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

ICT-2007.6.1: ICT for intelligent
vehicles and mobility services

GeoNet STREP N°216269

D4.1 Conformance Test Plan and Results

DATE	31 May 2010
CONTRACTUAL DATE OF DELIVERY TO THE EC	M21 – October 2009
ACTUAL DATE OF DELIVERY TO THE EC	M23 – December 2009
EDITOR, COMPANY	Lajos Nagy, BroadBit
WORKPACKAGE	WP4 Conformance Testing
DOCUMENT CODE	GeoNet-D.4.1_final_revised-1.1
SECURITY	Public

DOCUMENT HISTORY

Release	Date	Reason of change	Status	Distribution
0.1	09/11/2009	Lajos Nagy: Initial version	Draft	Internal
0.2	09/12/2009	Andras Kovacs: Text update	Draft	Internal
0.3	21/12/2009	Carsten Schulze: 1st review	Review	Internal
0.4	2010-01-20	Lajos Nagy: Test results updated	Review	Public
1.0	2010-05-07	Andras Kovacs: Finalisation of test descriptions	Final	Public
1.0.1	2010-05-31	Andras Kovacs: update of terminology and follow-up test standardisation	Update	Public
1.0.1	2010-06-10	Thierry Ernst: Brush up and compliance with other GeoNet deliverables - review	Update	Internal
1.0.2	2010-06-17	Lajos Nagy: Review comments addressed	Update	Internal
1.1	2010-06-28	Thierry Ernst: Final quality check	Final	EC

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

Contents

1. Executive Summary

The GeoNet project aims at providing a standard solution for IPv6 GeoNetworking to all Intelligent Transportation Systems (ITS), relying on the IPv6 standard and on Car-to-Car Communication Consortium's (C2C-CC) GeoNetworking. The primary objective of the conformance testing reported in the present document is to validate that the functional behaviour of the GeoNet implementations of IPv6 GeoNetworking [GeoNetD3.1] exactly conforms to the IPv6 GeoNetworking reference specification [GeoNetD2.2]. For proper understanding, the reader is assumed to be familiar with the GeoNet architecture (see Figure 1) and the terminology presented in [GeoNetD1.2].

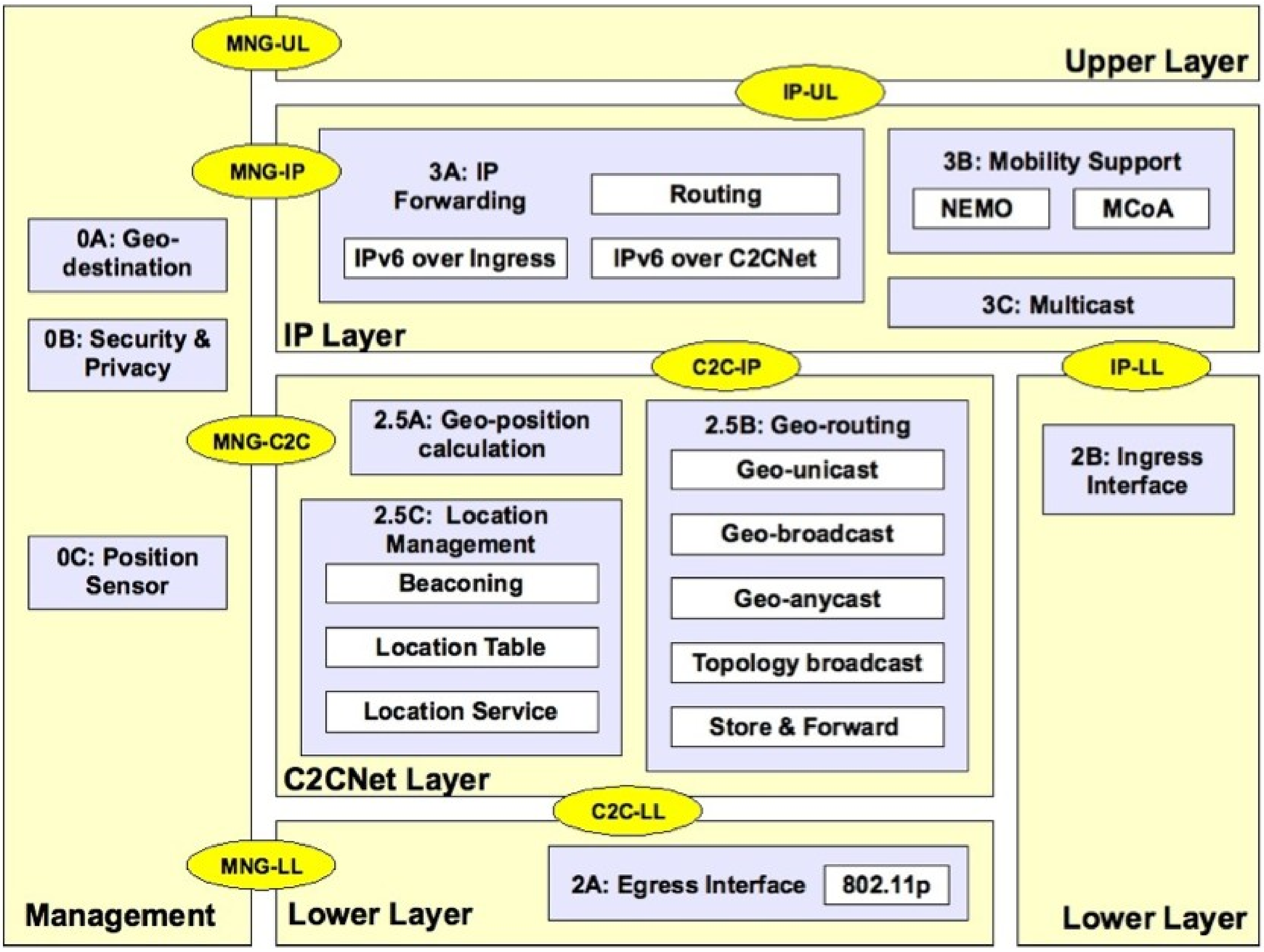


Figure 1: GeoNet Architecture - Functional Modules

Conformance testing or 'type testing' is a testing procedure to determine whether a system meets some specified standard functionality. This document describes the conformance testbed set-up, test case details, and test results. The system under test means the two reference implementations of the C2CNet layer (developed by Hitachi and NEC) and the implementation of the IPv6 over C2CNet sub-module (developed by INRIA). The implementation of the PositionSensor module (developed by Lesswire) is tested to ensure that the received implementation-relevant position data is valid and conforms to the specification.

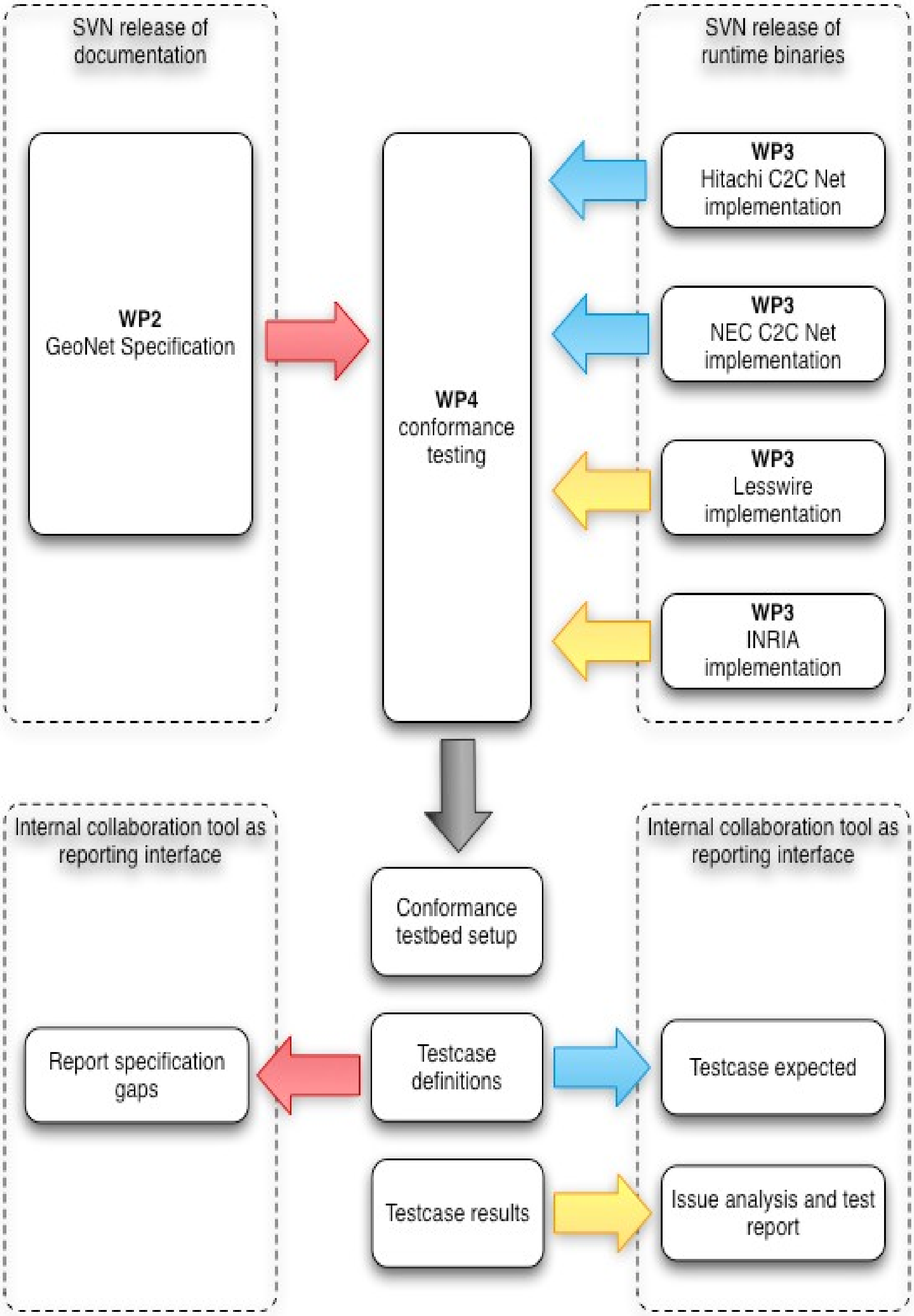

SVN release of documentation
WP2
GeoNet Specification
WP4
conformance testing
SVN release of runtime binaries
WP3
Hitachi C2C Net implementation
WP3
NEC C2C Net implementation
WP3
Lesswire implementation
WP3
INRIA implementation
Internal collaboration tool as reporting interface
Conformance testbed setup
Report specification gaps
Testcase definitions
Testcase results
Internal collaboration tool as reporting interface
Testcase expected
Issue analysis and test report

Figure 2: GeoNet Conformance Testing Workflow

2. Structure of the Document

The present document is structured as follows:

- Definition of Test Purposes
- Test cases of C2CNet implementation
- Test cases of PositionSensor implementation
- Test reports
- Conformance test results
- Annex A: Contributors - lists contributors of this document
- Annex B: Terminology - lists the terms and acronyms used in this document
- Annex C: References: lists all the references presented in this document

3. Definition of Test Purposes

The purpose of GeoNet implementation conformance testing is to determine to what extent multiple – Hitachi, NEC and Lesswire - implementations of a particular standard conforms to the individual requirements of that standard. DESCRIPTION [ETSI-TS-102-636-3] contains the standardisation methodology of protocol and profile conformance testing specifications. ETSI conformance testing uses the principles in the ISO/IEC 9646 series as a basis, but not as strict guidelines, for developing conformance test specifications.

The components of conformance testing specifications are Test Purposes, the Abstract Test Method/Architecture, Abstract Test Cases, the Abstract Test Suite and its Test Suite Structure, and Implementation Conformance Statements (ICS).

3.1 Purpose

Conformance Test Purpose is a 'prose description' of a well-defined testing objective focusing on a single requirement or a set of requirements as specified in the base requirements or another appropriate specification. GeoNet Test Purposes have mandatory and optional elements that vary according to the type of testing.

Test purpose is a non-formal high-level description of a test. This test description can be used as the basis for a formal test specification (e.g., Abstract Test Suite in TTCN).

For each of the testing in the test groups described, valid behaviour and/or invalid behaviour test cases are performed.

Expected behaviour statuses:

- **Valid Behaviour** (BV) tests: Group or Subgroup provides testing verifying that the IUT reacts in conformity with the specified functionality.

- **Invalid Behaviour** (BI) tests: Group or Subgroup provides testing verifying that the IUT reacts in conformity with the specified functionality, after receipt of a syntactically, semantically or functionality invalid event.

3.2 TP Name Convention

Test purposes are numbered, starting at 01, within each group and they are organised according to the TSS. Additional references are added to identify the actual test suite. TP name means a unique identification of each test purposes of an implementation in test environment.

IUT: Implementation Under Test

Name	Values
<**IUT**> (type of IUT)	C2CNet_Hitachi, C2CNet_NEC
<**Main Functionality**>	TB: Transmitted beacons
<**Access Method**>	UDP: UPD port, IP6: IPv6
<**Test Group**>	BV: Valid Behaviour, BI: Invalid Behaviour
<*nn*>	sequential number (01-99)

Example format:

- **C2CNet_Hitachi/TB/UDP/BV/01** - testing Hitachi implementation of C2CNet layer by UDP in group Valid Behaviour (BV) with number 01

3.3 Definition Conventions

The test purposes are following particular rules as defined here:

- **TP ID**: The Test Purpose (TP) ID is a unique identifier what is specified according to the TP naming convention defined in previous section;
- **Reference**: Source of testing or testing requirement;
- **Test purpose**: The Test Purpose describes the objectives of the test;
- **Selection criteria**: Expression to select the execution of the test;
- **Comments**: Extra information for this TP.

Example table:

TP ID	C2CNet_Hitachi/TB/UDP/BV/01
Reference	Commonheader.doc from specification
Test purpose	Testing that transmitted Beacons conform to specification and contain correct position sensor data.
Selection criteria	no
Comments	Ticket #16

3.4 Abstract Test Method, Architecture

The Abstract Test Method is the test architecture or configuration. It describes how an GeoNet Implementation Under Test (IUT) is to be tested at a level of abstraction independent of any Means of Testing (MoT), but with enough detail to allow specification of abstract test cases.

3.4.1 Specification as Source

These initial and final GeoNet specification [GeoNetD2.1, GeoNetD2.2] have been used as baseline for designing conformance test cases. Altogether, the designed test cases verify all relevant aspects of the protocol behaviour.

3.4.2 Conformance Test Cycle

Each test case has been re-executed at every update of the GeoNet modules. This test cycle is illustrated in Figure 3.

