

1 **LoRaWAN Certification Protocol Specification TS009-1.0.0**

2 Copyright © 2020 LoRa Alliance, Inc. All rights reserved.

3

4 **NOTICE OF USE AND DISCLOSURE**

5

6 Copyright © LoRa Alliance, Inc. (2020). All Rights Reserved. The information within this
7 document is the property of the LoRa Alliance (“The Alliance”) and its use and disclosure are
8 subject to LoRa Alliance Corporate Bylaws, Intellectual Property Rights (IPR) Policy and
9 Membership Agreements.

10

11 Elements of LoRa Alliance specifications may be subject to third-party intellectual property
12 rights, including without limitation, patent, copyright or trademark rights (such a third party
13 may or may not be a member of the LoRa Alliance). The Alliance is not responsible and
14 shall not be held responsible in any manner for identifying or failing to identify any or all such
15 third-party intellectual property rights.

16

17 This document and the information contained herein are provided on an “AS IS” basis and
18 THE ALLIANCE DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING
19 BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION
20 HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT
21 LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT,
22 COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF
23 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR
24 NONINFRINGEMENT. IN NO EVENT WILL THE ALLIANCE BE LIABLE FOR ANY LOSS
25 OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF
26 BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY,
27 INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN
28 CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE
29 INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF
30 SUCH LOSS OR DAMAGE.

31

32 The above notice and this paragraph must be included on all copies of this document.

33

34 LoRa Alliance®
35 5177 Brandin Court
36 Fremont, CA 94538
37 United States

38

39 Note: LoRa Alliance® and LoRaWAN® are trademarks of the LoRa Alliance, used by permission. All
40 company, brand and product names may be trademarks that are the sole property of their respective
41 owners.

42

43



44

45

46

LoRaWAN Certification Protocol Specification TS009-1.0.0

47

48

49

50 **Authored by the Certification Working Group of the LoRa Alliance Technical**
51 **Committee**

52

53 **Technical Committee Chair and Vice-Chair:**

54 A.YEGIN(Actility), O.SELLER (Semtech)

55

56 **Working Group Chairs:**

57 D.HUNT (LoRa Alliance)

58

59 **Editor:**

60 M.LUIS (Semtech)

61

62 **Contributors:**

63 N. BHASKARAN (LoRa Alliance), D.HUNT(LoRa Alliance), M.LUIS (Semtech),

64 C.NFOR (7 Layers), J.PETRUNO (Senet)

65

66 **Version:** 1.0.0

67 **Date:** October 2020

68 **Status:** Released

69

70	Contents	
71	1	Conventions..... 5
72	2	Introduction..... 6
73	2.1	Scope of LoRaWAN Certification 7
74	2.2	LoRaWAN Certification Process..... 7
75	3	Functional Test Description for LoRaWAN Certification 8
76	4	End-Device Certification Description..... 10
77	4.1	Overview..... 10
78	4.1.1	Over-The-Air DUT 10
79	4.1.2	Personalized DUT 10
80	4.2	Certification Commands..... 10
81	4.2.1	Downlink Counter..... 10
82	4.2.2	Commands handling 10
83	5	Certification Protocol Commands..... 11
84	5.1	Package Version Commands (<i>PackageVersionReq</i> , <i>PackageVersionAns</i>)..... 13
85	5.2	DUT Reset Command (<i>DutResetReq</i>) 13
86	5.3	Dut JoinReq Command (<i>DutJoinReq</i>) 13
87	5.4	LoRaWAN Class Selection Command (<i>SwitchClassReq</i>) 13
88	5.5	ADR Control Command (<i>AdrBitChangeReq</i>)..... 14
89	5.6	Regional Duty-Cycle Enforcement Command (<i>RegionalDutyCycleCtrlReq</i>) 14
90	5.7	Application Transmission Periodicity Control Command
91		(<i>TxPeriodicityChangeReq</i>) 14
92	5.8	Uplink Frames Control Command (<i>TxFramesCtrlReq</i>) 15
93	5.9	Echo Frame Request Commands (<i>EchoIncPayloadReq</i> , <i>EchoIncPayloadAns</i>) 15
94	5.10	Applicative Rx Counter Commands (<i>RxAppCntReq</i> , <i>RxAppCntAns</i> ,
95		<i>RxAppCntRstReq</i>)..... 16
96	5.11	Link Check Request Command (<i>LinkCheckReq</i>)..... 16
97	5.12	DeviceTimeReq MAC Command (<i>DeviceTimeReq</i>) 17
98	5.13	PingSlotInfoReq MAC Command (<i>PingSlotInfoReq</i>) 17
99	5.14	Transmit Continuous Wave Request Command (<i>TxCwReq</i>) 17
100	5.15	DUT Disables FPort 224 (<i>DutFPort224DisableReq</i>) The
101		<i>DutFPort224DisableReq</i> command has no payload 17
102	5.16	DUT Versions Command (<i>DutVersionsReq</i> , <i>DutVersionsAns</i>) 18
103	6	Glossary 19
104	7	Bibliography 20
105	7.1	References..... 20
106		

