CONFORMANCE TEST SPECIFICATION

V1.0.00

Version	Modifications	Date
0.1.01 0.1.02 0.1.03	First draft document for the Conformance Test Specification	2021-05-31
0.2.00 0.2.01 0.2.02	Draft for circulation after TC meeting 2021-06-29	2021-07-09
1.0.00	First preliminary release	2021-10-15

Summary

This document specifies the test to show conformance to the Wize protocol in order to provide interoperation with the Wize network infrastructure. The tests cover the lower layers (PHY and DLL), as well as transport layer (TPL) and the common application layers (APL). This release of the Test Specification applies to the European 169 MHz band.

Table of contents

Summary *1*
Table of contents *2*
Tables and figures *4*
Contributors ..... ***Error! Bookmark not defined.***

1 Introduction 5
2 Standard documents 6
3 Abbreviations 7
4 General requirements and manufacturer declarations 7
4.1 [GEN-01] Device information 7
4.2 [GEN-02] RED Conformity 7
4.3 [GEN-03] Transmission Duty Cycle 7
4.4 [GEN-04] Radiated power 8
4.5 [GEN-05] Minimum sensitivity 8
4.6 [GEN-06] Random transmission time 8
5 Physical Layer (PHY) test specification 8
5.1 [PHY-01] Uplink channel 8
5.2 [PHY-02] Uplink modulation (data rate) 8
5.3 [PHY-03] Downlink channel 9
5.4 [PHY-04] Downlink modulation (data rate) 9
5.5 [PHY-05] Frequency accuracy 9
5.6 [PHY-06] Frequency adjustment 10
5.7 [PHY-07] Output power management 10
6 Medium Access sublayer (MAC) test specification 10
6.1 [MAC-01] Random transmission time 10
6.2 [MAC-02] Exchange response window 11
6.3 [MAC-03] Installation response window 11
6.4 [MAC-04] Clock synchronization 11
7 Data Link Layer (DLL) test specification 11
7.1 [DLL-01] Length field 12
7.2 [DLL-02] Length field error 12
7.3 [DLL-03] C-field 12
7.4 [DLL-04] CRC field 12
7.5 [DLL-05] CRC field error 12
8 Transport Layer (TPL) test specification 13
8.1 [TPL-01] CI-field - Control Information 13
8.2 [TPL-02] L6Ctrl – L6 control 13
8.3 [TPL-03] L6NetwID - Network identifier 13
8.4 [TPL-04] L6Cpt – Counter 13
8.5 [TPL-05] L6App - Application identifier 14
8.6 [TPL-06] L7 Ciph - Encryption 14
8.7 [TPL-07] L6HashKenc (MAC1) 14
8.8 [TPL-08] L6TStamp – Time Stamp 15
8.9 [TPL-09] L6HashKmac (MAC2) 15
8.10 [TPL-10] Downlink L6HashKmac (MAC2) error 15
8.11 [TPL-11] Downlink L6hashKenc (MAC1) error 15
8.12 [TPL-12] Downlink L6NetwID error 15
8.13 [TPL-11] Downlink L6Ctrl L6KeySel error 16
8.14 [TPL-12] Downlink CI-field error 16

8.15 [TPL-13] Roaming (optional) 16

9 Common Application Layer (APL) test specification 16

9.1 [APL-01] Reception of INSTPONG messages 16
9.2 [APL-02] READPARAMETERS - Read parameters 17
9.3 [APL-03] WRITEPARAMETERS - Write parameters 17
9.4 [APL-04] WRITEKEY - Write Key 17
9.5 [APL-05] EXECINSTPING - Execute installation Ping 17
9.6 [APL-06] CMDERROR – Unsupported command error response 18
9.7 [APL-07] CMDERROR – Supported command error response 18

10 Download Application layer (DWL) test specification 18

10.1 General 18
10.2 [DWL-01] ANNDOWNLOAD – Download announcement 18
10.3 [DWL-02] Block reception 18
10.4 [DWL-03] Verification of complete download 18
10.5 [DWL-03] Activation of new FW 18
10.6 [DWL-03] Fallback in case of failure to boot with new FW 19

11 Test interface for DUT 19

12 Test system 21

Tables and figures

No table of figures entries found.

1 Introduction

This document is the Conformance Test Specification (CTS) outlining the tests to be performed in the Wize Certification process.

The purpose of the Wize Certification process is to test and verify that;

- A device can be connected to a Wize network without negative impact to the network operation or performance (Essential requirements),
- A device will be accepted by the Wize network, and that data can be received (Network compatibility),
- A device meets the performance requirements of the Wize protocol, and that the functionalities of the Wize protocol has been correctly implemented (Wize compliance).

The tests cover all layers of the Wize protocol; PHY (L1), DLL (L2), TPL (L6) and APL (L7). As for the APL, only the Common Application Layers are covered (APP-INSTALL, APP-ADMIN, APP-DOWNLOAD). It will not cover the specific application layers, such as those for Water metering (APP-METER-WTR) or Gas metering (APP-METER-GAS).

In addition to outlining specific tests to be performed, there are also conformance tests and parameters to be declared by the manufacturer.