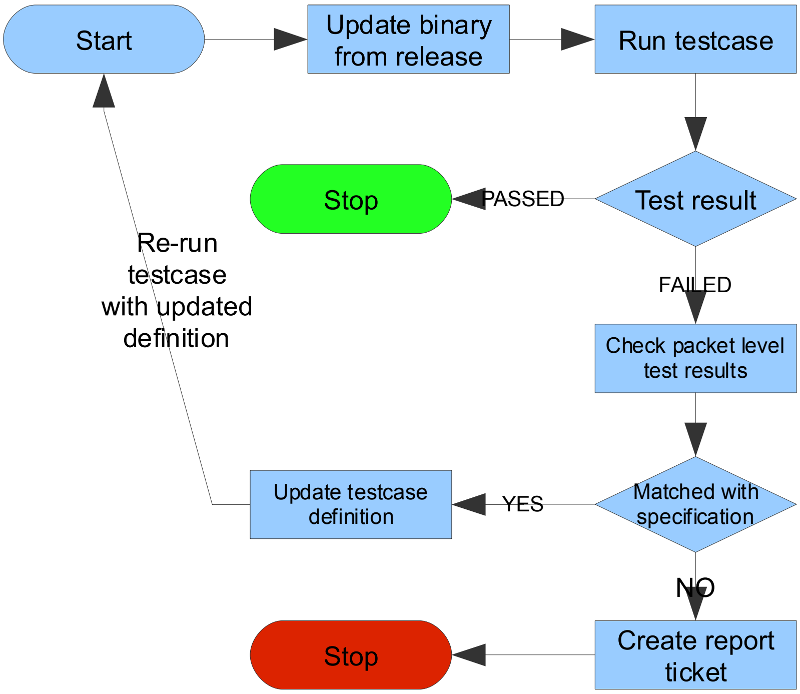

Start
Update binary from release
Run testcase
Stop
PASSED
Test result
Re-run testcase with updated definition
FAILED
Check packet level test results
Update testcase definition
YES
Matched with specification
NO
Stop
Create report ticket

Figure 3: Conformance Test Cycle

3.4.3 Conformance Test platforms and Tools

Runtime environment:

- OS: Linux, kernel: 2.6.28-15-generic;
- wireless device: Card with “Atheros” chipset;
- GPS device simulator.

We used BTT (BroadBit Test Tool) as the test system. BTT is a comprehensive TTCN-3 tool for conformance and protocol testing, it supports both Tabular and ASN.1 based protocol definitions. Main features of the utilised test platform include:

- Support of basic TTCN-3 syntax;
- Support of both tabular and ASN.1 based definitions;
- Easy mapping between port definitions and port handlers;
- Port handlers included for generic sockets, such as Ethernet, UDP, or Unix sockets;
- Platform independent, open source.

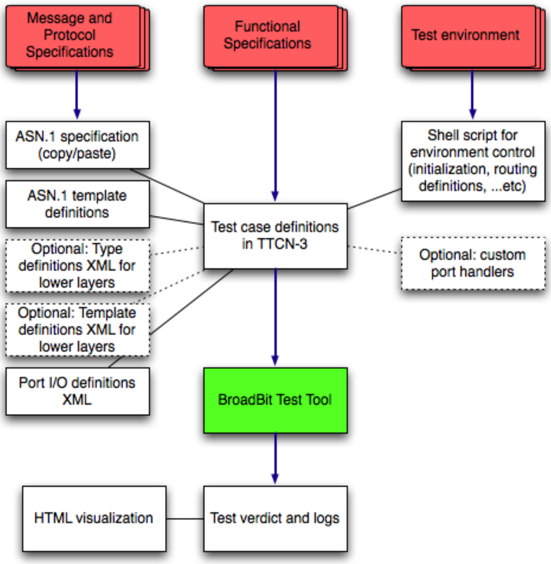

Message and Protocol Specifications
Functional Specifications
Test environment
ASN.1 specification (copy/paste)
ASN.1 template definitions
Optional: Type definitions XML for lower layers
Optional: Template definitions XML for lower layers
Port I/O definitions XML
Test case definitions in TTCN-3
Shell script for environment control (initialization, routing definitions, ...etc)
Optional: custom port handlers
BroadBit Test Tool
HTML visualization
Test verdict and logs

Figure 4: BroadBit Test Tool Architecture Diagram for ASN.1-based Applications

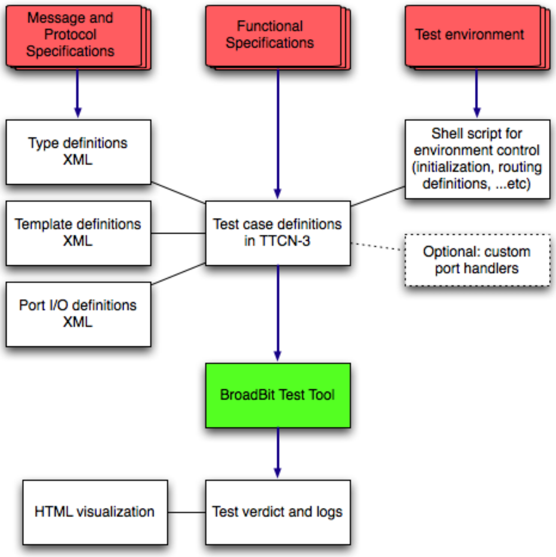

Message and Protocol Specifications
Functional Specifications
Test environment
Type definitions XML
Template definitions XML
Port I/O definitions XML
Test case definitions in TTCN-3
Shell script for environment control (initialization, routing definitions, ...etc)
Optional: custom port handlers
BroadBit Test Tool
HTML visualization
Test verdict and logs

Figure 5: BroadBit Test Tool Architecture Diagram

3.4.4 Conformance Test Environment Set-Up

Conformance Test environment requires PHP command-line (“php5-cli”) package installed. The current stable PHP version is 5.2, above this version, there are deprecated function warnings (relating to split function).

Set environment variable "GEONET_ROOT" to your local repository location (without ending “/”) before starting test scripts, example:

- $ export GEONET_ROOT=~/development/svn/geonet

Verify GEONET_ROOT location:

- $ ls $GEONET_ROOT/Binaries
  - wp3hitachi wp3inria wp3lesswire wp3nec wp4broadbit

Conformance test folder structure:

- **wp4broadbit**: store of main scripts of Conformance Test
  - **doc**: documentation part of BTT
  - **packet_manager**: packet generator and parser part of BTT
  - **ports**: network port handlers of BTT
  - **ttcn_compiler**: TTCN compiler part of BTT
  - **modules**: test cases of GeoNet modules
    - **c2cnw-hitachi**: numbered test cases for Hitachi implementation of GeoNet C2CNW
    - **c2cnw-nec**: numbered test cases for NEC implementation of GeoNet C2CNW
    - **positionsensor**: numbered test cases of PositionSensor implementation

Implementation module and test case folder structure:

- **modules**
  - *module name* (example: c2cnw-hitachi)
    - **common_begin.sh**: begin of common test script parts for module

- **common_end.sh**: end of common test script parts for module
- **common_mnpp_begin.sh**: begin of common MNPP test script parts for module
- **common_mnpp_end.sh**: end of common MNPP test script parts for module
- **test.sh**: main test script to run all test cases, and print results
- *test case number* (example: 01, 02, ...)
  - **ttcn**: test cases and scripts
    - **test.sh**: startup script for test case (run from this directory)
    - **.ttcn*: TTCN description of test case
    - **port_definitions.xml**: simulated port definitions
    - **packet_description.xml**: protocol description
    - **packet_description_templates.xml**: templates to fill protocol with data
  - **log**: test result log files
    - **ttcn.log**: main log with generated and parsed packets
    - **tcpdump.log**: network packet monitoring log file
    - **strace.log**: kernel debugger log file
  - **visualization**: test result HTML files, TTCN visualization of test case
    - **index.html**: main HTML file of visualization result

3.4.5 Test environment for Hitachi Implementation

Sources of binary implementation, including application and configuration files:

- https://geonet-project.eu/geonet/browser/Binaries/wp3hitachi/final

Documentation of binary implementation:

- https://geonet-project.eu/geonet/browser/Binaries/wp3hitachi/final/README

Test environment sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi

Test environment and start-up flow:

- Initialise BTT environment;
- Initialise GEONET environment;
- Set-up TTCN compiler environment;
- Set-up LOG files;
- Set-up path to Hitachi implementation sources (GEONET_C2CNW_BINARY);
- Cleanup BTT environment and port handlers;
- Set-up tun device;
- Set-up IPv6 networking and routing parameters;
- Parse and execute TTCN script;
- Execute Hitachi binary with predefined configuration parameters;
- Inject, receive and parse packets;
- Create reports in text and HTML formats.

3.4.6 Test environment for NEC implementation

Sources of binary implementation, including application and configuration files:

- https://geonet-project.eu/geonet/browser/Binaries/wp3nec/final

Documentation of binary implementation:

- https://geonet-project.eu/geonet/browser/Binaries/wp3nec/final/README.txt

Test environment sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec

Test environment and start-up flow:

- Initialise BTT environment;
- Initialise GEONET environment;
- Set-up TTCN compiler environment;
- Set-up LOG files;
- Set-up path to NEC implementation sources (GEONET_C2CNW_BINARY);
- Cleanup BTT environment and port handlers;
- Insert tun module to kernel;
- Execute NEC binary with predefined configuration parameters;
- Set-up IPv6 networking and routing parameters;
- Parse and execute TTCN script;
- Inject, receive and parse packets;
- Create reports in text and HTML formats.

3.4.7 Test Environment for PositionSensor Implementation

Sources of binary implementation, including application and configuration files:

- https://geonet-project.eu/geonet/browser/Binaries/wp3lesswire/PositionSensor/PositionSensor

Documentation of binary implementation:

- https://geonet-project.eu/geonet/browser/Binaries/wp3lesswire/PositionSensor/README

Test environment sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp3lesswire/PositionSensor

Test environment and start-up flow:

- Initialise BTT environment;

- Initialise GEONET environment;
- Set-up TTCN compiler environment;
- Set-up LOG files;
- Start GPS device simulator;
- Cleanup BTT environment and port handlers;
- Parse and execute TTCN script;
- Inject, receive and parse packets;
- Create reports in text and HTML formats.

3.4.8 Example GeoNet TTCN Script

Source: “C2CNWTester.ttcn” TTCN-3 definition file

```
module C2CNWTester {

              type port PositionSensorPort message {
                inout Position_information
              }

              type port LowerLayer message {
                in C2CNW_Unicast_incoming;
                out C2CNW_Unicast_outgoing
              }

              type port UpperLayer message {
                inout IPv6_Packet
              }

              type component EgoVehicleComponent {
                port PositionSensorPort gpsPort
                port LowerLayer lowerLayerPort
                port UpperLayer upperLayerPort
              }

              testcase TestLocationService()
              {
                  timer controlTimer;
                  timer replyTimer;

                  gpsPort.send ( sensor_data );

                  replyTimer.start (5.0);

                  alt {
                   [] lowerLayerPort.receive {
                   }
                   [] controlTimer.timeout {
                    setverdict (fail);
                   }
                  }

                  lowerLayerPort.send ( neighbor_Beacon );

                  upperLayerPort.send ( test_ip_packet );

                  replyTimer.start (5.0);

                  alt {
                           [] lowerLayerPort.receive ( expected_ls_request ) {
                               replyTimer.stop;

                               lowerLayerPort.send ( ls_answer );

                               replyTimer.start (5.0);

                               alt {
                                [] lowerLayerPort.receive ( expected_c2c_unicast ) {
                                setverdict (pass);
                                replyTimer.stop;
                                }
                                 [] replyTimer.timeout {
                                 setverdict (fail);
                                 }
                               }

                           }
                           [] replyTimer.timeout {
                               setverdict (fail);
                           }
                  }

              }

              control {
                           execute (TestLocationService());
              }
}
```

3.4.9 Example GeoNet Packet Definition

Source: “packet_description.xml” as the main protocol description

```xml
<?xml version="1.0" encoding="utf-8"?>
<protocolDescription>

    <PDUs>

        <Position_information>
            <fields>
                <!-- Header -->
                <Type>
                    <bits>8</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Type>

                <Accuracy>
                    <bits>8</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Accuracy>

                <Reserved>
                    <bits>16</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Reserved>

                <Timestamp>
                    <bits>32</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>timestamp</type>
                </Timestamp>

                <Latitude>
                    <bits>32</bits>
                    <endian>little</endian>
                    <signed>true</signed>
                    <type>integer</type>
                </Latitude>

                <Longitude>
                    <bits>32</bits>
                    <endian>little</endian>
                    <signed>true</signed>
                    <type>integer</type>
                </Longitude>

                <Speed>
                    <bits>16</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Speed>

                <Heading>
                    <bits>16</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Heading>

                <Altitude>
                    <bits>16</bits>
                    <endian>little</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Altitude>
            </fields>
        </Position_information>

        <C2CNW_Beacon>
            <fields>
            <!-- SAP 4.7 Header -->
                <SAP47_Type>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </SAP47_Type>
                <SAP47_Channel>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </SAP47_Channel>
                <SAP47_TxPower>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </SAP47_TxPower>
                <SAP47_DataRate>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </SAP47_DataRate>
                <SAP47_Priority>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </SAP47_Priority>
                <SAP47_Reserved1>
                    <bits>24</bits>
                    <type>hexstring</type>
                </SAP47_Reserved1>
                <SAP47_DestinationMAC>
                    <bits>48</bits>
                    <type>hexstring</type>
                </SAP47_DestinationMAC>
                <SAP47_Reserved2>
                    <bits>16</bits>
                    <type>hexstring</type>
                </SAP47_Reserved2>
                <SAP47_SourceMAC>
                    <bits>48</bits>
                    <type>hexstring</type>
                </SAP47_SourceMAC>
                <SAP47_Reserved3>
                    <bits>16</bits>
                    <type>hexstring</type>
                </SAP47_Reserved3>

                <!-- C2C-NET common header -->

                <Version>
                    <bits>4</bits>
                    <type>bitstring</type>
                </Version>

                <Priority>
                    <bits>4</bits>
                    <type>bitstring</type>
                </Priority>

                <HeaderType>
                    <bits>4</bits>
                    <type>bitstring</type>
                </HeaderType>

                <HeaderSubType>
                    <bits>4</bits>
                    <type>bitstring</type>
                </HeaderSubType>

                <TxPower>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </TxPower>

                <Reserved>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </Reserved>

                <PayloadLength>
                    <bits>16</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </PayloadLength>

                <NextHeader>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </NextHeader>

                <HopLimit>
                    <bits>8</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </HopLimit>

                <C2CId>
                    <bits>64</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </C2CId>

                <BeaconTimestamp>
                    <bits>32</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>timestamp</type>
                    <format>millisecond</format>
                </BeaconTimestamp>

                <BeaconLatitude>
                    <bits>32</bits>
                    <endian>big</endian>
                    <signed>true</signed>
                    <type>integer</type>
                </BeaconLatitude>

                <BeaconLongitude>
                    <bits>32</bits>
                    <endian>big</endian>
                    <signed>true</signed>
                    <type>integer</type>
                </BeaconLongitude>

                <BeaconSpeed>
                    <bits>16</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </BeaconSpeed>

                <BeaconHeading>
                    <bits>16</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </BeaconHeading>

                <BeaconAltitude>
                    <bits>16</bits>
                    <endian>big</endian>
                    <signed>false</signed>
                    <type>integer</type>
                </BeaconAltitude>

                <TimAc>
                    <bits>4</bits>
                    <type>bitstring</type>
                </TimAc>

                <PosAc>
                    <bits>4</bits>
                    <type>bitstring</type>
                </PosAc>

                <SAc>
                    <bits>3</bits>
                    <type>bitstring</type>
                </SAc>

                <HAc>
                    <bits>3</bits>
                    <type>bitstring</type>
                </HAc>

                <AAc>
                    <bits>2</bits>
                    <type>bitstring</type>
                </AAc>
            </fields>
        </C2CNW_Beacon>

    </PDUs>

</protocolDescription>
```

