHICULAR (value to be assigned by IANA from the reserved ones).
- *group ID* identifies the multicast group, in the solution identifies the GeoDestination according to the previously described encoding; 3 bits for identifying the square size, 32 for the lower left corner of the square and the rest with 0.

```
|      77 bits          | 3 |    32 bits      |
+---------------------------------------------+
|00000000000000000000000|siz|  square coords  |
+---------------------------------------------+
```

siz is the square size. The values are as follows:

0	Tables 1 and 2 values of arc length for factor value 9
1	Tables 1 and 2 values of arc length for factor value 10
10	Tables 1 and 2 values of arc length for factor value 11
11	Tables 1 and 2 values of arc length for factor value 12
100	Tables 1 and 2 values of arc length for factor value 13
101	Tables 1 and 2 values of arc length for factor value 14
110	Tables 1 and 2 values of arc length for factor value 15
111	Tables 1 and 2 values of arc length for factor value 16

square coords are the coordinates of the lower left corner of the square. The sequence for address formation given the coordinates of a point is:

- Normalize the coordinates for origin of the allowed space (the European continent square 35N-70N and 25W-40E)
  - For latitude subtract 35 from the value
  - For west longitude subtract the value from 25
  - For east longitude add 25 to the value
- Divide resulting values by 35/2^factor for latitude and 65/2^factor for longitude.
- Convert each of the integers to 16-digit binary
- Prepend latitude to longitude into 32-bit result

Approach B: Encoding geographic information into an IPv6 Extension Header

The second approach is to encode the GeoDestination into an IPv6 extension header field of the IPv6 multicast packet originated at the AU as shown on Figure 37.

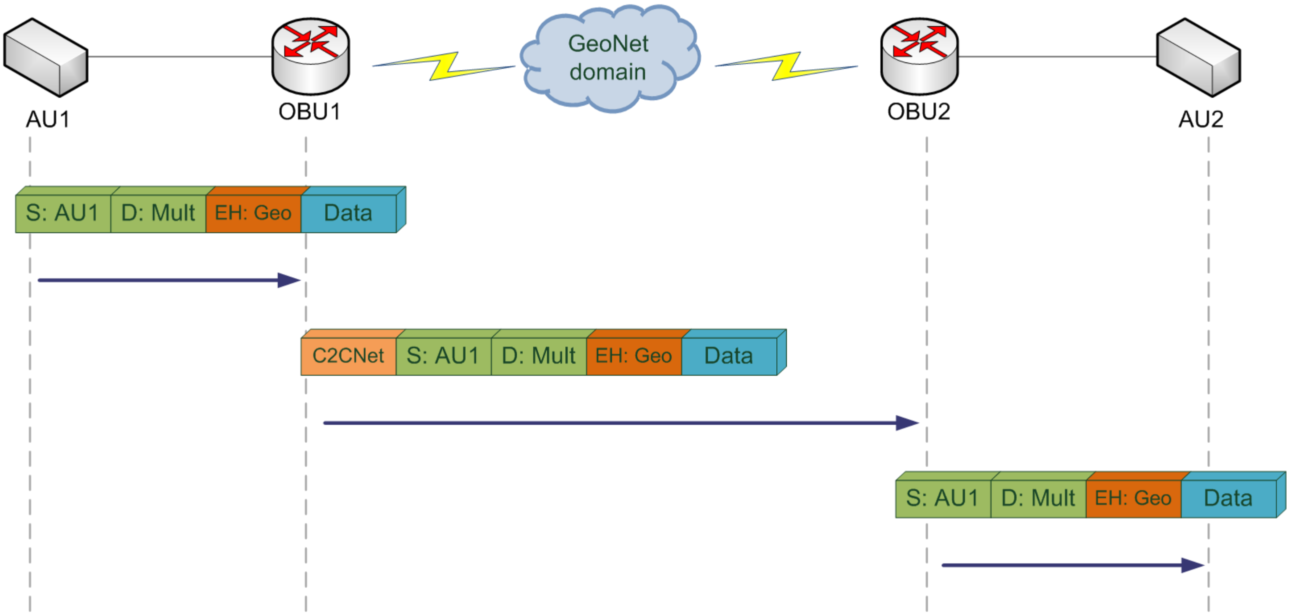


Figure 37: Approach B

The AU (AU1) generates an IPv6 multicast packets and adds a Hop-by-Hop Option extension header containing the GeoDestination. When the packet reaches AU's OBU (OBU1) the extension header is processed (processing Hop-by-Hop headers at every hop is mandatory in IPv6) and from it the OBU obtains the GeoDestination indicating where the packet shall be forwarded.

