LoRa Alliance
5177 Brandin Court,
Fremont, CA 94538 USA

LoRaWAN 1.0.4

End Device Certification Requirements for All Regions

Version 1.0

Table of Contents

Figures

Revision History

Version	Author	Updates	Date
Version 1.0	Nisha Bhaskaran / Derek Hunt	Release for Class A devices	October 2020

173 Glossary

LoRaWAN	Protocol specification developed and maintained by the LoRa Alliance.
Certification Logo	LoRa Alliance defined logo that can be displayed on the Certified product and any documentation and marketing information about the End-Device.
LoRa Test House	Organization and corresponding facility accredited by the LoRa Alliance to perform Certification testing.
End Device	Device submitted to a LoRa Test House for Certification.

174

175 Abbreviations & Acronyms

ABP	Activation by Personalization
ADR	Adaptive Data Rate
AS	Application Server
ATH	Authorized Test House
AWG	Arbitrary Waveform Generator
DR	Data Rate
DUT	Device Under Test
ERP	Equivalent Radiated Power compared to a dipole antenna (expressed in dBd)
EIRP	Equivalent Isotropic Radiated Power: ERP = EIRP – 2.15dB (expressed in dBi)
ETSI	European Telecommunications Standards Institute
FSK	Frequency Shift Keying modulation technique.
GW	Gateway
LCTT	LoRaWAN Conformance Test Tool
MAC	Media Access Control
NS	Network Server

OTAA	Over-the-Air Activation
TCL	Test Control Layer of the Test Harness
TRP	Total Radiated Power

Definitions of terms used in this document

XXXX	X is a valid number which would vary based on the region being tested
DC	Dynamic Channel
FC	Fixed Channel
MinDR	Minimum Data Rate for the region tested, as specified in the Regional Parameters Specification RP2 1.0.1
Max125kHzDR	Maximum Data Rate using 125 kHz bandwidth for the region tested, as specified in the Regional Parameters Specification RP2 1.0.1
Nb	Number
R	Repeat until the condition specified within brackets [] is completed

References

[1]	LoRaWAN Specifications L2 1.0.4.
[2]	LoRaWAN Regional Parameters Specification RP2 1.0.1.
[3]	LoRaWAN Certification Protocol Specification 1.0.0.

1. Introduction

This document specifies the minimum testing requirements for an End-Device to be designated "**LoRaWAN Certified**". LoRaWAN Certification will confirm that the End-Device meets the Functional Requirements of the *LoRaWAN Specification Version L2 1.0.4* [1] for the corresponding regional parameters as defined in the *LoRaWAN Regional Parameters Version RP2 1.0.1* [2].

1.1. Scope of LoRaWAN Certification

The scope of this test specification is limited to validating compliant implementation of the LoRaWAN protocol for Class A Devices.

For LoRaWAN Certification it is not mandatory that the End-Device has all the Regulatory approvals, but these will be required before the product can be sold or operated in the respective countries. Intended or otherwise, the inevitable variability of performance and quality of the radio implementation among End-Devices is too high to allow normalized-, practical evaluation. RF performance measurement, whether radiated or conducted, is therefore considered out of scope for the tests described herein. The RF performance of the End-Device will be tested solely as part of the RF Performance Testing, which will be a separate test conducted at the Authorized Test House to evaluate the RF performance of the device as part of LoRaWAN Certification testing.

1.2. LoRaWAN Certification Process

A party seeking LoRaWAN Certification for their End-Device must be a member of the LoRa Alliance in good standing and only Test Houses designated accredited by the LoRa Alliance may perform the tests described herein to earn it.

The Authorized Test Houses must communicate the complete results to the LoRa Alliance. If the device has passed all mandatory tests, the LoRa Alliance will issue a certificate for the End-Device with respect to a version of this document and in turn corresponding versions of [1] and [2]. Additionally, the Alliance will publish both the status and a results summary on its web site along with data for any optional features tested.

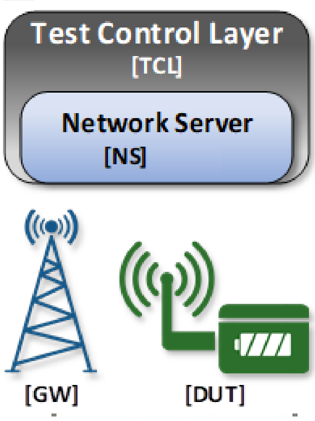
1.3. Changes made in this version for the requirement changes from LoRaWAN Specifications v1.0.2 to LoRaWAN Specification v1.0.4

- Combined all 5 regional Certification specifications into one.
- Certification Application has been removed. All testing will be done in Application mode. Corresponding updates are made throughout the document to
  - Note down the default datarate of the device and reset to default if modified during the test
  - Set the ADR bit if not already set, before performing LinkADRReq command
  - Turn off Duty cycle
- MAC commands added: DutyCycleReq and DeviceTimeReq
- Requirement changes made
  - DeviceStatusReq: SNR Margin has been renamed to RadioStatus

    - NewChannelReq: For Fixed Channel plan devices, no test must be performed
    - TXParamSetupReq: This requirement must be tested for regions other than Asian region as well to confirm that the command is ignored
    - LinkADRReq: The value [0x]F of either DataRate or TXPower means that the DUT must ignore that field and keep the current parameter values
    - LinkADRReq - TXPower: When commanded to a valid TX power level lower than it is capable of, the DUT must respond with an unsuccessful LinkADRAns and operates at its previously configured TX power
    - LinkADRReq – TXPower: When commanded to a valid TX power level greater than it is capable of, the DUT must respond with a successful LinkADRAns and operates at its maximum TX power
    - New section added: Multiple MAC command prioritization
- Section added for Test Case mapping with LW 1.0.4
- Formatting:
    - Uplinks and the corresponding downlink are combined the same step to indicate the exact sequence

2. Functional Test Description for LoRaWAN Certification

The list of tests specified below reflects the functional requirements of a Class-A End-Device as defined in [1]. The tests are conducted in a test harness generally comprised of:



- A Test Control Layer [**TCL**]
- A LoRaWAN Network Server [**NS**]
- 8/16/64 channel LoRaWAN gateway [**GW**]
- The End-Device Under Test [**DUT**]

Figure 1: Test Harness Architecture

Note:

- *A 16-channel gateway device will be used for all official Dynamic Channel Plan device certification testing*
- *A 64-channel gateway device will be used for all official Fixed Channel Plan device certification testing*
- *An 8-channel gateway device could be used for unofficial pre-testing conducted at LoRaWAN member labs. However, for official certification testing, 8-channel gateway devices will not be used. When using an 8-channel gateway, 125kHz Channels 0-7 and 500kHz Channel 64 must be configured on the gateway.*

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

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

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

The LoRaWAN gateway [**GW**] and **DUT** are collocated in an RF-isolated environment, provisioned as necessary for reliable bi-directional communication. It is nonetheless expected that both the **DUT** and **GW** will not receive every frame intended for reception. The **TCL** should make reasonable effort to accommodate this inevitability. The RF-isolated environment mentioned above should mitigate any potential interference.

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

The **TCL** must verify the following throughout the course of this certification test suite:

- The **DUT**'s uplinks' size must respect the maximum allowed uplink size for the data rate used.
- The length of the **DUT**'s uplinks based on the expected content to ensure no extraneous and unnecessary content is present.

When the **TCL** is restarted, the **DUT** must be set to the factory reset mode.

MIC check Test Note: When the tool encounters an invalid MIC, it must fail the specific test being performed.

Downlink FPort Test Note: All MAC commands sent by the TCL will be sent on FPort 0, unless specified otherwise in the Sequence charts in this document.

Channel Mask configuration for LinkADRReq MAC command for pre-testing using an 8-channel gateway for Fixed Channel plan devices

For Fixed channel plan devices, when the MAC-CMD LinkADRReq is required to be sent by the TCL, if the device is being pre-tested using an 8-channel gateway, the LinkADRReq in the sequence charts must be replaced by the LinkADRReq commands mentioned below. The LinkADRReq must first disable all 125kHz channels, enable only the channel 64 – 500kHz, and then enable Channels 0-7 using a second LinkADRReq.

MAC-CMD LinkADRReq
ChMaskCntl = 7
ChMask = [0x]0001

MAC-CMD LinkADRReq
ChMaskCntl = 0
ChMask = [0x]00FF

Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX

If a 64-channel gateway is being used for testing, then the LinkADRReq must be the same as mentioned in the Sequence charts.

2.1. Activation Pre-test

Section 2.1.1 – DUT Pre-condition Activation tests - must be executed as the first test when executing a single test or multiple tests.

Note: The TCL sends frames only on the RX2 window of the previous frame of the DUT for all tests in this document, unless specified otherwise in the Sequence Charts of the Test cases.

The DUT must support either over-the-air (OTA) activation or activation by personalization (ABP) or both. If the device supports both OTAA and ABP, the device vendor must provide 2 separate devices, one supporting OTAA and the other supporting ABP method of activation to the Authorised Test House (ATH). The ATH will use the appropriate device for each activation test. In the case where the device supports both OTAA and ABP for the same firmware version, the ATH will run the complete test cycle for the ABP device and then test only the OTAA specific tests for the OTAA device.

2.1.1. DUT Pre-condition Activation

After initial power-up – and activation if the **DUT** supports OTAA instead of ABP – the **DUT** must transmit an uplink packet as soon as possible (recommended within 10 seconds). Contents of this “I’m alive” packet are unimportant.

The **TCL** replies to this packet with a downlink payload frame of [0x]0601 (*TxPeriodicityChangeReq*) sent to port 224, setting the Uplink Periodicity value to 5 seconds. Upon setting the uplink periodicity value, the **DUT** must try to send an unconfirmed/confirmed uplink every 5 seconds.

If the uplink sent by the **DUT** is a Confirmed frame, the **TCL** sends a *TxFramesCtrlReq* command to the **DUT** to send Unconfirmed frames thereafter.

The **TCL** then checks the ADR Bit setting of the DUT and if disabled, it enables the ADR Bit using the *AdrBitChannelReq* command.

The **TCL** then sets the Data Rate to Max125kHzDR, refer [2], using the *LinkADRReq* command.

TCL finally sends the *DutVersionsReq* command to the **DUT** to obtain the version of the device.

Verify that

- **DUT** increments the DevNonce on reset
- **DUT** Uplink Periodicity is set to 5 seconds
- **DUT** sends Unconfirmed uplink frames
- **DUT** enables its ADR Bit
- **DUT** sets the Data Rate to Max125kHzDR
- **DUT** sends the version information in the *DutVersionsAns* response. TCL must store this version number for display in the final Test Report and Certificate.

For more details on the Certification Protocol implementation, refer to the Certification Protocol Specification [3].

2.1.1.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	If DUT = OTA device, DUT sends a Join-Request frame If DUT = ABP device, skip Step 1.	→	DataRate (DR) = Any allowed DR, refer [2] Note down the DevNonce	
	TCL sends a Join-Accept frame	←		
2	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices and any random number for ABP devices)	→	If the DUT is an ABP device and ADR Bit is set, DataRate (DR) = Minimum DR allowed by the DUT, refer [2]	

	The TCL sends Unconfirmed frame	←	If DUT sent Confirmed uplink frame, TCL must Acknowledge
			CP-CMD DutResetReq
			FPort = 224
			Payload = [0x]01	

3	If DUT = OTA device, DUT sends a Join-Request frame If DUT = ABP device, skip Step 3.	→	DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response on RX1 window	←		
4	DUT sends Confirmed or Unconfirmed frame FCntUp = m For OTA device, m = 0 or 1 For ABP device, m > n	→	If the DUT is an ABP device, and ADR Bit is set, DataRate (DR) = Minimum DR allowed by the DUT, refer [2]	

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

5	DUT sends Confirmed or Unconfirmed frame FCntUp = m + 1	→	FPort = any allowed port except 224	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
6	DUT sends Unconfirmed frame FCntUp = m + 2	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

7	DUT sends Unconfirmed frame FCntUp = m + 3	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			
			ChMaskCntl:
			DC = 0,
			FC = 6
			
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF	

8	DUT sends Unconfirmed frame in 5 seconds FCntUp = m + 4	→	MAC-CMD LinkADRAns Payload = [0x]0307	Changed the DR to Max125kHz DR

	The TCL sends Unconfirmed frame	←	CP-CMD DutVersionsReq
			FPort = 224
			Payload = [0x]7F	

9	DUT sends Unconfirmed frame
	FCntUp = m + 5
			CP-CMD DutVersionsAns
			FPort = 224
			Payload = [0x]7FXXXXXXXXXXXXXXXXX
				DUT version obtained and stored for future

***Note:** The FCntUp value can either start form 0 or 1. Some secure-elements implementations verify that the counter value is strictly greater than the previous value before performing the crypto operations. Thus, when resetting to 0 the value of the counter in OTAA mode, and the first time the crypto operations are performed, the counter is incremented. On other implementations, the counter value is initialized to the maximum 32-bit unsigned value [0x]FFFFFFFF which then becomes 0 when incremented by 1 prior to performing the crypto operation.*

2.2.Over the Air Activation

This test must be performed if the **DUT** supports over the air activation. The test verifies the correct functionality of the over-the-air activation. Furthermore, the fields within the Join-Accept frame (DLSettings and RXDelay) used to modify the data rates and receive window timing are tested to verify compliance.

2.2.1. Pre-Join Behaviour

2.2.1.a. For Dynamic Channel (DC) plan devices

The **TCL** commands a re-join, and the **DUT** must respond with a Join-Request frame. The **TCL** ignores the Join-Request frames until the **DUT** responds on all the default channels, refer [2].

After this, the **TCL** responds with a Join-Accept frame. The **TCL** waits for a maximum of (number of default channels * 3) Join-Request frames before responding with a Join-Accept frame.

Verify

- **DUT** sends Join-Request frames until the **TCL** responds with a Join-Accept frame.
- All the default channels must be used at least once in these requests.
- Record the DataRate of the initial Join Requests.
- The duration between the Join-Request frames is greater than the JOIN_ACCEPT_DELAY2 which is 6 seconds.
- Check if the Major version in the MAC header is correct and the RFU bits are set to 0
- The DevNonce value sent by the **DUT** in the Join-Request must be incremented in each Join-Request.
- **DUT** successfully joins the network.

<u>JoinNonce check</u>

The **TCL** must trigger a Join-Request twice. The JoinNonce value of the second Join-Accept frame must be the same as the first Join-Accept frame.
The **TCL** must trigger a Join-Request again and send a Join-Accept frame with a different JoinNonce value.

Verify

- **DUT** accepts the first Join-Accept frame
- **DUT** rejects the second Join-Accept frame
- **DUT** resends the Join-Request after rejecting the second Join-Accept frame
- **DUT** joins the network after the **TCL** sends a Join-Accept frame with a different JoinNonce value.

2.2.1.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends a maximum of (3 * number of default channels) Join-Request frames, i.e. until Join-Request frames are sent on all the default channels	→ R [3*NbCh] or [JR on AllCh]	- All Join-Request frames must be sent on all the default channels, refer [2], at least once. - Duration between previous Join-Request and next Join-Request > 6 seconds for all JoinRequest frames - Record the DataRate of the Join-Request frames - Major version is correct - RFU bits = 0 - DevNonce is greater than DevNonce of previous JR	
	TCL ignores all the Join-Request frames until all the default channels are used for the Join-Request frames	*R [3*NbCh] or [JR on AllCh]*		
3	DUT sends Join-Request frame again	→	DataRate (DR) = any allowed DR, refer [2] DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response on RX1 window	←		Join accepted
4	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

5	DUT sends Unconfirmed or Confirmed frame FCntUp = n + 1	→	FPort = any allowed port except 224	Next Uplink sent in 5 seconds

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
6	DUT sends Unconfirmed frame FCntUp = n + 2	→		

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

7	DUT sends Unconfirmed frame FCntUp = n + 3	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX	

8	DUT sends Unconfirmed frame FCntUp = n + 4	→	MAC-CMD LinkADRAns Payload = [0x]0307	Changed the DR to Max125kHz DR
9	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

10	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response	←	JoinNonce = a	
11	DUT sends Unconfirmed or Confirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Device Reinitializes [Not Joined]

12	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response	←	JoinNonce = a	JoinNonce value must be the same as the previous one

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
13	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	DUT rejects the JoinAccept response and sends JoinReq again
	TCL sends Join-Accept response	←	JoinNonce = b, where b NOT = a	Join accepted
14	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

15	DUT sends Confirmed or Unconfirmed frame FCntUp = n + 1	→	FPort = any allowed port except 224	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
16	DUT sends Unconfirmed frame FCntUp = n + 2	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

17	DUT sends Unconfirmed frame FCntUp = n + 3	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX	

18	DUT sends Unconfirmed frame within 5 seconds FCntUp = n + 4	→	MAC-CMD LinkADRAns Payload = [0x]0307	Changed the DR to Max125kHz DR

381

2.2.1.b. For Fixed Channel (FC) plan devices

After the **TCL** commands a re-join to the DUT, the **DUT** sends a Join-Request frame on 125 kHz and 500 kHz channels using the minimum data rate allowed for these channels respectively (DR-X and DR-Y respectively), refer [2]. This is ignored by the server until a channel change from 125kHz at DR-X to 500kHz at DR-Y or vice versa is recognized by the **TCL**. The last Join-Request message indicating the channel change is processed and the server responds with a Join-Accept message.

Verify

- **DUT** sends Join-Request frames on 125 kHz channels using DR-X and 500 kHz channels using DR-Y. For example: For the US902-928 region, Join-Request must be sent on a random 125 kHz channel at DR0 and a random 500kHz channel at DR4.
- The duration between the Join-Request frames is greater than the JOIN_ACCEPT_DELAY2 which is 6 seconds.
- Check if the Major version in the MAC header is correct and the RFU bits are set to 0
- The DevNonce value sent by the DUT in the Join-Request must be incremented in each Join-Request.
- **DUT** successfully joins the network.

The **TCL** again commands a re-join, the DUT sends Join-Request messages on 125 kHz channels using DR-X and 500 kHz channels using DR-Y. The **TCL** does not respond to requests with the same DR as the one responded in the previous test. It responds only to Join Requests with the alternate DR.

Verify

- **DUT** joins successfully by way of a Join-Request sent at the DR not responded earlier.

JoinNonce check

The **TCL** must trigger a Join-Request twice. The JoinNonce value of the second Join-Accept frame must be the same as the first Join-Accept frame.
The **TCL** must trigger a Join-Request again and send a Join-Accept frame with the correct JoinNonce value.

Verify

- **DUT** accepts the first Join-Accept frame
- **DUT** rejects the second Join-Accept frame
- **DUT** resends the Join-Request after rejecting the second Join-Accept frame
- **DUT** joins the network after the **TCL** sends a Join-Accept frame with the correct JoinNonce value.

2.2.1.b.i. Test Procedure Frame Sequence Chart

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends Join-Request frames.	→ R [2*NbCh] or [DR-X NOT = DR-Y]	Join-Request @ minimum default data rate for - 125 kHz on Channel A and - 500 kHz channel on Channel B (where A is a random upstream channel utilizing 125 kHz and B is a random upstream channel utilizing 500 kHz), refer [2] i.e. Join-Request1 @ DR-X on Channel A Join-Request2 @ DR-Y on Channel B DevNonce is greater than DevNonce of previous JR	
	*TCL ignores the Join-Request frames and waits for a maximum of (2 * number of channels configured) uplink transmissions, until a channel change from 125kHz at DR-X to 500kHz at DR-Y or vice versa is recognized by the TCL.*	*R [2*NbCh] or [DR-X NOT = DR-Y]*		

	TCL sends Join-Accept response when it receives a Join-Request with a channel change, on RX1 window.	← [DR-X → DR-Y]	Join-Accept is sent to DUT without a CFList.
			
