Specification

Z-Wave Long Range MAC Layer Test Specification

Version:	1.0
Description:	Test specification for testing the MAC layer of the Z-Wave Long Range protocol
Written By:	CSWG
Date:	2021.08.27
Reviewed By:	CSWG
Restrictions:	Public

Approved by:
Z-Wave Alliance Board of Directors

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
0.1	2020.09.29	CSWG	Initial version	
0.5	2021.02.12	CSWG	All	Complete set of test cases added to document
0.8	2021.03.01	CSWG	None	Updated to revision 0.8 after WG review. Ready for TC review
0.9	2021.03.29	CSWG	Frontpage	Cleanup for IPR review
1.0	2021.08.27	ZWA Board		Approved for Publication

Table of Contents

1 ABBREVIATIONS

Abbreviation	Explanation
ACK	Acknowledgement
AL	Always Listening
API	Application Program Interface
ERTT	Encoder Receiver Transmitter Test
FCS	Frame Check Sequence
FL	Frequently Listening
LR	Long Range
MAC	Media Access Control
MFR	MAC Footer
MHR	MAC Header
MPDU	MAC Protocol Data Unit
MSDU	MAC Service Data Unit
NL	Not Listening
NOP	No Operation
PHY	Physical (Layer)
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
S2	Security 2 (Command Class)
TX	Transmission
US_LR	United States Long Range

2 INTRODUCTION

2.1 Purpose

The purpose of this specification is to provide a set of tests that verifies compliance with the MAC layer of the Z-Wave Long Range protocol.

2.2 Audience and prerequisites

Developers and testers of the Z-Wave Long Range protocol.
An RF Sniffer hardware and analyzer software that can be tuned in on the valid Long Range frequency for Z-Wave or a Sniffer module and PC Application. Z-Wave Controller or equivalent to execute communication between the different nodes.

3 MAC-LAYER TEST CASE DESCRIPTIONS

3.1 General assumptions

For performing this test it is assumed that the MAC layer is running on a PHY layer that is compliant with the "Z-Wave Long Range PHY Layer specification"**Error! Reference source not found.** and is verified by the set of tests in the "Z-Wave Long Range PHY Layer Test Specification."

All components are defined in "Z-Wave Long Range PHY and MAC Specification" and that document is the sole reference for the present Test Plan.

All **times** and **time-out periods** must be compliant with the values described in tables 6-32 & 6-33 from "Z-Wave Long Range PHY and MAC Specification." **Error! Reference source not found.**

Inclusion in Z-Wave Long Range refers to Bootstrapping in "Z-Wave Long Range PHY and MAC Specification" - 6.1.2, each device has the same HomeID and different NodeID. The controller needs to have each End Node registered in its Node List with a unique NodeID. It is performed by Smart Start inclusion for Long Range Devices.

From here on all requirement numbers refer to sections in "Z-Wave Long Range PHY and MAC Specification" **Error! Reference source not found.**it is also referred to as [LRMAC].

3.2 Format of MPDU, Singlecast in Long Range

A device must be able to produce the 2 types of frames: Single Cast & Acknowledge in Long Range [LRMAC] 6.1.3.1.

3.2.1 Prerequisites

1 x Sniffer.
1 x Controller
1 x LR End node

3.2.2 Test Setup

1. Include End Node to the Controller network.
2. Controller sends Singlecast frame to End Node with Data Payload (MSDU) = 0x00 (NOP).

3.2.3 Test Result

2. Verify on Sniffer that End Node responds with an Acknowledgement frame to the Controller (Header type: 0x03).

3.2.4 Pass Criteria

If the frames are displayed in the Sniffer, that means the PHY-layer header and EOF Delimiter are structured correctly (6.3.2).

1. The singlecast frame sent to the End Node has the format from [LRMAC] figure 6-4 (6.3.2).
2. The singlecast frame sent to the End Node has the frame type set to: 0x01 ([LRMAC]6.2.2.3.1 – Table 6-5).
3. The singlecast frame sent to the End Node has the ACK bit set to 0x01 ([LRMAC] 6.3.1.5)
4. The End Node responds with an Acknowledgement frame ([LRMAC] 6.1.3.2.2).
5. This Acknowledgement frame matches the description ([LRMAC] 6.1.3.1.2).
6. This Acknowledgement singlecast responded has the frame type set to: 0x03 ([LRMAC] 6.2.2.3.1 – Table 6-5).
7. The Ack Req bit (byte 8, bit 7) in the Acknowledgement frame is set to 0 ([LRMAC] 6.3.1.5.1 – Table 6-19).
8. This singlecast acknowledgement responded has the same HomeID as the sent singlecast ([LRMAC] 6.1.2).
9. This singlecast acknowledgement responded has the Destination ID set to the Node ID of the Controller ([LRMAC] 6.1.2).

3.2.5 Fail Criteria

1. The single cast does Not have the format described by figure 6-4 ([LRMAC] 6.3.2).
2. The singlecast frame sent to the End Node doesn't have the frame type set to: 0x01 ([LRMAC] 6.2.2.3.1 – Table 6-5).
3. The singlecast frame sent to the End Node doesn't have the ACK bit set to 0x01 ([LRMAC] 6.3.1.5).
4. The End Node did not respond using an Acknowledgement frame ([LRMAC] 6.1.3.2.2).
5. The Acknowledgement singlecast frame does not match the description ([LRMAC] 6.1.3.1.2).
6. This Acknowledgement singlecast responded does not have the frame type set to: 0x03 ([LRMAC] 6.2.2.3.1 – Table 6-5).
7. The ACK bit (byte 8, bit 7) in the Acknowledgement frame is NOT set to 0 ([LRMAC] 6.3.1.5.1 – Table 6-19).
8. This singlecast acknowledgement responded has a Different HomeID than the singlecast ([LRMAC] 6.1.2).
9. This singlecast acknowledgement responded has a different destination ID than the node ID of the Controller ([LRMAC] 6.1.2).

3.3 Network Robustness, Clear channel assessment in Long Range

A device must ensure robustness in data transmission. This is achieved by the mechanisms: Backoff Algorithm, Frame Acknowledgement, Data Verification and Frame Retransmission ([LRMAC] 6.1.3.2).

3.3.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node
1 x Noise Generator

1 x Spectrum Analyzer or equivalent for monitoring RF traffic in a specified bandwidth

3.3.2 Test Setup

1. Include End Node to the Controller network.
2. Configure Noise Generator to operate in the same frequency (US_LR) as the Z-Wave devices.
3. Use the Spectrum Analyzer or equivalent device to monitor the RF traffic around the channels of the used frequency.
4. On the Sniffer observe that communication between Controller and End Node is possible and which channels are used.
5. Start the Noise Generator and observe on the Spectrum Analyzer that there is higher volume of traffic on the chosen frequency: There is a spike on the plot of RF frequencies corresponding to the channel were the noise Generator is operating.
6. On the Controller Set up an ERTT test with 100 iterations, 100ms delay, alternating Basic Set Value 0/255 and disable “Stop on Error”. Observing how the transmissions start in one channel and come into a pause waiting for lower traffic from the noise Generator.

3.3.3 Test Result

6. Observe on the Sniffer how the singlecast are not sent immediately as the PHY layer is waiting for an opening in traffic and that the command frames are on the channel with less traffic (the one in which the Noise Generator is not transmitting).

