ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Ανάπτυξη και Υλοποίηση πρωτοκόλλου για δρομολόγηση ασύρματων σταθμών με βάση Γεωγραφικές Συντεταγμένες

Μεταπτυχιακή Εργασία

Μακρής Νικόλαος
Οκτώβριος 2013

Επιβλέπων Καθηγητής
Κοράκης Αθανάσιος, Λέκτορας

Μέλη Επιτροπής
Τασιούλας Λέανδρος, Καθηγητής
Κατσαρός Δημήτριος, Λέκτορας

DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF THESSALY

Design and Implementation of the GeoNetworking protocol for Geo-Location based Routing in Wireless Networks

Master Thesis

Makris Nikolaos
October 2013

Supervisor
Korakis Thanasis, Lecturer

Committee members
Tassiulas Leandros, Professor
Katsaros Dimitrios, Lecturer

This page is intentionally left blank

Ευχαριστίες

Με την εκπόνηση της παρούσας μεταπτυχιακής εργασίας, φέρνω εις πέρας τις μεταπτυχιακές μου σπουδές στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας.

Θα ήθελα αρχικά να ευχαριστήσω θερμά τους Λέκτορες του τμήματος ΗΜΜΥ κ. Κοράκη Αθανάσιο και κ. Κατσαρό Δημήτριο, και τον καθηγητή του τμήματος κ. Λέανδρο Τασιούλα, για τις χρήσιμες συμβουλές και υποδείξεις του καθώς και για την υποστήριξη που μου προσφέρανε κατά τη διάρκεια της φοίτησής μου, αλλά και κατά την εκπόνηση της διπλωματικής μου εργασίας. Επιπροσθέτως να τους ευχαριστήσω για την δυνατότητα που μου δίνουν να βρίσκομαι στην ομάδα του NITlab και να ασχολούμαι συνεχώς με νέες τεχνολογίες και να κάνω έρευνα στον χώρο τον ασύρματων δικτύων και υποδομών.

Από καρδιάς να ευχαριστήσω όλη την ομάδα του NITlab μέσα από την οποία μαθαίνω διαρκώς νέα πράγματα και τεχνολογίες και κάνω σημαντικές συνεργασίες. Να ευχαριστήσω επίσης όλη την ομάδα των παιδιών που οπουδήποτε κολλούσα σε κάποιο σημείο βοηθούσαν πολύ με το brainstorming που κάναμε.

Τέλος, ευχαριστώ θερμά την οικογένειά μου για την αμέριστη συμπαράσταση που μου παρείχε όλα αυτά τα χρόνια για την ολοκλήρωση των μεταπτυχιακών και προπτυχιακών σπουδών μου.

Περίληψη

Σε αυτή τη μεταπτυχιακή εργασία, ασχολούμαστε με την υλοποίηση του πρωτοκόλλου GeoNetworking, βασιζόμενοι στο standard ETSI TS 102 636-4-1. Το πρωτόκολλο αυτό είναι σχεδιασμένο να παρέχει δυνατότητες δρομολόγησης σε ασύρματα δίκτυα οχημάτων (VANETs) βασιζόμενο σε πληροφορίες διευθυνσιοδότησης που προέρχονται από κάποια συσκευή εντοπισμού θέσης (GPS). Το πρωτόκολλο αυτό τοποθετείται στο επίπεδο δικτύου της στοίβας πρωτοκόλλων OSI. Η υλοποίηση έγινε σε συνεργασία του Πανεπιστημίου Θεσσαλίας και της DELPHI, εταιρείας που δραστηριοποιείται στον χώρο και το αποτέλεσμα της συνεργασίας αυτής δεν αποτελεί ανοιχτό κώδικα. Για την αξιολόγηση του πρωτοκόλλου χρειάστηκαν να υλοποιηθούν κάποιοι βοηθητικοί δαίμονες (daemons) που παρέχουν λειτουργικότητα υψηλότερων επιπέδων (Facilities Layer) και επιπέδου διαχείρισης του πρωτοκόλλου (Management Layer). Η αξιολόγηση του πρωτοκόλλου έγινε στις πειραματικές υποδομές του Πανεπιστημίου Θεσσαλίας NITOS (Network Implementation Testbed using Open Source code) χρησιμοποιώντας και ασύρματες αλλά και ασύρματες τεχνικές δικτύωσης.

Abstract

In this Master Thesis we have been engaged in implementing the GeoNetworking protocol, based on the ETSI TS 102 636-4-1 standard. This protocol is designed to provide routing capabilities in Vehicular Adhoc Networks (VANETs), based on addressing information coming from a GPS device. This protocol is a Network layer protocol, in the OSI Networking stack. The implementation is a collaboration of University of Thessaly and Delphi, a company active on this field, and the results of this thesis are not Open Source code. For the evaluation of the protocol extra daemons had to be developed, emulating the operation of the higher layer (Facilities layer) and the management layer. The evaluation of the protocol took place in the NITOS (Network Implementation Testbed using Open Source code) testbed, using both wired and wireless networking connections.

Table of Contents

Contents

Table of Figures

Abbreviations

API	Application Programming Interface
ASL	Adaptation Sub Layer
AU	Application Unit
BRAN	Broadband Radio Access Networks
BTP	Basic Transport Protocol
CAM	Cooperative Awareness Messages
CBF	Contention Based Forwarding
CCU	Control and Communication Unit
CW	Contention Window
DENM	Decentralized Environment Notification Messages
DFS	Dynamic Frequency Selection
DSRC	Ddedicated Short Range Communications
ECC	European Control Conference
EDCA	Enhanced Distributed Channel Access
ERC	Energy Regulatory Commission
FIRE	Future Internet Research and Experimentation
FCS	Frame Check Summary
GF	Greedy Forwarding
GN	GeoNetworking
GN_ADDR	GeoNetworking Address
GPL	General Public Licence
HCCA	Hybrid Coordination Function Controlled Channel Access
HCF	Hybrid Coordination Function
HL	Hop Limit

HST	Header Subtype
HT	Header Type
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITS	Intelligent Transport Systems
LAN	Local Area Network
LL	Link Layer
LL_ADDR	Link Layer Address
LocT	Location Table
LocTE	Location Table Entry
LPV	Long Position Vector
LS	Location Service
LT	Lifetime
LTE	Long Term Evolution
MAC	Medium Access Control
MANET	Mobile Adhoc Networks
MFR	Most Forward within Radius
MGMT	Management
MIB	Management Information Base
MID	MAC ID
MLME	MAC sublayer Management Entity
NFP	Nearest with Forward Process
NH	Next Header
NITOS	Network Implementation Testbed using Open Source code
NMEA	National Marine Electronics Association

OBU	On Board Unit
OMF	cOntrol and Management Framework
OML	OMF Measurement Library
OSI	Open Systems Interconnection
PCF	Point Coordination Function
PDU	Protocol Data Unit
PL	Payload Length
PLME	PHY layer Management Entity
POS	Position
POTI	Position and Time
PV	Position Vector
QoS	Quality of Service
RHW	Road Hazard Warning
RLAN	Radio Local Area Network
RSU	Road Side Unit
SAP	Service Access Point
SDN	Software Defined Networking
SDR	Software Defined Radios
SDU	Service Data Unit
SE	Sender
SHB	Single Hop Broadcast
SN	Sequence Number
SO	Source
SPV	Short Position Vector
STA	Station
TPC	Transmission Power Control

TSB	Topologically Scoped Broadcast
TST	Timestamp
USRP	Universal Serial Radio Peripheral
UTH	University of Thessaly
V2I	Vehicle to Infrastructure Communication
V2V	Vehicle to Vehicle Communication
V2X	Vehicle to Vehicle/Infrastructure Communication
WLAN	Wireless LAN
WSN	Wireless Sensor Networks

1. Introduction

Ad hoc networks consist of mobile or stationary nodes that communicate over wireless links. There is neither fixed infrastructure to support the communication nor any centralized administration or standard support services. Nodes can self-organize dynamically in an arbitrary and temporary manner allowing people and devices to seamlessly communicate in areas with no pre-existing communication infrastructure; therefore, the nodes themselves act as routers as well. In addition, due to the limited transmission range of wireless nodes, intermediate nodes may be required to collaborate in forwarding a packet from source to destination. Therefore, nodes beyond direct wireless transmission range of each other will be able to communicate via multihop routing. The focus in this thesis is specifically on Mobile Ad Hoc Networks (MANETs) based on Intelligent Transport Systems (ITS). We implement a protocol based on geographical addressing, usually found in MANETs, based on the standard ETSI TS 102 636-4-1 [1], that give to a simple node full GeoNetworking protocol capabilities. The implementation has taken place in the NITOS testbed [2], in collaboration with Delphi[3] and the code produced in this project is not Open Source. The implementation is based on GPL v3 based libraries, and is owned by NITLab University of Thessaly.

The following sections are organized as follows: In Section 2, an overview of existing geographical addressing protocols is presented. Section 3 describes the generic ITS architecture. Sections 3-6 describe the different protocols that run on each layer of the GeoNetworking Stack. Section 7 describes implementation specific details, and Sections 8 and 9 present basic structures needed for creating the GN packets. Sections 10 and 11 present how the communication takes place with other ITS layers. In section 12 an overview of the tools needed for the implementation are presented. Section 13 contains the evaluation steps for ensuring proper execution of the protocol. Last but not least, section 14 describes the environment under which the protocol was evaluated, and the steps we took in order to enable GN capability to the NITOS testbed. Finally, section 15 describes some future issues that might need to be resolved.

2. Existing Geographical Addressing and Routing Protocols

Geographic routing (also called georouting or position-based routing) is a routing principle that relies on geographic position information. It is mainly proposed for wireless networks and based on the idea that the source sends a message to the geographic location of the destination instead of using the network address. The idea of using position information for routing was first proposed in the 1980s in the area of packet radio networks [4] and interconnection networks [5]. Geographic routing requires that each node can determine its own location and that the source is aware of the location of the destination. With this information a message can be routed to the destination without knowledge of the network topology or a prior route discovery.

There are various approaches, such as single-path, multi-path and flooding-based strategies [6]. Most single-path strategies rely on two techniques: greedy forwarding and face routing. Greedy forwarding tries to bring the message closer to the destination in each step using only local information. Thus, each node forwards the message to the neighbor that is most suitable from a local point of view. The most suitable neighbor can be the one who minimizes the distance to the destination in each step (Greedy). Alternatively, one can consider another notion of progress, namely the projected distance on the source-destination-line (MFR, NFP), or the minimum angle between neighbor and destination (Compass Routing). Not all of these strategies are loop-free, i.e. a message can circulate among nodes in a certain constellation. It is known that the basic greedy strategy and MFR are loop free, while NFP and Compass Routing are not [7].

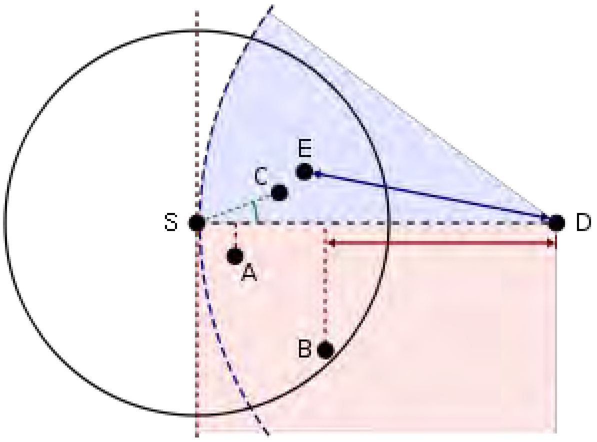


Figure 1: Greedy forwarding variants: The source node (S) has different choices to find a relay node for further forwarding a message to the destination (D). A = Nearest with Forwarding Progress (NFP), B = Most Forwarding progress within Radius (MFR), C = Compass Routing, E = Greedy

There are several greedy routing strategies. They can be defined in terms of progress, distance and direction towards the destination. The progress is the distance between a node S and the projection A' of a neighbour node A onto the line connecting S and final destination D (see Figure 1). The larger this distance, the more progress the corresponding neighbour can make. For instance, the Most Forward within Radius (MFR) [4] scheme is based on this progress notion.

In MFR, the packet destined to destination D is forwarded to the next neighbour who maximizes the progress towards D (e.g., node A in Figure 1). This scheme minimizes the number of hops to reach D.

Under this category, there is another scheme called Nearest with Forward Progress (NFP) [8], which forwards the packet to the nearest neighbour of the sender that is closer to the destination (node C in Figure 1). It is shown that if all nodes employ NFP, the probability of packet collision is reduced significantly [9]. Therefore, this strategy performs better than MFR. Another greedy strategy, which is widely used, applies the same principle, but uses the notion of distance, and more accurately, the Euclidean distance. That is, an intermediate node forwards the packet to the neighbour with least distance d to the destination, who is closer to D than S (e.g., node B in Figure 1). Direction-based schemes, also called compass routing [10], use the deviation as a criterion. The deviation is defined as the angle between two lines: the line connecting the current node and the next hop, and the line connecting the source and the destination. The deviation is used to select the neighbour closest in the direction to destination D (e.g., node C in Figure 1.). This scheme aims at minimizing the spatial distance a packet travels.

3. Intelligent Transport Systems Architecture

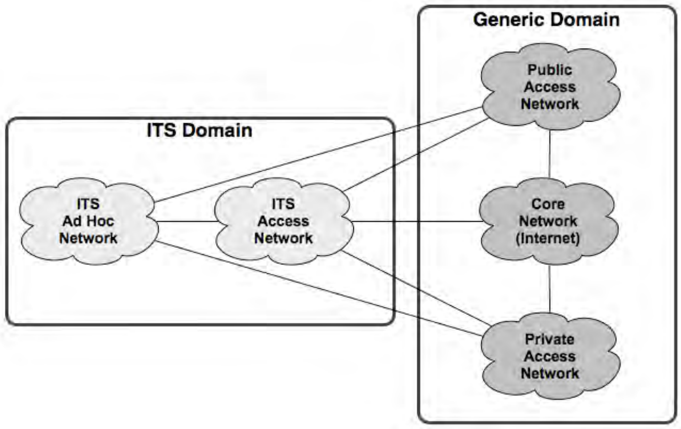


Figure 2: ITS Network Architecture

Figure 2 represents the highest level of abstraction of the ITS network architecture, where the external networks, represented by clouds are connected. The networks can be categorized into an ITS domain and a generic domain as specified in [11]. The external networks can be described as follows:

The ITS ad hoc network enables ad hoc communication among vehicle, roadside and personal ITS stations. The communication is based on wireless technologies, that typically provide a limited communication range referred to as 'short-range wireless technology') and allow for mobility of the ITS stations forming arbitrary network topologies without the need for a coordinating communication infrastructure. An example of an ITS ad hoc network is a network of vehicle, roadside and personal ITS stations interconnected by ITS-G5 [12] wireless technology. Generally, an access network enables ITS stations to access networks.

An ITS access network is a dedicated network that provides access to specific ITS services and applications and can be operated by a road operator or other operators. The ITS access network also interconnects roadside ITS stations and provides communication in between these as well as among vehicle ITS stations via the roadside ITS stations that are interconnected in the ITS access network. This local network can then enable the vehicle ITS stations to communicate via a roadside infrastructure communication network instead directly in ad hoc mode. As an example, an ITS access network can connect roadside ITS stations along a highway with a central ITS station (e.g. a road traffic management centre). In the case that short-range wireless technology is used for communication via roadside ITS stations, the connectivity to the ITS access network is typically provided intermittently.

A public access network provides access to general purpose networks that are publicly accessible. An example is an IMT-2000 [13] network that connects vehicle ITS stations to the Internet and provides mobile Internet access. A private access network, in contrast to a public access network, provides data services to a closed user group for a secured access to another network. For example, a private access network can connect vehicle ITS stations to a company's intranet.

The access networks and the core network provide access to various services:

- legacy services , such as WWW, email and many others;
- ITS services provided by road traffic management centres and backend services;
- ITS operational support services required to operate the ITS, such as security services.

Core component of the architecture is the ITS station, which has two main roles: In its first role, the ITS station is a network node and acts as a communication source or sink. Likewise an ITS station can be a forwarder of data, e.g. in the ITS ad hoc network. In its second role, the ITS station is placed at the network edge and connect the different networks via an ITS station internal network (see Figure 3).

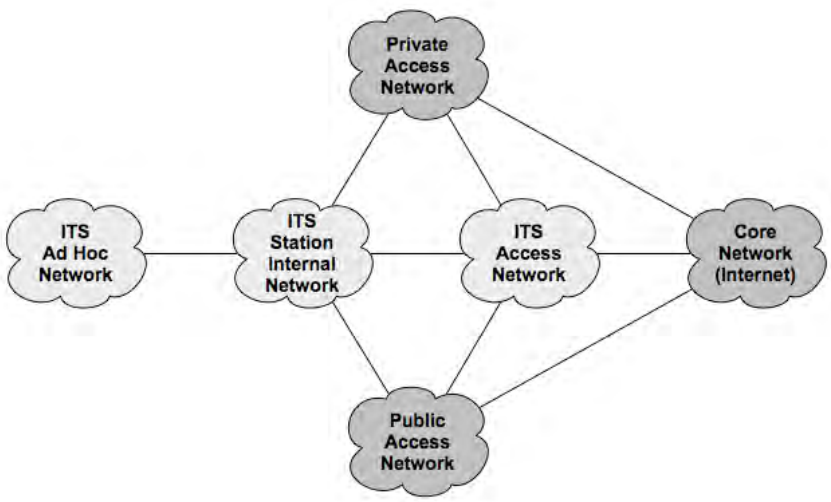


Figure 3 : ITS Network architecture (general)

ITS stations shall be able to communicate via at least one of the following means (see Figure 3):

1. via an ITS ad hoc network;
2. via an ITS access network;
3. via a public access network;
4. via a private access network;
5. via one of the access networks into the core network (e.g. the Internet).

In addition to the networks listed above, an ITS station can also be attached to proprietary local networks of e.g. vehicle ITS sub-systems and roadside ITS sub-system as presented in [14]. Typical examples are:

a) Controller Area Network (CAN) in a vehicle ITS sub-system.
b) Legacy roadside infrastructure in a roadside ITS sub-system.

According to ETSI TS 102 636-3 [15] ITS standard, an ITS (Intelligent Transport System) Station is comprised out of two units:

- The Application Unit (AU)
- The Computing and Communication Unit (CCU)

AU runs a single or a set of applications and utilizes the CCU's communication capabilities. In a possible implementation, the CCU executes the ITS access technology, ITS network & transport, and the ITS facilities layers, whereas the ITS application layer resides in the AU. The distinction between AU and OBU is logical; all layers can also be implemented in a single physical unit.

The CCU shall be equipped with at least a single ITS external communication interface to provide connectivity to the ITS ad hoc network or the different access networks (ITS access network, public access network, private access network). The CCU and the AU can be equipped with one or multiple ITS internal communication interfaces.

Moreover, an AU can have an external communication interface for access to the proprietary local network. The ITS internal communication interface shall connect AUs with CCUs, AUs with other AUs, and CCUs with other CCUs via the ITS station-internal network. AUs and CCUs can form a mobile network [16], where the AUs obtain connectivity to the networks via the external communication interface of the CCU. AU and CCU can reside in a single physical unit.

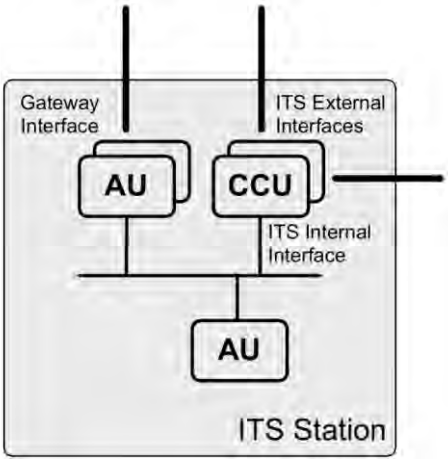


Figure 4 : ITS Station Architecture: Application Unit and Control and Communication Units in a single On Board Unit

Due to the different communication needs that a vehicular environment has, an ITS Station protocol stack has been proposed, based on the existing OSI protocol stack.

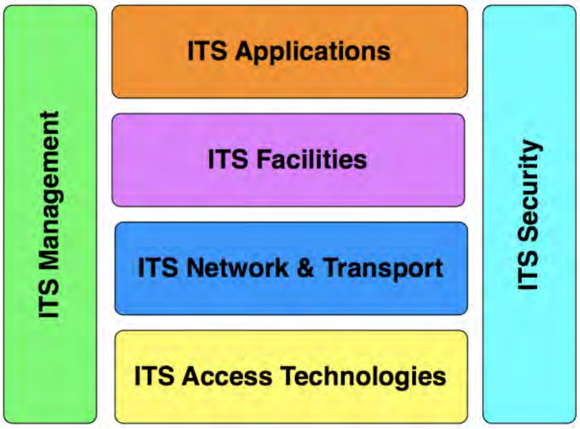


Figure 5 : General ITS Station protocol stack

ITS access technologies layer covers various communication media and related protocols for the physical and data link layers. The access technologies are not restricted to specific type of media, though most of the access technologies are based on wireless communication. The access technologies are used for communication inside of an ITS station (among its internal components) and for external communication (for example with other ITS stations). For external communication, some of the ITS access technologies represent complete, non-ITS specific communication systems (such as GPRS, UMTS, WiMAX) that are regarded as 'logical links over which ITS data is transparently transported.

The **ITS network & transport layer** comprises protocols for data delivery among ITS stations and from ITS stations to other network nodes, such as network nodes in the core network (e.g. the Internet). ITS network protocols particularly include the routing of data from source to destination through intermediate nodes and the efficient dissemination of data in geographical areas. ITS transport protocols provide the end-to-end delivery of data and, depending on requirements of ITS facilities and applications, additional services, such as reliable data transfer, flow control and congestion avoidance. A particular protocol in the ITS network & transport layer is the Internet protocol IP version 6 (IPv6). The usage of IPv6 includes the transmission of IPv6 packets over ITS network protocols, dynamic selection of ITS access technologies and handover between them, as well as interoperability issues of IPv6 and IPv4.

The **ITS facilities layer** provides a collection of functions to support ITS applications. The facilities provide data structures to store, aggregate and maintain data of different type and source (such as from vehicle sensors and from data received by means of communication). As for communication, ITS facilities enable various types of addressing to applications, provide ITS-specific message handling and support establishment and maintenance of communication sessions. An important facility is the management of services, including discovery and download of services as software modules and their management in the ITS station.

The **ITS applications layer** refers to ITS applications and use cases for road safety, traffic efficiency, infotainment and business.

4. Facilities Layer protocols

The protocols running on the facilities layer of an ITS station are based on the exchange of two types of messages:

- Cooperative Awareness Messages (CAM)
- Decentralized Environment Notification Messages (DENM)

An overview of where these messages are applicable follows.

4.1 Cooperative Awareness Messages (CAM)

The Cooperative Awareness Messages (CAMs) are distributed within the ITS-G5 (802.11p) network and provide information of presence, positions as well as basic status of communicating ITS stations to neighbouring ITS stations that are located within a single hop distance. All ITS stations shall be able to generate, send and receive CAMs, as long as they participate in V2X networks. By receiving CAMs, the ITS station is aware of other stations in its neighbourhood area as well as their positions, movement, basic attributes and basic sensor information. At receiver side, reasonable efforts can be taken to evaluate the relevance of the messages and the information. This allows ITS stations to get information about its situation and act accordingly.

Information distributed by CAM Management is commonly used by related use cases and therefore the CAM Management is a mandatory facility. The Approaching Emergency Vehicle and Slow Vehicle Warning are just two use cases which benefit from CAM.

4.2 Decentralized Environment Notification Messages (DENM)

Decentralized Environmental Notification Messages (DENMs) are mainly used by the Cooperative Road Hazard Warning (RHW) application in order to alert road users of the detected events. The RHW application is an event-based application composed of multiple use cases. The general processing procedure of a RHW use case is as follows:

- Upon detection of an event that corresponds to a RHW use case, the ITS station immediately broadcasts a DENM to other ITS stations located inside a geographical area and which are concerned by the event.
- The transmission of a DENM is repeated with a certain frequency.
- This DENM broadcasting persists as long as the event is present.
- The termination of the DENM broadcasting is either automatically achieved once the event disappears after a predefined expiry time, or by an ITS station that generates a special DENM to inform that the event has disappeared.

- ITS stations, which receive the DENMs, process the information and decide to present appropriate warnings or information to users, as long as the information in the DENM is relevant for the ITS station.

5. Network and Transport Layer protocols

The ITS network & transport layer comprises several network and transport protocols (Figure 6). In detail an ITS station can execute the following protocols at the ITS network & transport layer:

- GeoNetworking protocol. For usage of the GeoNetworking over different ITS access technologies, the specification of the protocol is split into a media-independent part and a media-dependent part (potentially multiple parts), such as for ITS-G5 [12].
- Transport protocols over GeoNetworking, such as the Basic Transport Protocol and other GeoNetworking transport protocols as they may be defined later.
- Internet protocol IP version 6 [17] with IP mobility support [18] and optionally support for network mobility
- (NEMO) [19] or other approaches depending on the deployment scenario.
- Internet protocol IP version 4 for transition to IPv6 [20].
- User Datagram Protocol UDP [17].
- Transmission Control Protocols TCP [18].
- Other network protocols.
- Other transport protocols, such as SCTP.