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted

3	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
4	DUT sends Join-Request frames.	→ R [2*NbCh] or [DR-X NOT = DR-Y]	Join-Request @ minimum default data rate for - 125 kHz on Channel A and - 500 kHz channel on Channel B (where A is a random upstream channel utilizing 125 kHz and B is a random upstream channel utilizing 500 kHz), refer [2] i.e. Join-Request1 @ DR-X on Channel A Join-Request2 @ DR-Y on Channel B DevNonce is greater than DevNonce of previous JR	
	*TCL ignores the Join-Request frames and waits for a maximum of (2 * number of channels configured) uplink transmissions, until the DUT sends a Join-Request with a different DataRate from the one sent earlier.*	*R* *[2*NbCh] or [JR-DR-X NOT = JR-DR-Y]*		

	TCL sends Join-Accept response when it receives the Join-Request with the alternate DR	←
		[JR-DR-X NOT = JR-DR-Y]
			Join-Accept is sent to DUT without a CFList.
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted

5	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

6	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response	←	JoinNonce = a	
7	DUT sends Unconfirmed or Confirmed frame	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Device Reinitializes [Not Joined]

8	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	
	TCL sends Join-Accept response	←	JoinNonce = a	JoinNonce value must be the same as the previous one
9	DUT sends Join-Request	→	DevNonce is greater than DevNonce of previous JR	DUT rejects the JoinAccept response and sends JoinReq again

	TCL sends Join-Accept response	←	JoinNonce = b, where b NOT = a
			
			Official certification (64-channel gateway):
			Join-Accept is sent to DUT without a CFList
			
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted

10	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

11	DUT sends Confirmed or Unconfirmed frame FCntUp = n + 1	→	FPort = any allowed port except 224	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
12	DUT sends Unconfirmed frame FCntUp = n + 2	→		

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

13	DUT sends Unconfirmed frame FCntUp = n + 3	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			ChMaskCntl = 6
			ChMask = [0x]00FF	

14	DUT sends Unconfirmed frame in 5 seconds FCntUp = n + 4	→	MAC-CMD LinkADRAns Payload = [0x]0307	Changed the DR to Max125kHz DR

2.2.2. Join-Accept with DLSettings

After the **TCL** triggers a Join-Request, the **DUT** starts the Join procedure for over-the-air activation. The **TCL** responds with a Join-Accept frame with RX1DRoffset = 2 and RX2DataRate = any applicable DataRate, except the default RX2DataRate as defined in [2]. After the join procedure succeeds, the **TCL** downlinks an echo command targeting the RX1 window to which the **DUT** must respond correctly. Next the **TCL** downlinks an echo command targeting the RX2 window to which the **DUT** must respond correctly.

Verify

- **DUT** successfully joins the network
- **DUT** implements RX1DRoffset correctly after processing the Join-Accept
- **DUT** implements RX2DataRate correctly after processing the Join-Accept

2.2.2.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends Join-Request	→	DR = Z (where Z = any allowed DR, refer [2]) DevNonce is greater than DevNonce of previous JR	

	TCL sends Join-Accept response	←	RX1DROffset = 2
			RX2DataRate = Any DR except default RX2 DR, as defined in [2]
			
			Join-Accept is sent to DUT without a CFList.
			
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted with modified parameters

3	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

4	DUT sends Unconfirmed or Confirmed frame FCntUp = n + 1	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
6	DUT sends Unconfirmed frame	→	FCtrl ADR Bit = true	ADR Bit turned ON

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			
			ChMaskCntl:
			DC = 0,
			FC = 6
			
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF	

7	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			Payload = [0x]0307
			
			DR = Max125kHzDR	

	The TCL sends Unconfirmed frame on RX1 window	←	DR = Max125kHzDR - 2
			
			CP-CMD EchoPayloadReq
			FPort 224
			Payload = [0x]08010203	

8	DUT sends Unconfirmed frame	→	DR = Max125kHzDR
			
			CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				RX1 reply sent

	The TCL sends Unconfirmed frame on RX2 window	←	RX2DataRate - As set in Join-Accept frame above CP-CMD EchoPayloadReq FPort 224 Payload = [0x]080A0B0C	

9	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]080B0C0D
				RX2 reply sent

10	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Revert the device to default values

11	DUT sends Join-Request	→	DR = Z (where Z = any allowed DR, refer [2]) DevNonce is greater than DevNonce of previous JR	

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	

	TCL sends Join-Accept response	←	Join-Accept is sent to DUT without a CFList.
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001	

12	DUT sends Unconfirmed or Confirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD
			TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

13	DUT sends Unconfirmed or Confirmed frame	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
14	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

15	DUT sends Unconfirmed frame	→	FCtrl ADR Bit = true	ADR Bit turned ON

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

16	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			Payload = [0x]0307
			
			DR = Max125kHzDR	

2.2.3. Join-Accept with Delay Settings

After the **TCL** triggers a JoinRequest, the **DUT** starts the Join procedure for over-the-air activation as above. The **TCL** responds with a Join-Accept frame containing Delay Settings on RX2 window, such that RX1 and subsequently RX2 timing is increased (at least 2 seconds are recommended). The **TCL** sends an echo command targeted to RX1 to which the **DUT** must respond correctly. The **TCL** repeats this same downlink test against the RX2 window, to which the **DUT** must respond correctly.

Verify

- **DUT** successfully joins the network
- **DUT** implements the new (non-default) Delay Settings
- **DUT** restores the default settings for RXDelay

2.2.3.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends Join-Request	→	DR = Any allowed DR, refer [2] DevNonce is greater than DevNonce of previous JR	

	TCL sends Join-Accept response on RX2 window	←	RXDelay = n, where 2 <= n <=15
			
			Join-Accept is sent to DUT without a CFList.
			
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted with modified parameters

3	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

4	DUT sends Unconfirmed or Confirmed frame FCntUp = n + 1	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame on RX1 window	←	RX1Delay = n seconds CP-CMD EchoPayloadReq FPort 224 Payload [0x]08010203	

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	
6	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns FPort 224 Payload [0x]08020304	RX1 delay reply sent
	The TCL sends Unconfirmed frame on RX2 window	←	RX2Delay = n + 1 seconds CP-CMD EchoPayloadReq FPort 224 Payload [0x]080A0B0C	
7	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns FPort = 224 Payload [0x]080B0C0D	RX2 delay reply sent

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Revert the device to default values

8	DUT sends Join-Request	→	DR = Z (where Z = any allowed DR, refer [2]) DevNonce is greater than DevNonce of previous JR	

	TCL sends Join-Accept response	←	Join-Accept is sent to DUT without a CFList.
			
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001	

9	DUT sends Unconfirmed or Confirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

10	DUT sends Unconfirmed or Confirmed frame	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
11	DUT sends Unconfirmed frame	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

12	DUT sends Unconfirmed frame	→	FCtrl ADR Bit = true	ADR Bit turned ON

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF	

13	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			Payload = [0x]0307
			DR = Max125kHzDR	

2.2.4. Join-Accept with CFList

2.2.4.a. For Dynamic Channel (DC) plan devices

After the **TCL** triggers a Join-Request, the **DUT** starts the Join procedure for over-the-air activation as above. The **TCL** responds with a Join-Accept frame containing an additional single channel in the CFList field. The **DUT** must use this additional channel together with the default channels within the following uplinks.

The **TCL** again triggers the DUT to send a Join-Request. The **TCL** responds with a Join-Accept frame containing a CFListType = 1. The **DUT** rejects the CFList and enables all default channels. The **DUT** must not use the additional channel added earlier.

Verify

- **DUT** successfully joins the network
- **DUT** uses the new channel in its random selection of frequencies
- **DUT** removes the additional channel added earlier.

2.2.4.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends Join-Request	→	DR = Any allowed DR, refer [2] DevNonce is greater than DevNonce of previous JR	

	TCL sends Join-Accept response	←	CFList = add single channel
			CFListType = 0
				Join accepted

3	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

4	DUT sends Unconfirmed or Confirmed frame FCntUp = n + 1	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	DUT sends Unconfirmed frame FCntUp = n + 2	→		
6	Wait until the new channel which was added has been used at least once. Wait for a maximum of [5*(number of channels configured)] uplink packets to be sent.	→ R [5*NbCh] or [AllCh used]	Channel added is used at least once	DUT adds the additional channel to its default channel plan
7	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

8	DUT sends Join-Request	→	DR = Any allowed DR, refer [2] DevNonce is greater than DevNonce of previous JR	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

	TCL sends Join-Accept response	←	CFList = add another channel
			CFListType = 1
				Join-Accept sent with CFListType = 1

9	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

10	DUT sends Unconfirmed or Confirmed frame FCntUp = n + 1	→		
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
11	DUT sends Unconfirmed frame FCntUp = n + 2	→		
12	Wait for [5* (number of channels configured)] uplink packets to be sent.	→ R [5*NbCh] or [AllCh used]	- Default channels are used at least once. - The additional channel is not used.	DUT removes the additional channel from its default channel plan

2.2.4.b. For Fixed Channel (FC) plan devices

After the TCL triggers a Join-Request, the DUT starts the Join procedure for over-the-air activation as above.

The TCL responds with a Join-Accept frame with CFListType set to [0x]01 in the CFList field. The ChMask fields must enable 3 channels (Channel 0, 1 and 64) and disable all other channels.

Verify

- DUT successfully joins the network
- The DUT must use only the channels enabled for the uplinks.

The TCL triggers a Join-Request again and responds with a Join-Accept frame with CFListType set to [0x]00 in the CFList field. The ChMask fields must enable 3 channels (Channel 0, 1 and 64) and disable all other channels.

Verify

- DUT successfully joins the network
- The DUT must reject the CFList as the value of the CFListType = [0x]00 and must use all channels for uplinks. For the sake of verification, verify that any other channel other than the channels enabled in the ChMask bits are used for the uplinks.

Finally, the TCL triggers a Join-Request again and responds with a Join-Accept frame with CFListType set to [0x]01 in the CFList field. The ChMask fields must be set to [0x]FFFF.

Verify

- DUT successfully joins the network
- The DUT must use all the channels for the uplinks.

2.2.4.b.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
				Device Reinitializes [Not Joined]

2	DUT sends Join-Request frame	→	DevNonce is greater than DevNonce of previous JR	

	TCL sends Join-Accept response	←	CFListType = [0x]01
			ChMask0 = [0x]0003
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted with CFList

3	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

4	DUT sends Unconfirmed or Confirmed frame in 5 seconds FCntUp = n + 1	→	FPort = any allowed port except 224	Next Uplink sent in 5 seconds
5	Wait for a maximum of 5 uplinks to be sent.	→ R [max 5]	If DR = 125kHz DR, only Channels 0 and 1 must be used for all uplinks If DR = 500kHz DR, only channel 64 must be used for uplinks	Only the enabled channels are used for uplinks
6	DUT sends a Confirmed or Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Device Reinitializes [Not Joined]

7	DUT sends Join-Request frames.	→	DevNonce is greater than DevNonce of previous JR	
	*TCL ignores the Join-Request frames and waits for a maximum of (2 * number of channels configured) uplink transmissions, until the Join-Request channel is a 500kHz channel*	*R* *[2*NbCh] or [500kHz channel]*		

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device – TCL	**Frame**	

	TCL sends Join-Accept response	←	CFListType = [0x]00
			ChMask0 = [0x]0003
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted with CFListType = [0x]00

8	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

9	DUT sends Unconfirmed or Confirmed frame in 5 seconds FCntUp = n + 1	→	FPort = any allowed port except 224	Next Uplink sent in 5 seconds
10	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

11	DUT sends Confirmed or Unconfirmed frame in 5 seconds FCntUp = n + 1	→	FPort = any allowed port except 224	Next Uplink sent in 5 seconds
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device – TCL	**Frame**	
12	If Data Rate = 500kHz DR, wait for (5 * number of 500kHz channels configured) uplink transmissions (max. 16 packets for 500kHz channels), i.e. until all 500kHz channels configured are used at least once If Data Rate = 125kHz DR, skip this step	→ R [5*Nb500kHzCh] or [AllCh used]	*Official certification (64-channel gateway):* All 500kHz channels must be used at least once *Pre-testing with 8-channel gateway:* Channel 64 must be used at least once.	All 500kHz channels must be used at least once
13	DUT sends a Confirmed or Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD DutJoinReq
			FPort = 224
			Payload = [0x]02
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Device Reinitializes [Not Joined]

14	DUT sends Join-Request frames.	→	DevNonce is greater than DevNonce of previous JR	
	*TCL ignores the Join-Request frames and waits for a maximum of (2 * number of channels configured) uplink transmissions, until the Join-Request channel is a 125kHz channel*	*R [2*NbCh] or [125kHz channel]*		

	TCL sends Join-Accept response	←	Join-Accept must be sent without a CFList.
			Pre-testing with 8-channel gateway:
			CFListType = [0x]01
			ChMask0 = [0x]00FF
			ChMask1 = [0x]0000
			ChMask2 = [0x]0000
			ChMask3 = [0x]0000
			ChMask4 = [0x]0001
				Join accepted without a CFList

15	DUT sends Unconfirmed or Confirmed frame FCntUp = n (where n = 0 or 1 for OTA devices)	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Uplink Periodicity set

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
16	DUT sends Confirmed or Unconfirmed frame in 5 seconds FCntUp = n + 1	→	FPort = any allowed port except 224	Next Uplink sent in 5 seconds
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
17	Wait for (2 * number of 125kHz channels configured) uplink transmissions (max. 128 packets for 125kHz channels), i.e. until all 125kHz channels configured are used at least once	→ R [2*Nb125kHzCh] or [AllCh used]	*Official certification (64-channel gateway):* All 125kHz channels must be used at least once *Pre-testing with 8-channel gateway:* Channels 0-7 must be used at least once.	All 125kHz channels must be used at least once
18	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	The TCL sends Unconfirmed frame
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

19	DUT sends Unconfirmed frame	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			ChMaskCntl = 6
			ChMask = [0x]00FF	

20	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Changed the DR to Max125kHz DR

2.3. Activation by Personalization

This test must be performed if the **DUT** supports activation by personalization.

For this test, the Authorised Test Lab must use the device provided by the device vendor to test the ABP activation functionality.

After initial power-up, the **DUT** must be enabled for testing as specified in Section 2.1.1.

Verify

- **DUT** successfully joins the network.

The Test Procedure Message Sequence Chart for this test is the same as Section 2.1.1.

Additionally, the DUT must also retain its previous settings even after reset.
TCL sets the **DUT**'s parameters using the MAC commands RXParamSetupReq, DlChannelReq and RXTimingSetupReq.
The **DUT** is then reset using the CP-CMD DutResetReq.

Verify

- The **DUT**'s parameters must be retained even after reset.

2.3.1. Dynamic channel plan devices

2.3.1.a. All regions - Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	Perform all steps mentioned in Section 2.1.1		Same results as mentioned in Section 2.1.1	
2	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq TXPower = Minimum, refer [2] DataRate = Max125kHzDR, refer [2] ChMaskCntl = 0 ChMask = [0x]0001 MAC-CMD RxParamSetupReq RX1DRoffset = any allowed offset value except default, refer [2] RX2DataRate = Any DataRate allowed except default, refer [2] RX2Frequency = Y (where Y = any frequency allowed except default, refer [2]) MAC-CMD DlChannelReq ChIndex = C (where C = Any default channel, refer [2]) Freq = X (where X = any allowed frequency except default, refer [2]) MAC-CMD RxTimingSetupReq Delay (i) = 2 Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX [0x]0AXXXXXXXX[0x]08XX	Channel 0 enabled RxParamSetupReq, DlChannelReq, RxTimingSetupReq commands executed
3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns MAC-CMD DlChannelAns MAC-CMD RxTimingSetupAns Payload = [0x]0307[0x]0507[0x]0A03[0x]08	

	The TCL sends Unconfirmed frame	←	CP-CMD DutResetReq
			FPort = 224
			Payload = [0x]01
				Reset the DUT

4	DUT sends Confirmed or Unconfirmed frame	→	If the ADR Bit is set, DataRate (DR) = Minimum DR allowed by the DUT, refer [2]	

	The TCL sends **Confirmed** frame on RX1 window	←	RX1DRoffset = as set in Step 2
			RX1Delay = as set in Step 2
			Freq = as set in Step 2
			
			CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				RX1DRoffset, RX1Delay and Freq retained after reset

5	DUT sends Confirmed or Unconfirmed frame	→	ACK Bit = True	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
6	Wait for a maximum of 5 * (number of default channels)	→ R [5*NbChDC] or [AllCh used]	All default channels must be used at least once	All channels must be enabled after the reset
7	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

8	DUT sends Unconfirmed frame	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends **Confirmed** frame on RX2 window	←	RX2DataRate = as set in Step 2
			RX2Frequency = as set in Step 2
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = default, refer [2]
			RX2DataRate = default, refer [2]
			RX2Frequency = default, refer [2]
			
			MAC-CMD DlChannelReq
			ChIndex = C
			Freq = default, refer [2]
			
			MAC-CMD RxTimingSetupReq
			Delay (i) = default, refer [2]
			
			MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			
			Payload = [0x]05XXXXXXXX[0x]0AXXXXXXXX[0x]08XX[0x]03XXXXXXXX
				RX2DataRate, RX2Freq setting retained after reset
				
				Later, revert to default for all settings.
				
				Set DR to Max125kHz DR

9	DUT sends Unconfirmed frame	→	ACK Bit = True MAC-CMD RxParamSetupAns MAC-CMD DlChannelAns MAC-CMD RxTimingSetupAns MAC-CMD LinkADRAns Payload = [0x]0507[0x]0A03[0x]08[0x]0307	

2.3.1.b. For regions with Dwell Time limitation only - Test Procedure Frame Sequence Chart

These tests must be performed only for regions with Dwell Time limitation. **TCL** sets the **DUT**'s UplinkDwellTime using the TXParamSetupReq MAC command. The **DUT** is then reset using the CP-CMD DutResetReq.

Verify

- The UplinkDwellTime setting must be retained even after reset.