This CTS specify tests for each protocol layer, in addition to some general requirements and manufacturer declarations.

The table below show how the tests will cover the Essential requirements (step 1), Network compatibility (step 2) and Wize compliance (step 3).

Layer - Tests	Step 1	Step 2	Step 3
General – GEN-xx	All	All	All
Physical Layer – PHY-xx		PHY-01, PHY-02 (WM2400)	All
Medium Access sublayer – MAC-xx		MAC-01 MAC-02 MAC-03	All
Data Link Layer – DLL-xx		All	All
Transport Layer – TPL-xx		All	All
Application Layer – APL-xx			All
Download Application Layer – DWL-xx			Optional

Note: The L6, OSI model presentation layer, is in this CTS referred to as Transport layer to be in line with the layering of EN 13757,

.

2 Standard documents

Reference	Document	Version
N[1]	WIZE - 01 LAN Protocol	v1.2
N[2]	WIZE - 02 Regional Parameters	v1.1
N[3]	WIZE - 03 Common Application Layers	v1.2
N[4]	WIZE – T1 – L6AppID table	v1.0
N[5]	EN 300 220-1 Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods	V3.1.1 2017-02
N[6]	EN 300 220-4 Short Range Devices (SRD) operating in the frequency range 25 MHz to 1 000 MHz; Part 4: Harmonised Standard covering the essential requirements of article 3.2 of the Directive 2014/53/EU; Metering devices operating in designated band 169,400 MHz to 169,475 MHz	V1.1.1 2017-02
N[7]	EN 13757-4 Communication systems for meters – Part 4: Wireless M-Bus	2018
N[8]	WIZE – T2 – NetwID table	v1.0
N[9]	WIZE - Guidelines	V1.0

3 Abbreviations

CTS - Conformance Test Specification
DUT - Device Under Test
GWE - Wize Gateway Emulator

4 General requirements and manufacturer declarations

4.1 [GEN-01] Device information

The manufacturer shall provide at minimum the following information about the device:

- Device address consisting of Manufacturer ID, Serial number, version, and device type
- Network key Kmac
- Unique encryption keys Kenc 1-14
- Kchg (Key change key)
- Klog (Key for FW download)
- L6NetwID, and whether roaming is supported
- If EPOCH2013 is supported in terms of a real time clock (used by L6TStamp)

The Manufacturer ID shall have been registered in the FLAG association to ensure uniqueness.

A test interface shall be provided as specified in section 11.

For deliverables to test FW download, see section 10.1.

4.2 [GEN-02] RED Conformity

The manufacturer shall provide an RED DoC (Radio Equipment Directive – Declaration of Conformity).

The assessment shall have been based on EN 300 220-4.

4.3 [GEN-03] Transmission Duty Cycle

The manufacturer shall specify the transmission duty cycle to show compliance with the Wize duty cycle requirements. Reference N2, section 4 and 4.2 and N8.

Note: The Wize protocol includes in its scope metering applications with battery operated devices and potentially high density of end points. Being a radio communication network, it is therefore very important that the bandwidth sharing is optimized. The requirement in N2 will be changed from 10% to a lower value to improve interoperation in dense networks as described in the reference N8 (WIZE – Guidelines).

The Duty Cycle is calculated as
Number of data frames per day
x bytes per data frame (including preamble, synch and all overhead in addition to the payload)
x 1/Data rate

The Duty Cycle is calculated as an average over a one-day transmission window (duration 20 hours). Any weekly, monthly or event trigged frames shall be included with a daily average value.

The Duty Cycle shall be maximum 0.02% in order to limit radio packet collisions within the Wize network.

Note: Up to 10% duty cycle (maximum 6 minutes per 1 hour) is allowed under EN 300 220-4.

4.4 [GEN-04] Radiated power

The manufacturer shall specify the minimum EIRP, as measured according to ETSI EN 300 220-1:2017 section 5.2, as per EN 300 220-4:2017 section 4.3.1.

To comply with “deep indoor” use cases the EIRP shall be minimum 18 dBm.

For other use cases the EIRP may be lower.

4.5 [GEN-05] Minimum sensitivity

The manufacturer shall specify the minimum sensitivity at WM2400, as measured according to ETSI EN 300 220-1:2017 section 5.14, as per EN 300 220-4:2017 section 4.4.1.

To comply with “deep indoor” use cases the sensitivity shall be minimum -110 dBm (WM2400).

For other use cases the sensitivity may be lower.

4.6 [GEN-06] Random transmission time

The manufacturer shall specify the normal transmission schedule during one day, and the means of randomization of the transmission time in order to minimize the risk of repetitive message collisions between devices.

The randomized transmission time may depend on pseudo random sequences and somewhat related to the serial number but shall have at least a small spread based on a true random value (noise).

This manufacturer statement may replace the MAC test MAC-01.

5 Physical Layer (PHY) test specification

5.1 [PHY-01] Uplink channel

Configure the DUT for uplink channels (RF_UPLINK_CHANNEL, 0x08) to 100, 110, 120, 130, 140, 150. Reference N2, section 4.2 and 4.3.1.