3.4.10 Example GeoNet Packet Template Definition

Source: “packet_description_templates.xml” as the content of data to send and expect

```
<?xml version="1.0" encoding="utf-8"?>
<templateDescription>

    <templates>
```

```
        <!-- sensor data packet -->
        <sensor_data>
            <Position_information>
                <Type>0</Type>
                <Accuracy>127</Accuracy>
                <Reserved>0</Reserved>
                <Timestamp>5555</Timestamp>
                <Latitude>1234</Latitude>
                <Longitude>5678</Longitude>
                <Speed>25</Speed>
                <Heading>96</Heading>
                <Altitude>211</Altitude>
            </Position_information>
        </sensor_data>

        <!-- expected beacon packet -->
        <expected_Beacon>
            <C2CNW_Beacon>
                <SAP47_Type>0</SAP47_Type>
                <SAP47_Channel>*</SAP47_Channel>
                <SAP47_TxPower>*</SAP47_TxPower>
                <SAP47_DataRate>*</SAP47_DataRate>
                <SAP47_Priority>*</SAP47_Priority>
                <SAP47_Reserved1>*</SAP47_Reserved1>
                <SAP47_DestinationMAC>*</SAP47_DestinationMAC>
                <SAP47_Reserved2>*</SAP47_Reserved2>
                <SAP47_SourceMAC>*</SAP47_SourceMAC>
                <SAP47_Reserved3>*</SAP47_Reserved3>
                <Version>0001</Version>
                <Priority>0000</Priority>
                <HeaderType>0001</HeaderType>
                <HeaderSubType>0000</HeaderSubType>
                <TxPower>*</TxPower>
                <Reserved>0</Reserved>
                <PayloadLength>*</PayloadLength>
                <NextHeader>0</NextHeader>
                <HopLimit>1</HopLimit>
                <C2CId>*</C2CId>
                <BeaconTimestamp>*</BeaconTimestamp>
                <BeaconLatitude>1234</BeaconLatitude>
                <BeaconLongitude>5678</BeaconLongitude>
                <BeaconSpeed>25</BeaconSpeed>
                <BeaconHeading>96</BeaconHeading>
                <BeaconAltitude>211</BeaconAltitude>
                <TimAc>*</TimAc>
                <PosAc>*</PosAc>
                <SAc>*</SAc>
                <HAc>*</HAc>
                <AAc>00</AAc>
            </C2CNW_Beacon>
        </expected_Beacon>

        <!-- other header type values -->
        <other_header_type>
            <C2CNW_Beacon>
                <SAP47_Type>1..255</SAP47_Type>
                <SAP47_Channel>*</SAP47_Channel>
                <SAP47_TxPower>*</SAP47_TxPower>
                <SAP47_DataRate>*</SAP47_DataRate>
                <SAP47_Priority>*</SAP47_Priority>
                <SAP47_Reserved1>*</SAP47_Reserved1>
                <SAP47_DestinationMAC>*</SAP47_DestinationMAC>
                <SAP47_Reserved2>*</SAP47_Reserved2>
                <SAP47_SourceMAC>*</SAP47_SourceMAC>
                <SAP47_Reserved3>*</SAP47_Reserved3>
                <Version>*</Version>
                <Priority>*</Priority>
                <HeaderType>*</HeaderType>
                <HeaderSubType>*</HeaderSubType>
                <TxPower>*</TxPower>
                <Reserved>*</Reserved>
                <PayloadLength>*</PayloadLength>
                <NextHeader>*</NextHeader>
                <HopLimit>*</HopLimit>
                <C2CId>*</C2CId>
                <BeaconTimestamp>*</BeaconTimestamp>
                <BeaconLatitude>*</BeaconLatitude>
                <BeaconLongitude>*</BeaconLongitude>
                <BeaconSpeed>*</BeaconSpeed>
                <BeaconHeading>*</BeaconHeading>
                <BeaconAltitude>*</BeaconAltitude>
                <TimAc>*</TimAc>
                <PosAc>*</PosAc>
                <SAc>*</SAc>
                <HAc>*</HAc>
                <AAc>*</AAc>
            </C2CNW_Beacon>
        </other_header_type>

    </templates>

</templateDescription>
```

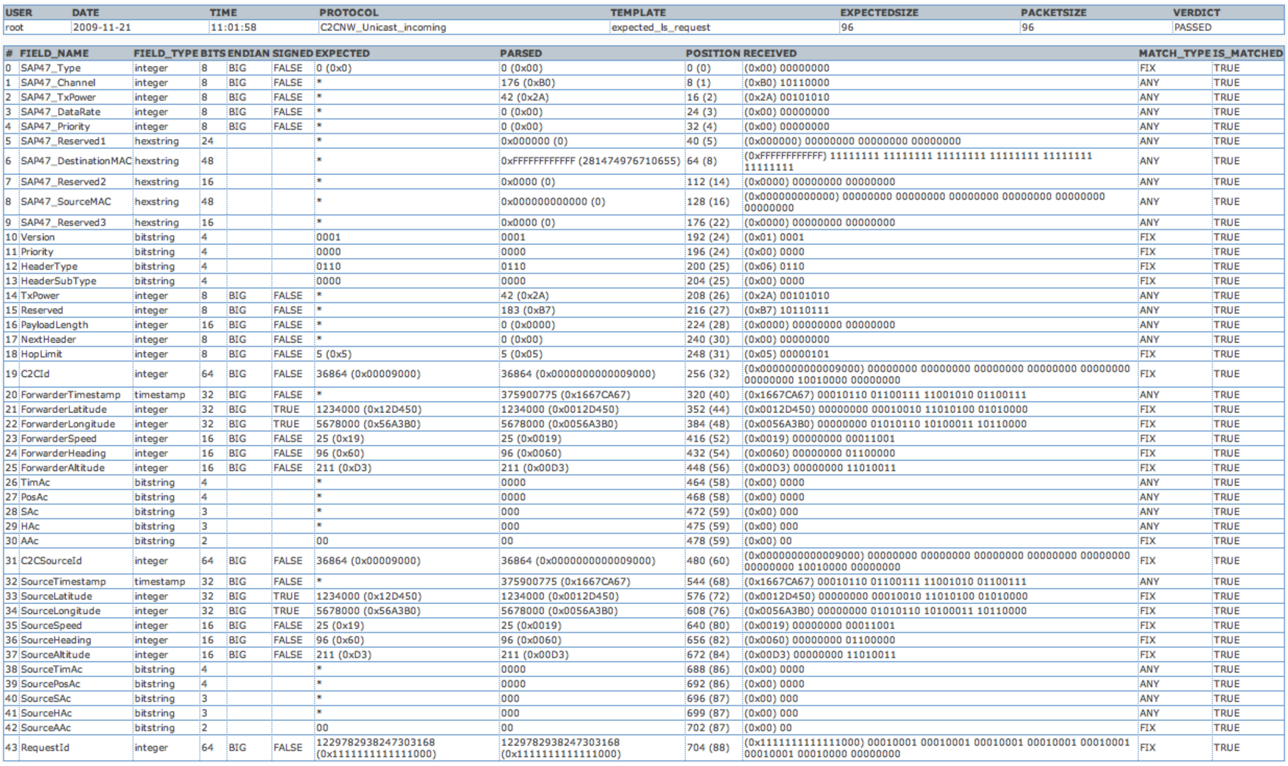
3.4.11 Example GeoNet Packet Parse Result

A C2CNet incoming unicast packet with content “ls_request” parse result with matching of template definition.

Packet content in binary (bitstring) and hexadecimal format

```
Packet reader and parser - BroadBit ASN.1/TTCN compiler

Selected template name: "expected_ls_request"

Received packet in bitstring format: 00000000 10110000 00101010 00000000 00000000 00000000 00000000 00000000
11111111 11111111 11111111 11111111 11111111 11111111 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00010000 01100000 00101010 10110111 00000000 00000000 00000000 00000101
00000000 00000000 00000000 00000000 00000000 00000000 10010000 00000000 00010110 01100111 11001010 01100111
00000000 00010010 11010100 01010000 00000000 01010110 10100011 10110000 00000000 00011001 00000000 01100000
00000000 11010011 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 10010000 00000000
00010110 01100111 11001010 01100111 00000000 00010010 11010100 01010000 00000000 01010110 10100011 10110000
00000000 00011001 00000000 01100000 00000000 11010011 00000000 00000000 00010001 00010001 00010001 00010001
00010001 00010001 00010000 00000000

Received packet in hexstring format: 00 b0 2a 00 00 00 00 00 ff ff ff ff ff ff 00 00 00 00 00 00 00 00 10 60
2a b7 00 00 00 05 00 00 00 00 00 00 90 00 16 67 ca 67 00 12 d4 50 00 56 a3 b0 00 19 00 60 00 d3 00 00 00 00 00 00
00 00 90 00 16 67 ca 67 00 12 d4 50 00 56 a3 b0 00 19 00 60 00 d3 00 00 11 11 11 11 11 11 10 00

Protocol definition to use: "C2CNW_Location_Service_request"

Parse "expected_ls_request" result: condition check passed.
```

Parsed packet contents table as output



USER	DATE	TIME	PROTOCOL	TEMPLATE	EXPECTEDSIZE	PACKETSIZE	VERDICT
root	2009-11-21	11:01:58	C2CNW_Unicast_incoming	expected_ls_request	96	96	PASSED

#	FIELD_NAME	FIELD_TYPE	BITS	ENDIAN	SIGNED	EXPECTED	PARSED	POSITION	RECEIVED	MATCH_TYPE	IS_MATCHED
0	SAP47_Type	integer	8	BIG	FALSE	0 (0x0)	0 (0x00)	0 (0)	(0x00) 00000000	FIX	TRUE
1	SAP47_Channel	integer	8	BIG	FALSE	*	176 (0xB0)	8 (1)	(0xB0) 10110000	ANY	TRUE
2	SAP47_TxPower	integer	8	BIG	FALSE	*	42 (0x2A)	16 (2)	(0x2A) 00101010	ANY	TRUE
3	SAP47_DataRate	integer	8	BIG	FALSE	*	0 (0x00)	24 (3)	(0x00) 00000000	ANY	TRUE
4	SAP47_Priority	integer	8	BIG	FALSE	*	0 (0x00)	32 (4)	(0x00) 00000000	ANY	TRUE
5	SAP47_Reserved1	hexstring	24			*	0x000000 (0)	40 (5)	(0x000000) 00000000 00000000 00000000	ANY	TRUE
6	SAP47_DestinationMAC	hexstring	48			*	0xFFFFFFFFFFFF (281474976710655)	64 (8)	(0xFFFFFFFFFFFF) 11111111 11111111 11111111 11111111 11111111 11111111	ANY	TRUE
7	SAP47_Reserved2	hexstring	16			*	0x0000 (0)	112 (14)	(0x0000) 00000000 00000000	ANY	TRUE
8	SAP47_SourceMAC	hexstring	48			*	0x000000000000 (0)	128 (16)	(0x000000000000) 00000000 00000000 00000000 00000000 00000000 00000000	ANY	TRUE
9	SAP47_Reserved3	hexstring	16			*	0x0000 (0)	176 (22)	(0x0000) 00000000 00000000	ANY	TRUE
10	Version	bitstring	4			0001	0001	192 (24)	(0x01) 0001	FIX	TRUE
11	Priority	bitstring	4			0000	0000	196 (24)	(0x00) 0000	FIX	TRUE
12	HeaderType	bitstring	4			0110	0110	200 (25)	(0x06) 0110	FIX	TRUE
13	HeaderSubType	bitstring	4			0000	0000	204 (25)	(0x00) 0000	FIX	TRUE
14	TxPower	integer	8	BIG	FALSE	*	42 (0x2A)	208 (26)	(0x2A) 00101010	ANY	TRUE
15	Reserved	integer	8	BIG	FALSE	*	183 (0xB7)	216 (27)	(0xB7) 10110111	ANY	TRUE
16	PayloadLength	integer	16	BIG	FALSE	*	0 (0x0000)	224 (28)	(0x0000) 00000000 00000000	ANY	TRUE
17	NextHeader	integer	8	BIG	FALSE	*	0 (0x00)	240 (30)	(0x00) 00000000	ANY	TRUE
18	HopLimit	integer	8	BIG	FALSE	5 (0x5)	5 (0x05)	248 (31)	(0x05) 00000101	FIX	TRUE
19	C2CId	integer	64	BIG	FALSE	36864 (0x00009000)	36864 (0x0000000000009000)	256 (32)	(0x0000000000009000) 00000000 00000000 00000000 00000000 00000000 00000000 10010000 00000000	FIX	TRUE
20	ForwarderTimestamp	timestamp	32	BIG	FALSE	*	375900775 (0x1667CA67)	320 (40)	(0x1667CA67) 00010110 01100111 11001010 01100111	ANY	TRUE
21	ForwarderLatitude	integer	32	BIG	TRUE	1234000 (0x12D450)	1234000 (0x0012D450)	352 (44)	(0x0012D450) 00000000 00010010 11010100 01010000	FIX	TRUE
22	ForwarderLongitude	integer	32	BIG	TRUE	5678000 (0x56A3B0)	5678000 (0x0056A3B0)	384 (48)	(0x0056A3B0) 00000000 01010110 10100011 10110000	FIX	TRUE
23	ForwarderSpeed	integer	16	BIG	FALSE	25 (0x19)	25 (0x0019)	416 (52)	(0x0019) 00000000 00011001	FIX	TRUE
24	ForwarderHeading	integer	16	BIG	FALSE	96 (0x60)	96 (0x0060)	432 (54)	(0x0060) 00000000 01100000	FIX	TRUE
25	ForwarderAltitude	integer	16	BIG	FALSE	211 (0xD3)	211 (0x00D3)	448 (56)	(0x00D3) 00000000 11010011	FIX	TRUE
26	TimAc	bitstring	4			*	0000	464 (58)	(0x00) 0000	ANY	TRUE
27	PosAc	bitstring	4			*	0000	468 (58)	(0x00) 0000	ANY	TRUE
28	SAc	bitstring	3			*	000	472 (59)	(0x00) 000	ANY	TRUE
29	HAc	bitstring	3			*	000	475 (59)	(0x00) 000	ANY	TRUE
30	AAc	bitstring	2			00	00	478 (59)	(0x00) 00	FIX	TRUE
31	C2CSourceId	integer	64	BIG	FALSE	36864 (0x00009000)	36864 (0x0000000000009000)	480 (60)	(0x0000000000009000) 00000000 00000000 00000000 00000000 00000000 00000000 10010000 00000000	FIX	TRUE
32	SourceTimestamp	timestamp	32	BIG	FALSE	*	375900775 (0x1667CA67)	544 (68)	(0x1667CA67) 00010110 01100111 11001010 01100111	ANY	TRUE
33	SourceLatitude	integer	32	BIG	TRUE	1234000 (0x12D450)	1234000 (0x0012D450)	576 (72)	(0x0012D450) 00000000 00010010 11010100 01010000	FIX	TRUE
34	SourceLongitude	integer	32	BIG	TRUE	5678000 (0x56A3B0)	5678000 (0x0056A3B0)	608 (76)	(0x0056A3B0) 00000000 01010110 10100011 10110000	FIX	TRUE
35	SourceSpeed	integer	16	BIG	FALSE	25 (0x19)	25 (0x0019)	640 (80)	(0x0019) 00000000 00011001	FIX	TRUE
36	SourceHeading	integer	16	BIG	FALSE	96 (0x60)	96 (0x0060)	656 (82)	(0x0060) 00000000 01100000	FIX	TRUE
37	SourceAltitude	integer	16	BIG	FALSE	211 (0xD3)	211 (0x00D3)	672 (84)	(0x00D3) 00000000 11010011	FIX	TRUE
38	SourceTimAc	bitstring	4			*	0000	688 (86)	(0x00) 0000	ANY	TRUE
39	SourcePosAc	bitstring	4			*	0000	692 (86)	(0x00) 0000	ANY	TRUE
40	SourceSAc	bitstring	3			*	000	696 (87)	(0x00) 000	ANY	TRUE
41	SourceHAc	bitstring	3			*	000	699 (87)	(0x00) 000	ANY	TRUE
42	SourceAAc	bitstring	2			00	00	702 (87)	(0x00) 00	FIX	TRUE
43	RequestId	integer	64	BIG	FALSE	1229782938247303168 (0x1111111111111000)	1229782938247303168 (0x1111111111111000)	704 (88)	(0x1111111111111000) 00010001 00010001 00010001 00010001 00010001 00010001 00010000 00000000	FIX	TRUE

3.5 Abstract Test Cases

An abstract test case is a complete and independent specification of the actions required to achieve a specific test purpose at the Abstract Test Method's level of abstraction. It starts and ends in a stable testing state. It can include implementations running sequentially or concurrently. Test cases are written in a programming language, usually specific to testing, such as the advanced language or – in this case - TTCN language.