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX
				TXParamSetupReq command executed

2	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	

	The TCL sends Unconfirmed frame	←	CP-CMD DutResetReq
			FPort = 224
			Payload = [0x]01
				Reset the DUT

3	DUT sends Confirmed or Unconfirmed frame	→		

	The TCL sends **Confirmed** frame on RX1 window	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Change periodicity to 5sec

4	DUT sends Confirmed or Unconfirmed frame	→	ACK Bit = True	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

6	DUT sends Unconfirmed frame	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame on RX2 window	←	MAC-CMD LinkADRReq
			DataRate = MinDR, refer [2]
			Payload = [0x]03XXXXXXXX
				UplinkDwellTime setting retained after reset-tested by setting to MinDR

7	DUT sends Unconfirmed frame in 5 seconds	→	MAC-CMD LinkADRAns Payload = [0x]0307	
8	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			
			MAC-CMD TXParamSetupReq
			UplinkDwellTime = default, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]09XX
				Revert to Max125kHz DR and revert UplinkDwellTime to default

9	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD TXParamSetupAns Payload = [0x]0307[0x]09	

2.3.2. Fixed channel plan devices

2.3.2.a. All regions - Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	Perform all steps mentioned in Section 2.1.1		Same results as mentioned in Section 2.1.1	
2	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Minimum, refer [2]
			DataRate = Max500kHzDR, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0001
			
			MAC-CMD LinkADRReq
			TXPower = Minimum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0003
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = any allowed offset value except default, refer [2]
			RX2DataRate = Any DataRate allowed except default, refer [2]
			RX2Frequency = Y (where Y = any frequency allowed except default, refer [2])
			
			MAC-CMD RxTimingSetupReq
			Delay (i) = 2
			
			Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX[0x]05XXXXXXXX[0x]08XX
				Channels 0, 1 and 64 enabled
				
				RxParamSetupReq, RxTimingSetupReq commands executed

3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns MAC-CMD RxTimingSetupAns Payload = [0x]0307[0x]0307[0x]0507[0x]08	

4	The TCL sends Unconfirmed frame	←	CP-CMD DutResetReq
			FPort = 224
			Payload = [0x]01
				Reset the DUT

	DUT sends Confirmed or Unconfirmed frame	→		

	The TCL sends **Confirmed** frame on RX1 window	←	RX1DRoffset = as set in Step 2
			RX1Delay = as set in Step 2
			Freq = as set in Step 2
			
			CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				RX1DRoffset, RX1Delay and Freq retained after reset

5	DUT sends Confirmed or Unconfirmed frame	→	ACK Bit = True	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
6	If DR = 125kHz DR, wait for a maximum of 2 * (number of 125kHz channels configured) uplink packets to be sent, i.e. until all channels are used at least once. If DR = 500kHz DR, wait for a maximum of 5 * (number of 500kHz channels configured) uplink packets to be sent, i.e. until all channels are used at least once.	→ R [2*Nb 125kHzCh] or [5*Nb 500kHzCh] or [AllCh used]	If DR = 125kHz DR, all 125kHz channels including the channels which were disabled earlier must be used at least once If DR = 500kHz DR, all 500kHz channels including the channels which were disabled earlier must be used at least once *Pre-testing for FC plan with 8-channel gateway:* Channels 0-7 (if DR = 125kHz DR) or channel 64 (if DR = 500kHz DR) must be used at least once.	All channels must be enabled after the reset
7	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

8	DUT sends Unconfirmed frame	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends **Confirmed** frame on RX2 window	←	RX2DataRate = as set in Step 2
			RX2Frequency = as set in Step 2
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = default, refer [2]
			RX2DataRate = default, refer [2]
			RX2Frequency = default, refer [2]
			
			MAC-CMD RxTimingSetupReq
			Delay (i) = default, refer [2]
			
			MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			
			Payload = [0x]05XXXXXXXX[0x]08XX[0x]03XXXXXXXX
				RX2DataRate, RX2Freq, retained after reset
				
				Revert to default for all settings.
				
				Changed the DR to Max125kHz DR

9	DUT sends Unconfirmed frame	→	ACK Bit = True MAC-CMD RxParamSetupAns MAC-CMD RxTimingSetupAns MAC-CMD LinkADRAns Payload = [0x]0507[0x]08[0x]0307	

2.3.2.b. For regions with Dwell Time limitation only - Test Procedure Frame Sequence Chart

These tests must be performed only for regions with Dwell Time limitation. **TCL** sets the **DUT**’s UplinkDwellTime using the TXParamSetupReq MAC command. The **DUT** is then reset using the CP-CMD DutResetReq.

Verify

- The UplinkDwellTime setting must be retained even after reset.

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX
				TXParamSetupReq command executed

2	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	

	The TCL sends Unconfirmed frame	←	CP-CMD DutResetReq
			FPort = 224
			Payload = [0x]01
				Reset the DUT

3	DUT sends Confirmed or Unconfirmed frame	→		

	The TCL sends **Confirmed** frame on RX1 window	←	CP-CMD TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Change periodicity to 5sec

4	DUT sends Confirmed or Unconfirmed frame	→	ACK Bit = True	
	If DUT sent a Confirmed frame, then The TCL sends Unconfirmed frame Else, this step must be skipped	←	CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	DUT sends Unconfirmed frame	→		

	If FCtrl ADR Bit = false, then
	
	The TCL sends Unconfirmed frame
	
	Else, this step is skipped
		←	CP-CMD AdrBitChangeReq-ON
			FPort = 224
			Payload = [0x]0401	

6	DUT sends Unconfirmed frame	→	FCtrl ADR bit = true	Turn on the ADR bit

	The TCL sends Unconfirmed frame on RX2 window	←	MAC-CMD LinkADRReq
			DataRate = MinDR, refer [2]
			Payload = [0x]03XXXXXXXX
				UplinkDwellTime setting retained after reset-tested by setting to MinDR

7	DUT sends Unconfirmed frame in 5 seconds	→	MAC-CMD LinkADRAns Payload = [0x]0307	
8	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			
			MAC-CMD TXParamSetupReq
			UplinkDwellTime = default, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]09XX
				Revert to Max125kHzDR and revert UplinkDwellTime to default

9	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD TXParamSetupAns Payload = [0x]0307[0x]09	

549

2.4. Device Functionality Tests

2.4.1. Default Setting Tests

The **TCL** will test the basic functionality of the **DUT** using the current applicative 'RxAppCnt' value and using the [0x]08 Echo command. The purpose of these tests is to detect implementation errors in the **DUT** early, instead of failing later tests.

2.4.1.a. Cryptography:

Verifies that AES encryption and message integrity code (MIC) algorithms are correctly implemented by the **DUT**.

2.4.1.a.i. AES Encryption

TCL will send multiple [0x]08 Echo commands with varying length payloads to the **DUT**. The varying payloads lengths must contain lengths

- Test with a physical frame size smaller than 16 bytes. (Less than block of 16 bytes,)
- Test with a physical frame size equal to 16 bytes. (16 bytes,)
- Test with a physical frame where the size is between 17 and 31 bytes. (Greater than block of 16 bytes but lesser than the second block,)
- Test with a physical frame size equal to the maximum which is 255 bytes (Max payload length). For this test used data rate is important as the max length depends on it.

The echo-replies from the **DUT** are verified to contain the same payload where each byte is incremented by one. This test will only use echo commands whose payload is less than or equal to the maximum allowed payload of the **DUT**'s current uplink DR. Frame counter must increase for every received package. Oversized payloads are tested separately in Section 2.5.11.d.

576 ### 2.4.1.a.i.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frames	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08 (Various)
			
			Note: See description for Various details	

3	DUT sends Unconfirmed frame
	FCntUp = n + 1 + i
		→ R [3]	CP-CMD EchoPayloadAns (repeat i times- where i = 1 to 3)
			FPort = 224
			Payload = [0x]08 (Various)'
			
			No ACK is sent for the previous Unconfirmed frame
				Multiple Echo replies sent

	The TCL sends Unconfirmed frames	← R [3]	CP-CMD EchoPayloadReq (repeat i times- where i = 1 to 3)
			FPort = 224
			Payload = [0x]08 (Various)	

4	DUT sends Unconfirmed frames
	FCntUp = n + 5
		→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08 (Various)'
				Echo reply sent

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

5	DUT sends Unconfirmed frame
	FCntUp = n + 6
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 5
				Downlink counter incremented

577

578 ### 2.4.1.a.ii. Message Integrity Code

579 **TCL** will send packets with purposely invalid message integrity codes. The **DUT** must
580 ignore these packets.

2.4.1.a.ii.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frames	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08 (Various)
			MIC Invalid	

3	DUT sends Unconfirmed frame FCntUp = n + 1 + i	→ R [4]		MIC Invalid packets ignored

	The TCL sends Unconfirmed frames	← R [4]	CP-CMD EchoPayloadReq (repeat i times- where i = 1 to 4)
			FPort = 224
			Payload = [0x]08 (Various)
			MIC Invalid	

4	DUT sends Unconfirmed frame FCntUp = n + 6	→		MIC Invalid packet ignored

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

5	DUT sends Unconfirmed frame FCntUp = n + 7	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 1
				Downlink counter not incremented

2.4.1.b. Downlink Sequence Number

This test verifies the **DUT** properly handles the frame sequence numbers that are used to prevent replay attacks on the communication.

TCL sends several packets with decreasing sequence number i such that:

$$1 \leq i < Current\ FCntDown$$

The **DUT** must ignore downlinks whose sequence numbers are less than its current internal downlink counter value.

2.4.1.b.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frame
	FCntDown = a
		←	CP-CMD TxFramesCtrlReq
			FPort = 224
			Frame type = No change
			Payload = [0x]0700	

3	DUT sends Unconfirmed frame FCntUp = n + 1 + y *Repeat y times*	→ R [y]		

	The TCL sends Unconfirmed frame
	FCntDown = a – i (where i = [1, a-1])
	Repeat y times
		← R [y]	CP-CMD TxFramesCtrlReq
			FPort = 224
			Frame type = No change
			Payload = [0x]0700	

4	DUT sends Unconfirmed frame FCntUp = n + 2 + y	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

5	DUT sends Unconfirmed frame
	FCntUp = n + 3 + y
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 2
				DUT ignores **y** downlinks with FCntDown < current downlink counter

2.4.2. Confirmed Frames

This test verifies that the **DUT** properly handles confirmed frames as both the sender (uplinks) and receiver (downlinks).

2.4.2.a. Confirmed Uplinks

The **TCL** sends the [0x]0702 TxFramesCtrlReq command to the **DUT**. Verify all subsequent uplinks from the **DUT** are frame type *ConfirmedUp*.

2.4.2.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frame	←	CP-CMD TxFramesCtrlReq FPort 224 Frame Type = Confirmed Payload [0x]0702	
3	DUT sends **Confirmed** frame FCntUp >= n + 2	→		Confirmed frame sent
	The TCL sends Unconfirmed frame	←	Acknowledge No FPort and no payload	
4	DUT sends **Confirmed** frame FCntUp >= n + 3	→		Confirmed frame sent
	The TCL sends Unconfirmed frame	←	Acknowledge CP-CMD TxFramesCtrlReq FPort = 224 Frame type = No change Payload [0x]0700	
5	DUT sends **Confirmed** frame FCntUp >= n + 4	→		Confirmed frame sent

	The TCL sends Unconfirmed frames	←	Acknowledge
			
			CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

6	DUT sends **Confirmed** frame
	FCntUp >= n + 5
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt >= x + 4	

7	DUT sends **Confirmed** frame FCntUp >= n + 6	→		DUT increments the FCntUp even when no ACK is sent by TCL
	The TCL sends Unconfirmed frame	←	Acknowledge CP-CMD TxFramesCtrlReq FPort 224 Frame type = Unconfirmed Payload [0x]0701	

8	DUT sends Unconfirmed frame FCntUp >= n + 7	→		Switch back to Unconfirmed frame

2.4.2.b. **Confirmed Downlinks**

TCL sends a *ConfirmedDown* packet. Verify the **DUT** sets the *ACK* bit in the subsequent uplink. The test also verifies that when the **TCL** sends a retransmission with the same FCntDown, the DUT ignores the downlink.

2.4.2.b.i.Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n			

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL send **Confirmed** frame FCntDown = m	←	CP-CMD TxFramesCtrlReq FPort 224 Frame type = Confirmed Payload [0x]0702	TCL sends Confirmed frame
3	DUT sends **Confirmed** frame FCntUp = n + 2	→	ACK Bit = True Note: The DUT may split this frame into first an empty Unconfirmed frame with ACK, followed by a Confirmed frame. This must be accepted as well.	
	The TCL sends **Confirmed** frame FCntDown = m + 1	←	Acknowledge CP-CMD TxFramesCtrlReq FPort 224 Frame type = Unconfirmed Payload [0x]0701	TCL sends Confirmed frame
4	DUT sends Unconfirmed frame FCntUp = n + 3	→	ACK Bit = True	
	The TCL sends **Confirmed** frames FCntDown = m + 2	←	CP-CMD TxFramesCtrlReq FPort 224 Frame type = No change Payload [0x]0700	
5	DUT sends Unconfirmed frame FCntUp = n + 4	→	ACK Bit = True	
	The TCL sends **Confirmed** frames FCntDown = m + 2	←	CP-CMD TxFramesCtrlReq FPort 224 Frame type = No change Payload [0x]0700	
6	DUT sends Unconfirmed frame FCntUp = n + 5	→	No acknowledgement is sent	DUT ignores the downlink with incorrect frame counter

	The TCL sends Unconfirmed frames
	FCntDown = m + 3
		←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

7	DUT sends Unconfirmed frame
	FCntUp = n + 6
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 4	

2.5. MAC Command Tests

The following tests will validate the **DUT**'s implementation of MAC command processing and the associated functional areas being controlled by the **TCL**. As previously stated, the **TCL** should allow for some reasonable amount of packet loss while facilitating tests. Specifically, where MAC commands are concerned, it is acceptable to retry commanding the device in the absence of a MAC command answer. Retries should be limited to a maximum of 5 attempts. After 5 uplinks are received from the **DUT**, if the **TCL** still does not get the expected response from the **DUT**, the test must fail, and the tool must move to the next test.

2.5.1. DevStatusReq

TCL sends a *DevStatusReq* command to the **DUT**.

Verify that

- The **DUT** must reply with a *DevStatusAns* packet.
- The signal-to-noise information in the RadioStatus field in the reply is a signed integer of 6 bits with a minimum value of -32 and maximum value of 31.

2.5.1.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD DevStatusReq Payload [0x]06	
2	DUT sends Unconfirmed frame	→	MAC-CMD DevStatusAns RadioStatus >= - 32 and <= 31 Payload [0x]06XXXX	DevStatusAns sent and encoded value tested

2.5.2. NewChannelReq

2.5.2.a. Fixed Channel plan devices

For Fixed channel plan devices, the NewChannelReq MAC command must be rejected and the **DUT** must silently drop the request packet. The **DUT** must continue normal operation.

2.5.2.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq
			ChIndex = 4
			Freq = any applicable frequency, refer [2]
			Payload = [0x]07XXXXXXXXXX	

2	DUT sends Unconfirmed frame	→		No response to the command but DUT continues normal operation

2.5.2.b. For Dynamic Channel plan devices only

TCL sends a *NewChannelReq* command to the **DUT** for configuring new channel frequencies. *NewChannelReq* command is supported for only Dynamic channel plan devices. **For Fixed channel plan devices**, **DUT** must silently drop the request packet.

2.5.2.b.i. Read-only default channels

The mandatory default channels are defined in [2]. The **TCL** will send *NewChannelReq* that tries to modify these channels. The **DUT** must reply with an unsuccessful *NewChannelAns NOT = ([0x]0703)*.

2.5.2.b.ii. Addition of a channel

TCL sends a MAC command to add a single new channel.

The **DUT** must reply with a successful *NewChannelAns* and begin using the new channel in its random selection of frequencies. This test may use any frequency applicable for that region [2].

2.5.2.b.iii. Removal of a channel

TCL sends *NewChannelReq* to set the previously configured channel to 0 MHz frequency.

The **DUT** must reply with a successful *NewChannelAns* and stop using the additional channel in its uplink transmissions.

2.5.2.b.iv. Addition and removal of multiple channels

TCL sends multiple MAC commands in a single frame to configure the additional channels. Multiple such frames may be sent to configure all the additional non-default channels and to remove them.

The **DUT** must reply with a successful *NewChannelAns* to each request. The DUT must send an uplink on each channel configured.

2.5.2.b.v. Invalid command processing - Frequency

TCL sends *NewChannelReq* command including an invalid frequency located out of the band to check that the **DUT** replies correctly.

2.5.2.b.vi. Invalid command processing – Data Rate Range

TCL sends *NewChannelReq* command including an invalid data rate range to check that the **DUT** replies correctly.

2.5.2.b.vii. Removal of default channels – not allowed

TCL tries to remove the default channels. **DUT** must reject the command and must continue to use the default channels.

2.5.2.b.viii. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq ChIndex = 0 Freq = Any allowed frequency for the channel, refer [2] DRRange = Any valid range, refer [2] Payload = [0x]07XXXXXXXXXX	TCL attempts to modify default channels
2	DUT sends Unconfirmed frame	→ R [All default Ch]	MAC-CMD NewChannelAns Payload NOT = [0x]0703	
	The TCL sends Unconfirmed frame	← R [All default Ch]	MAC-CMD NewChannelReq ChIndex = All other default channels, refer [2] Freq = Any allowed frequency for that channel, refer [2] DRRange = Any valid range, refer [2] Payload = [0x]07XXXXXXXXXX	
3	DUT sends Unconfirmed frame	→	MAC-CMD NewChannelAns Payload NOT = [0x]0703	DUT shall not change its channel plan or transmission behaviour

	The TCL sends Unconfirmed frame	←	FPort = 0
			MAC-CMD NewChannelReq
			ChIndex = 15
			Freq = any applicable frequency, refer [2]
			Payload = [0x]07XXXXXXXXXX	

4	DUT sends Unconfirmed frame	→	MAC-CMD NewChannelAns Payload = [0x]0703	
5	Wait for a maximum of (5 * number of channels configured) uplink packets, i.e. until the channel configured is used at least once	→ R [5*NbCh] or [AllCh used]	The new channel configured must be used at least once	New channel added
6	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq
			ChIndex = 15
			Freq = 0 MHz
			Payload = [0x]07XXXXXXXXXX	

7	DUT sends Unconfirmed frame	→	MAC-CMD NewChannelAns Payload = [0x]0703	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
8	Wait for 5 * number of channels configured uplink packets, to confirm that the removed channel is not used.	→ R [5*NbCh]	The channel removed must not be used	
9	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	FPort = 0 MAC-CMD NewChannelReq ChIndex = all non-default channel indexes, refer [2] Freq = any frequency applicable for that region, refer [2]. Each channel must have a different frequency, as supported by the gateway. Payload = [0x]07XXXXXXXXXX[0x]07XXXXXXXXXX… [repeat [0x]07XXXXXXXXXX up to (16 - NbDefaultChannels)] ***Note1***: This downlink may be split into multiple downlinks so that the maximum FRMPayload is not exceeded ***Note2***: When performing pre-testing using an 8-channel gateway for DC plan devices, the Freq field must be set to 8 different allowed frequencies for that region as supported by the gateway for the first 8 channels. The same set of 8 frequencies must be repeated for the next set of 8 channels as well.	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
10	DUT sends Unconfirmed frame	→	FPort = 0 MAC-CMD NewChannelAns Payload = [0x]0703[0x]0703[0x]0703…[Repeat [0x]0703 “Y” times] – where Y is the number of channels configured Note: This uplink may be split into multiple uplinks so that the maximum FRMPayload is not exceeded	16 channels (default + additional channels) configured

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			ChMaskCntl = 0
			ChMask = [0x]F000
			Payload = [0x]03XXXXXXXX	

11	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Enable channel IDs 12-15
12	Wait for 5 * 4 (number of channels enabled) = 20 uplink packets to be sent	→ R [5*NbCh]	Only the 4 enabled channels must be used at least once, and the disabled channels must not be used	Checking channel usage
13	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			ChMaskCntl = 0
			ChMask = [0x]0F00
			Payload = [0x]03XXXXXXXX	

14	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Enable channel IDs 8-11
15	Wait for 5 * 4 (number of channels enabled) = 20 uplink packets to be sent	→ R [5*NbCh]	Only the 4 enabled channels must be used at least once, and the disabled channels must not be used	Checking channel usage
16	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			ChMaskCntl = 0
			ChMask = [0x]00F0
			Payload = [0x]03XXXXXXXX	

17	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Enable channel IDs 4-7
18	Wait for 5 * 4 (number of channels enabled) = 20 uplink packets to be sent	→ R [5*NbCh]	Only the 4 enabled channels must be used at least once, and the disabled channels must not be used	Checking channel usage
19	DUT sends Unconfirmed frame	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			ChMaskCntl = 0
			ChMask = [0x]000F
			Payload = [0x]03XXXXXXXX	

20	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Enable channel IDs 0-3
21	Wait for 5 * 4 (number of channels enabled) = 20 uplink packets to be sent	→ R [5*NbCh]	Only the 4 enabled channels must be used at least once, and the disabled channels must not be used	Checking channel usage
22	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	FPort = 0 MAC-CMD NewChannelReq ChIndex = all non-default channel indexes, refer [2] Freq = 0 MHz Payload = [0x]07XXXXXXXXXX[0x]07XXXXXXXXXX… [repeat [0x]07XXXXXXXXXX for all channels removed]	
23	DUT sends Unconfirmed frame	→	FPort = 0 MAC-CMD NewChannelAns Payload = [0x]0703[0x]0703[0x]0703… [Repeat [0x]0703 "Y" times] – where Y is the number of channels removed	All additional Channels removed
24	Wait for 5 * (default number of channels) uplink packets to be sent	→ R [5*NbDefCh]	The additional channels which were removed must not be used and only the default channels must be used	Checking channel usage
25	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CMD NewChannelReq
			ChIndex = Any channel index other than the default, refer [2]
			Freq = Any invalid Freq, refer [2]
			Payload = [0x]07XXXXXXXXXX	

26	DUT sends Unconfirmed frame	→	CMD NewChannelAns Payload NOT = [0x]0703	DUT shall not modify its frequency or transmission behaviour

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq ChIndex = Any channel index other than the default, refer [2] Freq = Default, refer [2] DRRange = An invalid data range, refer [2] Payload = [0x]07XXXXXXXX	
27	DUT sends Unconfirmed frame	→	MAC-CMD NewChannelAns Payload is NOT = [0x]0703	DUT shall not add the channel due to invalid Data Range
	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq ChIndex = All default channels, refer [2] Freq = 0 MHz Payload = [0x]07XXXXXXXX[Repeat [0x]07XXXXXXXX "Y" times] – where Y is the number of default channels	
28	DUT sends Unconfirmed frame	→	MAC-CMD NewChannelAns Payload NOT =[0x]0703[Repeat "Y" times] – where Y is the number of default channels	Default channels not affected
29	Wait for 5 * (number of channels configured) uplink packets to be sent	→ R [5*NbDefCh]	Only the default channels must be used at least once	Checking channel usage

2.5.3. DlChannelReq

2.5.3.a. Fixed Channel plan devices

For Fixed channel plan devices, the **DUT** must silently drop the DlChannelReq MAC command request. The **DUT** must continue normal operation.