3.3.4 Pass criteria

1. The singlecast frames don’t appear on the Sniffer right after being commanded, instead the node waits for an idle channel before transmitting ([LRMAC] 6.1.3.2.1)
2. The singlecast frames wait a bare minimum of 1100ms before being sent ([LRMAC] 6.4.1 – Table 6-32)
3. The singlecast uses the channel with higher communication rate ([LRMAC] 6.1.3.2.1)
4. The singlecast is responded with an acknowledgement frame from the End Node.
5. The singlecast is responded with an acknowledgement frame from the End Node in the same channel.
6. The MAC Layer chooses the active channel used for transmission ([LRMAC] 6.1.3.2.5).

3.3.5 Fail criteria

1. The node does not wait for an idle channel before transmitting ([LRMAC] 6.1.3.2.1).
2. The singlecast fails being transmitted by trying to use a high traffic channel ([LRMAC] 6.1.3.2.1).
3. The singlecast is not sent in the channel with higher communication rate ([LRMAC] 6.1.3.2.1)
4. The singlecast is not responded with an acknowledgement frame at all.
5. The singlecast is not responded with an acknowledgement frame from the End Node in the same channel.
6. The active channel is selected by a different Layer ([LRMAC] 6.1.3.2.5).

3.4 Network Robustness, Acknowledgement in Long Range

A device must ensure robustness in data transmission. This is achieved by the mechanisms: Backoff Algorithm, Frame Acknowledgement, Data Verification and Frame Retransmission ([LRMAC] 6.1.3.2).

3.4.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR Not-Listening (NL) End node

3.4.2 Test Setup

1. Include End Node to the Controller network.
2. On the Sniffer observe that communication between Controllers and End Node is possible.
3. Set on Controller a queue of 30 Version Get commands to send to the NL End Node. On the Command Classes view, enable “Expect Command” select Version Report for each command queued. Wait or wake the End Node up.
4. Once the commands start being sent after a couple of seconds, remove power, or do a power reset on the End Node, in order to prevent it from responding to the commands.
5. After a few seconds restore power to the End Node and let the Controller finish sending its command queue.

3.4.3 Test Result

2. The End Node answers only when it has awakened, and Controller waits until the End Node wakes up to transmit frames to it.
3. Observe on the Sniffer how the End Node replies to the controller’s commands with one acknowledgement frame using the same sequence number present in each singlecast.
4. When the End Node does not answer, the controller retransmits the frame with the same sequence number. It then waits for the End Node to continue responding.
5. When the power returns to the End Node, the Controller waits for the End Node to answer with an acknowledgement frame and followed by responding to the query with the Version Report command as described.

3.4.4 Pass criteria

1. The singlecast frames are answered with an acknowledgement frame ([LRMAC] 6.1.3.2.2).
2. This Acknowledgement frame matches the description ([LRMAC] 6.3.3).
3. This Acknowledgement singlecast responded has the frame type set to: 0x03 ([LRMAC] 6.2.2.3.1– Table 6-5).
4. The ACK bit (byte 8, bit 7) in the Acknowledgement frame is set to 0 ([LRMAC] 6.3.1.5.1– Table 6-19).
5. This singlecast acknowledgement responded has the same HomeID as the sent singlecast ([LRMAC] 6.1.2).
6. This singlecast acknowledgement responded has the destination ID set to the node ID of the Controller that sent it ([LRMAC] 6.1.2).

7. The controller retransmits the frames that weren't answered with an Acknowledgement ([LRMAC] 6.1.3.2.3)
8. When the Controller retransmit the frames, the End Node answers with an acknowledgement using the same Sequence Number as the received frame ([LRMAC] 6.3.1.6).

3.4.5 Fail criteria

1. The End Node did not respond using an Acknowledgement frame ([LRMAC] 6.1.3.2.2).
2. The Acknowledgement singlecast frame does not match the description ([LRMAC] 6.1.3.3).
3. This Acknowledgement singlecast responded does not have the frame type set to: 0x03 ([LRMAC] 6.2.2.1.1– Table 6-3).
4. The Ack bit (byte 8, bit 7) in the Acknowledgement frame is NOT set to 0 ([LRMAC] 6.3.1.5.1– Table 6-19).
5. This singlecast acknowledgement responded has a Different HomeID than the singlecast ([LRMAC] 6.1.2).
6. This singlecast acknowledgement responded has a different destination ID than the node ID of the Controller ([LRMAC] 6.1.2).
7. The controller does not retransmit the frames that weren't answered with an Acknowledgement ([LRMAC] 6.1.3.2.3)
8. When the Controllers retransmit the frames, the End Node does not answer or answers with an Acknowledgement using a different Sequence Number as the received frame ([LRMAC] 6.3.1.6).

3.5 Network Robustness, Acknowledgement OFF in Long Range

A device must ensure robustness in data transmission. This is achieved by the mechanisms: Backoff Algorithm, Frame Acknowledgement, Data Verification and Frame Retransmission ([LRMAC] 6.1.3.2).

3.5.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node
1 x Frame Generator*

3.5.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections of the frame modified individually in order to test the behavior of the receiver.

1. Include End Node to the Controller network.
2. On the Sniffer observe that communication between both Controllers and End Node is possible.
3. Generate a frame that sends to the End Node a singlecast with MDSU = 0x00 (NOP) making sure ACK bit (byte 8, bit 7) is set to 0x00.
4. Send this frame as singlecast to the End Node.

3.5.3 Test Result

4. Observe on the Sniffer how the End Node ignores the singlecast and the Controller re-transmits the frame.

3.5.4 Pass criteria

1. The ACK bit (byte 8, bit 7) in the singlecast frame is set to 0 ([LRMAC] 6.3.1.5.1 – Table 6.19).
2. The singlecast frames are not answered with an acknowledgement frame ([LRMAC] 6.1.3.2.2).

3.5.5 Fail criteria

1. The End Node did respond using an Acknowledgement frame ([LRMAC] 6.1.3.2.2).

3.6 Network Robustness, Retransmission

A device must ensure robustness in data transmission. This is achieved by the mechanisms: Backoff Algorithm, Frame Acknowledgement, Data Verification and Frame Retransmission ([LRMAC] 6.1.3.2).

3.6.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node

3.6.2 Test Setup

1. Include End Node to the Controller network.
2. Disable the End Node device by removing power or removing the antenna from it.
3. Send a singlecast from the controller to the End Node. With MSDU = 0x00 (NOP).
4. On the Sniffer observe that communication between both Devices is not possible and the Controller sending the frame re-tries sending it.

3.6.3 Test Result

4. Observe on the Sniffer the transmission is attempted up to 2 times more (the maximum number of frame transmission retries “aMacLRMaxFrameRetries”) before increasing the Sequence Number and each re-transmission waits a random period to prevent collisions with other frames that may be being sent at the same time.

3.6.4 Pass criteria

1. The Controller sends only 2 retransmissions (“aMacLRMaxFrameRetries”) with the same Sequence Number waiting a random period of time after each attempt ([LRMAC] 6.1.3.2.3).
2. The Controller issues a new frame with the same contents but with its Sequence Number value increased by one and sent also only up to the value of “aMacLRMaxFrameRetries” waiting a random period of time after each attempt ([LRMAC] 6.5.1.5.5).