The test cases are defined following particular rules as defined here:

- **TC ID**: The Test Case (TC) ID is a unique identifier;
- **TC Name**: Name of the Test Case;
- **List of Test Purposes**: Complete list of Test Purposes attached to this Test Case;
- **Result**: Test result.

Example:

TC ID	TC Name	List of Test Purposes	Result
TC/01	C2CNet - Hitachi	C2CNet_Hitachi/TB/UDP/BV/01, C2CNet_Hitachi/TB/UDP/BV/02, ...	valid
TC/02	C2CNet - NEC	C2CNet_NEC/TB/UDP/BV/01, C2CNet_Hitachi/TB/UDP/BV/02, ...	invalid

4. C2CNet Test Cases

This section describes details of executed conformance test cases. The resulting conformance testing suite comprises of the sum of these test cases.

- Notice: all source URLs in test case definition point to GeoNet Internal Collaboration Site, and requires user name / password for access!

Legend of colours used in flowcharts elements:

Colour	Entity description
Yellow	Start position of test case
Green	Skip or no-operation position
Red	Failed test position
White	Fork by result of a decide
Pink	Module to test
Blue	Test port in TTCN terminology, equivalent to GeoNet SAP entity
Green	Passed test position

TTCN port group list:

Port group name	Port description
PositionSensor	Incoming position information (GPS data)
LowerLayer	Incoming and outgoing C2C Net, or IP packages on LowerLayer
UpperLayer	Incoming and outgoing C2C Net, or IP packages on UpperLayer

4.1 Beacon Data Corresponding to Sensor Input and Beacon Fields Specification

This test case verifies whether generated beacons contain position data which corresponds to position sensor input. The test case verifies that data of generated beacons corresponds to simulated position sensor data and to the specifications.

4.1.1 Test Flow

The flowchart in Figure 6 shows test cycle elements and states.

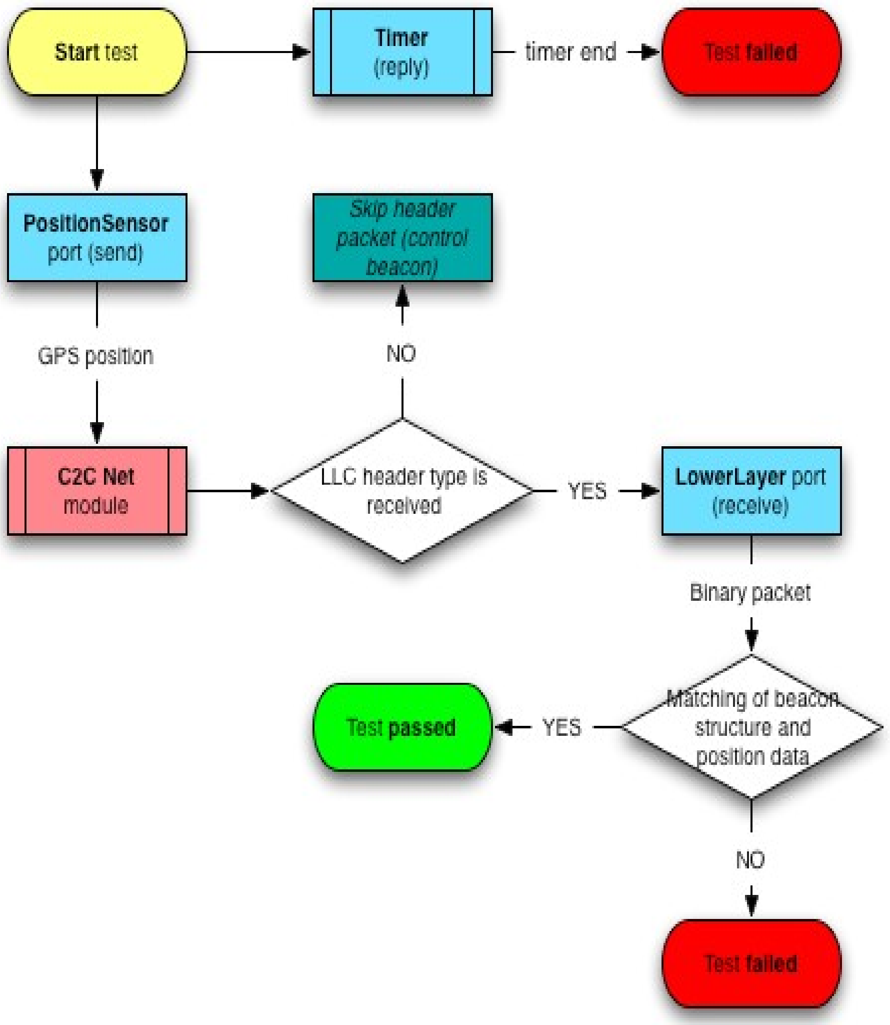

Start test
Timer (reply)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Test passed
YES
Matching of beacon structure and position data
NO
Test failed

Figure 6: Verification that Beacon Data Corresponds to Sensor Input and to the Specification of Beacon Fields

4.1.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Beacon.

4.1.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/01/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/01/ttcn/C2CNWTester.ttcn

4.1.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/01/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/01/ttcn/packet_description.xml

4.1.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/01/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/01/ttcn/packet_description_template.xml

4.2 Correct IP Packet Delivery to Lower Layer SAP - Sender Side

This test case verifies that IPv6 data is delivered to Lower Layer with C2CNet destination correct address. The correspondence between a generated test IPv6 packet and the resulting C2CNet message on the lower layer is tested, including correct addressing to the destination.

4.2.1 Test Flow

The flowchart in Figure 7 shows test cycle elements and states.

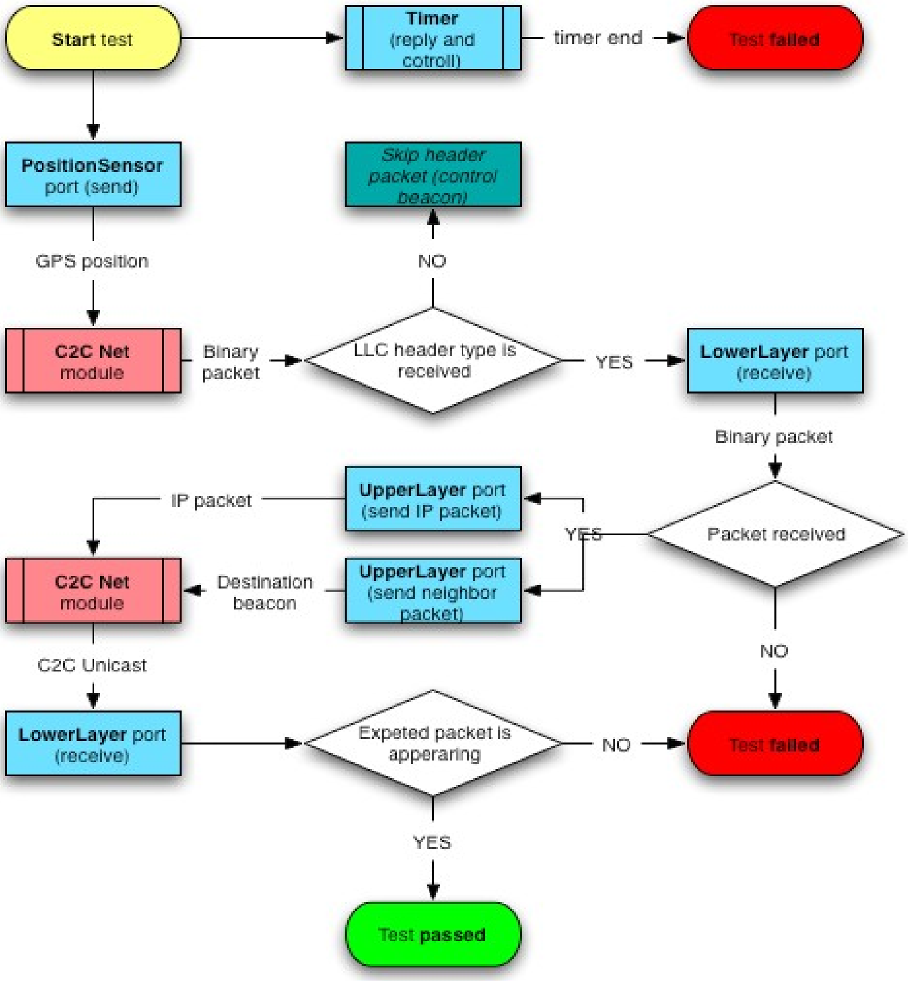

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
IP packet
UpperLayer port (send IP packet)
YES
Packet received
C2C Net module
Destination beacon
UpperLayer port (send neighbor packet)
C2C Unicast
NO
LowerLayer port (receive)
Expeted packet is apperaring
NO
Test failed
YES
Test passed

Figure 7: Verification that IP Packets are Correctly Sent out at Lower Layer SAP on Sender Side

4.2.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** Ipv6_Packet.

4.2.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/02/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/02/ttcn/C2CNWTester.ttcn

4.2.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/02/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/02/ttcn/packet_description.xml

4.2.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/02/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/02/ttcn/packet_description_template.xml

4.3 Correct Delivery of Incoming lower Layer Packet to IP layer

This test case verifies that Lower Layer unicast is delivered to the IPv6 layer. It is tested if a generated C2CNet message on the lower layer results in the correct IPV6 packet.

4.3.1 Test Flow

The flowchart in Figure 8 shows test cycle elements and states.

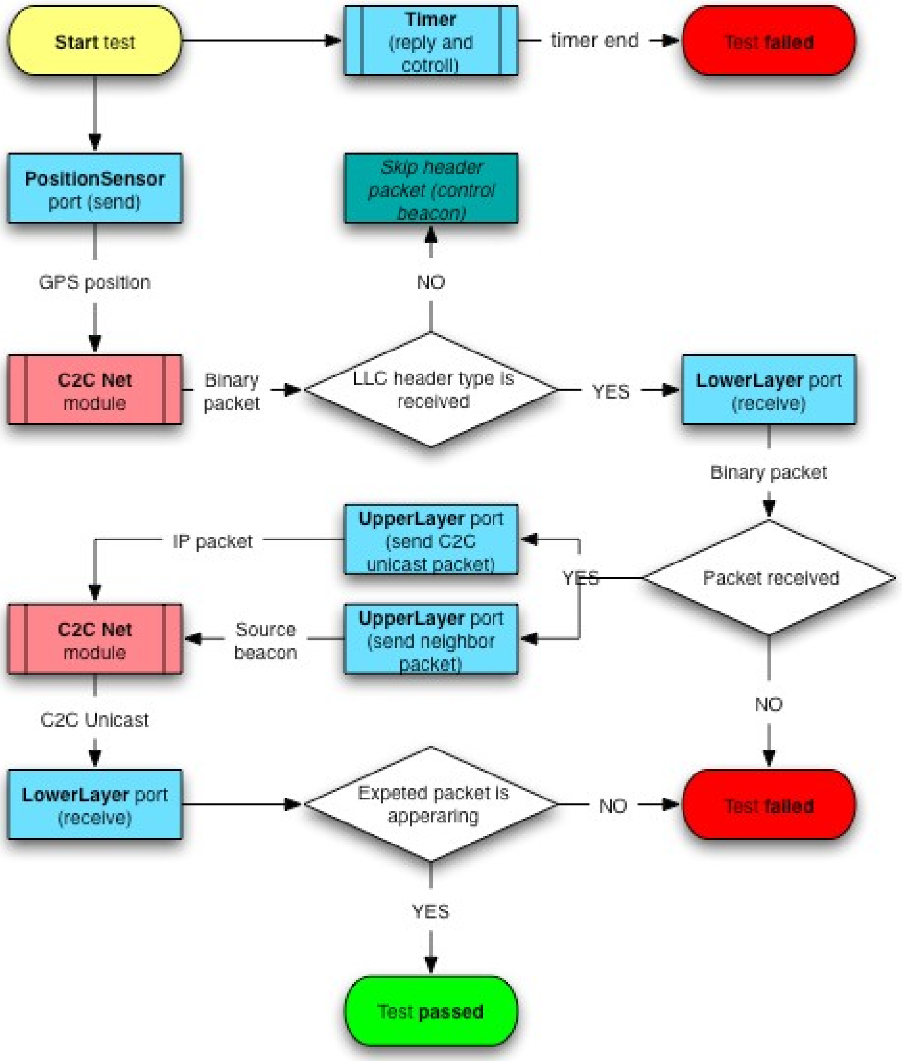

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
UpperLayer port (send C2C unicast packet)
IP packet
YES
Packet received
C2C Net module
Source beacon
UpperLayer port (send neighbor packet)
C2C Unicast
NO
LowerLayer port (receive)
Expeted packet is apperaring
NO
Test failed
YES
Test passed

Figure 8: Verification that Incoming Lower Layer Packet is Delivered to the IP Layer

4.3.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** Ipv6_Packet.

4.3.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/03/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/03/ttcn/C2CNWTester.ttcn

4.3.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/03/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/03/ttcn/packet_description.xml

4.3.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/03/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/03/ttcn/packet_description_template.xml

4.4 Correct Next Forwarder Selection

This test case verifies that Location Service request is sent out when destination location is unknown, and data is sent to destination upon receiving Location Service answer. The correspondence between a generated test IPv6 packet, the resulting Location Service request, and the final sending of C2CNet message to a suitable forwarder on the lower layer is tested.

4.4.1 Test Flow

The flowchart in Figure 9 shows test cycle elements and states.