The packet is transmitted to the C2CNet layer where it is forwarded from source OBU (OBU1) to destination OBUs (e.g., OBU2) using geographic routing. When the packet reaches an OBU within the destination area, the C2CNet layer transmits the packet up to the IPv6 layer which forwards it onto the in-vehicle network if there are AUs subscribed to the IPv6 multicast group.

Encoding the GeoDestination into the IPv6 extension header allows using relative geographic addressing because this information is only used in AU's OBU of the sender. However, forwarding using an extension header instead of the destination address field is not the normal behavior of IPv6. This does not break IP because it only introduces changes in the OBU forwarding process. This is a debatable solution.

Benefits:

- Allows relative geographic addresses (e.g., 200 meters around the vehicle). The first approach does not allow it because destinations need to subscribe to the IPv6 geographic addresses thus they need to be absolute.

Drawbacks:

- Uses Hop-by-Hop Option extension header, this header is processed at every node although its information is only necessary at the first OBU.

- Unusual behaviour at OBUs' IP layer. OBU uses an extension header to obtain the layer 2 destination address.
- Radio interface overhead. The GeoDestination is transported two times, once at the C2CNet layer and once at the IPv6 layer within an extension header.

Approach C: Encoding geographic information into IPv6 Extension Headers of an IPv6 Tunnel from AU to OBU

The third approach is encoding the GeoDestination into an IPv6 extension header field of an IPv6 tunnel originated at the AU with destination AU's OBU as shown in the following figure.

The AU (AU1) generates an IPv6 multicast packet and encapsulates it into an IPv6 packet which destination address is the AU's OBU (OBU1). The GeoDestination is added as a Destination Option extension header of the outer packet. When the OBU receives the packet, it obtains the GeoDestination from the extension header and removes the tunnel, forwarding the IPv6 multicast packet within the C2CNet layer without introducing additional overhead.

As in the second solution, AU's OBU forwards the packet using an extension header instead of destination address thus this solution also forces an unusual IPv6 behaviour at the OBUs which is debatable.