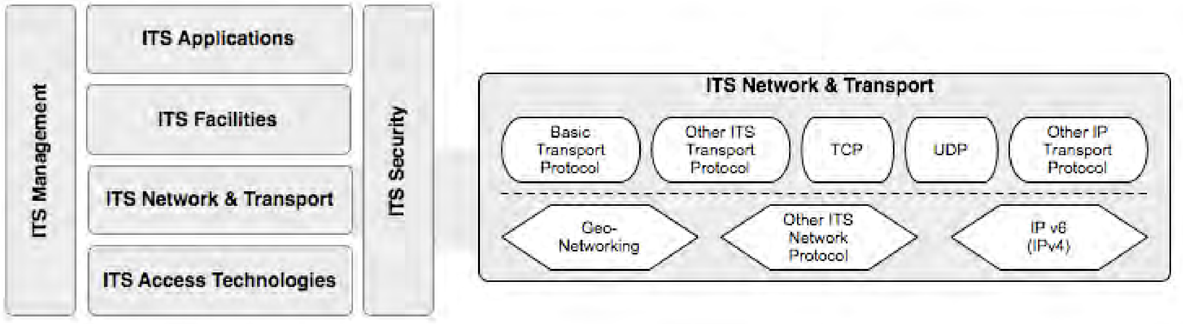


Figure 6 : ITS Network and Transport Layer protocols

5.1 Assembly of network and transport protocols in the ITS station protocol stack

For protocol stacks involving the GeoNetworking protocol and IPv6, protocols shall be assembled in one of the following ways described.

For the GeoNetworking protocol, the underlying ITS access technologies are limited to short-range wireless technologies, such as ITS-G5.

5.2 GeoNetworking protocol stack

The GeoNetworking protocol stack may be assembled with the GeoNetworking protocol and ITS-specific transport protocols as envisaged in TS 102 636-5 [21] at the top of the GeoNetwork protocol as depicted in Figure 7.

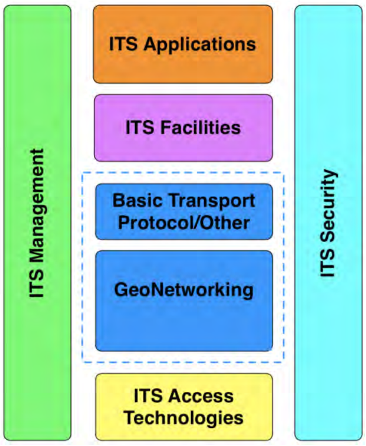


Figure 7 : The GeoNetworking protocol stack

5.3 ITS IPv6 stack

The IPv6 stack may be assembled with the IPv6 protocol and related transport protocols UDP [22], TCP [23] and others as depicted in the following figure.

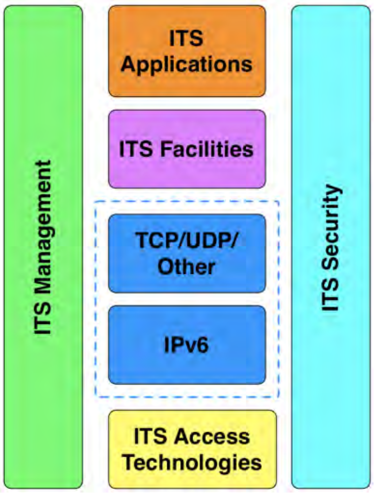


Figure 8 : ITS IPv6 protocol stack

5.4 Combination of the GeoNetworking protocol and IPv6

This protocol stack combines the stacks in the previous clauses. In this protocol stack (Figure 8), IP shall run at the top of the GeoNetworking protocol or directly at the top of the ITS access technologies.

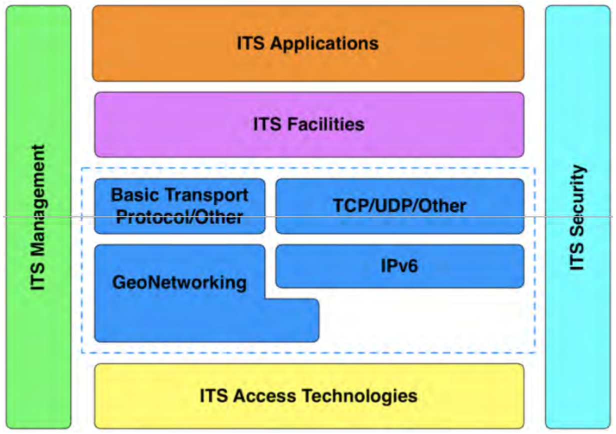


Figure 9 : ITS GeoNetworking with IPv6 support protocol stack

5.5 Protocol stacks for other network protocols

Further protocol stacks can be defined for other network protocols. In order to meet application and system requirements, the usage of other network protocols in parallel to the protocol stacks defined in the previous clauses can be restricted. For example, other network protocols must not be used at the top of ITS-G5 operating in the ITS-G5A frequency band.

6. Access Layer protocols

Figure 10 shows the part of the ITS AL that is covered by the present document. It is based on the OSI layered communications model with a detailed view of the ITS Access Technology layer. The mapping between the ITS-G5 elements specified in the present document and the ITS AL model is shown in figure 10.

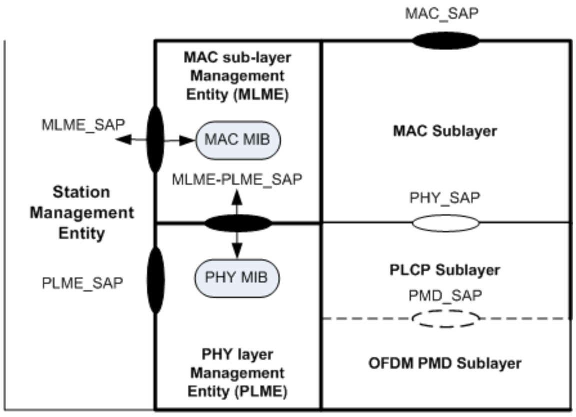


Figure 10 : ITS-G5 Elements

The following elements of ITS-G5 are defined:

- a physical layer,
- a medium access control sub-layer,
- a MAC sub-layer Management Entity (MLME)
- and a Physical Layer Management entity (PLME).

ITS-G5 also includes the SAPs MAC_SAP and MLME_SAP (depicted in black in figure 10). The internal SAPs PHY_SAP and PMD_SAP (depicted in white in figure 10) are not part of ITS-G5 specification, which implies that an ITS-G5 STA may not implement these SAPs. However, a STA implementing these SAPs in compliance with 802.11 [24] is also considered compliant to ITS-G5.

In figure 10 the ITS-G5 physical layer is composed of the two sub-layers PLCP and PMD. The distinction of the two sub-layers is only presented for homogeneity with [25]. In fact, ITS-G5 only supports the OFDM PHY specification. Consequently, ITS-G5 STAs may not have the two sub-layers PLCP and PMD and, instead, may have one single physical layer.

As compared to [25], ITS-G5 does not provide the SME-PLME_SAP. Consequently, only the MLME can access the PHY MIB via the MLME-PLME_SAP. The MLME-PLME_SAP is therefore part of ITS-G5.

6.1 Physical Layer Requirements

6.1.1 Frequency Allocation

Table 1 shows the frequency ranges and related regulatory requirements and harmonized standards.

Table 1 : Frequencies and applications on ITS

Frequency Range	Usage	Regulation	Harmonized Standard
5.905-5.925 GHz	Future ITS applications	ECC Decision [28]	ETSI EN 302 571 [26]
5.875-5.905 GHz	ITS Road Safety	ECC Decision[28], Commission Decision [29]	
5.855-5.875 GHz	ITS non-safety applications	ECC Recommendation [30]	
5.475-5.725 GHz	RLAN (BRAN, WLAN)	ERC Decision [31], Commission Decisions [32] [33]	ETSI EN 301 893 [27]

Figure 11 illustrates the requirements for spectral power density in the 5 GHz range for the frequency bands listed in table 1. It also shows the European bands for Dedicated Short Range Communication (DSRC).

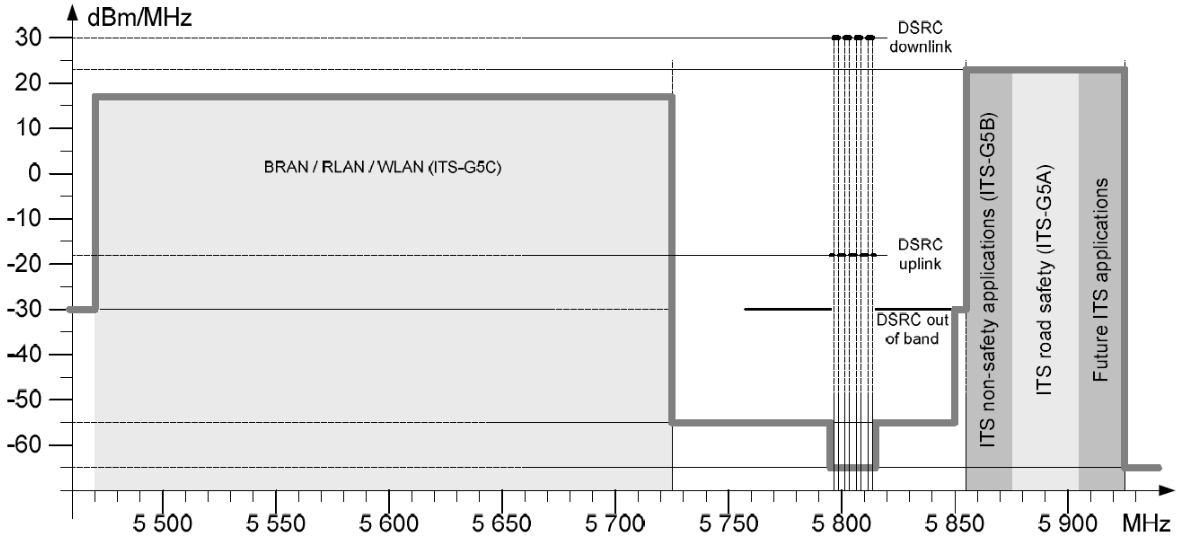
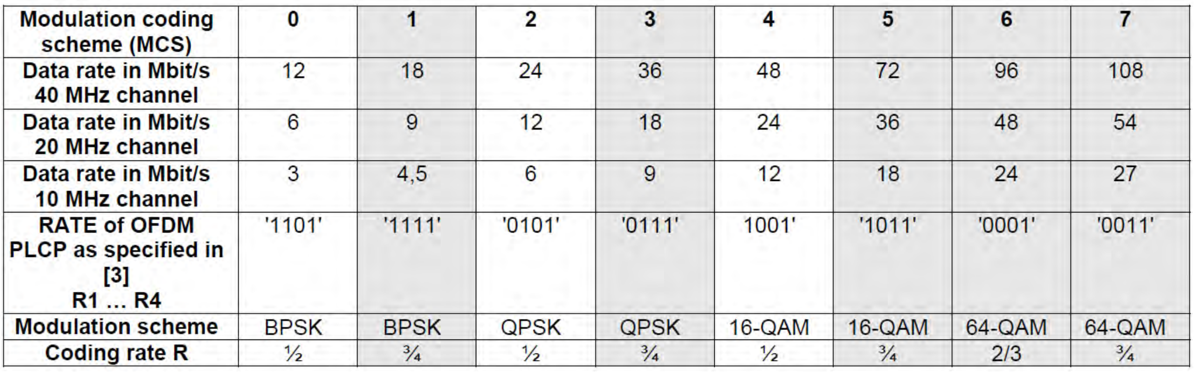


Figure 11: Requirements for spectral power density in the 5 GHz range for ITS stations



Modulation coding scheme (MCS)	0	1	2	3	4	5	6	7
Data rate in Mbit/s 40 MHz channel	12	18	24	36	48	72	96	108
Data rate in Mbit/s 20 MHz channel	6	9	12	18	24	36	48	54
Data rate in Mbit/s 10 MHz channel	3	4,5	6	9	12	18	24	27
RATE of OFDM PLCP as specified in [3] R1 … R4	'1101'	'1111'	'0101'	'0111'	1001'	'1011'	'0001'	'0011'
Modulation scheme	BPSK	BPSK	QPSK	QPSK	16-QAM	16-QAM	64-QAM	64-QAM
Coding rate R	½	¾	½	¾	½	¾	2/3	¾

Figure 12 : Modulation scheme and data rates for ITS G5 applications

6.1.2 Transmit Power Control (TPC)

TPC limits shall be as specified in [26] for ITS-G5A and ITS-G5B, and as specified in [27] for ITS-G5C. Additional TPC requirements are provided by the DCC.

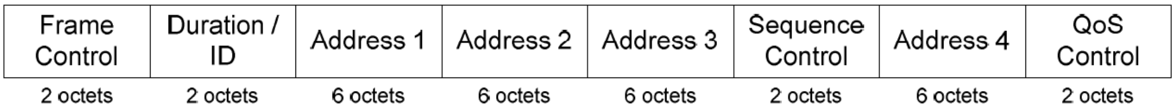
These additional requirements for ITS-G5A and ITS-G5B are encompassed in [34].

6.2 Medium Access Control Sub-layer

The medium access control sub-layer of the ITS-G5 shall be compliant with the profile of IEEE 802.11 [25].
A MAC frame shall consist of the following basic components:

- a MAC header;
- a frame body of variable length;
- and a frame check sequence (FCS).

Figure 13 shows the MAC header as specified in [25]:



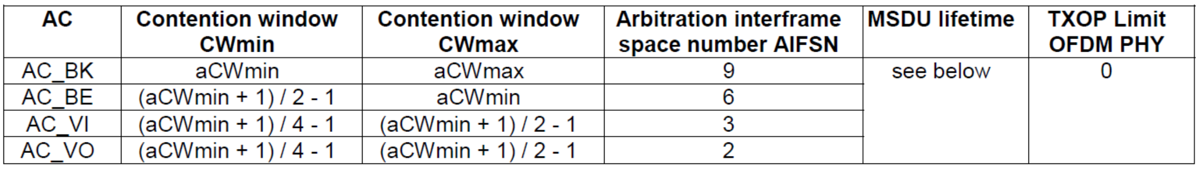
Frame Control	Duration / ID	Address 1	Address 2	Address 3	Sequence Control	Address 4	QoS Control
2 octets	2 octets	6 octets	6 octets	6 octets	2 octets	6 octets	2 octets

Figure 13 : MAC header of IEEE 802.11

6.2.1 Quality of Service

Since ITS-G5 STAs always operate outside the context of a BSS, mechanisms such as Point Coordination Function (PCF) and hybrid coordination function (HCF) controlled channel access (HCCA) are not applicable.

Enhanced Distributed Channel Access (EDCA) shall be applied. The set of EDCA parameters as specified in figure 14 must not be negotiated over the air link, but shall be statically set.



AC	Contention window CWmin	Contention window CWmax	Arbitration interframe space number AIFSN	MSDU lifetime	TXOP Limit OFDM PHY
AC_BK	aCWmin	aCWmax	9	see below	0
AC_BE	(aCWmin + 1) / 2 - 1	aCWmin	6		
AC_VI	(aCWmin + 1) / 4 - 1	(aCWmin + 1) / 2 - 1	3		
AC_VO	(aCWmin + 1) / 4 - 1	(aCWmin + 1) / 2 - 1	2		

Figure 14 : EDCA parameters for Contention Window in IEEE 802.11

6.2.2 Dynamic Frequency Selection (DFS)

Dynamic frequency selection (DFS) is required for communications in the RLAN band [27] and [33]. Only a DFS master/slave mode of operation shall be supported. A fixed ITS-G5 STA operating as a service provider shall provide the DFS master functionality. Mobile ITS-G5 STAs operating as a service client shall provide the DFS slave function functionality.

The frequency band inside ITS-G5C actually to be used as the G5SC shall be announced in the service advertisement frame broadcasted in the G5CC.

7. GeoNetworking Implementation

In this thesis, we start from scratch and build an implementation of the ETSI TS 102 636-4-1 standard, where the GeoNetworking protocol is defined. Along with GeoNetworking, Basic Transport protocol A and B (BTP-A, BTP-B) are implemented as a transport layer protocol.

The goals for this implementation is the ability to interact with a higher facility layer, able to generate either CAM or DENM messages, that then are encapsulated in proper GeoNetworking packets and passed to a lower layer through a defined Application Interface (API).

This implementation is developed as a user space daemon process, which is able to communicate with already existing applications developed by Delphi Co., simulating a facilities layer or a lower layer. Initially, for testing purposes only the applications of Delphi were used, but then we moved to new lightweight implementations of the facilities and management layer, able to allow for proper protocol operation.

In the following section, the protocol operation is described, according to [1], how addressing is defined and the type of messages supported.

7.1 GeoNetworking Addressing

Every GeoAdhoc router shall have a unique GeoNetworking address. This address shall be used in the header of a GeoNetworking packet and identify the communicating GeoNetworking entities. The format of the GeoNetworking address is described in the following figure:

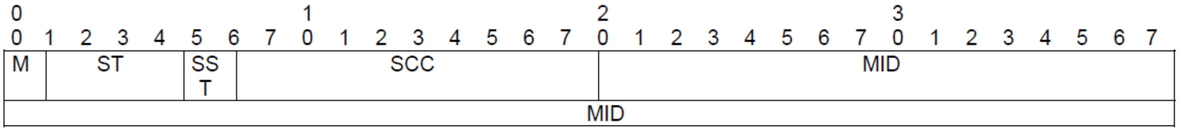
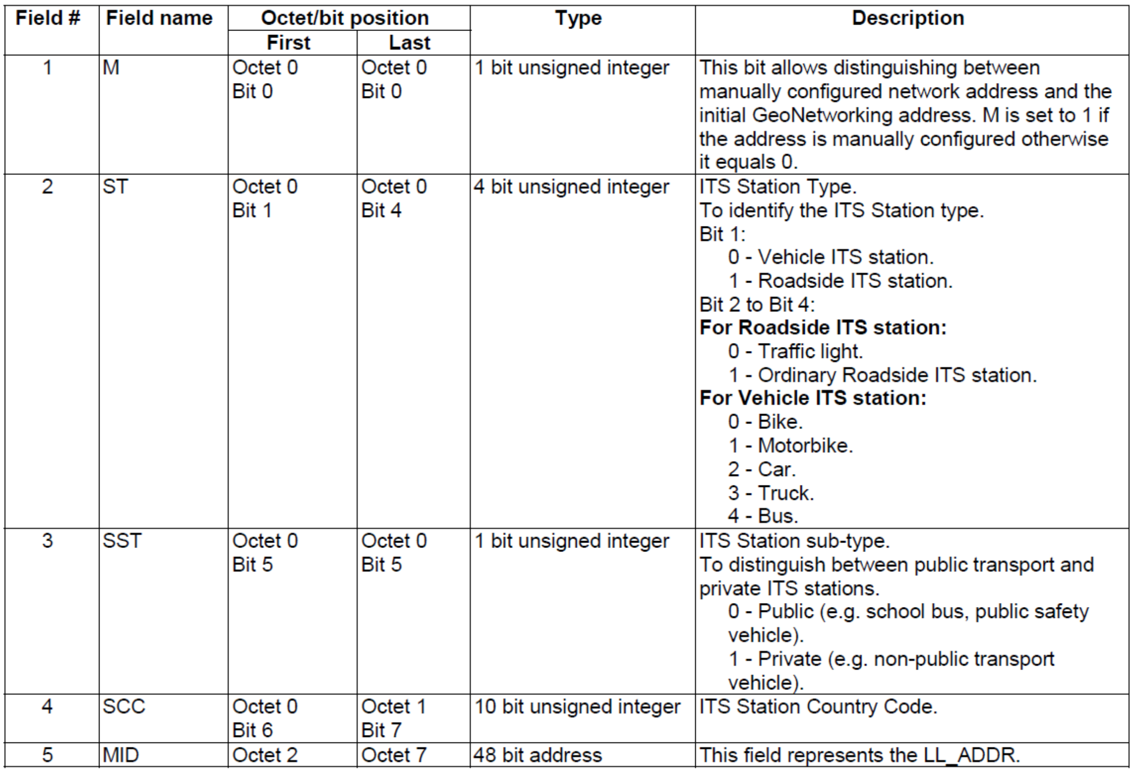


Figure 15 : GeoNetworking Address format

The fields of the GeoNetworking address shall be configured according to the following table:



Field #	Field name	Octet/bit position		Type	Description
		First	Last		
1	M	Octet 0 Bit 0	Octet 0 Bit 0	1 bit unsigned integer	This bit allows distinguishing between manually configured network address and the initial GeoNetworking address. M is set to 1 if the address is manually configured otherwise it equals 0.
2	ST	Octet 0 Bit 1	Octet 0 Bit 4	4 bit unsigned integer	ITS Station Type. To identify the ITS Station type. Bit 1: 0 - Vehicle ITS station. 1 - Roadside ITS station. Bit 2 to Bit 4: **For Roadside ITS station:** 0 - Traffic light. 1 - Ordinary Roadside ITS station. **For Vehicle ITS station:** 0 - Bike. 1 - Motorbike. 2 - Car. 3 - Truck. 4 - Bus.
3	SST	Octet 0 Bit 5	Octet 0 Bit 5	1 bit unsigned integer	ITS Station sub-type. To distinguish between public transport and private ITS stations. 0 - Public (e.g. school bus, public safety vehicle). 1 - Private (e.g. non-public transport vehicle).
4	SCC	Octet 0 Bit 6	Octet 1 Bit 7	10 bit unsigned integer	ITS Station Country Code.
5	MID	Octet 2	Octet 7	48 bit address	This field represents the LL_ADDR.

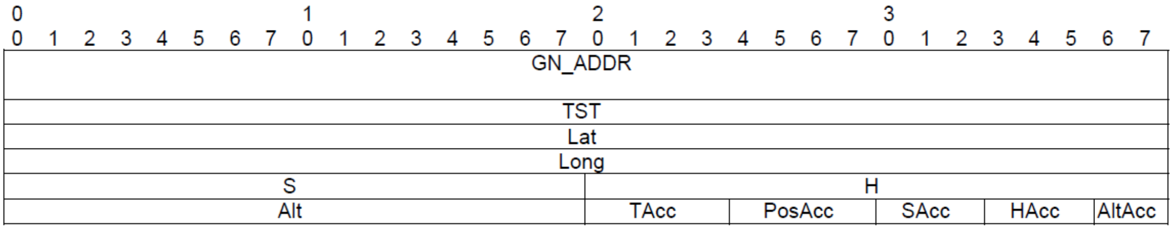
Figure 16 : Configuration parameters of the GN address

The first bit is reserved for the recognition of manual configured GeoNetworking addresses. The allocation of ITS Station Country Codes (SCC) is administered by the ITU-T.

The MID field corresponds to the access layer address. In case of IEEE 802.11p MAC layer, the 48-bit MAC layer address shall be used.

7.2 GeoNetworking Long Position Vectors (LPVs)

The LPVs are used complementary to the GeoNetworking address, and are used in the messages sent out of a single GeoNetworking station. They contain geographical information of a GeoNetworking Station. Their format is described in the following picture:



<table>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="32">GN_ADDR</td></tr>
<tr><td colspan="32">TST</td></tr>
<tr><td colspan="32">Lat</td></tr>
<tr><td colspan="32">Long</td></tr>
<tr><td colspan="16">S</td><td colspan="16">H</td></tr>
<tr><td colspan="16">Alt</td><td colspan="4">TAcc</td><td colspan="4">PosAcc</td><td colspan="3">SAcc</td><td colspan="3">HAcc</td><td colspan="2">AltAcc</td></tr>
</table>

Figure 17 : LPV format

Field #	**Field name**	**Octet/bit position**		**Type**	**Unit**	**Description**
		First	**Last**			
1	*GN_ADDR*	Octet 0	Octet 7	64 bit address	n/a	The network address for the GeoAdhoc router entity in the ITS station.
2	*TST*	Octet 8	Octet 11	32 bit unsigned integer	[ms]	Expresses the time in milliseconds at which the latitude and longitude of the ITS station were acquired by the GeoAdhoc router. The time is encoded as: $TST = TST(UET) \bmod 2^{32}$ where *TST(UET)* is the number of milliseconds since the Unix Epoch 1970-01-01T00:00.
3	*Lat*	Octet 12	Octet 15	32 bit signed integer	[1/10 micro-degree]	WGS-84 latitude of the GeoAdhoc router expressed in 1/10 micro degree.
4	*Long*	Octet 16	Octet 19	32 bit signed integer	[1/10 micro-degree]	WGS84 longitude of the GeoAdhoc router expressed in 1/10 micro degree.
5	*S*	Octet 20	Octet 21	16 bit signed integer	[1/100 m/s]	Speed of the GeoAdhoc router expressed in signed units of 0,01 meters per second.
6	*H*	Octet 22	Octet 23	16 bit unsigned integer	[1/10 degrees]	Heading of the GeoAdhoc router from which the Network Header originates, expressed in unsigned units of 0,1 degrees from North
7	*Alt*	Octet 24	Octet 25	16 bit signed integer	[m]	Altitude of the GeoAdhoc router expressed in signed units of 1 meter.
8	*TAcc*	Octet 26 Bit 0	Octet 26 Bit 3	4 bit unsigned integer	n/a	Accuracy indicator for the value expressed in the field TST. The accuracy interval of the 95 % accuracy level for the value of the TST field defined as: $Accuracy_{95\%} = Valueof\,(LSbit(TST) \cdot 2^{TAcc} = 1ms \cdot 2^{TAcc}$ The minimum and maximum value of the time accuracy interval is then: $1\,ms \times 2^{0 \ldots 15} = 1 \ldots 32\,768\,ms$.