107	Tables	
108	Table 1:	Control messages format 11
109	Table 2:	Certification protocol messages summary 12
110	Table 3:	PackageVersionAns 13
111	Table 4:	SwitchClassReq fields 13
112	Table 5:	AdrBitChangeReq fields 14
113	Table 6:	RegionalDutyCycleCtrlReq fields..... 14
114	Table 7:	TxPeriodicityChangeReq fields 14
115	Table 8:	Periodicity field encoding..... 15
116	Table 9:	TxFramesCtrlReq fields..... 15
117	Table 10:	FrameType values description..... 15
118	Table 11:	EchoIncPayloadReq..... 16
119	Table 12:	RxAppCntAns fields 16

120	Table 13: PingSlotInfoReq fields	17
121	Table 14: TxCwReq fields	17
122	Table 15: DutVersionsAns fields	18

123

124 **Figures**

125	Figure 1: Certification Process Flow Chart	6
126	Figure 2: Test Harness Architecture	8

127

128 **1 Conventions**

129

 130 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
 131 "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in
 132 this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and
 133 only when, they appear in all capitals, as shown here.

134

 135 The tables in this document are normative. The figures in this document are informative. The
 136 notes in this document are informative.

137

 138 Commands are written *PackageVersionReq*, bits and bit fields are written
 139 *PackageIdentifier*, constants are written RECEIVE_DELAY1, variables are written *N*.

140 In this document,

- 141 • The octet order for all multi-octet fields SHALL be little endian.
-
- 142 • EUI are 8-octet fields and SHALL be transmitted as little endian.
-
- 143 • By default, RFU bits are Reserved for Future Use and SHALL be set to 0 by the transmitter
-
- 144 of the packet and SHALL be silently ignored by the receiver.

145

146 **2 Introduction**

147 All messages described in this document are transported as application layer messages on
 148 a dedicated port. As such, all unicast messages (uplink or downlink) are encrypted by the
 149 LoRaWAN MAC layer using the end-device’s AppSKey.

150
 151 This protocol specification allows a Certification test harness to fully validate compliance of
 152 the end-device to the LoRaWAN Link Layer Specification [TS001-1.0.4] and the LoRaWAN
 153 Regional Parameters Specification [RP002-1.0.1] for Class A end devices.

154
 155
 156 **In order for an end-device to be designated “LoRaWAN Certified^{CM}” it SHALL**
 157 **implement this application layer specification and have the FPort 224 enabled for the**
 158 **duration of the Certification tests. This application SHALL be disabled on any device**
 159 **in production, otherwise it may be intentionally or accidentally be activated to harm**
 160 **the device itself or the networks in its radio coverage.**

161
 162 The end-device to be certified SHALL be sent to the Authorized Test House (ATH) with the
 163 FPort enabled for the Certification test, then returned to the end-device manufacturer for
 164 the FPort to be disabled.