For each setting the following tests shall be done:
Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit an DATA frame,
Trig the GWE to transmit a COMMAND frame and receive the RESPONSE from the DUT.

The uplink channel used for INSTPING, DATA and RESPONSE frame shall be consistent with the configured uplink channel, as verified implicit with the GWE.

5.2 [PHY-02] Uplink modulation (data rate)

Configure the DUT for uplink modulations (RF_UPLINK_MOD, 0x0A) WM2400, WM4800, HSPEED (corresponding to EN 13757-4 data rates 2.4, 4.8 and 6.4 kbps). Reference N2, section 4.1, and 4.3.2, 4.3.3 and 4.3.4.

For each setting the following tests shall be done:
Trig the DUT to transmit an INSTPING frame,

Trig the DUT to transmit an DATA frame,
Trig the GWE to transmit a COMMAND frame and receive the RESPONSE from the DUT.

The uplink channel used for INSTPING, DATA and RESPONSE frame shall be consistent with the configured uplink channel, as verified implicit with the GWE.

5.3 [PHY-03] Downlink channel

Configure the DUT for downlink channels (RF_DOWNLINK_CHANNEL, 0x09) to 100, 110, 120, 130, 140, 150. Reference N2, section 4.1. and 4.3.1.

Set the GWE to respond with the selected downlink channel. Keep the uplink channel fixed during the tests.

For each setting the following tests shall be done:

Trig the DUT to transmit an INSTPING frame, and the GWE respond with an INSTPONG. Verify that the INSTPONG is received by the DUT by reading back the recorded Pongs.

Trig the DUT to transmit a DATA frame, and the GWE to transmit a COMMAND frame on the selected downlink channel. Verify that the RESPONSE from the DUT is transmitted, as verified implicit with the GWE.

The INSTPONG and COMMAND frame shall be received by the DUT when the configured downlink channel corresponds to the selected channel and verified as described above.

5.4 [PHY-04] Downlink modulation (data rate)

Configure the DUT for downlink modulation (RF_DOWNLINK_MOD, 0x0B) to WM2400, WM4800, HSPEED (corresponding to EN 13757-4 data rates 2.4, 4.8 and 6.4 kbps). Set the GWE to respond with the selected downlink modulation. Keep the uplink modulation fixed during the tests. Reference N2, section 4.1.

For each setting the following tests shall be done:

Trig the DUT to transmit an INSTPING frame, and the GWE respond with an INSTPONG. Verify that the INSTPONG is received by reading back the recorded Pongs.

Trig the DUT to transmit a DATA frame, and the GWE to transmit a COMMAND frame using the selected downlink modulation. Verify that the RESPONSE from the DUT is transmitted, as verified implicit with the GWE.

The INSTPONG and COMMAND frame shall be received by the DUT when the configured downlink modulation corresponds to the selected modulation and verified as described above.

5.5 [PHY-05] Frequency accuracy

This test is to verify the RF center frequency accuracy. Reference N2, section 4.3.1.

The nominal frequency is shown in the table below:

Frequency channel number	Central frequency
100	169.406250 MHz
110	169.418750 MHz
120	169.431250 MHz
130	169.443750 MHz
140	169.456250 MHz
150	169.468750 MHz

Use a Spectrum Analyzer (with < 1 ppm reference) to measure the center frequency. If the DUT provide a test mode for unmodulated carrier (CW), this may be used. The duration of the transmission must be short to avoid self-heating of the transmitter which may cause excessive frequency drift.

The RF center frequency shall be within +/- 1.5 kHz of the nominal frequency of the radio channel.

Note: The +/- 1.5 kHz requirement apply to WM4800. At WM2400 the requirement is +/- 2.0 kHz. But in general the same frequency reference (TCXO) is used for all modulations and should meet the strictest tolerance.

5.6 [PHY-06] Frequency adjustment

This test is to verify the possibility to adjust the central frequency of the Device by reading and writing the TX_FREQ_OFFSET parameter (0x??). Reference N1, section 5.5.

Set TX_ FREQ_OFFSET to + 1kHz as per manufacturers instruction. The following transmissions shall use the increased RF frequency as measured with a Spectrum Analyzer before and after the change.

5.7 [PHY-07] Output power management

This test is to verify the possibility to adjust the output power of the Device by reading and writing the TX_POWER parameter (0x10). If the DUT support the reduction of output power, there shall be an automatic mechanism to revert to full output power if the DUT did not receive any COMMAND messages within TX_DELAY_FULLPOWEER (configurable, days). Reference N1, section 5.5.

Set TX_DELAY_FULLPOWER to the minimum. Set the TX_POWER to the minimum (0x02). Verify that the following transmissions use reduced output power. The output power shall revert to maximum power after the configured timeout as measured with a Spectrum Analyzer or inspecting the RSSI of the receive frame at the GWE.

6 Medium Access sublayer (MAC) test specification

6.1 [MAC-01] Random transmission time

This test shall verify the true randomness of the transmission time of DATA frames. Reference N1, section 5.2.

Configure two DUTs to transmit DATA frames over a period of one week, and note the transmission times.