9	*PosAcc*	Octet 26 Bit 4	Octet 26 Bit 7	4 bit unsigned integer	n/a	Encoded accuracy indicator for the value of the position *POS*. The accuracy interval of the 95 % accuracy level for POS is defined as: $Accuracy_{95\%} = Area(POS, R)$ with $POS = Lat, Long$ and $R = LSbit(Long) \cdot 2^{PosAcc}$ *PosAcc* = 0000 is reserved to indicate that no specific accuracy interval is specified. The maximum error the 95 % accuracy interval can cover, represented by *PosAcc* = 1111, is 1/10 micro degree * 2^{15} equivalent to about 365 m. (see notes 1 to 3)
10	*SAcc*	Octet 27 Bit 0	Octet 27 Bit 2	3 bit unsigned integer	n/a	Encoded accuracy indicator for the value expressed in the field Speed *S*. The accuracy interval of the 95 % accuracy level for the currently reported value is defined as: $Accuracy_{95\%} = Valueof(LSbit(S)) \cdot 2^{SAcc} = 0.01m/s \cdot 2^{SAcc}$ SAcc = 00 is reserved to indicate that the accuracy interval is not specified. The maximum error the 95 % accuracy interval can cover, represented by *SAcc* = 111, is 0,01 m/s * 2^7 equivalent to 0,01 m/s * 128 = 1,28 m/s.
11	*HAcc*	Octet 27 Bit 3	Octet 27 Bit 5	3 bit unsigned integer	n/a	Encoded accuracy indicator for the value of the field Heading *H*. The interval of the 95 % accuracy level for the currently Heading value H is defined as: $Accuracy_{95\%} = Valueof(LSbit(H)) \cdot 2^{HAcc} = 0.005493247 \deg \cdot 2^{HAcc}$ *HAcc* = 000 is reserved to indicate that no specific accuracy interval is specified. The maximum error the 95 % accuracy interval can cover, represented by the *HAcc* = 111, is 0,005493247 deg * 2^7 equivalent to about 0,005493247 deg * 128 ≈ 0,7 deg.
12	*AltAcc*	Octet 27 Bit 6	Octet 27 Bit 7	2 bit unsigned integer	n/a	Encoded accuracy indicator for the value in the field Altitude (*Alt*). The accuracy interval of the 95 % accuracy level is defined as: $Accuracy_{95\%} =$ $Valueof(LSbit(Alt)) \cdot 2^{AAcc} =$ $1m \cdot 2^{AAcc}$ *AltAcc* = 00 is reserved to indicate that no specific accuracy interval is specified. The maximum error the 95 % accuracy interval can cover, represented by *AltAcc*=11, is 1 m ˙ 2^3 equivalent to about 1 m * 8 = 8 m.

Figure 18 : Configguration Parameters of LPV

Similar to LPVs are the Short Position Vectors (SPVs) that are used in some specific packet exchanges. SPVs have the following format:

0 0	 1	 2	 3	 4	 5	 6	 7	1 0	 1	 2	 3	 4	 5	 6	 7	2 0	 1	 2	 3	 4	 5	 6	 7	3 0	 1	 2	 3	 4	 5	 6	 7
GN_ADDR																															
TST																															
Lat																															
Long																															

Figure 19 : SPV format

7.3 GeoNetworking Location Table

Similar to other routing protocols, that have a routing table, the GeoNetworking implementation should hold a location table that will contain information about each ITS STA that participates in a message exchange. The location table should contain a flag of whether another ITS STA is a neighbor or not, and the pending operations to a specific station (such as if the location service is invoked).

The location table is used for taking all the forwarding decisions, so it should be as more precise as possible. The minimum location table elements, as defined by ETSI TS 102 636-4-1 are the following:

- GeoNetwork address of the ITS station GN_ADDR.
- LL address of the ITS station LL_ADDR.
- Type of the ITS station (vehicle ITS station, roadside ITS station).
- Position vector, i.e. Long Position Vector of the ITS station, comprised of:
    - Geographical position POS(GN_ADDR).
    - Speed S(GN_ADDR).
    - Heading H(GN_ADDR).
    - Timestamp of the geographical position TST(POS, GN_ADDR).
    - Accuracy of the geographical position Acc(POS, GN_ADDR).
    - Accuracy of the speed Acc(S, GN_ADDR).
    - Accuracy of the heading Acc(H, GN_ADDR).
- Flag LS_PENDING(GN_ADDR): Flag indicating that a Location Service (LS) (clause 9.2.4) is in progress.
- Flag IS_NEIGHBOUR(GN_ADDR): Flag indicating that the GeoAdhoc router is in direct communication range, i.e. is a neighbour.
- Sequence number SN(GN_ADDR): The sequence number of the last packet from the source GN_ADDR that was identified as 'not duplicated'.

8. GeoNetworking Packets

The packet structure of GeoNetworking protocol messages shall follow the format depicted in the following picture:

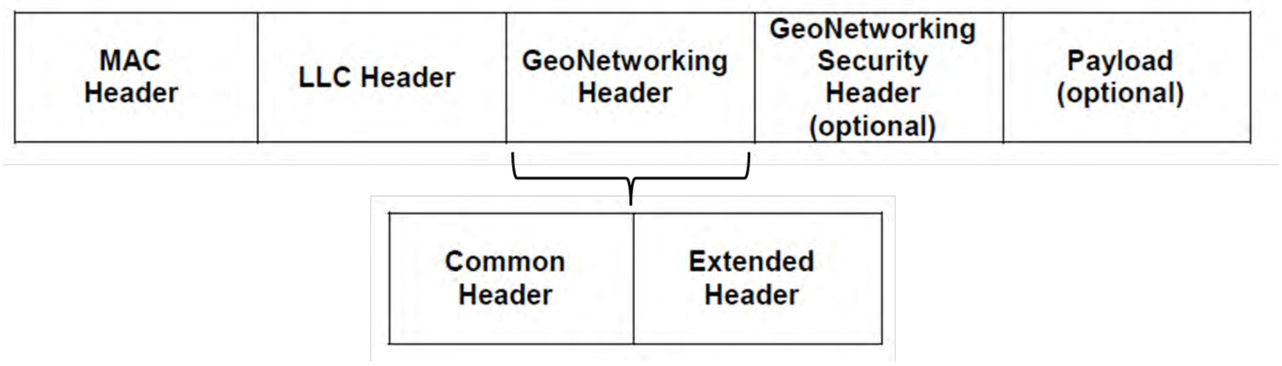


Figure 20 : GN packet structure

The GeoNetworking packets are encapsulated from the MAC layer protocol, according to what protocol has been adopted at the network access layer. Every GeoNetworking packet has a common header (which is the same for all the types of GN packets) and a packet specific extended header. The packet's common header is depicted in the following picture:

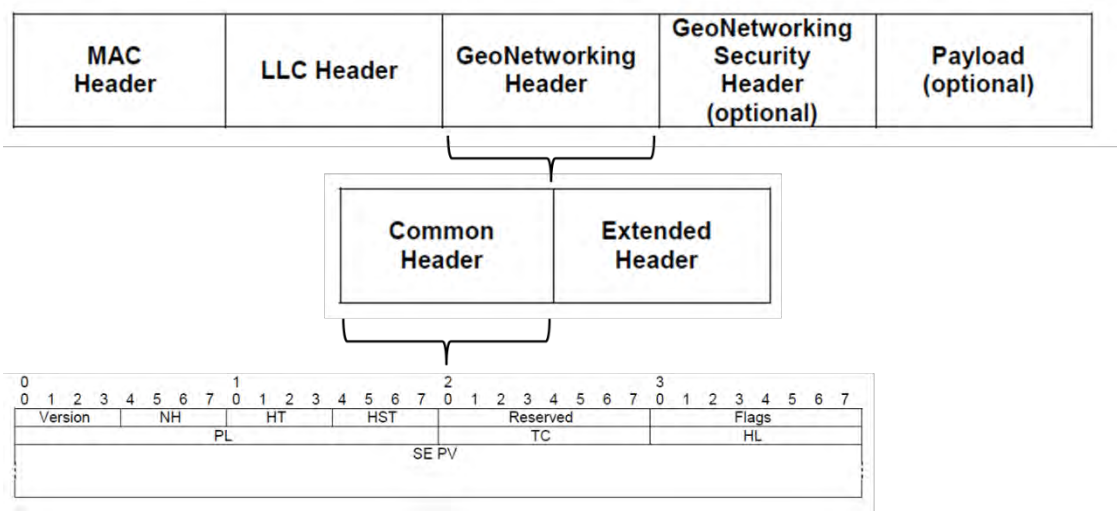
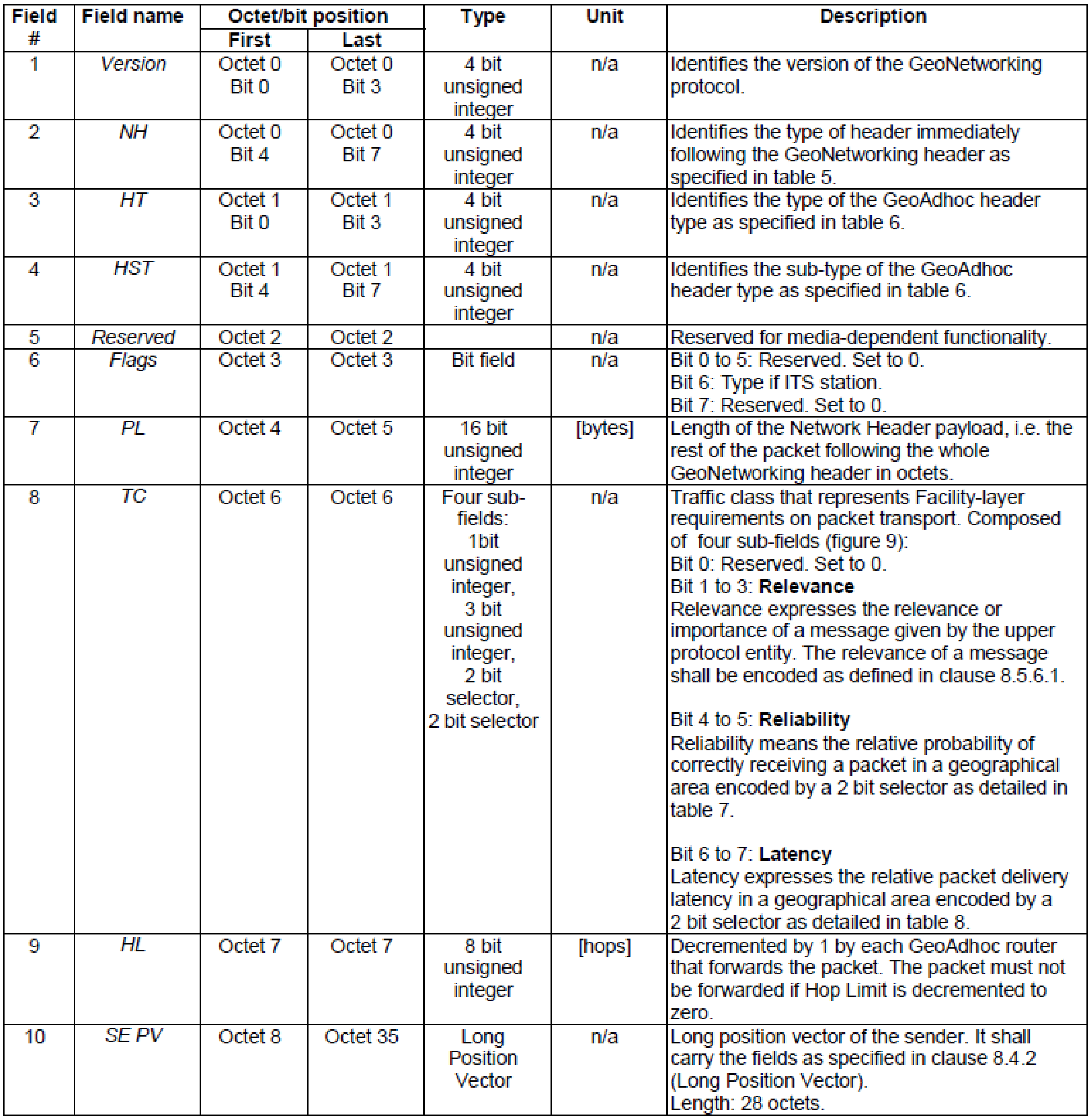


Figure 21 : Common Header Format

The common header consists of the following fields:



Field #	Field name	Octet/bit position		Type	Unit	Description
		First	Last			
1	*Version*	Octet 0 Bit 0	Octet 0 Bit 3	4 bit unsigned integer	n/a	Identifies the version of the GeoNetworking protocol.
2	*NH*	Octet 0 Bit 4	Octet 0 Bit 7	4 bit unsigned integer	n/a	Identifies the type of header immediately following the GeoNetworking header as specified in table 5.
3	*HT*	Octet 1 Bit 0	Octet 1 Bit 3	4 bit unsigned integer	n/a	Identifies the type of the GeoAdhoc header type as specified in table 6.
4	*HST*	Octet 1 Bit 4	Octet 1 Bit 7	4 bit unsigned integer	n/a	Identifies the sub-type of the GeoAdhoc header type as specified in table 6.
5	*Reserved*	Octet 2	Octet 2		n/a	Reserved for media-dependent functionality.
6	*Flags*	Octet 3	Octet 3	Bit field	n/a	Bit 0 to 5: Reserved. Set to 0. Bit 6: Type if ITS station. Bit 7: Reserved. Set to 0.
7	*PL*	Octet 4	Octet 5	16 bit unsigned integer	[bytes]	Length of the Network Header payload, i.e. the rest of the packet following the whole GeoNetworking header in octets.
8	*TC*	Octet 6	Octet 6	Four sub-fields: 1bit unsigned integer, 3 bit unsigned integer, 2 bit selector, 2 bit selector	n/a	Traffic class that represents Facility-layer requirements on packet transport. Composed of four sub-fields (figure 9): Bit 0: Reserved. Set to 0. Bit 1 to 3: **Relevance** Relevance expresses the relevance or importance of a message given by the upper protocol entity. The relevance of a message shall be encoded as defined in clause 8.5.6.1. Bit 4 to 5: **Reliability** Reliability means the relative probability of correctly receiving a packet in a geographical area encoded by a 2 bit selector as detailed in table 7. Bit 6 to 7: **Latency** Latency expresses the relative packet delivery latency in a geographical area encoded by a 2 bit selector as detailed in table 8.
9	*HL*	Octet 7	Octet 7	8 bit unsigned integer	[hops]	Decremented by 1 by each GeoAdhoc router that forwards the packet. The packet must not be forwarded if Hop Limit is decremented to zero.
10	*SE PV*	Octet 8	Octet 35	Long Position Vector	n/a	Long position vector of the sender. It shall carry the fields as specified in clause 8.4.2 (Long Position Vector). Length: 28 octets.

Figure 22 : Configuration Parameters for Common Header

The different packet types supported by the protocol are the following:

- Beacons
- Single Hop Broadcast packets (SHB)
- Topologically scoped Broadcast packets (TSB)
- GeoBroadcast packets
- GeoAnycast packets
- GeoUnicast packets
- Location Service Packets

Each one of the aforementioned packets, the packet structure and every operation is described in the following session:

8.1 Beacons

A beacon packet is consists of only the common header part of a packet. It is sent by any ITS STA to advertise its position at certain time intervals, if no other packet transmission has happened in this interval.

8.2 Single Hop Broadcast packets

SHB packets are broadcasted in all ITS stations within a 1-hop distance. They are used to encapsulate packets received from upper layers if indicated so by the “next header” field. Usually, they are used to encapsulate CAM messages.

8.3 Topologically Scoped Broadcast packets

TSB packets are packets broadcasted within a distance indicated by a hop limit. Due to multiple retransmissions of a packet by several ITS stations in the same geographical area, a duplicate packet detection mechanism should exist.

The extended packet header should contain the following fields, as indicated by the two next figures:

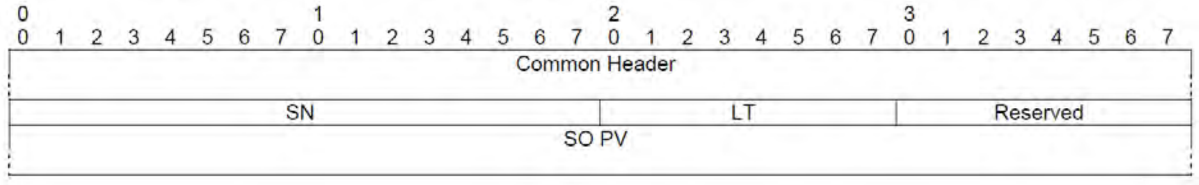
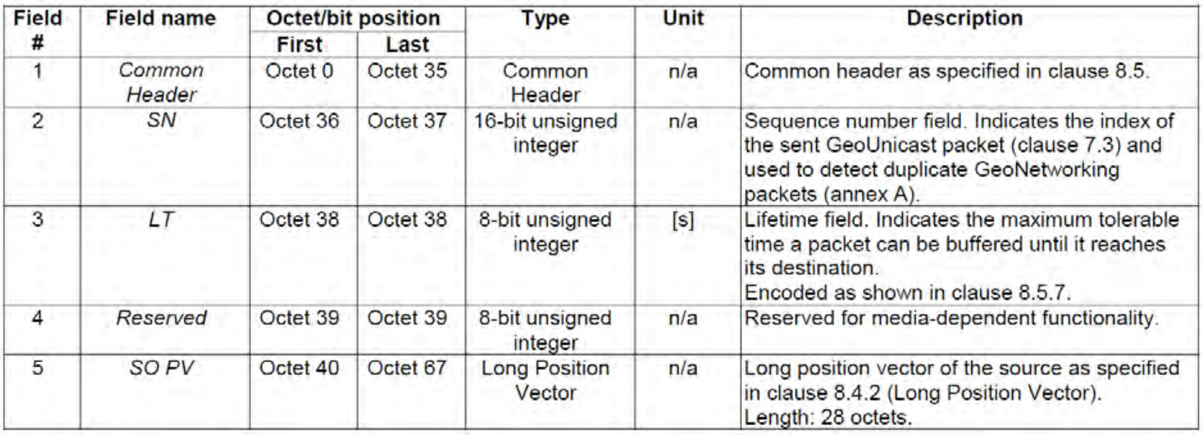


Figure 23 : TSB packet header



Field #	Field name	Octet/bit position		Type	Unit	Description
		First	Last			
1	*Common Header*	Octet 0	Octet 35	Common Header	n/a	Common header as specified in clause 8.5.
2	*SN*	Octet 36	Octet 37	16-bit unsigned integer	n/a	Sequence number field. Indicates the index of the sent GeoUnicast packet (clause 7.3) and used to detect duplicate GeoNetworking packets (annex A).
3	*LT*	Octet 38	Octet 38	8-bit unsigned integer	[s]	Lifetime field. Indicates the maximum tolerable time a packet can be buffered until it reaches its destination. Encoded as shown in clause 8.5.7.
4	*Reserved*	Octet 39	Octet 39	8-bit unsigned integer	n/a	Reserved for media-dependent functionality.
5	*SO PV*	Octet 40	Octet 67	Long Position Vector	n/a	Long position vector of the source as specified in clause 8.4.2 (Long Position Vector). Length: 28 octets.

Figure 24 : TSB Extended Header configuration parameters

8.4 GeoBroadcast/GeoAnycast packets

Geobroadcast/GeoAnycast packets perform the operation of broadcasting/anycasting (as anycast is defined in IPv6) in a specific geographical area. The definition of a geographical area is described in the next sections.

The GeoBroadcast and GeoAnycast packets shall have the same header structure. They are distinguished by the HT field in the Common Header. The header shall be comprised of the Common Header and the Extended Header as shown in the next figure.

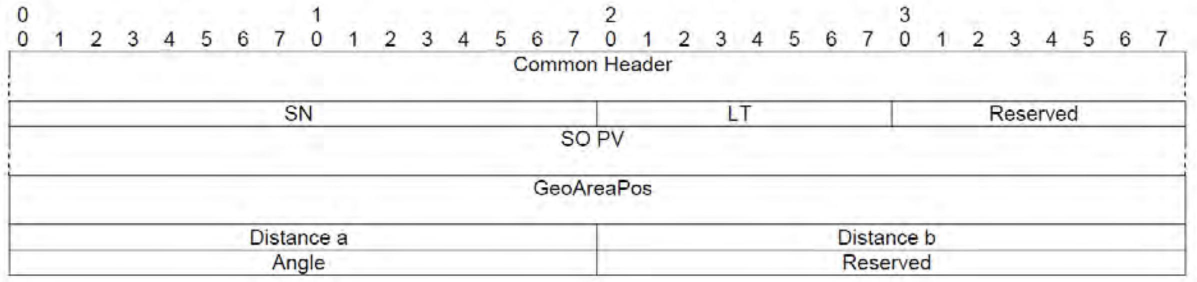
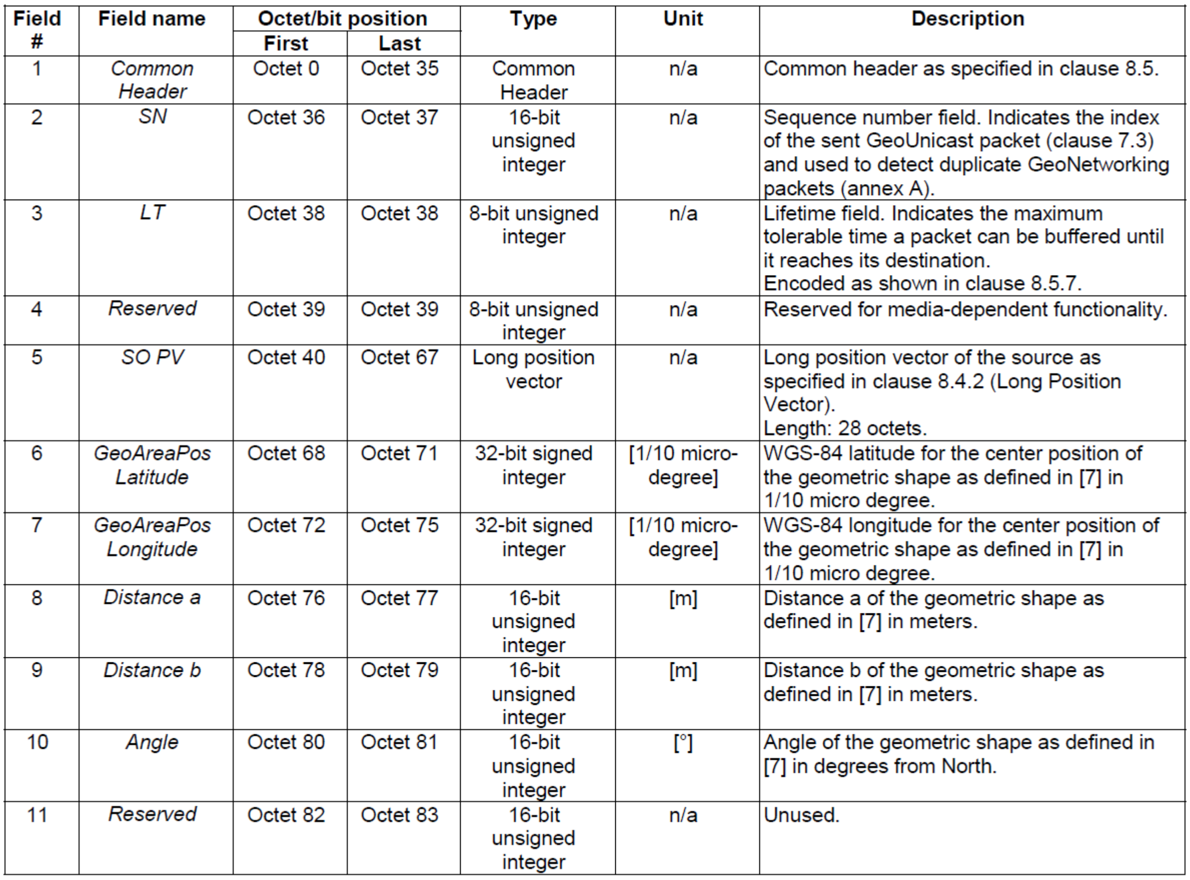


Figure 25 : GeoBroadcast/GeoAnycast packet header

The fields should carry the information indicated by the following table:



Field #	Field name	Octet/bit position		Type	Unit	Description
		First	Last			
1	*Common Header*	Octet 0	Octet 35	Common Header	n/a	Common header as specified in clause 8.5.
2	*SN*	Octet 36	Octet 37	16-bit unsigned integer	n/a	Sequence number field. Indicates the index of the sent GeoUnicast packet (clause 7.3) and used to detect duplicate GeoNetworking packets (annex A).
3	*LT*	Octet 38	Octet 38	8-bit unsigned integer	n/a	Lifetime field. Indicates the maximum tolerable time a packet can be buffered until it reaches its destination. Encoded as shown in clause 8.5.7.
4	*Reserved*	Octet 39	Octet 39	8-bit unsigned integer	n/a	Reserved for media-dependent functionality.
5	*SO PV*	Octet 40	Octet 67	Long position vector	n/a	Long position vector of the source as specified in clause 8.4.2 (Long Position Vector). Length: 28 octets.
6	*GeoAreaPos Latitude*	Octet 68	Octet 71	32-bit signed integer	[1/10 micro-degree]	WGS-84 latitude for the center position of the geometric shape as defined in [7] in 1/10 micro degree.
7	*GeoAreaPos Longitude*	Octet 72	Octet 75	32-bit signed integer	[1/10 micro-degree]	WGS-84 longitude for the center position of the geometric shape as defined in [7] in 1/10 micro degree.
8	*Distance a*	Octet 76	Octet 77	16-bit unsigned integer	[m]	Distance a of the geometric shape as defined in [7] in meters.
9	*Distance b*	Octet 78	Octet 79	16-bit unsigned integer	[m]	Distance b of the geometric shape as defined in [7] in meters.
10	*Angle*	Octet 80	Octet 81	16-bit unsigned integer	[°]	Angle of the geometric shape as defined in [7] in degrees from North.
11	*Reserved*	Octet 82	Octet 83	16-bit unsigned integer	n/a	Unused.

Figure 26 : GeoBroadcast/GeoAnycast Extended Header configuration parameters

As one can easily see, a geographical area is defined using three fields; distance a, distance b and angle. Specifically in the case of a circular area, a is the radius distance of the area, b is set to zero, and angle is set to zero.