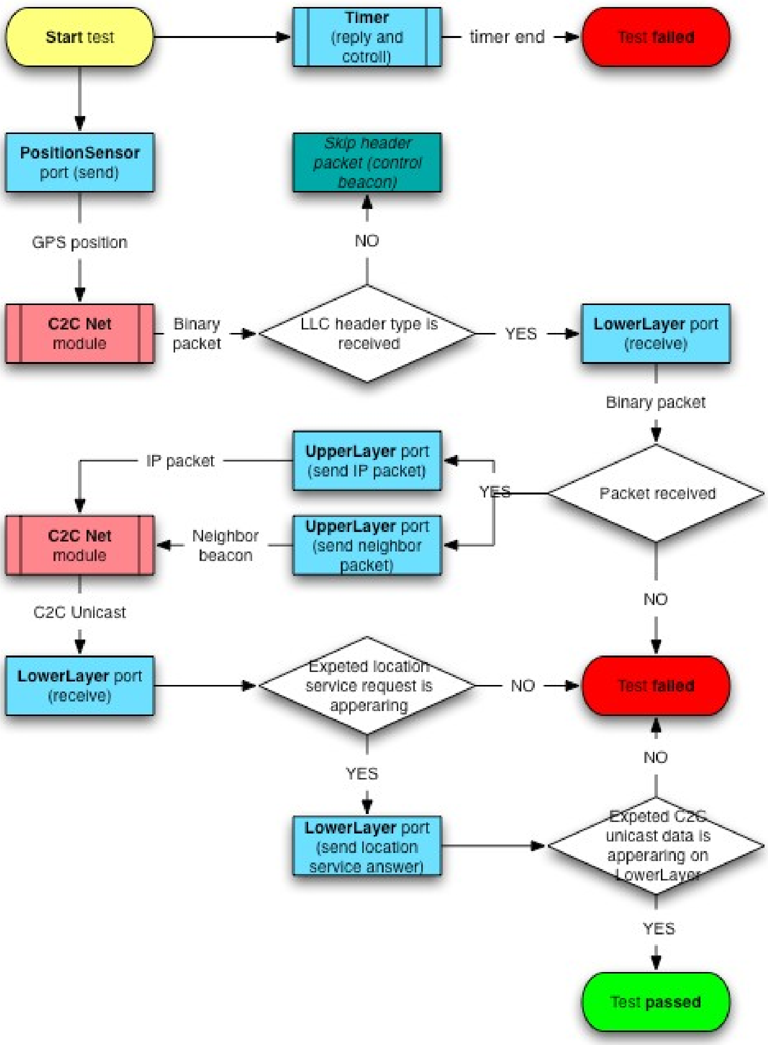

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
IP packet
UpperLayer port (send IP packet)
YES
Packet received
C2C Net module
Neighbor beacon
UpperLayer port (send neighbor packet)
C2C Unicast
NO
LowerLayer port (receive)
Expeted location service request is apperaring
NO
Test failed
YES
NO
LowerLayer port (send location service answer)
Expeted C2C unicast data is apperaring on LowerLayer
YES
Test passed

Figure 9: Verification that Routing over the C2CNet Layer Selects Appropriate Next Forwarder Among Candidates

4.4.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.4.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/04/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/04/ttcn/C2CNWTester.ttcn

4.4.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/04/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/04/ttcn/packet_description.xml

4.4.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/04/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/04/ttcn/packet_description_template.xml

4.5 Correct Next Forwarder for Multi-Hop Forwarder

This test case verifies that the C2CNet layer selects correct next forwarder for multi-hop forwarding. This test case is almost the same as test case number 4, with the difference of checking that the most suitable next forwarder is selected among candidates - evaluated according to the geo-networked routing rules.

4.5.1 Test Flow

The flowchart in Figure 10 shows test cycle elements and states.

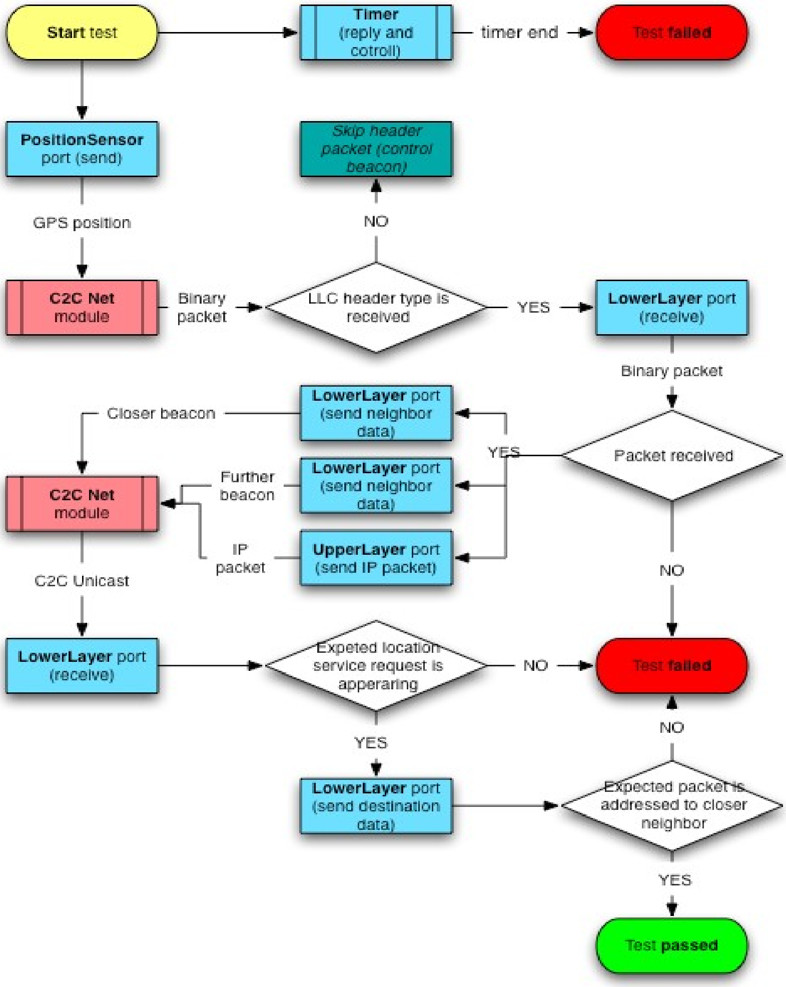

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Closer beacon
LowerLayer port (send neighbor data)
YES
Packet received
Further beacon
LowerLayer port (send neighbor data)
C2C Net module
IP packet
UpperLayer port (send IP packet)
C2C Unicast
NO
LowerLayer port (receive)
Expeted location service request is apperaring
NO
Test failed
YES
NO
LowerLayer port (send destination data)
Expected packet is addressed to closer neighbor
YES
Test passed

Figure 10: Verification that C2CNet Layer Selects Correct Next Forwarder for Multi-Hop Forwarding

4.5.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.5.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/05/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/05/ttcn/C2CNWTester.ttcn

4.5.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/05/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/05/ttcn/packet_description.xml

4.5.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/05/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/05/ttcn/packet_description_template.xml

4.6 Correct Multi-hop Forwarding at Intermediate Node

This test case verifies multi-hop forwarding to correct next forwarder, initiated by reception of Lower Layer packet. It is tested if an incoming C2CNet message on the Lower Layer results in the correct outgoing C2CNet message.

4.6.1 Test Flow

The flowchart in Figure 11 shows test cycle elements and states.

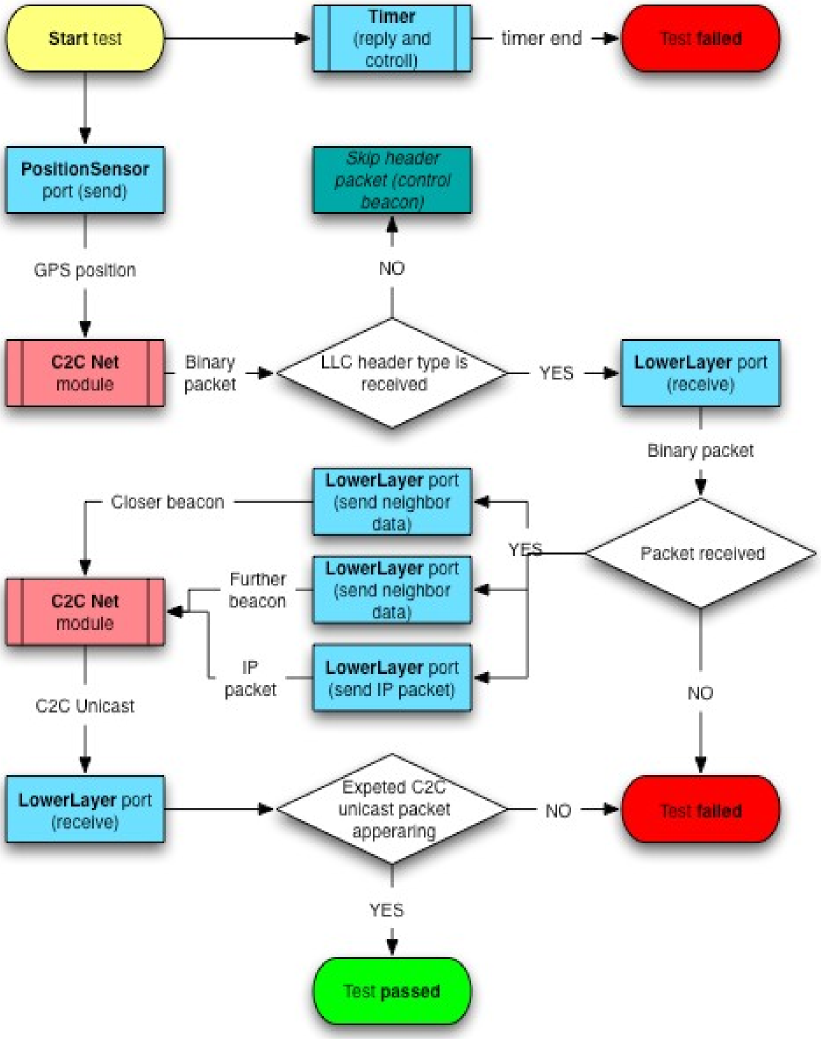

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Closer beacon
LowerLayer port (send neighbor data)
YES
Packet received
Further beacon
LowerLayer port (send neighbor data)
C2C Net module
IP packet
LowerLayer port (send IP packet)
NO
C2C Unicast
LowerLayer port (receive)
Expeted C2C unicast packet appearing
NO
Test failed
YES
Test passed

Figure 11: Multi-hop Forwarding Validation at Intermediate Node, Initiated by Packet Reception at Lower Layer SAP

4.6.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.6.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/06/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/06/ttcn/C2CNWTester.ttcn

4.6.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/06/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/06/ttcn/packet_description.xml

4.6.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/06/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/06/ttcn/packet_description_template.xml

4.7 Correct Store and Forward Buffering

This test case verifies the correct use of the store and forward buffer in the C2CNet layer. Initially there is no suitable forwarding candidate for the generated test message, as both simulated “neighbours” are situated in wrong direction. The test case verifies that the message is buffered until a “neighbour” is appearing in the direction leading closer to destination.

4.7.1 Test Flow

The flowchart in Figure 12 shows test cycle elements and states.

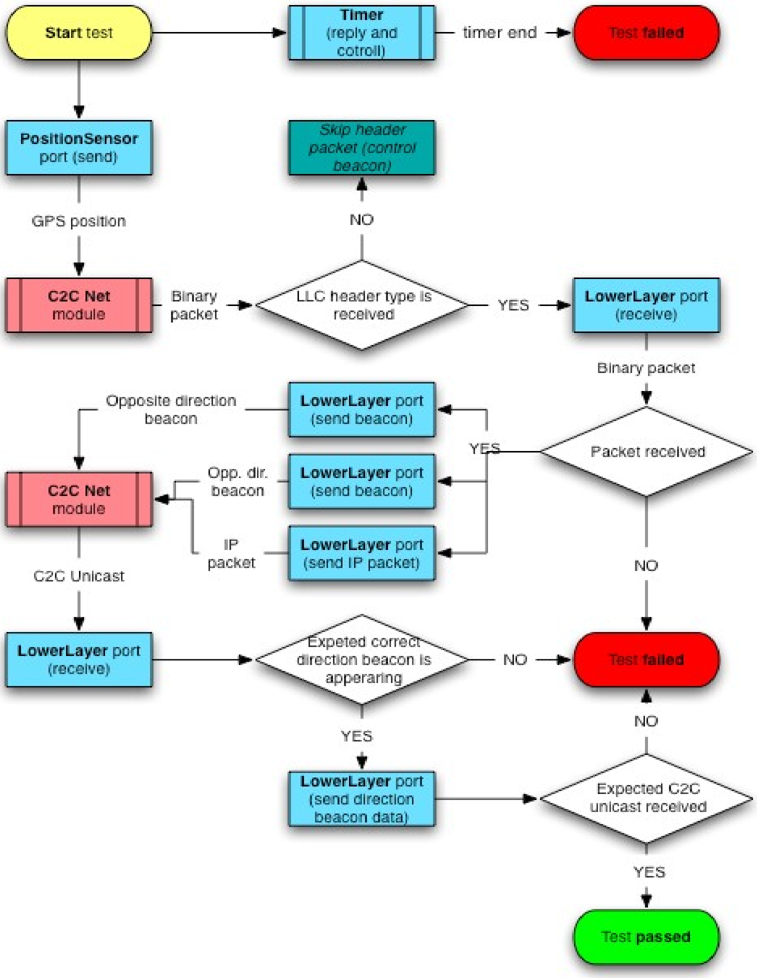

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Opposite direction beacon
LowerLayer port (send beacon)
YES
Packet received
C2C Net module
Opp. dir. beacon
LowerLayer port (send beacon)
IP packet
LowerLayer port (send IP packet)
NO
C2C Unicast
LowerLayer port (receive)
Expeted correct direction beacon is apperaring
NO
Test failed
NO
YES
LowerLayer port (send direction beacon data)
Expected C2C unicast received
YES
Test passed

Figure 12: Validation of Store and Forward Buffering Mechanism

4.7.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.7.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/07/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/07/ttcn/C2CNWTester.ttcn

4.7.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/07/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/07/ttcn/packet_description.xml

4.7.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/07/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/07/ttcn/packet_description_template.xml

4.8 GeoBroadcasting Verification

This test case verifies the correct use of GeoBroadcasting; including re-broadcasting, geo-area check, and duplicate broadcast check.

The point in this test case is to verify that the received GeoBroadcast is correctly processed the first time, and is discarded when being received the second time. This test case also verifies that a received out-of-area GeoBroadcast is not being processed.

4.8.1 Test Flow

The flowchart in Figure 13 shows test cycle elements and states.

