Sidewalk Compliance Test Implementation Guide

Version 1.0

2026-04-22

Change History

Version	Date	Description
1.0	2026-04-17	Initial release

Table of Contents

Test Execution Summary

BLE

Test Case	Executable by Customer
CMN/EN/API/INF/BV-01	Yes
BLE/EN/CONN/REG/BV/01	Yes
BLE/EN/CONN/REG/BV/02	Yes
BLE/EN/CONN/REG/BV/03	Yes
BLE/EN/DATA/DL/BV/01	Yes
BLE/EN/DATA/UL/BV/01	Yes
BLE-EP-CONN-EST-BV/01	Yes
BLE/EN/CONN/EST/BV/02	Yes
BLE/EN/CONN/EST/BV/03	Yes
BLE/EN/CONN/DSC/BV/01	Yes
BLE/EN/CONN/DSC/BV/02	Yes
BLE/EN/CONN/DSC/BV/03	Yes
BLE/EN/CONN/DSC/BV/06	Yes
BLE/EN/CONN/DSC/BV/07	Yes
BLE/EN/CONN/BCN/BV/01	Yes
BLE/EN/CONN/BCN/BV/02	Yes
BLE/EN/CONN/BCN/BV/03	Yes
BLE/EN/CONN/BCN/BV/04	Yes
BLE-EP-CONN-DUP-BV/01	Yes
BLE-EP-CONN-DUP-BV/02	Yes
BLE-EP-CONN-DUP-BV/03	Yes
BLE-EP-CONN-DUP-BV/04	Yes
BLE-EP-CONN-DUP-BV/05	Yes
BLE/EN/NW-SYNC/TIME/BV/01	Yes

FSK

Test Case	Executable by Customer
FSK/EN/BCN/DISCO/BV-01	Yes
FSK/EN/BCN/DISCO/BV-02	Yes
FSK/EN/BCN/DISCO/BV-03	Yes
FSK/EN/CONN/REG/BV-02	Yes
FSK/EN/CONN/REG/BV-03	Yes
FSK/EN/CONN/SEQ/BV-01	Yes
FSK/EN/CONN/SEQ/BV-02	Yes
FSK/EN/CONN/SEQ/BV-03	Yes
FSK/EN/CONN/SEQ/BV-05	Yes
FSK/EN/CONN/SEQ/BV-06	Yes
FSK-EP-CONN-SEQ-BV-07	Yes
FSK/EN/CONN/SEQ/BV-08	Yes
FSK/EN/CONN/SEQ/BV-09	Yes
FSK/EN/DATA/DL/BV-01	Yes
FSK/EN/DATA/UL/BV-01	Yes
FSK/EN/NW-SYNC/TIME/BV-01	Yes
FSK/EN/NW-SYNC/JOIN/BV-01	Yes
FSK/EN/NW-SYNC/JOIN/BV-02	Yes
FSK/EN/SEC/UUID/BV-01	Yes
FSK/EN/SEC/UUID/BV-02	Yes
FSK/EN/SEC/UUID/BV-03	Yes

LORA

Test Case	Executable by Customer
LORA/EN/DATA/DL/BV-01	Yes
LORA/EN/DATA/UL/BV-01	Yes
LORA/EN/SEC/UUID/BV-01	Yes
LORA/EN/SEC/UUID/BV-02	Yes
LORA/EN/SEC/UUID/BV-03	Yes
LORA/EN/CONN/REG/BV-02	Yes
LORA/EN/CONN/LP/BV-01	Yes
LORA/EN/CONN/LP/BV-02	Yes
LORA/EN/CONN/SEQ/BV-01	Yes
LORA/EN/CONN/SEQ/BV-02	Yes
LORA/EN/CONN/SEQ/BV-03	Yes
LORA/EN/CONN/SEQ/BV-05	Yes
LORA/EN/CONN/SEQ/BV-06	Yes
LORA/EN/CONN/SEQ/BV-07	Yes
LORA/EN/CONN/SEQ/BV-08	Yes
LORA/EN/CONN/SEQ/BV-09	Yes
LORA/EN/NW-SYNC/JOIN/BV-02	Yes
LORA/EN/NW-SYNC/JOIN/BV-03	Yes
LORA/EN/NW-SYNC/TIME/BV-01	Yes

MLM

Test Case	Executable by Customer
MULTI-LINK/EP/BLE-AUTO-CONNECT/UL/BI/01	Yes
MULTI-LINK/EP/BLE-AUTO-CONNECT/UL/BV/02	Yes
MULTI-LINK/EP/FSK-AUTO-CONNECT/UL/BI/01	Yes
MULTI-LINK/EP/FSK-AUTO-CONNECT/UL/BV/02	Yes
MULTI-LINK/EP/LoRa-AUTO-CONNECT/UL/BI/01	Yes
MULTI-LINK/EP/LoRa-AUTO-CONNECT/UL/BV/02	Yes
MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/01	No
MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/02	No
MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/03	No
MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/04	No
MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/01	No
MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/02	No
MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/03	No
MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/04	No
MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/05	No
MULTI-LINK/EP/MLM-ACTIVE/UL/BV/01	No
MULTI-LINK/EP/MLM-ACTIVE/UL/BV/02	No
MULTI-LINK/EP/MLM-ACTIVE/UL/BV/03	No
MULTI-LINK/EP/MLM-ACTIVE/UL/BV/04	No
MULTI-LINK/EP/MLM-ACTIVE/UL/BV/05	No
MULTI-LINK/EP/MLM-POWER/UL/BV/01	No
MULTI-LINK/EP/MLM-POWER/UL/BV/02	No
MULTI-LINK/EP/MLM-POWER/UL/BV/03	No
MULTI-LINK/EP/MLM-POWER/UL/BV/04	No
MULTI-LINK/EP/MLM-POWER/UL/BV/05	No
MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/01	No
MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/02	No
MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/03	No
MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/04	No

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/05	No
MULTI-LINK/EP/MLM-LATENCY/UL/BV/01	No
MULTI-LINK/EP/MLM-LATENCY/UL/BV/02	No
MULTI-LINK/EP/MLM-LATENCY/UL/BV/03	No
MULTI-LINK/EP/MLM-LATENCY/UL/BV/04	No
MULTI-LINK/EP/MLM-LATENCY/UL/BV/05	No
MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/01	No
MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/02	No
MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/03	No
MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/04	No
MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/05	No

Multi-Radio

Test Case	Executable by Customer
MULTI/EN/API/INIT/BV-01	Yes
MULTI/EN/CONN/DL/BV-01	Yes
MULTI/EN/CONN/DL/BV-02	Yes
MULTI/EN/CONN/DL/BV-03	Yes
MULTI/EN/CONN/SEND-LINK/BV-01	Yes
MULTI/EN/CONN/SEND-LINK/BV-02	Yes
MULTI/EN/CONN/SEND-LINK/BV-03	Yes
MULTI/EN/CONN/SEND-LINK/BV-04	Yes
MULTI/EN/CONN/SEND-LINK/BV-05	Yes

File Transfer

Test Case	Executable by Customer
BLE/EP/API/FILETRANSFER/BV/01	Yes
BLE/EP/API/FILETRANSFER/BV/02	Yes
BLE/EP/API/FILETRANSFER/BV/03	Yes
BLE/EP/API/FILETRANSFER/BV/04	Yes
BLE/EP/API/FILETRANSFER/RECOVERY/BI/01	Yes
BLE/EP/API/FILETRANSFER/RECOVERY/BI/02	Yes
BLE/EP/API/FILETRANSFER/RECOVERY/BI/03	No
BLE/EP/API/FILETRANSFER/RECOVERY/BI/04	No
BLE/EP/API/FILETRANSFER/RECOVERY/BI/05	Yes

EP Metrics

Test Case	Executable by Customer
CMN/EP/EPMETRICS/METRICS/BV/01	No
BLE/EP/EPMETRICS/REPORT/BV/01	No
BLE/EP/EPMETRICS/REPORT/BV/02	No
BLE/EP/EPMETRICS/REPORT/BV/03	No
BLE/EP/EPMETRICS/UL/BV/01	No
BLE/EP/EPMETRICS/DL/BV/01	No
BLE/EP/EPMETRICS/CONN-BV-01	No
FSK/EP/EPMETRICS/REPORT/BV/01	No
FSK/EP/EPMETRICS/REPORT/BV/02	No
FSK/EP/EPMETRICS/REPORT/BV/03	No
FSK/EP/EPMETRICS/UL/BV/01	No
FSK/EP/EPMETRICS/DL/BV/01	No
LORA/EP/EPMETRICS/REPORT/BV/01	No

LORA/EP/EPMETRICS/REPORT/BV/02	No
LORA/EP/EPMETRICS/REPORT/BV/03	No
LORA/EP/EPMETRICS/UL/BV/01	No
LORA/EP/EPMETRICS/DL/BV/01	No

EP Capability

Test Case	Executable by Customer
CMN-EP-CAPABILITY-REPORT-BV-01	Yes
CMN-EP-CAPABILITY-REPORT-BV-02	Yes
CMN-EP-CAPABILITY-REPORT-BV-03	No
CMN-EP-CAPABILITY-REQUEST-BV-01	No
CMN-EP-CAPABILITY-CONFIGURE-BV-01	No
CMN-EP-CAPABILITY-REGISTER-BV-01	No
CMN-EP-CAPABILITY-THRESHOLD-BV-01	No
CMN-EP-CAPABILITY-PERIODICITY-BV-01	Yes
CMN-EP-CAPABILITY-PERIODICITY-BV-02	Yes
BLE-EP-CAPABILITY-THRESHOLD-BV-01	No
FSK-EP-CAPABILITY-THRESHOLD-BV-01	No
LORA-EP-CAPABILITY-THRESHOLD-BV-01	No

MAC Address Rotation

Test Case	Executable by Customer
BLE/EP/BCN/MAC/BV/01	Yes
BLE/EP/BCN/MAC/BV/02	Yes
BLE/EP/BCN/MAC/BV/03	Yes
BLE/EP/BCN/MAC/BV/04	Yes
BLE/EP/BCN/MAC/BV/05	Yes
BLE/EP/BCN/MAC/BV/06	Yes
BLE/EP/BCN/MAC/BV/07	No
BLE/EP/BCN/MAC/BV/08	Yes

Power Optimization

Test Case	Executable by Customer
FSK/EP/PWR/OPT/BV/01	Yes
FSK/EP/PWR/OPT/BV/02	Yes
FSK/EP/PWR/OPT/BV/03	Yes
FSK/EP/PWR/OPT/BV/04	Yes
FSK/EP/PWR/OPT/BV/05	Yes
FSK/EP/PWR/OPT/BV/06	Yes
FSK/EP/PWR/OPT/BV/07	Yes
LORA/EP/PWR/OPT/BV/01	Yes
LORA/EP/PWR/OPT/BV/02	Yes
LORA/EP/PWR/OPT/BV/03	Yes
LORA/EP/PWR/OPT/BV/04	Yes

BLE Conn Policy

Test Case	Executable by Customer
BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-01	Yes
BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-02	Yes

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-03	Yes
BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-04	Yes
BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-05	Yes
BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-01	Yes
BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-02	Yes
BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-03	Yes
BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-04	Yes
BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING-BV-01	Yes
BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-02	Yes
BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-03	Yes
BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-04	Yes

Required Log Patterns

Pattern Name	Log Example	Description
SID_LAST_STATUS	EVENT SID STATUS: State: <s>, Reg: <r>, Time: <t>, Link_Mask: <m> EVENT SID STATUS LINK MODE: LORA: <n>, FSK: <n>, BLE: <n>	EN current status: registration state, time sync state, and active link mask.
SID_STATUS_TIME_SYNC_COMPLETED	EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: <n>	EN completed time sync. Link_Mask shows active links.
TIME_REQUEST_SEQ	Sent time req [<n>] to CldID	EN sent time sync request to cloud (sequence number).
TIME_RESPONSE_TR_STORE	tr_store: tr from:(radio|spi|stack), c_cls:0x0000, c_id:0x108(rsp)...	EN stored time response from cloud.
TIME_UPDATED	Updated time:	EN successfully updated its clock from cloud time.
TIME_FROM_THE_CLOUD	Received time from the cloud round trip	EN received time from cloud and completed round trip.
BLE_CONNECTED	EVENT SID STATUS: ..., Link_Mask: 1 EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1	BLE connection established (Link_Mask: 1).
BLE_DISCONNECTED	EVENT SID STATUS: ..., Link_Mask: 0 EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0	BLE connection dropped (Link_Mask: 0).
SID_UPLINK_EVENT	EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: <link> LINK_MODE: 0	EN successfully sent an uplink packet to cloud.
SID_DOWNLINK_EVENT	EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: <link>, LINK_MODE: 0, ACK: 1 Data: <data>	EN received a downlink packet from cloud.
JOIN_TERMINATE_SUCCEED	Njoin(<n>, <n>) st <n>->6	EN completed join procedure, FSK/LoRa link established.
SID_STATUS_REGISTERED	Reg: 0	EN is registered with Sidewalk network.
SID_STATUS_NOT_REGISTERED	Reg: 1	EN is NOT registered (deregistered).
FAILED_COMMAND	CMD: ERR: -<n>	EN command failed with error code (e.g. -40 = no connection available).
CAPABILITY_ACK_RECV	DEC: M:15 [0:2:3]	EN received ACK for capability response sent to cloud.
GWD_SCAN_CONNECTED	GWD(<n>) st <n>->6	GW discovery scan completed, GW found and connected.
TX_DONE_APP_LAYER_DST_ID	FskTx:TxDone sqn:<sqn> dst_id:0x<id> ch:<ch> pktlen:<len> tx_st:0	FSK/LoRa TX packet sent to cloud (dst_id=0x4101).
TX_UUID	TXID: 0x<uuid>	EN current TX UUID. Used to verify UUID rotation every 15 minutes.
CAPABILITY_NTFY_SENT	ENC: M:15 [0:2:2]	EN sent capability notification to cloud.
CAPABILITY_WRITE_RECV	DEC: M:15 [0:2:1]	EN received capability write (configuration) from cloud.
TRAFFIC_THRESHOLD_EXCEED	TRAFFIC THRESHOLD RATE EXCEEDED	EN exceeded configured traffic rate limit.
SEND_METRICS	ENC: M:15 [B:E:2]	EN sending endpoint metrics to cloud via uplink.
NDB_METRICS	MET: NDB5 st:0	Remaining metrics queued for piggybacking on next uplink.
FILE_TRANSFER_COMPLETED_COMMAND	DEC: M:15 [3:D:2]	File transfer (SBDT) completed successfully.
FILE_TRANSFER_SENDING_BLOCKS	DEC: M:15 [3:5:1]	File transfer block data being received.
BEACON_RECEIVED	BCN:pkt rcvd gw_id:<id> ch:<ch> rssi:<rssi>	EN received a FSK beacon from GW.
AUTO_CONNECT_LOG	ACM: Connect <link> for <n> sec err 0	Auto-connect manager attempting to connect on specified link.
MLM_CONNECT_LOG	MLM: Connect <link> for <n> sec err 0	Multi-link manager attempting to connect on specified link.
AC_TERMINATE	ACM(<n>) st <n>->0	Auto-connect manager terminated, all retry attempts exhausted.
SID_SDK_VERSION	SID_SDK_VERSION: <version>	EN SDK version string.
BLE_POLICY	CMD: ERR: 0 BLE conn policy <n>	BLE connection policy set (0=Default, 1=LongLived, 2=OptimalAdv).
BLE_CONFIG_ADV	CMD: ERR: 0 BLE adv, fast_int <n>(<ms>ms), fast_to <n>,	BLE advertisement interval parameters

	slow_int <n>(<ms>ms), slow_to <n>	configured.
BLE_CONFIG_PARAM	CMD: ERR: 0 BLE conn, min_int: <n>(<ms>ms), max_int <n>, sl <n>, timeout <n>	BLE connection interval parameters configured.
BLE_INACTIVITY_TIMEOUT	CMD: ERR: 0 BLE inactivity timeout <n>s	BLE inactivity timeout configured.