8.5 GeoUnicast packets

GeoUnicast packets are used for unicast transmission to an ITS STA which we know the geographical characteristics of it. GeoUnicast header is defined as follows:

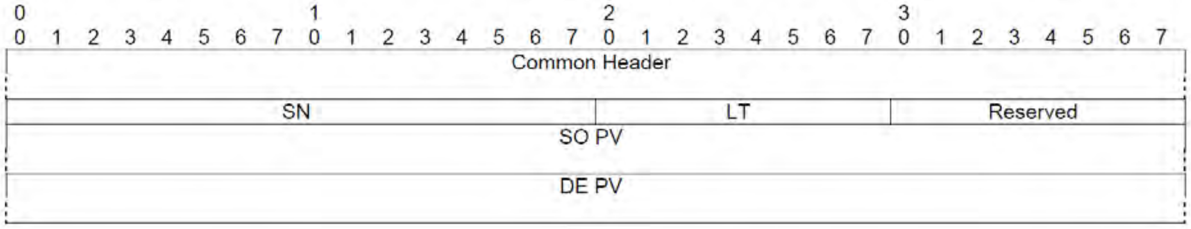
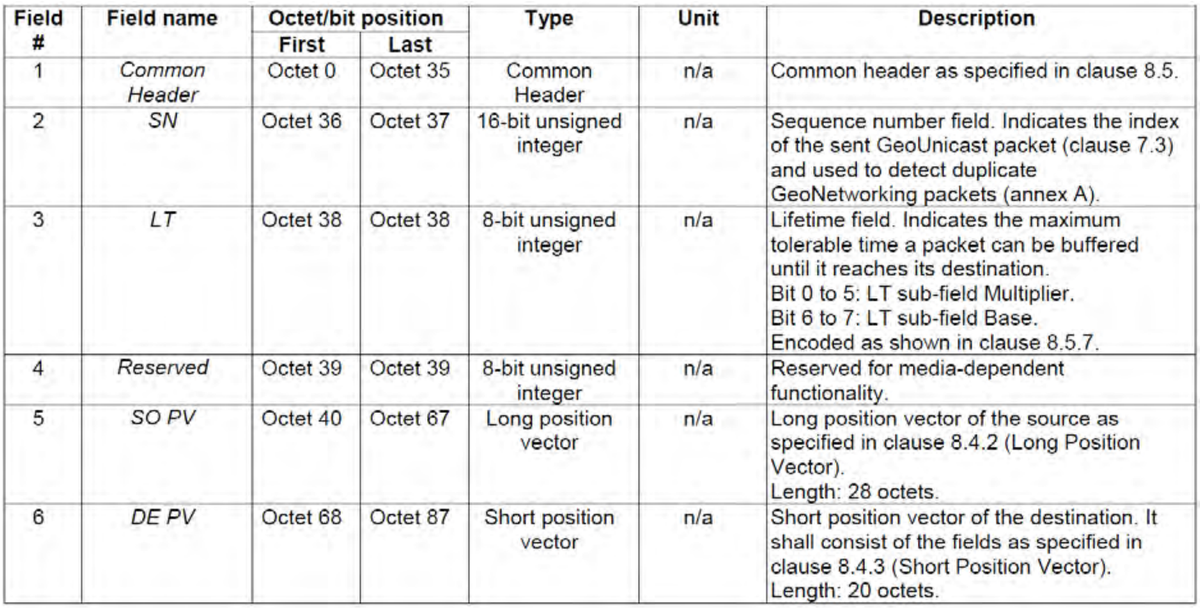


Figure 27 : GeoUnicast packet header

GeoUnicast extended header fields should be filled according to the next figure:



Field #	Field name	Octet/bit position		Type	Unit	Description
		First	Last			
1	*Common Header*	Octet 0	Octet 35	Common Header	n/a	Common header as specified in clause 8.5.
2	*SN*	Octet 36	Octet 37	16-bit unsigned integer	n/a	Sequence number field. Indicates the index of the sent GeoUnicast packet (clause 7.3) and used to detect duplicate GeoNetworking packets (annex A).
3	*LT*	Octet 38	Octet 38	8-bit unsigned integer	n/a	Lifetime field. Indicates the maximum tolerable time a packet can be buffered until it reaches its destination. Bit 0 to 5: LT sub-field Multiplier. Bit 6 to 7: LT sub-field Base. Encoded as shown in clause 8.5.7.
4	*Reserved*	Octet 39	Octet 39	8-bit unsigned integer	n/a	Reserved for media-dependent functionality.
5	*SO PV*	Octet 40	Octet 67	Long position vector	n/a	Long position vector of the source as specified in clause 8.4.2 (Long Position Vector). Length: 28 octets.
6	*DE PV*	Octet 68	Octet 87	Short position vector	n/a	Short position vector of the destination. It shall consist of the fields as specified in clause 8.4.3 (Short Position Vector). Length: 20 octets.

Figure 28 : GeoUnicast Extended Header configuration parameters

8.6 Location Service

The location service is used if a GeoAdhoc router needs to determine the position of another GeoAdhoc router. This is the case if a GeoAdhoc router is in the process to send a T/GN6-SDU as a GeoUnicast packet to another GeoAdhoc router, i.e. from the source to the destination, and does not have the position information for the destination in its LocT.

The execution of a location service is fully transparent to protocol entities of higher layers. The location service function resides on top of the forwarding functions and can therefore use any forwarding type. The location service is based on the exchange of control packets between GeoAdhoc routers.

The querying GeoAdhoc router (source) issues a LS Request packet with the GN_ADDR of the sought GeoAdhoc router (destination). The LS Request packet is forwarded by intermediate GeoAdhoc routers (forwarders) until it reaches the destination. The destination replies with a LS Reply packet.

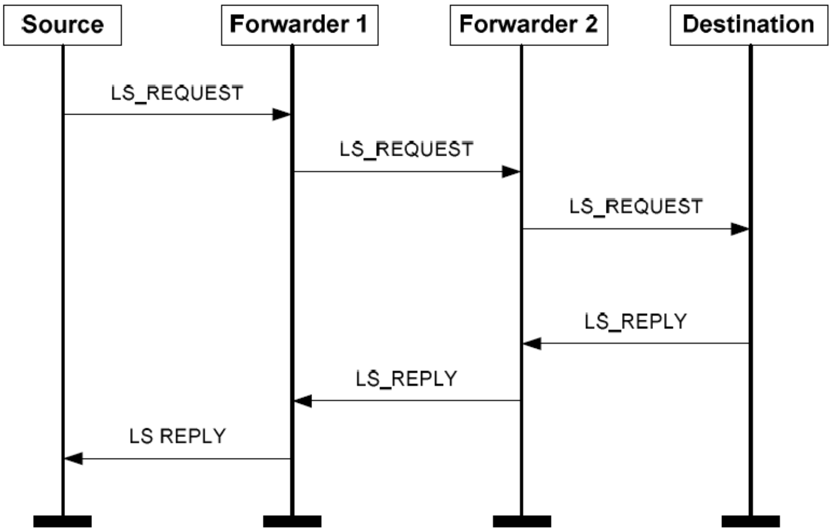


Figure 29 : Location Service example

The fields of the Common Header in the case of a Location Service message are defined in the following table:

Field name	Field setting	Description
Version	MIB attribute itsGnProtocolVersion	Version of the GeoNetworking protocol.
NH	0	Next header (table 5 in clause 8.5.3).
HT	6 (LS)	Header type (table 6 in clause 8.5.4).
HST	0 (LS_REQUEST) 1 (LS_REPLY)	Header sub-type (specified in table 6 in clause 8.5.4).
Reserved	0	Unused fields.
Flags	Bit 0 to 5: 0 Bit 6: itsGnStationType (MIB attribute) Bit 7: 0	Only Bit 6 used. Other fields reserved for future use.
PL	0	Payload length (packet does not carry payload).
TC	Bit 0: 0 (Reserved) Bit 1 to 3: MIB attribute itsGnTrafficClassRelevance (Relevance) Bit 4 to 5: MIB attribute itsGnTrafficClassReliability (Reliability) Bit 6 to 7: MIB attribute itsGnTrafficClassLatency (Latency)	Traffic class as specified in clauses 8.5.5 and 8.5.6.
HL	MIB attribute itsGnDefaultHopLimit	Hop limit.
SE PV	Actual values of the LPV (clause 7.2).	PV of the local GeoAdhoc router (sender of the LS Request and LS Reply packet).

Figure 30 : Common Header Location Service parameters

9. Defining a Geographical area

The definition of a geographical area is done in the standard ETSI EN 302 931. Geographical areas shall be specified by geometric shapes. The following geographical areas are defined:

- circular area;
- rectangular area; and
- elliptical area

9.1 Circular Area

The circular area shall be described by a circular shape with a single point A that represents the center of the circle and a radius r as shown in figure 31.

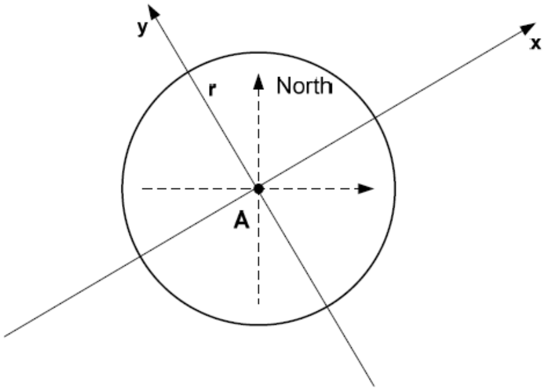


Figure 31 : Definition of a Circular Area

9.2 Rectangular Area

The rectangular area shall be defined by a rectangular shape (figure 3) with point A that represents the center of the rectangle and the following parameters:

- a - the distance between the center point and the short side of the rectangle (perpendicular bisector of the short side);
- b - the distance between the center point and the long side of the rectangle (perpendicular bisector of the long side);
- θ - azimuth angle of the long side of the rectangle.

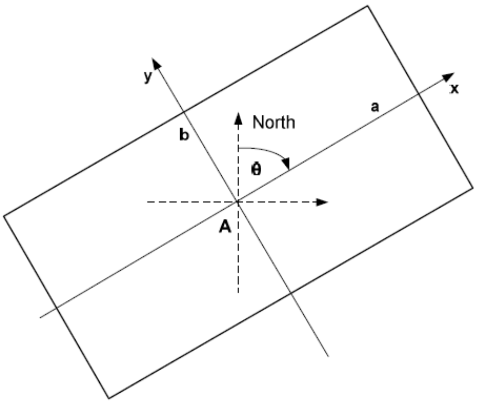


Figure 32 : Definition of a Rectangular Area

9.3 Elliptical Area

The elliptical area shall be defined by an elliptical shape with point A that represents the center of the ellipse and the following parameters:

- a - the length of the long semi-axis;
- b - the length of the short semi-axis;
- θ - azimuth angle of the long semi-axis.

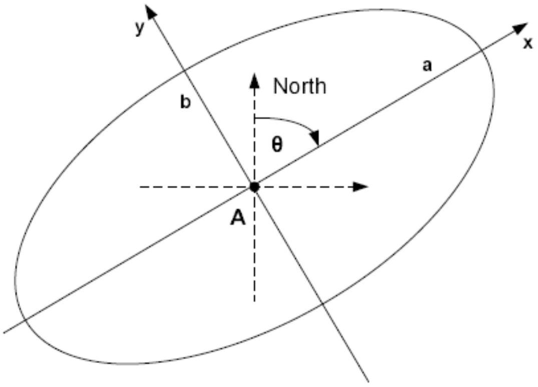


Figure 33 : Definition of an Elliptical Area

9.4 Geometric Function to determine spatial characteristics of a point

A function F that an ITS station can use to determine whether a point P(x,y) is located inside, outside, at the centre, or at the border of a geographical area should have the following properties:

$$F(x,y)\begin{cases} =1 & \text{for } x=0 \text{ and } y=0 \text{ (at the centre point)} \\ >0 & \text{inside the geographical area} \\ =0 & \text{at the border of the geographical area} \\ <0 & \text{outside the geographical area} \end{cases}$$

Function F(x,y) is depedent of the type of area defined. Therefore for each one of the available areas, we define the following functions:

1. For a circular area:

$$F(x,y)=1-\left(\frac{x}{r}\right)^2-\left(\frac{y}{r}\right)^2$$

2. For a rectangular area:

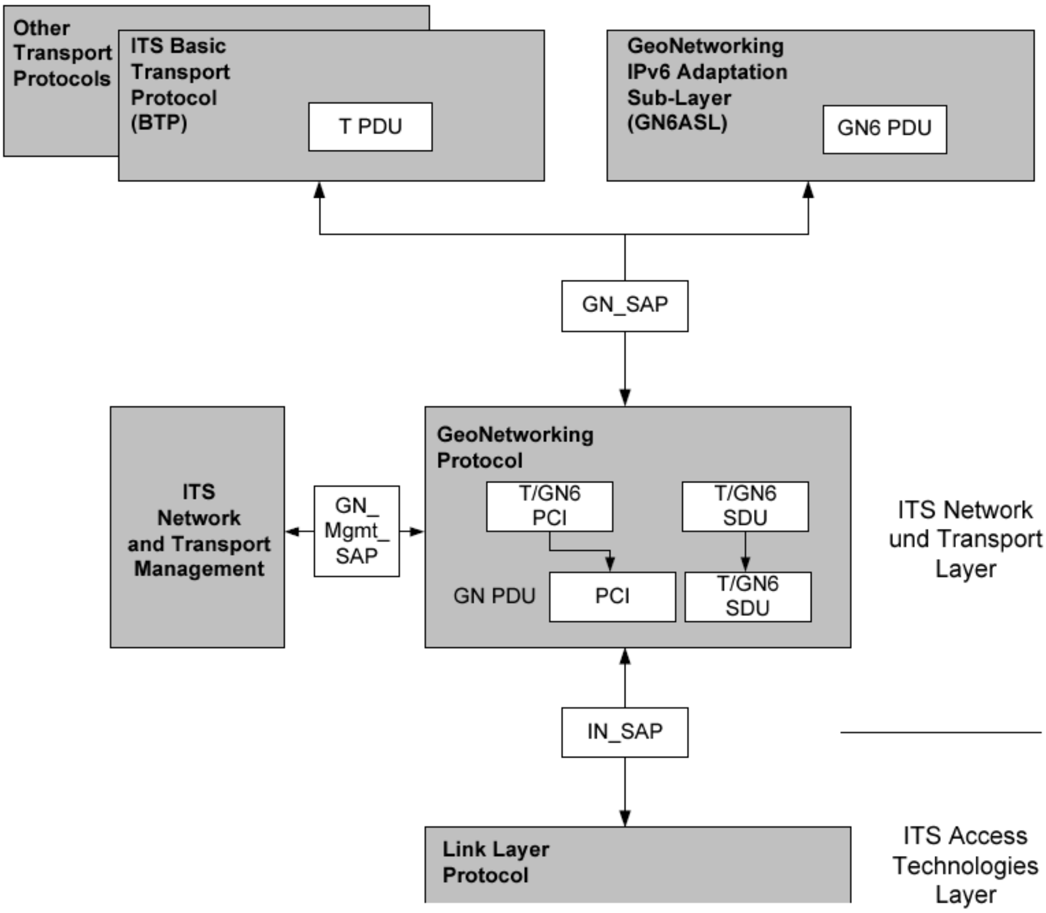
$$F(x,y)=Minimum\left(1-\left(\frac{x}{a}\right)^2,1-\left(\frac{y}{b}\right)^2\right)$$

3. For an elliptical area:

$$F(x,y)=1-\left(\frac{x}{a}\right)^2-\left(\frac{y}{b}\right)^2$$

10. Communication with upper layer

As we can deduce from the following picture, the communication between different layer entities is done via Service Access Points (SAP). The functions defined by the GN SAP are the GN-Data.request, the GN-Data.confirm and the GN-Data.indication messages.



Other Transport Protocols
ITS Basic Transport Protocol (BTP)
T PDU
GeoNetworking IPv6 Adaptation Sub-Layer (GN6ASL)
GN6 PDU
GN_SAP
ITS Network and Transport Management
GN_ Mgmt_ SAP
GeoNetworking Protocol
T/GN6 PCI
T/GN6 SDU
GN PDU
PCI
T/GN6 SDU
ITS Network und Transport Layer
IN_SAP
Link Layer Protocol
ITS Access Technologies Layer

Figure 34 : The GeoNetworking SAPs

The GN-DATA.request primitive is used by the ITS transport protocol entity to request sending a GeoNetworking packet. Upon reception of the GN-DATA.request primitive, the GeoNetworking protocol delivers the GeoNetworking packet to the LLC protocol entity via the IN_SAP.

The parameters of the GN-DATA.request are as follows:

GN-DATA.request (
Upper protocol entity,
Packet transport type,
Destination,
Communication profile,
Maximum packet lifetime, (optional)
Repetition interval, (optional)
Traffic class,

Length,
Data
)

The Upper protocol entity parameter specifies whether the primitive was triggered by an ITS Transport protocol (e.g. BTP) or by the GeoNetworking to IPv6 Adaptation Sub-Layer (GN6ASL).

The Packet transport type parameter specifies the packet transport type (GeoUnicast, SHB, TSB, GeoBroadcast, GeoAnycast).

The Destination parameter specifies the destination address for GeoUnicast or the geographical area for GeoBroadcast/GeoAnycast. The destination address for GeoUnicast can optionally contain the MID field only; with the other fields set to 0.

The Communication profile parameter determines the LL protocol entity (unspecified, ITS-G5A). The Maximum lifetime parameter specifies the maximum tolerable time in [s] a GeoNetworking packet can be buffered until it reaches its destination. The parameter is optional. If it is not used, the MIB attribute itsGnMaxPacketLifetime is used.

The Repetition interval parameter specifies the duration between two consecutive transmissions of the same GeoNetworking packet during the lifetime of a packet in [ms]. The parameter is optional. If it is not used, the packet should not be repeated.

The Traffic class parameter specifies the traffic class for the message as a triple of Relevance, Reliability, and Latency. The Length parameter indicates the length of the Data parameter. The Data parameter represents the payload of the GeoNetworking packet to be sent, i.e. the T-SDU/GN6-SDU.

The GN-DATA.confirm primitive is used to confirm that the GeoNetworking packet was successfully processed in response to a GN-DATA.request. For the reception of the primitive, no behaviour is specified.

The parameters of the primitive are as follows:

GN-DATA.confirm (
ResultCode
)

The ResultCode parameter specifies whether the GN-DATA.request primitive:

- has been accepted;
- rejected due to maximum length exceeded if the size of the T/GN6-PDU exceeds the MIB attribute itsGnMaxSduSize;
- reject due to maximum lifetime exceeded if the lifetime exceeds the maximum value of the MIB attribute itsGnMaxPacketLifetime;
- reject due to repetition interval too small, if the repetition interval is smaller than the MIB attribute itsGnMinPacketRepetitionInterval;
- rejected due to unsupported traffic class; or

- rejected for unspecified reasons if the GN-DATA.request primitive cannot accepted for any other reason.

The GN-DATA.indication primitive indicates to an upper protocol entity that a GeoNetworking packet has been received. The primitive is generated by the GeoNetworking protocol to deliver data contained in a received GeoNetworking packet to upper protocol entity. The data of the GeoNetworking packet are processed as determined by the receiving upper protocol entity.

The parameters of the GN-DATA.indication primitive are as follows:

GN-DATA.indication (

Upper protocol entity,
Packet transport type,
Destination Source position vector,
Traffic class,
Remaining packet lifetime (optional),
Length,
Data -- T/GN6-PDU
)

The Upper protocol entity parameter determines the protocol entity that processes the service primitive (BTP or GN6). The Packet transport type parameter is the packet transport type (GeoUnicast, SHB, TSB, GeoBroadcast, GeoAnycast) of the received packet.

The Destination parameter is the destination address for GeoUnicast or the geographical area for GeoBroadcast/GeoAnycast, with which the GeoNetworking packet was generated by the source. The Source position vector parameter is the geographical position for the source of the received GeoNetworking packet. The Remaining packet lifetime parameter is the remaining lifetime of the packet. The Traffic Class parameter is the traffic class, with which the GeoNetworking packet was generated by the source. The Length parameter is the length of the Data parameter. The Data parameter is the payload of the received GeoNetworking packet, i.e. the T-PDU/GN6-PDU.

11. Communication with the management layer

Communication with the management layer is done via the GN_MGMT_SAP. The messages exchanged are the GN-MGMT.request and the GN-MGMT.response.

The GN-MGMT.request primitive is generated by the GeoNetworking protocol entity at the initialization phase in order to request management information, i.e. time, position vector, GeoNetworking address. After receiving the GN-MGMT.request primitive, the ITS Network and Transport Layer Management entity is in charge of providing the GeoNetworking entity with the requested management information.

The parameters of the GN-MGMT.request are as follows:

GN-MGMT.request (
Request cause
)

The Request cause parameter specifies the type of requested information, i.e. time, position vector, GeoNetworking address. In case the GeoNetworking address is requested, the parameter also indicates whether the address request is caused by duplicate address detection or is an initial request.

The GN-MGMT.response primitive is generated by the ITS Network and Transport Layer Management entity to indicate an update of management information, i.e. time, position vector and GeoNetworking address. The primitive can be triggered upon reception of a GN-MGMT.request primitive or can be generated unsolicited, i.e. without a GN-MGMT.request primitive.

The parameters of the GN-MGMT.response are as follows:
GN-MGMT.response (
Time (optional)
Local position vector (optional)
GeoNetworking address (optional)
)

The Time parameter specifies the timestamp that is used as a reference to determine the freshness of received information carried in packets. The Local position vector parameter specifies the ITS station's most recent position vector (geographical position, speed, heading, timestamp when the position vector was generated, and corresponding accuracy information).

The GeoNetworking address parameter specifies the GeoNetworking address that shall be used by the GeoNetworking protocol entity. All parameters are optional, whereas at least one parameter should be present.

12. Other Implementation Details

12.1 Daemon Implementation details

For this thesis, we developed a user space daemon that provides full GeoNetworking capabilities to an ITS STA, by using its API. The architecture and the communication APIs that the GeoNetworking daemon has are depicted in the following picture.

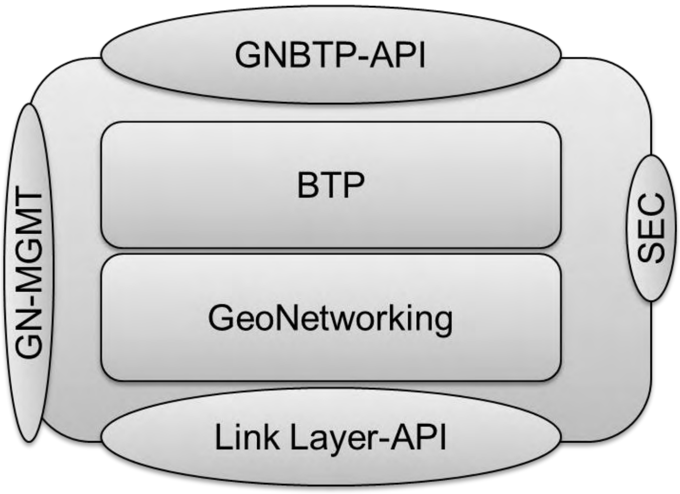


Figure 35 : GeoNetworking implementation architecture

The communication through the GNBTP-API and the GN-MGMT API are done using UDP sockets. As a communication scheme with the Link Layer, we used the pcap library.

12.1.1 Pcap Library

The pcap library provides a high level interface to the programmer, to access several features of a network interface. These could be from directly cloning a stream on the card to an application, to directly injecting traffic over a networking interface.

Therefore, for the proper operation of the GeoNetworking daemon, we needed to make calls to the pcap library, once we build the GN packet, one for injecting the created packet on the network interface and one thread listening for incoming packets marked as GeoNetworking.

12.1.2 Target Platform

The target platform for the evaluation of the system is DELPHI's embedded ITS CCU unit which is depicted in Figure 36.

Figure 36 : DELPHI's OBU

It features an Intel Atom processor at 1GHz and 1 GB of RAM memory. It has integrated GPS receiver and a 2 GB SSD. It has an option of adding LTE communication interfaces on top of it. The operating system that it runs is a customized by Delphi Linux Operating System.

As a communication interface with other CCU's it features an Atheros based WiFi card, supporting the IEEE 802.11p protocol. The frequency it is transmitting is at 5.9 GHz WAVE band with a channel bandwidth of 10MHz.

12.2 Management and POTI

For testing purposes, we used DELPHI's implementation of a facilities and management layer. The implementation was in java, using the OSGi [35] framework. Since this implementation is under the copyright law, only a binary form was given of it. Howerver, the high processing and memory requirements that are needed to run it, rendered it impossible to run on DELPHI's MyCCU.

Therefore, we developed lightweight versions of the facilities and management layer, written in C, which are able to provide basic support for some operations of these layers, and are able to operate on Delphi's CCUs. The operations supported so far are:

- Generation of CAM messages, that are send via the GNBTP-API to GeoNetworking layer via a GN-Data.request format
- Reception of CAM messages, received from the GNBTP-API in a GN-Data.indication format
- Generation of time and position update messages send via the GN-MGMT API as time and position update messages
- Generation of Address update messages sent to GN via the GN-MGMT API.

These operations run as two separate application daemons, able to give full GeoNetworking capabilities to an ITS STA.

Specifically, the position and time update module (POTI) is using another interface to communicate with the gpsd service, that is serving position updates in a JSON format. These are translated properly to GN compatible messages for proper handling on the GeoNetworking side.

12.2.1 GPSD and gps.h library

GPSD is a daemon that receives data from a GPS receiver, and provides the data back to multiple applications. It thus provides a unified interface to receivers of different types, and allows concurrent access by multiple applications.

GPSD provides a TCP/IP service by binding to port 2947. It accepts commands from that socket, and returns results back to it. These commands use a JSON-based syntax and return JSON responses (older, now obsolete versions used single-letter commands). Concurrent operation is supported. Most GPS receivers are supported, whether serial, USB, or Bluetooth. Additionally gpsd supports interfacing with the UNIX network time protocol daemon ntpd via shared memory to enable setting the host platform's time via the GPS clock.

The corresponding UNIX gps.h library provides an interface to the programmer to interact with the gpsd daemon and send or receive data to its listening socket. By sending a WATCH message, a JSON file is returned by gpsd containing current GPS data.

12.3 Logging Functions

For proper debugging of the protocol’s operation, we have implemented logging mechanisms at the different levels that the developed software operates. Logging support is provided not only for the Geonetworking layer, but for the management and the facilities layer as well. Each time a position update is sent (defined by a MIB value), it is logged by the logging thread running at this layer. Similar process is happening at the facilities layer, whenever a packet is sent to or received from the GeoNetworking layer.