2.5.3.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD DlChannelReq ChIndex = Any default channel, refer [2] Freq = Any frequency other than default freq, refer [2]) Payload = [0x]0AXXXXXXXX	
2	DUT sends Unconfirmed frame	→		No response to the command but DUT continues normal operation

2.5.3.b. Dynamic Channel plan devices

The **TCL** sends a MAC command to change the downlink frequency in RX1 for an existing channel. The **TCL** tests RX1 downlink window using the new frequency and then restores the default values. Retransmission is tested, as well. The **TCL** waits for an uplink, while the MAC answer should be [0x]0A03. Then the **TCL** sends a downlink and waits for the next uplink, while the MAC answer should not contain [0x]0A03. Finally, standard settings are applied and tested again. Additionally, the **TCL** sends *DlChannelReq* commands including invalid frequency and channel values to check that the **DUT** replies appropriately.

2.5.3.b.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame on RX1 window	←	MAC-CMD DlChannelReq ChIndex = C (where C = Any default channel, refer [2]) Freq = X (where X = any allowed frequency other than the default frequency, refer [2]) Payload = [0x]0AXXXXXXXX	
2	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the DlChannelAns*	→ R [max 3]	MAC-CMD DlChannelAns Payload = [0x]0A03	
3	DUT sends Unconfirmed frame	→	MAC-CMD DlChannelAns Payload = [0x]0A03	

	The TCL must send an Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]080A0B0C
			
			ChIndex = Any one of the default channels, refer [2]
			Freq = The frequency set
				Tests downlink frequency settings for the default channel

4	DUT sends Unconfirmed frame
	
	Repeat for a maximum of (5 * number of default channels) until the DUT sends an uplink on all default channels
		→ R
		[5*NbDefCh]
		OR
		[All default channels]
			If the EchoPayloadReq was sent in the previous downlink,
			
			CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]080B0C0D	

	If the DUT sends an uplink on another default channel, then the TCL sends Unconfirmed frame on RX1 window
	
	Repeat for all default channels
		← R
		[All default channels]
			CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]080A0B0C
			
			ChIndex = Each of the remaining default channels, refer [2]
			Freq = The frequency set
				Tests downlink frequency settings for all default channels

5	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]080B0C0D
				Tests downlink frequency settings for the last default channel

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

	TCL sends Unconfirmed frame	←	MAC-CMD DlChannelReq
			ChIndex = C
			
			Freq = default frequency, refer [2]
			
			Payload = [0x]0AXXXXXXXX	

6	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the DlChannelAns*	→ R [max 3]	MAC-CMD DlChannelAns Payload = [0x]0A03	DUT returns to its default settings
7	DUT sends Unconfirmed frame	→	MAC-CMD DlChannelAns Payload = [0x]0A03	

	The TCL must send an Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]080A0B0C
			
			ChIndex = Any one of the default channels, refer [2]
			Freq = Default frequency
				Tests default frequency settings for the default channel

8	DUT sends Unconfirmed frame
	
	Repeat for a maximum of (5 * number of default channels) until the DUT sends an uplink on all default channels
		→ R
		[5*NbDefCh]
		OR
		[All default channels]
			If the EchoPayloadReq was sent in the previous downlink,
			
			CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]080B0C0D	

	If the DUT sends an uplink on another default channel, then the TCL sends Unconfirmed frame on RX1 window
	
	Repeat for all default channels
		← R
		[All default channels]
			CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]080A0B0C
			
			ChIndex = Each of the remaining default channels, refer [2]
			Freq = Default frequency
				Tests default frequency settings for all default channels

9	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]080B0C0D
				Tests default frequency settings for the last default channel

	The TCL sends Unconfirmed frame	←	MAC-CMD DlChannelReq
			ChIndex = C
			
			Freq = Any invalid frequency, refer [2]
			Payload = [0x]0AXXXXXXXX	

10	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the DlChannelAns*	→ R [max 3]	MAC-CMD DlChannelAns Payload is NOT = [0x]0A03	Unsuccessful for invalid frequency

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
11	DUT sends Unconfirmed frame	→	MAC-CMD DlChannelAns Payload is NOT = [0x]0A03	

	The TCL must wait until the DUT sends an uplink on the same channel configured in Step 4 and then sends Unconfirmed frame	←	MAC-CMD DlChannelReq
			ChIndex = Any channel not configured, refer [2]
			
			Freq = default frequency
			Payload = [0x]0AXXXXXXXX	

12	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the DlChannelAns*	→ R [max 3]	MAC-CMD DlChannelAns Payload is NOT = [0x]0A03	Unsuccessful for channel not configured
13	DUT sends Unconfirmed frame	→	MAC-CMD DlChannelAns Payload is NOT = [0x]0A03	

	The TCL sends Unconfirmed frame	←	CP-CMD TxFramesCtrlReq
			FPort = 224
			Frame type = No change
			Payload = [0x]0700	

14	DUT sends Unconfirmed frame	→		

2.5.4. RXParamSetupReq

The **TCL** sends a MAC command to configure new *RX1DRoffset, RX2DataRate* and *Frequency*. The **DUT** must include a successful *RXParamSetupAns* in all subsequent uplinks until a downlink is received. The **TCL** will wait for more than one uplink containing the affirmative *RXParamSetupAns* of [0x]0507. The **TCL** then sends a downlink and verifies the next uplink does not contain the *RXParamSetupAns* [0x]0507. Both the RX1 and RX2 downlink windows are then tested using the new parameters. Finally, default settings are restored by way of MAC command and both RX windows tested again.

The **TCL** sends a MAC command with invalid RX2DRoffset, RX2DataRate and Frequency that is not supported. Verify that the **DUT** ignores the command and the previous parameters are kept.

2.5.4.a. Test Procedure Frame Sequence Chart

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = y + 1	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame
	FCntUp = y + 2
		→	MAC-CMD LinkADRAns
			DataRate = X
			Payload = [0x]0307
				DataRate set to value configured

	The TCL sends Unconfirmed frame	←	MAC-CMD RxParamSetupReq RX1DRoffset = any allowed offset value, refer [2] RX2DataRate = Any DataRate allowed except the one set in Step 2 Frequency = Y (where Y = any frequency allowed, refer [2]) Payload = [0x]05XXXXXXXX	
3	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the RxParamSetupAns* FCntUp >= y + 3	→ R [max 3]	MAC-CMD RxParamSetupAns Payload = [0x]0507	
4	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload = [0x]0507	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			DR = X - RX1DRoffset
			Payload = [0x]08010203	

5	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Tested for new parameters on RX1

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314
			Freq = Y
			DataRate = RX2DataRate as set	

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	

6	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Tested for new parameters on RX2

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = Default
			Frequency = Default
			Payload = [0x]05XXXXXXXX
			The default values are defined in [2]	

7	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload = [0x]0507	Restored to default settings

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203	

8	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			RX1DRoffset = 0
			Payload = [0x]08020304
				Default Rx1 Params verified

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			RX2DataRate = Default
			Frequency = Default
			FPort = 224
			Payload = [0x]08121314	

9	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Default Rx2 Params verified

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxParamSetupReq
			RX1DRoffset = Invalid, refer [2]
			RX2DataRate = default
			Frequency = default
			Payload = [0x]05XXXXXXXX
				Invalid RX1DRoffset

10	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload is NOT = [0x]0507	DUT confirms that the parameters were not set

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxParamSetupReq
			RX2DataRate = Invalid, refer [2]
			RX1DROffset = default
			Frequency = default
			
			Payload = [0x]05XXXXXXXX
				Invalid RX2DataRate

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
11	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload is NOT = [0x]0507	DUT confirms that the parameters were not set

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxParamSetupReq
			RX1DROffset = default
			RX2DataRate = default
			Frequency = Invalid, refer [2]
			
			Payload = [0x]05XXXXXXXX
				Invalid Freq

12	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload is NOT = [0x]0507	DUT confirms that the parameters were not set

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203	

13	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Default Rx1 Params verified

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314	

14	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Default Rx2 Params verified

2.5.5. RXTimingSetupReq

The **TCL** sends a MAC command to change the timing of the reception windows. The **DUT** must reply with [0x]08 (and no *RXTimingSetupAns* payload). The **TCL** tests RX1 and RX2 downlink windows using the new parameters.

Additionally, retransmission is tested. The **TCL** sends a MAC command to change the timing of the reception windows. The **TCL** waits for an uplink with *RXTimingSetupAns* [0x]08. The **TCL** does not send a downlink and instead waits for subsequent uplinks that must contain the *RXTimingSetupAns*. Once satisfied, the **TCL** sends a downlink and waits for the next uplink to ensure it does not contain the *RXTimingSetupAns* [0x]08. Finally, standard settings are applied and tested again with Echo command.

2.5.5.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = y	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD RxTimingSetupReq Payload = [0x]08XX Delay (i) = [3-14]	
2	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the RxTimingSetupAns* FCntUp >= y + n	→ R [max 3]	MAC-CMD RxTimingSetupAns Payload = [0x]08	
3	DUT sends Unconfirmed frame	→	MAC-CMD RxTimingSetupAns Payload = [0x]08	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203
			TXDelay = (i) seconds	

4	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Tested for new parameters on RX1

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314
			TXDelay = (i + 1) seconds	

5	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Tested for new parameters on RX2

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxTimingSetupReq
			Payload = [0x]08XX
			Delay = 2
				Setting Delay = 2

6	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the RxTimingSetupAns*	→ R [max 3]	MAC-CMD RxTimingSetupAns Payload = [0x]08	
7	DUT sends Unconfirmed frame	→	MAC-CMD RxTimingSetupAns Payload = [0x]08	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203
			TXDelay = 2 sec	

8	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Tested for delay settings on RX1

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314
			
			TXDelay = 3 seconds	

9	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Tested for delay settings on RX2

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxTimingSetupReq
			Payload = [0x]08XX
			Delay = 15
				Setting Delay = 15

10	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the RxTimingSetupAns*	→ R [max 3]	MAC-CMD RxTimingSetupAns Payload = [0x]08	
11	DUT sends Unconfirmed frame	→	MAC-CMD RxTimingSetupAns Payload = [0x]08	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203
			
			TXDelay = 15 sec	

12	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Tested for delay settings on RX1

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314
			
			TXDelay = 16 seconds	

13	DUT sends Unconfirmed frame	→ R	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Tested for delay settings on RX2

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxTimingSetupReq
			Payload = [0x]08XX
			Delay = 0
				Setting Delay to default = 0

14	DUT sends Unconfirmed frame *Repeat up to 3 times until a downlink is received confirming the receipt of the RxTimingSetupAns*	→ R [max 3]	MAC-CMD RxTimingSetupAns Payload = [0x]08	
15	DUT sends Unconfirmed frame	→	MAC-CMD RxTimingSetupAns Payload = [0x]08	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08010203
			
			TXDelay = 1 sec	

16	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08020304
				Tested for delay settings on RX1

17	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq
			FPort = 224
			Payload = [0x]08121314
			
			TXDelay = 2 seconds	

18	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns
			FPort = 224
			Payload = [0x]08131415
				Tested for delay settings on RX2

2.5.6. TXParamSetupReq MAC command

Refer to [2] to obtain the list of regions for which the TXParamSetupReq command is applicable. If the TXParamSetupReq command is not applicable for the region being tested, the **DUT** must continue normal operation after receiving the TXParamSetupReq command hence ignoring the command.

2.5.6.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq
			Payload = [0x]09XX
			UplinkDwellTime = 0	

2	DUT sends Unconfirmed frame	→		No response to the command but DUT continues normal operation

If the TXParamSetupReq command is applicable for the region being tested, the following tests must be performed.

1. No dwell time setting: The **TCL** sends *LinkADRReq* to set the **DUT** to each valid DR which is not in the Join-Request Data Rate range, refer [2]. **DUT** must reply with an unsuccessful *LinkADRAns* and the **TCL** verifies that no packet is received using this data rate.
2. *UplinkDwellTime* set to 0 (unlimited): The **TCL** sends a *TXParamSetupReq* to set *UplinkDwellTime* to 0 (unlimited) and the **DUT** must reply with a *TXParamSetupAns*. Then, the **TCL** sends *LinkADRReq* to set the **DUT** to the Minimum Data Rate up to the Maximum Data Rate using 125kHz, refer [2].

The **DUT** must reply to each request with a successful *LinkADRAns.* The **TCL** verifies that the uplink data rate is as requested.

3. *UplinkDwellTime* set to 1 (400ms): The **TCL** sends a *TXParamSetupReq* to set *UplinkDwellTime* to 1 (400ms) and the **DUT** must reply with a *TXParamSetupAns*. Then, the **TCL** sends *LinkADRReq* to set the **DUT** to each DR which is in the invalid Data Rate range, refer [2]. The **DUT** must reply with an unsuccessful *LinkADRAns* and the server verifies that no packet is received using this data rate. For the data rates in the Join-Request Data Range, the **DUT** must reply to each request with a successful *LinkADRAns.* The **TCL** verifies that the uplink data rate is as requested.
4. MaxEIRP setting:
    a. Set *MaxEIRP = Highest:* Uplink signal level is monitored on the gateway side (1).
    b. Set *MaxEIRP = Lowest*: Uplink signal level is monitored on the gateway side (2).
    c. Check that (1) – (2) is greater than 6 dB

2.5.6.b. Test Procedure Frame Sequence Chart- if TXParamSetupReq command is applicable for the region tested

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq DataRate = The first DR which is not in the Join-Request Data Rate range, refer [2] Payload = [0x]03XXXXXXXX	
2	DUT sends Unconfirmed frame	→ R [All DR not in JR range]	MAC-CMD LinkADRAns Payload NOT = [0x]0307	
	The TCL sends Unconfirmed frame	← R [All DR not in JR range]	MAC-CMD LinkADRReq DataRate = All other DR which are not in the Join-Request Data Rate range, refer [2] Payload = [0x]03XXXXXXXX	
3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload NOT = [0x]0307 Note: This response is for the last LinkADRReq with DR not in the JR range	LinkADRAns Status = Unsuccessful

	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX	

4	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	Uplink Dwell Time set to 0

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = MinDR, refer [2]
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame	→ R
		[All DR < Max125kHz DR]
			MAC-CMD LinkADRAns
			Payload = [0x]0307
			DataRate = X
				DataRate set as required

	The TCL sends Unconfirmed frame
	
	Repeat this step for each DR up to the maximum DR
		← R
		[All DR up to Max125kHz DR]
			MAC-CMD LinkADRReq
			DataRate = Each other allowed DR up to Max125kHzDR, refer [2]
			Payload = [0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			Payload = [0x]0307
			DataRate = Max125kHzDR, refer [2]
			
			Note: This response is for the last LinkADRReq with Max125kHzDR.
				DataRate set as required

Testing UplinkDwellTime = 1				
7	DUT sends Unconfirmed frame	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

	The TCL sends Unconfirmed frame	←	MAC-CMD
			TXParamSetupReq
			UplinkDwellTime = 1 (400ms)
			Payload = [0x]09XX	

8	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	Uplink Dwell Time set to 1 (400ms)

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = The first DR in the invalid Data Rate range, refer [2]
			Payload = [0x]03XXXXXXXX	

9	DUT sends Unconfirmed frame	→ R [All invalid data rate range]	MAC-CMD LinkADRAns Payload NOT = [0x]0307	
	The TCL sends Unconfirmed frame	← R [All invalid data rate range]	MAC-CMD LinkADRReq DataRate = Each other DR which is in the invalid Data Rate range, refer [2] Payload = [0x]03XXXXXXXX	
10	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload NOT = [0x]0307	LinkADRAns Status = Unsuccessful
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq Data Rate = The first DR in the Join-Request Data Rate range, refer [2]) Payload = [0x]03XXXXXXXX	

11	DUT sends Unconfirmed frame	→ R
		[All DR in the JR range]
			MAC-CMD LinkADRAns
			Payload = [0x]0307
			DataRate = Y	

	The TCL sends Unconfirmed frame Repeat this step until DataRate reaches the maximum data rate in the Join-Request DataRange, refer [2]	← R [All DR in JR range]	MAC-CMD LinkADRReq Data Rate = Y (where Y = Each other DR in the Join-Request Data Rate range, refer [2]) Payload = [0x]03XXXXXXXX	

12	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			Payload = [0x]0307
			DataRate = Y
				DataRate set to Y

MaxEIRP testing				
13	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq Max EIRP = Highest, refer [2] Payload = [0x]09XX	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
14	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	Max EIRP set to max value

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			Payload = [0x]03XXXXXXXX	

15	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	TXPower set to max value
16	DUT sends Unconfirmed frame	→	Get RSSI value	
17	DUT sends Unconfirmed frame	→	Get RSSI value	
18	DUT sends Unconfirmed frame	→	Get RSSI value X = Avg of last 3 RSSI value	RSSI value avg checked
	The TCL sends Unconfirmed frame	←	MAC-CMD TXParamSetupReq Max EIRP = Lowest, refer [2] Payload = [0x]09XX	
19	DUT sends Unconfirmed frame	→	MAC-CMD TXParamSetupAns Payload = [0x]09	Max EIRP set to min value

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			Payload = [0x]03XXXXXXXX	

20	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	TXPower set to max value
21	DUT sends Unconfirmed frame	→	Get RSSI value	
22	DUT sends Unconfirmed frame	→	Get RSSI value	
23	DUT sends Unconfirmed frame	→	Get RSSI value Y = Avg of last 3 RSSI value Confirm: X – Y > 6 dB	RSSI value avg checked Difference of RSSI values checked

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			MAC-CMD TXParamSetupReq
			Max EIRP = Highest, refer [2]
			UplinkDwellTime = default, refer [2]
			Payload = [0x]03XXXXXXXX [0x]09XX	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
24	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD TXParamSetupAns Payload = [0x]0307[0x]09	DUT reverted to default settings

2.5.7. LinkCheckReq tests

After the **TCL** triggers a LinkCheckReq, the **DUT** must send a LinkCheckReq frame. The **TCL** responds with a LinkCheckAns frame.