165
 166
 167
 168 Flow chart for process

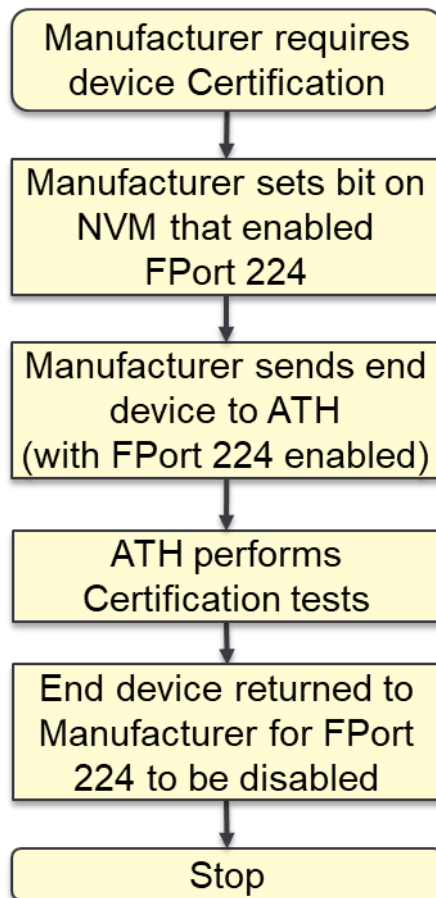


Figure 1: Certification Process Flow Chart

169
 170

171
172 When an OTAA device has connected to the Test Control Layer (**TCL refer to Chapter 3**)
173 after the Join-Request and Join-Accept frame exchange the device SHOULD send an uplink
174 message as soon as possible.

175
176 When an ABP device has connected to the (**TCL**) there SHALL be a way to trigger an uplink
177 message.

178 **2.1 Scope of LoRaWAN Certification**

179 The scope of this specification is limited to validating compliant implementation of the
180 LoRaWAN protocol.

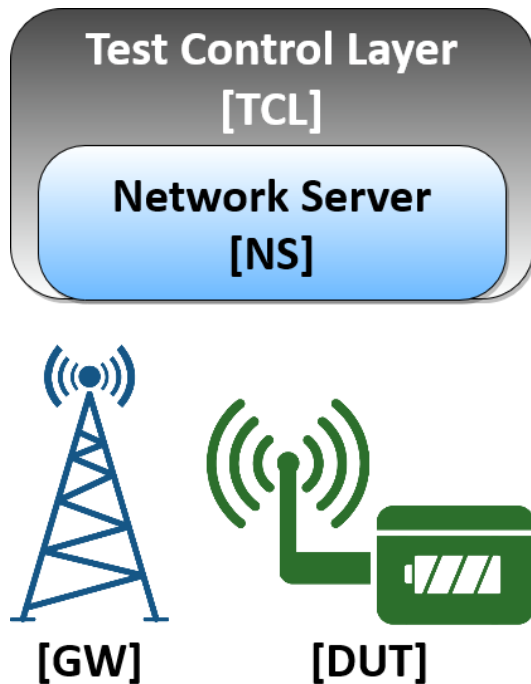
181 Intended or otherwise, the inevitable variability of performance and quality of the radio
182 implementation among end-devices is too high to allow normalized, practical evaluation. RF
183 performance measurement, whether radiated or conducted, which are therefore considered
184 out of scope of the tests described herein. Subsequently, all methodologies describing RF
185 provisioning or adjustments (e.g. device attenuation, etc.) are intentionally absent. It is the
186 shared responsibility of the Authorized Test Houses and those parties seeking certification to
187 best accommodate the submitted end-device RF characteristics for LoRaWAN protocol
188 certification. This is intended to optimize reliability and consistency of bi-directional
189 communication of the test harness.