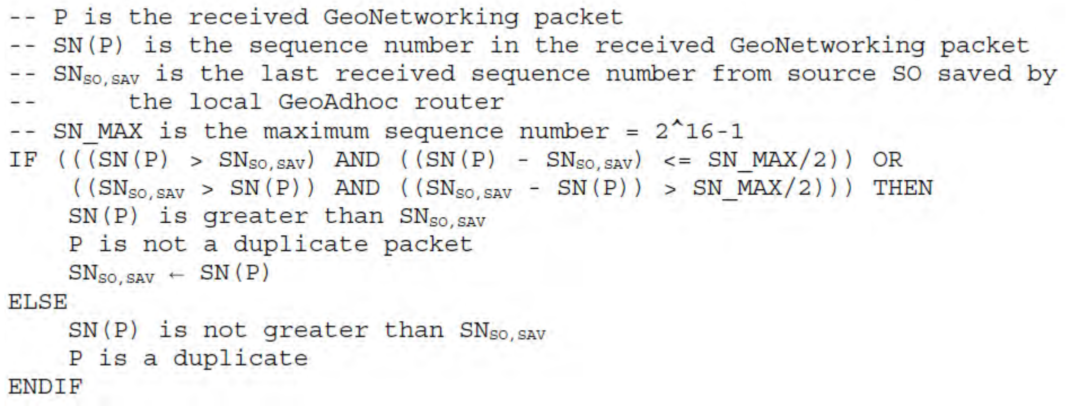
For the implementation of the logging process, we engaged C++ file streams library. Log files are renewed in a circular way, and every time a timer expires (originally set to a daylong duration), the log files are archived. The compression of the files is happening using a system call to the `tar` and `gunzip` UNIX commands. For the compression of the files, we could have engaged the `zlib` library and actually implement the compression process using our own code. Since the logging archiving is done only once per day, we used the system calls solution, which can ensure interoperability among different UNIX based systems. Implementation of compression functions using the `zlib` library is one of our future goals.

12.4 Duplicate packet detection

As it is expected from the protocol’s operation, several of the packets are retransmitted over the network more than once, depending on the number of ITS Stations in a geographical area. Therefore, proper handling of duplicate packets shall be implemented.

The GeoNetworking protocol applies duplicate packet detection to multi-hop packets (GeoUnicast, TSB, GeoBroadcast, GeoAnycast, LS Request, LS Reply). The mechanism is based on sequence numbers and implies that every GeoNetworking packet carries a sequence number in its header. For duplicate packet detection, a GeoAdhoc router maintains the sequence number of the last packet from the source that was identified as 'not duplicated' in its LocT, i.e. SNGN_ADDR. When the GeoAdhoc router processes a GeoNetworking packet, it compares the value of the SN field carried in the GeoNetworking packet SN(P) and SNSO,SAV. If SN(P) is greater than SNGN_ADDR the received packet is regarded as 'not duplicated' and SNSO,SAV is updated.

The sequence numbers in the GeoNetworking protocol are limited in the number of bits that represent the sequence number. In order to handle the wrap-around of sequence numbers (that the sequence number is incremented from the maximum possible value to zero), the following algorithm shall be used:



```
-- P is the received GeoNetworking packet
-- SN(P) is the sequence number in the received GeoNetworking packet
-- $SN_{SO,SAV}$ is the last received sequence number from source SO saved by
--      the local GeoAdhoc router
-- SN_MAX is the maximum sequence number = 2^16-1
IF (((SN(P) > $SN_{SO,SAV}$) AND ((SN(P) - $SN_{SO,SAV}$) <= SN_MAX/2)) OR
    (($SN_{SO,SAV}$ > SN(P)) AND (($SN_{SO,SAV}$ - SN(P)) > SN_MAX/2))) THEN
    SN(P) is greater than $SN_{SO,SAV}$
    P is not a duplicate packet
    $SN_{SO,SAV}$ ← SN(P)
ELSE
    SN(P) is not greater than $SN_{SO,SAV}$
    P is a duplicate
ENDIF
```

12.5 GeoNetworking Management Information Base (MIB)

In the protocol’s specification, a Management Information Base (MIB) is defined, with values that define its operation. However, in order to avoid adding more complexity to the developed daemon, we created a sample configuration file which is parsed at the beginning of execution of the daemon. This file is used to initialize a class that contains all the MIB values used by the protocol. The sample configuration file that is read by the GeoNetworking daemon and the MIB values stored are depicted in the following codebox and the following table:

```
#################################################
#                                               #
# GeoNetworking Daemon Configuration File       #
# Proposal for Discussion                       #
#   Version: 0.0.1                              #
#                                               #
#################################################
# $Rev: 0001 $
########## <Configuration Table>
# (1) Log Functionality
# (2) Device Confiuration
# (3) Data Communication API Settings
# (4) ITS Station Settings
# (5) Default Local Position Vector Settings
# (6) Protocol Configuration
# (7) Default Transmission Parameters
# (8) Testing Configuration
##########
## (1) Log Functionality
# File path
GN_SYSLOG_FILENM gn_sys_log
GN_USRLOG_FILENM gn_usr_log
## Log level
GN_SYSLOG_LV              3       # System log level (0-3)
GN_USRLOG_LV              2       # User log level (0-3)
# Log rotation parameters
GN_LOG_MAX_NLINES50000    # Maximum # of lines by file (rotate mode = 1)
GN_LOG_MAX_SIZE           4000000 # Maximum size by file (rotate mode = 2)
GN_LOG_N_ROTATE           2            # Number of log rotate files (>= 2)
## (2) Device Confioguration
AL_CCH_IF                 eth0    # Interface name used for CCH
AL_SCH_IF                 eth0   # Interface name used for SCH
## (3) Data Communication API Settings
# Listen port for communication with Facility
GN_NWT_LISTEN_PORT        1301
# Destination Port for communication with Facility
GN_FAC_DEST_ADDR 127.0.0.1
GN_FAC_DEST_PORT 1302
```

```
# Listen port for communication with Management
GN_MNGT_LISTEN_PORT       1401
# Destination Port for communication with Management
GN_MNGT_DEST_ADDR127.0.0.1
GN_MNGT_DEST_PORT1402
## (4) Node Settings Configuration
# ITS Station Type
#  (bike, motorbike, car, truck, bus, trafficlight, rsu)
GN_ITSST_TYPE             rsu
# ITS Station SubType
#  (public, private)
GN_ITSST_SUBTYPE private
# Address Configuration Mode (MIB.itsGnLocalAddrConfMethod)
#  (auto, managed, manual)
GN_ADDRCONF_MODE auto
# MID of Geonetworking address (only if GN_ADDRCONF_MODE = manual)
#   Format: XX:XX:XX:XX:XX:XX (X is hexdecimal character)
GN_ADDR_MID               00:00:00:00:00:00
## (5) Default LPV Settings
GN_DEF_LAT                40.4
GN_DEF_LON                10.2
GN_DEF_SPEED              10
GN_DEF_HEADING            0
GN_DEF_ALT                20
## (6) Protocol Configuration
# Algorithm Type Settings
# GeoUnicast Algorithm Type (MIB.itsGnGeoUnicastForwardingAlgorithm)
#  (0: Unspecified, 1: Greedy, 2: ETSI-CBF, 3: Revised-CBF)
GN_UCALG_TYPE             0
# GeoBroadcast Algorithm Type (MIB.itsGnGeoBroadcastForwardingAlgorithm)
#  (0: Unspecified, 1: Simple, 2: Advanced(not available now))
GN_BCALG_TYPE             0
# Default Hop Limit (0-255) (MIB.itsGnDefaultHopLimit)
GN_HOP_LIMIT              10
# Upper Limit of Packet Lifetime (1-6300000) [ms] (MIB.itsGnMaxPacketLifetime)
GN_MAX_PCKT_LIFETM        600000
# Lower Limit of the Packet Repetition Interval [ms] (MIB.itsGnMinPacketRepetitionInterval)
GN_MIN_PCKT_REPINT        100
```

```
## (8) Testing Configuration

### Radio Range Emulation

# Whether emulation functionality is enabled or not

#  0: Disabled, 1: Enabled

GN_RADIO_EMU_MODE       0

# Emulated radio range [m]

GN_RADIO_EMU_RANGE        300
```

Table 2 : MIB values and their initial configuration

MIB attribute value	Default/Initial value	Comment
itsGnLocalGnAddr	1	GeoNetworking address of the GeoAdhoc router
itsGnLocalAddrConfMethod	AUTO (0) MANAGED (1)	AUTO: Local GN_ADDR is configured from MIB MANAGED: Local GN_ADDR is configured via the GN-MGMTt primitive (annex I)
itsGnProtocolVersion	TS 102 636-4-1 (V1.1.1) (0)	Version of the GeoNetworking protocol set in the GeoNetworking protocol headers
itsGnStationType	Vehicle ITS Station (0) Roadside ITS Station (1)	Type of ITS Station
itsGnMinimumUpdateFrequencyLPV	Vehicle ITS Station (1 000) Roadside ITS Station (0)	Minimum update frequency of local position vector (LPV) in ms
itsGnMaxSduSize	1 398	Maximum size of GN-SDU [bytes] 1 500- GN_MAX (88) - GNSEC_MAX (0)
itsGnMaxGeoNetworkingHeaderSize	88	GN_MAX: Maximum size of GeoNetworking header [bytes] Determined by the GeoUnicast header as defined in clause 8.6.2
itsGnLifetimeLocTE	20	Lifetime of location table entry [s]
itsGnLocationServiceMaxRetrans	10	Maximum number of retransmissions of LS Request packets
itsGnLocationServiceRetransmitTimer	1 000	Duration of Location service retransmit timer [ms]
itsGnLocationServicePacketBufferSize	1 024	Size of Location service packet buffer [Byte]
itsGnBeaconServiceRetransmitTimer	3 000	Duration of Beacon service retransmit timer [ms]
itsGnBeaconServiceMaxJitter	itsGnMaxPacketLifetime/4	Maximum beacon jitter [ms]
itsGnDefaultHopLimit	10	Default hop limit indicating the maximum number of hops a packet

		travels
itsGnMaxPacketLifetime	600	Upper limit of the maximum lifetime [s]
itsGnMinPacketRepetitionInterval	100	Lower limit of the packet repetition interval [ms]
itsGnGeoUnicastForwardingAlgorithm	UNSPECIFIED (0) GREEDY (1) CBF (2)	Default GeoUnicast forwarding algorithm
itsGnGeoBroadcastForwardingAlgorithm	UNSPECIFIED (0) SIMPLE (1)	Default GeoBroadcast forwarding algorithm
itsGnGeoUnicastCbfMinTime	1	Minimum duration a packet shall be buffered in the CBF packet buffer [ms]
itsGnGeoUnicastCbfMaxTime	100	Maximum duration a packet shall be buffered in the CBF packet buffer [ms]
itsGnDefaultMaxCommunicationRange	1 000	Default theoretical maximum communication range [m]
itsGnGeoAreaLineForwarding	DISABLED (0) ENABLED (1)	Forwarding of GeoBroadcast/GeoAnycast packet if GeoAdhoc router is located outside the GeoArea.
itsGnUcForwardingPacketBufferSize	256	Size of UC forwarding packet buffer [Kbytes].
itsGnBcForwardingPacketBufferSize	1 024	Size of BC forwarding packet buffer [Kbytes].
itsGnCbfPacketBufferSize	256	Size of CBF packet buffer [Kbytes]
itsGnTrafficClassRelevance	3	Forwarding: Default traffic class Relevance
itsGnTrafficClassReliability	Medium (10)	Forwarding: Default traffic class Reliability
itsGnTrafficClassLatency	Medium (10)	Forwarding: Default traffic class Latency

12.6 GeoNetworking Operation

The ETSI TS 102 636-4-1 standard, defines different operation of the protocol depending on the packet type sent, and whether the GeoNetworking STA is the transmitter, the forwarder or the receiver of the packet.

12.6.1 Transmission of packets

In the case of receiving a GN-Data.indication from an upper layer, the GeoNetworking daemon checks the type of the packet that shall be transmitted and performs some packet type specific actions that are defined in the following sections:

12.6.2 Single Hop Broadcast

On reception of a GN-DATA.request primitive with a Packet transport type parameter set to SHB, the source shall execute the following operations:

1) create a GN-PDU with the T/GN6-SDU as payload and a SHB packet header:
    a) set the fields of the Common Header (clause 9.3.2);
2) if no neighbour exists, i.e. the LocT does not contain a LocTE with the IS_NEIGHBOUR flag set to TRUE, then buffer the SHB packet in the BC forwarding packet buffer and omit the execution of further steps;
3) if the optional Repetition interval parameter in the GN-DATA.request parameter is set:
    a) save the SHB packet;
    b) retransmit the packet with period as specified in Repetition interval until the maximum lifetime of the packet is expired;
4) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
    a) UNSPECIFIED then omit this operation;
    b) is set to ITS-G5A then execute the operations as specified;
5) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the Broadcast address of the LL entity.

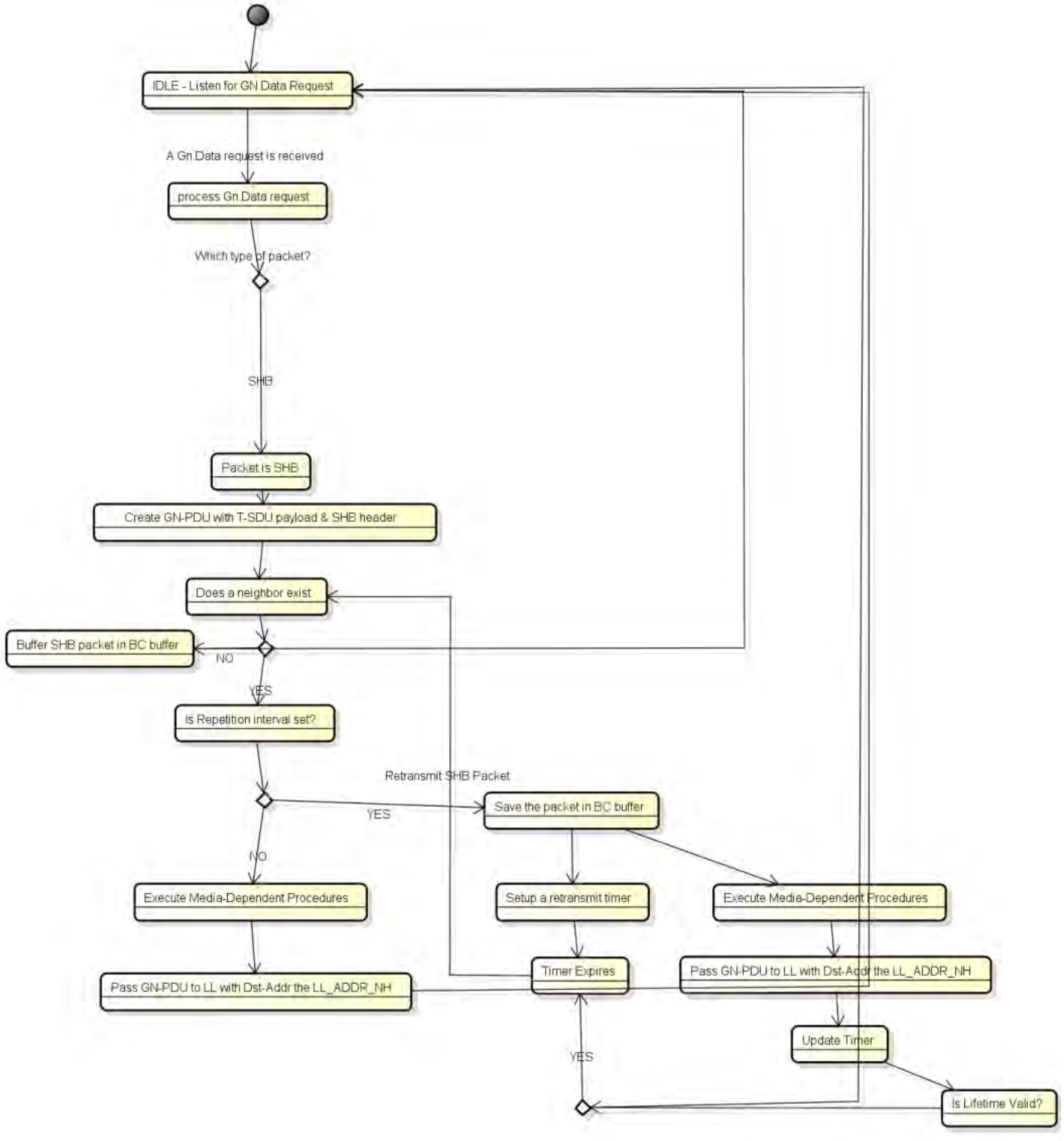

IDLE - Listen for GN Data Request
A Gn.Data request is received
process Gn.Data request
Which type of packet?
SHB
Packet is SHB
Create GN-PDU with T-SDU payload & SHB header
Does a neighbor exist
Buffer SHB packet in BC buffer
NO
YES
Is Repetition interval set?
Retransmit SHB Packet
YES
Save the packet in BC buffer
NO
Execute Media-Dependent Procedures
Setup a retransmit timer
Execute Media-Dependent Procedures
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Timer Expires
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Update Timer
YES
Is Lifetime Valid?

Figure 37: SHB transmission flowchart

12.6.2.1 Topologically Scoped Broadcast

On reception of a GN-DATA.request primitive with a Packet transport type parameter set to TSB, the source shall execute the following operations:

1) create a GN-PDU with the T/GN6-SDU as payload and a TSB packet header:
   a) set the fields of the Common Header
   b) set the fields of the TSB Extended Header

2) if no neighbour exists, i.e. the LocT does not contain a LocTE with the IS_NEIGHBOUR flag set to TRUE, then buffer the TSB packet in the BC forwarding packet buffer and omit the execution of further steps;
3) if the optional Repetition interval parameter in the GN-DATA.request parameter is set:
   a) save the TSB packet
   b) retransmit the packet with period as specified in Repetition interval until the maximum lifetime of the packet is expired;
4) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
   a) then omit this operation;
   b) is set to ITS-G5A then execute the operations as specified
5) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the Broadcast address of the LL entity.

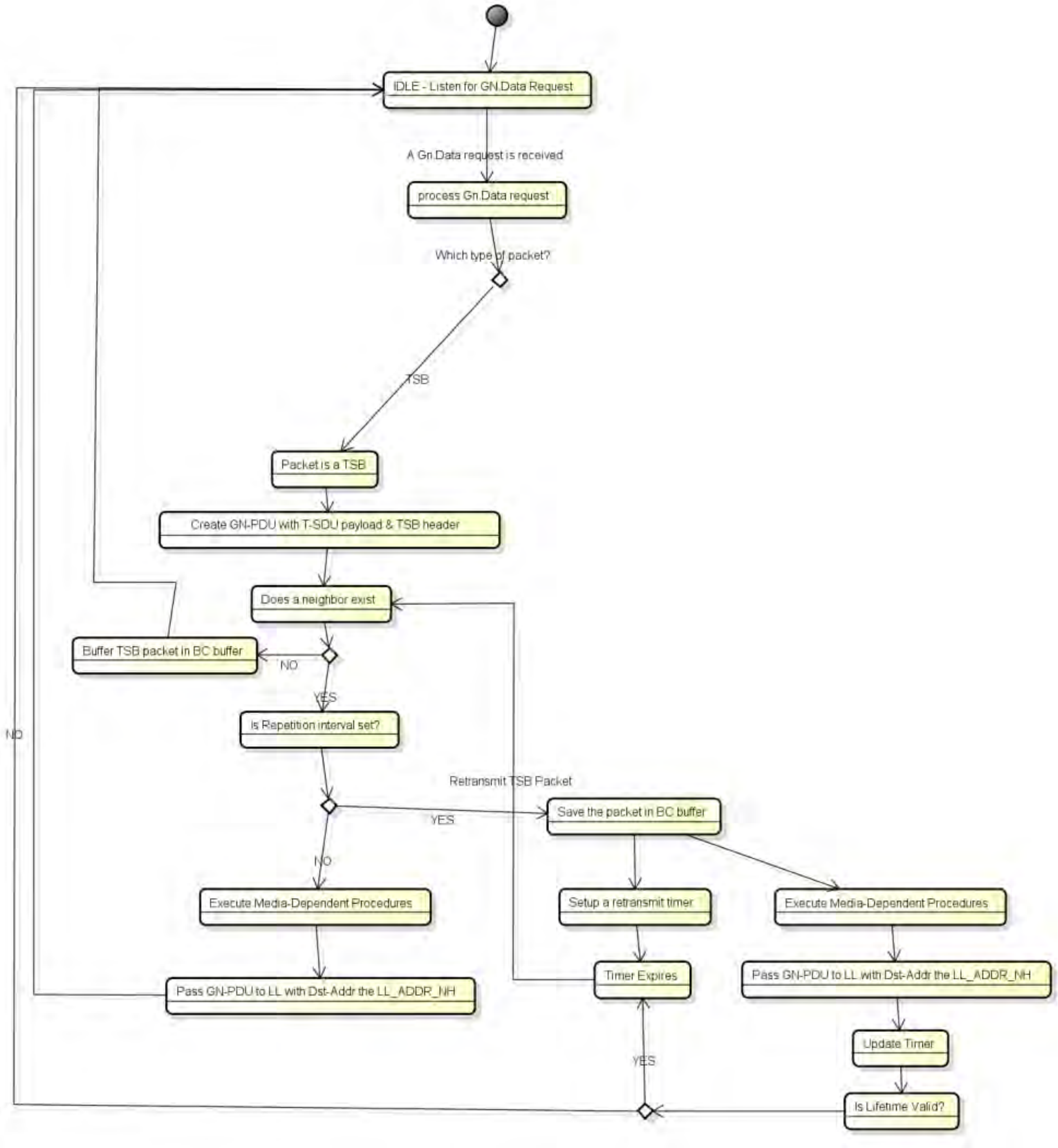

IDLE - Listen for GN.Data Request
A Gn.Data request is received
process Gn.Data request
Which type of packet?
TSB
Packet is a TSB
Create GN-PDU with T-SDU payload & TSB header
Does a neighbor exist
Buffer TSB packet in BC buffer
NO
YES
NO
Is Repetition interval set?
Retransmit TSB Packet
YES
Save the packet in BC buffer
NO
Execute Media-Dependent Procedures
Setup a retransmit timer
Execute Media-Dependent Procedures
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Timer Expires
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Update Timer
YES
Is Lifetime Valid?

Figure 38 : TSB transmission flowchart

12.6.2.2 GeoBroadcast

On reception of a GN-DATA.request primitive with a Packet transport type parameter set to GeoBroadcast, the source shall execute the following operations:

1) create a GN-PDU with the T/GN6-SDU as payload and a GeoBroadcast packet header:
    a) set the fields of the Common Header

    b) set the fields of the GeoBroadcast Extended Header

2) if no neighbour exists, i.e. the LocT does not contain a LocTE with the IS_NEIGHBOUR flag set to TRUE, then buffer the GeoBroadcast packet in the BC forwarding packet buffer and omit the execution of further steps;
3) if the optional Repetition interval parameter in the GN-DATA.request parameter is set:
    a) save the GeoBroadcast packet
    b) retransmit the packet with period as specified in Repetition interval until the maximum lifetime of the packet is expired
4) determine the link-layer address LL_ADDR_NH of the next hop
    a) if the MIB attribute itsGnGeoBroadcastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the Simple GeoBroadcast with line forwarding algorithm
    b) if the MIB attribute itsGnGeoBroadcastForwardingAlgorithm is set to 1 (SIMPLE), execute the Simple GeoBroadcast with line forwarding algorithm
5) if LL_ADDR_NH = 0, then buffer the GeoBroadcast packet in the BC forwarding packet buffer and omit the execution of further steps;
6) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
    a) UNSPECIFIED then omit this operation
    b) is set to ITS-G5A then execute the operations
7) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the LL address of the next hop LL_ADDR_NH.