BLE Test Cases

CMN/EN/API/INF/BV-01: End Node has correct MCU version after launching Sidewalk application

Case ID: C103619213

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Check response on command:

sid sdk_version

Expected Result:

Response example:

<info> app: 1.19.5-36 (rev. 0)

BLE/EN/CONN/REG/BV/01: An unregistered End Node successfully completes Sidewalk device registration over BLE

Case ID: C103619399

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following commands on the EN device:

sid init 1

sid start

Expected Result:

Each command returns: CMD: ERR: 0

Step 2

Description:

Place EN next to a registered GW and wait for BLE registration to complete.

Verify registration status with:

sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0

Key check: Reg: 0 (registered)

BLE/EN/CONN/REG/BV/02: An unregistered End Node successfully completes Sidewalk device registration with use of FFN

Case ID: C103619414

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following commands on the EN device:

sid init 1

sid start

Expected Result:

Each command returns: CMD: ERR: 0

Step 2

Description:

Place the EN next to a GW

Expected Result:

Registration process should start automatically

Step 3

Description:

Verify registration status with:

sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0 Key check: Reg: 0 (registered)

BLE/EN/CONN/REG/BV/03: End Node can be deregistered from Sidewalk network

Case ID: C103619400

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following command to deregister the device:

sid factory_reset

Expected Result:

CMD: ERR: 0

Step 2

Description:

Verify deregistration status with:

sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 1, Time: 1, Link_Mask: 0

Key check: Reg: 1 (deregistered)

BLE/EN/DATA/DL/BV/01: Endpoint receives downlink packets with various sizes of Command data

Case ID: C103619402

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:
Send 1 downlink packet from AWS IoT to EN with 1 byte payload
Expected Result:
EVENT SID RECEIVED: TYPE: 1, ID: <id>, LEN: 1, LINK: 1, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x> Data: <payload> Key check: ID matches sent packet, LEN: 1

Step 2

Description:
Send 1 downlink packet from AWS IoT to EN with 125 bytes payload
Expected Result:
EVENT SID RECEIVED: TYPE: 1, ID: <id>, LEN: 125, LINK: 1, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x> Data: <payload> Key check: ID matches sent packet, LEN: 125

Step 3

Description:
Send 1 downlink packet from AWS IoT to EN with 255 bytes payload
Expected Result:
EVENT SID RECEIVED: TYPE: 1, ID: <id>, LEN: 255, LINK: 1, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x> Data: <payload> Key check: ID matches sent packet, LEN: 255

BLE/EN/DATA/UL/BV/01: Gateway receives uplink packet with different payload sizes from End Node

Case ID: C103619403

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized
BT connection is established (sid conn_req 1)

Test Steps:

Step 1

Description:
Run the following command on EN to send 1 UL packet with 1 byte payload:
sid send <1-byte-payload>
Expected Result:
CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: SID ERR: 0 (sent successfully), packet received on AWS IoT

Step 2

Description:
Run the following command on EN to send 1 UL packet with 125 bytes payload:
sid send <125-byte-payload>
Expected Result:
CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: SID ERR: 0 (sent successfully), packet received on AWS IoT

Step 3

Description:
Run the following command on EN to send 1 UL packet with 255 bytes payload:
sid send <255-byte-payload>
Expected Result:
CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: SID ERR: 0 (sent successfully), packet received on AWS IoT

BLE-EP-CONN-EST-BV/01: Gateway establishes BLE connection on Endpoint’s request.

Case ID: C103619417

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:

Run the following command on EN:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (BLE connected)

BLE/EN/CONN/EST/BV/02: End Node can connect with Gateway via BLE, after reinitializing Sidewalk library

Case ID: C103619412

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:
Deinitialize Sidewalk:
sid deinit
Expected Result:
CMD: ERR: 0

Step 2

Description:
Initialize and start Sidewalk, then wait for time sync:
sid init 1
sid start
Expected Result:
Each command returns: CMD: ERR: 0
Then wait for time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1 Key check: Time: 0 (time synced)

Step 3

Description:
Wait for BLE disconnected, verify with:
sid last_status
Expected Result:
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0 Key check: BLE: 0 (disconnected)

Step 4

Description:
Set connection request on EN:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1 Key check: BLE: 1 (BLE connected)

BLE/EN/CONN/EST/BV/03: End Node can connect with Gateway via BLE, after restarting Sidewalk library

Case ID: C103619413

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:
Stop Sidewalk:
sid stop
Expected Result:
CMD: ERR: 0

Step 2

Description:
Start Sidewalk:
sid start
Expected Result:
CMD: ERR: 0

Step 3

Description:
Verify BLE is disconnected:
sid last_status
Expected Result:
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected)

Step 4

Description:
Set connection request on EN:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (BLE connected)

BLE/EN/CONN/DSC/BV/01: BLE connection between Gateway and End Node is dropped after 30 seconds of being idle

Case ID: C103619404

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Do not send any UL or DL for 30 seconds.
Verify BLE is disconnected:
sid last_status

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected after 30s idle)

BLE/EN/CONN/DSC/BV/02: BLE connection between Gateway and End Node is extended by uplink packet and dropped after additional 30 seconds

Case ID: C103619405

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:
Wait 15 seconds. Do not send any UL or DL.
Expected Result:
BLE connection still active
BLE: 1

Step 3

Description:
Send UL packet from EN:
sid send TEST
Expected Result:
CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: SID ERR: 0 (sent successfully)

Step 4

Description:
Wait 30 seconds after UL sent.
Verify BLE is disconnected:
sid last_status
Expected Result:
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected after 30s)

BLE/EN/CONN/DSC/BV/03: BLE connection between Gateway and End Node is extended by downlink packet and dropped after additional 30 seconds

Case ID: C103619406

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN is time synchronized

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:
Wait 15 seconds. Do not send any UL or DL.
Expected Result:
BLE connection still active
BLE: 1

Step 3

Description:
Send DL packet from AWS IoT to EN via REST API
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: DL received on EN

Step 4

Description:
Wait 30 seconds after DL received.
Verify BLE is disconnected:
sid last_status
Expected Result:
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected after 30s)

BLE/EN/CONN/DSC/BV/06: End Node drops BLE connection after stopping Sidewalk library

Case ID: C103619410

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network\nEN is registered in Sidewalk network\nEN has time synchronized

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:

sid conn_req 1

Expected Result:

CMD: ERR: 0

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Stop Sidewalk:

sid stop

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: Link_Mask: 0, BLE: 0

BLE/EN/CONN/DSC/BV/07: End Node drops BLE connection after deinitializing Sidewalk library

Case ID: C103619411

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network\nEN is registered in Sidewalk network\nEN has time synchronized

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Deinitialize Sidewalk:
sid deinit

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: Link_Mask: 0, BLE: 0

BLE/EN/CONN/BCN/BV/01: End Node transmits beacons only when BLE connection is not established

Case ID: C103619407

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network\nEN is registered in Sidewalk network\nEN has time synchronized\nBT sniffer is ready

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

While BLE connection is active, monitor BT sniffer for 10 seconds

Expected Result:

No beacons captured by BT sniffer

Key check: Beacon is invisible when BLE connected

Step 3

Description:

Wait for BLE to disconnect (idle 30s).
Monitor BT sniffer for 10 seconds.

Expected Result:

BLE disconnected:
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Beacons captured by BT sniffer

Key check: Beacon is visible after BLE disconnected

BLE/EN/CONN/BCN/BV/02: End Node transmits beacons that contain Identifier (TX-ID) which changes every 15 minutes

Case ID: C103619416

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network

Test Steps:

Step 1

Description:
Initialize BLE stack and wait for BLE disconnected:
sid init 1
sid start
Expected Result:
EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE disconnected

Step 2

Description:
Use BLE sniffer to collect 2 TX-ID rotations (wait up to 20 min each):
Monitor beacon TX-ID changes in application specific data
Expected Result:
TX-ID rotates twice
Each rotation logged with elapsed time
Key check: TX-ID changes observed in beacon data

Step 3

Description:
Verify TX-ID rotation interval is within 14~16 minutes
Expected Result:
Rotation time: 840s~960s (14~16 min)
Key check: TX-ID rotates every ~15 minutes per Sidewalk spec

BLE/EN/CONN/BCN/BV/03: End Node transmits beacons with 160ms interval for the first 30 seconds after starting Sidewalk library

Case ID: C103619408

Executable by Customer: Yes

Preconditions:

EN is registered in Sidewalk network\nGW is turned off\nBT sniffer is ready

Test Steps:

Step 1

Description:

Power on device, catch beacons

Expected Result:

Measure the time between beacons for 30 seconds. The mean interval should be around 160 ms.

BLE/EN/CONN/BCN/BV/04: End Node transmits beacons with 1 s interval, 30 seconds after starting Sidewalk library

Case ID: C103619409

Executable by Customer: Yes

Preconditions:

EN is registered in Sidewalk network\nGW is turned off\nBT sniffer is ready

Test Steps:

Step 1

Description:

Power on device, wait 2 min, catch beacons

Expected Result:

Measure the time between beacons for 30 seconds. The mean interval should be around 1000ms

BLE-EP-CONN-DUP-BV/01: BLE connection is closed after Endpoint receives 10 packets with duplicate SEQ number within one minute (filter duplicates enabled)

Case ID: C103619419

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized
Duplicate filter enabled: sid option -d 0

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0 EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1 Key check: BLE: 1 (connected)

Step 2

Description:
Send 1 DL packet from AWS IoT with SEQ number <seq>
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x> Data: <payload> Key check: DL received on EN

Step 3

Description:
Send 10 DL packets with the same SEQ number as Step 2, all within 1 minute
Expected Result:
BLE connection terminated after 10th duplicate DL received: [INFO]: BLE (Disconnected|DISCONNECTED) Key check: BLE disconnected

BLE-EP-CONN-DUP-BV/02: BLE connection is closed after Endpoint receives 10 packets with various duplicate SEQ numbers within one minute (filter duplicates enabled).

Case ID: C103619418

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized
Duplicate filter enabled: sid option -d 0

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0 EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1 Key check: BLE: 1 (connected)

Step 2

Description:
Send 3 DL packets with various SEQ numbers (e.g. SEQ=1, SEQ=2, SEQ=3)
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, ... Data: <payload> Key check: All 3 DL packets received on EN

Step 3

Description:
Send within 1 minute:
- 3x DL with SEQ=1
- 3x DL with SEQ=2
- 4x DL with SEQ=3
Expected Result:
BLE connection terminated after 10th duplicate DL received: [INFO]: BLE (Disconnected|DISCONNECTED) Key check: BLE disconnected

BLE-EP-CONN-DUP-BV/03: BLE connection is not closed after Endpoint receives 10 packets with duplicate SEQ number within one minute (filter duplicates disabled).

Case ID: C103619420

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized
Duplicate filter disabled: sid option -d 1

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
Key check: BLE: 1 (connected)

Step 2

Description:
Send 1 DL packet from AWS IoT with SEQ number <seq>
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, ...
Data: <payload>
Key check: DL received on EN

Step 3

Description:
Send 10 DL packets with the same SEQ number as Step 2, all within 1 minute
Expected Result:
BLE connection remains active after 10th duplicate DL received
Key check: BLE NOT disconnected

BLE-EP-CONN-DUP-BV/04: BLE connection is not closed after Endpoint receives 10 packets with various duplicate SEQ numbers within one minute (filter duplicates disabled).

Case ID: C103619421

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network\nEN is registered in Sidewalk network\nEN has time synchronized\nDuplicate filter disabled: sid option -d 1

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Send 3 DL packets with various SEQ numbers (e.g. SEQ=1, SEQ=2, SEQ=3)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, ...
Data: <payload>

Key check: All 3 DL packets received on EN

Step 3

Description:

Send within 1 minute:
- 3x DL with SEQ=1
- 3x DL with SEQ=2
- 4x DL with SEQ=3

Expected Result:

BLE connection remains active after 10th duplicate DL received

Key check: BLE NOT disconnected

BLE-EP-CONN-DUP-BV/05: BLE connection is not closed after 10 packets with duplicate SEQ number reach Endpoint, if not all of those packets are sent within one minute.