Verify

- After receiving the LinkCheckAns, the **DUT** continues normal operation and sends uplinks.

2.5.7.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	CP-CMD LinkCheckReq FPort 224 Payload [0x]20	
2	DUT sends Unconfirmed frame	→	MAC-CMD LinkCheckReq Payload = [0x]02	
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkCheckAns Payload = [0x]02XXXX	
3	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	CP-CMD EchoPayloadReq FPort 224 Payload [0x]08010203	
4	DUT sends Unconfirmed frame	→	CP-CMD EchoPayloadAns FPort = 224 Payload [0x]08020304	DUT responds normally after LinkCheck

2.5.8. LinkADRReq

The following tests validate each aspect of the *LinkADRReq* command.

2.5.8.a. TXPower

This test validates the **DUT** handles the TXPower indicated in the *LinkADRReq* command as an upper-limit (i.e. allowed maximum).

The **TCL** separately tests *LinkADRReq* commands with TXPower within and outside the range specified in [2]. As part of testing values within the valid range, the **TCL** will command the device to its minimum and maximum allowable TX power levels. TCL also checks for a TXPower value of [0x]F.

Verify

- When commanded to a valid TX power level lower than it is capable of, the **DUT** responds with an unsuccessful LinkADRAns and operates at its previously configured TX power
- When commanded to a valid TX power level greater than it is capable of, the **DUT** responds with a successful LinkADRAns and operates at its maximum TX power
- The **DUT** responds with a successful *LinkADRAns* for those commands whose TXPower is within the specified range. The **TCL** will monitor the RSSI reported by the gateway and verify a minimum difference of at least 6 dB between these settings.
- Commands whose TX power is outside the range or [0x]F, the **DUT** responds with an unsuccessful *LinkADRAns* and keep the current parameter values.

***Note**: The ChMaskCntl and ChMask for Dynamic channel (DC) and Fixed channel (FC) plan devices are different. The difference is specified in the Sequence charts below.*

2.5.8.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Minimum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns If minimum TXPower the DUT is capable of is greater than the minimum allowed TXPower as refer [2], then PowerACK = false Payload NOT = [0x]0307 and DUT retains its previous TXPower Else, PowerACK = true Payload = [0x]0307	Command is rejected if minimum TXPower the DUT is capable of is greater than the minimum allowed TXPower else the command is accepted
3	If in Step 2, the minimum TXPower the DUT is capable of is greater than the minimum allowed TXPower, then perform Steps 3 and 4. Else, skip to the step 5. DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Minimum supported by DUT, as mentioned in the Certification Questionnaire.
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF (for FC)
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			PowerACK = true
			Payload = [0x]0307	

5	DUT sends Unconfirmed frame	→	Get the RSSI value	
6	DUT sends Unconfirmed frame	→	Get the RSSI value	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
7	DUT sends Unconfirmed frame	→	Get the RSSI value X = Avg of last 3 RSSI value	Avg RSSI checked

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR for the region, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

8	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			PowerACK = true
			DataRateACK = true
			ChannelMaskACK = true
			Payload = [0x]0307
				Command accepted

9	DUT sends Unconfirmed frame	→	Get the RSSI value	
10	DUT sends Unconfirmed frame	→	Get the RSSI value	
11	DUT sends Unconfirmed frame	→	Get the RSSI value Y = Avg of last 3 RSSI value Confirm: X – Y > 6dB	RSSI value avg checked Difference of RSSI values checked

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = RFU, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX
			
			Note: For US902-928, this step must be skipped as the TXPower cannot be set to RFU	

12	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload NOT = [0x]0307 Note: For US902-928, this step must be skipped as the TXPower cannot be set to RFU	Command rejected

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = [0x]F
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

13	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			PowerACK = true
			DataRateACK = true
			ChannelMaskACK = true
			Payload = [0x]0307
				Command accepted

14	DUT sends Unconfirmed frame	→	Get the RSSI value	
15	DUT sends Unconfirmed frame	→	Get the RSSI value	
16	DUT sends Unconfirmed frame	→	Get the RSSI value Z = Avg of last 3 RSSI value Confirm: Y = Z ± 3dB There must be no change to the TXPower and it must be the same as set earlier.	RSSI value avg checked No change to TXPower

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

17	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b. Uplink Channel Management

2.5.8.b.i. For Dynamic channel plan devices -

This section is applicable for only Dynamic channel plan devices.

2.5.8.b.i.1. Unsupported data rates

The **TCL** sends MAC commands to change the **DUT** to an unsupported data rate, refer [2].

The **DUT** must reply with an unsuccessful *LinkADRAns* and the uplink data rate must not change.

Verify

- The **DUT** replies to each valid request with an unsuccessful *LinkADRAns*.
- The uplink data rate does not change.

790 2.5.8.b.i.1.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum
			DataRate = Unsupported data rate, refer [2]
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n	→ R	MAC-CMD LinkADRAns Payload NOT = [0x]0307	Request rejected

	The TCL sends Unconfirmed frame	← R
		[All optional data rates]
			MAC-CMD LinkADRReq
			TXPower = Maximum
			DataRate = An optional data rate except default data rate, refer [2]
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX
			
			Note: Repeat for all optional data rates.	

3	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns Payload NOT = [0x]0307	Request rejected

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum
			DataRate = [0x]F
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame FCntUp = n + 2	→	MAC-CMD LinkADRAns Payload = [0x]0307	Request accepted and no change to DR

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame FCntUp = n + 3	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.i.2. ChannelMask functionality & Disable all Channels

The **TCL** sends *NewChannelReq* command to configure a new channel and *LinkADRReq* to disable that channel using the channel mask. The **TCL** verifies that no transmissions are sent on the new channel. The **TCL** then sends a *LinkADRReq* to enable the new channel. The **TCL** verifies that the **DUT** begins using the new channel in a random sequence.

Finally, the additional channel is removed.

Disable all Channels

Lastly, the **TCL** tries to disable all channels by sending a *LinkADRReq* command with channel mask control index of 0 and a channel mask of [0x]0000. The **DUT** must reply with an unsuccessful *LinkADRAns,* (i.e. not [0x]0307).

The **TCL** verifies that **DUT** still uses the default uplink channels in random sequence at an appropriate data rate.

2.5.8.b.i.2.1. Test Procedure Frame Sequence Chart for Channel Mask and Disable all Channels

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq
			ChIndex = X (where X = any unused optional channel index)
			Freq = Any allowed frequency except the default frequency, refer [2]
			DRRange = Max125kHzDR to Max125kHzDR, refer [2]
			
			MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			
			Payload = [0x]07XXXXXXXXXX[0x]03XXXXXXXX
				Adds new channel and disables it

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	DataRate = Max125kHzDR, refer [2]
			
			MAC-CMD NewChannelAns
			
			MAC-CMD LinkADRAns
			Payload = [0x]0703[0x]0307	

3	Wait until all channels configured are used at least once to confirm the channel plan [Wait for a maximum of (5 * number of channels configured) uplink packets]	→ R [5*NbCh]	All default channels are used at least once. The newly added channel is not used.	DUT does not modify its channel plan
4	DUT sends Unconfirmed frame FCntUp = i	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask – Enable the default channels and the newly added channel only, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX
				Enables the new channel

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

5	DUT sends Unconfirmed frame
	FCntUp = i + 1
		→	DataRate = Max125kHzDR, refer [2]
			
			MAC-CMD LinkADRAns
			Payload = [0x]0307	

6	Wait until all channels configured are used at least once to confirm the channel plan [Wait for a maximum of (5 * number of channels configured) uplink packets]	→ R [5*NbCh]	All default channels and the newly added channel are used at least once	New channel added to default plan
7	DUT sends Unconfirmed frame FCntUp = j	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD NewChannelReq
			ChIndex = X
			Freq = 0 MHz
			Payload = [0x]07XXXXXXXXXX
				Removes new channel

8	DUT sends Unconfirmed frame FCntUp = j + 1	→	MAC-CMD NewChannelAns Payload = [0x]0703	Channel removed
9	DUT sends Unconfirmed frame FCntUp = k	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0000
			NbTrans = 1
			Payload = [0x]03XXXXXXXX
				Tries to disable all channels

10	DUT sends Unconfirmed frame FCntUp = k + 1	→	MAC-CMD LinkADRAns Payload NOT = [0x]0307	Channel plan not modified

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask - Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

11	DUT sends Unconfirmed frame FCntUp = k + 2	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.ii. For Fixed channel plan Devices

This section is applicable for only Fixed channel plan devices.

2.5.8.b.ii.1. 125khz Uplink Channel Management

This test validates the **DUT**’s ability to process commands to operate at Data Rates allowed for 125 kHz uplink channels as defined in [2]. It also validates the **DUT** replies appropriately to invalid *LinkADRReq* commands with unsuccessful *LinkADRAns.*

2.5.8.b.ii.1.1. Valid Command Processing

The **TCL** sends *LinkADRReq* commands to change the **DUT**’s DataRate to each of the allowed Data Rates for 125 kHz uplink channels, refer [2]. The commands use channel mask control 6 and a channel mask value of [0x]00FF.

Verify

- The **DUT** replies to each valid request with a successful LinkADRAns.
- The DataRate of subsequent uplinks is as commanded.

2.5.8.b.ii.1.1.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			Payload = [0x]0500
			FPort = 224	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = The first allowed data rates for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

3	DUT sends Unconfirmed frame FCntUp >= n + 2	→ R [All DR for 125kHz]	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = [0x]0307	

	The TCL sends Unconfirmed frame
	
	Repeat the test for all the remaining allowed Data Rates for 125 kHz uplink channels
		← R [All DR for 125kHz]	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = Each of the other allowed data rates for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = [0x]0307	All allowed DRs tested

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum
			DataRate = [0x]F
			ChMaskCntl = 6
			ChMask – [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX
				Testing for DR = [0x]F

5	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	Request accepted and no change to DR

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

7	DUT sends Unconfirmed frame	→		

2.5.8.b.ii.1.2. Invalid Command Processing

Using separate downlinks, the **TCL** sends two commands: the first is a valid command disabling all 500 kHz uplink Channels with mask index 6, channel mask of [0x]0000 and an allowed Data Rate for 125 kHz uplink channels, refer [2]. The second is an invalid *LinkADRReq* command with channel mask control 6, channel mask of [0x]0000 and an allowed Data Rates for 500 kHz uplink channels, refer [2]

Verify

- The **DUT** replies to the first command with a successful LinkADRAns ([0x]0307) and to the second with an unsuccessful LinkADRAns. Acceptable values of the second LinkADRAns include (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304.

Next, using separate downlinks, the **TCL** sends two commands: the first is a valid command disabling all 125 kHz uplink Channels with mask index 7, channel mask of [0x]00FF and an allowed Data Rate for 500 kHz uplink channels, refer [2]. The second is an invalid command with a mask index 0, channel mask of [0x]0001 and an allowed Data Rate for 125 kHz uplink channels, refer [2].

Verify

- The **DUT** replies to the first command with a successful LinkADRAns ([0x]0307) and to the second with an unsuccessful LinkADRAns. Acceptable values of the second LinkADRAns include (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304.

2.5.8.b.ii.1.2.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]0000
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = [0x]0307	Successful LinkADRAns

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rates for 500 kHz uplink channels, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]0000
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

3	DUT sends Unconfirmed frame FCntUp = n + 2	→	MAC-CMD LinkADRAns Payload = Any one of these- (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304.	Unsuccessful LinkADRAns

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 500 kHz uplink channels, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame FCntUp = n + 3	→	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = [0x]0307	Successful LinkADRAns

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0001
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame FCntUp = n + 4	→	MAC-CMD LinkADRAns Payload = Any one of these- (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304.	Unsuccessful LinkADRAns

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.ii.1.3. Enable All-Channels

The **TCL** then sends a LinkADRReq to enable all channels. In this case the command's channel mask control index is 6 and the DR specified must be appropriate for 125 kHz channels. The channel mask must be applied to the 500 kHz channels. The **DUT** *must* reply with a successful *LinkADRAns*. The TCL verifies that DUT resumes using the full range (64) of 125kHz uplink channels in random sequence at an appropriate DR.

2.5.8.b.ii.1.3.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→	]	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = Max data rate for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns DataRate = As set in the previous step Payload = [0x]0307	
3	Wait for a maximum of 2 * (number of channels configured) uplink packets to be sent, i.e. until all channels are used at least once.	→ R [2*NbCh] or [All Ch used]	*Official certification:* DUT resumes using the full range (64) of 125kHz uplink channels in random sequence at the DR set *Pre-testing with 8-channel gateway:* Channels 0-7 and Channel 64 must be used at least once.	All channels must be used at least once
4	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.ii.2. 500 kHz Uplink Channel Management

This test validates the **DUT**'s ability to process commands to operate at Data Rates allowed for 500 kHz uplink channels as defined in [2]. It also validates the **DUT** replies appropriately to invalid *LinkADRReq* commands with an unsuccessful *LinkADRAns*.

2.5.8.b.ii.2.1. Valid Command Processing

The **TCL** sends *LinkADRReq* commands the **DUT**'s Data Rate to an allowed DataRate for 500 kHz uplink channels, refer [2]. The commands use channel mask control 7 and a channel mask value of [0x]00FF.

Verify

- The **DUT** replies to the valid request with a successful LinkADRAns ([0x]0307)
- The Data Rate of subsequent uplinks is as set by **TCL** command, and all 500kHz uplink channels are used.

Next the **TCL** sends a single *LinkADRReq* command to the **DUT** with an allowed DataRate for 500 kHz uplink channels, refer [2], channel mask control 7, and channel mask value of [0x]0001.

Verify

- The **DUT** replies to the valid request with a successful LinkADRAns ([0x]0307)
- All subsequent uplinks are sent on Channel 64 at the DataRate set by the **TCL** command.

2.5.8.b.ii.2.1.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 500 kHz uplink channels, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns DataRate = As set in the previous step Payload = [0x]0307	All allowed DRs tested

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = An allowed DR for 500 kHz, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0001
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

3	DUT sends Unconfirmed frame FCntUp = n + 2	→	MAC-CMD LinkADRAns Payload = [0x]0307	
4	DUT sends Unconfirmed frame FCntUp = n + 3	→	DUT sends uplinks only on Channel 64 at the DataRate set	Channel and DR tested
5	DUT sends Unconfirmed frame FCntUp = n + 4	→	DUT sends uplinks only on Channel 64 at the DataRate set	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame FCntUp = n + 5	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.ii.2.2. Invalid Command Processing

Using separate downlinks, the **TCL** sends two commands: the first is a valid command disabling all 125 kHz uplink channels with mask index 7, channel mask of [0x]00FF and an allowed DataRate for 500 kHz uplink channels, refer [2]. The second is an invalid *LinkADRReq* command with channel mask control 7, channel mask of [0x]0000 (or [0x]FF00) and an allowed DataRate for 125 kHz uplink channels, refer [2].

Verify

- The **DUT** must reply to the first command with a successful LinkADRAns ([0x]0307) and to the second with an unsuccessful LinkADRAns of (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304. ([0x]0304).
- The DataRate of subsequent uplinks is as set by the **TCL** command.

2.5.8.b.ii.2.2.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 500 kHz uplink channels, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = [0x]0307	Successful LinkADRAns

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0000
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

3	DUT sends Unconfirmed frame FCntUp = n + 2	→	MAC-CMD LinkADRAns Payload = Any one of these-(bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad Channel) [0x]0304.	Unsuccessful LinkADRAns

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.b.ii.3. Disable all Channels (125kHz and 500kHz channels)

Lastly, the **TCL** then tries to disable all channels by sending a LinkADRReq command with channel mask control index of 7 and a channel mask of [0x]0000. The **DUT** *must* reply with an unsuccessful *LinkADRAns* of (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad ChMask) [0x]0304.

2.5.8.b.ii.3.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum refer [2]
			DataRate = An allowed data rate, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0000
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD LinkADRAns DataRate = As set in the LinkADRReq Payload = Any one of these- (bad DR) [0x]0305, or (bad ChMask) [0x]0306 or (both bad DR & bad ChMask) [0x]0304.	Unsuccessful LinkADRAns when disabling all channels

2.5.8.c. Redundancy

This test validates the **DUT**'s correct implementation *NbTrans* setting within the *LinkADRReq* command. The **TCL** sends *LinkADRReq* requesting a repetition count of 2.

Verify

- All subsequent unconfirmed/confirmed uplink frames from the **DUT** are transmitted twice (the same sequence number is received twice by the **TCL**). The number of redundant uplinks verified by the harness is at the discretion of the Test Tool vendor.
- The **DUT** does not repeat the transmission if a downlink is received during the RX1 window.
- The **DUT** does not repeat the transmission if a downlink is received during the RX2 window.
- For Fixed channel plan devices only: The DUT appropriately performs frequency hopping for repeated transmissions.

Once the above criteria are verified for a *NbTrans* setting of 2, the **DUT** is commanded back to the default of 1 by the **TCL** using a *LinkADRReq* command wherein the *NbTrans* value is 0. This default setting is also validated.

The above test is repeated for NbTrans = 3 within the LinkADRReq command.

2.5.8.c.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

3	DUT sends Unconfirmed frame FCntUp = n + 2	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 2
			Payload = [0x]03XXXXXXXX	

4	DUT sends Unconfirmed frame FCntUp = n + 3	→	MAC-CMD LinkADRAns Payload = [0x]0307	
5	DUT sends Unconfirmed frame FCntUp = n + 3	→	MAC-CMD LinkADRAns Payload = [0x]0307 FC plan only: The frequency is different from the previous uplink.	Uplink sent twice
6	DUT sends Unconfirmed frame FCntUp = n + 4	→		
7	DUT sends Unconfirmed frame FCntUp = n + 4	→		
8	DUT sends Unconfirmed frame FCntUp = n + 5	→		

	The TCL sends Unconfirmed frames on **RX1** window	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

9	DUT sends Unconfirmed frame
	FCntUp = n + 6
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 1
				Transmit not repeated when downlink received on RX1

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

10	DUT sends Unconfirmed frame
	FCntUp = n + 6
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 1
			
			FC plan only: The frequency is different from the previous uplink.	

11	DUT sends Unconfirmed frame FCntUp = n + 7	→		

	The TCL sends Unconfirmed frame on **RX2** window	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

12	DUT sends Unconfirmed frame
	FCntUp = n + 8
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 2
				Transmit not repeated when downlink received on RX2

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

13	DUT sends Unconfirmed frame FCntUp = n + 9	→	MAC-CMD LinkADRAns Payload = [0x]0307	Uplink sent once
14	DUT sends Unconfirmed frame FCntUp = n + 10	→		

	The TCL sends Unconfirmed frame	←	CP-CMD TxFramesCtrlReq
			FPort = 224
			Frame type = Confirmed
			Payload = [0x]0702
				Switch to Confirmed frames

15	DUT sends Confirmed frame FCntUp = n + 11	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	Acknowledge
			
			MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 3
			Payload = [0x]03XXXXXXXX	

16	DUT sends **Confirmed** frame FCntUp = n + 12	→	MAC-CMD LinkADRAns Payload = [0x]0307	
17	DUT sends **Confirmed** frame FCntUp = n + 12	→	MAC-CMD LinkADRAns Payload = [0x]0307 FC plan only: The frequency is different from the previous uplink.	
18	DUT sends **Confirmed** frame FCntUp = n + 12	→	MAC-CMD LinkADRAns Payload = [0x]0307 FC plan only: The frequency is different from the previous 2 uplinks.	Uplink sent thrice
19	DUT sends Confirmed frame FCntUp = n + 13	→		
	The TCL sends Unconfirmed frame	←	Acknowledge CP-CMD TxFramesCtrlReq FPort = 224 Frame type = Unconfirmed Payload = [0x]0701	Revert to Unconfirmed frames
20	DUT sends Unconfirmed frame FCntUp = n + 14	→		

	The TCL sends Unconfirmed frame	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

21	DUT sends Unconfirmed frame
	FCntUp = n + 15
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 6
				Transmit not repeated when downlink received

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

22	DUT sends Unconfirmed frame
	FCntUp = n + 15
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 6
			
			FC plan only: The frequency is different from the previous uplink.	