3.6.5 Fail criteria

1. When the Controllers retransmit the frames, the sequence number changes each time and does it a different amount of times than the one defined by “aMacLRMaxFrameRetries” ([LRMAC] 6.1.3.2.3).
2. The Controller does not issue any new frame, or it issues them with a Sequence Number entirely unrelated to the previously used one ([LRMAC] 6.5.1.5.5).

3.7 Network Robustness, Data Validation Corrupt FCS, Long Range

A device must ensure robustness in data transmission. This is achieved by the mechanisms: Backoff Algorithm, Frame Acknowledgement, Data Verification and Frame Retransmission ([LRMAC] 6.1.3.2).

3.7.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node
1 x Frame Generator*

3.7.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include End Node to the Controller network.
2. On the Sniffer observe that communication between both Controllers and End Node is possible.
3. Generate a frame that sends to the End Node a singlecast with MDSU = 0x00 (NOP) making sure the 8 bits for FCS are random and not generated automatically.
4. Send this frame as singlecast to the End Node.

3.7.3 Test Result

4. Observe on the Sniffer how the End Node ignores the singlecast and the Controller re-transmits the frame.

3.7.4 Pass criteria

1. The singlecast frame is not answered with an acknowledgement frame ([LRMAC] 6.1.3.2.4).
2. The controller retransmits the frames that weren’t answered with an Acknowledgement ([LRMAC] 6.1.3.2.4)
3. When the Controller retransmit the frames, the End Node remains without answering with an acknowledgement ([LRMAC] 6.1.3.2.2).

3.7.5 Fail criteria

1. The End Node does respond using an Acknowledgement frame ([LRMAC] 6.1.3.2.4).

2. The End Node answers the singlecasts with Acknowledgement frame even if it has bit errors as per FCS manipulation ([LRMAC] 6.1.3.2.4).
3. The controller does not retransmit the frames that weren't answered with an Acknowledgement ([LRMAC] 6.1.3.2.2)

3.8 Power Consumption considerations, Sleeping nodes

Power consumption considerations are handled by the PHY layer. ([LRMAC] 6.1.3.3)

Beaming frames are covered in the MPDU Format for section ([LRMAC] 6.3.6)

3.9 General MPDU Format, Long Range

The MAC Protocol Data Unit (MPDU), consists of three basic components: A MAC Header (MHR), a MAC data payload (MAC Service Data Unit (MSDU)) and a MAC Footer (MFR) ([LRMAC] 6.3.1).

3.9.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node

3.9.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver

1. Include the End Node to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to one End Node.
3. Observe the structure of the singlecast sent.

3.9.3 Test Result

3. The singlecast is displalyed correctly on the Sniffer.

3.9.4 Pass Criteria

1. The singlecast shows:
    a. MHR: ([LRMAC] 6.1.3 – Figure 6-4)
        i. Home ID: 4 bytes ([LRMAC] 6.3.1.1)
        ii. Source Node ID: 12 bits ([LRMAC] 6.3.1.2)
        iii. Destination Node ID: 12 bits ([LRMAC] 6.3.1.3)
        iv. Length: 1 byte ([LRMAC] 6.3.1.4)
        v. Frame Control (8 bits): ([LRMAC] 6.3.1.5 – Table 6-19)
            1. Ack Req: 1 bit ([LRMAC] 6.3.1.5.1)
            2. Extended: 1 bit ([LRMAC] 6.3.1.5.2)
            3. Header type: 3 bits ([LRMAC] 6.3.1.5.3)

4. Reserved: 3 bits ([LRMAC] 6.3.1.5.4)

vi. Sequence Number: 8 bits ([LRMAC] 6.3.1.6)

vii. Noise Floor: 8 bits ([LRMAC] 6.3.1.7)

viii. Tx Power: 8 bits ([LRMAC] 6.3.1.8)

b. MSDU: Payload: ([LRMAC] 6.3.1.9)

i. 1 Byte = 0x00 (NOP)

c. MFR: FCS (not described in the structure in the Sniffer): ([LRMAC] 6.3.1.10)

i. FCS: 2 bytes

3.9.5 Fail Criteria

1. At Least one of the components of the format of the MPDU for singlecast has a different length or values ([LRMAC] 6.3.1).

3.10 MPDU Format, Home ID

The MAC Protocol Data Unit (MPDU), consists of three basic components: A MAC Header (MHR), a MAC data payload (MAC Service Data Unit (MSDU)) and a MAC Footer (MFR). Home ID are 4 bytes that identify all nodes in the same domain ([LRMAC] 6.3.1.1).

3.10.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.10.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include both End Node devices to the Network of the Controller.
2. Verify the Controller can communicate with both End Nodes by sending Singlecast to each of them and S2 Multicast to both.
3. Send a singlecast to one of the End Nodes, modifying the Home ID to be different from the original value.
4. Send an S2 multicast to both End Nodes with modified Home ID.
5. Send a singlecast to one of the End Nodes extending the length of the Home ID value*.
6. Send a singlecast to one of the End Nodes shortening the length of the Home ID value*.
*only possible if the Frame generator allows to modify the length of the different fields.

3.10.3 Test Result

2. Both End Nodes answer with an Acknowledgement frame as expected to the singlecast and the singlecast follow-up after the Multicast.
3. The End Node does not answer, since the Home ID is not the Home ID it has been included to.

4. Neither End Node answers since the Home ID is different from the one, they have been included to.
5. After the Home Id component, the rest of the frame will not match against the expected values, therefore the frame should be rejected.
6. After the Home Id component, the rest of the frame will not match against the expected values, therefore the frame should be rejected.

3.10.4 Pass Criteria

1. On the singlecast the Home ID occupies only 4 bytes ([LRMAC] 6.3.1.1)
2. No node responds to any frame that holds a modified Home Id in any way, because of mis-matching Home ID value or because the frame is affected because of re-sizing the HomeID ([LRMAC] 6.3.1.1)

3.10.5 Fail Criteria

1. Any of the methods for altering the Home ID Component is accepted by the receiving node and answered with an acknowledgement frame.

3.11 MPDU Format, Source NodeID

The MAC Protocol Data Unit (MPDU), consists of three basic components: A MAC Header (MHR), a MAC data payload (MAC Service Data Unit (MSDU)) and a MAC Footer (MFR). Source Node ID are 12 bits that identify the node within one domain that have transmitted the frame ([LRMAC] 6.3.1.2).

3.11.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End nodes

3.11.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver

1. Include End Nodes to Controller's network.
2. Verify the Controller can communicate with both End Nodes by sending Singlecast to each of them and S2 Multicast to both.
3. Send a singlecast to one of the End Nodes, modifying the Source Node ID to be different from the original value.
4. Send an S2 Multicast to both End Nodes, modifying the Source Node Id to be different from the original value.
5. Send a singlecast to one of the End Nodes, modifying the Source Node ID to be 0x000.
6. Send a singlecast to one of the End Nodes, modifying the Source Node ID to be a value between 0xFA6 & 0xFFF (reserved values, Table 6-15).
7. Send a singlecast to one of the End Nodes, modifying the Source node ID to be more than 12 bits long *
   *only possible if the Frame generator allows to modify the length of the different fields.

3.11.3 Test Result

3. End Node received the singlecast and responds to the modified Source Node ID with an Acknowledgement frame.
4. The End Nodes do not answer the S2 multicast frame but answer to the singlecast follow-up frames originated after the S2 Multicast
5. The End Node receiving the singlecast with Source Node ID set to 0x000, will answer to it with an Acknowledgement frame.
6. The End Node receiving the singlecast with Source Node ID set to a reserved value will not answer, since a network is limited to that number of nodes.
7. As the structure of the frame will be affected by an oversized Source Node ID component, it won't be received correctly by the End Node and won't answer with an Acknowledgement frame. The Controller should try to retransmit the modified frame.