The transmission time shall show true randomness, and the independence of the two DUTs.

This test can be omitted provided a sufficient statement is given in GEN-06.

6.2 [MAC-02] Exchange response window

This test shall verify the response window (listening window of the Device) used for two-way communication. Reference N1, section 5.3.

The response window (listening window) may be set by parameters EXCH_RX_DELAY and EXCH_RX_LENGHT, but shall as a minimum support the Extended Response Delay as specified in the EN 13757-4, which is 5.000 second.

Trig the DUT to transmit a DATA frame,
Prepare the GWE to transmit a COMMAND frame (after a DATA frame is received) and receive the RESPONSE from the DUT.

Note; the GWE COMMAND frame (first preamble bit) must be transmitted exactly 5.0 seconds after the end of the DATA frame (last CRC bit).

6.3 [MAC-03] Installation response window

This test shall verify the timing of the installation sequence (the INSTPONG listening window of the Device). Reference N1, section 5.4.

The INSTPONG listening window may be set by parameters PING_RX_DELAY and PING_RX_LENGHT, but shall as a minimum support a delay of 10 seconds, and a length of 3 seconds.

Trig the DUT to transmit a INSTPING frame,
Prepare the GWE to transmit an INSTPONG frame (after an INSTPING frame is received) and verify the frame is received by the DUT by reading back the recorded Pongs.

Note; the GWE INSTPONG frame (first preamble bit) must be transmitted exactly 10 – 13 seconds after the end of the INSTPING frame (last CRC bit).

6.4 [MAC-04] Clock synchronization

This test is to verify the clock synchronization of the Device to a Head End system by using the COMMAND to read and write CLOCK_CURRENT_EPOCH (@0x20), CLOCK_OFFSET_CORRECTION (@0x21) and CLOCK_DRIFT_CORRECTION (@0x22). Reference N1, section 5.7.1.

Read the CLOCK_CURRENT_EPOCH
Write the CLOCK_CURRENT_EPOCH to a the above read value + 100 (new absolute value)
Read the CLOCK_CURRENT_EPOCH
The last clock reading shall return a value approximately 100 seconds higher.

Read the CLOCK_CURRENT_EPOCH
Write the CLOCK_OFFSET_CORRECTION by +100 (offset to be added)
Read the CLOCK_CURRENT_EPOCH
The last clock reading shall return a value approximately 100 seconds higher.

Read the CLOCK_CURRENT_EPOCH
Write the CLOCK_DRIFT_CORRECTION = (+100, 1) (adding 100 secs every day)
After > 24 hours read the CLOCK_CURRENT_EPOCH
The last clock reading shall return a value approximately 100 seconds higher that the expected time one day later.

7 Data Link Layer (DLL) test specification

7.1 [DLL-01] Length field

This test is to verify the correctness of the Length field. Reference N1, section 4.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND frame and receive the RESPONSE from the DUT.

The Length field of the INSTPING, DATA and RESPONSE frame shall be consistent with the message (bytes counted from C-field to and including the CRC), as verified implicit with the GWE.

7.2 [DLL-02] Length field error

This test is to verify the non-acceptance of in-correctness Length field. Reference N1, section 4.1.

Set the GWE to respond with an error in the Length field (e.g. actual length minus one).

Trig the DUT to transmit an INSTPING frame, and the GWE respond with error in Length field. Verify that the INSTPONG is discarded by the DUT by reading parameters 0x35 to 0x3D (number of INSTPONGs received, followed by 8 INSTPONG values. The INSTPONG with wrong Length field shall not been received.

7.3 [DLL-03] C-field

This test is to verify the correctness of the C-field for different message types. Reference N1, section 4.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND frame and receive the RESPONSE from the DUT.

The C-field of the INSTPING (0x46), DATA (0x44 or 0x54) and RESPONSE (0x08) frame shall be consistent with the message.

7.4 [DLL-04] CRC field

This test shall verify the correct generation of the CRC field. Reference N1, section 4.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Prepare the GWE to transmit a COMMAND frame (after a DATA frame is received) and receive the RESPONSE from the DUT.

The CRC field of the INSTPING, DATA and RESPONSE frame shall be consistent with the message (as per EN 13757-4), as verified implicit with the GWE.

7.5 [DLL-05] CRC field error

This test is to verify the non-acceptance of in-correct CRC field. Reference N1, section 4.1.

Set the GWE to respond with an error in the CRC field (for example, actual value plus one).

Trig the DUT to transmit an INSTPING frame, and the GWE respond with error in CRC field. Verify that the INSTPONG is discarded by the DUT by reading parameters 0x35 to 0x3D (number of INSTPONGs received, followed by 8 INSTPONG values. The INSTPONG with wrong CRC field shall not been received.

8 Transport Layer (TPL) test specification

8.1 [TPL-01] CI-field - Control Information

This test will verify the CI-field is 0x20 to indicate this is a Wize message. Reference N1, section 4.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND frame and receive the RESPONSE from the DUT.

The CI-field of the INSTPING, DATA, and RESPONSE frame shall be 0x20 in all cases.