Case ID: C103619422

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network
EN has time synchronized
Duplicate filter enabled: sid option -d 0

Test Steps:

Step 1

Description:
Verify BLE is disconnected, then set connection request:
sid conn_req 1
Expected Result:
CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
Key check: BLE: 1 (connected)

Step 2

Description:
Send 1 DL packet from AWS IoT with SEQ number <seq>
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 1, ...
Data: <payload>
Key check: DL received on EN

Step 3

Description:
Send 10 DL packets with the same SEQ number as Step 2, with 7s interval between each
(10th duplicate DL received more than 1 minute after 1st duplicate)
Expected Result:
BLE connection remains active after 10th duplicate DL received
Key check: BLE NOT disconnected (packets spread over >1 minute)

BLE/EN/NW-SYNC/TIME/BV/01: End Node completes the initial time sync procedure

Case ID: C103619401

Executable by Customer: Yes

Preconditions:

GW is registered in Sidewalk network
EN is registered in Sidewalk network

Test Steps:

Step 1

Description:

Initialize and start Sidewalk, then wait for time sync:
sid init 1
sid start

Expected Result:

Each command returns: CMD: ERR: 0 Then wait for time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1 Key check: Time: 0 (time synced)

FSK Test Cases

FSK/EN/BCN/DISCO/BV-01: FSK-WAN Endpoint discovers neighbor Gateway beacon and synchronizes with it if roaming is allowed

Case ID: C83932917

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Initialize and start FSK stack:
sid init 2
sid start

Expected Result:

Each command returns: CMD: ERR: 0
EN detects beacon and synchronizes with GW within 40 seconds

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0

Key check: Time: 0 (time synced), FSK: 1

FSK/EN/BCN/DISCO/BV-02: FSK-WAN Endpoint discovers home Gateway and synchronizes with it even if roaming is forbidden

Case ID: C83932918

Executable by Customer: Yes

Preconditions:

PAN device.

Test Steps:

Step 1

Description:

Initialize and start FSK stack:
sid init 2
sid start

Expected Result:

Each command returns: CMD: ERR: 0

Step 2

Description:

EN gets GW FSK beacon

Expected Result:

```
[INFO]: Disco active
[INFO]: GWD scan started
```

Step 3

Description:

EN accepts beacon and completes time sync.
Verify with:
sid last_status

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0

Key check: Time: 0 (time synced), FSK: 1

FSK/EN/BCN/DISCO/BV-03: FSK-WAN Endpoint detects a neighbor Gateway beacon but continues with the GW discovery process if roaming is forbidden

Case ID: C83932919

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Disable Sidewalk on GW via Amazon Alexa app:

Amazon Alexa app -> More -> Account Settings -> Amazon Sidewalk -> Enabled -> Disable

Expected Result:

Amazon Sidewalk set to OFF on Mobile.

Step 2

Description:

Restart EN:

sid init 2

sid start

Expected Result:

Each command returns: CMD: ERR: 0

Step 3

Description:

Verify EN starts GW discovery (Disco active)

Expected Result:

`[INFO]: Disco active. LSC=<x>, symb=<x>, rx_to_ms=<x>`

Step 4

Description:

Wait for Disco stop sampling success

Expected Result:

`[INFO]: Disco stop sampling success. Time elapsed = 120002 ms (+-1000ms) Key check: EN continues GW discovery without synchronizing to neighbor GW`

FSK/EN/CONN/REG/BV-02: Endpoint can be deregistered from Sidewalk network

Case ID: C83932921

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following command to deregister the device:

sid factory_reset

Expected Result:

CMD: ERR: 0

EVENT SID STATUS: State: 1, Reg: 1, Time: 1, Link_Mask: 0

Key check: Reg: 1 (deregistered)

FSK/EN/CONN/REG/BV-03: An unregistered Endpoint successfully completes Sidewalk device registration with use of FFN

Case ID: C83932922

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following commands on the EN device:

sid init 2

sid start

Expected Result:

Each command returns: CMD: ERR: 0

Step 2

Description:

Place EN next to a GW.

Verify registration status with:

sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0

Key check: Reg: 0 (registered)

FSK/EN/CONN/SEQ/BV-01: Uplink packets contain SEQ number

Case ID: C83932923

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/46066068
And
https://issues.labcollab.net/browse/HALO-31618

Test Steps:

Step 1

Description:

Send 5 UL packets from EN with 10 byte payload:
sid send <10-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 2 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: UL packets received on AWS

FSK/EN/CONN/SEQ/BV-02: Endpoint increments uplink SEQ until UUID changes

Case ID: C83932925

Executable by Customer: Yes

Preconditions:

Spec: https://issues.labcollab.net/browse/HALO-31619
Step: https://ring.testrail.net/index.php?/cases/view/46066076

Test Steps:

Step 1

Description:

Send 1 UL packet:
sid send TEST

Expected Result:

FskTx:EnqPkt sqn:<seq> src_frmt:1 src_id:0x<txid> dst_frmt:0 dst_id:0x<gw_id> msg_id:<id>

Key check: Note the SEQ number

Step 2

Description:

Wait for UUID rotation (~15 minutes after time sync)

Expected Result:

UUID rotated (TX UUID changes)

Step 3

Description:

Send 1 UL packet after UUID rotation:
sid send TEST

Expected Result:

FskTx:EnqPkt sqn:<new_seq> src_frmt:1 src_id:0x<new_txid> dst_frmt:0 dst_id:0x<gw_id> msg_id:<id>

Key check: New SEQ number is different from Step 1 (reset after UUID change)

FSK/EN/CONN/SEQ/BV-03: Endpoint accepts and decodes downlink packets with increased and decreased SEQ

Case ID: C83932926

Executable by Customer: Yes

Preconditions:

increased SEQ -> https://ring.testrail.net/index.php?/cases/view/46066069
decreased SEQ -> https://ring.testrail.net/index.php?/cases/view/46066070

Test Steps:

Step 1

Description:
Send 3 DL packets from AWS IoT with SEQ: 1, 2, 3
Expected Result:
All 3 DL packets received and accepted by EN:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: Delivery ratio 100%

Step 2

Description:
Send 3 DL packets from AWS IoT with SEQ: 6, 5, 4
Expected Result:
All 3 DL packets received and accepted by EN:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: Delivery ratio 100%

FSK/EN/CONN/SEQ/BV-05: Endpoint rejects downlink packet with duplicated SEQ

Case ID: C83932927

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/46066071

Test Steps:

Step 1

Description:

Send DL with SEQ=1

Then send DL with SEQ=1 again (duplicate)

Expected Result:

First: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ... (accepted)

Second: No EVENT SID RECEIVED within 20 seconds (duplicate rejected)

Key check: Duplicate rejected

Step 2

Description:

Send DL with SEQ=2

Then send DL with SEQ=2 again (duplicate)

Expected Result:

First: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ... (accepted)

Second: No EVENT SID RECEIVED within 20 seconds (duplicate rejected)

Key check: Duplicate rejected

FSK/EN/CONN/SEQ/BV-06: Endpoint rejects downlink packet with duplicated SEQ range

Case ID: C83932928

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 5 DL packets with SEQ: 1, 2, 3, 4, 5

Expected Result:

EVENT SID RECEIVED: TYPE: 2, ID: <id>, LEN: <len>, LINK: 2, LINK_MODE: 1, ACK_REQUESTED: 0, ACK: 0, DUP: 0, RSSI: <rssi>, SNR: <snr>

Key check: All 5 packets received

Step 2

Description:

Send 5 DL packets with SEQ: 1, 2, 3, 4, 5 again (all duplicates)

Expected Result:

No EVENT SID RECEIVED for any packet

Key check: All duplicates rejected

FSK-EP-CONN-SEQ-BV-07: SEQ number can be accepted again after getting 10 other messages

Case ID: C83932930

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 10 DL packets from AWS IoT with SEQ numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Expected Result:

Each DL packet received and accepted by EN:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: - , ...
Data: <payload>

Key check: Delivery ratio 100%

Step 2

Description:

Send 1 DL packet with SEQ=1 (duplicate of Step 1)

Expected Result:

DL packet rejected by EN (duplicate)

Key check: Delivery ratio 0%

Step 3

Description:

Send 1 DL packet with SEQ=11 (new, non-duplicate)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: - , ...
Data: <payload>

Key check: Delivery ratio 100%

Step 4

Description:

Send 1 DL packet with SEQ=1 again (now outside the 10-SEQ window)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: - , ...
Data: <payload>

Key check: SEQ=1 accepted again after 10 other messages, Delivery ratio 100%

FSK/EN/CONN/SEQ/BV-08: SEQ number can be accepted again after End Node reinitialization

Case ID: C83932933

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send DL packet with SEQ=1

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: Packet accepted

Step 2

Description:

Send DL packet with SEQ=1 again (duplicate)

Expected Result:

No EVENT SID RECEIVED within 20 seconds

Key check: Duplicate rejected

Step 3

Description:

Reinitialize Sidewalk stack:
sid stop
sid deinit
sid init 2
sid start

Expected Result:

Each command returns: CMD: ERR: 0
Wait for time sync:
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Wait for join:
Nsync st ->2

Step 4

Description:

Send DL packet with SEQ=1 again (after reinit)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: SEQ=1 accepted again after reinit

FSK/EN/CONN/SEQ/BV-09: Endpoint drops the packet with SEQ which is out of range

Case ID: C83932934

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send DL packet with SEQ=0 (minimum valid)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: Accepted

Step 2

Description:

Send DL packet with SEQ=16383 (maximum valid)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 2, ...

Key check: Accepted

Step 3

Description:

Send DL packet with SEQ=-1 (out of range)

Expected Result:

No EVENT SID RECEIVED within 20 seconds

Key check: Rejected (out of range)

Step 4

Description:

Send DL packet with SEQ=16384 (out of range)

Expected Result:

No EVENT SID RECEIVED within 20 seconds

Key check: Rejected (out of range)

FSK/EN/DATA/DL/BV-01: Endpoint receives downlink packets with different payload size from Gateway

Case ID: C83932935

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/45856669 -> Refer to Ble

Test Steps:

Step 1

Description:

Send 1 DL packet from AWS IoT with 1 byte payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 1, LINK: 2, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

Step 2

Description:

Send 1 DL packet from AWS IoT with 50 bytes payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 50, LINK: 2, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

Step 3

Description:

Send 1 DL packet from AWS IoT with 200 bytes payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 200, LINK: 2, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

FSK/EN/DATA/UL/BV-01: Gateway receives uplink packet with different payload sizes from Endpoint

Case ID: C83932936

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/45548595 -> Refer to Ble

Test Steps:

Step 1

Description:

Send 1 UL packet from EN with 1 byte payload:
sid send <1-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 2 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

Step 2

Description:

Send 1 UL packet from EN with 50 bytes payload:
sid send <50-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 2 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

Step 3

Description:

Send 1 UL packet from EN with 200 bytes payload:
sid send <200-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 2 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

FSK/EN/NW-SYNC/TIME/BV-01: Endpoint completes the initial time sync procedure

Case ID: C83932937

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/68272482

Test Steps:

Step 1

Description:

Reboot EN, then initialize and start FSK stack:
reboot
sid init 2
sid start
Wait for time sync

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0

Key check: Time: 0 (time synced)

FSK/EN/NW-SYNC/JOIN/BV-01: Endpoint completes the initial join procedure

Case ID: C83932938

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/49547140

Test Steps:

Step 1

Description:

Initialize and start FSK stack, then wait for time sync:
sid init 2
sid start

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0

Key check: Time: 0 (time synced)

Step 2

Description:

Wait for join response

Expected Result:

Njoin(1, 2) st 2->6

Key check: Join response received

FSK/EN/NW-SYNC/JOIN/BV-02: Endpoint keeps synchronization with the network by periodic transmission of Join request

Case ID: C83932939

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Wait 14 minutes after time sync. Do not send any UL.

Expected Result:

No UL packets sent during this period

Step 2

Description:

Wait for join response

Expected Result:

Njoin(1, 2) st 2->6

Key check: Join response received (periodic join request)

FSK/EN/SEC/UUID/BV-01: Endpoint rotates UUID immediately after time sync

Case ID: C83932941

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/45748340

Test Steps:

Step 1

Description:

Get initial TX UUID:

print txid

Expected Result:

TXID: 0x<uuid>

Key check: Note the UUID value

Step 2

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0

Key check: Time: 0 (time synced)

Step 3

Description:

Get TX UUID again:

print txid

Expected Result:

TXID: 0x<new_uuid>

Key check: UUID is different from Step 1 (rotated after time sync)

FSK/EN/SEC/UUID/BV-02: UUID rotates every 15 minutes after time sync

Case ID: C83932942

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/45722316

Test Steps:

Step 1

Description:

Get initial TX UUID:

print txid

Expected Result:

TXID: 0x<uuid>

Key check: Note the UUID value

Step 2

Description:

Monitor UUID rotation for 2 rounds.

Check UUID every 30 seconds:

print txid

Expected Result:

Round 1: UUID rotated after ~15 minutes

Round 2: UUID rotated after ~15 minutes

Key check: Each rotation within 14-16 minutes

Step 3

Description:

Verify all collected UUIDs are unique

Expected Result:

All UUIDs are unique

Key check: No duplicate UUID detected

FSK/EN/SEC/UUID/BV-03: Endpoint can exchange UL/DL during UUID rotation

Case ID: C83932943

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/45722322

Test Steps:

Step 1

Description:

Get initial TX UUID:
print txid

Expected Result:

TXID: 0x<uuid>

Step 2

Description:

Wait for 1st UUID rotation (check every 30s):
print txid

Expected Result:

TXID: 0x<new_uuid>

Key check: UUID rotated (1st rotation detected)

Step 3

Description:

Wait ~14 minutes, then send UL/DL continuously until 2nd UUID rotation:
sid send <5-byte-payload> (UL)
Send DL from AWS IoT (DL)
After 2nd rotation, send one more UL/DL to confirm

Expected Result:

UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 2 LINK_MODE: 0
DL: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 5, LINK: 2, ...
2nd UUID rotation detected
UL/DL still works after rotation

Key check: UL/DL successful before and after UUID rotation

LORA Test Cases

LORA/EN/DATA/DL/BV-01: End Node receives downlink packets with different payload size from Gateway

Case ID: C83934651

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 1 DL packet from AWS IoT with 1 byte payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 1, LINK: 4, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

Step 2

Description:

Send 1 DL packet from AWS IoT with 10 bytes payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 10, LINK: 4, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

Step 3

Description:

Send 1 DL packet from AWS IoT with 19 bytes payload

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 19, LINK: 4, LINK_MODE: <x>, ACK_REQUESTED: <x>, ACK: <x>, DUP: <x>, RSSI: <x>, SNR: <x>
Data: <payload>

Key check: Delivery ratio 100%

LORA/EN/DATA/UL/BV-01: Gateway receives uplink packets with different payload sizes from End Node

Case ID: C83934652

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 1 UL packet from EN with 1 byte payload:

sid send <1-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

Step 2

Description:

Send 1 UL packet from EN with 10 bytes payload:

sid send <10-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

Step 3

Description:

Send 1 UL packet from EN with 19 bytes payload:

sid send <19-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: Packet received on AWS

LORA/EN/SEC/UUID/BV-01: End Node rotates UUID immediately after time sync

Case ID: C83934653

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Get initial TX UUID:

print txid

Expected Result:

TXID: 0x<uuid>

Key check: Note the UUID value

Step 2

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4

EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0

Key check: Time: 0 (time synced)

Step 3

Description:

Get TX UUID again:

print txid

Expected Result:

TXID: 0x<new_uuid>

Key check: UUID is different from Step 1 (rotated after time sync)

LORA/EN/SEC/UUID/BV-02: UUID rotates every 15 minutes after time sync

Case ID: C83934654

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Get initial TX UUID:

print txid

Expected Result:

TXID: 0x<uuid>

Key check: Note the UUID value

Step 2

Description:

Monitor UUID rotation for 2 rounds.