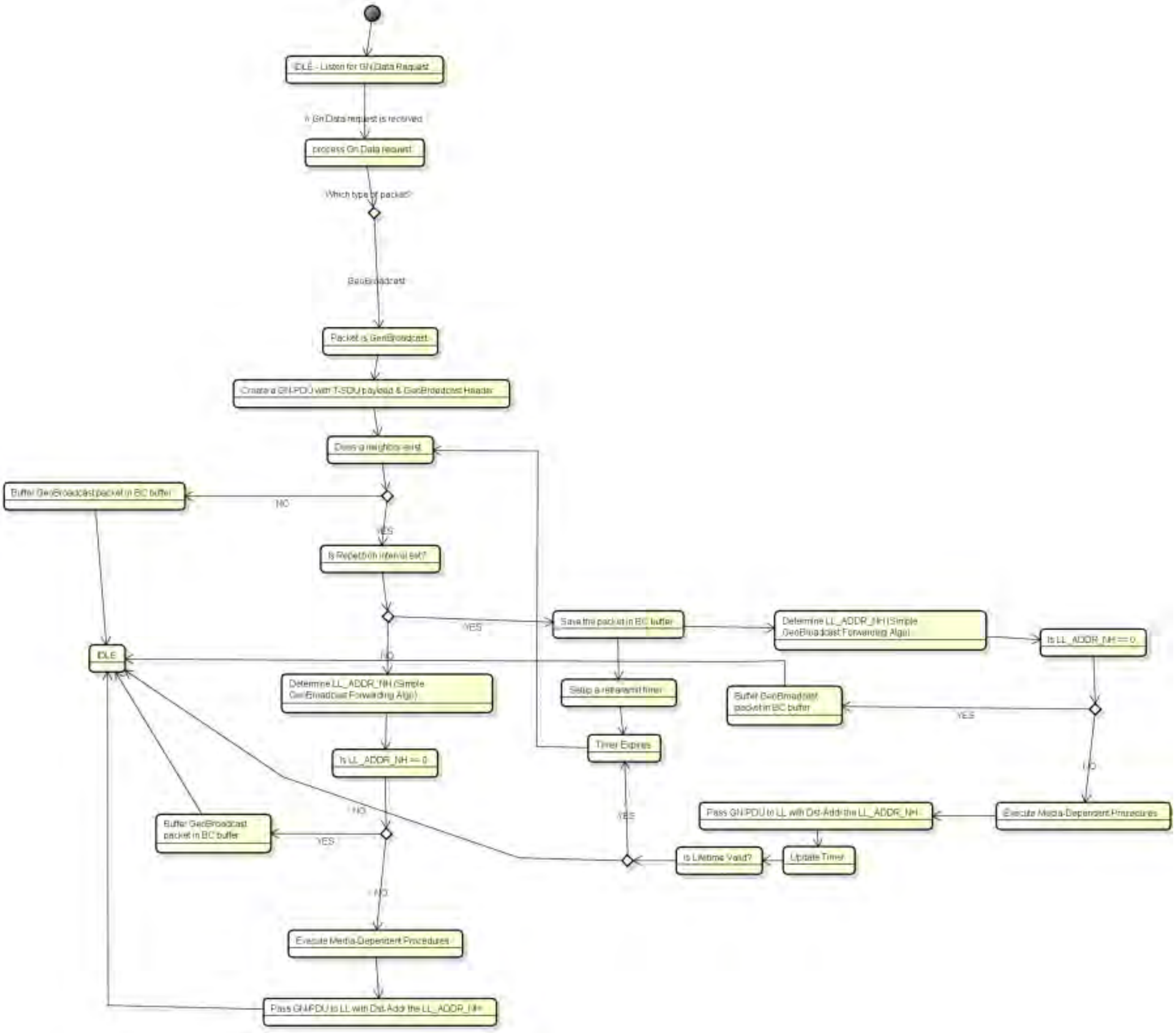

IDLE - Listen for GN.Data Request
/i GnData request is received
process Gn.Data request
Which type of packet?
GeoBroadcast
Packet is GeoBroadcast
Create a GN-PDU with T-SDU payload & GeoBroadcast Header
Does a neighbor exist
Buffer GeoBroadcast packet in BC buffer
NO
YES
Is Repetition interval set?
YES
Save this packet in BC buffer
Determine LL_ADDR_NH (Simple GeoBroadcast Forwarding Algo)
Is LL_ADDR_NH == 0
IDLE
NO
Determine LL_ADDR_NH (Simple GeoBroadcast Forwarding Algo)
Setup a retransmit timer
Buffer GeoBroadcast packet in BC buffer
YES
Timer Expires
Is LL_ADDR_NH == 0
NO
YES
NO
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Execute Media-Dependent Procedures
Buffer GeoBroadcast packet in BC buffer
YES
Is Lifetime Valid?
Update Timer
NO
Execute Media-Dependent Procedures
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH

Figure 39 : GeoBroadcasst transmission flowchart

12.6.2.3 GeoAnycast

The operations of the source of a GeoAnycast packet are identical with the source of a GeoBroadcast packet, except the operation in step 5. Instead, the source shall execute the following operation:

1) determine function F(x,y)
   a) if F (x, y) < 0 (GeoAdhoc router is outside the geographical area) and the MIB attribute itsGnGeoAreaLineForwarding is set to TRUE, execute the GeoUnicast forwarding algorithm and determine the link-layer address LL_ADDR_NH of the next hop.
   b) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the GF algorithm
   c) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 1 (GREEDY), execute the GF algorithm
   d) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 2 (CBF), execute the CBF algorithm

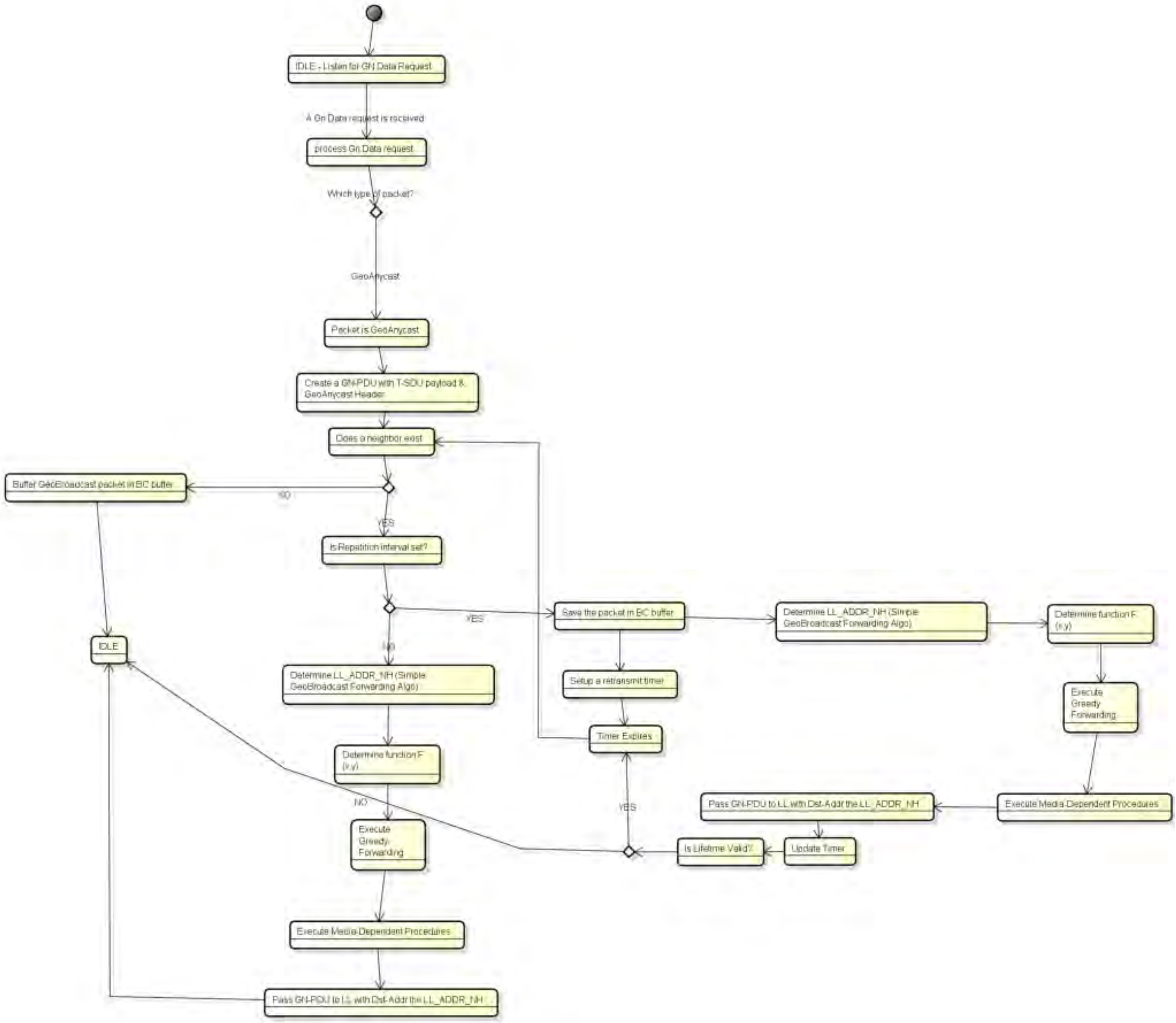

IDLE - Listen for GN Data Request
A Gn Data request is received
process Gn Data request
Which type of packet?
GeoAnycast
Packet is GeoAnycast
Create a GN-PDU with T-SDU payload & GeoAnycast Header
Does a neighbor exist
Buffer GeoBroadcast packet in BC buffer
NO
YES
Is Repetition interval set?
YES
Save the packet in BC buffer
Determine LL_ADDR_NH (Simple GeoBroadcast Forwarding Algo)
Determine function F (x,y)
NO
IDLE
Determine LL_ADDR_NH (Simple GeoBroadcast Forwarding Algo)
Setup a retransmit timer
Execute Greedy Forwarding
Determine function F (x,y)
Timer Expires
NO
YES
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH
Execute Media Dependent Procedures
Execute Greedy Forwarding
Is Lifetime Valid?
Update Timer
Execute Media Dependent Procedures
Pass GN-PDU to LL with Dst-Addr the LL_ADDR_NH

Figure 40 : GeoAnycast transmission flowchart

12.6.2.4 GeoUnicast

On reception of a GN-DATA.request primitive with a Packet transport type parameter set to GeoUnicast, the source shall execute the following operations:

1) check whether it has a valid position vector for DE in its LocT:
    a) If no valid position vector information is available, the source shall invoke the location service and omit the execution of further steps. Otherwise, the source shall proceed with step 2;
2) determine the link-layer address LL_ADDR_NH of the next hop:

    a) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the GF algorithm
    b) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 1 (GREEDY), execute the GF algorithm.
    c) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 2 (CBF), execute the CBF algorithm.

3) create a GN-PDU with the T/GN6-SDU as payload and a GeoUnicast packet header:
    a) set the fields of the Common Header.
    b) set the fields of the GeoUnicast Extended Header
4) if LL_ADDR_NH = 0, then buffer the GeoUnicast packet in the UC forwarding packet buffer and omit the execution of further steps;
5) if the optional Repetition interval parameter in the GN-DATA.request parameter is set:
    a) save the GeoUnicast packet;
    b) retransmit the packet with period as specified in Repetition interval until the maximum lifetime of the packet is expired;
6) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
    a) UNSPECIFIED then omit this operation
    b) ITS-G5A then execute the operations.
7) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the LL address of the next hop LL_ADDR_NH.

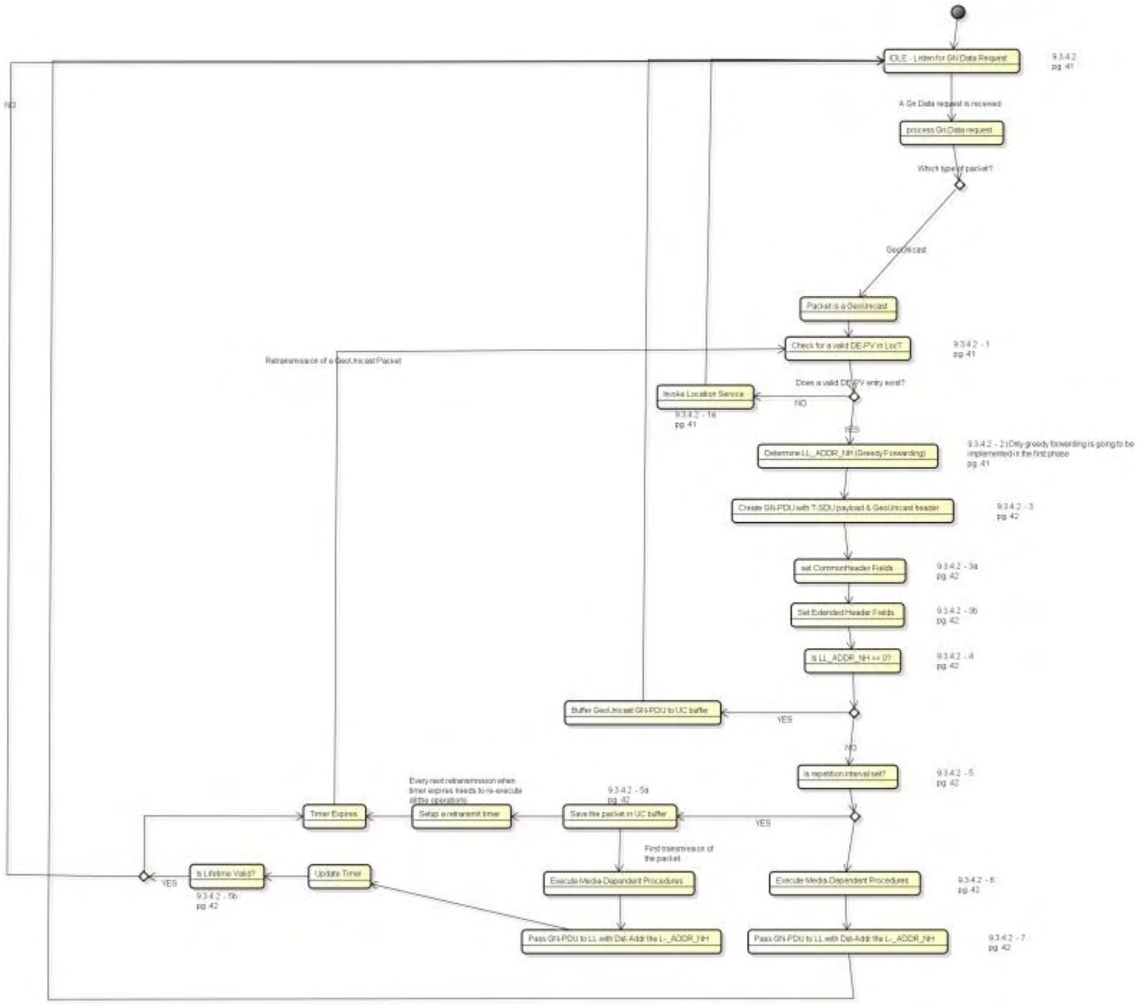

IDLE - Listen for GN Data Request
9.3.4.2
pg. 41
A Gn Data request is received
process Gn Data request
Which type of packet?
GeoUnicast
Packet is a GeoUnicast
Check for a valid DE-PV in LocT
9.3.4.2 - 1
pg. 41
Does a valid DE-PV entry exist?
NO
Invoke Location Service
9.3.4.2 - 1a
pg. 41
YES
Determine LL_ADDR_NH (Greedy Forwarding)
9.3.4.2 - 2 (Only greedy forwarding is going to be implemented in the first phase
pg. 41
Create GN-PDU with T-SDU payload & GeoUnicast header
9.3.4.2 - 3
pg. 42
set CommonHeader Fields
9.3.4.2 - 3a
pg. 42
Set Extended Header Fields
9.3.4.2 - 3b
pg. 42
Is LL_ADDR_NH == 0?
9.3.4.2 - 4
pg. 42
Buffer GeoUnicast GN-PDU to UC buffer
YES
NO
Is repetition interval set?
9.3.4.2 - 5
pg. 42
Retransmission of a GeoUnicast Packet
Every next retransmission when timer expires needs to re-execute all the operations
9.3.4.2 - 5a
pg. 42
Timer Expires
Setup a retransmit timer
Save the packet in UC buffer
YES
First transmission of the packet
NO
YES
Is Lifetime Valid?
9.3.4.2 - 5b
pg. 42
Update Timer
Execute Media-Dependent Procedures
Execute Media-Dependent Procedures
9.3.4.2 - 6
pg. 42
Pass GN-PDU to LL with Dst-Addr the L-_ADDR_NH
Pass GN-PDU to LL with Dst-Addr the L-_ADDR_NH
9.3.4.2 - 7
pg. 42

Figure 41 : GeoUnicast transmission flowchart

12.6.3 Reception/Forwarding of packets

Upon reception of a GeoNetworking packet, the GeoAdhoc STA should decide whether it is the intended recipient of the packet, or will act as a forwarder. Therefore, some common actions are defined for all types of packets, including duplicate packet detection, processing of the GeoNetworking Common Header, flushing buffers and checking if Location Service is pending for the sender GeoNetworking STA. In the following flowchart, the common actions applied upon reception of a GeoNetworking packet are shown:

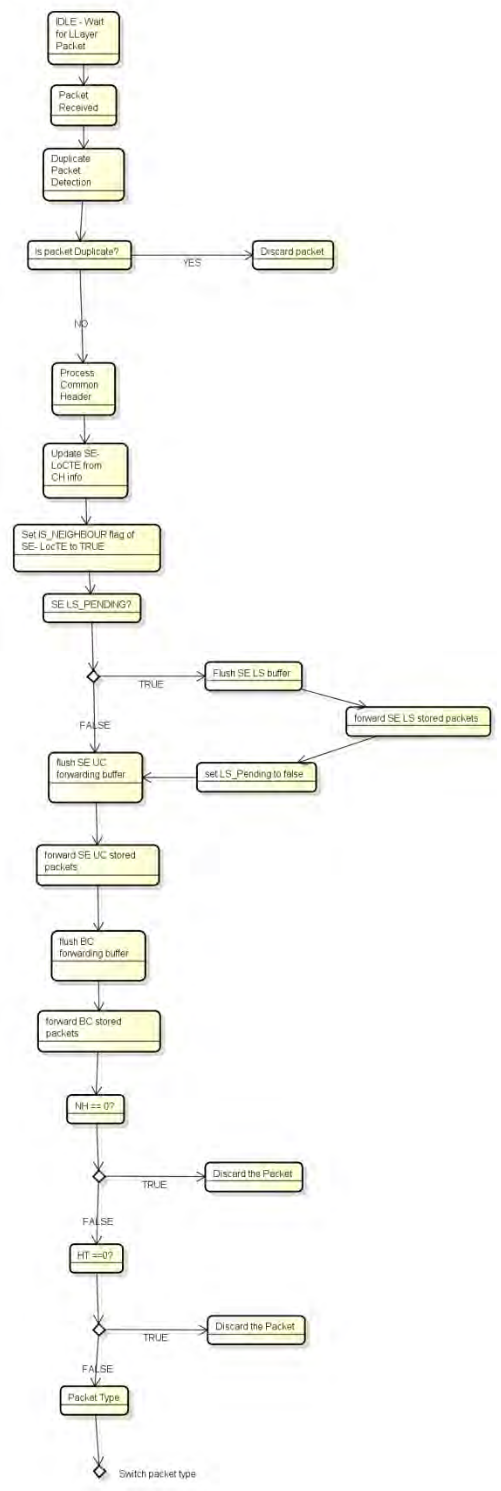

IDLE - Wait for LLayer Packet
Packet Received
Duplicate Packet Detection
Is packet Duplicate?
YES
Discard packet
NO
Process Common Header
Update SE-LoCTE from CH info
Set IS_NEIGHBOUR flag of SE- LocTE to TRUE
SE LS_PENDING?
TRUE
Flush SE LS buffer
forward SE LS stored packets
FALSE
flush SE UC forwarding buffer
set LS_Pending to false
forward SE UC stored packets
flush BC forwarding buffer
forward BC stored packets
NH == 0?
TRUE
Discard the Packet
FALSE
HT ==0?
TRUE
Discard the Packet
FALSE
Packet Type
Switch packet type

Figure 42 : Common actions upon reception of GN packets

12.6.3.1 Single Hop Broadcast / Topologically Scoped Broadcast

SHB packets should not be retransmitted. Therefore, upon reception of an SHB packet, the GeoAdhoc router shall execute the following operations:

1) Common Header processing
2) pass the payload of the GN-PDU to the upper protocol entity by means of a GN-DATA.indication primitive.

In the case of reception of a TSB packet, the GeoAdhoc router shall execute the following operations:

1) Common Header processing
2) execute duplicate packet detection. If the TSB packet is a duplicate, discard the packet and omit the execution of further steps;
3) update the PV(SO) in the LocT with the SO PV fields of the TSB Extended Header.
4) set the IS_NEIGHBOUR(SO) flag to FALSE if SO GN_ADDR does not equal SE GN_ADDR;
5) pass the payload of the GN-PDU to the upper protocol entity by means of a GN-DATA.indication primitive.
6) decrement the value of the HL field by one. If HL is decremented to zero, discard the GN-PDU and omit the execution of following operations:
7) update the fields of the Common Header, i.e.:
    a) the HL field with the decremented HL value.
    b) the SE PV fields with the LPV.
8) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
    a) UNSPECIFIED then omit this operation.
    b) ITS-G5A then execute the operations.
9) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the Broadcast address of the LL entity.

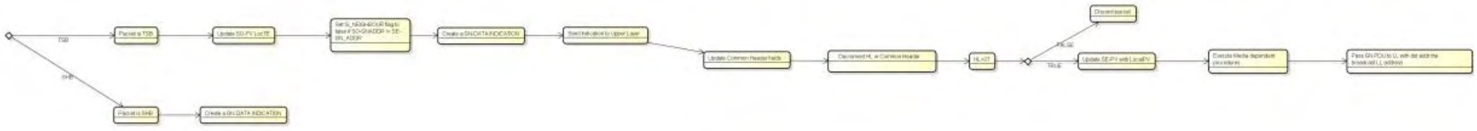

TSB
SHB
Packet is TSB
Update SO-PV LocTE
Set IS_NEIGHBOUR flag to false if SO-GNADDR != SE-GN_ADDR
Create a GN-DATA.INDICATION
Send Indication to Upper Layer
Update Common Header fields
Decrement HL in Common Header
HL<0?
FALSE
TRUE
Discard packet
Update SE-PV with LocalPV
Execute Media dependent procedures
Pass GN PDU to LL with dst addr the broadcast LL address
Packet is SHB
Create a GN DATA INDICATION

Figure 43 : TSB/SHB packet reception flowchart

12.6.3.2 GeoBroadcast/GeoAnycast

On reception of a GeoBroadcast packet, the GeoAdhoc router shall execute the following operations:

1) Common Header processing.
2) execute duplicate packet detection. If the GeoBroadcast packet is a duplicate, discard the packet and omit the execution of further steps.
3) update the PV(SO) in the LocT with the SO PV fields of the GeoBroadcast Extended Header.

4) set the IS_NEIGHBOUR(SO) flag to FALSE if SO GN_ADDR does not equal SE GN_ADDR;
5) determine the link-layer address LL_ADDR_NH of the next hop.
   a) if the MIB attribute itsGnGeoBroadcastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the Simple GeoBroadcast with line forwarding algorithm.
   b) if the MIB attribute itsGnGeoBroadcastForwardingAlgorithm is set to 1 (SIMPLE), execute the Simple GeoBroadcast with line forwarding algorithm.
6) if LL_ADDR_NH = 0, then buffer the GeoBroadcast packet in the BC forwarding packet buffer and omit the execution of further steps
7) if F (x, y) ≥ 0 (GeoAdhoc router is inside or at the border of the geographical area) pass the payload of the GN-PDU to the upper protocol entity by means of a GN-DATA.indication primitive with the parameter settings.
8) decrement the value of the HL field by one; if HL is decremented to zero, discard the GN-PDU and omit the execution of following operations;
9) update the fields of the Common Header, i.e.:
   a) the HL field with the decremented HL value;
   b) the SE PV fields with the LPV.
10) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
   a) UNSPECIFIED then omit this operation.
   b) ITS-G5A then execute the operations.
11) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the LL address of the next hop LL_ADDR_NH.

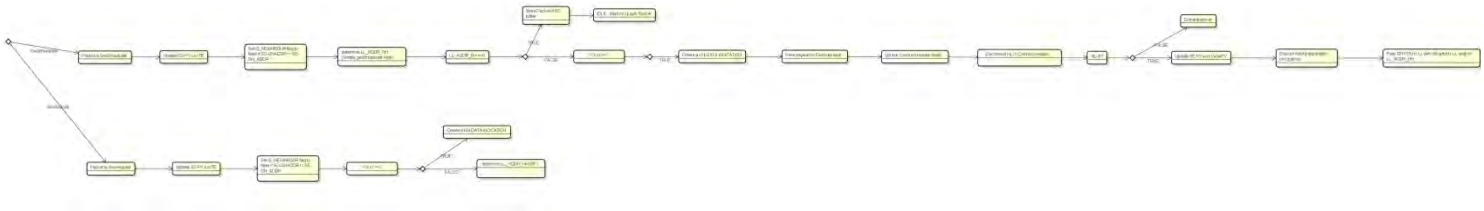


Figure 44 : GeoBroadcast/GeoAnycast packet reception flowchart

On reception of a GeoAnycast packet, the GeoAdhoc router shall execute the following operations:

1) Common Header processing.
2) execute duplicate packet detection. If the GeoAnycast packet is a duplicate, discard the packet and omit the execution of further steps.
3) update the PV(SO) in the LocT with the SO PV fields of the GeoAnycast Extended Header.
4) set the IS_NEIGHBOUR(SO) flag to FALSE if SO GN_ADDR does not equal SE GN_ADDR;
5) determine function F(x,y).
   a) if F (x, y) < 0 (GeoAdhoc router is outside the geographical area) and the MIB attribute itsGnGeoAreaLineForwarding is set to TRUE, execute the GeoUnicast forwarding algorithm and determine the link-layer address LL_ADDR_NH of the next hop.
      i) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the GF algorithm.
      ii) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 1 (GREEDY), execute the GF algorithm.

        iii) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 2 (CBF), execute the CBF algorithm.
    b) if F (x, y) ≥ 0 (GeoAdhoc router is inside or at the border of the geographical area) pass the payload of the GN-PDU to the upper protocol entity by means of a GN-DATA.indication primitive.

12.6.3.3 GeoUnicast

In the case of GeoUnicast packet reception, the GeoAdhoc STA decides whether it is the recipient of a forwarder for the packet. Different cases are defined for the two cases.

12.6.3.3.1 ITS STA is the destination

On reception of a GeoUnicast packet, the GeoAdhoc router shall check the GN_ADDR field in the DE PV of the GeoUnicast packet header. If this address does not match its GN_ADDR, the GeoAdhoc router shall execute the following operations:

1) Common Header processing.
2) execute duplicate packet detection. If the GeoUnicast packet is a duplicate, discard the packet and omit the execution of further steps;
3) update the PV(SO) in the LocT with the SO PV fields of the GeoUnicast Extended Header.
4) set the IS_NEIGHBOUR(SO) flag to FALSE if SO GN_ADDR does not equal SE GN_ADDR.
5) flush packet buffers (SO LS packet buffer, SO UC forwarding packet buffer):
    a) if LS_pending(SO) is TRUE:
        (1) flush the SO LS packet buffer.
        (2) forward the stored packets.
        (3) set LS_pending(SO) to false.
    b) if the UC forwarding packet buffer for SO is not empty, flush the UC forwarding buffer and forward the stored packets;
6) update the DE PV(DE) in the LocT with DE PV fields in the GeoUnicast Extended Header.
7) update the fields of the Common Header, i.e.:
    a) the HL field with the decremented HL value.
    b) the SE PV fields with the LPV.
8) update the DE PV fields with the PV(DE) in the LocT.
9) decrement the value of the HL field by one; if HL is decremented to zero, discard the GN-PDU and omit the execution of further steps;
10) determine the link-layer address LL_ADDR_NH of the next hop.
    a) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 0 (UNSPECIFIED), execute the GF algorithm.
    b) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 1 (GREEDY), execute the GF algorithm.
    c) if the MIB attribute itsGnGeoUnicastForwardingAlgorithm is set to 2 (CBF), execute the CBF algorithm.