190 **2.2 LoRaWAN Certification Process**

191 For details of the LoRaWAN Certification Process see:
192 <https://lora-alliance.org/lorawan-certification>

193 **3 Functional Test Description for LoRaWAN Certification**

194 The list of tests specified on Regional Certification documents reflect functional requirements
 195 of an end-device as defined by the targeted LoRaWAN Specification. The tests are conducted
 196 in a test harness generally comprised of:



- A Test Control Layer [TCL]
- A LoRaWAN Network Server [NS]
- A LoRa gateway [GW]
- The end-device Under Test [DUT]

Figure 2: Test Harness Architecture

197 Implementation of this harness architecture is expected to vary among test houses. The Test
 198 Control Layer [TCL] is assumed to be a framework of automated scripts and tools that
 199 manipulates the LoRaWAN Network Server [NS] to facilitate the tests. Specifically, the TCL
 200 drives events in the harness, controlling application and network-control content of downlinks.
 201 It also decrypts, inspects and validates content of uplinks sent by the DUT. This allows test
 202 coverage to include:

- Cryptography
- Timing of the DUT Receive Windows
- Frequency Channel usage and Data Rate adaptation
- Maximum Payload length handling

208 For brevity, this document makes procedural reference to only the **TCL**, **NS**, **GW** and **DUT**.

209 The LoRaWAN gateway [GW] and DUT are collocated in an RF-isolated environment,
 210 provisioned as necessary for reliable bi-directional communication. It is nonetheless expected
 211 that both the DUT and GW will not receive every frame intended for reception. The TCL
 212 SHOULD make reasonable effort to accommodate this inevitability.

213 The DUT is required to implement this LoRaWAN Certification Protocol Specification in order
 214 to provide a way to control DUT application. The RF-isolated environment mentioned above
 215 SHOULD mitigate any potential interference.

216

217 Testing occurs to certify the **DUT** for each supported activation method, be it over-the-air
218 activation (**OTAA**), activation-by-personalized (**ABP**), or both.

219 Between each test section described in specific regional documents, the **TCL** will return the
220 **DUT** to a known state.

221 The **TCL** SHALL verify the following throughout the course of this Certification test suite:

- 222 • The **DUT** uplink frames size SHALL respect the maximum allowed uplink frame size
223 for the data rate currently in use.
- 224 • The size of the **DUT**'s uplink frames SHALL match the expected content to ensure no
225 extraneous and unnecessary content is present.

226 4 End-Device Certification Description

227 4.1 Overview

228 Every LoRaWAN end-device SHALL implement this applicative protocol specification in its
 229 application layer. This allows the test harness to fully validate compliance of the end-
 230 device's LoRaWAN MAC layer implementation.

231 The Port Field (F_{Port}) value 224 is dedicated to the LoRaWAN MAC layer certification
 232 protocol.

233 The **DUT** SHALL return to its normal application behaviour by a command disabling
 234 certification F_{Port} 224 processing. Additionally, the **DUT** SHALL be reset with a dedicated
 235 command, returning to a join state from which it can then establish a new session.

236 All defined test commands SHALL be sent by **TCL** to the **DUT** using F_{Port} 224 when the
 237 end-device is in its normal operation. The **DUT** SHOULD execute the given command as
 238 soon as possible.

239 4.1.1 Over-The-Air Activated DUT

240 When the **DUT** is first powered up if it uses OTAA it SHOULD join the network by issuing a
 241 Join-request, the **TCL** will respond with Join-accept. For best practice operation of the
 242 Certification process, upon receipt of the Join-accept frame the **DUT** SHOULD then send a
 243 (possibly empty) uplink frame. The **TCL** is then able to send certification test commands on
 244 F_{Port} 224 if enabled.

245 4.1.2 Activated by Personalization DUT

246 A Personalized **DUT** is one that comes with session keys pre-programmed. The
 247 personalization information SHALL be supplied to the Test House. When the **DUT** is first
 248 powered on, it SHOULD send an uplink frame. The **TCL** is then able to send certification test
 249 commands on F_{Port} 224 if enabled.