8.2 [TPL-02] L6Ctrl – L6 control

The L6Ctrl is the concatenation of L6Vers (3 bits protocol version), Reserv (1 reserved bit) and L6KeySel (4 bits key selection). Reference N1, section 6.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND_READPARAMETER frame and receive the RESPONSE from the DUT using Key number 1.
Trig the GWE to transmit a COMMAND_WRITEKEY frame and receive the RESPONSE from the DUT using Key number 15.
Optional test: Trig the GWE to transmit a COMMAND_ANNDOWNLOAD frame and receive the RESPONSE from the DUT using Key number 15.

The L6Ctrl.L6Vers shall be 001 for all versions 1.x (including 1.0, 1,1 and 1.2).
The L6Ctrl.reserved bit shall be 0.
The L6Ctrl.L6KeySel shall be 0 for un-encrypted messages

The L6SCtrl of the frames shall therefore be;
0x20 for INSTPING
0x21 for DATA
0x21 for RESPONSE_READPARAMETER
0x2F for RESPONSE_WRITEKEY
0x2F for RESPONSE_ANNDOWNLOAD

8.3 [TPL-03] L6NetwID - Network identifier

The L6NetwID is the network identifier, using the L6NETWID (@0x2A) in the uplink, and accepting only the same value in the downlink. Reference N1, section 6.1. See also Reference N8 for a list of L6NetwIDs.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using the same L6NETWID) and receive the RESPONSE from the DUT.

The L6NetwID of the frames shall be the same as L6NETWID (@0x2A).

8.4 [TPL-04] L6Cpt – Counter

The L6Cpt is the message counter, incremented for every new message. Response messages shall use the same counter value as the request. Reference N1, section 6.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using the same L6NETWID) and receive the RESPONSE from the DUT.

Repeat the frame transmissions above.

The L6Cpt of the INSTPING frame shall increment from the first to the second transmission.
The L6Cpt of the DATA frame shall increment from the first to the second transmission.
The L6Cpt of the RESPONSE frame shall be the same as of the COMMAND frame in the first and in the second transmission (and also incrementing).

8.5 [TPL-05] L6App - Application identifier

The L6App is the application identifier, using the L6APP (@04) in the uplink, and accepting only the same value in the downlink. Reference N1, section 6.1, and N3 and N4.

NOTE: APP-INSTALL (0xFC) and APP-ADMIN (0xFD) is not required used, and the specific application layer L6App value may be used because the C-field will identify the application in this case.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using the same L6APP) and receive the RESPONSE from the DUT.

The L6App of the frames shall be the same as L6APP (@04).

8.6 [TPL-06] L7 Ciph - Encryption

The application frame shall be encrypted by AES-128-CTR using a 16 byte Key. Reference N1, section 6.1 Figure 24, and section 7 Appendix A.

Trig the DUT to transmit an INSTPING frame, L6KeySel = 0 (un-encrypted)
Trig the DUT to transmit a DATA frame, L6KeySel = 1
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) and receive the RESPONSE from the DUT.

The L7Ciph frame
-shall not be encrypted for the INSPING frame
-shall be encrypted for the DATA and RESPONSE frame
-shall be successfully decrypted by the GWE for all frames

8.7 [TPL-07] L6HashKenc (MAC1)

The L6 control information and application data frame shall be authenticated by AES-128-CMAC using a 16 byte Key to generate a 4 byte Message Authentication Code (MAC1), called L6HashKenc. Reference N1, section 6.1 Figure 24, and section 8 Appendix B.

Trig the DUT to transmit an INSTPING frame, L6KeySel = 0 (un-encrypted), Kmac used to generate MAC1
Trig the DUT to transmit a DATA frame, L6KeySel = 1
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) and receive the RESPONSE from the DUT.

The L6HashKenc
- shall be successfully authenticated by the GWE for INSTPING frame (using Kmac, not Kenc)
-shall be successfully authenticated by the GWE for the DATA and RESPONSE frame (using Kenc selected

by L6KeySel = 1).

8.8 [TPL-08] L6TStamp – Time Stamp

The L6 frame contains a timestamp, L6TStamp, which is made up of the two LSB of the EPOCH2013. Reference N1, section 6.1 Figure 24, and Table 12.

Trig the DUT to transmit an INSTPING frame, L6KeySel = 0 (un-encrypted)
Trig the DUT to transmit a DATA frame, L6KeySel = 1
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) and receive the RESPONSE from the DUT.

After a few seconds (e.g. 5 seconds) repeat the frame transmissions.

The L6TStamp shall increment for each frame transmission, corresponding to the time (in seconds) between the transmissions.

NOTE: If the device does not contain an RTC, the L6TStamp shall be 0x0000 for all frames.

8.9 [TPL-09] L6HashKmac (MAC2)

The L6 control information, application data frame and Tijme stamp shall be authenticated by AES-128-CMAC using a 16 byte Key (Kmac) to generate a 2 byte Message Authentication Code (MAC2), called L6HashKmac. Reference N1, section 6.1 Figure 24, and section 8 Appendix B.