3.11.4 Pass Criteria

1. When the Controller does not receive the Acknowledgement frame, the Controller re-tries sending the same frame up to 3 times and then routing through the other End Node node because Each time the receiving End Node node answers to the modified Source Node ID. This happens for single cast, singlecast follow-up, frames addressed to Node 0 or frames with a structure affected by a longer Source Node ID field. ([LRMAC] 6.3.2.1, 6.1.3.2.3)
2. The End Nodes won't answer to a frame with Source Node Id set to a reserved value (outside the valid values defined by Table 6-15), causing the controller to retransmit the frame. ([LRMAC] 6.3.2.1)

3.11.5 Fail Criteria

1. The End Node answers to the controller with an Acknowledgement frame directly, ignoring the field Source Node ID ([LRMAC] 6.3.2.1)
2. The Controller does not re-transmit when the Acknowledgement frames are not addressed to it. ([LRMAC] 6.1.3.2.3)

3.12 MPDU Format, Destination Node ID

The destination Node ID specifies a destination node in the same domain identified by the HomeID. It shall comply with table 6-17 ([LRMAC] 6.3.1.3).

3.12.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.12.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include End Nodes to Controler's network.
2. Send a singlecast with MPDU = 0x00 (NOP) to each End Node.
3. Look for the Destination NodeID field in the frames on the Sniffer.
4. Generate a singlecast frame with Destination ID value different to either End Node, send it.
5. Generate a singlecast frame with Destination ID value higher than 0xFA6 (4006), send it.
6. Generate a singlecast frame with Destination ID field longer than 1 2bits, send it.
7. Generate a singlecast frame with Destination ID value of 0xFFF (4095), send it.

3.12.3 Test Result

2. Communication is correct. End Nodes answer with an Acknowledgement frame.
3. Destination NodeID is byte 12 in the frames. It holds the value of the End Node's NodeID.
4. The Controller tries to reach this End Node but can't reach it.
5. The Controller tries to reach this End Node but can't reach it.

6. The Controller tries to reach this End Node but can't reach it.
7. The Controller sends this frame as a Broadcast.
    a. The End Nodes don't respond to this Broadcast frame.

3.12.4 Pass Criteria

1. The Destination Node ID is 12 bits in length. ([LRMAC] 6.3.1.3)
2. The Destination Node ID can be any value up to 0xFA6 (4006) ([LRMAC] 6.3.1.3)
3. The Destination Node ID is in Bytes 5 & 6 of the frame ([LRMAC] 6.3.1.3 – Table 6-16)

3.12.5 Fail Criteria

1. The Destination Node ID can be more than 12 bits in length ([LRMAC] 6.3.1.3).
2. The Destination Node ID can be any value beyond 0xFA6 (4006) ([LRMAC] 6.3.1.3).
3. Always Listening devices respond or try to route singlecast addressed to 0xFFF (Broadcast) ([LRMAC] 6.3.1.3).

3.13 MPDU Format, Length

The length field is 1 byte that indicates the length of the MPDU in bytes. It's limited by "aMacLRMaxMSDUSize" defined on table 6-33. A receiving node shall not read more than the maximum length allowed ([LRMAC] 6.3.1.4).

3.13.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node

3.13.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include the End Node to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to the End Node.
3. Look for the Length field.
4. Populate the MPDU with a large amount of random data, less than "aMacLRMaxMSDUSize" and send it to the End Node.
5. Generate a singlecast with MPDU = 0x00 (NOP) and modify the Length field to be more than 16 and less than 59. Send it to the End Node.
6. Generate a singlecast with MPDU = 0x00 (NOP) and modify the Length field to be more than 59 Bytes. Send it to the End Node.
7. Generate a singlecast with MPDU = 0x00 (NOP) and modify the Length field to be more than 1 byte. Send it to the End Node.
8. Generate a singlecast populating the MPDU with a long amount of random data, less than "aMacLRMaxMSDUSize" and modify the value of the Length field to be 16. Send it to the End Node.

3.13.3 Test Result

2. Communication is possible and End Node answers with an Acknowledgement frame. Check on the Singlecast the Length field.
3. The Length field should be in Byte 7 of the frame and be show value 16 (0x10) for a NOP MPDU.
4. The singlecast should show the corresponding size in length.
5. When the End Node receives it, the stated size and the actual size as well as the FCS values do not correspond, and the End Node ignores the singlecast.
    a. The Controller tries re-transmitting the same singlecast because of not receiving an Acknowledgement frame.
6. When the End Node receives it, the stated size is larger than “aMacLRMaxMSDUSize” and the End Node ignores the frame.
    a. The Controller tries re-transmitting the same singlecast because of not receiving an Acknowledgement frame.
7. When the End Node receives it, the structure of the frame is outside specifications and the End Node ignores the frame.
    a. The Controller tries re-transmitting the same singlecast because of not receiving an Acknowledgement frame.
8. When the End Node receives it, the stated size of the frame is smaller than it actually is and the End Node ignores the frame.
    a. The Controller tries re-transmitting the same singlecast because of not receiving an Acknowledgement frame.

3.13.4 Pass Criteria

1. The Length field is only one byte in length. ([LRMAC] 6.3.1.4)
2. The Length field is in byte 7 of the Frame ([LRMAC] 6.3.1.4– Figure 6-18)
3. The value of the Length field is always less or equal than “aMacLRMaxMSDUSize”. ([LRMAC] 6.3.1.6)
4. The receiving node ignores all instances where the Length field does not match the actual length of the frame. ([LRMAC] 6.3.1.6)

3.13.5 Fail Criteria

1. The Length field is different from one byte in length. ([LRMAC] 6.3.1.6)
2. The Length field is located outside byte 7 of the Frame ([LRMAC] 6.3.1.6 – Figure 6-18)
3. The value of the length field can be more than “aMacLRMaxMSDUSize”. ([LRMAC] 6.3.1.6)
4. The receiving node accepts and answers with an Acknowledgement frame any frame regardless of the size and value of the Length field. ([LRMAC] 6.3.1.6)

3.14 MPDU Format, Frame Control, Header Type, Singlecast

The Frame Control field is 8 bits (1 byte) in length. It defines the frame type and other control flags. The header type defines the frame Header type. A broadcast MPDU is a singlecast MPDU (type 0x01) carrying destination Node ID = 0xFFF ([LRMAC] 6.3.1.5.3).

3.14.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node

3.14.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include the End Node to Controller's Network.
2. Send one single cast to the End Node with MPDU 0x00 (NOP).
3. Generate a frame with Header going from 0x02 to 0x0F according to table 6-20 and send it to the End Node.
4. Generate a frame with Header going from 0x02 to 0x0F according to table 6-20 and send it to Node ID 0xFFF (4095).

3.14.3 Test Results

2. Singlecast is sent correctly and it's answered with an Acknowledgement frame.
    a. Its Header type is set to 0x01.
3. Each frame sent to the node is displayed as the corresponding type on the Sniffer, it's ignored by the End Node and no Acknowledgement frame is responded.
4. Each frame sent to Node ID 4095 is displayed as the corresponding type on the Sniffer*

* needs practical verification.