250 4.2 Certification Commands

251 4.2.1 Downlink Counter

252 The **DUT** creates a 16-bit unsigned counter (called $RxAppCnt$) which is incremented each
 253 time the **DUT** receives an applicative downlink frame ($F_{Port} > 0$). An empty downlink frame
 254 with F_{Ctrl} ACK bit set SHALL be considered as and applicative downlink.

255 The $RxAppCnt$ counter SHALL be initialized to 0 when **DUT** is reset or each time the **TCL**
 256 sends a *DownlinkCntRstReq* command on F_{Port} 224.

257 4.2.2 Commands handling

258 The **TCL** MAY send the certification application commands at any given time. The **DUT**
 259 SHOULD execute a command as soon as possible.

260 **5 Certification Protocol Commands**

261 The `PackageIdentifier` of the certification protocol transport package is 6. The
 262 `PackageVersion` of this package is version 1.
 263

264 **Note:** This version of the package is not compatible with any previous
 265 version of the Certification Protocol used with LW1.0.2 and earlier
 266 releases.
 267

268 This package supports all the commands necessary to execute the LoRaWAN end-device
 269 certification tests. The port value is 224 (see [TS008] FPort Assignments). This port SHALL
 270 NOT be used for any other purposes.
 271

272 All certification protocol command messages are exchanged on this port using application
 273 payload and encrypted using the end-device's `AppSKey`. All certification protocol command
 274 messages use the same format:
 275

Certification protocol command	Certification protocol command Payload
--------------------------------------	---

276 **Table 1: Certification protocol command messages format**

277 A frame SHALL NOT carry more than one Certification protocol command message. The
 278 length of Certification protocol command payload can be determined unambiguously as a
 279 function of the command.
 280

281 The following table summarizes the list of Certification protocol command messages.

CID	Certification protocol command name	Transmitted by		Short Description
		End-device	server	
0x00	PackageVersionReq		x	Used by the TCL to request the package version implemented by the end-device
0x00	PackageVersionAns	x		Conveys the answer to PackageVersionReq
0x01	DutResetReq		x	DUT SHALL reset the MCU
0x02	DutJoinReq		x	DUT SHALL start issuing Join-Request messages
0x03	SwitchClassReq		x	DUT SHALL change its Class of operation to A, B or C
0x04	AdrBitChangeReq		x	DUT SHALL activate/deactivate ADR
0x05	RegionalDutyCycleCtrlReq		x	DUT SHALL activate/deactivate the regional band duty-cycle enforcement
0x06	TxPeriodicityChangeReq		x	DUT SHALL change its uplink periodicity to the provided value
0x07	TxFramesCtrlReq		x	All subsequent DUT uplinks SHALL be of specified type
0x08	EchoPayloadReq		x	TCL requests the DUT to echo the provided payload where each byte is incremented by 1
0x08	EchoPayloadAns	x		Conveys the answer to EchoPayloadReq request
0x09	RxAppCntReq		x	TCL requests the DUT to provide the current applicative RxAppCnt value
0x09	RxAppCntAns	x		Conveys the answer to RxAppCntReq request
0x0A	RxAppCntResetReq		x	DUT SHALL reset the applicative RxAppCnt value to 0
0x0B-0x1F	RFU			
0x20	LinkCheckReq		x	DUT SHALL send a LinkCheckReq MAC command to the TCL
0x21	DeviceTimeReq		x	DUT SHALL send a DeviceTimeReq MAC command to the TCL
0x22	PingSlotInfoReq		x	DUT SHALL send a PingSlotInfoReq MAC command to the TCL Only required for Class B DUT
0x23-0x7C	RFU			
0x7D	TxCwReq		x	DUT SHALL set the radio in continuous wave transmission mode
0x7E	DutFPort224DisableReq		x	DUT SHALL disable the processing of data received on FPort 224
0x7F	DutVersionsReq		x	TCL requests the DUT to send its firmware version, LoRaWAN version and Regional parameters version
0x7F	DutVersionsAns	x		Conveys the answer to DutVersionsReq request
0x80 - 0xFF	Proprietary	x	x	Reserved for proprietary end-device command extensions

283 **5.1 Package Version Commands (*PackageVersionReq*,**
 284 ***PackageVersionAns*)**

285
 286 The *PackageVersionReq* command has no payload.