23	DUT sends Unconfirmed frame
	FCntUp = n + 15
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 6
			
			FC plan only: The frequency is different from the previous 2 uplinks.	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 0
			Payload = [0x]03XXXXXXXX	

24	DUT sends Unconfirmed frame FCntUp = n + 16	→	MAC-CMD LinkADRAns Payload = [0x]0307	Uplink sent once
25	DUT sends Unconfirmed frame FCntUp = n + 17	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

26	DUT sends Unconfirmed frame FCntUp = n + 18	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.d. Data Rate Decay

2.5.8.d.i. DR Decay test for all devices

This test validates the **DUT**'s implementation of the *ADR_ACK_LIMIT, ADR_ACK_DELAY*, and use of the *ADRACKReq* bit in its uplinks' frame header. A tolerance of +/- 2 uplinks is allowed for the **DUT**'s implementation of both the *ADR_ACK_LIMIT* and *ADR_ACK_DELAY* parameters.

The **TCL** adds a new channel.
The **TCL** sends *LinkADRReq* to configure the **DUT** to use

- the Max125kHzDR data rate for DC and 500kHz for FC,
- TXPower Index = 1,
- For DC devices: Disable all default channels and enable only the newly added channel,
- For FC devices: Enable channel 0, 1 and 64,
- NbTrans set to 2.

It then stops sending downlinks until the **DUT** decays to the minimum default data rate, refer [2] .

Verify

- After sending *ADR_ACK_LIMIT* (64) uplinks the **DUT** must - in a sustained absence of downlinks - set the *ADRACKReq* bit in the next *ADR_ACK_DELAY* (32) uplinks' frame headers.
- As of its 96th consecutive uplink in the continued absence of downlinks, the **DUT** must change the TXPower Index to 0 and leave the *ADRACKReq* bit set.
- As of its 128th consecutive uplink in the continued absence of downlinks, the **DUT** must lower its data rate to Default, leave the TXPower Index set to 0 and *ADRACKReq* bit set.
- The **DUT** further lowers its data rate by 1 each time another *ADR_ACK_DELAY* cycle occurs (32) uplinks are sent in the sustained absence of downlinks.
- For FC plan devices, when the DR is decayed from a 500kHz DR to a 125KHz DR, Channel 64 must be discontinued and Channels 0 and 1 must be used.
- After the DR reaches MinDR, in the continued absence of downlinks, the **DUT** must re-enable all default uplink frequency channels for DC plan devices, re-enable all channels for FC plan devices, reset *NbTrans* to 1, retain the TXPower at Default, retain the DataRate at Default and the *ADRACKReq* bit is kept set in subsequent uplinks.
- After receiving a downlink, the **DUT** unsets the ADRACKReq bit.

2.5.8.d.i.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

2	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame
	Note: This step is required for only regions with Dwell Time limitation
		←	MAC-CMD
			TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX
				Set TXParamSetupReq for regions with Dwell time limitation

3	DUT sends Unconfirmed frame	→	For regions with Dwell time limitation only MAC-CMD TXParamSetupAns Payload = [0x]09	

	The TCL sends Unconfirmed frame
	Note: This step must be performed for DC plan devices only
		←	MAC-CMD NewChannelReq
			ChIndex = The first non-default channel, refer [2]
			Freq = Any allowed frequency for the channel, refer [2]
			DRRange = 0-5, refer [2]
			Payload = [0x]07XXXXXXXXXX
				Create a new channel

4	DUT sends Unconfirmed frame Note: This step must be performed for DC plan devices only	→	MAC-CMD NewChannelAns Payload = [0x]0703	

	The TCL sends Unconfirmed frame
	Note: This step must be performed for DC plan devices only
		←	MAC-CMD LinkADRReq
			TXPower Index = 1, refer [2]
			DataRate = X (where X = Max125kHzDR), refer [2]
			ChMaskCntl = 0
			ChMask = Disable all default channels and enable only the newly added channel, refer [2]
			NbTrans = 2
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame	→	If the LinkADRReq was sent in the previous step,
			MAC-CMD LinkADRAns
			Payload = [0x]0307
			NbTrans = 2	

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame
	Note: This step must be performed for FC plan devices only
		←	MAC-CMD LinkADRReq
			TXPower Index = 1, refer [2]
			DataRate = Max500kHzDR, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0001
			NbTrans = 2
			MAC-CMD LinkADRReq
			TXPower Index = 1, refer [2]
			DataRate = X (where X = Max500kHzDR), refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0003
			NbTrans = 2
			Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame
	Note: This step must be performed for FC plan devices only
		→	MAC-CMD LinkADRAns
			MAC-CMD LinkADRAns
			NbTrans = 2
			Payload = [0x]0307[0x]0307	

	This is a repeated frame
	DUT sends Unconfirmed frame
	Note: This step must be performed for FC plan devices only
			MAC-CMD LinkADRAns
			MAC-CMD LinkADRAns
			NbTrans = 2
			Payload = [0x]0307[0x]0307
				Repeat as NbTrans = 2

7	DUT sends Unconfirmed frame FCntUp = n Repeat 63 times without receiving any downlinks	→ R [63]	FCtrl ADRAckReq = False DataRate = X NbTrans = 2 TXPower Index = 1 Only the channels enabled must be used	
	This is a repeated frame DUT sends Unconfirmed frame FCntUp = n Repeat 63 times without receiving any downlinks		FCtrl ADRAckReq = False DataRate = X NbTrans = 2 TXPower Index = 1 Only the channels enabled must be used	Repeat as NbTrans = 2
8	DUT sends Unconfirmed frame FCntUp = n + 63 Repeat 32 times without receiving any downlinks (starting with n + 63)	→ R [32]	FCtrl ADRAckReq = True DataRate = X NbTrans = 2 TXPower Index = 1 Only the channels enabled must be used	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
	This is a repeated frame DUT sends Unconfirmed frame FCntUp = n + 63 Repeat 32 times without receiving any downlinks (starting with n + 63)		FCtrl ADRAckReq = True DataRate = X NbTrans = 2 TXPower Index = 1 Only the channels enabled must be used	Repeat as NbTrans = 2
9	DUT sends Unconfirmed frame FCntUp = n + 95 Repeat 32 times without receiving any downlinks (starting with n + 95)	→ R [32]	FCtrl ADRAckReq = True DataRate = X NbTrans = 2 TXPower Index = 0 Only the channels enabled must be used	
	This is a repeated frame DUT sends Unconfirmed frame FCntUp = n + 95 Repeat 32 times without receiving any downlinks (starting with n + 95)		FCtrl ADRAckReq = True DataRate = X NbTrans = 2 TXPower Index = 0 Only the channels enabled must be used	Repeat as NbTrans = 2
10	DUT sends Unconfirmed frame FCntUp = FCntUp (previous) + 32 Repeat 32 times without receiving any downlinks **Note: Repeat this step until the DUT decays to the minimum DR for the region, refer [2]. The test is repeated even when DR = minDR**	→ R [32] until minDR	FCtrl ADRAckReq = True DataRate = Next lower DR, refer [2], until it reaches the minimum DR NbTrans = 2 TXPower Index = 0 Only the channels enabled must be used For FC only: If DataRate decays from 500kHz DR to 125kHz DR, DUT must discontinue usage of Channel 64, and instead use Channel 0 and 1	DUT switches to next lower DR, until it decays to the minimum DR
	This is a repeated frame DUT sends Unconfirmed frame FCntUp = FCntUp (previous) + 32 Repeat 32 times without receiving any downlinks **Note: Repeat this step until the DUT decays to the minimum DR for the region, refer [2]. The test is repeated even when DR = minDR**		FCtrl ADRAckReq = True DataRate = Next lower DR, refer [2], until it reaches the minimum DR NbTrans = 2 TXPower Index = 0 Only the channels enabled must be used For FC only: If DataRate decays from 500kHz DR to 125kHz DR, DUT must discontinue usage of Channel 64, and instead use Channel 0 and 1	DUT switches to next lower DR, until it decays to the minimum DR

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

11	DUT sends Unconfirmed frame	→	FCtrl ADRAckReq = True
			NbTrans = 1
			DataRate = MinDR, refer [2]
			TXPower = 0
				DUT switches to default settings for Nbtrans and channels

12	Wait for a Dynamic channel: maximum of 5 * (number of channels configured) Fixed channel: maximum of 2 * (number of channels configured) uplink packets to be sent, i.e. until all channels are used at least once.	→ R [5*NChDC] or [2*NbChFC] or [AllCh used]	*Official certification (DC plan and FC plan):* All channels configured must be used at least once *Pre-testing for DC Plan:* All default channels must be used at least once. *Pre-testing for FC plan with 8-channel gateway:* Channels 0-7 must be used at least once.	All channels must be used at least once
13	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0 (for DC) and 6 (for FC)
			ChMask = Enable only the default channels for DC, refer [2] and [0x]00FF for FC
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

14	DUT sends Unconfirmed frame	→	FCtrl ADRAckReq = False MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

	The TCL sends Unconfirmed frame	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

15	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame
	
	Note: This step must be performed only for DC plan devices
		←	MAC-CMD NewChannelReq
			ChIndex = The non-default channel added in Step 2 of this table.
			Freq = 0 MHz
			Payload = [0x]07XXXXXXXXXX
				Remove additional channel added

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
16	DUT sends Unconfirmed frame	→	If the NewChannelReq was sent in the previous step, MAC-CMD NewChannelAns Payload = [0x]0703	

	The TCL sends Unconfirmed frame
	Note: This step is required for only regions with Dwell Time limitation
		←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = default, refer [2]
			Payload = [0x]09XX
				Reset Dwell time setting to default

17	DUT sends Unconfirmed frame	→	For regions with Dwell time limitation only MAC-CMD TXParamSetupAns Payload = [0x]09	

2.5.8.d.ii. Additional DR Decay test for only DC plan devices which support the optional data rates

This test validates that the **DUT** re-enables all the default channels when it decays from the Maximum optional Data Rate to Max125kHzDR.

The **TCL** adds a new channel for each optional data rate supported.
The **TCL** sends *LinkADRReq* to configure the **DUT** to use

- the Maximum data rate,
- TXPower Index = 1,
- Disable all default channels and enable only the newly added channel(s),
- NbTrans set to 2.

It then stops sending downlinks until the **DUT** decays from maximum optional data rate to Max125kHzDR, refer [2].

Verify

- After sending *ADR_ACK_LIMIT* (64) uplinks the **DUT** must - in a sustained absence of downlinks - set the *ADRACKReq* bit in the next *ADR_ACK_DELAY* (32) uplinks' frame headers.
- As of its 96th consecutive uplink in the continued absence of downlinks, the **DUT** must change the TXPower Index to 0 and leave the *ADRACKReq* bit set.
- As of its 128th consecutive uplink in the continued absence of downlinks, the **DUT** must lower its data rate to Default, leave the TXPower Power set to 0 and *ADRACKReq* bit set.
- The **DUT** further lowers its data rate by 1 each time another *ADR_ACK_DELAY* cycle occurs (32) uplinks are sent in the sustained absence of downlinks.

- After the DR reaches decays from an optional DataRate to a Max125kHzDR, in the continued absence of downlinks, the **DUT** must re-enable all default uplink frequency channels.

2.5.8.d.ii.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

2	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame
	Note: This step is required for only regions with Dwell Time limitation
		←	MAC-CMD
			TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX
				Set TXParamSetupReq for regions with Dwell time limitation

3	DUT sends Unconfirmed frame	→	For regions with Dwell time limitation only MAC-CMD TXParamSetupAns Payload = [0x]09	

	The TCL sends Unconfirmed frame
	Note: This step must be performed only if the DUT supports 250kHz channels
		←	MAC-CMD NewChannelReq
			ChIndex = An unused non-default channel, refer [2]
			Freq = Any allowed freq for a 250kHz channel, refer [2]
			DRRange = BW250OptionalDR-BW250OptionalDR, refer [2]
			Payload = [0x]07XXXXXXXXXX
				Create a new channel for 250kHz channel

4	DUT sends Unconfirmed frame	→	If the NewChannelReq was sent in the previous step, MAC-CMD NewChannelAns Payload = [0x]0703	

	The TCL sends Unconfirmed frame
	Note: This step must be performed only if the DUT supports FSK channels
		←	MAC-CMD NewChannelReq
			ChIndex = An unused non-default channel, refer [2]
			Freq = Any allowed frequency for an FSK channel, refer [2]
			DRRange = FSK50OptionalDR-FSK50OptionalDR, refer [2]
			Payload = [0x]07XXXXXXXXXX
				Create a new channel for FSK channel if supported

5	DUT sends Unconfirmed frame	→	If the NewChannelReq was sent in the previous step, MAC-CMD NewChannelAns Payload = [0x]0703	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower Index = 1, refer [2]
			DataRate = X (where X = Maximum Supported Data Rate), refer [2]
			ChMaskCntl = 0
			ChMask = Disable all default channels and enable only the newly added channel(s), refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

6	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	
7	DUT sends Unconfirmed frame FCntUp = n Repeat 63 times without receiving any downlinks	→ R [63]	FCtrl ADRAckReq = False DataRate = X TXPower Index = 1 Only the channels enabled must be used	
8	DUT sends Unconfirmed frame FCntUp = n + 63 Repeat 32 times without receiving any downlinks (starting with n + 63)	→ R [32]	FCtrl ADRAckReq = True DataRate = X TXPower Index = 1 Only the channels enabled must be used	
9	DUT sends Unconfirmed frame FCntUp = n + 95 Repeat 32 times without receiving any downlinks (starting with n + 95)	→ R [32]	FCtrl ADRAckReq = True DataRate = X TXPower Index = 0 Only the channels enabled must be used	
10	DUT sends Unconfirmed frame FCntUp = FCntUp (previous) + 32 Repeat 32 times without receiving any downlinks **Note: Repeat this step until the DUT decays to the Max125kHzDR, refer [2]. When DR = Max125kHzDR, skip to the next step.**	→ R [32] Until DUT switches from the OptionalDR to Max125kHz DR	FCtrl ADRAckReq = True DataRate = Next lower DR, refer [2], until it reaches Max125kHzDR, refer [2] TXPower Index = 0 Only the channels enabled must be used	DUT switches to next lower DR, until it decays to the Max125kHz DR
11	DUT sends Unconfirmed frame	→	FCtrl ADRAckReq = True DataRate = Max125kHzDR, refer [2] TXPower Index = 0 All default channels are enabled	DUT reenables all default channels

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
12	Wait for a maximum of 5 * (number of channels configured) uplink packets to be sent, i.e. until all channels are used at least once.	→ R [5*NChDC] or [AllDefCh used]	All default channels must be used at least once	All default channels must be used at least once
13	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask = Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

14	DUT sends Unconfirmed frame	→	FCtrl ADRAckReq = False MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

	The TCL sends Unconfirmed frame	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

15	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame
	
	Note: This step is required for only regions with Dwell Time limitation
		←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = default, refer [2]
			Payload = [0x]09XX
				Reset Dwell time to default

16	DUT sends Unconfirmed frame	→	For regions with Dwell time limitation only MAC-CMD TXParamSetupAns Payload = [0x]09	

	The TCL sends Unconfirmed frame
	
	Note: This step must be performed only if the DUT supports 250kHz channels
		←	MAC-CMD NewChannelReq
			ChIndex = The channel used in Step3
			Freq = 0
			DRRange = BW250OptionalDR-BW250OptionalDR, refer [2]
			Payload = [0x]07XXXXXXXXXX
				Removes the 250kHz channel

17	DUT sends Unconfirmed frame	→	If the NewChannelReq was sent in the previous step, MAC-CMD NewChannelAns Payload = [0x]0703	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	

	The TCL sends Unconfirmed frame
	Note: This step must be performed only if the DUT supports FSK channels
		←	MAC-CMD NewChannelReq
			ChIndex = The channel used in Step4
			Freq = 0
			DRRange = FSK50OptionalDR-FSK50OptionalDR, refer [2]
			Payload = [0x]07XXXXXXXXXX
				Removes the FSK channel

18	DUT sends Unconfirmed frame	→	If the NewChannelReq was sent in the previous step, MAC-CMD NewChannelAns Payload = [0x]0703	

2.5.8.e. Command Block Channel Management

This test validates the **DUT** correctly processes blocks of multiple *LinkADRReq* commands included in the same downlink. The **DUT** is expected to service each of these MAC commands in the same sequence as it is ordered in the FOpts or FRMPayload field.

2.5.8.e.i. Dynamic channel plan devices

2.5.8.e.i.1.Successful LinkADRReq block

The **TCL** sends multiple *LinkADRReq* commands in a single downlink as shown**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.** in Table 1: Successful LinkADRReq block**Error! Reference source not found.Error! Reference source not found.Error! Reference source not found..** The first command tries to disable all channels. The channel mask the second command enables is *channel 0*, and any default DR except the minimum DR, is appropriate for this channel. The last command enables all defined channels using the channel mask control 6 and configures any other default data rate except the ones used earlier for this test. The **DUT** is expected to respond with *LinkADRAns [0x]0307* for all *LinkADRReq* commands.

LinkADRReq CMD Index	DR	TXPower	Channel Mask	MaskCntl	NbTrans
1	Minimum	Any allowed TXPower	[0x]0000	0	0
2	Any default DR except the one set above	Any allowed TXPower, other than the one set above	[0x]0001	0	0

3	Any default DR except the ones set above	Any other allowed TXPower	[0x]0000	6	1

Table 1: **Successful LinkADRReq block**

Note:

- The DR, TX Power, and NbTrans values of only the last command are implemented by the **DUT**. Values for these parameters in any preceding command must be ignored.
- After processing the command block shown above, the **DUT** is expected to be configured as follows:
  Channel Plan: Default channels only
  Data Rate: As set in Step 3 in Table 1: Successful LinkADRReq block
  TXPower: As set in Step 3 in **Error! Reference source not found.**Table 1: Successful LinkADRReq block

Verify

- The **DUT** replies with an uplink containing a successful *LinkADRAns* for each command in the block.
- The uplink containing the answers and all subsequent transmissions are sent using the default channels at the commanded DR.