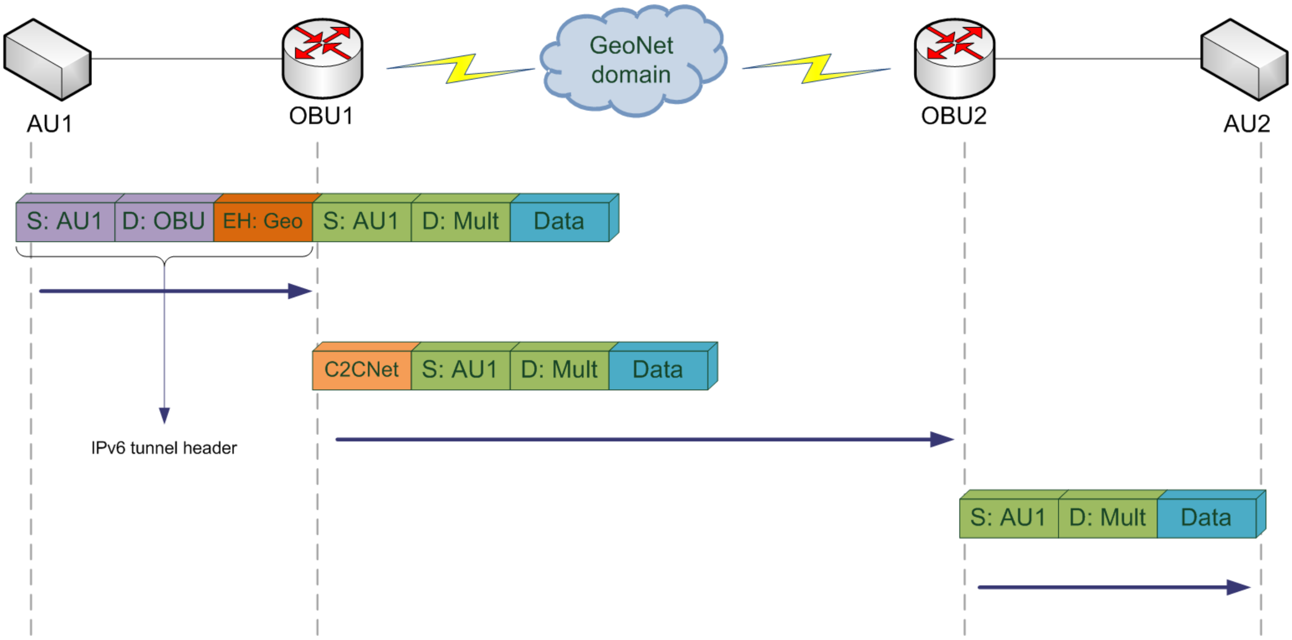


Figure 38: Approach C

Benefits:

- Allows relative geographic addresses.

- Does not introduce radio interface overhead. Geographic information at IPv6 layer is only propagated in the in-vehicle network.

- Geographic information at IPv6 layer is only processed once by using Destination Option extension header in the AU-OBU tunnel

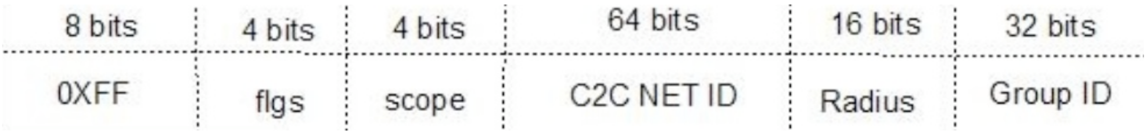
Drawbacks:

- Unusual behaviour at OBUs' IP layer. The OBU uses an extension header to obtain the layer 2 destination address.

Approach D: Encoding geographic information directly into the multicast group-ID

In this approach the GeoDestination is encoded directly into the 112-bits *group id* field of the IPv6 multicast address. There could be different GeoDestination encoding formats. A format similar to the one described in Approach A could be used. Figure 39 illustrates an other possible format of the GeoDestination where the C2CNet ID and radius of a GeoNet RSU around which packets should be GeoBroadcast is provided. Group ID indicates the type of application.

The part of IPv6 address information that will be directly processed by the C2CNet layer consists of: C2CNet ID and radius. Since this solution presumes knowledge of applicable C2CNet IDs, it can be used when the source AU is part of the vehicular network.



8 bits	4 bits	4 bits	64 bits	16 bits	32 bits
0XFF	flgs	scope	C2C NET ID	Radius	Group ID

Figure 39: Format of IPv6 multicast address using C2CNet GeoBroadcast

Notice that maintaining the group ID to 32 bits has a significance, since it provides a consistency between different types of IPv6 addresses. Indeed, in the case of addresses derived from unicast prefix, this field has a length of 32 bits [RFC3306].

Examples:

- The packet is delivered to a circular area around the source. When Radius is *1500 (0x5dc)*, GroupID is 1. The IPv6 multicast address could be considered as: *ff1e: 0000:0000:0000:0000:05dc:0001*.

- The packet is delivered to a geographic circular area around a selected specific node. When Radius is 1500 (0x5dc), GroupID is 1, C2CNet ID is *AAAABBBBCCCCDDDD*. The IPv6 multicast address of this area could be considered as: *ff1e:AAAA:BBBB:CCCC:DDDD:05dc:0001*.

Benefits:

- Allows relative geographic addresses and simple selection of relevant areas.
- Does not introduce radio interface overhead. Geographic information at IPv6 layer is only propagated in the in-vehicle internal network.