287
 288 The end-device SHALL answer this command with a *PackageVersionAns* command with the
 289 following payload.

Field	PackageIdentifier	PackageVersion
Size (octets)	1	1

290
 291 **Table 3: *PackageVersionAns***

292 *PackageIdentifier* uniquely identifies the package.

293 *PackageVersion* corresponds to the version of the package specification implemented by
 294 the end-device.

295 **5.2 DUT Reset Command (*DutResetReq*)**

296
 297 The *DutResetReq* command has no payload.

298
 299 Instructs the **DUT** to execute/simulate a full **DUT** MCU reset.

300
 301 This command allows the verification of the session context storage.

302 **5.3 Dut JoinReq Command (*DutJoinReq*)**

303
 304 The *DutJoinReq* command has no payload.

305
 306 Instructs the **DUT** to reset the LoRaWAN MAC layer and to start issuing Join-Request
 307 frames. The LoRaWAN MAC layer SHALL reinitialize such that all RF parameters are
 308 restored to default settings and the end-device SHALL then attempt to join the network as
 309 part of normal operation.

310
 311 This command allows testing the various Join-Accept test scenarios.

312
 313 **5.4 LoRaWAN Class Selection Command (*SwitchClassReq*)**

314
 315 The *SwitchClassReq* command payload is used to convey the new target end-device
 316 Class.

Fields	Class
Size (octets)	1

317
 318 **Table 4: *SwitchClassReq* fields**

319 The `Class` field tells the **DUT** to switch to that class of operation. A value of 0 for Class A,
 320 value of 1 for Class B and a value of 2 for Class C.

321

322 5.5 ADR Control Command (*AdrBitChangeReq*)

323

324 The *AdrBitChangeReq* command sent by the **TCL** requests the **DUT** to activate/deactivate
325 the ADR feature.

326

Fields	ADR
Size (octets)	1

327

Table 5: *AdrBitChangeReq* fields

328 The `ADR` field encodes the ADR state. A value of 1 means ADR ON and a value of 0 means
329 ADR OFF.

330

331 The **TCL** can verify the correct operation by checking every uplink frame ADR bit state.

332 5.6 Regional Duty-Cycle Enforcement Command 333 (*RegionalDutyCycleCtrlReq*)

334

335 The *RegionalDutyCycleCtrlReq* command sent by the **TCL** requests the **DUT** to
336 activate/deactivate the regional duty-cycle enforcement for regions requiring it.

337

Fields	DutyCycle
Size (octets)	1

338

Table 6: *RegionalDutyCycleCtrlReq* fields

339 The `DutyCycle` field encodes the regional duty-cycle enforcement state. A value of 1
340 means that the regional duty-cycle enforcement is ON and a value of 0 means that the
341 regional duty-cycle enforcement is OFF.

342

343 The **TCL** can verify the correct operation by checking that the uplinks aren't anymore
344 delayed.

345 5.7 Application Transmission Periodicity Control Command 346 (*TxPeriodicityChangeReq*)

347

348 The *TxPeriodicityChangeReq* command payload is used to convey the periodicity of uplink
349 frames.

350

Fields	Periodicity
Size (octets)	1

351

Table 7: *TxPeriodicityChangeReq* fields

352 The `Periodicity` field encodes time values in seconds which will allow to run all test
353 scenarios.

354

Periodicity	Value [s]
0	Default DUT application behavior
1	5
2	10
3	20

4	30
5	40
6	50
7	60
8	120
9	240
10	480
11-255	RFU

Table 8: Periodicity field encoding

355

356

 357 The **TCL** can verify the correct operation by checking that time between successive physical
 358 uplink packets has changed as requested.