3.14.4 Pass Criteria

1. Each frame sent by the Controller in 3. & 4. is displayed as the corresponding type on the Sniffer, making each frame correctly defined. ([LRMAC] 6.3.1.5.3)
2. None of the frames sent in 3. Are answered by definition.
3. None of the frames sent in 4. Are answered by definition nor by being addressed to a reserved Destination Node ID.

3.14.5 Fail Criteria

1. Any frame is displayed as singlecast on the Sniffer regardless of the different Header. ([LRMAC] 6.3.1.5.3)
2. Any frame sent in 3. received an Acknowledgement frame.
3. Any frame sent in 4. received an Acknowledgement frame.

3.15 MPDU Format, Frame Control, Header Type, Acknowledgement

The Frame Control field is 8 bits (1 byte) in length. It defines the frame type and other control flags. The header type defines the frame Header type. A broadcast MPDU is a singlecast MPDU (type 0x01) carrying destination Node ID = 0xFF ([LRMAC] 6.3.1.5.3).

3.15.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR Z-Wave Serial API End Node that can manipulate frames

3.15.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include the Serial API End Node to Primary Controller's network.
2. Send one singlecast to the End Node with MPDU 0x00 (NOP).
3. Generate an Acknowledgement frame with Header going from 0x01 to 0x0F (except 0x03) according to table 6-20 to be answered by the End Node to the Primary Controller and send a singlecast to the End Node for each header type.
4. Generate an Acknowledgement frame with Header going from 0x01 to 0x0F (except 0x03) according to table 6-20 to be answered by the End Node to Node ID 0xff (255) and send a singlecast to the End Node for each header type.

3.15.3 Test Result

2. The singlecast is sent correctly and it's answered with an Acknowledgement frame from the End Node.
3. Each frame answered is constructed as an Acknowledgement frame, but with the corresponding header from table 6-20.
    a. Therefore, it's displayed on the Sniffer as the expected type.
    b. Since the frames are not identified as Acknowledgement Header Type, the Primary Controller tries to retransmit each time. *
4. Each frame answered is constructed as an Acknowledgement frame, but with the corresponding header from table 6-20 and addressed to Node ID 0xFF.
    a. Therefore, it's displayed on the Sniffer as the expected type except for type 0x01, showing as a Broadcast.
    b. Since the frames are not identified as Acknowledgement Header Type, the Primary Controller tries to retransmit each time. *

* needs practical verification.

3.15.4 Pass Criteria

1. Each frame sent by the Controller in 3. is answered with the corresponding type on the Sniffer, making each frame correctly defined. ([LRMAC] 6.3.1.5.3, 6.3.3.2.2)
2. None of the frames sent in 3. Are answered by the Primary Controller by definition.
3. None of the frames sent in 4. Are answered by the Primary Controller by definition.

3.15.5 Fail Criteria

1. Any frame is displayed as Acknowledgement on the Sniffer regardless of the different Header. ([LRMAC] 6.3.1.5.3, 6.3.3.2.2)
2. Any frame sent in 3. received an Acknowledgement frame.

3. Any frame sent in 4. received an Acknowledgement frame.

3.16 MPDU Format, Sequence Number

The sequence number is an 8-bit field provided by higher layers when transmitting. The same Sequence Number shall be used for all retransmissions of a given MPDU that first fails being delivered. A receiving node shall return the same value in an Acknowledgement frame if the Ack bit is present in the received frame ([LRMAC] 6.3.1.6).

3.16.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node
1 x LR Z-Wave Serial API End Node that can manipulate frames

3.16.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include both End Nodes and Serial API End Node to the Primary Controller's network.
2. Disable one of the End Nodes' antenna and send a singlecast with MPDU = 0x00 (NOP) to it.
3. Enable the End Node again and try sending a singlecast to it again.
4. Select both End nodes and send an S2 multicast from the controller.
5. Configure an Acknowledgement frame so that its Sequence Number has a random non-zero static value on the Serial API End Node when the Primary Controller transmits a singlecast to it and proceed to send a singlecast to the Serial API End Node.
6. Configure a singlecast frame with static sequence number and send it twice or more to one of the End Nodes from the Primary Controller.
7. Configure a singlecast frame with static sequence number set to 0 and send it twice or more to one of the End Nodes from the Primary Controller.
8. Configure a singlecast frame with a longer Sequence Number value than 8 bits and send it to one of the End Nodes.

3.16.3 Test Result

2. Observe that the Controller re-tries sending the command to the disabled End Node and all frames have the same sequence number.
3. The Controller transmits directly and the frame correctly reaches the destination Node.
    a. The End Node Answers with an Acknowledgement frame using the same sequence number as the singlecast.
4. The multicast frame and its respective single cast follow-up frames have their own sequence numbers
5. When receiving the Singlecast, the Serial API End Node responds with the generated Acknowledgement frame with the static Sequence Number value.

   a. When the Controller received this Acknowledgement frame with a Sequence Number value different from the singlecast it transmitted, it tries to retransmit the frame again. Since this is equivalent to not having received the proper Ackowledgement frame*
6. The End Node receiving the Singlecast answers correctly to the first one with an Acknowledgement frame using the same Sequence Number.
   a. In the following frames, it ignores the frame, as it holds the same Sequence Number value as previous frames.
   b. Since the frames go unanswered, the Primary Controller tries re-transmitting the frames again.
7. The End Node receiving the Singlecast with Sequence Number set to 0 answers to it.
   a. The Primary Controller tries re-transmitting the following attempted frames with the same Sequence Number set to 0.
   b. The receiver it ignores those frames.
8. Since the sequence number has a larger size than it's supposed to have by definition, the Receiving node finds a miscalculation in the Frame Check Sequence (FCS).
   *Needs practical verification.

3.16.4 Pass Criteria

1. The re-transitted frames when the Controller doesn't reach the End Node have the same Sequence Number. ([LRMAC] 6.3.1.6)
2. The Ack frames have the same Sequence number as the original singlecast frame sent from the Controller. ([LRMAC] 6.3.1.6, 6.3.3.3)
3. The S2 Multicast and its follow-up singlecast have successive Sequence numbers. ([LRMAC] 6.3.1.6)
4. Acknowledgement frames with value that do not match the one of the singlecast that originated them are rejected by the Controller and re-transmitted by the receiving node. ([LRMAC] 6.3.1.6)
5. Sequence Number only has 8 bits going from 0x0 to 0xFF. ([LRMAC] 6.3.1.6)

3.16.5 Fail Criteria

1. The retransmitted frames have their own Sequence Number value. ([LRMAC] 6.3.1.6)
2. The Ack frames from the Controller to the repeater have different Sequence Number value. ([LRMAC] 6.3.1.6)
3. The S2 Multicast and its successive Follow-up singelcast frames have the same Sequence Number value. ([LRMAC] 6.3.1.6)
4. All Acknowledgement frames with non-zero value different from the singlecast that originated them are accepted. ([LRMAC] 6.3.1.6)
5. Sequence Number can have more than 8 bits of length. ([LRMAC] 6.3.1.6)

3.17 MPDU Format, Noise Floor

The Noise Floor field is an 8-bit signed field that indicates the radio noise level present on the channel the frame is being transmitted. Its format follows Table 6-22 ([LRMAC] 6.3.1.7).