11) if LL_ADDR_NH = 0, then buffer the GeoUnicast packet in the UC forwarding packet buffer and omit the execution of further steps.
12) execute media-dependent procedures: if the Communication profile parameter of the GN-DATA.request primitive is set to:
    a) UNSPECIFIED then omit this operation;
    b) ITS-G5A then execute the operations.
13) pass the GN-PDU to the LL protocol entity via the IN interface and set the destination address to the LL address of the next hop LL_ADDR_NH.

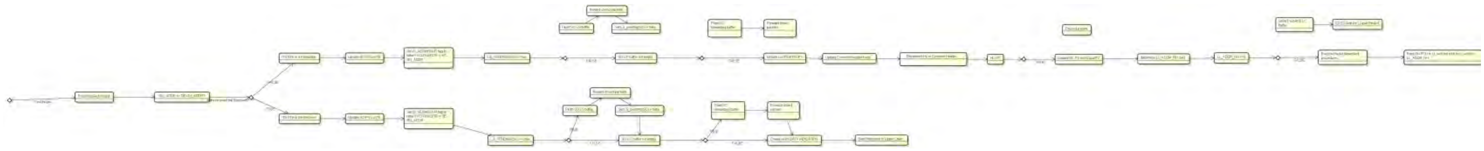


Figure 45 : GeoUnicast packet reception flowchart

12.6.3.3.2 ITS STA is a Forwarder

On reception of a GeoUnicast packet, the GeoAdhoc router shall check the GN_ADDR field in the DE PV of the GeoUnicast packet header. If this address matches its GN_ADDR, the GeoAdhoc router shall execute the following operations:

1) Common Header processing.
2) execute duplicate packet detection. If the GeoUnicast packet is a duplicate, discard the packet and omit the execution of further steps.
3) update the PV(SO) in the LocT with the SO PV fields of the GeoUnicast Extended Header.
4) set the IS_NEIGHBOUR(SO) flag to FALSE if SO GN_ADDR does not equal the SE GN_ADDR.
5) flush packet buffers (SO LS packet buffer, SO UC forwarding packet buffer):
    a) if LS_pending(SO) is TRUE:
        i. flush the SO LS packet buffer.
        ii. forward the stored packets.
        iii. set LS_pending(SO) to false.
    b) if the UC forwarding packet buffer for SO is not empty, flush the UC forwarding buffer and forward the stored packets.
6) pass the payload of the GN-PDU to the upper protocol entity by means of a GN-DATA.indication primitive.

12.7 Forwarding Algorithms

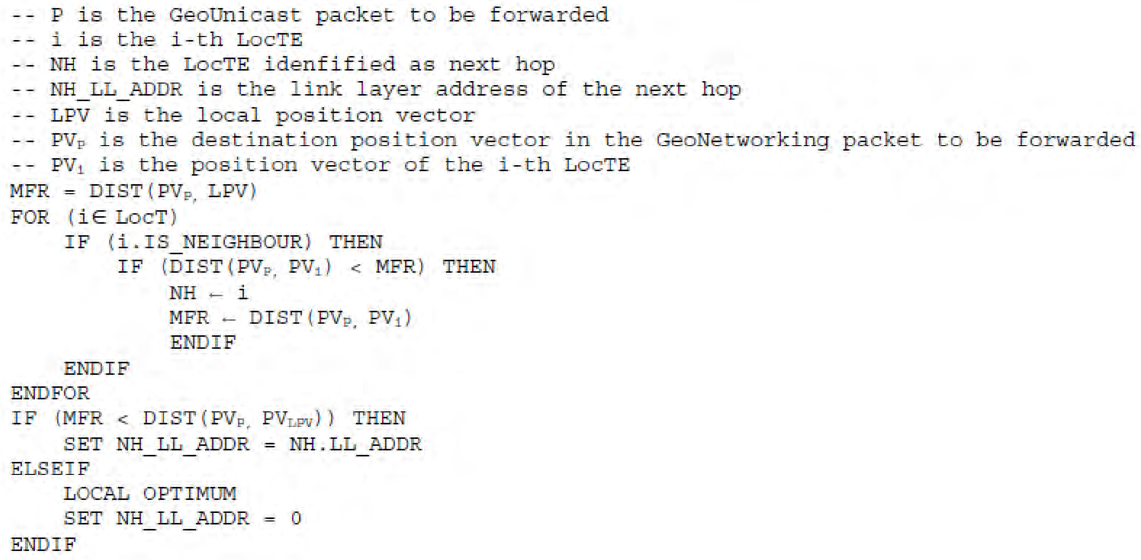
Although in the standard defined by ETSI several forwarding algorithms are defined in the case of taking a routing decision, in order to keep our implementation as simple as possible we only implemented one for GeoUnicast transmissions and one for GeoBroadcast/Anycast packets. The algorithms are the “Greedy Forwarding” (GF) and the “Simple GeoBroadcast Algorithm”, that is dependent of the former.

12.7.1 Greedy Forwarding Algorithm

With the Greedy Forwarding (GF) algorithm, the GeoAdhoc router uses the location information of the destination carried in the GeoUnicast packet header and selects one of the neighbors as the next hop.

The algorithm applies the most forward within radius (MFR) policy, which selects the neighbor with the smallest geographical distance to the destination, thus providing the greatest progress when the GeoUnicast packet is forwarded.

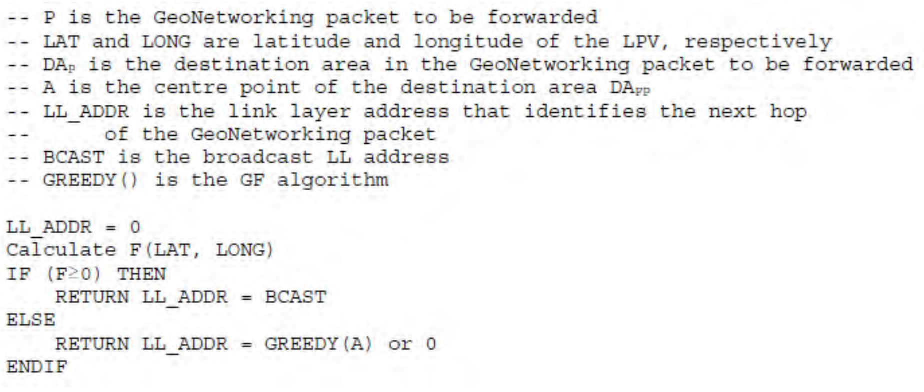
If no neighbor with greater progress than the local GeoAdhoc router exists, the packet has reached a local optimum.



```
-- P is the GeoUnicast packet to be forwarded
-- i is the i-th LocTE
-- NH is the LocTE idenfified as next hop
-- NH_LL_ADDR is the link layer address of the next hop
-- LPV is the local position vector
-- $PV_P$ is the destination position vector in the GeoNetworking packet to be forwarded
-- $PV_1$ is the position vector of the i-th LocTE
MFR = DIST($PV_P$, LPV)
FOR (i∈ LocT)
    IF (i.IS_NEIGHBOUR) THEN
        IF (DIST($PV_P$, $PV_1$) < MFR) THEN
            NH ← i
            MFR ← DIST($PV_P$, $PV_1$)
            ENDIF
    ENDIF
ENDFOR
IF (MFR < DIST($PV_P$, $PV_{LPV}$)) THEN
    SET NH_LL_ADDR = NH.LL_ADDR
ELSEIF
    LOCAL OPTIMUM
    SET NH_LL_ADDR = 0
ENDIF
```

12.7.2 Simple GeoBroadcast Algorithm

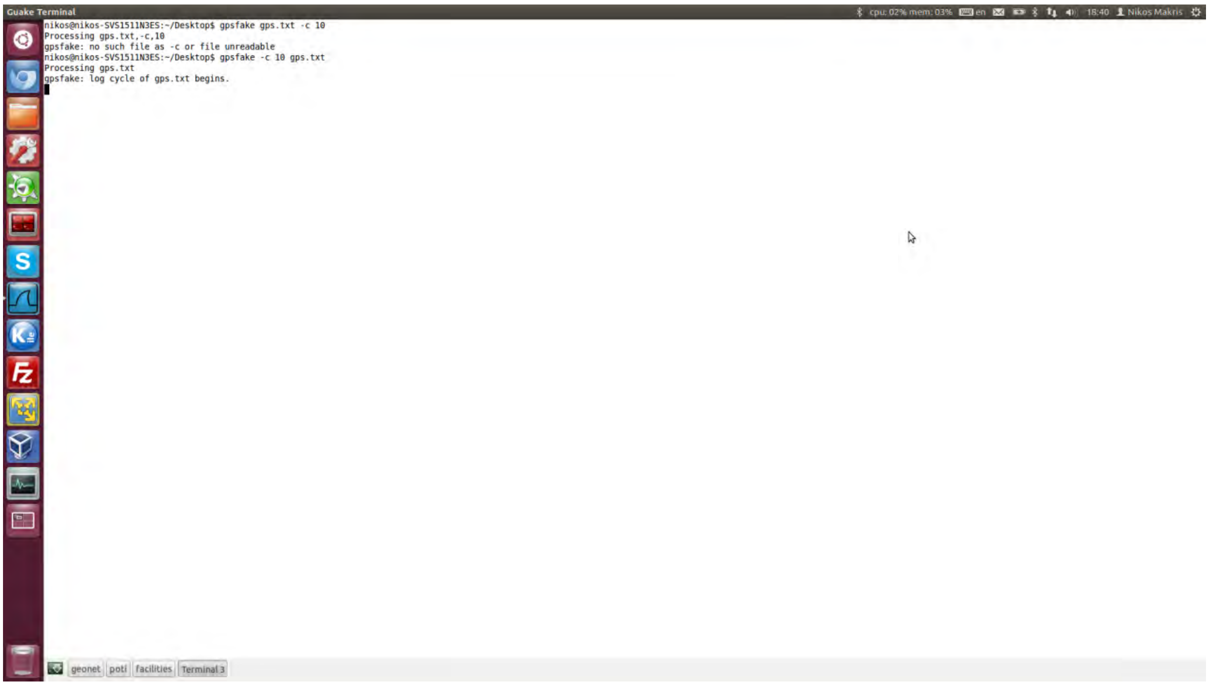
The algorithm utilizes the function F(x,y) in order to determine whether the GeoAdhoc router is located inside, at the border or outside the geographical target area carried in the GeoBroadcast packet header. If the GeoAdhoc router is inside or at the border of the area, the packet shall be re-broadcasted. If it is outside the area, the packet shall be forwarded by the GF algorithm.



```
-- P is the GeoNetworking packet to be forwarded
-- LAT and LONG are latitude and longitude of the LPV, respectively
-- $DA_P$ is the destination area in the GeoNetworking packet to be forwarded
-- A is the centre point of the destination area $DA_{PP}$
-- LL_ADDR is the link layer address that identifies the next hop
--       of the GeoNetworking packet
-- BCAST is the broadcast LL address
-- GREEDY() is the GF algorithm

LL_ADDR = 0
Calculate F(LAT, LONG)
IF (F≥0) THEN
    RETURN LL_ADDR = BCAST
ELSE
    RETURN LL_ADDR = GREEDY(A) or 0
ENDIF
```

13. Evaluation of the GeoNetworking Daemon

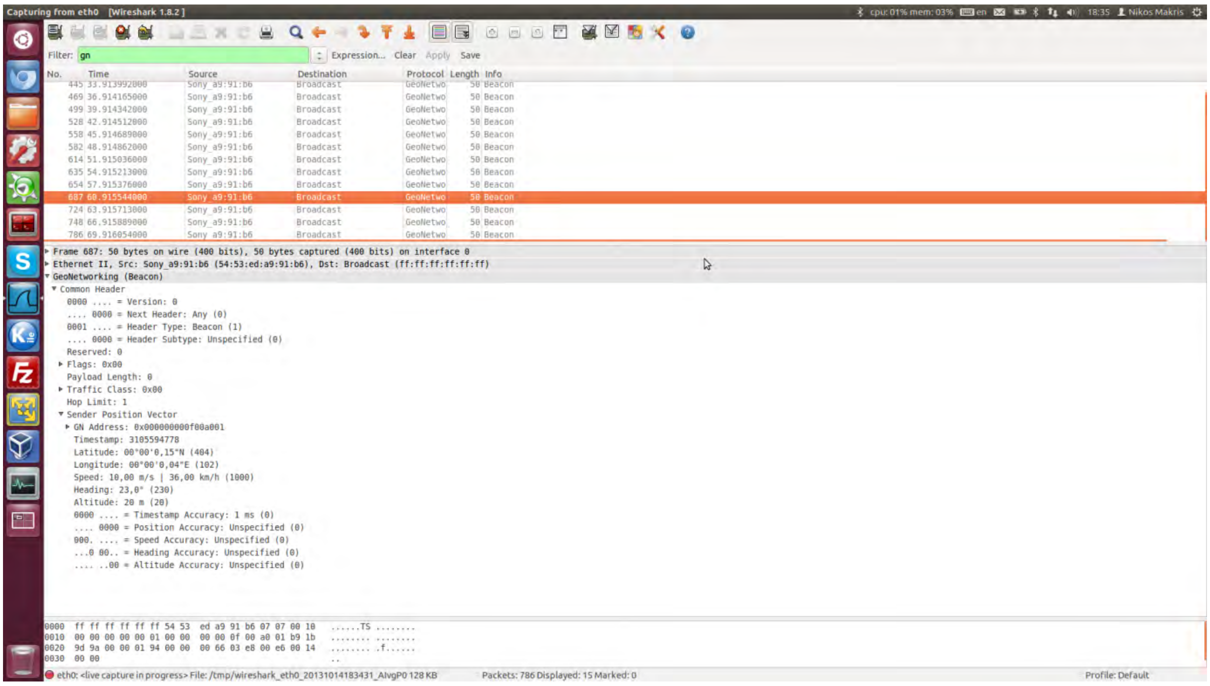
For the evaluation of the overall platform, we engaged three of the NITOS nodes, setup with the tools described. In order to emulate mobility we used the gpsfake application, as we can see in Figure 46.



```
nikos@nikos-SVS1511N3ES:~/Desktop$ gpsfake gps.txt -c 10
Processing gps.txt,-c,10
gpsfake: no such file as -c or file unreadable
nikos@nikos-SVS1511N3ES:~/Desktop$ gpsfake -c 10 gps.txt
Processing gps.txt
gpsfake: log cycle of gps.txt begins.
```

Figure 46 : Starting the gpsfake application

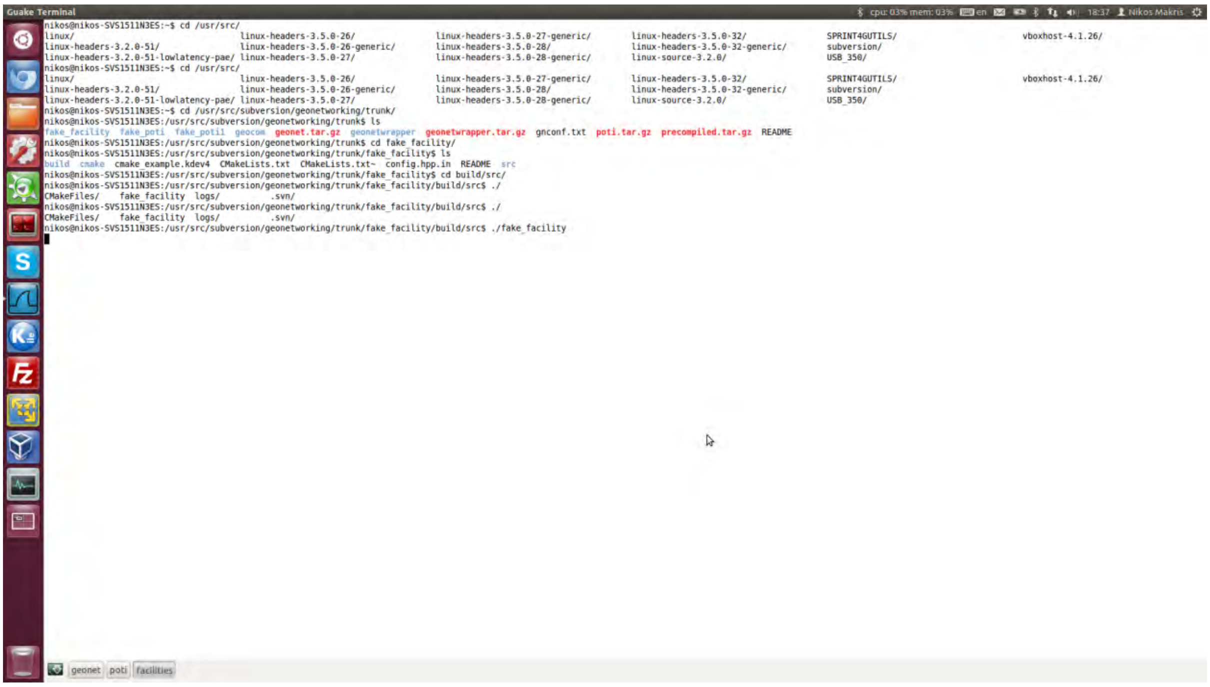
Once the GeoNetworking daemon starts, it prints out the MID that will be used in its GNADDR. By using the Wireshark packet sniffer, with ITS dissectors installed, we observe that every three seconds a beacon packet is sent, and correctly received on the receiver side.



```
Filter: gn

No.   Time          Source          Destination   Protocol  Length  Info
  445 33.913992000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  469 36.914165000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  499 39.914342000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  528 42.914512000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  558 45.914689000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  582 48.914862000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  614 51.915036000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  635 54.915213000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  654 57.915376000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  687 60.915544000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  724 63.915713000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  748 66.915889000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon
  786 69.916054000  Sony_a9:91:b6   Broadcast     GeoNetwo      50  Beacon

▸ Frame 687: 50 bytes on wire (400 bits), 50 bytes captured (400 bits) on interface 0
▸ Ethernet II, Src: Sony_a9:91:b6 (54:53:ed:a9:91:b6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ GeoNetworking (Beacon)
  ▾ Common Header
      0000 .... = Version: 0
      .... 0000 = Next Header: Any (0)
      0001 .... = Header Type: Beacon (1)
      .... 0000 = Header Subtype: Unspecified (0)
      Reserved: 0
    ▸ Flags: 0x00
      Payload Length: 0
    ▸ Traffic Class: 0x00
      Hop Limit: 1
    ▾ Sender Position Vector
      ▸ GN Address: 0x000000000f00a001
        Timestamp: 3105594778
        Latitude: 00°00'0,15"N (404)
        Longitude: 00°00'0,04"E (102)
        Speed: 10,00 m/s | 36,00 km/h (1000)
        Heading: 23,0° (230)
        Altitude: 20 m (20)
        0000 .... = Timestamp Accuracy: 1 ms (0)
        .... 0000 = Position Accuracy: Unspecified (0)
        000. .... = Speed Accuracy: Unspecified (0)
        ...0 00.. = Heading Accuracy: Unspecified (0)
        .... ..00 = Altitude Accuracy: Unspecified (0)

0000  ff ff ff ff ff ff 54 53  ed a9 91 b6 07 07 00 10   ......TS ........
0010  00 00 00 00 00 01 00 00  00 00 0f 00 a0 01 b9 1b   ........ ........
0020  9d 9a 00 00 01 94 00 00  00 66 03 e8 00 e6 00 14   ........ .f......
0030  00 00                                              ..

eth0: <live capture in progress> File: /tmp/wireshark_eth0_20131014183431_AlvgP0 128 KB    Packets: 786 Displayed: 15 Marked: 0
```

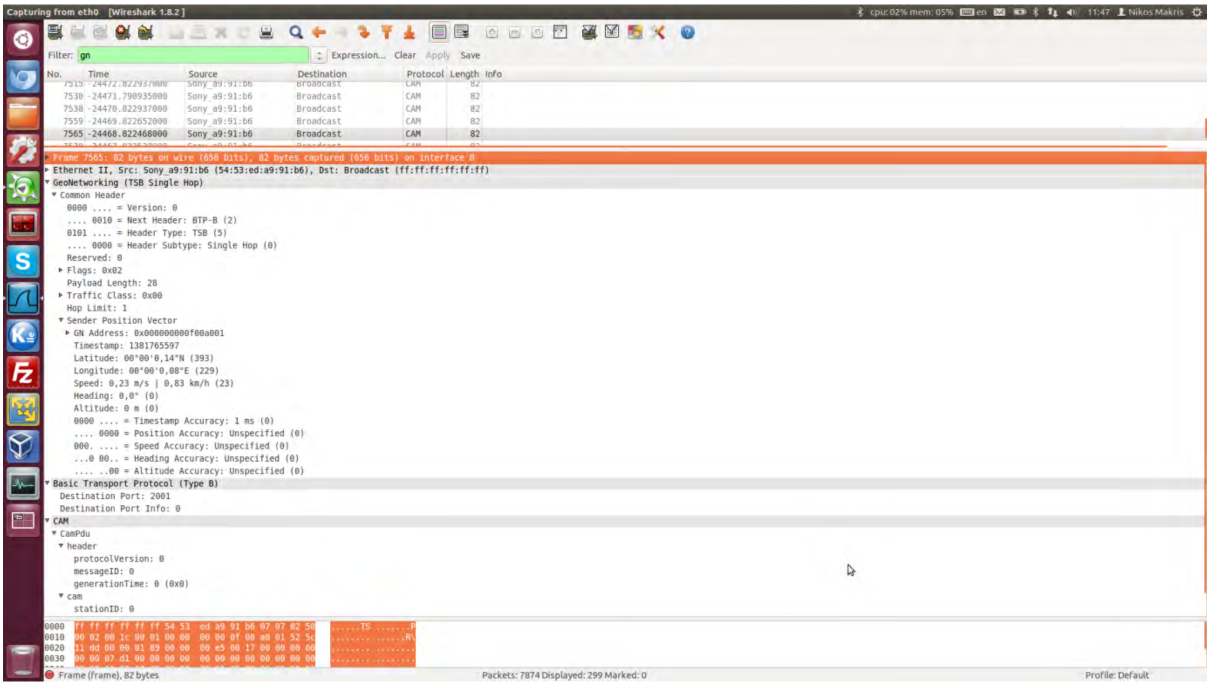
Figure 47 : Transmission of Beacons

If we initialize the facility layer, that is used to create CAM packets every one second, we see that packets are correctly encapsulated in SHB packets, with a BTP-A transport protocol.



```
nikos@nikos-SVS1511N3ES:~$ cd /usr/src/
linux/                             linux-headers-3.5.0-26/            linux-headers-3.5.0-27-generic/    linux-headers-3.5.0-32/            SPRINT4GUTILS/                     vboxhost-4.1.26/
linux-headers-3.2.0-51/            linux-headers-3.5.0-26-generic/    linux-headers-3.5.0-28/            linux-headers-3.5.0-32-generic/    subversion/
linux-headers-3.2.0-51-lowlatency-pae/ linux-headers-3.5.0-27/        linux-headers-3.5.0-28-generic/    linux-source-3.2.0/                USB_350/
nikos@nikos-SVS1511N3ES:~$ cd /usr/src/
linux/                             linux-headers-3.5.0-26/            linux-headers-3.5.0-27-generic/    linux-headers-3.5.0-32/            SPRINT4GUTILS/                     vboxhost-4.1.26/
linux-headers-3.2.0-51/            linux-headers-3.5.0-26-generic/    linux-headers-3.5.0-28/            linux-headers-3.5.0-32-generic/    subversion/
linux-headers-3.2.0-51-lowlatency-pae/ linux-headers-3.5.0-27/        linux-headers-3.5.0-28-generic/    linux-source-3.2.0/                USB_350/
nikos@nikos-SVS1511N3ES:~$ cd /usr/src/subversion/geonetworking/trunk/
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk$ ls
fake_facility  fake_poti  fake_poti1  geocom  geonet.tar.gz  geonetwrapper  geonetwrapper.tar.gz  gnconf.txt  poti.tar.gz  precompiled.tar.gz  README
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk$ cd fake_facility/
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk/fake_facility$ ls
build  cmake  cmake_example.kdev4  CMakeLists.txt  CMakeLists.txt~  config.hpp.in  README  src
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk/fake_facility$ cd build/src/
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk/fake_facility/build/src$ ./
CMakeFiles/    fake_facility  logs/          .svn/
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk/fake_facility/build/src$ ./
CMakeFiles/    fake_facility  logs/          .svn/
nikos@nikos-SVS1511N3ES:/usr/src/subversion/geonetworking/trunk/fake_facility/build/src$ ./fake_facility
█
```

Figure 48 : Starting facility layer



```
No.   Time              Source           Destination   Protocol Length Info
7515  -24472.822937000  Sony_a9:91:b6    Broadcast     CAM      82
7530  -24471.790935000  Sony_a9:91:b6    Broadcast     CAM      82
7538  -24470.822937000  Sony_a9:91:b6    Broadcast     CAM      82
7559  -24469.822652000  Sony_a9:91:b6    Broadcast     CAM      82
7565  -24468.822468000  Sony_a9:91:b6    Broadcast     CAM      82
```