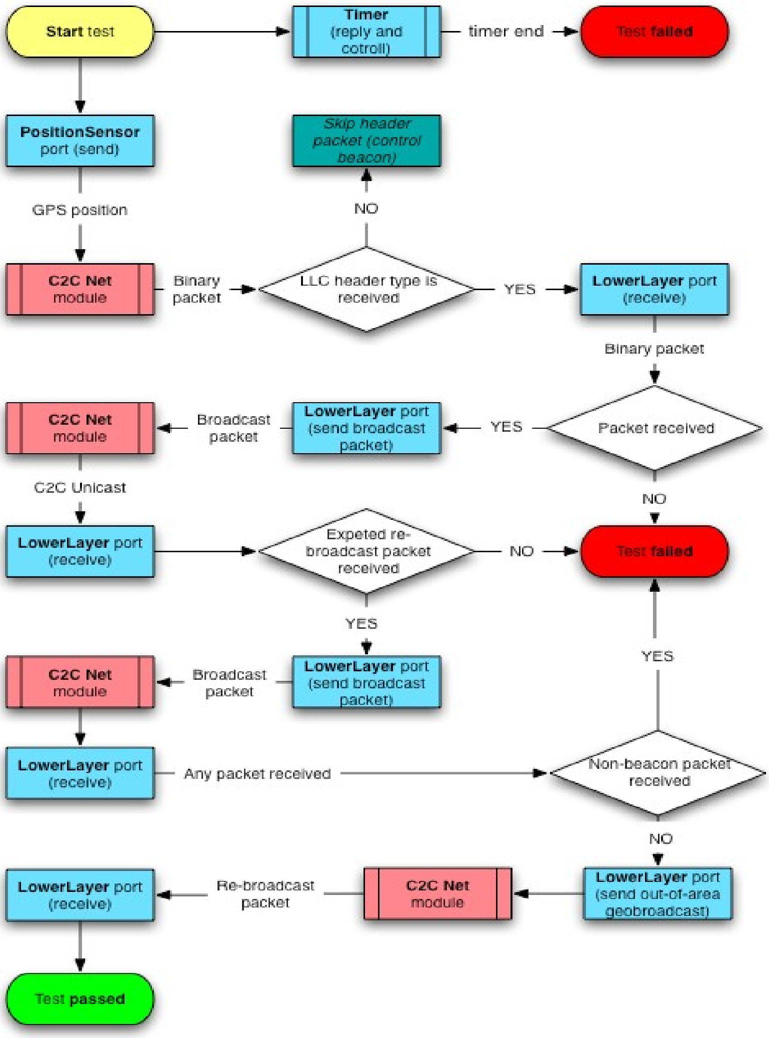

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
C2C Net module
Broadcast packet
LowerLayer port (send broadcast packet)
YES
Packet received
C2C Unicast
NO
LowerLayer port (receive)
Expeted re-broadcast packet received
NO
Test failed
YES
YES
C2C Net module
Broadcast packet
LowerLayer port (send broadcast packet)
LowerLayer port (receive)
Any packet received
Non-beacon packet received
NO
LowerLayer port (receive)
Re-broadcast packet
C2C Net module
LowerLayer port (send out-of-area geobroadcast)
Test passed

Figure 13. GeoBroadcasting Verification

4.8.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.8.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/08/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/08/ttcn/C2CNWTester.ttcn

4.8.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/08/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/08/ttcn/packet_description.xml

4.8.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/08/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/08/ttcn/packet_description_template.xml

4.9 Correct GeoBroadcast Reception at Upper Layer SAP

This test case verifies proper delivery of GeoBroadcasting to Upper Layer. The generated test scenario is the same as in test case #8, but the validation of protocol behaviour is now checking correct message delivery to the IPv6 layer. The re-broadcast handling is not checked here. The point in this test case is to verify that the received GeoBroadcast is correctly processed the first time, and is discarded when being received the second time (that would happen during a flooding process).

4.9.1 Test Flow

The flowchart in Figure 14 shows test cycle elements and states.

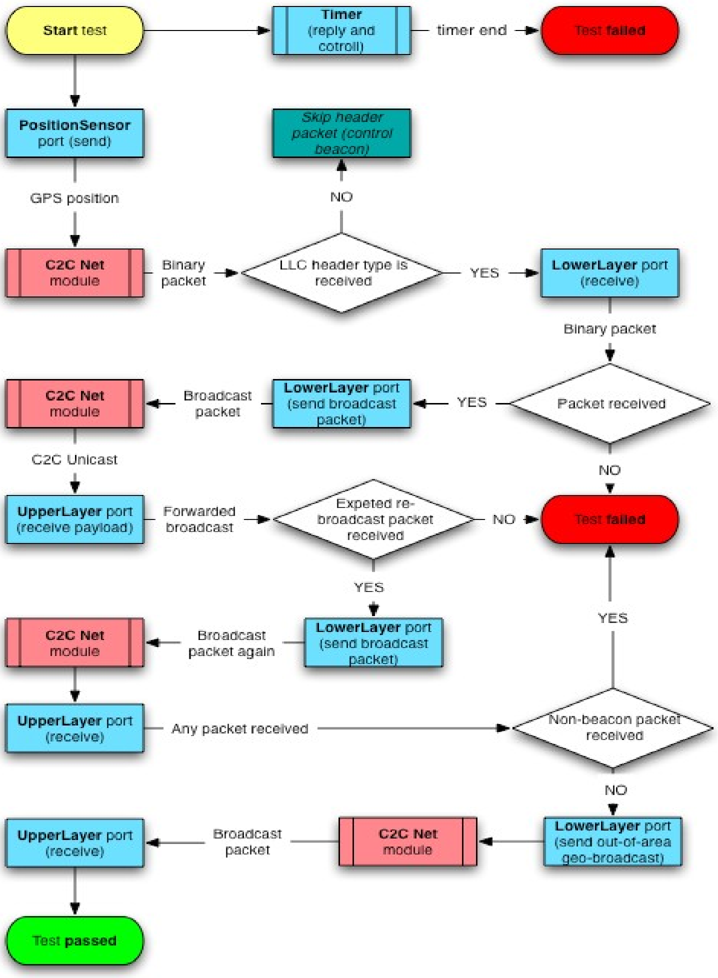

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
C2C Net module
Broadcast packet
LowerLayer port (send broadcast packet)
YES
Packet received
C2C Unicast
NO
UpperLayer port (receive payload)
Forwarded broadcast
Expeted re-broadcast packet received
NO
Test failed
YES
YES
C2C Net module
Broadcast packet again
LowerLayer port (send broadcast packet)
UpperLayer port (receive)
Any packet received
Non-beacon packet received
NO
UpperLayer port (receive)
Broadcast packet
C2C Net module
LowerLayer port (send out-of-area geo-broadcast)
Test passed

Figure 14: Validation of Correct GeoBroadcast Reception at Upper Layer SAP

4.9.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.9.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/09/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/09/ttcn/C2CNWTester.ttcn

4.9.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/09/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/09/ttcn/packet_description.xml

4.9.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/09/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/09/ttcn/packet_description_template.xml

4.10 Correct Handling of Location Table Entries' Expiration

This test case verifies the lifetime of an entry in the location table. It is tested that in the absence of beacon update the corresponding location table entry is removed when its lifetime expires.

4.10.1 Test Flow

The flowchart in Figure 15 shows test cycle elements and states.

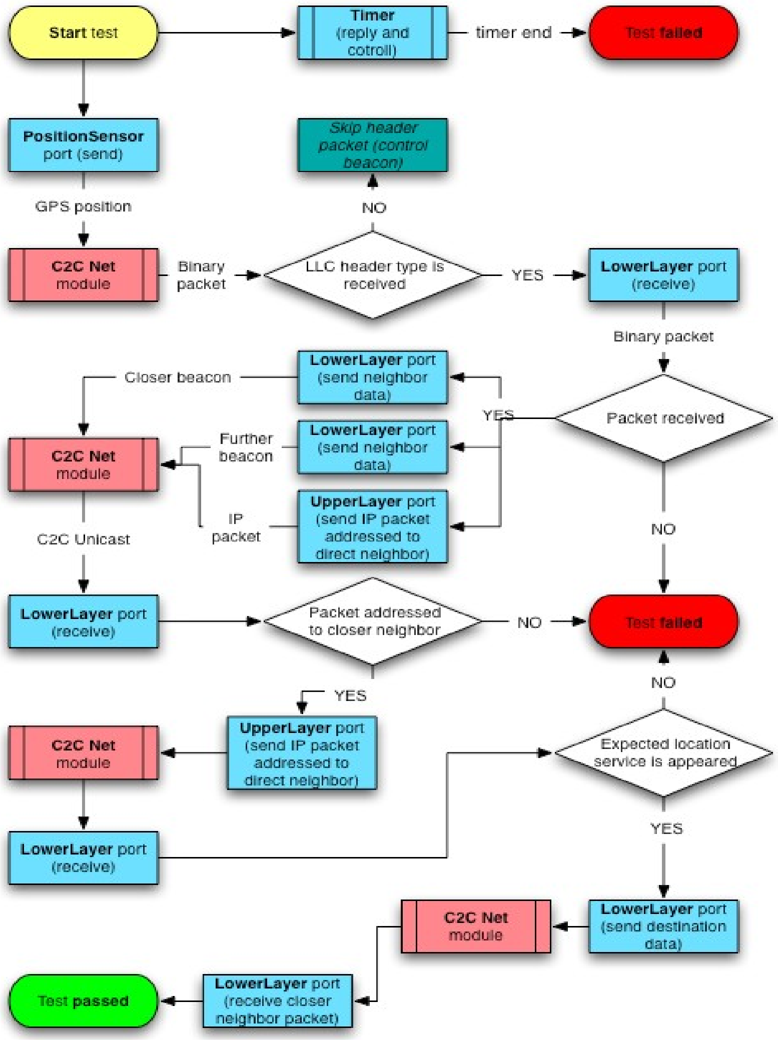

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Closer beacon
LowerLayer port (send neighbor data)
YES
Packet received
Further beacon
LowerLayer port (send neighbor data)
C2C Net module
IP packet
UpperLayer port (send IP packet addressed to direct neighbor)
C2C Unicast
NO
LowerLayer port (receive)
Packet addressed to closer neighbor
NO
Test failed
YES
NO
C2C Net module
UpperLayer port (send IP packet addressed to direct neighbor)
Expected location service is appeared
LowerLayer port (receive)
YES
C2C Net module
LowerLayer port (send destination data)
Test passed
LowerLayer port (receive closer neighbor packet)

Figure 15: Verification of Location Table Entries' Expiration Handling

4.10.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.10.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/11/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/11/ttcn/C2CNWTester.ttcn

4.10.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/11/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/11/ttcn/packet_description.xml

4.10.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/11/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/11/ttcn/packet_description_template.xml

4.11 End-to-end Validation of MNPP

This test case verifies the correct functioning of the MNPP (Mobile Network Prefix Provisioning) mechanism part of sub-module "IPv6 over C2Net" (Module 3A "IP Forwarding"). At the C2CNet layer it is verified that MNPP advertisement messages are GeoBroadcast as expected. At the IPv6 layer it is verified that the "radvd" (Linux IPv6 Router Advertisement Daemon) module updates the IPv6 routing table from received MNPP advertisement information.

4.11.1 Test Flow

The flowchart in Figure 16 shows test cycle elements and states.

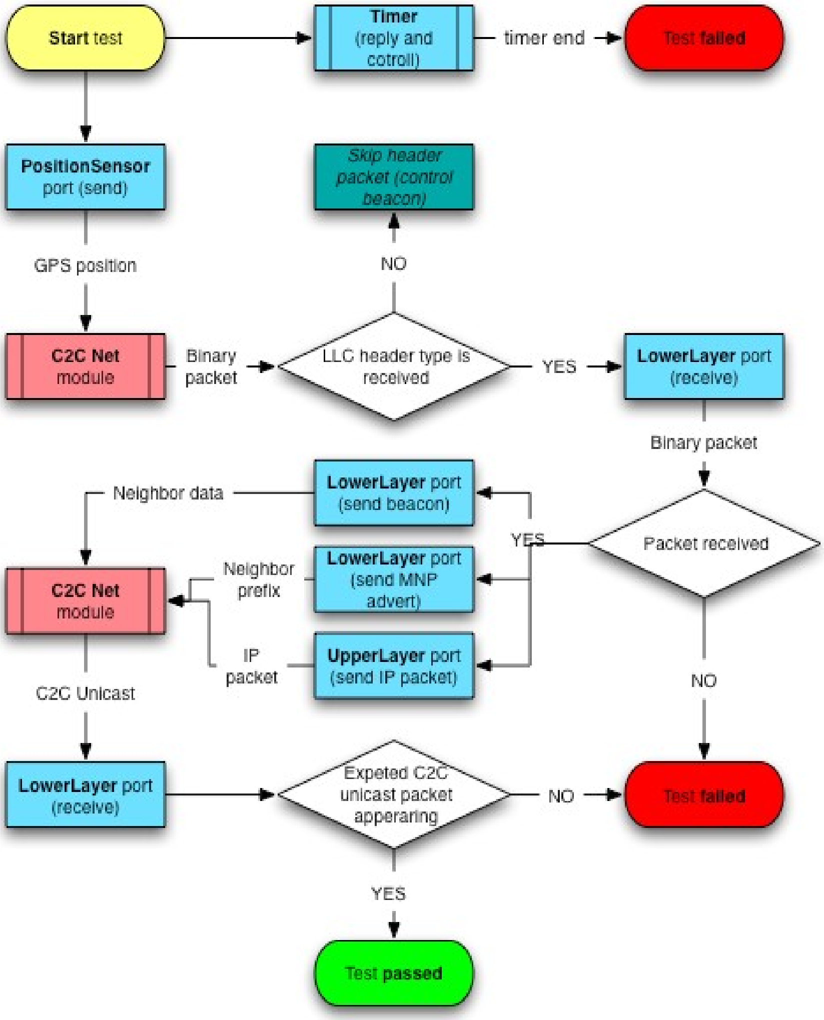

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Neighbor data
LowerLayer port (send beacon)
YES
Packet received
C2C Net module
Neighbor prefix
LowerLayer port (send MNP advert)
IP packet
UpperLayer port (send IP packet)
C2C Unicast
NO
LowerLayer port (receive)
Expeted C2C unicast packet apperaring
NO
Test failed
YES
Test passed

Figure 16: End-to-End Validation of MNPP

4.11.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.11.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/12/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/12/ttcn/C2CNWTester.ttcn

4.11.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/12/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/12/ttcn/packet_description.xml

4.11.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/12/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/12/ttcn/packet_description_template.xml

4.12 Correct Handling of Entry Expiration in the Store & Forward Message Buffer

This test case verifies the correct use of data expiration from message forwarding buffer. It is tested that a buffer entry is removed when its lifetime expires.

4.12.1 Test Flow

The flowchart in Figure 17 shows test cycle elements and states.

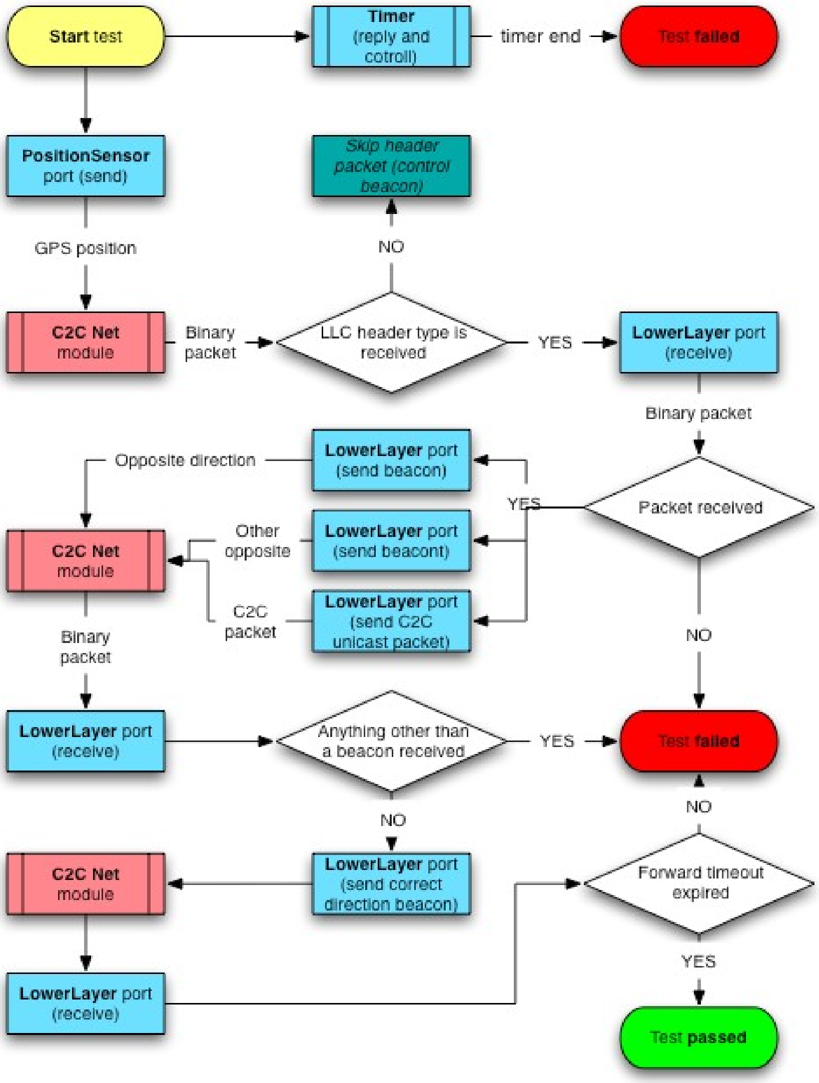

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Opposite direction
LowerLayer port (send beacon)
YES
Packet received
C2C Net module
Other opposite
LowerLayer port (send beacont)
C2C packet
LowerLayer port (send C2C unicast packet)
NO
Binary packet
LowerLayer port (receive)
Anything other than a beacon received
YES
Test failed
NO
NO
C2C Net module
LowerLayer port (send correct direction beacon)
Forward timeout expired
YES
LowerLayer port (receive)
Test passed

FIgure 17: Verification of Entry Expiration Handling in the Store & Forward message Buffer

4.12.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.12.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/13/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/13/ttcn/C2CNWTester.ttcn

4.12.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/13/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/13/ttcn/packet_description.xml

4.12.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/13/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/13/ttcn/packet_description_template.xml

4.13 Correct Re-evaluation of Stored Entries' in the Store & Forward Message Buffer

This test case verifies the re-evaluation of the store and forward buffer triggered by the movement of a “neighbouring” forwarder. A received message is initially stored in the message forwarding buffer because of the lack of suitable forwarder candidates. It is tested that when an existing moving “neighbours” becomes a suitable candidate, the buffered message is forwarded to it.