Drawbacks:

- Reduced range of multicast group IDs
- Source AU should be part of the vehicular network

Approach E: Group ID identifies a GeoDestination known at the C2CNet layer

In this approach each possible multicast IPv6 address is mapped in advance to the corresponding GeoDestination at the C2CNet layer. The GeoDestination information is not carried at all at the IP layer. The application and C2CNet layer must be coordinated through the use of module “0A: Geo-Destination” which contains a table recording mapping between GeoDestination ID and corresponding information defining the boundary of the geographical area with respect to the vehicle’s current position. The multicast packet would be generated by Geo-aware application on the MNN with an IPv6 multicast address embedding the GeoDestination ID (i.e. within the group ID field). The packet is received by the GeoNet OBU and transmitted to the C2CNet layer through SAP C2C-IP. The GeoDestination ID is thus learnt by the C2CNet layer. The C2CNet layer would determine the GeoDestination associated to the GeoDestination ID by querying module “0A: Geo-destination” through SAP MNG-C2C.

Quantitative evaluation

In this section a quantitative evaluation of the four proposed approaches for providing IP multicast services over geocast for vehicular networks is given. This evaluation does not cover approach E because that solution has been introduced in a late stage of the project work.

Overhead

The overhead introduced by the solutions is the following:

Sol.	In-vehicle network overhead	Radio interface overhead
A	40 bytes (IPv6 tunnel)	40 bytes (IPv6 tunnel)
B	16 bytes (12 bytes C2CNet destination position + 4 bytes Hop-by-Hop Option header)	16 bytes (12 bytes C2CNet destination position + 4 bytes Hop-by-Hop Option header)
C	56 bytes (40 bytes IPv6 tunnel + 12 bytes C2CNet destination position + 4 bytes Hop-by-Hop Option header)	None
D	None	None

Table 6: Solutions' overhead comparison

As seen in table 6, although approach C introduces the biggest overhead on the in-vehicle network it does not introduce any overhead on the radio interface. Therefore from a overhead point of view, approach C is the best solution because minimising overhead on the radio interface is an important requirement for the solution. With respect to the in-vehicle network, it is supposed that bandwidth should support the overhead of all proposed solutions.

Modifications to legacy IPv6 nodes

As previously said, solutions B and C impose a non-standard forwarding behaviour (or at least a non-common one) on the GeoNet OBU which forwards the geographic IP traffic in function of the value of an extension header instead of using IPv6 destination address. This does not break IP because it only introduces changes to the GeoNet OBU forwarding process but is a debatable solution. Solution D entails a reduction of the freely usable multicast group-ID range. Therefore, for a standard accomplishment point of view solution A should be preferred.

At the AU the solutions impose the use of the Advanced Socket API for managing tunnels and extension headers. This is a reasonable requirement since applications that make use of geonetworking (GeoAware applications) are specifically designed for vehicular environments.

Flooding efficiency

Solutions B, C, and D use the same geographic area definition as the C2CNet layer so its flooding efficiency is the same and depends on how precise the definition of the geographic area is.

However, solution A limits the allowed geographic areas to a set of predefined squares of fixed length. Due to that, for a specific GeoDestination the application must use the square or squares that contain that area. This makes that C2CNet layer forwards the traffic to an area greater than optimal. Therefore its flooding efficiency depends on how close to the predefined squares the desired forwarding area is and logically, it is lower than the efficiencies of solutions B and C.

ANNEX B: Example of SQL structures for module “0A: Geo-destination”