359 **5.8 Uplink Frames Control Command (*TxFramesCtrlReq*)**

360

 361 The *TxFramesCtrlReq* command is used to convey the frame type to be used by
 362 subsequent uplink frames. This command MAY also convey N extra octets.
 363

Fields	FrameType	0..N
Size (octets)	1	N-1

Table 9: TxFramesCtrlReq fields

364

 365 The `FrameType` field encodes the frame type to be used by the **DUT** on all subsequent
 366 uplink frames.
 367

FrameType	Name	Remarks
0	No change	Allows to perform a no operation downlink from TCL
1	Unconfirmed	L2 Unconfirmed <code>FType = 2</code> frames
2	Confirmed	L2 Confirmed <code>FType = 4</code> frames
3-255	RFU	

Table 10: FrameType values description

368

369

 370 The **TCL** can verify the correct operation by checking that `FType` of MHDR field of
 371 subsequent uplink frames has changed as requested.

372 **5.9 Echo Frame Request Commands (*EchoIncPayloadReq*, 373 *EchoIncPayloadAns*)**

374

 375 The *EchoIncPayloadReq* command payload contains the N bytes to be echoed plus one.
 376 The N value is arbitrary. In case the N value is bigger than the application payload buffer
 377 then the echoed packet SHALL be clipped to the maximum payload buffer size.
 378

Fields	Octet1	Octet2	...	Octet N-1
Size (octets)	1	1	...	1

379

Table 11: EchoIncPayloadReq fields

380 **EchoIncPayloadReq** instructs the **DUT** to respond with a subsequent uplink whose payload
 381 content is the downlink's data incremented octet by octet, excepting the first octet which
 382 remains 0x08.
 383

384 Assume the received payload length is N , where N is any value between zero and maximum
 385 allowed LoRaWAN region payload size. Then the bytes composing the command payload
 386 are:

387 $[0x08, \text{octet}_1, \text{octet}_2, \dots, \text{octet}_{N-1}]$
 388

389

390 **EchoIncPayloadAns** SHALL convey a payload whose content is as follows:

391 $[0x08, \text{mod}(\text{octet}_1 + 0x01, 256), \text{mod}(\text{octet}_2 + 0x01, 256), \dots, \text{mod}(\text{octet}_{N-1} + 0x01, 256)]$
 392

393

394 ...where $\text{mod}()$ indicates modulo arithmetic.

395

396 For example, if the **DUT** receives a payload of $[8\ 1\ 5\ 255]$ on **FPort** 224, it will respond
 397 with $[8\ 2\ 6\ 0]$ on the **FPort** 224. This echo functionality is used to validate the **DUT**
 398 cryptography implementation as well as its handling of the maximum payload for both
 399 uplinks and downlinks.

400 The **EchoIncPayloadAns** SHALL be clipped to maximum Regional Parameters allowed
 401 uplink frame payload size.

402 5.10 Applicative Rx Counter Commands (**RxAppCntReq**, **RxAppCntAns**, 403 **RxAppCntRstReq**)

404

405 The **RxAppCntReq** command sent by the **TCL** requests the **DUT** to provide the current
 406 **RxAppCnt** value. This command has no payload.

407

408 The **DUT** answers to the **RxAppCntReq** with an **RxAppCntAns** command.

409

Fields	RxAppCnt
Size (octets)	2

410

Table 12: RxAppCntAns fields

411 The **RxAppCntRstReq** command sent by the **TCL** requests the **DUT** to reset the **RxAppCnt**
 412 value to 0. This command has no payload.

413

414 The **TCL** can verify **RxAppCntRstReq** command correct operation by issuing an
 415 **RxAppCntReq** command.

416 5.11 Link Check Request Command (**LinkCheckReq**)

417

418 The **LinkCheckReq** command has no payload.

419

420 Instructs the **DUT** to send a **LinkCheckReq** MAC command.

421 5.12 DeviceTimeReq MAC Command (*DeviceTimeReq*)

422

423 The *DeviceTimeReq* command has no payload

424

425 Instructs the **DUT** to send a `DeviceTimeReq` MAC command.

426 5.13 PingSlotInfoReq MAC Command (*PingSlotInfoReq*)