Check UUID every 30 seconds:

print txid

Expected Result:

Round 1: UUID rotated after ~15 minutes

Round 2: UUID rotated after ~15 minutes

Key check: Each rotation within 14-16 minutes

Step 3

Description:

Verify all collected UUIDs are unique

Expected Result:

All UUIDs are unique

Key check: No duplicate UUID detected

LORA/EN/SEC/UUID/BV-03: End Node can exchange UL/DL during UUID rotation

Case ID: C83934655

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Get initial TX UUID:

print txid

Expected Result:

TXID: 0x<uuid>

Step 2

Description:

Wait for 1st UUID rotation (check every 30s):

print txid

Expected Result:

TXID: 0x<new_uuid>

Key check: UUID rotated (1st rotation detected)

Step 3

Description:

Wait ~14 minutes, then send UL/DL continuously until 2nd UUID rotation:

sid send <5-byte-payload> (UL)

Send DL from AWS IoT (DL)

After 2nd rotation, send one more UL/DL to confirm

Expected Result:

UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 0

DL: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 5, LINK: 4, ...

2nd UUID rotation detected

UL/DL still works after rotation

Key check: UL/DL successful before and after UUID rotation

LORA/EN/CONN/REG/BV-02: End Node can be deregistered from Sidewalk network

Case ID: C83934657

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Run the following command to deregister the device:

sid factory_reset

Expected Result:

CMD: ERR: 0

EVENT SID STATUS: State: 1, Reg: 1, Time: 1, Link_Mask: 0

Key check: Reg: 1 (deregistered)

LORA/EN/CONN/LP/BV-01: End Node can select Profile A and complete Join procedure

Case ID: C83934658

Executable by Customer: Yes

Preconditions:

sid option -lp_set 128 5

sid option -lp_get_l3

Test Steps:

Step 1

Description:

Initialize and start LoRa stack with Profile A:
sid init 3
sid option -lp_set 128 5
sid start

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0
Njoin(1, 2) st 2->6

Key check: Time: 0 (time synced), join completed

LORA/EN/CONN/LP/BV-02: End Node can select Profile B and complete Join procedure

Case ID: C83934659

Executable by Customer: Yes

Preconditions:

sid option -lp_set 129 0

sid option -lp_get_l3

Test Steps:

Step 1

Description:

Initialize and start LoRa stack with Profile B:
sid init 3
sid option -lp_set 129 0
sid start

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0
Njoin(1, 2) st 2->6

Key check: Time: 0 (time synced), join completed

LORA/EN/CONN/SEQ/BV-01: Uplink packets contain SEQ number

Case ID: C83934660

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 5 UL packets from EN with 10 byte payload:

sid send <10-byte-payload>

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: UL packets received on AWS

LORA/EN/CONN/SEQ/BV-02: End Node increments uplink SEQ number until UUID changes

Case ID: C83934661

Executable by Customer: Yes

Preconditions:

The seq num field should be properly incremented by one for each subsequent packet sent by DUT within a maximum of 15 minutes from the previous UUID change, then a new sequence number should be generated in the range between 0 and 16383.

Test Steps:

Step 1

Description:

Send 1 UL packet:
sid send TEST

Expected Result:

LoraTx:EnqPkt sqn:<seq> src_frmt:1 src_id:0x<txid> dst_frmt:0 dst_id:0x<gw_id> msg_id:<id>

Key check: Note the SEQ number

Step 2

Description:

Wait for UUID rotation (~15 minutes after time sync)

Expected Result:

UUID rotated (TX UUID changes)

Step 3

Description:

Send 1 UL packet after UUID rotation:
sid send TEST

Expected Result:

LoraTx:EnqPkt sqn:<new_seq> src_frmt:1 src_id:0x<new_txid> dst_frmt:0 dst_id:0x<gw_id> msg_id:<id>

Key check: New SEQ number is different from Step 1 (reset after UUID change)

LORA/EN/CONN/SEQ/BV-03: End Node accepts and decodes downlink packets with increased and decreased SEQ

Case ID: C83934662

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 3 DL packets from AWS IoT with SEQ: 1, 2, 3

Expected Result:

All 3 DL packets received and accepted by EN:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Delivery ratio 100%

Step 2

Description:

Send 3 DL packets from AWS IoT with SEQ: 6, 5, 4

Expected Result:

All 3 DL packets received and accepted by EN:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Delivery ratio 100%

LORA/EN/CONN/SEQ/BV-05: End Node rejects downlink packet with duplicated SEQ

Case ID: C83934663

Executable by Customer: Yes

Preconditions:

EN MCU <= 1.13

Test Steps:

Step 1

Description:

Send DL with SEQ=1

Then send DL with SEQ=1 again (duplicate)

Expected Result:

First: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ... (accepted)

Second: No EVENT SID RECEIVED within 20 seconds (duplicate rejected)

Key check: Duplicate rejected

Step 2

Description:

Send DL with SEQ=2

Then send DL with SEQ=2 again (duplicate)

Expected Result:

First: EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ... (accepted)

Second: No EVENT SID RECEIVED within 20 seconds (duplicate rejected)

Key check: Duplicate rejected

LORA/EN/CONN/SEQ/BV-06: End Node rejects downlink packet with duplicated SEQ range

Case ID: C83934664

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send 5 DL packets with SEQ: 1, 2, 3, 4, 5

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: All 5 packets received

Step 2

Description:

Send 5 DL packets with SEQ: 1, 2, 3, 4, 5 again (all duplicates)

Expected Result:

No EVENT SID RECEIVED for any packet

Key check: All duplicates rejected

LORA/EN/CONN/SEQ/BV-07: SEQ number can be accepted again after getting 10 other messages

Case ID: C83934665

Executable by Customer: Yes

Preconditions:

EN MCU <= 1.13
Seq=1,2,3,4,5,6,7,8,9,10,1.11,1

Test Steps:

Step 1

Description:
Send 10 DL packets from AWS IoT with SEQ: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Expected Result:
All 10 DL packets received and accepted by EN
Delivery ratio 100%

Step 2

Description:
Send 1 DL packet with SEQ=1 (duplicate)
Expected Result:
No EVENT SID RECEIVED within 20 seconds

Key check: Delivery ratio 0%

Step 3

Description:
Send 1 DL packet with SEQ=11 (new, non-duplicate)
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Delivery ratio 100%

Step 4

Description:
Send 1 DL packet with SEQ=1 again (now outside the 10-SEQ window)
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: SEQ=1 accepted again after 10 other messages

LORA/EN/CONN/SEQ/BV-08: SEQ number can be accepted again after End Node reinitialization

Case ID: C83934666

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Send DL packet with SEQ=1

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Packet accepted

Step 2

Description:

Send DL packet with SEQ=1 again (duplicate)

Expected Result:

No EVENT SID RECEIVED within 20 seconds

Key check: Duplicate rejected

Step 3

Description:

Reinitialize Sidewalk stack:
sid stop
sid deinit
sid init 3
sid start

Expected Result:

Each command returns: CMD: ERR: 0
Wait for time sync:
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Wait for join:
Nsync st ->2

Step 4

Description:

Send DL packet with SEQ=1 again (after reinit)

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: SEQ=1 accepted again after reinit

LORA/EN/CONN/SEQ/BV-09: End Node drops the packet with SEQ which is out of range

Case ID: C83934667

Executable by Customer: Yes

Test Steps:

Step 1

Description:
Send DL packet with SEQ=0 (minimum valid)
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Accepted

Step 2

Description:
Send DL packet with SEQ=16383 (maximum valid)
Expected Result:
EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: <len>, LINK: 4, ...

Key check: Accepted

Step 3

Description:
Send DL packet with SEQ=-1 (out of range)
Expected Result:
No EVENT SID RECEIVED within 20 seconds

Key check: Rejected (out of range)

Step 4

Description:
Send DL packet with SEQ=16384 (out of range)
Expected Result:
No EVENT SID RECEIVED within 20 seconds

Key check: Rejected (out of range)

LORA/EN/NW-SYNC/JOIN/BV-02: End Node keeps synchronization with the network by periodic transmission of Join request

Case ID: C83934668

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Initialize and start LoRa stack:
sid init 3
sid start
Wait for time sync and join

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6

Key check: Time synced and joined

Step 2

Description:

Wait 14 minutes after time sync. Do not send any UL.

Expected Result:

No UL packets sent during this period

Step 3

Description:

Wait for join response

Expected Result:

Njoin(1, 2) st 2->6

Key check: Join response received (periodic join request)

LORA/EN/NW-SYNC/JOIN/BV-03: End Node keeps synchronization with the network by embedded sync request

Case ID: C83934669

Executable by Customer: Yes

Test Steps:

Step 1

Description:

EN time-sync and joined

Step 2

Description:

Wait 150 seconds, EN send a UL packet.

Expected Result:

EN receive NTFY_JOIN_REQ

Step 3

Description:

Wait 170 seconds

Expected Result:

EN does not send SET_JOIN_REQ in this period.

LORA/EN/NW-SYNC/TIME/BV-01: End Node completes the initial time sync procedure

Case ID: C83934670

Executable by Customer: Yes

Test Steps:

Step 1

Description:

Reboot EN, then initialize and start LoRa stack:
reboot
sid init 3
sid start
Wait for time sync

Expected Result:

Each command returns: CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0

Key check: Time: 0 (time synced)

MLM Test Cases

MULTI-LINK/EP/BLE-AUTO-CONNECT/UL/BI/01: Send UL without auto-connect, DUT is BLE disconnected (Link Mask 0)

Case ID: C103619231

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027002

Test Steps:

Step 1

Description:

Initialize and start BLE stack:
sid init 1
sid start

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: EN initialized with BLE, Link_Mask: 0

Step 2

Description:

Wait for BLE disconnected:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE is disconnected

Step 3

Description:

Send UL without auto-connect:
sid send -l 1 TEST

Expected Result:

CMD: ERR: -40
Key check: UL rejected, no connection available

MULTI-LINK/EP/BLE-AUTO-CONNECT/UL/BV/02: Send UL with auto-connect, ack is true, retry is 3, TTL is 60

Case ID: C103619232

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027034

Test Steps:

Step 1

Description:

Initialize and start BLE stack:
sid init 1
sid start

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: EN initialized with BLE

Step 2

Description:

Wait for BLE disconnected:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE is disconnected

Step 3

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 1 1 5 30

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 4

Description:

Send UL with auto-connect (ack=true, retry=3, TTL=60):
sid send -l 1 -a 1 3 60 TEST

Expected Result:

ACM(0) st 0->1
CMD: ERR: 0 TYPE: 2 ID: 1
ACM: Connect 1 for 30 sec err 0
BLE Connected
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0
Key check: BLE auto-connected, UL sent successfully, SID ERR: 0

MULTI-LINK/EP/FSK-AUTO-CONNECT/UL/BI/01: SendUL without auto-connect, DUT is FSK disconnected (Link Mask0)

Case ID: C103619233

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84103974

Test Steps:

Step 1

Description:

Initialize and start FSK stack, wait for time sync:
sid init 2
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 1, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN time-synced via FSK

Step 2

Description:

Send UL without auto-connect:
sid send -l 2 TEST

Expected Result:

CMD: ERR: -40
Key check: UL rejected, no connection available

MULTI-LINK/EP/FSK-AUTO-CONNECT/UL/BV/02: Send UL with auto-connect, ack is true, retry is 3, TTL is 60

Case ID: C103619234

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027170

Test Steps:

Step 1

Description:

Initialize and start FSK stack, wait for time sync:
sid init 2
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 1, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN time-synced via FSK

Step 2

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 2 1 20 90

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 3

Description:

Send UL with auto-connect (ack=true, retry=3, TTL=60):
sid send -l 2 -a 1 3 60 TEST

Expected Result:

ACM(0) st 0->1
CMD: ERR: 0 TYPE: 2 ID: 1
ACM: Connect 2 for 90 sec err 0
Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 2 LINK_MODE: 0
Key check: FSK auto-connected, UL sent successfully, SID ERR: 0

MULTI-LINK/EP/LoRa-AUTO-CONNECT/UL/BI/01: Send UL without auto-connect, DUT is LoRa disconnected (Link Mask 0)

Case ID: C103619235

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104344

Test Steps:

Step 1

Description:

Initialize and start LoRa stack, wait for time sync:
sid init 3
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 1, Link_Mask: 4
Njoin(1, 4) st 2->6
Key check: EN time-synced via LoRa

Step 2

Description:

Send UL without auto-connect:
sid send -l 3 TEST

Expected Result:

CMD: ERR: -40
Key check: UL rejected, no connection available

MULTI-LINK/EP/LoRa-AUTO-CONNECT/UL/BV/02: Send UL with auto-connect, ack is true, retry is 3, TTL is 60

Case ID: C103619236

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027051

Test Steps:

Step 1

Description:

Initialize and start LoRa stack, wait for time sync:
sid init 3
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 1, Link_Mask: 4
Njoin(1, 4) st 2->6
Key check: EN time-synced via LoRa

Step 2

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 3

Description:

Send UL with auto-connect (ack=true, retry=3, TTL=60):
sid send -l 3 -a 1 3 60 TEST

Expected Result:

ACM(0) st 0->1
CMD: ERR: 0 TYPE: 2 ID: 1
ACM: Connect 4 for 120 sec err 0
Njoin(1, 4) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 4 LINK_MODE: 0
Key check: LoRa auto-connected, UL sent successfully, SID ERR: 0

MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/01: Send UL with auto-conenct when BLE link priority is higher than LoRa, EN Link Mask is 0, Specify BLE+LoRa link

Case ID: C103619237

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027057

Test Steps:

Step 1

Description:

Disable GW 900MHz (FSK + LoRa):
adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled (FSK and LoRa unavailable)

Step 2

Description:

Initialize and start BLE+LoRa stack, set auto-connect policy:
sid init 4
sid start
sid option -m 1
sid option -c 1 1 10 30
sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0
Key check: EN initialized with BLE+LoRa, auto-connect policy set

Step 3

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
Key check: EN time-synced

Step 4

Description:

Wait for Link_Mask = 0:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE disconnected, Link_Mask: 0

Step 5

Description:

Send UL with BLE+LoRa link specified:

sid send -l 4 TEST

Expected Result:

ACM: Connect 1 for 30 sec err 0

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: BLE auto-connected (higher priority than LoRa), UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/02: Send UL with auto-conenct when BLE link priority is higher than LoRa, EN Link Mask is 0, Specify BLE+LoRa link, disable BLE on GW

Case ID: C103619238

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027058

Test Steps:

Step 1

Description:

Disable GW BLE:

adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE disabled

Step 2

Description:

Initialize and start BLE+LoRa stack, set auto-connect policy:

sid init 4

sid start

sid option -m 1

sid option -c 1 1 10 30

sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0

Key check: EN initialized with BLE+LoRa, auto-connect policy set

Step 3

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0

Key check: EN time-synced

Step 4

Description:

Send UL with BLE+LoRa link specified:

sid send -l 4 TEST

Expected Result:

ACM: Connect 1 for 30 sec err 0

(BLE connect fails)

ACM: Connect 4 for 120 sec err 0

Njoin(1, 4) st 2->6

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 4 LINK_MODE: 0
Key check: BLE failed, LoRa auto-connected, UL sent via LoRa, SID ERR: 0

MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/03: Send UL with auto-conenct when BLE link priority is higher than LoRa, EN Link Mask is 0, disables BLE & LoRa on GW

Case ID: C103619239

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81027116

Test Steps:

Step 1

Description:

Disable GW 900MHz (FSK + LoRa):
adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled (FSK and LoRa unavailable)

Step 2

Description:

Initialize and start BLE+LoRa stack, set auto-connect policy:
sid init 4
sid start
sid option -m 1
sid option -c 1 1 10 30
sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0
Key check: EN initialized with BLE+LoRa, auto-connect policy set

Step 3

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
Key check: EN time-synced

Step 4

Description:

Wait for Link_Mask = 0:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE disconnected, Link_Mask: 0

Step 5

Description:

Disable GW BLE:

adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE disabled

Step 6

Description:

Send UL with auto-connect:

sid send -l 4 -a 1 1 60 TEST

Expected Result:

ACM: Connect 1 for 30 sec err 0

(BLE connect fails)

ACM: Connect 4 for 120 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 4 LINK_MODE: 0

Key check: Both BLE and LoRa failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/BLE-LoRa-AUTO-CONNECT/UL/BV/04: Send UL with auto-conenct when BLE link priority is higher than LoRa, EN Link Mask is 0, Specify FSK link

Case ID: C103619240

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/86491534

Test Steps:

Step 1

Description:

Disable GW 900MHz (FSK + LoRa):
adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled (FSK and LoRa unavailable)

Step 2

Description:

Initialize and start BLE+LoRa stack, set auto-connect policy:
sid init 4
sid start
sid option -m 1
sid option -c 1 1 10 30
sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0
Key check: EN initialized with BLE+LoRa, auto-connect policy set

Step 3

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
Key check: EN time-synced

Step 4

Description:

Wait for Link_Mask = 0:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: BLE disconnected, Link_Mask: 0

Step 5

Description:

Send UL specifying FSK link (not initialized):

sid send -l 2 TEST

Expected Result:

CMD: ERR: -40

Key check: UL rejected, FSK link not available in BLE+LoRa stack

MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/01: UL sent through BLE when no connection

Case ID: C103619241

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080904

Test Steps:

Step 1

Description:

Initialize and start BLE+FSK+LoRa stack:
sid init 7
sid start

Expected Result:

CMD: ERR: 0
Key check: EN initialized with BLE+FSK+LoRa

Step 2

Description:

Disable GW 900MHz (FSK + LoRa):
adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled (FSK and LoRa unavailable)

Step 3

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 1 1 10 30
sid option -c 2 1 20 90
sid option -c 3 1 30 120

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 4

Description:

Wait for time sync

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
Key check: EN time-synced

Step 5

Description:

Wait for Link_Mask = 0:
sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
Key check: all links disconnected

Step 6

Description:

Send UL (no link specified):
sid send TEST

Expected Result:

ACM: Connect 1 for 30 sec err 0
BLE Connected
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0
Key check: BLE auto-connected (highest priority), UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/02:
UL sent through FSK when no connection

Case ID: C103619242

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080905

Test Steps:

Step 1

Description:
Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7
Expected Result:
CMD: ERR: 0
Key check: EN initialized

Step 2

Description:
Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp
Expected Result:
GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:
Start stack, wait for time sync and FSK join:
sid start
Expected Result:
Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:
Set auto-connect policy:
sid option -m 1
sid option -c 1 1 10 30
sid option -c 2 1 20 90
sid option -c 3 1 30 20
Expected Result:
CMD: ERR: 0
Key check: auto-connect policy set

Step 5

Description:

Force Link_Mask = 0 by sending LoRa UL with short TTL:

sid send -l 3 -a 1 1 10 TEST

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 6

Description:

Wait for auto-connect to terminate

Expected Result:

ACM(<n>) st <n>->0

Key check: previous auto-connect terminated

Step 7

Description:

Send UL with auto-connect:

sid send -a 1 1 120 TEST

Expected Result:

ACM: Connect 1 for 30 sec err 0

(BLE connect fails)

ACM: Connect 2 for 90 sec err 0

Njoin(1, 2) st 2->6

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 2 LINK_MODE: 0

Key check: BLE failed, FSK auto-connected, UL sent via FSK, SID ERR: 0

MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/03:
UL sent through LoRa when no connection

Case ID: C103619243

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080906

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 1 1 10 10
sid option -c 2 1 20 30
sid option -c 3 1 30 20

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 5

Description:

Force Link_Mask = 0 by sending LoRa UL with short TTL:

sid send -l 3 -a 1 1 10 TEST

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 6

Description:

Wait for auto-connect to terminate

Expected Result:

ACM(<n>) st <n>->0

Key check: previous auto-connect terminated

Step 7

Description:

Enable GW LoRa only (disable FSK):

adb shell halo_cli -c 0 -i 875 0002

sid option -c 3 1 30 90

Expected Result:

GW LoRa enabled, FSK disabled

Key check: Only LoRa available on GW

Step 8

Description:

Send UL with auto-connect:

sid send -a 1 1 120 TEST

Expected Result:

ACM: Connect 1 for 10 sec err 0

(BLE connect fails)

ACM: Connect 2 for 30 sec err 0

(FSK connect fails)

ACM: Connect 4 for 90 sec err 0

Njoin(1, 4) st 2->6

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 4 LINK_MODE: 0

Key check: BLE and FSK failed, LoRa auto-connected, UL sent via LoRa, SID ERR: 0

MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/04:
UL not send when no connection and BLE/FSK/LoRa unavailable

Case ID: C103619244

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080907

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 1 1 10 10
sid option -c 2 1 20 30
sid option -c 3 1 30 20

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 5

Description:

Force Link_Mask = 0 by sending LoRa UL with short TTL:

sid send -l 3 -a 1 1 10 TEST

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 6

Description:

Wait for auto-connect to terminate

Expected Result:

ACM(<n>) st <n>->0

Key check: previous auto-connect terminated

Step 7

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 8

Description:

Send UL with auto-connect:

sid send -a 1 1 50 TEST

Expected Result:

ACM: Connect 1 for 10 sec err 0

(BLE connect fails)

ACM: Connect 2 for 30 sec err 0

(FSK connect fails)

ACM: Connect 4 for 20 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/BLE-FSK-LoRa-AUTO-CONNECT/UL/BV/05: UL sent through BLE when EN's Link Mask is 1

Case ID: C103619245

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104354

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW 900MHz (FSK + LoRa):
adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled
Key check: Only BLE available on GW

Step 3

Description:

Start stack, wait for time sync:
sid start

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 1, Link_Mask: 0
Key check: EN time-synced

Step 4

Description:

Set auto-connect policy:
sid option -m 1
sid option -c 1 1 10 10
sid option -c 2 1 20 30
sid option -c 3 1 30 20

Expected Result:

CMD: ERR: 0
Key check: auto-connect policy set

Step 5

Description:

Wait for Link_Mask = 0:

sid last_status

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: all links disconnected

Step 6

Description:

Establish BLE connection via conn_req:

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/MLM-ACTIVE/UL/BV/01: UL sent throughBLE when no connection

Case ID: C103619246

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080927

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-ACTIVE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-ACTIVE/UL/BV/02: UL sent throughFSK when no connection

Case ID: C103619247

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080928

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-ACTIVE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-ACTIVE/UL/BV/03: UL sent throughLoRa when no connection

Case ID: C103619248

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080929
Update add step 5:
QA disables BLE on GW
$ adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-ACTIVE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-ACTIVE/UL/BV/04: UL not sendwhen no connection and BLE/FSK/LoRa unavailable

Case ID: C103619249

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080930

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 0

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

MLM: Connect 1 for 30 sec err 0

(BLE connect fails)

MLM: Connect 2 for 60 sec err 0

(FSK connect fails)

MLM: Connect 4 for 60 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/MLM-ACTIVE/UL/BV/05: UL sent throughBLE when EN's Link Mask is 1

Case ID: C103619250

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104355

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 0

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Enable GW BLE and establish BLE connection:

adb shell start trackerd

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 8

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/MLM-POWER/UL/BV/01: UL sent through BLE when no connection

Case ID: C103619251

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080944

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-POWER/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-POWER/UL/BV/02: UL sent through FSK when no connection

Case ID: C103619252

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080945

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-POWER/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-POWER/UL/BV/03: UL sent through LoRa when no connection

Case ID: C103619253

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080946

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-POWER/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-POWER/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable

Case ID: C103619254

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080947

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 1

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

MLM: Connect 1 for 30 sec err 0

(BLE connect fails)

MLM: Connect 2 for 60 sec err 0

(FSK connect fails)

MLM: Connect 4 for 60 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/MLM-POWER/UL/BV/05: UL sent through BLE when EN's Link Mask is 1

Case ID: C103619255

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104356

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 1

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Enable GW BLE and establish BLE connection:

adb shell start trackerd

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 8

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/01: UL sent through BLE when no connection

Case ID: C103619256

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080964

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/02: UL sent through FSK when no connection

Case ID: C103619257

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080965

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/03: UL sent through LoRa when no connection

Case ID: C103619258

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080966

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable

Case ID: C103619259

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81080967

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 2

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

MLM: Connect 1 for 30 sec err 0

(BLE connect fails)

MLM: Connect 2 for 60 sec err 0

(FSK connect fails)

MLM: Connect 4 for 30 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/MLM-PERFORMANCE/UL/BV/05: UL sent through BLE when EN's Link Mask is 1

Case ID: C103619260

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104357

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 2

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Enable GW BLE and establish BLE connection:

adb shell start trackerd

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 8

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/MLM-LATENCY/UL/BV/01: UL sent through BLE when no connection

Case ID: C103619261

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081000

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-LATENCY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-LATENCY/UL/BV/02: UL sent through FSK when no connection

Case ID: C103619262

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081001

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-LATENCY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-LATENCY/UL/BV/03: UL sent through LoRa when no connection

Case ID: C103619263

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081002

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-LATENCY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-LATENCY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable

Case ID: C103619264

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081003

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 3

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

MLM: Connect 1 for 15 sec err 0

(BLE connect fails)

MLM: Connect 2 for 30 sec err 0

(FSK connect fails)

MLM: Connect 4 for 60 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/MLM-LATENCY/UL/BV/05: UL sent through BLE when EN's Link Mask is 1

Case ID: C103619265

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104358

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 3

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Enable GW BLE and establish BLE connection:

adb shell start trackerd

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 8

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/01: UL sent through BLE when no connection

Case ID: C103619266

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081674

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/02: UL sent through FSK when no connection

Case ID: C103619267

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081675

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/03: UL sent through LoRa when no connection

Case ID: C103619268

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081676

Test Steps:

Step 1

Description:

Note: This scenario is covered by MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable.

No separate execution required.