Trig the DUT to transmit an INSTPING frame, L6KeySel = 0 (un-encrypted)
Trig the DUT to transmit a DATA frame, L6KeySel = 1
Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) and receive the RESPONSE from the DUT.

The L6HashKmac shall be successfully authenticated by the GWE for all frames (using Kmac).

8.10 [TPL-10] Downlink L6HashKmac (MAC2) error

Referring to test TPL-09, an invalid MAC shall not be accepted by the device.

Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) with an error in the L6HashKmac (e.g. L6HashKmac + 1) and attempt to receive the RESPONSE from the DUT.

The L6HashKmac shall be rejected by the DUT (using Kmac), and no RESPONSE frame shall be generated.

8.11 [TPL-11] Downlink L6hashKenc (MAC1) error

Referring to test TPL-07, an invalid MAC shall not be accepted by the device.

Trig the GWE to transmit a COMMAND_READPARAMETER frame (using L6KeySel =1) with an error in the L6HashKenc (e.g. L6HashKenc + 1) and attempt to receive the RESPONSE from the DUT.

The L6HashKenc shall be rejected by the DUT (using Kenc), and no RESPONSE frame shall be generated.

8.12 [TPL-12] Downlink L6NetwID error

The L6NetwID is the network identifier, using the L6NETWID (@2A) in the uplink, and accepting only the same value in the downlink. Reference N1, section 6.1.

Trig the DUT to transmit an INSTPING frame,
Trig the DUT to transmit a DATA frame,

Trig the GWE to transmit a COMMAND_READPARAMETER frame (using a different L6NETWID, e.g. L6NETWID+1) and attempt to receive the RESPONSE from the DUT.

The COMMAND frame shall not be accepted when L6NetwID is different from L6NETWID (@2A), and no RESPONSE frame shall be generated.

8.13 [TPL-11] Downlink L6Ctrl L6KeySel error

Referring to test TPL-02, an invalid L6KeySel shall not be accepted by the device. Reference N1, section 6.1.

Trig the GWE to transmit a COMMAND_READPARAMETER frame using L6KeySel = 2 and attempt to receive the RESPONSE from the DUT. The DUT should be using and expecting L6KeySel = 1.

Trig the GWE to transmit a COMMAND_WRITEKEY frame using L6KeySel = 2 and attempt to receive the RESPONSE from the DUT. The DUT should be expecting L6KeySel = 15 (Kchg).

Optional test: Trig the GWE to transmit a COMMAND_ANNDOWNLOAD frame using L6KeySel = 2 and attempt to receive the RESPONSE from the DUT. The DUT should be expecting L6KeySel = 15 (Kchg).

The COMMAND frames shall not be accepted when an illegal key is used, and no RESPONSE frame shall be generated.

8.14 [TPL-12] Downlink CI-field error

Referring to test TPL-01, an invalid CI-field (e.g. a non-Wize frame) shall not be accepted by the device. Reference N1, section 4.1.

Trig the GWE to transmit a COMMAND_READPARAMETER frame using L6KeySel = 1 and attempt to receive the RESPONSE from the DUT. The DUT should be using and expecting L6KeySel = 1.

Trig the GWE to transmit a COMMAND_WRITEKEY frame using L6KeySel = 15 and attempt to receive the RESPONSE from the DUT. The DUT should be expecting L6KeySel = 15 (Kchg).

Optional test: Trig the GWE to transmit a COMMAND_ANNDOWNLOAD frame using L6KeySel = 15 and attempt to receive the RESPONSE from the DUT. The DUT should be expecting L6KeySel = 15 (Kchg).

The COMMAND frames shall not be accepted when a different CI-field (not 0x20) is used, and no RESPONSE frame shall be generated.

8.15 [TPL-13] Roaming (optional)

In order to support roaming between different networks, the gateway would be able to handle End Devices with different L6NetwIDs. Currently there is no requirements to the DUT to enable roaming (Reference N1). Roaming is planned to be added to LAN specification v2.0 (Reference N1, updated to v2.0).

9 Common Application Layer (APL) test specification

9.1 [APL-01] Reception of INSTPONG messages

This test is to verify that the DUT can receive multiple INSTPONG messages and store the associated data. Reference N1, section 5.4.

After a INSTPING/INSTPONG sequence where the GWE transmit several INSTPONGS with unique IDs, the recorded INSTPONGS shall be readable using the COMMAND_READPARAMETERS, reading parameters 0x35 to 0x3D (number of INSTPONGs received, followed by 8 INSTPONG values). The values

of the parameters shall correspond to the actual INSTPONG test messages.

9.2 [APL-02] READPARAMETERS - Read parameters

Reference N3, section 4.4.1.1 (COMMAND) / 4.4.2.3 (RESPONSE).

Trig the GWE to transmit a COMMAND_READPARAMETER frame and receive the RESPONSE from the DUT. Request parameter RF_UPLINK_CHANNEL (@08) and RF_DOWNLONK_CHANNEL (@09),

The RESPONSE frame shall
-have L7ErrorCode = 0x00 (no error).
-contain ID and value of RF_UPLINK_CHANNEL and RF_DOWNLONK_CHANNEL corresponding to the actual uplink/downlink channel.