427

428 The *PingSlotInfoReq* command payload is used to convey the ping slots periodicity.

429

430 Instructs the **DUT** to send a `PingSlotInfoReq` MAC command.

431

432

Fields	Periodicity
Size (octets)	1

433

Table 13: `PingSlotInfoReq` fields

434 The `Periodicity` field follows the same rules as the ones provided by the Link Layer
435 specification [TS001].

436 5.14 Transmit Continuous Wave Request Command (*TxCwReq*)

437

438 The *TxCwReq* command payload is used to define the timeout, radio frequency and radio
439 transmission output power.

Fields	Timeout	Frequency	TxPower
Size (octets)	2	3	1

440

Table 14: `TxCwReq` fields

441 The `Timeout` field is a 16-bit unsigned integer indicating the number of seconds that the
442 **DUT** will spend in Continuous Wave (**CW**) mode.

443

444 The `Frequency` field is a 24-bit unsigned integer. The actual channel frequency in Hz is
445 $100 \times \text{Frequency}$ whereby values representing frequencies below 100 MHz are reserved
446 for future use.

447

448 The `TxPower` field is an 8-bit signed integer. The value in dBm represents the output power
449 applied to the **CW**.

450

451 *Example: If the **DUT** receives 7 bytes on the port 224, [0x7D 0x08 0x00 0xD8 0xB2 0x83*
452 *0x0E] (0x7D indicating the command ID), it SHALL enter the CW-mode for 8 seconds using*
453 *frequency 863.1 MHz and 14 dBm.*

454 5.15 DUT Disables FPort 224 (*DutFPort224DisableReq*)

455

456 The *DutFPort224DisableReq* command has no payload

457

458 Instructs the **DUT** to disable access to `FPort` 224 and executes a full reset of the **DUT**.

459 **5.16 DUT Versions Command (*DutVersionsReq*, *DutVersionsAns*)**

460

 461 The *DutVersionsReq* command sent by the **TCL** requests the **DUT** to provide its firmware
 462 version, Link Layer specification [TS001] version and Regional Parameters specification
 463 [RP002] version. This command has no payload.

464

 465 The **DUT** answers to the *DutVersionsReq* with a *DutVersionsAns* command.

466

Fields	FwVersion	LrwanVersion	LrwanRpVersion
Size (octets)	4	4	4

467

 Table 15: *DutVersionsAns* fields

 468 The versions (*FwVersion*, *LrwanVersion* and *LrwanRpVersion*) fields **SHALL** be
 469 encoded as *Major.Minor.Patch.Revision*: 1 octet for *Major*, 1 octet for *Minor*, 1
 470 octet for *Patch* and 1 octet for *Revision*. [Semantic Versioning 2.0.0: <https://semver.org/>]

471

 472 **Note:** In some regions (i.e: US915) the answer to this request will not fit
 473 the allowed frame size at lowest data rates. As such the **TCL** may only
 474 issue the *DutVersionReq* when the **DUT** is set with a data rate allowing
 475 the *DutVersionAns* to fit the corresponding frame size.

476 6 Glossary

477		
478	ABP	Activation by Personalization
479	ADR	Adaptive Data Rate
480	CW	Continuous Wave
481	DR	Data Rate
482	DUT	Device Under Test
483	GW	LoRaWAN Gateway
484	NS	LoRaWAN Network Server
485	MAC	Media Access Control
486	OTAA	Over-the-Air-Activation
487	TCL	Test Control Layer of the Test Harness

488 7 Bibliography**489 7.1 References**

- 490 [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP
491 14, RFC 2119, March 1997
- 492 [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP
493 14, RFC8174, May 1997
- 494 [TS001-1.0.4]: LoRaWAN L2 1.0.4 Specification, LoRa Alliance, October 2020
- 495 [RP002-1.0.1]: LoRaWAN Regional Parameters, LoRa Alliance, November 2019
- 496 [TS008]: TS008 LoRa Alliance Assigned Value Registries May 2020