Expected Result:

N/A

MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/04: UL not send when no connection and BLE/FSK/LoRa unavailable

Case ID: C103619269

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/81081677

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 4

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Key check: All links unavailable on GW

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Send UL:

sid send TEST

Expected Result:

MLM: Connect 1 for 15 sec err 0

(BLE connect fails)

MLM: Connect 2 for 60 sec err 0

(FSK connect fails)

MLM: Connect 4 for 120 sec err 0

(LoRa connect fails)

EVENT SID SEND STATUS: SID ERR: -2, TYPE: 2, ID: 1, LINK: 7 LINK_MODE: 0

Key check: All links failed, UL not sent, SID ERR: -2

MULTI-LINK/EP/MLM-RELIABILITY/UL/BV/05: UL sent through BLE when EN's Link Mask is 1

Case ID: C103619270

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84104359

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack (do not start yet):
sid init 7

Expected Result:

CMD: ERR: 0
Key check: EN initialized

Step 2

Description:

Disable GW BLE and LoRa (FSK only mode):
adb shell halo_cli -c 0 -i 875 0001
adb shell stop trackerd; adb shell stop btmanagerd; adb shell stop halo-core-swp; adb shell start halo-core-swp

Expected Result:

GW BLE and LoRa disabled
Key check: Only FSK available on GW

Step 3

Description:

Start stack, wait for time sync and FSK join:
sid start

Expected Result:

Njoin(1, 2) st 2->6
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Key check: EN joined via FSK

Step 4

Description:

Set MLM policy:
sid option -m 2
sid option -ml 4

Expected Result:

CMD: ERR: 0
Key check: MLM policy set

Step 5

Description:

Disable GW 900MHz (FSK + LoRa):

adb shell halo_cli set_protocol_on off

Expected Result:

GW 900MHz disabled

Step 6

Description:

Wait for Link_Mask = 0

Expected Result:

EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 0

Key check: Link_Mask = 0

Step 7

Description:

Enable GW BLE and establish BLE connection:

adb shell start trackerd

sid conn_req 1

Expected Result:

BLE Connected

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connected, Link_Mask: 1

Step 8

Description:

Send UL:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: 1, LINK: 1 LINK_MODE: 0

Key check: UL sent via BLE, SID ERR: 0

Multi-Radio Test Cases

MULTI/EN/API/INIT/BV-01: Switch between 900MHz and BLE stack multiple times

Case ID: C103619216

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/68652428

Test Steps:

Step 1

Description:

Initialize LoRa stack and wait for time sync:
sid deinit
sid init 3
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4

Key check: Time sync received (Time: 0)

Step 2

Description:

Switch to BLE stack and wait for time sync:
sid deinit
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

Key check: Time sync received (Time: 0)

Step 3

Description:

Switch back to LoRa and wait for time sync:
sid deinit
sid init 3
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4

Key check: Time sync received (Time: 0)

Step 4

Description:

Switch back to BLE and wait for time sync:
sid deinit

sid init 1

sid start

Expected Result:

CMD: ERR: 0

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

Key check: Time sync received (Time: 0)

MULTI/EN/CONN/DL/BV-01: End Node is able to receive a downlink packet using BLE

Case ID: C103619222

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382381

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Send UL packet through BLE:
sid send -l 1 TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 1

Key check: LINK: 1 (BLE)

Step 3

Description:

Send DL packet from AWS IoT

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 4, LINK: 1, LINK_MODE: 1, ACK_REQUESTED: 0, ACK: 0, DUP: 0, RSSI: <rssi>, SNR: <snr>
Data: 54455354

Key check: LINK: 1 (BLE), data received

MULTI/EN/CONN/DL/BV-02: End Node is able to receive a downlink packet using LoRa

Case ID: C103619223

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382382

Test Steps:

Step 1

Description:

Verify BLE is disconnected:
sid last_status

Expected Result:

EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected)

Step 2

Description:

Send UL packet through LoRa:
sid send -l 3 TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 1

Key check: LINK: 4 (LoRa)

Step 3

Description:

Send DL packet from AWS IoT

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 4, LINK: 4, LINK_MODE: 1, ACK_REQUESTED: 0, ACK: 0, DUP: 0, RSSI: <rssi>, SNR: <snr>
Data: 54455354

Key check: LINK: 4 (LoRa), data received

MULTI/EN/CONN/DL/BV-03: End Node is able to receive a downlink packet using LoRa, while BLE connection with Gateway is established

Case ID: C103619224

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382383

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 1

Key check: BLE: 1 (connected), LoRa: 1 (active)

Step 2

Description:

While BLE connection is active, send UL packet through LoRa:
sid send -l 3 TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 1

Key check: LINK: 4 (LoRa)

Step 3

Description:

Send DL packet from AWS IoT

Expected Result:

EVENT SID RECEIVED: TYPE: <type>, ID: <id>, LEN: 4, LINK: 4, LINK_MODE: 1, ACK_REQUESTED: 0, ACK: 0, DUP: 0, RSSI: <rssi>, SNR: <snr>
Data: 54455354

Key check: LINK: 4 (LoRa), data received

MULTI/EN/CONN/SEND-LINK/BV-01: UL performed through BLE, when send_link 0 and active BT connection

Case ID: C103619217

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382376

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Send UL with send_link=0 (any):
sid send TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 1

Key check: LINK: 1 (BLE auto-selected when BLE connected)

MULTI/EN/CONN/SEND-LINK/BV-02: UL performed through BLE, when send_link 1 and active BT connection

Case ID: C103619218

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382377

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Send UL with send_link=1 (BLE):
sid send -l 1 TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1 LINK_MODE: 1

Key check: LINK: 1 (BLE)

MULTI/EN/CONN/SEND-LINK/BV-03: UL performed through LoRa, when send_link 3 and active BT connection

Case ID: C103619219

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382378

Test Steps:

Step 1

Description:

Verify BLE is disconnected, then set connection request:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 1

Key check: BLE: 1 (connected)

Step 2

Description:

Send UL with send_link=3 (LoRa):
sid send -l 3 TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>
EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 1

Key check: LINK: 4 (LoRa, forced even with BLE connected)

MULTI/EN/CONN/SEND-LINK/BV-04: UL performed through LoRa, when send_link 0 and terminated BT connection

Case ID: C103619220

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382379

Test Steps:

Step 1

Description:

Verify BLE is disconnected:

sid last_status

Expected Result:

EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected)

Step 2

Description:

Send UL with send_link=0 (any):

sid send TEST

Expected Result:

CMD: ERR: 0 TYPE: 2 ID: <id>

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4 LINK_MODE: 1

Key check: LINK: 4 (LoRa auto-selected when BLE disconnected)

MULTI/EN/CONN/SEND-LINK/BV-05: Uplink not performed, when send_link 1 and terminated BT connection

Case ID: C103619221

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69382380

Test Steps:

Step 1

Description:

Verify BLE is disconnected:
sid last_status

Expected Result:

EVENT SID STATUS LINK MODE: LORA: 1, FSK: 0, BLE: 0

Key check: BLE: 0 (disconnected)

Step 2

Description:

Send UL with send_link=1 (BLE) while BLE is disconnected:
sid send -l 1 TEST

Expected Result:

No EVENT SID SEND STATUS received

Key check: UL not sent (BLE link not available)

File Transfer Test Cases

BLE/EP/API/FILETRANSFER/BV/01: Initialize the feature and start receiving a file

Case ID: C103619294

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823766

Test Steps:

Step 1

Description:

Initialize BLE stack, start, and initialize SBDT:
sid init 1
sid start
sbdt init

Expected Result:

SBDT: [NO LOAD]
dup drop: 29000000 (3)
MET: SRC: 13
CMD: ERR: 0
Key check: SBDT initialized, no pending transfer

Step 2

Description:

Create a FUOTA task on AWS:
File: fuota_test_file_2048bytes.txt, block_size: 1024
Note the task_id from response

Expected Result:

FUOTA task created successfully
Response: { "Id": "<task_id>" }
Key check: task_id obtained

Step 3

Description:

Associate EN wireless device ID with the FUOTA task:
wireless_device_id: <wireless_device_id>
task_id: <task_id>

Expected Result:

Response: 2xx
Key check: EN associated with FUOTA task

Step 4

Description:

Schedule start time for the FUOTA task (at least 5 minutes from now):
task_id: <task_id>
start_time:

Expected Result:

Response: 2xx
Key check: FUOTA task scheduled

Step 5

Description:

Wait for BLE disconnected, establish BLE connection and send UL:
sid conn_req 1
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1
Key check: UL sent via BLE

Step 6

Description:

Wait for file transfer to complete (timeout: 600s):
Monitor EN log

Expected Result:

DEC: M:15 [3:D:2]
Key check: File transfer completed

Step 7

Description:

Check file transfer task status on cloud:
task_id: <task_id>

Expected Result:

Task status: FuotaDone
Wireless device status: Successful
Key check: Cloud confirms transfer complete

BLE/EP/API/FILETRANSFER/BV/02: Cancel ongoing transfer

Case ID: C103619295

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823788

Test Steps:

Step 1

Description:

Initialize BLE stack, start, and initialize SBDT:
sid init 1
sid start
sbdt init

Expected Result:

SBDT: [NO LOAD]
CMD: ERR: 0
Key check: SBDT initialized

Step 2

Description:

Create a FUOTA task on AWS:
File: fuota_test_file_10240bytes.txt, block_size: 1024
Note the task_id from response

Expected Result:

FUOTA task created successfully
Response: { "Id": "" }
Key check: task_id obtained

Step 3

Description:

Associate EN wireless device ID with the FUOTA task:
wireless_device_id:
task_id:

Expected Result:

Response: 2xx
Key check: EN associated with FUOTA task

Step 4

Description:

Schedule start time for the FUOTA task (at least 5 minutes from now):
task_id:
start_time:

Expected Result:

Response: 2xx
Key check: FUOTA task scheduled

Step 5

Description:

Wait for BLE disconnected, establish BLE connection and send UL:

sid conn_req 1

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 1

Key check: UL sent via BLE

Step 6

Description:

Wait for file transfer to start (blocks transferring):

Monitor EN log

Expected Result:

DEC: M:15 [3:5:1]

Key check: File transfer started

Step 7

Description:

Cancel the file transfer:

sbdt cancel 0 0x00

Expected Result:

CMD: ERR: 0

Key check: File transfer cancelled

The cancel is not immediate, there might be chunks that are in flight either from the gateway or the cloud and device will continue to receive them, but the device will tag them as errors and send a reponse [3:5:3](which is by design), additionally the sdk tells the user that there was an error with the on_sbdt_error trigger.

Step 8

Description:

Check file transfer task status on cloud:

task_id:

Expected Result:

Task status: FuotaDone

Wireless device status: Successful

Key check: Cloud confirms task complete

BLE/EP/API/FILETRANSFER/BV/03: Print the progress of the transfer and current file offset

Case ID: C103619296

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84102495

Test Steps:

Step 1

Description:

Initialize BLE stack, start, and initialize SBDT:
sid init 1
sid start
sbdt init

Expected Result:

SBDT: [NO LOAD]
CMD: ERR: 0
Key check: SBDT initialized

Step 2

Description:

Create a FUOTA task on AWS:
File: fuota_test_file_10240bytes.txt, block_size: 1024
Note the task_id from response

Expected Result:

FUOTA task created successfully
Response: { "Id": "" }
Key check: task_id obtained

Step 3

Description:

Associate EN wireless device ID with the FUOTA task:
wireless_device_id:
task_id:

Expected Result:

Response: 2xx
Key check: EN associated with FUOTA task

Step 4

Description:

Schedule start time for the FUOTA task (at least 5 minutes from now):
task_id:
start_time:

Expected Result:

Response: 2xx
Key check: FUOTA task scheduled

Step 5

Description:

Wait for BLE disconnected, establish BLE connection and send UL:
sid conn_req 1
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 1
Key check: UL sent via BLE

Step 6

Description:

Wait for file transfer to start (blocks transferring):
Monitor EN log

Expected Result:

DEC: M:15 [3:5:1]
Key check: File transfer started

Step 7

Description:

Print file transfer progress:
sbdt stats 0

Expected Result:

FILE_OFFSET: 2048, FILE_PROGRESS: 49
Key check: Progress and file offset retrieved successfully

BLE/EP/API/FILETRANSFER/BV/04: Print various parameters of the transfer

Case ID: C103619297

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84102497

Test Steps:

Step 1

Description:

Initialize BLE stack, start, and initialize SBDT:
sid init 1
sid start
sbdt init

Expected Result:

SBDT: [NO LOAD]
CMD: ERR: 0
Key check: SBDT initialized

Step 2

Description:

Create a FUOTA task on AWS:
File: fuota_test_file_10240bytes.txt, block_size: 1024
Note the task_id from response

Expected Result:

FUOTA task created successfully
Response: { "Id": "" }
Key check: task_id obtained

Step 3

Description:

Associate EN wireless device ID with the FUOTA task:
wireless_device_id:
task_id:

Expected Result:

Response: 2xx
Key check: EN associated with FUOTA task

Step 4

Description:

Schedule start time for the FUOTA task (at least 5 minutes from now):
task_id:
start_time:

Expected Result:

Response: 2xx
Key check: FUOTA task scheduled

Step 5

Description:

Wait for BLE disconnected, establish BLE connection and send UL:

sid conn_req 1

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 1

Key check: UL sent via BLE

Step 6

Description:

Wait for file transfer to start (blocks transferring):

Monitor EN log

Expected Result:

DEC: M:15 [3:5:1]

Key check: File transfer started

Step 7

Description:

Print file transfer parameters:

sbdt params 0

Expected Result:

FILE_ID: 0, FILE_SIZE: 4098, BLOCK_SIZE: 1024, MIN_SCRATCH_SPACE: 2712, FILE_DESCRIPTOR_SIZE: 0, SCRATCH_BUFFER_SIZE: 6144

Key check: Parameters retrieved successfully

BLE/EP/API/FILETRANSFER/RECOVERY/BI/01: Recover transfer after EP powered off and back on

Case ID: C103619298

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823809

Test Steps:

Step 1

Description:

Initialize BLE stack, start, and initialize SBDT:
sid init 1
sid start
sbdt init

Expected Result:

SBDT: [NO LOAD]
CMD: ERR: 0
Key check: SBDT initialized

Step 2

Description:

Create a FUOTA task on AWS:
File: fuota_test_file_10240bytes.txt, block_size: 1024
Note the task_id from response

Expected Result:

FUOTA task created successfully
Response: { "Id": "<task_id>" }
Key check: task_id obtained

Step 3

Description:

Associate EN wireless device ID with the FUOTA task:
wireless_device_id: <wireless_device_id>
task_id: <task_id>

Expected Result:

Response: 2xx
Key check: EN associated with FUOTA task

Step 4

Description:

Schedule start time for the FUOTA task (at least 5 minutes from now):
task_id: <task_id>
start_time:

Expected Result:

Response: 2xx
Key check: FUOTA task scheduled

Step 5

Description:

Wait for BLE disconnected, establish BLE connection and send UL:
sid conn_req 1
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 1
Key check: UL sent via BLE

Step 6

Description:

Wait for file transfer to start (blocks transferring):
Monitor EN log

Expected Result:

DEC: M:15 [3:5:1]
Key check: File transfer started

Step 7

Description:

Power off EN and re-initialize:
sbdt deinit
sid deinit
Reboot EN
Wait 10 seconds to simulate EN offline
sid init 1
sid start
sbdt init

Expected Result:

SBDT: LOAD [...]
CMD: ERR: 0
Key check: EN re-initialized, pending transfer detected (LOAD)

Step 8

Description:

Wait for file transfer to complete (timeout: 600s):
Monitor EN log

Expected Result:

DEC: M:15 [3:D:2]
Key check: File transfer resumed and completed

Step 9

Description:

Check file transfer task status on cloud:
task_id: <task_id>

Expected Result:

Task status: FuotaDone
Wireless device status: Successful
Key check: Cloud confirms transfer complete

BLE/EP/API/FILETRANSFER/RECOVERY/BI/02: Recover transfer after EP was out of range from GW

Case ID: C103619299

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823813

Test Steps:

Step 1

Description:

EN 'sid init 1'
EN 'sid start'
EN 'sbdt init'

Expected Result:

EN is started and time synced

Step 2

Description:

Call CreateFuotaTask API Call AssociateFuotaTask API
Call ScheduleFuotaTask API (Gamma 3 minutes later, Prod 5 minutes later)

Step 3

Description:

After data blocks sending, moving EN out of GW range

Step 4

Description:

After 1 minutes, moving EN back to GW range

Expected Result:

- The file transfer should be recovered, Sidewalk Cloud should resend the rest block of data then continue file transfer
- Device should receive file that is not corrupted
- Expected to see Finalize Response which indicates file transfer has been finished successfully

[Finalize Response]
<info> app: DEC: M:15 [3:8:1]
<info> app: ENC: M:15 [3:8:3]
<info> app: DEC: M:15 [3:D:2]

BLE/EP/API/FILETRANSFER/RECOVERY/BI/03: Recover transfer after GW that was using to send packets switched to another GW

Case ID: C103619300

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823814

Test Steps:

Step 1

Description:

EN 'sid init 1'
EN 'sid start'
EN 'sbdt init'

Expected Result:

EN is started and time synced

Step 2

Description:

Call CreateFuotaTask API Call AssociateFuotaTask API Call ScheduleFuotaTask API (Gamma 3 minutes later, Prod 5 minutes later)

Step 3

Description:

After data blocks sending, turn off GW1 and turn on GW2

Expected Result:

- The file transfer should be recovered, Sidewalk Cloud should resend the rest block of data then continue file transfer
- Device should receive file that is not corrupted
- Expected to see Finalize Response which indicates file transfer has been finished successfully

[Finalize Response]

<info> app: DEC: M:15 [3:8:1]
<info> app: ENC: M:15 [3:8:3]
<info> app: DEC: M:15 [3:D:2]

BLE/EP/API/FILETRANSFER/RECOVERY/BI/04: Recover transfer after GW that was using to send packets being opt- out

Case ID: C103619301

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823817

No test steps available.