3.17.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.17.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include End Nodes to Controller's Network.
2. Send a regular singlecast to one End Node with MDPU = 0x00 (NOP).
3. Check the Noise Floor field in the Header section in the Sniffer for both the Singlecast and the Acknowledgement frame.
4. Generate a Frame that has a custom random value set on Noise Floor field, send it to one End Node.
5. Send an S2 Multicast to both End Nodes. Observe the Noise Floor field.

3.17.3 Test Results

2. End Node answers with an Acknowledgement Frame to the Controller.
3. The Acknowledgement Frame has similar values as the Singlecast.
4. The End Node answers to the Controller with an Acknowledgement frame with a similar Noise Floor value.
5. The Multicast is sent with the same value as the Singlecast follow-ups.

3.17.4 Pass Criteria

1. Noise Floor consists of 8 bits at byte 10 of the Frame and only shows valid values ([LRMAC] 6.3.1.7, Table 6-22, Table 6-23).
2. The singlecast and Singlecast follow-up are sent with similar values, within the valid ones. ([LRMAC] 6.3.1.7 Table 6-22)
3. The End Node nodes answer with an Acknowledgement frame in similar value to the Noise Floor as the singlecast.
4. The modified singlecast is responded with an Acknowledgement frame set in the same Noise Floor.
5. The Multicast is not answered by any of the End Nodes ([LRMAC] 6.3.3).

3.17.5 Fail Criteria

1. Noise Floor is not 8 bits at byte 10 of the frame and holds different values from the valid ones ([LRMAC] 6.3.1.7, Table 6-22, Table 6-23).
2. The default singlecast are sent with different values and outside the valid ranges ([LRMAC] 6.3.1.7 Table 6-23).
3. The End Nodes answer to a regular singlecast with an Acknowledgement frame set to a different Noise Floor value than the singlecast.
4. The End Nodes answer to the modified singlecast with an Acknowledgement frame set to a different Noise Floor value.

5. The End Nodes answer to the Multicast frame directly with an Acknowledgement frame ([LRMAC] 6.3.3).

3.18 MPDU Format, Tx Power

The Tx Power field is an 8 bit signed field specifying the transmit power used to transmit this frame defined by the table 6-25 (6.3.1.8).

3.18.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node
1 x Spectrum Analyzer

3.18.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Configure Spectrum Analyzer to detect traffic in the RF determined for LR.
2. Include End Nodes to Controller's Network.
3. Send a regular singlecast to one End Node with MDPU = 0x00 (NOP).
4. Check the Tx Power field in the Header section in the Sniffer for both the Singlecast and the Acknowledgement frame.
5. Generate a Frame that has a random value set on Tx Power field, send it to one End Node.
6. Send an S2 Multicast to both End Nodes. Observe the Tx Power field.

3.18.3 Test Results

3. End Node answers with an Acknowledgement Frame to the Controller.
4. The Acknowledgement Frame has the same Tx Power value as the Singlecast.
5. The End Node answers to the Controller with an Acknowledgement frame in the same TX Power.
    a. Observe in the Spectrum Analyzer that the Acknowledgement frame was detected in the corresponding Power Level.
6. The Multicast is sent in higher Power while the Singlecast Follow-up frames are sent in the default value.

3.18.4 Pass Criteria

1. The singlecast and Singlecast follow-up are sent with similar values, within the valid ones. ([LRMAC] 6.3.1.8 Table 6-25)
2. The End Node nodes answer with an Acknowledgement frame in the same Tx Power as the singlecast.
3. The modified singlecast is responded with an Acknowledgement frame set in the same Tx Power value.
4. The Multicast is not answered by any of the End Nodes ([LRMAC] 6.3.3).

3.18.5 Fail Criteria

1. The default singlecast are sent with different values and outside the valid ranges ([LRMAC]6.3.1.8 Table 6-25).
2. The End Nodes answer to a regular singlecast with an Acknowledgement frame set to a different Tx Power value.
3. The End Nodes answer to the modified singlecast with an Acknowledgement frame set to a different Tx Power value.
4. The End Nodes answer to the Multicast frame directly with an Acknowledgement frame ([LRMAC] 6.3.3).

3.19 MPDU Format, Mac Footer (MFR): FCS

FCS is the 16-bit non-correcting Frame Check Sequence (FCS) used for validating the integrity of a frame. It shall be calculated from the HomeID field to the Data Payload, both included ([LRMAC] 6.3.1.10).

3.19.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.19.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include End Nodes to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to each End Node.
3. Look for the FCS field in each frame.
4. Generate a singlecast frame with MPDU = 0x00 (NOP) and modify its FCS value to a random value, send it to each End Node separately.
5. Generate a singlecast frame with MPDU = 0x00 (NOP) and modify its FCS value to be more than 16 bits, send it to each End Node separately.
6. Generate a singlecast frame with MPDU = 0x00 (NOP) and modify its FCS value to be less than 16 bits, send it to each End Node separately.
7. Send an unmodified S2 multicast frame with MPDU = 0x00 (NOP) to both End Nodes as an S2 Multicast. Look for the FCS in the multicast.
8. Look for the FCS field in each singlecast follow-up frame.
9. Generate an S2 multicast frame with MPDU = 0x00 (NOP) and modify its FCS value to a random value, send it to both End Nodes. Look for the FCS in the multicast.
10. Generate an S2 multicast frame with MPDU = 0x00 (NOP) and modify its FCS value to be more than 16 bits, send it to both End Nodes. Look for the FCS in the S2 multicast.
11. Generate an S2 multicast frame with MPDU = 0x00 (NOP) and modify its FCS value to be less than 16 bits, send it to both End Nodes. Look for the FCS in the multicast.

3.19.3 Test Result

2. Communication is correct with both End Nodes and they answer with an Acknowledgement frame each.
3. Make sure there are 16 MFR bits corresponding to the FCS section in each frame.
4. Neither End Node answers to the frame since the FCS does not allow verification of its integrity.
    a. Each time the frame is displayed on the Sniffer as a CRC error.
    b. The controller tries retransmitting the frame since it doesn't receive an Acknowledgement frame.
5. Neither End Node answers to the frame since the modified FCS does not allow verification of its integrity.
    a. Each time the frame is displayed on the Sniffer as a CRC error.
    b. The controller tries retransmitting the frame since it doesn't receive an Acknowledgement frame and tries routing it through the other End Node.
6. Neither End Node answers to the frame since the modified FCS does not allow to verify its integrity.
    a. Each time the frame is displayed on the Sniffer as a CRC error.
    b. The controller tries retransmitting the frame since it doesn't receive an Acknowledgement frame and tries routing it through the other End Node.
7. The S2 Multicast is not responded with an Acknowledgement frame by either End Node.
    a. The FCS in the multicast frame is 16 bits long.
8. Same as in step 3.
9. The S2 multicast is not responded with an Acknowledgement frame by either End Node.
    a. The multicast frame is displayed on the Sniffer as a CRC error.
    b. The FCS in the multicast frame is 16 bits long.
    c. The singlecast follow-up frames are displayed correctly and responded by each End Node with Acknowledgement frames.*
10. The S2 multicast is not responded with an Acknowledgement frame by either End Node.
    a. The multicast frame is displayed on the Sniffer as a CRC error.
    b. The FCS in the multicast frame is more than 16 bits long and makes the frame end in a series of zeroes.*
    c. The singlecast follow-up frames are displayed correctly and responded by each End Node with Acknowledgement frames.*
11. The S2 multicast is not responded with an Acknowledgement frame by either End Node.
    a. The multicast frame is displayed on the Sniffer as a CRC error.
    b. The FCS in the multicast frame is less than 16 bits long and makes the frame end in a series of zeroes.*
    c. The singlecast follow-up frames are displayed correctly and responded by each End Node with Acknowledgement frames.*

    * Needs practical verification.