9.3 [APL-03] WRITEPARAMETERS - Write parameters

Reference N3, section 4.4.1.2 (COMMAND) / 4.4.2.4 (RESPONSE).

Trig the GWE to transmit a COMMAND_WRITEPARAMETER frame and receive the RESPONSE from the DUT. Request to set parameter RF_UPLINK_CHANNEL (@08) and RF_DOWNLONK_CHANNEL (@09) to new values (e.g. one channel higher).

The RESPONSE frame shall
-have L7ErrorCode = 0x00 (no error).
Repeat the test APL-02 with the GWE using the new radio channel for uplink /downlink. The test shall be successful reporting the new channels.

9.4 [APL-04] WRITEKEY - Write Key

The protocol allows for changing the key for any local communication interface (Kmob) and the network key Kmac. Reference N3, section 4.4.1.3 (COMMAND) / 4.4.2.5 (RESPONSE). The test of writing of Kmob is not required since any local interface is manufacturer specific.

Trig the GWE to transmit a COMMAND_WRITEKEY frame and receive the RESPONSE from the DUT. Request to set parameter Kmac (L7KeyID = 0x02) to a new value (new random key). L7KIndex = 0x00.

The RESPONSE frame shall have L7ErrorCode = 0x00 (no error).

Repeat the test APL-02 with the GWE using the new Kmac. The test shall be successful reporting the new channels.

9.5 [APL-05] EXECINSTPING - Execute installation Ping

The EXECINSTPING command shall trig the DUT to start an Ping/Pong sequence. After the sequence, the RESPONSE_EXECINSTPING message shall be transmitted containing the INSTPONG information. Reference N3, section 4.3.1 / 4.4.1.5 (COMMAND) / 4.4.2.7 (RESPONSE).

Trig the GWE to transmit a COMMAND_EXECINSTPING frame.

The DUT shall start the Ping/pong sequence by transmitting the INSTPONG. The GWE should then send an INSTPONG, before receiving the RESPONSE_EXECINSTPING from the DUT.

The RESPONSE_EXECINSTPING frame
-shall have L7ErrorCode = 0x00 (no error)
-shall contain L7Pong1 corresponding to the INSTPONG transmitted by the GWE

9.6 [APL-06] CMDERROR – Unsupported command error response

In case of an error in the command/response sequence, the DUT shall report an error. Reference N3, section 4.4.2.1 (RESPONSE_CMDERROR).

Trig the GWE to transmit a COMMAND frame using L7CommandID = 0xEE (not supported) and receive the RESPONSE from the DUT.

The RESPONSE frame shall
-have L7ResponseID = 0xEE (same as requested).
-have L7ErrorCode = 0xFF (unsupported command)

9.7 [APL-07] CMDERROR – Supported command error response

In case of an error in the command/response sequence, the DUT shall report an error. Reference N3, section 4.4.2.2.

Trig the GWE to transmit a COMMAND_WRITEPARAMETER frame and receive the RESPONSE from the DUT. Request to set parameter VERS_HW_TRX (@01) and VERS_FW_TRX (@02) to any values (e.g. 0x0000).

The RESPONSE frame shall
-have L7ErrorCode = 0x02 (attempt to write a “read only” parameter)
-have L7ErrorParam = 0x01 (first parameter that failed)

10 Download Application layer (DWL) test specification

10.1 General

The Firmware Download use a different Data Link Layer frame format (LLC-DOWNLOAD) for downlink broadcast. In this case the first byte (Length field) is set to 0xFF which indicate this is a Download frame. The L6 and L7 (transport and application layer) of this frame is manufacturer specific, as well as the Forward Error Correction, if any. Reference N1 section 6.2.

The scheduling and preparation of the Firmware download session is done by the Download announcement command (ANNDOWNLOAD).

The format of the new FW (fragmented to be sent in multiple messages) is manufacturer specific.

The mechanism to verify Download announcement, Block reception, verification of complete download, activation, and fallback, is manufacturer specific.

The manufacturer shall provide a new “good” firmware image, and a “bad” image (in order to test the fallback mechanism), together with instructions how to use the FW download.

10.2 [DWL-01] ANNDOWNLOAD – Download announcement

Reference N3, section 4.4.1.4 / 4.4.2.6.

10.3 [DWL-02] Block reception

This test shall verify the reception of one block provided the L2Dwnld is correct.

10.4 [DWL-03] Verification of complete download

This test shall verify the completeness of the reception of all blocks.

10.5 [DWL-03] Activation of new FW

This test shall verify the activation of the new FW by reading the new FW revision.

10.6 [DWL-03] Fallback in case of failure to boot with new FW

This test shall verify fallback to previous FW revision if the booting of new FW failed. By using a failing FW it shall be tested that after attempt to booth with new FW, it shall fall back and use the previous version.

11 Test interface for DUT

11.1 Test interface overview

In order to perform the above tests, the Device must provide a standardized test command interface. The interface to the actual DUT can be a UART / RS232 interface, an IrDa interface, NFC interface, or similar, but the SW interface shall be a standardized Python script API. The Python drivers shall be provided by the manufacturer for sending commands to the Device. The software shall run on a Windows PC, or specific SW/HW must be provided to be used during the tests.