4.13.1 Test Flow

The flowchart in Figure 18 shows test cycle elements and states.

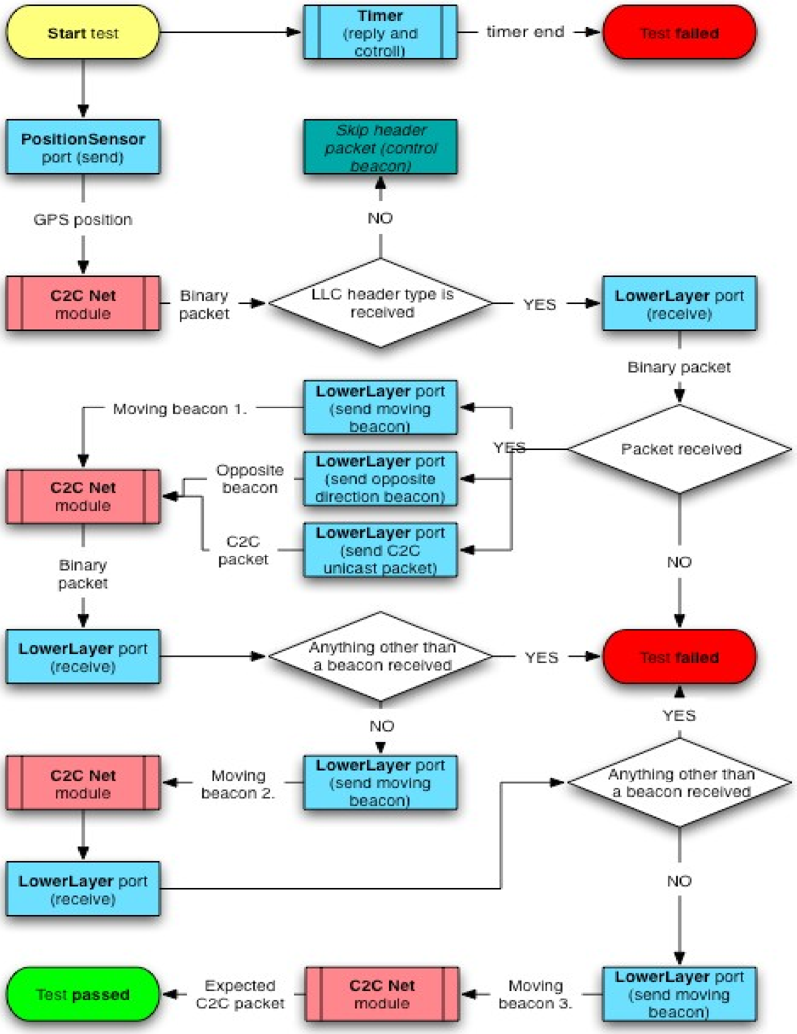

Start test
Timer (reply and cotroll)
timer end
Test failed
PositionSensor port (send)
Skip header packet (control beacon)
GPS position
NO
C2C Net module
Binary packet
LLC header type is received
YES
LowerLayer port (receive)
Binary packet
Moving beacon 1.
LowerLayer port (send moving beacon)
YES
Packet received
Opposite beacon
LowerLayer port (send opposite direction beacon)
C2C Net module
C2C packet
LowerLayer port (send C2C unicast packet)
NO
Binary packet
LowerLayer port (receive)
Anything other than a beacon received
YES
Test failed
NO
YES
C2C Net module
Moving beacon 2.
LowerLayer port (send moving beacon)
Anything other than a beacon received
LowerLayer port (receive)
NO
Test passed
Expected C2C packet
C2C Net module
Moving beacon 3.
LowerLayer port (send moving beacon)

Figure 18: Test of Stored Entries' Re-evaluation in the Store & Forward Message Buffer

4.13.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information;
- **LowerLayer:** C2CNW_Unicast_incoming, C2CNW_Unicast_outgoing;
- **UpperLayer:** IPv6_Packet.

4.13.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/14/ttcn/C2CNWTester.ttcn
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/14/ttcn/C2CNWTester.ttcn

4.13.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/14/ttcn/packet_description.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/14/ttcn/packet_description.xml

4.13.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-hitachi/14/ttcn/packet_description_template.xml
- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/c2cnw-nec/14/ttcn/packet_description_template.xml

5. PositionSensor Test Cases

There is only one test case in this section which verifies through a range of possible values that the position sensor module provides valid position information to the C2CNet layer. It is furthermore verified that this provided data is in the format that is specified by the Common Networking Header [GeoNetD2.2].

Note on Accuracy and Heading calculation

Accuracy should eventually reflect the accuracy of the GNSS method used, meaning GPS, EGNOS, Galileo or some other positioning technology. The assumption is that a higher value means a higher accuracy. But there is no general “rule” behind this right now.

Currently only GPS is used, therefore only two values are used:

- Value = 0 means that there is NO position data known.
- Value = 1 means that there is position data known.

“Heading” value is received directly from GPS daemon (“gpsd”) and converted by position sensor into units of 0.005493247 degrees. Processing of this value is only possible if the GPS device generates heading information.

5.1 Validation of Correct GPS Data Conversion

This test case verifies whether the GPS data received from the PositionSensor on the GPS-SAP corresponds to the real GPS data which is originated by the GPS hardware and then forwarded by the “gpsd” demon in NMEA format.

5.1.1 Test Flow

The flowchart in Figure 19 shows test cycle elements and states.

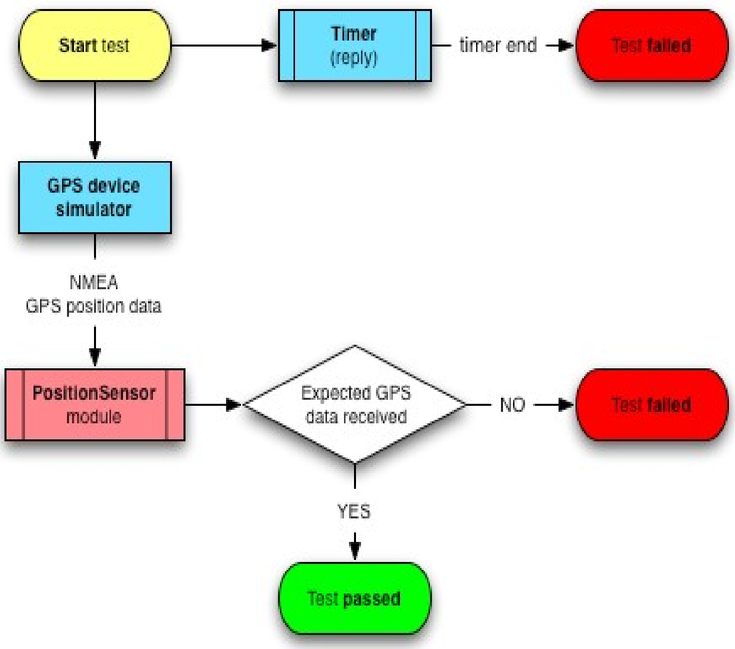


Figure 19: Validation of the Correct Conversion of GPS Data

5.1.2 TTCN Ports

Predefined TTCN ports to represent input/output interfaces between test environment and application. Following TTCN ports defined in TTCN script:

- **PositionSensor:** Position_information

5.1.3 TTCN Script

TTCN script describes testing flow with TTCN ports and status definitions. TTCN Script sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/positionsensor/01/ttcn/PositionSensorTester.ttcn

5.1.4 Packet Definition

Packet definition describes incoming and outgoing packet types with field and data type elements. Packet Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/positionsensor/01/ttcn/packet_description.xml

5.1.5 Template Definition

Template definition contains data to fill up outgoing packets and data to parse in incoming packets. Template Definition Sources:

- https://geonet-project.eu/geonet/browser/Binaries/wp4broadbit/modules/positionsensor/01/ttcn/packet_description_template.xml

6. Test Reports

6.1 Testing Timeline and Cycles

Following table shows conformance testing timeline by months and test cycles (binary release and install, run tests, report to developer).

Date	Hitachi imp. test cycles	NEC imp. test cycles	Lesswire imp. test cycles	Total
aug/2009	1	0	0	1
sept/2009	10	5	0	15
oct/2009	60	20	0	80
nov/2009	50	30	0	80
dec/2009	60	80	10	150
jan/2010	40	50	2	92
Totals*:*	221	185	12	**418**

6.2 Binary Release and Test Report Graph

Figure 20 presents graphs of binary releases and revisions, created report tickets and solved issues (closed tickets).

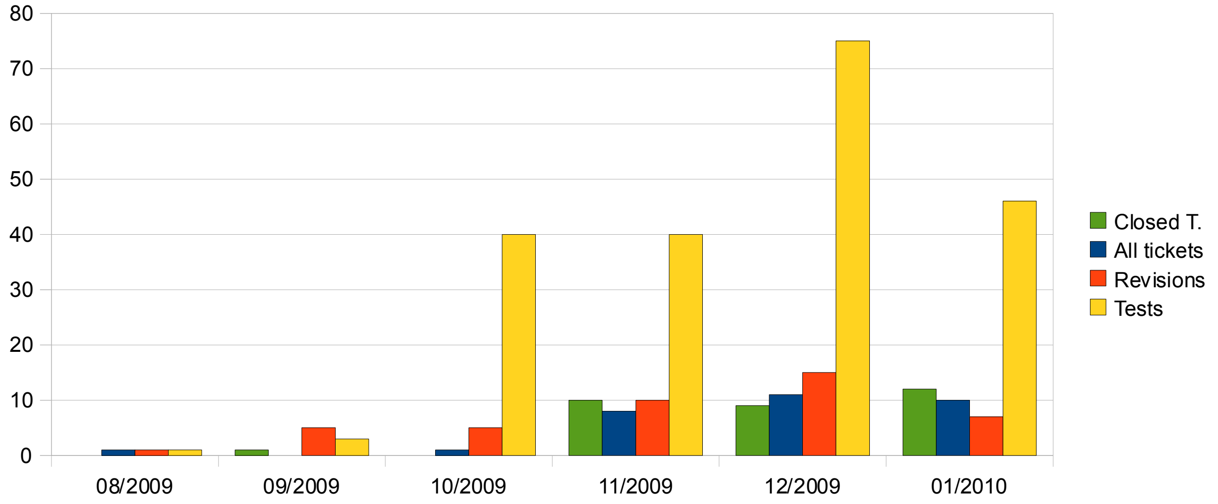
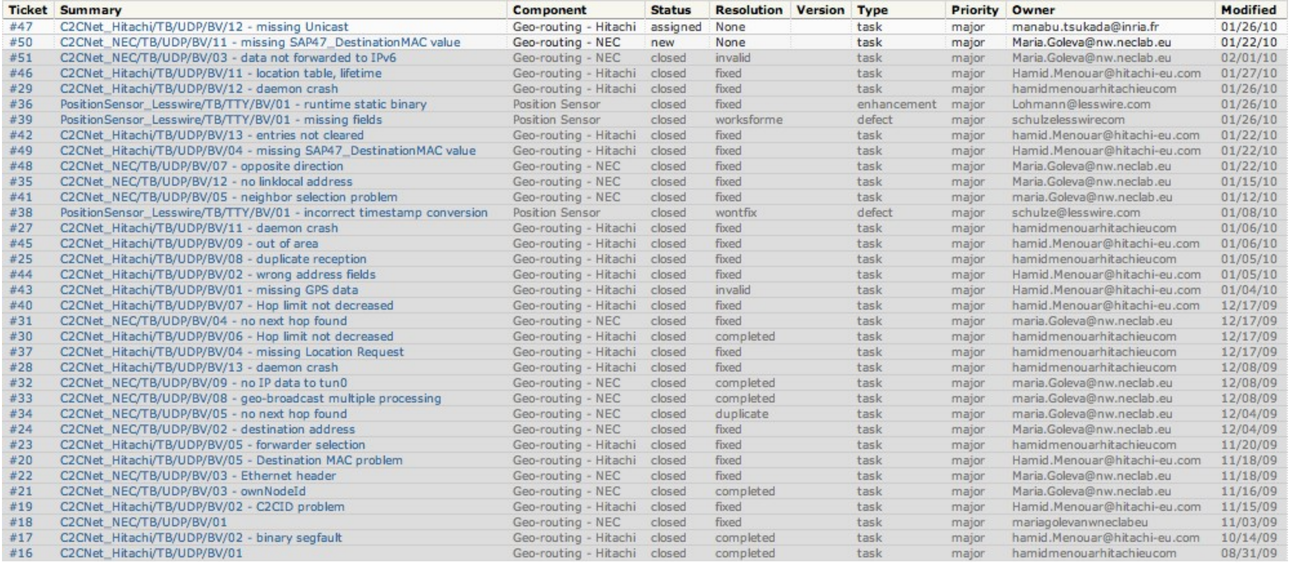


Figure 20: Binary Release and Test Report Graph

6.3 Report Tickets

Figure 21 shows the list of created issue tickets in internal collaboration tool addressed to developers. Source of ticket report: https://geonet-project.eu/geonet/report/6