3.19.4 Pass Criteria

1. The FCS can only be 16 bits long at the end of the frame ([LRMAC] 6.3.1.10)
2. It signalizes the integrity of the frame and when it is modified, the receiver and Sniffer are uncapable of identifying it ([LRMAC] 6.3.1.10)

3.19.5 Fail Criteria

1. The FCS can be different from 16 bits long at the end of the frame ([LRMAC] 6.3.1.10)
2. When it is modified, the receiver and Sniffer are able to identify and respond to the frame ([LRMAC] 6.3.1.10)

3.20 Acknowledgement MPDU Format

The Acknowledgement MPDU format uses the general MPDU format from Test 3.9. It must be returned in the same Long Range channel as the singlecast that triggered it and only when the singlecast has its Ack Req subfield in the Frame Control field set to 1 ([LRMAC] 6.3.3).

3.20.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR End node

3.20.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include the End Node to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to the End Node.
3. Observe the structure of the Acknowledgement frame in the Sniffer.

3.20.3 Test Result

2. Communication is correct it answers with an Acknowledgement.
3. The Acknowledgement frame is displayed correctly on the Sniffer.

3.20.4 Pass Criteria

1. The Acknowledgement shows the same structure as the singlecast, with the following differences:
    a. Destination NodeID must be set to the NodeID value of the source NodeID of the singlecast that triggered the Ack frame: 12 bits ([LRMAC] 6.3.3.1).
    b. Frame Control, Ack Req subfield is set to 0: 1 bit ([LRMAC] 6.3.3.2.1).
    c. Frame Control, Header Type subfield is set to Acknowledgement (0x03): 3 bits ([LRMAC] 6.3.3.2.2, 6.3.1.5.3).
    d. Received RSSI is included before the Data Payload, it shows a valid value: 8 bits ([LRMAC] 6.3.3.4, Table 6-27).
    e. Data Payload contains any data ([LRMAC] 6.3.3.5).

3.20.5 Fail Criteria

1. At least one of the components of the format of the MPDU for Acknowledgement has a different length or holds a value different from the mandatory ones ([LRMAC] 6.3.3).

3.21 Acknowledgement MPDU Format, Received RSSI

The Received RSSI is an 8-bit signed field present only in Acknowledgement frames that indicates the signal strength measured while the corresponding singlecast frame is received. It averages at least 1 sample during reception and complies with the values in Table 6-27 ([LRMAC] 6.3.3.4).

3.21.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR Serial API End node

3.21.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver

1. Include the End Node to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to one End Node.
3. Observe the Received RSSI field in the Acknowledgement frame returned.
4. Place the End Node at a 50 cm distance. Send a singlecast to it. Observe the RSSI value in the Acknowledgement frame answered.
5. Build an Acknowledgement frame in the End Node with an invalid value to be sent when the Controller sends a new singlecast. Send a singlecast to the End Node. Observe its TX Power value.
6. Send a new singlecast to the End Node. Observe its TX Power value.

3.21.3 Test Result

2. The singlecast is received and answered by the End Node with an Acknowledgement frame.
3. The Received RSSI field in the Acknowledgement holds values within the ones defined as valid in table 6-27.
4. The RSSI value changes in at least 1dB due to the difference in intensity of the received signal.
5. The End Node answers with the custom Acknowledgement frame, the Controller does not validate this value and doesn't adjust the TX Power value.
6. The Tx Power value is not adjusted after the RSSI value.

3.21.4 Pass Criteria

1. The RSSI field consists of 8 bits at Byte 12 of the Acknowledgement frame and its values are within the valid ones ([LRMAC] 6.3.3.4, Table 6-27).
2. The RSSI indicates the strength of the received singlecast depending on the distance between Controller and End Node ([LRMAC] 6.3.3.4).
3. The RSSI values outside the valid ones do not have an effect on the following singlecasts.

3.21.5 Fail Criteria

1. The RSSI field is not 8 bits long and its values can be outside the valid ones ([LRMAC] 6.3.3.4, Table 6-27).
2. The RSSI is not corresponding to the strength of the received singlecast ([LRMAC] 6.3.3.4).
3. The RSSI values outside the valid ones Do influence the following singlecasts.

3.22 Broadcast MPDU Format

The Broadcast MPDU format uses the general MPDU format from Test 3.9. A broadcast MPDU is a singlecast MPDU (type 0x01) carrying destination Node ID = 0xFFF ([LRMAC] 6.3.4).

3.22.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.22.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include both End Nodes to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to each End Node.
3. Select both End Nodes and send a frame with MPDU = 0x00 (NOP) to both End Nodes as an S2 Multicast
4. Observe the structure of the Broadcast (S2 Multicast) in the Sniffer.

3.22.3 Test Result

2. Communication is correct with both End Nodes and they answer with an Acknowledgement frame each.
3. The Controller sends a Broadcast frame directed to NodeID 0xFFF (4095) followed by singlecast follow-up frames to each End Node.
4. The Broadcast is displayed correctly on the Sniffer.

3.22.4 Pass Criteria

1. The Broadcast shows the same structure as the singlecast, with the following differences:
    a. Destination NodeID must be set to the broadcast NodeID value 0xFFF: 12 bits ([LRMAC] 6.3.4.1, 6.3.1.3).
    b. Frame Control, Ack Req subfield is set to 0: 1 bit ([LRMAC] 6.3.4.2.1).
    c. Frame Control, Header Type subfield is set to Singlecast (0x01): 3 bits ([LRMAC] 6.3.4.2.2, 6.3.1.5.3).
    d. Data Payload shall contain at least 1 byte of data ([LRMAC] 6.3.4.3).

3.22.5 Fail Criteria

1. At least one of the components of the format of the MPDU for broadcast has a different length or holds a value different from the mandatory for Broadcast ([LRMAC] 6.3.4).

3.23 MPDU Header Extension Format

The extended MPDU header format is an extension to the General MPDU frame format as in test 3.9. Follows the Format from Table 6-28 and the values from Table 6-29 ([LRMAC] 6.3.5).

3.23.1 Prerequisites

1 x Sniffer
1 x Controller
2 x LR End node

3.23.2 Test Setup

We assume a Frame Generator is available in order to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include both End Nodes to the Controller's Network.
2. Send a singlecast with MPDU = 0x00 (NOP) to each End Node.
3. Build a singlecast frame that sets the Extended subfield in the Frame Control field to 1; includes an Extension Control Field and Extension Data Field, in which the Extension Control Field describes correctly the contents of the Extension Data field. Send it to the End Node.
4. Observe the structure of the singlecast in the Sniffer.

3.23.3 Test Result

2. Communication is correct with both End Nodes and they answer with an Acknowledgement frame each.
3. The Controller sends a singlecast with the configured values.
4. The Singlecast contains the Extension Control and Extension Data fields as well as the Extended subfield in the Frame Control field is enabled, as configured.