2.5.8.e.i.1.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD1 LinkADRReq
			TXPower = Any allowed TXPower, refer [2]
			DataRate = MinDR, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0000
			NbTrans = 0
			
			MAC-CMD2 LinkADRReq
			TXPower = Any other allowed TXPower, refer [2]
			DataRate = Any default DR, except the one set above in this step, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0001
			NbTrans = 0
			
			MAC-CMD3 LinkADRReq
			TXPower = Any other TXPower, refer [2]
			DataRate = Any default DR, except the ones set above in this step, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]0000
			NbTrans = 1
			
			Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX[0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	DataRate = same as set in CMD3 above MAC-CMD1 LinkADRAns MAC-CMD2 LinkADRAns MAC-CMD3 LinkADRAns Payload = [0x]0307[0x]0307[0x]0307	Uplinks as configured
3	Wait for a maximum of 5 * (number of channels configured) = uplink packets to be sent, i.e. until all default channels are used at least once.	→ R [5*NbCh] or [All Ch used]	All default channels must be used at least once	All default channels must be used at least once
4	DUT sends Unconfirmed frame	→		

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask = Enable only the default channels, refer [2]
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

5	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.8.e.i.2.Unsuccessful LinkADRReq block

The **TCL** sends several *LinkADRReq* commands in a single downlink as shown in Table 2: Unsuccessful LinkADRReq block. **Error! Reference source not found.**In this case the last command requires all channels to be disabled, all commands must be rejected.

LinkADRReq CMD Index	DR	TXPower	Channel Mask	MaskCntl	NbTrans
1	Any mandatory DR, except Max125kHzDR	Any	[0x]0001	0	0
2	Any other mandatory DR, except Max125kHzDR	Any	[0x]0000	6	1
3	Max125kHzDR	Any	[0x]0000	0	1

Table 2**: Unsuccessful LinkADRReq block**

Verify

- The **DUT** replies with an uplink containing an unsuccessful LinkADRAns for each command in the block.
- The uplink data rate does not change and the **DUT** continues using the default channels at the previously configured data rate.

2.5.8.e.i.2.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD1 LinkADRReq
			TXPower = Any allowed value, refer [2]
			DataRate = X (where X = any mandatory DR, except Max125kHzDR, refer [2])
			ChMaskCntl = 0
			ChMask = [0x]0001
			NbTrans = 0
			
			MAC-CMD2 LinkADRReq
			TXPower = Any allowed value, refer [2]
			DataRate = Any other mandatory DR, except Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]0000
			NbTrans = 1
			
			MAC-CMD3 LinkADRReq
			TXPower = Any allowed value, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]0000
			NbTrans = 1
			
			Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX[0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD1 LinkADRAns MAC-CMD2 LinkADRAns MAC-CMD3 LinkADRAns Payload = [0x]030X[0x]030X[0x]030X (where X is NOT = 7)	All commands are rejected as the last command requires all channels to be disabled

2.5.8.e.ii. Fixed channel plan devices

2.5.8.e.ii.1. 125kHz Sub-Band Channel Plan

The **TCL** sends two *LinkADRReq* commands in a single downlink as shown in Table 3: 125kHz Sub-Band Channel Plan Command Block. The first command disables all 125 kHz channels and simultaneously enables the *channel 64* (500 kHz channel). The DataRate of the first command is any data rate allowed for uplinking on 500 kHz enabled channels. The channel mask control of the second command enables *channel 0 – 7*. The DataRate must be an allowed data rate allowed for 125 kHz uplink channels, refer [2]. The **DUT** is expected to respond with *LinkADRAns [0x]0307* for both *LinkADRReq* commands.

LinkADRReq CMD Index	DR	TXPower	Channel Mask	Mask Index	NbTrans
1	Any DR allowed for 500 kHz uplink channels	Maximum	[0x]0001	7	1
2	Any DR allowed for 125 kHz uplink channels	Any allowed TXPower other than max TXPower	[0x]00FF	0	1

Table 3: **125kHz Sub-Band Channel Plan Command Block**

Note:

- The DR, TX Power, and NbTrans values of only the last command are implemented by the **DUT**. Values for these parameters in any preceding command must be ignored.
- After processing the command block shown above, the **DUT** is expected to be configured as follows:
  Channel Plan: Channels 0 – 7
  Data Rate: DataRate set in the second command above
  TXPower: TXPower set in the second command above

Verify

- The **DUT** replies with an uplink containing a successful *LinkADRAns* for each command in the block.
- The uplink containing the answers and all subsequent transmissions are sent only on the enabled 125kHz channel plan at the commanded DR.

2.5.8.e.ii.1.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD1 LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Any DR allowed for 500 kHz uplink channels, refer [2]
			ChMaskCntl = 7
			ChMask = [0x]0001
			NbTrans = 1
			
			MAC-CMD2 LinkADRReq
			TXPower = Any allowed TXPower other than max TXPower, refer [2]
			DataRate = Any DR allowed for 125 kHz uplink channels, refer [2]
			ChMaskCntl = 0
			ChMask = [0x]00FF
			NbTrans = 1
			
			Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame FCntUp = n + 1	→	DataRate and TXPower = same as set in CMD2 above Channel Plan = Channel 0-7 MAC-CMD1 LinkADRAns MAC-CMD2 LinkADRAns Payload = [0x]0307[0x]0307	Uplinks as configured

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			TXPower = Maximum, refer [2]
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl = 6
			ChMask = [0x]00FF
			NbTrans = 1
			Payload = [0x]03XXXXXXXX	

3	DUT sends Unconfirmed frame FCntUp = n + 2	→	MAC-CMD LinkADRAns Payload = [0x]0307	DUT reverted to default settings

2.5.9. DutyCycleReq

The test validates that the **DUT** correctly updates its Duty Cycle – maximum aggregated transmit duty time.

TCL must then set the **DUT** to the maximum DataRate, refer [2].
DUT sends uplink frames and **TCL** stores the time of arrival of 2 consecutive uplink frames.
TCL sends *DutyCycleReq* MAC command with a *MaxDutyCycle* value of 7 (Duty Cycle is smaller than 1% duty-cycle used for default channels)
DUT again sends uplink frames and **TCL** stores the time of arrival of the next 2 consecutive uplink frames.

Verify

- **DUT** responds with a *DutyCycleAns* command, adjusts its duty cycle as requested by the **TCL** and transmits the uplink frames less frequently after the *DutyCycleReq* is sent.

2.5.9.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125KHzDR, refer [2]
			Payload = [0x]03XXXXXXXX
			
			ChMaskCntl:
			DC = 0,
			FC = 6
			
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
				Set DataRate to Max125kHzDR

2	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns Payload = [0x]0307 Time of Arrival = A	
3	DUT sends Unconfirmed frame	→	Time of Arrival = B	
	The TCL sends Unconfirmed frame	←	MAC-CMD DutyCycleReq Payload = [0x]0407	
4	DUT sends Unconfirmed frame	→	MAC-CMD DutyCycleAns Payload = [0x]04 Time of Arrival = C	Max Duty Cycle set to 7
5	DUT sends Unconfirmed frame	→	Time of Arrival = D Verify that ((B - A) + 5 sec) < (D – C)	Uplink frames are sent less frequently after the duty cycle is changed.
6	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD DutyCycleReq Payload = [0x]0400	
7	DUT sends Unconfirmed frame	→	MAC-CMD DutyCycleAns Payload = [0x]04	Revert to default Duty cycle

2.5.10. DeviceTimeReq

The **DUT** is triggered to request the **TCL** for the current network time and the **TCL** must correctly send the network time for the **DUT** to synchronize its time.

TCL must trigger the **DUT** to send a *DeviceTimeReq* to the **TCL**. The **TCL** must reply with the *DeviceTimeAns* providing the current network time to the **DUT.** The **DUT** must return the value received on *DeviceTimeAns* in the Payload of the next uplink.

Verify

- **DUT** sends a *DeviceTimeReq* to the **TCL**.
- **DUT** resumes normal operation after **TCL** sends *DeviceTimeAns*.

2.5.10.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

2	TCL sends Unconfirmed frame	←	CP-CMD DeviceTimeReq
			FPort = 224
			Payload = [0x]21	

3	DUT sends Unconfirmed frame	→	MAC-CMD DeviceTimeReq Payload = [0x]0D	
4	The TCL sends Unconfirmed frame	←	MAC-CMD DeviceTimeAns Payload = [0x]0DXXXXXXXXXX	DeviceTimeAns sent
5	DUT sends Unconfirmed frame	→		

2.5.11. RX Window test

2.5.11.a. RX1 Receive Window Test

This test validates the **DUT**'s capability to receive data on RX1 for data rates as specified in RX1DRoffset table in [2].

For each combination of Operating Uplink DR to RX1 offset:

- The **TCL** commands the **DUT** to implement the target configuration and subsequently validates corresponding *LinkADRAns* and *RXParamSetupAns* from the **DUT** indicating it has implemented the target configuration.
- The **TCL** then sends 5 consecutive downlinks to the **DUT** on the RX1 window where:
    - At least one of the frames is the maximum allowed payload for the current RX1 DR (with respect to the current offset).
    - At least one of the frames is confirmed, requiring the **DUT** to set the ACK bit in the subsequent uplink.
- The **TCL** validates reception by verifying for each downlink that the subsequent uplink contains a payload indicating the **DUT**'s downlink counter has incremented by one. The uplink following the confirmed downlink must have its ACK bit set to true as well.

For the given RX1DRoffset, the **TCL** then transitions the **DUT** through the remaining Uplink DR settings, verifying the respective *LinkADRAns* and repeating the above procedure for each.

Once the combinations for the current RX1DRoffset are exhausted, the **TCL** commands the **DUT** to the next RX1DRoffset, verifies the *RXParamSetupAns* and repeats the same series of downlink tests. The **TCL** proceeds in this fashion exhausting all combinations.

For each unique RX1 DR in the RX1DRoffset table, the **TCL** will perform the Sufficient Reception test. For these tests RX1DRoffset will be set to default. The **TCL** sends 60 downlink packets in the RX1 window and verifies that at least 57 frames were received by the **DUT**.

2.5.11.a.i. Test Procedure Frame Sequence Chart

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		
	For DC only – If the DUT supports optional data rate(s), then The TCL sends Unconfirmed frame	←	CMD NewChannelReq ChannelIndex = any unused optional channel Frequency = any allowed frequency for the channel, refer [2] Payload = 0x07XXXXXXXXXX Note: If the DUT supports more than one optional data rate, a NewChannelReq must be sent for each of them.	
2	If the NewChannelReq was sent by the TCL, then DUT sends Unconfirmed frame	→	CMD NewChannelAns Payload = [0x]0703[Repeat for all channel added]	Added new channel(s) for the optional data rate(s) supported
	*For each RX1DRoffset combination in the RX1DRoffset table in [2], the **TCL** transitions the **DUT** to each Uplink Data Rate by repeating below Steps 3-9.*	**R** Steps 3-9 for [All RX1DR in RX1DRoffset table]		
3	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq DataRate = X (where X = DR as defined in the RX1DRoffset table in [2], if DR is supported) ChMaskCntl: DC = 0, FC = 6 ChMask: DC - Enable only default channels FC = [0x]00FF MAC-CMD RxParamSetupReq RX1DRoffset = Offset value as defined in the RX1DRoffset table in [2] RX2DataRate = default DR, refer [2] RX2Frequency = default frequency, refer [2] Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
4	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

5	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

6	DUT sends Unconfirmed frame *Repeat 4 times*	→ R [4]		

	The TCL sends Unconfirmed frames on RX1 window
	Repeat 4 times
		← R [4]	CP-CMD TxFramesCtrlReq
			FPort = 224
			Frame type = No change
			Payload = [0x]0700	

7	DUT sends Unconfirmed frame	→		
	The TCL sends a **Confirmed** frame on RX1 window	←	Payload = Max allowed payload as defined in Maximum Payload size table in [2]	
8	DUT sends Unconfirmed frame	→	ACK Bit = True	

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

9	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt >= x + 4	

	Repeat steps 3-9 for each RX1DRoffset combination in the RX1DRoffset table in [2].			
	For each unique RX1 Data Rate in the RX1DRoffset table in [2]*, the* ***TCL*** *will perform the Sufficient Reception test by repeating Steps 10-15.*	**R** Steps 10-15 for [All RX1DR in RX1DRoffset table]		
10	DUT sends Unconfirmed frame	→		

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq DataRate = X (where X = DR as defined in the RX1DRoffset table in [2], if DR is supported) ChMaskCntl: DC = 0, FC = 6 ChMask: DC - Enable only default channels FC = [0x]00FF MAC-CMD RxParamSetupReq RX1DRoffset = 0 Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	
11	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

12	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = y	

13	DUT sends Unconfirmed frame *Repeat 60 times*	→ R [60]		

	The TCL sends Unconfirmed frame on RX1 window
	
	Repeat 60 times
		← R [60]	CP-CMD TxFramesCtrlReq
			DataRate = X
			FPort = 224
			Frame type = No change
			Payload = [0x]0700 (MaxLen for RX1 DR)	

14	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

15	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns FPort = 224 Payload = [0x]09XXXX RxAppCnt <= y + 60 + 1 RxAppCnt >= y + 57 + 1	Reception failure rate <=5%

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
	Repeat steps 10-15 for each unique RX1 Data Rate in the RX1DRoffset table in [2]			
16	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = default DR, refer [2]
			Freq = default frequency, refer [2]
			
			Payload = [0x]03XXXXXXXX
			[0x]05XXXXXXXX	

17	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	DUT reverted to default settings

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

18	DUT sends Unconfirmed frame	→		

	For DC only –
	If the DUT supports optional data rate(s), then
	
	The TCL sends Unconfirmed frame
		←	CMD NewChannelReq
			ChannelIndex = as added in Step 1
			Frequency = 0
			Payload = 0x07XXXXXXXXXX
			
			Repeat the command for all channels added in Step 1 of this table	

19	If the NewChannelReq was sent by the TCL, then DUT sends Unconfirmed frame	→	CMD NewChannelAns Payload = 0x0703[Repeat for all optional channels added]	Removed channel(s) added

2.5.11.b. RX2 Receive Window Test

This test validates the **DUT**’s capability to receive data on RX2 for all data rates.

For each unique RX2DataRate, the **TCL** will perform the Sufficient Reception. For these tests RX1DROffset will be set to default. The **TCL** sends 60 downlink packets in the RX2 window and verifies that at least 57 messages were received by the **DUT**.

2.5.11.b.i. Test Procedure Message Sequence Chart

Step	Procedure	Message Sequence		Test Purpose
		End Device - TCL	**Message**	
1	DUT sends Unconfirmed frame	→		
	For each RX2DataRate supported by the DUT, the ***TCL*** *will perform the Sufficient Reception test by repeating Steps 2-7.*	**R** Steps 2-7 for [All RX2DR]		
2	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD RxParamSetupReq RX1DROffset = default, refer [2] RX2DataRate = X (where X = each supported data rate for RX2) RX2 Frequency = default, refer [2] Payload = [0x]05XXXXXXXX	
3	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload = [0x]0507	

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

4	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = y	

5	DUT sends Unconfirmed frame	→ R [60]		

	The TCL sends Unconfirmed frame on RX2 window
	Repeat 60 times
	For each unique RX2 Data Rate, the TCL will perform the Sufficient Reception test.
		← R [60]	CP-CMD TxFramesCtrlReq
			FrameType = No change
			DataRate = X
			FPort = 224
			Payload = [0x]0700 (MaxLen for RX2 DR)	

6	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

7	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns FPort = 224 Payload = [0x]09XXXX RxAppCnt <= y + 60 +1 RxAppCnt >= y + 57 +1	Reception failure rate <=5%
	Repeat above steps 2-7 until all RX2 DataRates are tested			
8	DUT sends Unconfirmed frame	→		

Step	Procedure	Message Sequence		Test Purpose
		End Device - TCL	Message	

	The TCL sends Unconfirmed frame	←	MAC-CMD
			RxParamSetupReq
			RX1DROffset = default, refer [2]
			RX2DataRate = default, refer [2]
			RX2 Frequency = default, refer [2]
			Payload = [0x]05XXXXXXXX
				Revert to default RX2 DR

9	DUT sends Unconfirmed frame	→	MAC-CMD RxParamSetupAns Payload = [0x]0507	

2.5.11.c. RX1 and RX2 simultaneous frames

This test validates that when **TCL** sends frames on both RX1 and RX2 windows simultaneously, **DUT** responds to the frame on the RX1 window and rejects the frame on the RX2 window.

2.5.11.c.i. Test Procedure Message Sequence Chart

Step	Procedure	Message Sequence		Test Purpose
		End Device - TCL	Message	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD TxFramesCtrlReq
			FrameType = No change
			FPort = 224
			Payload = [0x]0700	

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD TxFramesCtrlReq
			FrameType = No change
			FPort = 224
			Payload = [0x]0700	

3	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

4	DUT sends Unconfirmed frame	→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 2
				RX2 frame is ignored if DUT received downlink in RX1

2.5.11.d. RX Oversized Payload

A follow-up negative test must be performed for each oversized scenario. The Max Payload size for each region is defined in [2]. After commanding the **DUT** such that the target RX DR is achieved, the TCL sends a downlink whose payload is one byte greater than the scenario's maximum. Note the payload content is random (i.e. not the echo command).

The TCL must use both confirmed and unconfirmed oversized frames; as well as target both receive windows.

Verify

- The **DUT** continues normal operation in the presence of oversized downlinks. DUT must silently discard the oversized downlinks.

2.5.11.d.i. Test Procedure Frame Sequence Chart

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

	Repeat Steps 2-7 'i' times, where i = all possible combinations in the Maximum Payload size table in [2]	**R** Steps 2-7 for [All DR in Max Payload table]		
2	DUT sends Unconfirmed frame	→		
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq DataRate = X (where X = Each DR as defined in the Maximum Payload size table in [2]) ChMaskCntl: DC = 0, FC = 6 ChMask: DC - Enable only default channels FC = [0x]00FF MAC-CMD RxParamSetupReq RX1DRoffset = 0 RX2DataRate = Y (where Y = Each RX2DataRate as defined in the Maximum Payload size table in [2]) Freq = default frequency [2] Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	
3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	

	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD TxFramesCtrlReq
			FPort = 224
			FrameType = Confirmed
			Payload (i) = [0x]0702(MaxLen + 1 for UL DR-X)	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
4	DUT sends Unconfirmed frame	→		DUT discards the oversized frame and sends Unconfirmed frame
	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD TxFramesCtrlReq FPort = 224 FrameType = No change Payload (i) = [0x]0700 (MaxLen + 1 for UL DR-Y)	
5	DUT sends Unconfirmed frame	→		DUT continues normal operation
	The TCL sends **Confirmed** frame on RX1 window	←	CP-CMD TxFramesCtrlReq FPort = 224 FrameType = No change Payload (i) = [0x]0700 (MaxLen + 1 for UL DR-X)	
6	DUT sends Unconfirmed frame	→	No ACK	DUT continues normal operation (RX1)
	The TCL sends **Confirmed** frame on RX2 window	←	CP-CMD TxFramesCtrlReq FPort = 224 FrameType = No change Payload (i) = [0x]0700 (MaxLen + 1 for UL DR-Y)	
7	DUT sends Unconfirmed frame	→	No ACK	DUT continues normal operation (RX2)
	Repeat above steps 2-7 until all Data Rates in the Max Payload table are tested			
8	DUT sends Unconfirmed frame	→		

Step	**Procedure**	**Frame Sequence**		**Test Purpose**
		End Device - TCL	**Frame**	

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = default, refer [2]
			Freq = default frequency, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

9	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	DUT reverted to default settings

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

10	DUT sends Unconfirmed frame	→		

2.5.11.e. **Maximum Allowed Payload**

The purpose of this test is to validate the maximum allowable payload sizes in uplink transmissions. The test depends on properly functioning *LinkADRReq* and *RXParamSetupReq* MAC command control. The first part validates the maximum uplink payload for each of the five uplink data rates against both receive windows, testing both positive and negative scenarios. The second part validates the maximum downlink payload for the various RX1DR offsets as well as various RX2 data rates.

The **TCL** commands the **DUT** with some combination of the following settings and validates the device successfully implements them before proceeding to test max payload handling:

- RX1DRoffset = 0
- RX2DataRate = Maximum default Data Rate, refer [2]

The Maximum Payload size for each region is defined in [2].