BLE/EP/API/FILETRANSFER/RECOVERY/BI/05: Recover transfer after GW that was using to send packets powered off for a while and back on

Case ID: C103619302

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/80823806

Test Steps:

Step 1

Description:

- Wait for the device to start receiving parts of the file

- Turn off GW for 5 minutes then turn it on again

Expected Result:

- The file transfer should be recovered, Sidewalk Cloud should resend the rest block of data then continue file transfer

- Device should receive file that is not corrupted

- Expected to see Finalize Response which indicates file transfer has been finished successfully

[Finalize Response]

<info> app: DEC: M:15 [3:8:1]
<info> app: ENC: M:15 [3:8:3]
<info> app: DEC: M:15 [3:D:2]

EP Metrics Test Cases

CMN/EP/EPMETRICS/METRICS/BV/01: Verify Endpoint has corresponding metrics but not invalid metrics enabled

Case ID: C103619324

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84025466

Test Steps:

Step 1

Description:

Initialize stack and start:
- BLE only: sid init 1 / sid start
- FSK only: sid init 2 / sid start
- LoRa only: sid init 3 / sid start
- BLE+LoRa: sid init 4 / sid start
- BLE+FSK: sid init 5 / sid start
- FSK+LoRa: sid init 6 / sid start
- BLE+FSK+LoRa: sid init 7 / sid start

Expected Result:

CMD: ERR: 0

Step 2

Description:

For non-LoRa stacks (BLE, FSK, BLE+LoRa, BLE+FSK, FSK+LoRa, BLE+FSK+LoRa), initialize SBDT:
sbdt init

Expected Result:

CMD: ERR: 0

Step 3

Description:

Verify expected metrics categories exist, unexpected ones do not:
print metrics 0x36

Expected Result:

Integration Type	Present	Not Present
BLE only	BLE_LINK, SBDT	LORA_MAC, FSK_MAC, NWK_SYNC, GWD
FSK only	FSK_MAC, FSK_LINK, NWK_SYNC, GWD, FFN, SBDT	LORA_MAC, BLE_LINK
LoRa only	LORA_MAC, LORA_LINK, NWK_SYNC	FSK_MAC, BLE_LINK, GWD, SBDT
BLE + LoRa	LORA_MAC, LORA_LINK, BLE_LINK, NWK_SYNC, SBDT	FSK_MAC, GWD
BLE + FSK	FSK_MAC, FSK_LINK, BLE_LINK, NWK_SYNC, GWD, FFN, SBDT	LORA_MAC
FSK + LoRa	LORA_MAC, LORA_LINK, FSK_MAC, FSK_LINK, NWK_SYNC, GWD, FFN, SBDT	BLE_LINK
BLE + FSK + LoRa	All categories present	None

BLE/EP/EPMETRICS/REPORT/BV/01: Verify Endpoint sends Endpoint Metrics to Cloud over BLE in a specific time interval

Case ID: C103619325

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972855

Test Steps:

Step 1

Description:

Initialize BLE stack and wait for time sync:
sid init 1 (or 4, 5, 7 - any stack with BLE enabled)
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
Key check: Time synced (Time: 0)

Step 2

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows metrics data

Step 3

Description:

Wait uploading interval for periodic reporting or trigger by command: metrics send
Default uploading interval is 24 hours
Check current interval: print get_cap 3 -> metrics periodicity

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
Key check: EN creates BLE connection and sends metrics to cloud

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data uploaded to cloud

BLE/EP/EPMETRICS/REPORT/BV/02: Verify Endpoint sends remaining Endpoint Metrics to Cloud over BLE via piggybacking on a periodic reporting

Case ID: C103619326

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/87023642

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack and wait for time sync:
sid init 7
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
Key check: Time synced (Time: 0)

Step 2

Description:

Initialize SBDT:
sbdt init

Expected Result:

CMD: ERR: 0

Step 3

Description:

Set all metrics to boundary value to exceed BLE MTU (255):

LoRa MAC Metrics

metrics set 0x01 0x01 -128 (dl_avg_rssi)
metrics set 0x01 0x02 -128 (dl_min_rssi)
metrics set 0x01 0x03 65535 (radio_error)
metrics set 0x01 0x04 255 (critical_radio_error)
metrics set 0x01 0x05 65535 (lora_rx_pkts)
metrics set 0x01 0x06 65535 (lora_tx_normal_pkts)
metrics set 0x01 0x07 65535 (lora_tx_low_latency_pkts)
metrics set 0x01 0x08 65535 (lora_tx_ll_in_time_pkts)
metrics set 0x01 0x09 65535 (lora_tx_ll_not_in_time_by_rate_limit)
metrics set 0x01 0x0a 255 (lora_tx_ll_not_in_time_by_preemption)
metrics set 0x01 0x0b 255 (lora_crc_error)

LoRa Link Metrics

metrics set 0x02 0x01 65535 (ul_msg)
metrics set 0x02 0x02 255 (ul_msg_burst)
metrics set 0x02 0x03 65535 (dl_msg)
metrics set 0x02 0x04 65535 (ul_msg_err)
metrics set 0x02 0x05 65535 (dl_msg_dec_err)

metrics set 0x02 0x06 255 (dl_msg_urun_err)
metrics set 0x02 0x07 255 (ul_avg_app_pld_size)
metrics set 0x02 0x08 255 (dl_avg_app_pld_size)
metrics set 0x02 0x09 65535 (ul_msg_nwk_server)

FSK MAC Metrics

metrics set 0x03 0x01 -128 (dl_avg_rssi)
metrics set 0x03 0x02 -128 (dl_min_rssi)
metrics set 0x03 0x03 65535 (radio_error)
metrics set 0x03 0x04 255 (critical_radio_error)
metrics set 0x03 0x05 4294967295 (fsk_tx_pkts)
metrics set 0x03 0x06 4294967295 (fsk_tx_fail)
metrics set 0x03 0x07 4294967295 (fsk_ack_not_received)
metrics set 0x03 0x08 4294967295 (fsk_rx_pkts)
metrics set 0x03 0x09 4294967295 (fsk_tx_ack_pkts)
metrics set 0x03 0x0a 4294967295 (fsk_crc_error)

FSK Link Metrics

metrics set 0x04 0x01 4294967295 (ul_msg)
metrics set 0x04 0x02 255 (ul_msg_burst)
metrics set 0x04 0x03 4294967295 (dl_msg)
metrics set 0x04 0x04 65535 (ul_msg_err)
metrics set 0x04 0x05 65535 (dl_msg_dec_err)
metrics set 0x04 0x06 255 (dl_msg_urun_err)
metrics set 0x04 0x07 255 (ul_avg_app_pld_size)
metrics set 0x04 0x08 255 (dl_avg_app_pld_size)
metrics set 0x04 0x09 65535 (ul_msg_nwk_server)

BLE Link Metrics

metrics set 0x05 0x01 4294967295 (ul_msg)
metrics set 0x05 0x02 65535 (ul_msg_burst)
metrics set 0x05 0x03 4294967295 (dl_msg)
metrics set 0x05 0x04 4294967295 (ul_msg_err)
metrics set 0x05 0x05 4294967295 (dl_msg_dec_err)
metrics set 0x05 0x06 255 (dl_msg_urun_err)
metrics set 0x05 0x07 255 (ul_avg_app_pld_size)
metrics set 0x05 0x08 255 (dl_avg_app_pld_size)
metrics set 0x05 0x09 65535 (conn_success)
metrics set 0x05 0x0a 65535 (conn_attempts)
metrics set 0x05 0x0b 65535 (conn_crit_error_types)
metrics set 0x05 0x0c -128 (dl_avg_rssi)
metrics set 0x05 0x0d -128 (dl_min_rssi)

Time Sync Metrics

metrics set 0x06 0x01 65535 (time_requests)
metrics set 0x06 0x02 65535 (time_responses)
metrics set 0x06 0x03 65535 (dynamic_periodicity)
metrics set 0x06 0x04 65535 (dynamic_drift)

Network Sync Metrics

metrics set 0x07 0x01 65535 (join_req_fsk)
metrics set 0x07 0x02 65535 (join_rsp_fsk)
metrics set 0x07 0x03 65535 (join_req_lora)

metrics set 0x07 0x04 65535 (join_rsp_lora)

Link Message Manager Metrics

metrics set 0x08 0x01 4294967295 (ul_msg_retries_ble)
metrics set 0x08 0x02 65535 (ul_msg_retries_fsk)
metrics set 0x08 0x03 65535 (ul_msg_retries_lora)
metrics set 0x08 0x04 4294967295 (dl_msg_dup_ble)
metrics set 0x08 0x05 65535 (dl_msg_dup_fsk)
metrics set 0x08 0x06 65535 (dl_msg_dup_lora)
metrics set 0x08 0x07 255 (avg_msg_roundtrip_time)

Security State Machine Metrics

metrics set 0x09 0x01 65535 (decrypt_err)
metrics set 0x09 0x02 65535 (auth_err)

Gateway Discovery and Tracking Metrics

metrics set 0x0a 0x01 -128 (avg_discovered_gw_rssi)
metrics set 0x0a 0x02 255 (gw_discovery_att_per_conn)
metrics set 0x0a 0x03 65535 (gw_discovery_duration_secs)
metrics set 0x0a 0x04 65535 (num_invalid_beacon_detect)
metrics set 0x0a 0x05 65535 (total_beacons_received)
metrics set 0x0a 0x06 65535 (total_beacons_missed)
metrics set 0x0a 0x07 65535 (total_sync_losses_due_to_beacon_misses)
metrics set 0x0a 0x08 65535 (times_gw_discovery_invoked)

Registration and Key Refresh Metrics

metrics set 0x0b 0x01 255 (key_refresh_att)
metrics set 0x0b 0x02 255 (key_refresh_err)
metrics set 0x0b 0x03 255 (reg_att)
metrics set 0x0b 0x04 255 (reg_err)

FFS/FFN Metrics

metrics set 0x0c 0x01 255 (reg_retries)

Sidewalk Bulk Data Transfer Metrics

metrics set 0x0d 0x01 255 (attempts)
metrics set 0x0d 0x02 255 (errors)

Multi-Link Manager Metrics

metrics set 0x0e 0x01 65535 (conn_att_ble)
metrics set 0x0e 0x02 65535 (conn_att_fsk)
metrics set 0x0e 0x03 65535 (conn_att_lora)
metrics set 0x0e 0x04 65535 (conn_timeout_ble)
metrics set 0x0e 0x05 65535 (conn_timeout_fsk)
metrics set 0x0e 0x06 65535 (conn_timeout_lora)
metrics set 0x0e 0x07 255 (avg_msg_roundtrip_time)
metrics set 0x0e 0x08 65535 (max_time_to_succ_conn)
metrics set 0x0e 0x09 255 (num_links_travrsd_to_succ_conn)
metrics set 0x0e 0x0a 255 (last_selected_policy)

Autoconnect Manager Metrics

metrics set 0x0f 0x01 65535 (conn_att_ble)

metrics set 0x0f 0x02 65535 (conn_att_fsk)
metrics set 0x0f 0x03 65535 (conn_att_lora)
metrics set 0x0f 0x04 65535 (conn_timeout_ble)
metrics set 0x0f 0x05 65535 (conn_timeout_fsk)
metrics set 0x0f 0x06 65535 (conn_timeout_lora)
metrics set 0x0f 0x07 255 (all_links_last_priority_cfg)
metrics set 0x0f 0x08 65535 (last_timeout_cfg_ble)
metrics set 0x0f 0x09 65535 (last_timeout_cfg_fsk)
metrics set 0x0f 0x0a 65535 (last_timeout_cfg_lora)
metrics set 0x0f 0x0b 65535 (min_time_to_succ_conn)
metrics set 0x0f 0x0c 65535 (max_time_to_succ_conn)

Gateway Scanning and Selection Metrics

metrics set 0x10 0x01 255 (uniq_pan_gw_found)
metrics set 0x10 0x02 255 (uniq_wan_gw_found)
metrics set 0x10 0x03 255 (total_numb_scan_fail)
metrics set 0x10 0x04 255 (total_numb_sync_lost_low_rssi)
metrics set 0x10 0x05 255 (total_numb_sync_lost_gw_load)
metrics set 0x10 0x06 255 (numb_switch_policies)
metrics set 0x10 0x07 255 (last_select_policy)

Expected Result:

CMD: ERR: 0 for each command

Step 4

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows all metrics data

Step 5

Description:

Establish BLE connection and trigger metrics send or wait for periodic reporting (check periodic by print get_cap 3):
sid conn_req 1
metrics send