3.23.4 Pass Criteria

1. The Extended subfield in the Frame Control field is set to 1 ([LRMAC] 6.3.1.5.2).
2. The Extension Control field is located in the Byte 12 of the frame and consists of 8 bits of length ([LRMAC] 6.3.5.1).
3. The extension Type subfield consists of only 3 bits from bits 4 to 6 of the Frame Control field and it follows the values of Table 6-29 ([LRMAC] 6.3.5.1.1)
4. The Discard unknown subfield is only bit number 3 in the Extension Control field and it's set depending on whether the Controller considers possible to discard the Extension Data in case the receiving end doesn't know the Extension Type.
5. The extension Length subfield consist of three bits (0 – 2) in the Extension Control field and holds the value of the number of bytes that the Extension Data field holds ([LRMAC] 6.3.5.1.3).

3.23.5 Fail Criteria

1. At least one of the components of the format of the Extension for singlecast has a different length or holds a value different from the ones configured ([LRMAC] 6.3.5.1).

3.24 Beam Frame MPDU Format

Beam frames are used to awake Frequently Listening (FL) nodes. They are transmitted back to back to ensure an FL node can detect a beam within a short time window. Each beam frame shall carry the Beam Tag and NodeID fields. The NodeID field should be followed by the optional HomeID Hash field. An FL node shall stay awake to receive the MPDU that follows if there is a match with the Hash or NodeID, else it may return to sleep ([LRMAC] 6.3.6).

3.24.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR FL End node

3.24.2 Test Setup

1. Include FL End Node to the Controller's Network.
2. Send right away a singlecast with MPDU 0x00 (NOP) to the End Node.
3. Wait 10 seconds to make sure FL is asleep. Send the frame again.
4. Observe the Beam frame is sent to the End Node.
5. Observe the Beam Stop frame.
6. Once the beaming reaches the FL in a waking up state, the FL stays awake so that the Controller tries with a transmission of the original singlecast again.

3.24.3 Test Result

2. Communication is correct and FL End Node answers with an Acknowledgement frame.
3. Observe that Controller can't deliver the frame and retransmits it.
    a. As soon as it fails, it starts sending Beam frames to the FL End Node.
4. The Beam is shown correctly in the Sniffer.
5. The Beam Stop frame shows some number of beam count.
6. After the beaming, the Controller tries with the original singlecast sent and the FL End Node answers with an Acknowledgement frame in response.

3.24.4 Pass Criteria

1. The Beam frame consists of: ([LRMAC] 6.3.6 – Figure 6-8)
    a. A Beam tag: 0x55 (1 byte). ([LRMAC] 6.3.6.1 – Table 6-30)
    b. Tx Power (4 bits). ([LRMAC] 6.3.6.2 – Table 6-31)
    c. Destination NodeID: 0x100 .. 0xFA0, 0xFFF (12 bits). ([LRMAC] 6.3.6.3, 6.3.1.3 – Table 6-15)
    d. Field "HomeID Hash": (1 byte). ([LRMAC] 6.3.6.4)

3.24.5 Fail Criteria

1. Any of the elements of the Beam frame deviates from the description ([LRMAC] 6.3.6 – Table 6-30 – Table 6-15)

3.25 Fragmented Frame MPDU Format

A Beam Fragment comprises a number of beam frames. The Beam Fragment duration is in the range 110-115 ms. Beam frames shall be sent back to back to ensure the FL node can detect it upon waking up. The next Beam Fragment shall begin 190 – 200 ms after the beginning of the previous one. They shall be sent in different channels. When recognizing a Beam the receiving node shall answer with an Acknowledgement frame, upon receiving it, the Controller shall send the original singlecast but only if the Acknowledgement frame matches the originating HomeID and the NodeID of the destination NodeID on the original singlecast. A Beam Fragment can be addressed to 0xFFF (4095) turning it into a broadcast Wake Up Beam, but it can't be answered directly by the End Node ([LRMAC] 6.3.7).

3.25.1 Prerequisites

1 x Sniffer
1 x Controller
1 x LR FL End node

3.25.2 Test Setup

We assume a Frame Generator is available to generate frames with individual bits, bytes or sections modified individually in order to test the behavior of the receiver.

1. Include FL End Node to the Controller's Network.
2. Wait 10 seconds for FL to sleep. Send a singlecast with MPDU 0x00 (NOP) to the End Node.
3. Generate a Wake up Beam Frame with Beam Tag different from 0x55, send it to the End Node.
4. Generate a Wake Up Beam Frame with a Destination ID different from the End Node, send it to the End Node.
5. Generate a Wake Up Beam Frame with a Destination ID 0xFFF "Broadcast", send it to the End Node.
6. Generate a Wake Up Beam Frame with a random HomeID Hash hardcoded, send it to the End Node.
7. Generate a Wake Up Beam Frame and set it to be delayed more 300ms, send it to the End Node.

3.25.3 Test Result

2. Observe there are 2 wake up Beam frames sent to the End Node.
    a. Each Beam Fragment is sent between 80 and 90ms after the Beam Stop from the previous Beam Fragment (as to begin between 190 – 200 ms from the beginning of the previous Beam Fragment).
    b. When the End Node recognizes the Beam, it answers with an Acknowledgement frame.
    c. Controller repeats the original singlecast.
3. Observe there are 2 wake up Beam frames sent to the End Node.

   a. The End Node never recognizes the beam and the Controller continues sending the Wake Up Fragment until it times out.
4. Observe there are 2 wake up Beam frames sent to the End Node.
   a. The End Node never recognizes the beam and the Controller continues sending the Wake Up Fragment until it times out.
5. Observe there are 2 wake up Beam frames sent to the End Node.
   a. The End Node recognizes the beam but does not respond to the “Broadcast” Wake Up Fragment with an Acknowledgement Frame, the Controller continues sending the Wake Up Fragment until it times out.
6. Observe there are 2 wake up Beam frames sent to the End Node.
   a. The End Node never recognizes the beam and the Controller continues sending the Wake Up Fragment until it times out.
7. Observe there are 2 wake up Beam frames sent to the End Node with 300ms between transmissions.
   a. The End Node never manages to catch the beam when waking up and the Controller continues sending the Wake Up Fragment until it times out.*

   * Needs practical verification.

3.25.4 Pass Criteria

1. There are more than one wake up Beam frames sent to the End Node ([LRMAC] 6.3.7).
2. The fragment lasts between 110-115ms ([LRMAC] 6.3.7).
3. There are between 190 – 200ms between two Wake Up Beam Start in the Fragment ([LRMAC] 6.3.7).
4. The Fragments are sent in different Channels (A & B) ([LRMAC] 6.3.7).
5. The Beams can be addressed to any node ([LRMAC] 6.3.7).
6. The Receiving Node can validate a Hash of the HomeID ([LRMAC] 6.3.7).
7. The End Node answers the Beam with an Acknowledgement frame and the Controller repeats the original singlecast ([LRMAC] 6.3.7, 6.3.7.1).
8. The End Node will answer to a Fragmented Beam addressed to NodeID 0xFFF if macLRenableFLBroadcast is set to 1 and the HomeID Hash corresponds to its own, otherwise it won’t answer with an acknowledgement frame ([LRMAC] 6.3.7.1)

3.25.5 Fail Criteria

1. Any of the Pass Criteria is not met.

REFERENCES

[1] Z-Wave Alliance, ZWA_Z-Wave Long Range PHY and MAC Layer Specification_SPE_1.x
[2] Z-Wave Alliance, ZWA_Z-Wave Long Range PHY Layer Test Specification_SPE_1.x