2.5.11.e.i. Max Payload via Echo

These tests are performed on both RX1 and RX2 Windows. For each ‘Uplink DR’, the **TCL** first commands the **DUT** to use the target ‘Uplink DR’ and verifies the configuration. It then sends an echo command ([0x]08) whose payload size is the current maximum allowed for the uplink DR as defined in [2].

Verify

- The **DUT** responds to each echo command with an echo answer.
- The payload size is in fact the maximum for the **DUT**’s current DR and its content is correct.

2.5.11.e.i.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

	Repeat Steps 2-5 'i' times, where i = all possible combinations in the Maximum Payload size table in [2]	**R** Steps 2-5 for [All DR in Max payload table]		
2	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = X (where X = Each DR as defined in the Maximum Payload size table in [2])
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD
			RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = Maximum default DR, refer [2]
			Freq = default frequency [2]
			
			Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507 DataRate = X	
	The TCL sends Unconfirmed frame on RX1 window	←	CP-CMD EchoPayloadReq FPort = 224 Payload (i) = [0x]08XX (MaxLen for UL DR-X)	

4	DUT sends Unconfirmed frame	→	DataRate = X
			
			CP-CMD EchoPayloadAns
			FPort = 224
			Payload (i)' = [0x]08XX (MaxLen for DR-X)
				DUT echos MaxLen PDU for each Uplink DR on RX1

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq FPort = 224 Payload (i) = [0x]08XX (MaxLen for UL DR-X)	

5	DUT sends Unconfirmed frame	→	DataRate = X
			
			CP-CMD EchoPayloadAns
			FPort = 224
			Payload (i)' = [0x]08XX (MaxLen for DR-X)
				DUT echos MaxLen PDU for each Uplink DR on RX2

	Repeat above Steps 2-5 for all DR in Max payload size table			
6	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = default DR, refer [2]
			Freq = default frequency, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

7	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	DUT reverted to default settings

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

8	DUT sends Unconfirmed frame	→		

2.5.11.e.ii. Oversized Payload via Echo

These tests are performed on RX2 Window only. For each ‘Uplink DR’, the **TCL** first commands the **DUT** to use the target ‘Uplink DR’ and verifies the configuration. It then sends an echo command ([0x]08) whose payload size is one byte greater than the current maximum allowed for the uplink DR. The Maximum Payload size for each region is defined in [2].

Verify

- The **DUT** does not uplink an echo response for any of the echo commands due to uplink size limitation. It may either respond with an uplink that has no payload, or silently discard the echo-command.
- Subsequent to the first uplink sent after receiving the command, uplinks must resume containing only the default payload (downlink counter).

2.5.11.e.ii.1. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frames	←	CP-CMD
			RegionalDutyCycleCtrlReq-OFF
			FPort = 224
			Payload = [0x]0500	

	Repeat Steps 2-4 'i' times, where i = all possible combinations in the Maximum Payload size table in [2]	**R** Steps 4-6 [All DR in Max Payload table]		
2	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = X (where X = Each DR as defined in the Maximum Payload size table in [2])
			ChMaskCntl:
			DC = 0,
			FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = Maximum default DR, refer [2]
			Freq = default frequency, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

3	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns
			
			MAC-CMD RxParamSetupAns
			
			Payload = [0x]0307[0x]0507
			
			DataRate = X	

	The TCL sends Unconfirmed frame on RX2 window	←	CP-CMD EchoPayloadReq FPort = 224 Payload (i) = [0x]08XX (MaxLen + 1 for UL DR-X)	

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
4	DUT sends Unconfirmed frame	→	Payload does not exceed limits	DUT continues normal operation in the presence of oversized downlinks after each Uplink DR on RX2
	Repeat above Steps 2-4 for all DR in Max payload size table			
5	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			DataRate = Max125kHzDR, refer [2]
			ChMaskCntl:
			DC = 0; FC = 6
			ChMask:
			DC - Enable only default channels
			FC = [0x]00FF
			
			MAC-CMD RxParamSetupReq
			RX1DRoffset = 0
			RX2DataRate = default DR, refer [2]
			Freq = default frequency, refer [2]
			
			Payload = [0x]03XXXXXXXX[0x]05XXXXXXXX	

6	DUT sends Unconfirmed frame	→	MAC-CMD LinkADRAns MAC-CMD RxParamSetupAns Payload = [0x]0307[0x]0507	DUT reverted to default settings

	The TCL sends Unconfirmed frames	←	CP-CMD RegionalDutyCycleCtrlReq-ON
			FPort = 224
			Payload = [0x]0501	

7	DUT sends Unconfirmed frame	→		

2.5.12. MAC Command(s) in App-Payload and/or Frame Options

These tests ensure a **DUT** appropriately accepts and processes or discards a downlink whose contents include one or more MAC commands in the App-Payload (i.e. FRMPayload) and/or the Frame options (FOpts) portion of the frame.

2.5.12.a. App-Payload Only (FPort = 0)

Perform the test listed below such that all MAC commands are sent to the **DUT** within the App-Payload portion of a single downlink. Successful completion of each test indicates the MAC Command payload is properly decrypted using the Network Session Key and processed by the **DUT**. It is up to the **DUT** to choose how it answers: it can choose either the FOpts field or the FRMPayload. Some devices for example may use the FRMPayload when the MAC command answers are larger than the 15 bytes limit of the FOpts field.

- DevStatusReq
- RxParamSetupReq
- RxTimingSetupReq
- LinkADRReq

2.5.12.a.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		
	The TCL sends Unconfirmed frame	←	<u>App-Payload</u> MAC-CMD1 DevStatusReq MAC-CMD2 RxParamSetupReq MAC-CMD3 RxTimingSetupReq MAC-CMD4 LinkADRReq Payload = [0x]06[0x]05XXXXXXXX[0x]08XX[0x]03XXXXXXXX FPort = 0	
2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD1 DevStatusAns MAC-CMD2 RxParamSetupAns MAC-CMD3 RxTimingSetupAns MAC-CMD4 LinkADRAns Payload = [0x]06XXXX[0x]0507[0x]08[0x]0307	Successful completion of each command

3	The TCL sends Unconfirmed frame	←	CP-CMD TxFramesCtrlReq
			FrameType = No change
			FPort = 224
			Payload = [0x]0700	

	DUT sends Unconfirmed frame FCntUp = n + 2	→		

2.5.12.b. Frame Options Only (FPort NOT = 0)

Perform the test listed below such that all MAC commands are sent to the **DUT** within the Frame Options field (i.e. FOpts) portion of a single downlink. Successful completion of each test indicates the MAC Command payload is properly decrypted using the Application Session Key and processed by the **DUT**. It is up to the **DUT** to choose how it answers: it can choose either the FOpts field or the FRMPayload. Some devices for example may use the FRMPayload when the MAC command answers are larger than the 15 bytes limit of the FOpts field.

- DevStatusReq
- RxParamSetupReq
- RxTimingSetupReq
- LinkADRReq

2.5.12.b.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	**Frame**	
1	DUT sends Unconfirmed frame FCntUp = n	→		
	The TCL sends Unconfirmed frame	←	Frame Options MAC-CMD1 DevStatusReq MAC-CMD2 RxParamSetupReq MAC-CMD3 RxTimingSetupReq MAC-CMD4 LinkADRReq Payload = [0x]06[0x]05XXXXXXXX[0x]08XX[0x]03XXXXXXXX FPort NOT = 0	
2	DUT sends Unconfirmed frame FCntUp = n + 1	→	MAC-CMD1 DevStatusAns MAC-CMD2 RxParamSetupAns MAC-CMD3 RxTimingSetupAns MAC-CMD4 LinkADRAns Payload = [0x]06XXXX[0x]0507[0x]08[0x]0307	Successful completion of each command

3	The TCL sends Unconfirmed frame	←	CP-CMD TxFramesCtrlReq
			FrameType = No change
			FPort = 224
			Payload = [0x]0700	

	DUT sends Unconfirmed frame FCntUp = n + 2	→		

2.5.12.c. App-Payload and Frame Options

Verify the **DUT** ignores a downlink when MAC commands are simultaneously present in the App-Payload and Frame options (FOpts) fields.

2.5.12.c.i. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame FCntUp = n	→		

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

2	DUT sends Unconfirmed frame
	FCntUp = n + 1
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x	

	The TCL sends Unconfirmed frame	←	Frame Options MAC-CMD1 LinkADRReq MAC-CMD2 LinkADRReq MAC-CMD3 RxParamSetupReq Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX[0x]05XXXXXXXX App-Payload MAC-CMD1' LinkADRReq MAC-CMD2' RxTimingSetupReq MAC-CMD3' DevStatusReq Payload = [0x]03XXXXXXXX[0x]08XX[0x]06	
3	DUT sends Unconfirmed frame FCntUp = n + 2	→		Downlink ignored when MAC commands are present simultaneously in App-Payload and FOpts field
	The TCL sends **Confirmed** frame	←	Frame Options MAC-CMD1 LinkADRReq MAC-CMD2 LinkADRReq MAC-CMD3 RxParamSetupReq Payload = [0x]03XXXXXXXX[0x]03XXXXXXXX[0x]05XXXXXXXX App-Payload MAC-CMD1' LinkADRReq MAC-CMD2' RxTimingSetupReq MAC-CMD3' DevStatusReq Payload = [0x]03XXXXXXXX[0x]08XX[0x]06	

4	DUT sends Unconfirmed frame FCntUp = n + 3	→		Downlink ignored when MAC commands are present simulaneously in App-Payload and FOpts field

	The TCL sends Unconfirmed frames	←	CP-CMD RxAppCntReq
			FPort = 224
			Payload = [0x]09	

5	DUT sends Unconfirmed frame
	FCntUp = n + 4
		→	CP-CMD RxAppCntAns
			FPort = 224
			Payload = [0x]09XXXX
			RxAppCnt = x + 1
				Downlink counter not incremented with ignored downlinks

2.5.13. Incorrect MAC Commands

These tests ensure that a **DUT** behaves normally after it receives incorrect MAC commands from the **TCL.**

The invalid commands are:

- LinkADRReq with value out of spec - Payload: [0x]0380000000
- Incomplete MAC command - Payload: [0x]03010000
- Post Unknown MAC command ID - Payload: [0x]7F
- Valid MAC command followed by invalid MAC commands – Payload: [0x]0603010000

Verify:

- The **DUT** continues normal operation after receiving the invalid commands

2.5.13.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device - TCL	Frame	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			Payload = [0x]0380000000
			FPort = 0	

2	DUT sends Unconfirmed frame	→	LinkADRAns NOT = OK	DUT continues normal operation after receiving LinkADRReq with ‘out of spec’ Payload

	The TCL sends Unconfirmed frame	←	MAC-CMD LinkADRReq
			Payload = [0x]03010000
			FPort = 0	

3	DUT sends Unconfirmed frame	→	No response	DUT continues normal operation after incomplete MAC command Payload

	The TCL sends Unconfirmed frame	←	MAC-CMD
			Payload = [0x]7F
			FPort = 0	

4	DUT sends Unconfirmed frame	→	No response	DUT continues normal operation after unknown MAC command Payload

	The TCL sends Unconfirmed frame	←	MAC-CMD1 DevStatusReq
			MAC-CMD2 incomplete LinkADRReq
			Payload = [0x]0603010000
			FPort = 0	

5	DUT sends Unconfirmed frame	→	MAC-CMD1 DevStatusAns Payload = [0x]06XXXX	DUT continues normal operation after incomplete MAC command Payload
	The TCL sends Unconfirmed frame	←	MAC-CMD1 LinkADRReq MAC-CMD2 [0x]7F CMD3 DevStatusReq FPort = 0	
6	DUT sends Unconfirmed frame	→	MAC-CMD1 LinkADRAns Payload = [0x]03XXXXXXXX	DUT answers only the first MAC command and continues normal operation if the second MAC command is invalid
7	Wait for a Dynamic channel: maximum of 5 * (number of channels configured) Fixed channel: maximum of 2 * (number of channels configured) uplink packets to be sent.	→ R [5*NbChDC] or [2*NbChFC] or [AllCh used]	For DC plan devices: Only default channels must be used. The additional channel must not be added. For FC plan devices: The DUT must send an uplink on any of the channels enabled.	

2.5.14. Multiple MAC commands prioritization

This test verifies that when a combination of application payload and MAC answers, or new MAC commands are sent by the DUT, the priority for including information in the frame is as shown below.

Priority Level	Information type
Highest	MAC answers
	New MAC commands
Lowest	Application payload

Table 4: Transmit data insertion prioritization

Additionally, if the commands cannot fit in the same frame due to size restrictions, the message must be truncated.

Verification summary:

- Within a single frame, the DUT must send all higher-priority information before sending any lower- priority information.
- If the MAC command buffer is too large to fit in the frame, the DUT must truncate the buffer at the end of the last MAC command that is able to fit within the frame.
- The DUT must execute the full list of MAC commands even if the buffer containing the MAC answers is truncated

2.5.14.a. Test Procedure Frame Sequence Chart

Step	Procedure	Frame Sequence		Test Purpose
		End Device – TCL	**Frame**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	FPort = 224
			CP-CMD LinkCheckReq
			
			MAC-CMD DevStatusReq
			
			MAC-CMD LinkADRReq
			DataRate = Max125kHzDR
			
			Payload = [0x]20
			FOpts = [0x]06[0x]03XXXXXXXX	

2	DUT sends Unconfirmed frame	→	MAC-CMD DevStatusAns MAC-CMD LinkADRAns MAC-CMD LinkCheckReq Payload = [0x]06XXXX[0x]0307[0x]02	DUT prioritises MAC answers over new MAC commands and application payload
	The TCL sends Unconfirmed frame	←	MAC-CMD LinkCheckAns Payload = [0x]02XXXX	
3	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame
	
	Note: This step is required for only regions with Dwell Time limitation
		←	MAC-CMD TXParamSetupReq
			UplinkDwellTime = 0
			Payload = [0x]09XX
				Set TXParamSetupReq for regions with Dwell time limitation

4	DUT sends Unconfirmed frame	→	For regions with Dwell time limitation only
			
			MAC-CMD TXParamSetupAns
			Payload = [0x]09
			
			DataRate = Max125kHzDR	

	TCL also sends Unconfirmed frame	←	MAC-CMD1 DevStatusReq MAC-CMD2 RxParamSetupReq Repeat the MAC-CMD DevStatusReq until the MAC command uplink response buffer would be full for MinDR, refer [2] MAC-CMDX LinkADRReq (with DataRate = MinDR) *For example: For EU863-870, the Max payload size is 51 bytes for MinDR. Hence repeat the DevStatusReq command 15 times to ensure the MAC command response buffer is greater than 51 bytes.* Payload = [0x]06[0x]05XXXXXXXX[0x]06[Repeat as required][0x]03XXXXXXXX	
5	DUT sends Unconfirmed frame	→	DataRate = MinDR MAC-CMD1 DevStatusAns MAC-CMD2 RxParamSetupAns MAC-CMD3 DevStatusAns … MAC-CMDX DevStatusAns Payload = [0x]06XXXX[0x]0507[0x]06XXXX…[0x]06XXXX DUT truncates the MAC command when max payload size is exceeded. The sequence of the response must be exactly the same as described. The LinkADRAns is not sent in the response as it must be truncated due to payload size restrictions. However, the DR must be set to MinDR.	Successful completion of all MAC commands in the correct sequence. Message truncated.

	TCL also sends Unconfirmed frame	←	MAC-CMD1 DevStatusReq MAC-CMD2 RxParamSetupReq Repeat the MAC-CMD DevStatusReq until the MAC command uplink response buffer would be full for MinDR, refer [2] *(i.e. same number of DevStatusReq commands sent in the previous step)* MAC-CMDX LinkADRReq (with DataRate = Max125kHzDR) Payload = [0x]06[0x]05XXXXXXXX[0x]06[Repeat as required][0x]03XXXXXXXX	
6	DUT sends Unconfirmed frame	→	DataRate = Max125kHzDR MAC-CMD1 DevStatusAns MAC-CMD2 RxParamSetupAns MAC-CMD3 DevStatusAns … MAC-CMDX DevStatusAns MAC-CMDX+1 LinkADRAns Payload = [0x]06XXXX[0x]0507[0x]06XXXX…[0x]06XXXX[0x]0307 DUT must not truncate the frame.	Successful completion of all commands. Message is not truncated.

3. FPort 224 Deactivation

This test must be performed as the last step ever to be performed on the device. The test lab must ensure that all other tests are completed on the device before performing this test. When the FPort 224 is disabled, it cannot be re-enabled on the device again.

The **TCL** will send a downlink payload message of [0x]07E (*DutFPort224DisableReq*) over port 224, thus disabling FPort 224 for the DUT.

3.1. Test Procedure Message Sequence Chart

Step	Procedure	Message Sequence		Test Purpose
		End Device - TCL	**Message**	
1	DUT sends Unconfirmed frame	→		

	The TCL sends Unconfirmed frame	←	CP-CMD
			DutFPort224DisableReq
			FPort = 224
			Payload = [0x]07E
				Disable the FPort 224 for the DUT as the last step of the Test cycle

2	If the device is an OTAA device, DUT sends Join-Request frame Else, skip to the next step	→		
	If the device is an OTAA device, TCL sends Join-Accept response	←		Join accepted for OTAA device
3	DUT sends Unconfirmed or Confirmed frame	→	FPort = any allowed port except 224	

	The TCL sends Unconfirmed frame	←	CP-CMD
			TxPeriodicityChangeReq
			FPort = 224
			Periodicity = 5 sec
			Payload = [0x]0601
			
			If DUT sent Confirmed uplink frame, TCL must Acknowledge
				Try to set Uplink Periodicity

4	DUT sends Unconfirmed or Confirmed frame If this uplink is not received within 2 minutes, then the test can be ended, and the next steps can be skipped.	→		
	The TCL sends Unconfirmed frame	←	CP-CMD EchoPayloadReq FPort 224 Payload = [0x]08010203 If DUT sent Confirmed uplink frame, TCL must Acknowledge	
5	Wait for a maximum of 3 minutes for the DUT to send an Unconfirmed or Confirmed frame If no uplink is sent, the test can be ended, and the next steps can be skipped.	→	No EchoPayloadAns response received	FPort 224 downlinks are not accepted.
	The TCL sends Unconfirmed frame	←	CP-CMD EchoPayloadReq FPort 224 Payload = [0x]08010203 If DUT sent Confirmed uplink frame, TCL must Acknowledge	Repeat the EchoPayloadReq

6	Wait for a maximum of 3 minutes.	→	No EchoPayloadAns response received	FPort 224 downlinks are not accepted.

4. Test Case Mapping with LoRaWAN Specification [1]

The following table provides the section mapping between the LoRaWAN Specification [1] and this Certification Specification document

LoRaWAN Spec [1] **section**	**This document section**	**Description**
3.3	2.5.11	Receive Windows
4	0 2.4.1 and 2.4.2	MAC Packet Formats
5.1	2.5.7	LinkCheck
5.2	2.5.8	LinkADR
5.3	0	DutyCycle
5.4	2.5.4	RXParamSetup
5.5	2.5.1	DevStatus
5.6	2.5.2 and 2.5.3	NewChannel and DlChannel
5.7	2.5.5	RXTimingSetup
5.8	2.5.6	TXParamSetup
5.9	2.5.10	DeviceTime
6.2	2.2	Over the air Activation
6.3	2.3	Activation by Personalization

Table 5 LoRaWAN Spec vs Certification Spec mapping

5. Notice of Use and Disclosure

LoRa Alliance, Inc.
5177 Brandin Court,
Fremont, CA 94538 USA