The SQL tables for neighbour location awareness sharing in module “0A: Geo-destination” can be generated by the following SQL commands. Applications shall have the right to query the 'LocationTable' table through the MNG-UL SAP, but not to change its values.

```
-- Database: `GeoAreaManager`

CREATE TABLE IF NOT EXISTS `Configuration` (
  `Version` varchar(8) NOT NULL default '1.0' COMMENT 'Module version',
  `LocationTableEntryLifetime` int(10) unsigned NOT NULL default '20' COMMENT 'Lifetime of each entry in seconds',
  `LowerLayerReceivingPortNumber` int(11) NOT NULL,
  `LowerLayerSendingPortNumber` int(11) NOT NULL,
  `C2CNetReceivingPortNumber` int(11) NOT NULL,
  `C2CNetSendingPortNumber` int(11) NOT NULL,
  `OwnC2CNet ID` varchar(8) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `LocationTable` (
  `C2CNet ID` varchar(8) NOT NULL,
  `MAC` varchar(6) NOT NULL,
  `last_update_timestamp` datetime NOT NULL COMMENT 'the time when a position update was last received',
  `last_update_millisec` int(10) unsigned NOT NULL COMMENT 'millisecond of last update',
  `Latitude` double NOT NULL,
  `Longitude` double NOT NULL,
  `Altitude` float NOT NULL,
  `Vehicle_centered_EW_coordinate` float NOT NULL COMMENT 'Longitudinal coordinate in vehicle centered coordinate system, in units of meter',
  `Vehicle_centered_NS_coordinate` float NOT NULL COMMENT 'Latitudinal coordinate in vehicle centered coordinate system, in units of meter',
  `heading` float NOT NULL COMMENT 'counter-clockwise from North, measured in degrees',
  `speed` float NOT NULL COMMENT 'in units of m/s',
  `Tx_power` tinyint(4) NOT NULL COMMENT 'In dBm units',
  `link_attenuation` tinyint(4) NOT NULL COMMENT 'in dB units',
  PRIMARY KEY  (`C2CNet ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

ANNEX C: Service Metrics, Scalability, and Congestion Control

This chapter discusses possible geonetworked service metrics, scalability assurance, and wireless congestion control. It is not in the scope of GeoNet to specify an exact algorithm for wireless congestion control because there is no general consensus among stakeholders about the most suitable congestion control method. Therefore appropriate congestion control will be a future extension to this protocol. The requirements on future congestion control, and its expected qualities are presented here.

Discussion of service metrics

Service metrics can be measured in terms of the geonetworked communication's latency, scalability, and reliability. As the improvement of some metric generally causes a detriment of an other metric, supported geonetworked service assurance domain can be described by a parametric volume:

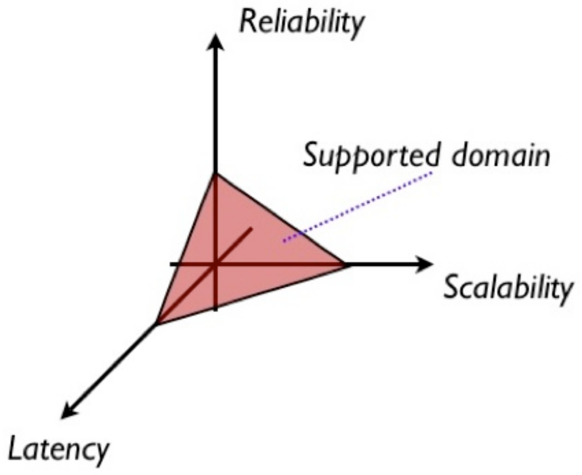


Figure 40: Parametric volume of a service assurance domain

Service metrics for Beaconing

Scalability: vehicles/m on a single road or vehicles/km^2 in urban area, at a given beaconing rate

Reliability: delivery probability percentage within target area

Latency: meters/sec information dissemination speed (e.g. 300 m / 0.6 s)

Service metrics for Event-driven safety broadcasts

Scalability: vehicles/m on a single road or vehicles/km^2 in urban area, at a given beaconing rate, without causing outage of more than N consecutive beacon deliveries to any vehicle

Reliability: delivery probability percentage within target area

Latency: meters/sec information dissemination speed (e.g. 300 m / 0.6 s)

Congestion control requirements

Following requirements are expected of congestion control functionality:

- Fair use of wireless resources
- Low-latency guarantee for emergency message delivery
- Resilience of system operation at a wide range of vehicle densities
- Resource utilization should converge to a stable, non-oscillating solution [Kovacs2008]
- The implementation of congestion control must consider priorities of information to be transmitted [Kovacs2008]

Qualities of congestion control

By the nature of vehicle-to-vehicle communication, congestion control algorithms will have following qualities:

- Distributed and decentralised mechanism
- Local approach, meaning it is based on measurements in a localised area around the vehicle
- Open loop approach, meaning that the congestion control algorithm in the vehicle does not get the feedback on result of its action. Instead it gets the feedback on cumulative action of congestion control in all surrounding vehicles.
- Multi-channel approach, meaning that congestion control algorithm must consider resource utilisation across available channels and media, and their effect on each other
- Both proactive and reactive methods can be used. Proactive methods work to prevent the building up of wireless congestion. Reactive methods react to an existing congestion scenario, with the aim of mitigating it.

Constraints and goals by basic scenarios

Periodic beaconing

Constraints:

- Beacons must be sent out at sufficient rate for safety applications (around 2 Hz for warning applications, and around 10 Hz for crash avoidance / mitigation)
- Beacon delivery probability must high enough - within radio reception range - to make multiple subsequent beacon losses very rare events
- Fixed control channel capacity constraint
- It is a priority to deliver beacons between vehicles with potential safety impact on one other but out of direct radio range because of some obstacle, for example a building corner in-between approaching vehicles

Goal functions:

- Maximize information flow delivered by beacons, measured as bits of decoded new information × m (single road) or bits of decoded new information × m^2 (urban area).

  This information flow is maximised by sustaining a proper rate of of control channel utilisation.
- Maximise probability of beacon reception within communication range. This probability is maximised by avoidance of hidden-terminal packet collisions.

Tools for implementation of goal functions:

- Use of optimal channel utilisation targeting power adjustment and channel rate adjustment mechanism
- Optimised scheduling of beacon transmissions (for example through the use of geo-mapped beacon scheduling)
- Optimisation of beacon forwarding, used only when needed for safety reasons

Event-driven GeoBroadcasts

Constraints:

- Safety information must be delivered at low latency within target area
- GeoBroadcast delivery probability must be very high after rebroadcasts - to make information delivery losses very rare events for any recipient in destination area
- Fixed control channel capacity constraint
- It is a priority to deliver beacons between vehicles with potential safety impact on one other but out of direct radio range because of some obstacle

Goal functions:

- Ensure an optimal rebroadcast density, measured as rebroadcasts / m (single road) or rebroadcasts / m^2 (urban area). This density should be achieved through a geographically even distribution of re-broadcasters for best delivery probability.

Tools for implementation of goal functions:

- Use of EMDV rebroadcast counting together with iterative-MHVB rebroadcast timing
- Use of short inter-frame spacing for priority control

Unicasts and Line-forwarding

Constraints:

- All vehicles must get fair share of channel capacity
- Fixed control/service channel capacity constraint with single media implementation. This capacity constraint is removed when directional media is available.
- Transmission of non-safety messages must not cause a significant change with delivery probability and latency of safety broadcasts

Goal functions:

- Ensure smallest co-channel interference with safety messaging
- Ensure highest end-to-end delivery probability (measured without rebroadcasts).

Tools for implementation of goal functions:

- Use of directional media priority and service channel power adjustment for minimising co-channel interference
- Use of intelligent routing for minimising the need for retransmission
- Use of transport layer scheduling (e.g. slow start) for control of fair-share access