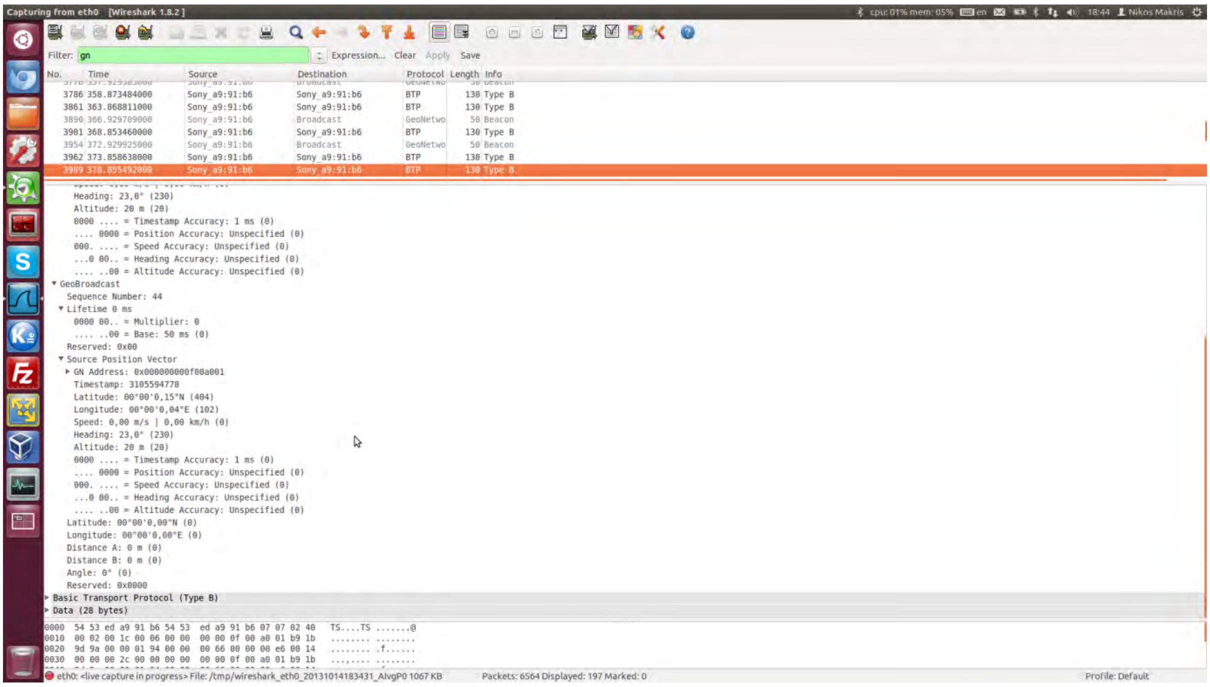
```
▸ Frame 7565: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
▸ Ethernet II, Src: Sony_a9:91:b6 (54:53:ed:a9:91:b6), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ GeoNetworking (TSB Single Hop)
  ▾ Common Header
     0000 .... = Version: 0
     .... 0010 = Next Header: BTP-B (2)
     0101 .... = Header Type: TSB (5)
     .... 0000 = Header Subtype: Single Hop (0)
     Reserved: 0
   ▸ Flags: 0x02
     Payload Length: 28
   ▸ Traffic Class: 0x00
     Hop Limit: 1
   ▾ Sender Position Vector
     ▸ GN Address: 0x000000000f00a001
       Timestamp: 1381765597
       Latitude: 00°00'0,14"N (393)
       Longitude: 00°00'0,08"E (229)
       Speed: 0,23 m/s | 0,83 km/h (23)
       Heading: 0,0° (0)
       Altitude: 0 m (0)
       0000 .... = Timestamp Accuracy: 1 ms (0)
       .... 0000 = Position Accuracy: Unspecified (0)
       000. .... = Speed Accuracy: Unspecified (0)
       ...0 00.. = Heading Accuracy: Unspecified (0)
       .... ..00 = Altitude Accuracy: Unspecified (0)
▾ Basic Transport Protocol (Type B)
   Destination Port: 2001
   Destination Port Info: 0
▾ CAM
  ▾ CamPdu
    ▾ header
       protocolVersion: 0
       messageID: 0
       generationTime: 0 (0x0)
    ▾ cam
       stationID: 0
```

```
0000  ff ff ff ff ff ff 54 53  ed a9 91 b6 07 07 02 50   ......TS .......P
0010  00 02 00 1c 00 01 00 00  00 00 0f 00 a0 01 52 5c   ........ ......R\
0020  11 dd 00 00 01 89 00 00  00 e5 00 17 00 00 00 00   ........ ........
0030  00 00 07 d1 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Frame (frame), 82 bytes   Packets: 7874 Displayed: 299 Marked: 0   Profile: Default

Figure 49 : Capturing CAM messages

Once we change the facility layer, to send fake DENM messages, aiming for a Geobroadcast packet, we get the following packet transmission (Figure 50):



```
No.   Time            Source          Destination     Protocol  Length Info
3786  358.873484000   Sony_a9:91:b6   Sony_a9:91:b6   BTP       130 Type B
3861  363.868811000   Sony_a9:91:b6   Sony_a9:91:b6   BTP       130 Type B
3890  366.929709000   Sony_a9:91:b6   Broadcast       GeoNetwo   50 Beacon
3901  368.853460000   Sony_a9:91:b6   Sony_a9:91:b6   BTP       130 Type B
3954  372.929925000   Sony_a9:91:b6   Broadcast       GeoNetwo   50 Beacon
3962  373.858638000   Sony_a9:91:b6   Sony_a9:91:b6   BTP       130 Type B
3989  378.855492000   Sony_a9:91:b6   Sony_a9:91:b6   BTP       130 Type B
```

```
       Heading: 23,0° (230)
       Altitude: 20 m (20)
       0000 .... = Timestamp Accuracy: 1 ms (0)
       .... 0000 = Position Accuracy: Unspecified (0)
       000. .... = Speed Accuracy: Unspecified (0)
       ...0 00.. = Heading Accuracy: Unspecified (0)
       .... ..00 = Altitude Accuracy: Unspecified (0)
  ▾ GeoBroadcast
     Sequence Number: 44
   ▾ Lifetime 0 ms
       0000 00.. = Multiplier: 0
       .... ..00 = Base: 50 ms (0)
     Reserved: 0x00
   ▾ Source Position Vector
     ▸ GN Address: 0x000000000f00a001
       Timestamp: 3105594778
       Latitude: 00°00'0,15"N (404)
       Longitude: 00°00'0,04"E (102)
       Speed: 0,00 m/s | 0,00 km/h (0)
       Heading: 23,0° (230)
       Altitude: 20 m (20)
       0000 .... = Timestamp Accuracy: 1 ms (0)
       .... 0000 = Position Accuracy: Unspecified (0)
       000. .... = Speed Accuracy: Unspecified (0)
       ...0 00.. = Heading Accuracy: Unspecified (0)
       .... ..00 = Altitude Accuracy: Unspecified (0)
     Latitude: 00°00'0,00"N (0)
     Longitude: 00°00'0,00"E (0)
     Distance A: 0 m (0)
     Distance B: 0 m (0)
     Angle: 0° (0)
     Reserved: 0x0000
▸ Basic Transport Protocol (Type B)
▸ Data (28 bytes)
```

```
0000  54 53 ed a9 91 b6 54 53  ed a9 91 b6 07 07 02 40   TS....TS .......@
0010  00 02 00 1c 00 06 00 00  00 00 0f 00 a0 01 b9 1b   ........ ........
0020  9d 9a 00 00 01 94 00 00  00 66 00 00 00 e6 00 14   ........ .f......
0030  00 00 00 2c 00 00 00 00  00 00 0f 00 a0 01 b9 1b   ...,.... ........
```

eth0: <live capture in progress> File: /tmp/wireshark_eth0_20131014183431_AlvgP0 1067 KB   Packets: 6564 Displayed: 197 Marked: 0   Profile: Default

Figure 50 : GeoBroadcast packet transmission

14. Testbed Related extensions

14.1 NITOS testbed

For the evaluation of the Geonetworking implementation we used the NITOS [2] Future Internet (FI) testbed and its corresponding control and management framework, OMF. NITOS (Network Implementation Testbed using Open Source code) FI is a key testbed in European research on Future Internet Research and Experimentation (FIRE) projects. It is a heterogeneous platform, where novel protocols and ideas can be evaluated under real conditions. The main experimental components of NITOS are the following:

1. A wireless experimentation testbed, which consists of powerful nodes (some of them mobile), that feature multiple wireless interfaces and allow for experimentation with heterogeneous (WiFi, Bluetooth, ZigBee) wireless technologies. NITOS has recently extended to a meso-scale testbed, by acquiring a WiMAX Base Station and two LTE small cells and by also enabling WiMAX/LTE connectivity to the wireless nodes. NITOS's 4G network spans throughout the metropolitan city of Volos.
2. A software defined radio (SDR) testbed that consists of Universal Software Radio Peripheral (USRP) devices attached to the NITOS wireless nodes. USRPs allow the researcher to program a number of physical layer features (e.g. modulation), thereby enabling dedicated PHY layer or cross-layer research.
3. A Software Defined Networking (SDN) testbed that consists of multiple OpenFlow technology enabled switches, connected to the NITOS nodes, thus enabling experimentation with switching and routing networking protocols. Experimentation using the OpenFlow technology can be combined with the wireless networking one, hence enabling the construction of more heterogeneous experimental scenarios.
4. A testbed for conducting video-transmission (wired or wireless) related experimentation, which consists of high definition digital cameras, mounted on the NITOS nodes. This component can be combined with the wired (OpenFlow) and wireless testbeds mentioned above, enabling the study of video transmission over heterogeneous communication technologies.
5. A distributed Wireless Sensor Network (WSN) testbed able to sense and gather environmental measurements from agricultural installations. The deployed facility consists of multiple clusters, each one comprised of wireless sensor devices and Gateway nodes, creating mesh networks utilizing the ZigBee technology. Measurements that are gathered by the sensor network are aggregated, stored and processed at a centralized cluster of servers, which controls the irrigating system. The distributed wireless sensor infrastructure is located in the agricultural installation of UTH and enables experimentation on agricultural development by sensing environmental conditions as well as controlling multiple parameters on the agricultural process.

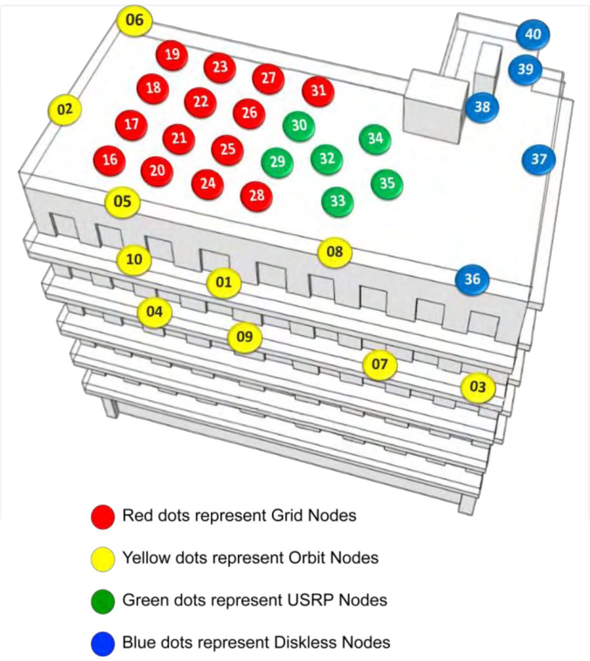

06
19
23
27
31
40
39
18
22
26
38
02
17
21
30
34
16
25
29
32
37
20
24
28
35
05
33
08
10
01
36
04
09
07
03
Red dots represent Grid Nodes
Yellow dots represent Orbit Nodes
Green dots represent USRP Nodes
Blue dots represent Diskless Nodes

Figure 51: NITOS testbed deployment

NITOS testbed is open to the research community 24/7 and it is remotely accessible through the NITOS reservation tool. Parallel experimentation of different users is enabled, through the utilization of the NITOS scheduler software. The testbed is based on open-source software that allows the design and implementation of new algorithms, enabling new functionalities on the existing hardware. Though OMF (cOntrol and Management Framework), NITOS supports evaluation of protocols and applications under real world settings. It is also designed to achieve reproducibility of experimentation though the CONCRETE (CONtrol and Classify REpeatable Testbed Experiments) tool developed in UTH.

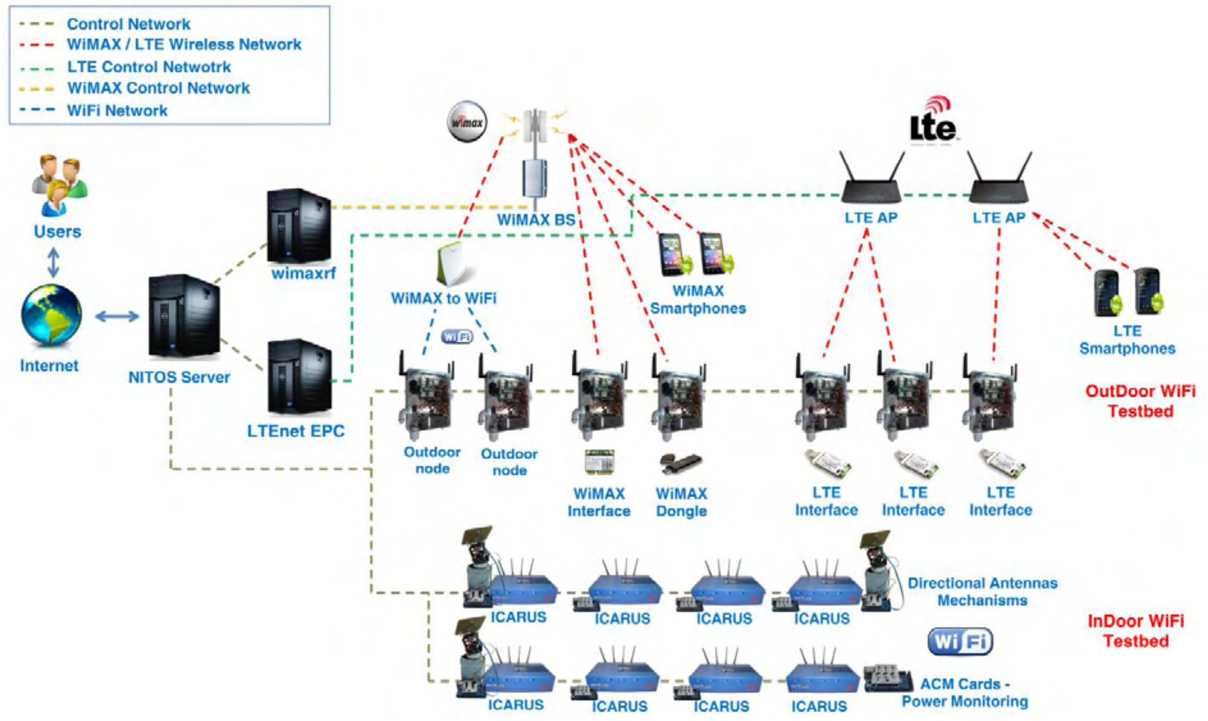


Figure 52 : NITOS testbed architecture

14.2 OMF Framework

OMF [36] is a free open-source testbed control framework that manages over 20 testbeds worldwide, including the NITOS testbed in Europe and six WiMAX meso-scale deployments in the US, along with the ORBIT facility at Rutgers University, the largest openly accessible wireless testbed facility in the world, for which OMF was originally developed in 2003. It provides an API for the experimenter to conduct experiments in a controlled manner, by using a simple experiment description written in OMF Experiment Description Language (OEDL), based on the Ruby language. OMF allows for organized control of a testbed’s resources (turning on/off the resources, loading an image on a node’s disk, saving the image, etc.) by using services running on an entity called Aggregate Manager (AM).

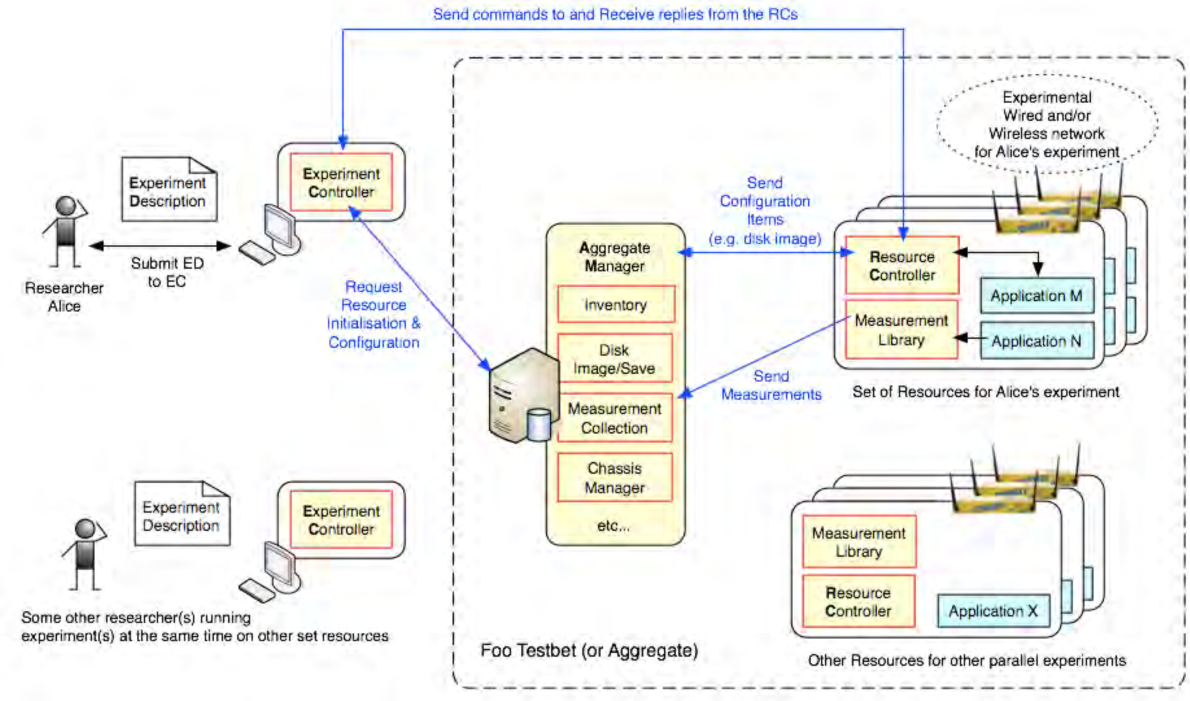

Send commands to and Receive replies from the RCs
Experimental Wired and/or Wireless network for Alice's experiment
Experiment Description
Experiment Controller
Researcher Alice
Submit ED to EC
Request Resource Initialisation & Configuration
Send Configuration Items (e.g. disk image)
Aggregate Manager
Inventory
Disk Image/Save
Measurement Collection
Chassis Manager
etc...
Resource Controller
Measurement Library
Application M
Application N
Send Measurements
Set of Resources for Alice's experiment
Experiment Description
Experiment Controller
Some other researcher(s) running experiment(s) at the same time on other set resources
Measurement Library
Resource Controller
Application X
Foo Testbet (or Aggregate)
Other Resources for other parallel experiments

Figure 53 : OMF architecture

Moreover, two more entities are vital for the effective use of the framework: The OMF Resource Controller (RC) and the Experiment Controller (EC). The RC is actually controlling the testbed's resources, and issues the appropriate commands needed for the effective setup of an experiment. The Experiment controller is used for the parsing the experiment's parameters and sending them to the RC. The communication among all the entities of an experiment takes place over an XMPP server based on the Openfire tool [37] installed on the NITOS's portal.

It provides an integrated measurement and instrumentation framework, as well powerful user tools and a portal that supports the entire experiment lifecycle.

14.3 OML Measurement Library

OMF's accompanying library is called OML (OMF Measurement Library). It exploits a client-server architecture and is used to send aggregated measurements from an application to a remote server side in an organized way. The server part of OML handles the measurements and passes them in a database file, to which the experimenter has access. OML can be used without using OMF, wherever an application creates measurements over a network. There are two ways of instrumenting an application with OML library:

- When we have the source code of the application, by using the open source libraries for OML

- By writing a wrapper around an existing application and parsing its output.

For the first case, implementations of the OML library exist for C/C++ language, Ruby and Python.

14.3.1 Instrumenting GeoNetworking with OML support

For our case, since we had the source code of our application, we used the OML implementation for C/C++. The measurements we receive from the GeoNetworking application are actually the reported messages written to the logs of the application. Therefore, the experimenter that executes an experiment with GeoNetworking can have access to a database, containing the sequence number of each packet transmitted and if this was correctly received.

14.4 Integrating GeoNetworking on the NITOS testbed

In order to provide the GeoNetworking protocol as an experimentation tool in the NITOS FI, we set up an image containing the following:

- The gps-clients and gps package from the Ubuntu repositories
- Two files containing prerecorded routes in the NMEA language
- OMF/OML support
- The `gpsfake` application
- gpsmm library.

Using the GPSfake application, the experimenter can emulate mobility conditions on the static NITOS node. Gpsfake application parses the NMEA sentences and feeds them into gpsd (gps-daemon) that runs on the node. Gpsd’s API provides us a JSON file with the current position of the “moving” object. This output is parsed by our POTI implementation, which then sends the data to the GeoNetworking daemon. Packets are sent every time they are triggered by our Facilities layer implementation.

15. Future Work

In this master thesis we have developed and evaluated an implementation of ETSI TS 102 636-4-1 standard for Intelligent Transport Systems. Current implementation can provide full GeoNetworking capability to a GeoAdhoc Station, with logging support and OML Measurement Points sent to a remote server. The following topics are identified as future research topics on this implementation:

- Performance evaluation of our implementation with other Open Source available, such as the CarGeo implementation [38]
- Evaluation of the implementation under real world settings. It is notable that in the context of the REDUCTION EU – FP7 project, DELPHI has offered to evaluate our protocol in a wide scale deployment on city buses at Nicosia, Cyprus.
- Extended measurements involving multiple transmission technologies, apart from Ethernet and IEEE 802.11 p, such as the IEEE 802.16e - WiMAX deployment at NITOS, using the Click platform [39].

16. References

[1] ETSI TS 102 636-4-1: "Intelligent Transportation System (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Subpart 1: Media independent functionalities".

[2] NITOS testbed, NITLab Laboratory, University of Thessaly – http://nitlab.inf.uth.gr

[3] Delphi Automotive LLP – http://www.delphi.com

[4] Takagi, H.; Kleinrock, L. (March 1984). "Optimal transmission ranges for randomly distributed packet radio terminals". IEEE Transactions on Communications 32 (3): 246–257.

[5] Finn, Gregory G. (March 1987). Routing and Addressing Problems in Large Metropolitan-Scale Internetworks, University of Southern California, ISI/RR-87-180

[6] Stojmenovic, Ivan (2002). "Position based routing in ad hoc networks". IEEE Communications Magazine 40 (7): 128–134.

[7] Stojmenovic, Ivan; Lin, Xu (2001). "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks". IEEE Transactions on Parallel and Distributed Systems 12 (10): 1023–1032.

[8] T.-C. Hou and V. O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," IEEE Trans. Commun., vol. 34, no. 1, Jan. 1986, pp. 38–44.

[9] M. Mauve, J. Widmer and H. Hartenstein, A Survey on Position Based Routing in Mobile Ad-hoc Networks, IEEE Network Magazine, 15(6):30–39, November 2001.

[10] E. Kranakis, H. Singh, and J. Urrutia, "Compass Routing on Geometric Networks," Proc. 11th Canadian Conf. Comp. Geo., Vancouver, Aug. 1999.

[11] ETSI EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture"

[12] IETF RFC 5213: "Proxy Mobile IPv6".

[13] ITU-R Recommendation M.687-2: "International Mobile Telecommunications 2000 (IMT-2000)".

[14] ETSI EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture"

[15] ETSI TS 102 636-3: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 3: Network architecture"

[16] IETF RFC 3753: "Mobility Related Terminology".

[17] IETF RFC 2460: "Internet Protocol, Version 6 (IPv6) Specification"

[18] IETF RFC 3775: "Mobility Support in IPv6".

[19] IETF RFC 3963: "Network Mobility (NEMO) Basic Support Protocol".

[20] IETF RFC 791: "Internet Protocol"

[21] ETSI TS 102 636-5-1: “Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol”

[22] IETF RFC 768: "User Datagram Protocol"

[23] IETF RFC 793: "Transmission Control Protocol"

[24] IEEE 802.11:2010: “IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments”

[25] IEEE 802.11:2007: "IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[26] ETSI EN 302 571 (V1.1.1): "Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive".

[27] ETSI EN 301 893 (V1.5.1): "Broadband Radio Access Networks (BRAN); 5 GHz high performance RLAN; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive".

[28] ECC/DEC(02)01: "ECC Decision of 15 March 2002 on the frequency bands to be designated for the co-ordinated introduction of Road Transport and Traffic Telematic Systems".

[29] Commission Decision 2008/671/EC of 5 August 2008 on the harmonised use of radio spectrum in the 5 875-5 905 MHz frequency band for safety-related applications of Intelligent Transport Systems (ITS).

[30] ECC/REC/(08)01: "ECC Recommendation (08)01 on the use of the band 5855-5875 MHz for Intelligent Transport Systems (ITS)".

[31] ERC/DEC(99)23: "ERC Decision of 29 November 1999 on the harmonised frequency bands to be designated for the introduction of High Performance Radio Local Area Networks (HIPERLANs)".

[32] Commission Decision 2005/513/EC of 11 July2005 on the harmonised use of radio spectrum in the 5 GHz frequency band for the implementation of wireless access systems including radio local area networks (WAS/RLANs).

[33] Commission Decision 2007/90/EC of 12 February 2007 amending Decision 2005/513/EC on the harmonized use of radio spectrum in the 5 GHz frequency band for the implementation of Wireless Access Systems including Radio Local Area Networks (WAS/RLANs).

[34] ETSI TS 102 687 (V1.1.1): "Intelligent Transport Systems (ITS); Transmitter Power Control Mechanism for Intelligent Transport Systems operating in the 5 GHz range"

[35] OSGi™ - The Dynamic Module System for Java – http://www.osgi.org

[36] cOntrol and Management Framework (OMF) – http://omf.mytestbed.net

[37] Openfire - Ignite Realtime: Openfire Server – http://www.igniterealtime.org/projects/openfire/

[38] An Open Source GeoNetworking Implementation - http://hal.inria.fr/hal-00760724/

[39] Click Modular Router - http://www.read.cs.ucla.edu/click