Ticket	Summary	Component	Status	Resolution	Version	Type	Priority	Owner	Modified
#47	C2CNet_Hitachi/TB/UDP/BV/12 - missing Unicast	Geo-routing - Hitachi	assigned	None		task	major	manabu.tsukada@inria.fr	01/26/10
#50	C2CNet_NEC/TB/UDP/BV/11 - missing SAP47_DestinationMAC value	Geo-routing - NEC	new	None		task	major	Maria.Goleva@nw.neclab.eu	01/22/10
#51	C2CNet_NEC/TB/UDP/BV/03 - data not forwarded to IPv6	Geo-routing - NEC	closed	invalid		task	major	Maria.Goleva@nw.neclab.eu	02/01/10
#46	C2CNet_Hitachi/TB/UDP/BV/11 - location table, lifetime	Geo-routing - Hitachi	closed	fixed		task	major	Hamid.Menouar@hitachi-eu.com	01/27/10
#29	C2CNet_Hitachi/TB/UDP/BV/12 - daemon crash	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	01/26/10
#36	PositionSensor_Lesswire/TB/TTY/BV/01 - runtime static binary	Position Sensor	closed	fixed		enhancement	major	Lohmann@lesswire.com	01/26/10
#39	PositionSensor_Lesswire/TB/TTY/BV/01 - missing fields	Position Sensor	closed	worksforme		defect	major	schulzelesswirecom	01/26/10
#42	C2CNet_Hitachi/TB/UDP/BV/13 - entries not cleared	Geo-routing - Hitachi	closed	fixed		task	major	hamid.Menouar@hitachi-eu.com	01/22/10
#49	C2CNet_Hitachi/TB/UDP/BV/04 - missing SAP47_DestinationMAC value	Geo-routing - Hitachi	closed	fixed		task	major	Hamid.Menouar@hitachi-eu.com	01/22/10
#48	C2CNet_NEC/TB/UDP/BV/07 - opposite direction	Geo-routing - NEC	closed	fixed		task	major	Maria.Goleva@nw.neclab.eu	01/22/10
#35	C2CNet_NEC/TB/UDP/BV/12 - no linklocal address	Geo-routing - NEC	closed	fixed		task	major	Maria.Goleva@nw.neclab.eu	01/15/10
#41	C2CNet_NEC/TB/UDP/BV/05 - neighbor selection problem	Geo-routing - NEC	closed	fixed		task	major	maria.Goleva@nw.neclab.eu	01/12/10
#38	PositionSensor_Lesswire/TB/TTY/BV/01 - incorrect timestamp conversion	Position Sensor	closed	wontfix		defect	major	schulze@lesswire.com	01/08/10
#27	C2CNet_Hitachi/TB/UDP/BV/11 - daemon crash	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	01/06/10
#45	C2CNet_Hitachi/TB/UDP/BV/09 - out of area	Geo-routing - Hitachi	closed	fixed		task	major	hamid.Menouar@hitachi-eu.com	01/06/10
#25	C2CNet_Hitachi/TB/UDP/BV/08 - duplicate reception	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	01/05/10
#44	C2CNet_Hitachi/TB/UDP/BV/02 - wrong address fields	Geo-routing - Hitachi	closed	fixed		task	major	Hamid.Menouar@hitachi-eu.com	01/05/10
#43	C2CNet_Hitachi/TB/UDP/BV/01 - missing GPS data	Geo-routing - Hitachi	closed	invalid		task	major	Hamid.Menouar@hitachi-eu.com	01/04/10
#40	C2CNet_Hitachi/TB/UDP/BV/07 - Hop limit not decreased	Geo-routing - Hitachi	closed	fixed		task	major	hamid.Menouar@hitachi-eu.com	12/17/09
#31	C2CNet_NEC/TB/UDP/BV/04 - no next hop found	Geo-routing - NEC	closed	fixed		task	major	maria.Goleva@nw.neclab.eu	12/17/09
#30	C2CNet_Hitachi/TB/UDP/BV/06 - Hop limit not decreased	Geo-routing - Hitachi	closed	completed		task	major	hamidmenouarhitachieucom	12/17/09
#37	C2CNet_Hitachi/TB/UDP/BV/04 - missing Location Request	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	12/17/09
#28	C2CNet_Hitachi/TB/UDP/BV/13 - daemon crash	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	12/08/09
#32	C2CNet_NEC/TB/UDP/BV/09 - no IP data to tun0	Geo-routing - NEC	closed	completed		task	major	maria.Goleva@nw.neclab.eu	12/08/09
#33	C2CNet_NEC/TB/UDP/BV/08 - geo-broadcast multiple processing	Geo-routing - NEC	closed	completed		task	major	maria.Goleva@nw.neclab.eu	12/08/09
#34	C2CNet_NEC/TB/UDP/BV/05 - no next hop found	Geo-routing - NEC	closed	duplicate		task	major	maria.Goleva@nw.neclab.eu	12/04/09
#24	C2CNet_NEC/TB/UDP/BV/02 - destination address	Geo-routing - NEC	closed	fixed		task	major	Maria.Goleva@nw.neclab.eu	12/04/09
#23	C2CNet_Hitachi/TB/UDP/BV/05 - forwarder selection	Geo-routing - Hitachi	closed	fixed		task	major	hamidmenouarhitachieucom	11/20/09
#20	C2CNet_Hitachi/TB/UDP/BV/05 - Destination MAC problem	Geo-routing - Hitachi	closed	fixed		task	major	Hamid.Menouar@hitachi-eu.com	11/18/09
#22	C2CNet_NEC/TB/UDP/BV/03 - Ethernet header	Geo-routing - NEC	closed	fixed		task	major	Maria.Goleva@nw.neclab.eu	11/18/09
#21	C2CNet_NEC/TB/UDP/BV/03 - ownNodeId	Geo-routing - NEC	closed	completed		task	major	Maria.Goleva@nw.neclab.eu	11/16/09
#19	C2CNet_Hitachi/TB/UDP/BV/02 - C2CID problem	Geo-routing - Hitachi	closed	fixed		task	major	Hamid.Menouar@hitachi-eu.com	11/15/09
#18	C2CNet_NEC/TB/UDP/BV/01	Geo-routing - NEC	closed	fixed		task	major	mariagolevanwneclabeu	11/03/09
#17	C2CNet_Hitachi/TB/UDP/BV/02 - binary segfault	Geo-routing - Hitachi	closed	completed		task	major	hamid.Menouar@hitachi-eu.com	10/14/09
#16	C2CNet_Hitachi/TB/UDP/BV/01	Geo-routing - Hitachi	closed	completed		task	major	hamidmenouarhitachieucom	08/31/09

Figure 21: Ticket Report Table

7. Conformance Test Results

7.1 GeoNet C2CNet Daemon

Conformance test result table of C2CNet daemon developed by Hitachi and NEC.

Revision number of Hitachi implementation: revision 781

Revision number of NEC implementation: revision 759

Test case num.	Last test date	Hitachi imp. result	NEC imp. result
01	26/jan/2010	passed	passed
02	26/jan/2010	passed	passed
03	26/jan/2010	passed	passed
04	26/jan/2010	passed	passed
05	26/jan/2010	passed	passed
06	26/jan/2010	passed	passed
07	26/jan/2010	passed	passed
08	26/jan/2010	passed	passed
09	26/jan/2010	passed	passed
10 (skipped test case numbering)	-	-	-
11	26/jan/2010	passed	failed
12	26/jan/2010	failed[1]	passed
13	26/jan/2010	passed	passed
14	26/jan/2010	passed	failed

7.2 PositionSensor

Revision number of Lesswire implementation: 548

Test case num.	Last test date	Lesswire imp. result	
01	17/dec/2009	passed, when considering the scope of position data required by the present GeoNet implementation[2]	

[1] The interaction of NEMO and MNPP in the GeoNetworking daemon, which is verified by this test case for IPv6 over GeoNetworking transactions, had not been finalised when the project ended by the end of January. This has been solved in the software update released in June. Re-testing the conformance of this functionality is left for future work.

[2] While all present requirements of the other GeoNet modules are met, full conformance with the GeoNet specification would also require correct processing of speed and heading data. This is a deficiency that should be fixed in the future.

This GeoNet conformance testing work is followed up by the development of conformance testing standards, undertaken by the ETSI specialist task force for “Conformance test specification for CAM & DENM, GeoNetworking, Basic Transport Protocol, and integration of IPv6 and GeoNetworking” (ETSI STF 405). That ETSI activity will use GeoNet conformance testing results as input and will also extend the scope of conformance testing in the following ways:

- Application layer testing is also performed for CAM and DENM applications, which rely on underlying GeoNetworking services

- The direct access of GeoNetworking layer is also considered, accessed through a 'Basic Transport Protocol' between the GeoNetworking layer and Application layer.

- Any changes between [GeoNetD2.2] and its adopted ETSI standard version are considered.

After revision by ETSI STF 405, the finalised TTCN test scripts will be published as part of the ETSI standards.

8. Conclusion and Results

The described test procedures follow the standardised TTCN-3 test methodology and tools. TTCN-3 (Testing and Test Control Notation version 3) is a strongly typed test scripting language used in conformance testing of communicating systems and a specification of test infrastructure interfaces that glue abstract test scripts with concrete communication environments. TTCN-3 has been developed by ETSI and its predecessor is TTCN-2. The use of this methodology results in a test suite that requires just O(N) tests to validate specification conformance and interoperability of N implementations – as opposed to testing each implementation against every other one. This test suite shall be made available for testing specification conformance of future IPv6 GeoNetworking implementations.

The chart in Figure 22 shows the monthly progress of testing work in terms of the new tickets assigned to a responsible implementation engineer (tickets refer to a detected deviation that must be corrected), the number of executed test cases, and the number of processed software revisions. At the time of the project's completion, the status of testing is the following:

- The C2CNet layer implementations (Modules 2.5) have successfully passed all test cases for the GeoNetworking functionality. Some unresolved test tickets related to the IPv6 over C2CNet sub-module (Module 3A “IP Forwarding”) still exist, specifically the support of MNPP and NEMO features. However, an updated implementation released some time after the end of the project in January successfully passed all test cases. In addition to the conformance tests reported in the present document, the validation of the two C2CNet layer implementations and the implementation of the IPv6 over C2CNet sub-module has been completed by a number of plug-tests and a performance analysis of IPv6 over C2CNet. This performance analysis has been realised by two means: by emulation means, and by experimental means using data collected during live tests conducted with vehicles. Last but not least, a live demonstration of a realistic IPv6 GeoNetworking scenario (traffic hazard notification) has been performed at the end of the project. The plug tests, the experimental performance analysis and the live demonstration are reported in [GeoNetD7.1] whereas the performance analysis by means of emulation is reported in [GeoNetD5.1].

- The Position Sensor (Module 0C) implementation has passed the test case, which has been designed to validate the hardware-independent provisioning of needed position data to other GeoNet modules. It is noted however, that the position sensor module implementation has some unresolved test tickets relating to the provisioning of vehicle speed and heading data. These data types are needed to achieve full conformance with the GeoNet specification [GeoNetD2.2], but are not needed by the present implementation of other software modules.

The present conformance testing work may continue in the future, subject to agreements among involved GeoNet partners and potential future implementers of the GeoNet system.

Such continued work may be needed for testing future improvements to the present implementations or for testing the conformance of new GeoNet system implementations.

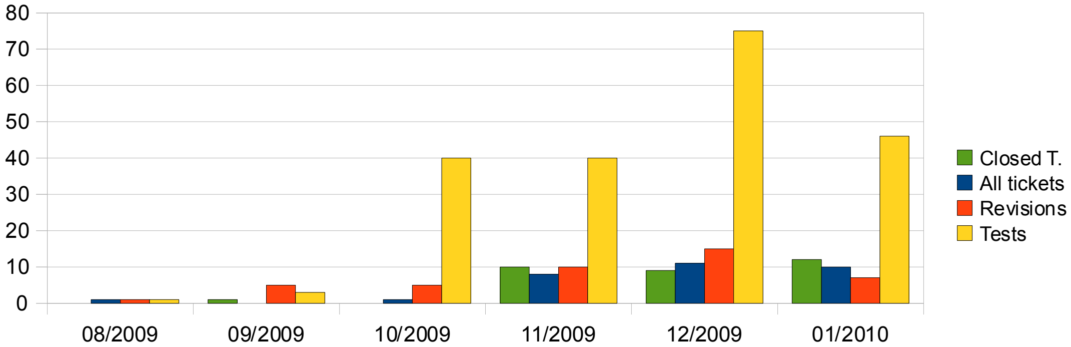


Figure 22: Conformance test cycles, reported tickets and binary revisions

The GeoNet conformance testing work is followed up by the development of conformance testing standards, undertaken by the ETSI specialist task force for “Conformance test specification for CAM & DENM, GeoNetworking, Basic Transport Protocol, and integration of IPv6 and GeoNetworking” (ETSI STF 405). This ETSI STF work will use GeoNet conformance testing results as input and the finalised TTCN test scripts will be published as part of the ETSI standards.

Annex A: Contributors

The following experts have contributed to this GeoNet document, in alphabetical order:

Thierry Ernst – INRIA

Maria Goleva – NEC

Andras Kovacs – BroadBit

Hamid Menouar – Hitachi Europe

Lajos Nagy – BroadBit

Carsten Schulze – Lesswire

Wenhui Zhang – NEC

Annex B: Terminology & Acronyms

The terminology used in the context of this work (IPv6, GeoNetworking and other generic networking terms) is defined Annex D of GeoNetD1.2]. In addition to these, the following terms which applies only to this document are also defined:

- **Conformance testing**: Conformance testing or type testing is testing to determine whether a system meets some specified standard. Conformance testing is often performed by external organisations, sometimes the standards body itself, to give greater guarantees of compliance. Products tested in such a manner are then advertised as being certified by that external organisation as complying with the standard.

- **TTCN:** TTCN is a programming language used for testing of communication protocols and web services. A TTCN test suite consists of many test cases written in the TTCN programming language.

- **Test case:** the fundamental building block in a test suite. A test case tests a particular feature of function in implementation under test (IUT). A test case has an identified test purpose and it assigns a verdict that depend on the outcome of the test case.

- **Test suite:** the highest level, encompassing all test components and serving as the root of the tree. A test suite can range from a large number of test groups and test cases to a single test event contained in test case.

- **TC ID:** The Test Case (TC) ID is a unique identifier.

- **TC Name:** Name of the Test Case.

- **List of Test Purposes**: Complete listing of Test Purposes attached to a Test Case

Annex C: References

[CAM] ETSI TS 102 637-2. Cooperative Awareness Messaging

[DENM] ETSI TS 102 637-3. Decentralized Environmental Notification Messaging

[ES 201 873-1] ETSI. “TTCN-3: Core Language”, 4.1.1

[ES 201 873-4] ETSI. ”TTCN-3: Operational Semantics”, 4.1.1

[ES 201 873-5] ETSI. “TTCN-3: TTCN-3 Runtime Interface”, 4.1.1

[ES 201 873-6] ETSI. “TTCN-3: TTCN-3 Control Interface”, 4.1.1

[ES 201 873-7] ETSI. “TTCN-3: The use of ASN.1”, 4.1.1

[ES 201 873-9] ETSI. “TTCN-3: Using XML schema with TTCN-3”, 4.1.1

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

[GeoNetD1.2] GeoNet. “Final GeoNet Architecture Design”, GeoNet Deliverable D1.2, January 2010.

[GeoNetD3.1] GeoNet. “GeoNet Development Results”. GeoNet Deliverable D3.1, January 2010.

[GeoNetD2.2] GeoNet. “Final GeoNet Specification”, GeoNet Deliverable D2.2, January 2010.

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

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

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

[TTCN-3] TTCN has been developed and is maintained by the Methods for Testing and Specification Technical Committee (TC-MTS) at ETSI. This group is composed of leading testing experts from industry and academia as well as members of ETSI's own Centre for Testing & Interoperability (CTI). TTCN-3 was born out of TTCN-2 which had been maintained by the same group. TTCN testing technology has been applied widely and

successfully in European industry, ETSI standardization, and certification for more than a decade.