Channel load monitoring

Following data are expected minimum available input for congestion control to assess channel load conditions:

- Network-layer statistics on received data rate per channel
- Number of neighbours in single-hop range
- The transmission power level assignment or received messages

Following data is possible additional input for congestion control:

- MAC-layer statistics on carrier sense channel utilisation rate per channel
- Furthest lateral/longitudinal distance to direct neighbours in the four directional quadrants (North-West, North-East, South-West, South-East), as described in [Kovacs2009].

ANNEX D: Contributors

The following experts have contributed to developing this specification:

- Andras Kovacs, BROADBIT
- Carlos J. Bernardos, IMDEA
- Carsten Schulze, LESSWIRE
- Hamid Menouar, HITACHI
- Ines Ben Jemaa, INRIA
- JinHyeock Choi, INRIA
- Jong-Hyouk Lee, INRIA
- Manabu Tsukada, INRIA
- Marco Gramaglia, IMDEA
- Maria Calderon, IMDEA
- Massimiliano Lenardi, HITACHI
- Thierry Ernst, INRIA
- Wenhui Zhang, NEC
- Wilfried Lohmann, LESSWIRE
- Yacine Khaled, INRIA

ANNEX E: References

[Brakemeier2008] Brakemeier, A.: “White Paper on Network Design Limits and VANET Performance”, Dec. 2008.

[C2CCC] Car-2-Car Communication Consortium’s “C2C-CC Manifesto”. Version 1.1. http://www.car-to-car.org. August 2007.

[Choi2008] J.H. Choi, Y. Khaled, M. Tsukada, T. Ernst “IPv6 support for VANET with geographical routing”, in 8th International Conference on Intelligent Transport System Telecommunications (ITST), Phuket, Thaïland, October 2008.

[Ernst2009] T. Ernst, V. Nebehaj, R. Sorasen “CVIS: CALM Proof of Concept Preliminary Results”, in 9th International Conference on Intelligent Transport System Telecommunications (ITST), Lille, France, October 2009.

[EtsiC2CDemo2009] ETSI TR 102 698 V1.1.1 - “Intelligent Transport Systems (ITS); Vehicular Communications; C2C-CC Demonstrator 2008; Use Cases and Technical Specifications”, June 2009.

[ETSI-ES-202-663] ETSI “Intelligent Transport Systems; European profile standard on the physical and medium access layer of 5 GHz ITS” ETSI ES 202 663 V<0.0.6> Draft for member approval, October 2009.

[ETSI-TR-102-698] ETSI. “Intelligent Transport Systems (ITS); Vehicular Communications; C2C-CC Demonstrator 2008; Use Cases and Technical Specifications”, ETSI TR 102 698 V1.1.1, June 2009.

[ETSI-TS-102-636-2] ETSI. “Intelligent Transportation Systems (ITS); Transport & Network: Vehicular Communications; GeoNetworking and Data Transport; Part 2: Scenarios for GeoNetworking”. ETSI, V0.3.1 Work in Progress, September 2009.

[ETSI-TS-102-636-3] ETSI. “Intelligent Transportation Systems (ITS); Vehicular Communications; GeoNetworking and Data Transport; Part 3: Network Architecture”. ETSI, V0.5.9 Work in Progress, September 2009.

[ETSI-TS-102-636-6-1] ETSI. “Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking and Data Transport; Part 6: Transmission of IPv6 Packets overs GeoNetworking Protocols”. ETSI, Work in Progress, December 2009.

[ETSI-TS-102-665] ETSI. “Intelligent Transport Systems (ITS); Vehicular Communications; Architecture”. ETSI, Work in Progress, December 2009.

[GeoNetD1.2] GeoNet D.1.2 deliverable - “Final GeoNet Architecture Design”. January 2010.

[GeoNetD4.1] GeoNet. “GeoNet Conformance Test Plan and Results”. GeoNet Deliverable D4.1, January 2010.

[GeoNetD5.1] GeoNet. “GeoNet Emulation Environment Results” GeoNet Deliverable D5.1, January 2010.

[GeoNetD7.1] GeoNet. “GeoNet Experimentation Results” GeoNet Deliverable D7.1, January 2010.

[Hain2008] Hain, T.: “An IPv6 Geographic Global Unicast Address Format”, Internet Draft (expired), 2008.

[Haversine] http://en.wikipedia.org/wiki/Haversine_formula

[ISO-21210] ISO TC204 WG16. “Intelligent Transport Systems – Communication for Land Mobiles (CALM) – IPv6 Networking”. ISO FDIS specification 21210, ISO, October 2009. Work in progress.