The minimum set of commands are:

- Set RF channel
- Set RF data rate
- Set RF output power
- Trig the transmission of INSTPING RF frames
- Trig the transmission of regular DATA RF frames (SND-NR). Application data may be “dummy data”.

Note; the Device shall send a RESPONSE frame, when a valid COMMAND frame is received. The COMMAND frame will be used to test the RESPONSE frame.

11.2 Python API

A Python language API shall be used for the DUT test interface. Below is shown the class definitions for the API.

```
_____________________________________________________________________
class WizeTestAPI:
    # Maps from channel number to frequency
    channel_map = { "100" : "169.406250",
                    "110" : "169.418750",
                    "120" : "169.431250",
                    "130" : "169.443750",
                    "140" : "169.456250",
                    "150" : "169.468750"
}

    # Maps from data rate standard to bits per second
    data_rate_map = { "WM2400" : "2400",
                      "WM4800" : "4800",
                      "HSPEED" : "6400" }

    def __init__(self):
        # Only do printout for this example. Implement specific (HW) driver here
        print("Init")


    def default_callback(self, response):
        # Only do printout for this example. Implement specific (HW) driver here
        print(response)


    def set_rf_channel(self, channel = "100"):
        try:
```

```
            frequency=self.channel_map[channel]
        except:
            print("No such channel. Using default 100.")
            frequency=self.channel_map["100"]

        # Only do printout for this example. Implement specific (HW) driver here
        print("Set RF Channel to channel {} ({} MHz)".format(channel, frequency))


    def set_rf_data_rate(self, data_rate = "WM2400"):
        try:
            bps=self.data_rate_map[data_rate]
        except:
            print("No such data rate specification. Using default WM2400.")
            bps=self.data_rate_map["WM2400"]

        # Only do printout for this example. Implement specific (HW) driver here
        print("Set RF data rate to {} ({} bps)".format(data_rate, bps))


    def set_rf_output_power(self, power = "0"):
        # Only do printout for this example. Implement specific (HW) driver here
        print("Set RF output power to {}".format(power))


    def trigger_rf_instping(self, response_callback = default_callback):
        print("Triggering RF instping")
        # Only do printout for this example. Implement specific (HW) driver here
        response = "Response from test system"
        response_callback(self, response)


    def trigger_rf_data_frame(self, data=b"abcd", response_callback =
default_callback):
        print("Triggering RF data frame with the following data: \n{}".format(data))
        # Only do printout for this example. Implement specific (HW) driver here
        response = "Response from test system"
        response_callback(self, response)



print("\n\n*********************************************")
print("Starting tests")
print("*********************************************\n")
#These are the actual tests
wize_tests = WizeTestAPI()

print("\n\n*********************************************")
print("Now testing with default values")
print("*********************************************\n")


# With default values
wize_tests.set_rf_channel()
wize_tests.set_rf_data_rate()
wize_tests.set_rf_output_power()
wize_tests.trigger_rf_instping()
wize_tests.trigger_rf_data_frame()

print("\n\n*********************************************")
print("Now testing with custom values")
print("*********************************************\n")

# With custom values
def my_instping_callback(self, response):
    print("Custom instping callback : {}".format(response))

def my_data_frame_callback(self, response):
    print("Custom data_frame_callback : {}".format(response))

wize_tests.set_rf_channel("110")
wize_tests.set_rf_data_rate("WN4800")
```

```
wize_tests.set_rf_output_power("1")
wize_tests.trigger_rf_instping(my_instping_callback)
wize_tests.trigger_rf_data_frame(data=b"CustomTest",
response_callback=my_data_frame_callback)
```

---

12 Test system

The test system consists of the Device Under Test, a PC to control the DUT, and a Wize Gateway Emulator (GWE) including both HW and SW.

The PC to control the DUT shall be equipped with software and interface to control the DUT as per manufacturers specification.

Optionally a Wize frame sniffer may be used to monitor and log the RF frames communicated between the DUT and the GWE.

The Wize Gateway Emulator shall be a Wize RF transceiver with control software in order to

- Receive INSTPING and respond with INSTPONG
- Receive DATA frames
- Send COMMAND frames and receive the RESPONSE

The Wize RF transceiver shall

- Support all 6 RF channels (configurable)
- Support all 3 data rates (configurable)
- Verify MAC1 and MAC2
- Decrypt received messages
- Transmit COMMAND frame exactly 5.0 seconds after receiving a DATA frame
- Transmit INSTPONG frame between 10 and 13 seconds after receiving an INSTPING frame

Special features shall be provided in order to

- Send incorrect Length field
- Send incorrect C-field (in INSTPONG and COMMAND frames)
- Send incorrect CRC field
- Send incorrect L6HashKenc
- Send incorrect L6HashKmac
- Adding a timestamp to the received messages with resolution in ms
- Transmit DOWNLOAD frames at a specific time (EPOCH 2013) and a specific interval (10 sec)
- Transmit several INSTPONG messages within the 10-13 seconds window to emulate multiple gateways

END OF DOCUMENT