[ISO-21217] ISO TC204 WG16. “Intelligent Transport Systems – Communication for Land Mobiles (CALM) – Architecture”. ISO FDIS specification 21217, ISO, October 2009. Work in progress.

[Karp2000] Karp, B. and Kung, H.T., "Greedy Perimeter Stateless Routing for Wireless Networks," in Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, August, 2000, pp. 243-254.

[Khaled2009a] Y. Khaled, M. Tsukada, T. Ernst “Geographical information extension for IPv6: application to VANET”, in 9th International Conference on Intelligent Transport System Telecommunications (ITST), Lille, France, October 2009.

[Khaled2009b] Y. Khaled, I. Ben Jemaa, M. Tsukada and T. Ernst “Application of IPv6 multicast to VANET”, in 9th International Conference on Intelligent Transport System Telecommunications (ITST), Lille, France, October 2009.

[Kovacs2008] Kovacs, A.: “Resource Sharing Principles for Vehicular Communications”, IEEE Communications Society AutoNet conference, New Orleans, USA, December 2008.

[Kovacs2009] Kovacs, A.: “GeoMapped Timing of Beacons and Unicast Messages”, 9th International Conference on Intelligent Transport System Telecommunications (ITST), Lille, France, October 2009

[Menouar2007] H. Menouar, M. Lenardi, F. Filali, “Movement Prediction-based Routing (MOPR) Concept for Position-based Routing in Vehicular Networks“, WiVec 2007, 1st IEEE International Symposium on Wireless Vehicular Communications.

[Mariyasagayam07a] M. Mariyasagayam, M. Lenardi, “Broadcast Algorithms for Active Safety Applications over Vehicular Ad-hoc Networks”, WIT 2007, 4th International Workshop on Intelligent Transportation, Hamburg, Germany, March 2007.

[Mariyasagayam07b] M. Mariyasagayam, T. Osafune, M. Lenardi, “Enhanced Multi-Hop Vehicular Broadcast (MHVB) for Active Safety Applications“ , ITST 2007, 7thInternational Conference on ITS Telecommunications, Sophia Antipolis, France, June 2007.

[Osafune2006] Tatsuaki Osafune, Lan Lin, Massimiliano Lenardi : “Multi-Hop Vehicular Broadcast (MHVB)”, Proceedings of 6th International Conference on ITS Telecommunications, 2006.

[RFC1075] D. Waitzman, S. Deering “Distance Vector Multicast Routing Protocol” RFC 1075. Internet Engineering Task Force. November 1988.

[RFC2328] J. Moy “OSPF Version 2”. RFC 2328. Internet Engineering Task Force. April 1998

[RFC3306] B. Haberman and D. Thaler. Unicast-prefix-based IPv6 multicast addresses. RFC 3306, IETF, 2002.

[RFC3810]. R. Vida, L. Costa “Multicast Listener Discovery Version 2 (MLDv2) for IP”, RFC 3810, Internet Engineering Task Force.June 2004

[RFC3963] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, “Network Mobility (NEMO) Basic Support Protocol,” RFC 3963 (Proposed Standard), Jan. 2005.

[RFC4191] R. Draves, D. Thaler “Default Router Preferences and More-specific Routes” RFC 4191. Internet Engineering Task Force. November 2005

[RFC4291] Hinden, R. Et al: “IPv6 Addressing Architecture”, IETF RFC 4291, 2006.

[RFC4601] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas “Protocol Independent Multicast – Sparse Mode (PIM-SM): Protocol Specification (Revised)”. RFC 4601. Internet Engineering Task Force. August 2006

[RFC4605] B. Fenner, H. He, B. Haberman, H. Sandick “Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding (“IGMP/MLD Proxying”)”. RFC 4605. Internet Engineering Task Force. August 2006

[RFC4861] T. Narten, E. Nordmark, W. Simpson and H. Soliman, “Neighbour Discovery for IP version 6 (IPv6),“ RFC 4861 (Standards Track), Sep. 2007.

[RFC4862] S. Thomson, T. Narten, and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,” RFC 4862 (Draft Standard), Sep. 2007.

[RFC4885] T. Ernst “Network Mobility Support Terminology”, RFC 4885. Internet Engineering Task Force. July 2007

[RFC5648] R. Wakikawa, T. Ernst, K. Nagami, and V. Devarapalli. Multiple Care-of Addresses Registration, January 2008. IETF, draft-ietf-monami6-multiplecoa-05.

[Torrent-Moreno2007] Torrent-Moreno, M.: “Achieving Safety in a Distributed Wireless Environment”, Dissertation at Universität Karlsruhe, 2007. pp. 78-84