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
MET: NDB5 st:0
Key check: EN sends metrics via BLE, remaining metrics via piggybacking (NDB)

Step 6

Description:

Send ULs until no more metrics in NDB (all metrics uploaded):
sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

While metrics remain: MET: NDB5 st:0 (metrics piggybacked on UL)
When complete: No MET: NDB5 st:0 after sid send
Key check: All metrics uploaded via piggybacking

Step 7

Description:

Check on Splunk by TX CRID:

index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data uploaded to cloud

BLE/EP/EPMETRICS/REPORT/BV/03: Verify Endpoint Metrics are uploaded via Piggybacking over BLE if the explicit reporting is disabled

Case ID: C103619327

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972868

Test Steps:

Step 1

Description:

Initialize BLE stack and wait for time sync:
sid init 1
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
Key check: Time synced (Time: 0)

Step 2

Description:

Disable metrics reporting via API: {"deviceId": "{<DEVICE_ID>}", "enableMetricsReport": false}

Expected Result:

Received write capability and config message:
[Rx][0:2:1:1] err:0
ECPFG(8) st 10->2
ECPFG(2) st 2->5
Response write capability and config message: [Tx][0:2:3:1] st:0 ECPFG(5) st 5->10
Key check: EN received configuration and sent response

Step 3

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows all metrics data

Step 4

Description:

Establish BLE connection and trigger metrics send, verify piggybacking:
sid conn_req 1
metrics send
sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

MET: NDB5 st:0 (remaining metrics piggybacked on UL)
When complete: No MET: NDB5 st:0 after sid send
Key check: All metrics uploaded via piggybacking even though explicit reporting is disabled

BLE/EP/EPMETRICS/UL/BV/01: Verify Uplink messages Metrics

Case ID: C103619328

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972874

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, establish BLE connection, clear metrics:
sid init 1
sid start
sid conn_req 1
clear metrics

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
CMD: ERR: 0

Step 2

Description:

Send 3 ULs and verify ul_msg counter increments after each:
sid send TEST (1st UL)
print metrics 0x05 (check BLE_LINK: ul_msg = 1)
sid send TEST (2nd UL)
print metrics 0x05 (check BLE_LINK: ul_msg = 2)
sid send TEST (3rd UL)
print metrics 0x05 (check BLE_LINK: ul_msg = 3)

Expected Result:

After 1st UL: BLE_LINK: ul_msg = 1
After 2nd UL: BLE_LINK: ul_msg = 2
After 3rd UL: BLE_LINK: ul_msg = 3
Key check: ul_msg counter increments by 1 after each UL

Step 3

Description:

Upload to Cloud to check

S360 to request the metrics
https://360.sidewalk.amazon/jobs/create
Job Information
- Job Type: Perform Action For Endpoint Metrics

- Job Reason: test

Job Input
- Action: Get Metrics

- Metrics Category: BLE LINK

- Individual Metrics: Uplink Messages Per Day

- Input Network Entities: please upload .csv which includes Endpoint Sidewalk Id you want to request (you can download the template from "Sample Input File")

Submit

Expected Result:

On endpoint

- <info> app: ENC: M:15 [B:E:3]

- <info> app: TX CRID: 000040B76CCFB4C45D

- Please remember the TX CRID 000040B76CCFB4C45D (On Splunk, it will be 000000B76CCFB4C45D)

[00040987] <info> app: TX CRID: 000040B76DAFC484CD
[00040987] <info> app: Tx Start (152)
[00040987] <info> app: Tx S:7 T:1 (0) I:0
[00040988] <info> app: ECPFG(5) st 5->6
[00041048] <info> app: ECPFG(5) st 6
[00041148] <info> app: ECPFG(2) st 6->2
[00042377] <info> app: New Rx S:7 T:0
[00042377] <info> app: Rx S:7 T:0 (238/258)
[00042390] <info> app: Rx S:7 T:0 (258/258)
[00042392] <info> app: RX CRID: 000040003AAB89ECC8
[00042392] <info> app: APP_LEN:0 [PAD]DL:220
[00042392] <info> app: DEC: M:15 [0:2:3]
[00042393] <info> app: ECPFG(6) st 2->10
[00042393] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [7] :: Transaction [0][00042493] <info> app: ECPFG(6) st 10

Step 4

Description:

Check on Splunk by TX CRID

Expected Result:

The metrics is in the event

BLE/EP/EPMETRICS/DL/BV/01: Verify Downlink messages Metrics.

Case ID: C103619329

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972876

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, clear metrics:
sid init 1 (or 4, 5, 7)
sid start
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
BLE_LINK: dl_msg = 0

Step 2

Description:

Send 3 DLs from AWS IoT and verify dl_msg counter increments after each:
Send DL (seq=0) -> print metrics 0x05
Send DL (seq=1) -> print metrics 0x05
Send DL (seq=2) -> print metrics 0x05

Expected Result:

After 1st DL: EVENT SID RECEIVED: ... LINK: 1 ... | BLE_LINK: dl_msg = 1
After 2nd DL: EVENT SID RECEIVED: ... LINK: 1 ... | BLE_LINK: dl_msg = 2
After 3rd DL: EVENT SID RECEIVED: ... LINK: 1 ... | BLE_LINK: dl_msg = 3

Key check: dl_msg counter increments by 1 after each DL

Step 3

Description:

Upload metrics to Cloud via S360:
https://gamma.360.sidewalk.amazon/jobs/create
Job Type: Periodic Metrics Upload

Expected Result:

ENC: M:15 [B:E:3]
TX CRID: <crid>

Key check: Metrics uploaded, note the TX CRID

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk with correct dl_msg value

BLE/EP/EPMETRICS/CONN-BV-01: Verify Connection attempts Metrics

Case ID: C103619330

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972883

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait for BLE disconnected, clear metrics:
sid init 1
sid start
Wait for BLE disconnected
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BLE_LINK: conn_attempts = 0

Step 2

Description:

Establish BLE connection 2 times and verify conn_attempts increments:
sid conn_req 1
print metrics 0x05 (check BLE_LINK: conn_attempts = 1)
sid conn_req 1
print metrics 0x05 (check BLE_LINK: conn_attempts = 2)

Expected Result:

After 1st connection: BLE_LINK: conn_attempts = 1
After 2nd connection: BLE_LINK: conn_attempts = 2
Key check: conn_attempts counter increments by 1 after each connection attempt

Step 3

Description:

Upload to Cloud to check

S360 to request the metrics

https://gamma.360.sidewalk.amazon/jobs/create
Job Information
- Job Type: Perform Action For Endpoint Metrics

- Job Reason: test

Job Input
- Action: Get Metrics

- Metrics Category: BLE LINK

- Individual Metrics: Connection Attempts Per Day

- Input Network Entities: please upload .csv which includes Endpoint Sidewalk Id you want to request (you can download the template from "Sample Input File")

Submit

Expected Result:

On endpoint

- <info> app: ENC: M:15 [B:E:3]

- <info> app: TX CRID: 000040B76CCFB4C45D

- Please remember the TX CRID 000040B76CCFB4C45D (On Splunk, it will be 000000B76CCFB4C45D)

[00039475] <info> app: BLE Connected
[00039788] <info> app: ama_gateway_set_state Gateway [91 60] state changing from 0 to 1
[00039803] <info> app: New Rx S:0 T:10
[00039803] <info> app: Rx S:0 T:10 (5/5)
[00039803] <info> app: Tx Start (6)
[00039803] <info> app: Tx S:0 T:3 (0) I:0
[00039804] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [0] :: Transaction [10]
[00039825] <info> app: New Rx S:0 T:11
[00039825] <info> app: Rx S:0 T:11 (5/5)
[00039826] <info> app: Tx Start (6)
[00039826] <info> app: Tx S:0 T:4 (0) I:0
[00039826] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [0] :: Transaction [11]
[00039855] <info> app: New Rx S:0 T:12
[00039855] <info> app: Rx S:0 T:12 (5/5)
[00039856] <info> app: Tx Start (9)
[00039856] <info> app: Tx S:0 T:5 (0) I:0
[00039856] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [0] :: Transaction [12]
[00039885] <info> app: ama_gateway_set_state Gateway [91 60] state changing from 1 to 2
[00039917] <info> app: New Rx S:7 T:15
[00039917] <info> app: Rx S:7 T:15 (173/173)
[00039920] <info> app: RX CRID: 000040003976E539CD
[00039920] <info> app: APP_LEN:2 [PAD]DL:130
[00039920] <info> app: DEC: M:15 [B:E:0]
[00039920] <info> app: MET:OndTr st:0
[00039921] <info> app: ENC: M:15 [B:E:3]
[00039921] <info> app: APP_LEN:119 [PAD]UL:132
[00039923] <info> app: TX CRID: 000040B76CCFB4C45D
[00039923] <info> app: Tx Start (294)
[00039923] <info> app: Tx S:7 T:0 (56) I:0
[00039924] <info> app: MET:Snd st:0 0
[00039924] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [7] :: Transaction [15]
[00039946] <info> app: Tx S:7 T:0 (0) I:238
[00039985] <info> app: ECPFG(4) st 2
[00040985] <info> app: ECPFG(2) st 2->5
[00040986] <info> app: ENC: M:15 [0:2:2]
[00040986] <info> app: APP_LEN:14 [PAD]UL:99
[00040987] <info> app: TX CRID: 000040B76DAFC484CD
[00040987] <info> app: Tx Start (152)
[00040987] <info> app: Tx S:7 T:1 (0) I:0

[00040988] <info> app: ECPFG(5) st 5->6
[00041048] <info> app: ECPFG(5) st 6
[00041148] <info> app: ECPFG(2) st 6->2
[00042377] <info> app: New Rx S:7 T:0
[00042377] <info> app: Rx S:7 T:0 (238/258)
[00042390] <info> app: Rx S:7 T:0 (258/258)
[00042392] <info> app: RX CRID: 000040003AAB89ECC8
[00042392] <info> app: APP_LEN:0 [PAD]DL:220
[00042392] <info> app: DEC: M:15 [0:2:3]
[00042393] <info> app: ECPFG(6) st 2->10
[00042393] <info> app: Delete rx_buffer :: Gateway [91 60] :: Stream [7] :: Transaction [0][00042493] <info> app: ECPFG(6) st 10

Step 4

Description:

Check on Splunk by TX CRID

Expected Result:

The metrics is in the event

FSK/EP/EPMETRICS/REPORT/BV/01: Verify Endpoint sends Endpoint Metrics to Cloud over FSK in a specific time interval

Case ID: C103619332

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972856

Test Steps:

Step 1

Description:

Initialize FSK stack and wait for time sync and join:
sid init 2 (or 5, 6, 7)
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: Time synced and joined

Step 2

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows metrics data

Step 3

Description:

Wait uploading interval for periodic reporting or trigger by command:
metrics send

Default uploading interval is 24 hours
Check current interval: print get_cap 3 -> metrics periodicity

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
Key check: EN sends metrics to cloud via FSK

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

FSK/EP/EPMETRICS/REPORT/BV/02: Verify Endpoint sends remaining Endpoint Metrics to Cloud over FSK via piggybacking on a periodic reporting

Case ID: C103619333

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/87023643

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack and wait for time sync and join:
sid init 7
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6

Step 2

Description:

Initialize SBDT:
sbdt init

Expected Result:

CMD: ERR: 0

Step 3

Description:

Set all metrics to boundary value to exceed FSK MTU(200)

FSK MAC Metrics

metrics set 0x03 0x01 -128 (dl_avg_rssi)
metrics set 0x03 0x02 -128 (dl_min_rssi)
metrics set 0x03 0x03 65535 (radio_error)
metrics set 0x03 0x04 255 (critical_radio_error)
metrics set 0x03 0x05 4294967295 (fsk_tx_pkts)
metrics set 0x03 0x06 4294967295 (fsk_tx_fail)
metrics set 0x03 0x07 4294967295 (fsk_ack_not_received)
metrics set 0x03 0x08 4294967295 (fsk_rx_pkts)
metrics set 0x03 0x09 4294967295 (fsk_tx_ack_pkts)metrics set 0x03 0x0a 4294967295 (fsk_crc_error)

FSK Link Metrics

metrics set 0x04 0x01 4294967295 (ul_msg)
metrics set 0x04 0x02 255 (ul_msg_burst)
metrics set 0x04 0x03 4294967295 (dl_msg)
metrics set 0x04 0x04 65535 (ul_msg_err)
metrics set 0x04 0x05 65535 (dl_msg_dec_err)
metrics set 0x04 0x06 255 (dl_msg_urun_err)
metrics set 0x04 0x07 255 (ul_avg_app_pld_size)

metrics set 0x04 0x08 255 (dl_avg_app_pld_size)metrics set 0x04 0x09 65535 (ul_msg_nwk_server)

Time Sync Metrics

metrics set 0x06 0x01 65535 (time_requests)
metrics set 0x06 0x02 65535 (time_responses)
metrics set 0x06 0x03 65535 (dynamic_periodicity)metrics set 0x06 0x04 65535 (dynamic_drift)

Network Sync Metrics

metrics set 0x07 0x01 65535 (join_req_fsk)
metrics set 0x07 0x02 65535 (join_rsp_fsk)
metrics set 0x07 0x03 65535 (join_req_lora)metrics set 0x07 0x04 65535 (join_rsp_lora)

Link Message Manager Metrics

metrics set 0x08 0x01 4294967295 (ul_msg_retries_ble)
metrics set 0x08 0x02 65535 (ul_msg_retries_fsk)
metrics set 0x08 0x03 65535 (ul_msg_retries_lora)
metrics set 0x08 0x04 4294967295 (dl_msg_dup_ble)
metrics set 0x08 0x05 65535 (dl_msg_dup_fsk)
metrics set 0x08 0x06 65535 (dl_msg_dup_lora)metrics set 0x08 0x07 255 (avg_msg_roundtrip_time)

Security State Machine Metrics

metrics set 0x09 0x01 65535 (decrypt_err)metrics set 0x09 0x02 65535 (auth_err)

Gateway Discovery and Tracking Metrics

metrics set 0x0a 0x01 -128 (avg_discovered_gw_rssi)
metrics set 0x0a 0x02 255 (gw_discovery_att_per_conn)
metrics set 0x0a 0x03 65535 (gw_discovery_duration_secs)
metrics set 0x0a 0x04 65535 (num_invalid_beacon_detect)
metrics set 0x0a 0x05 65535 (total_beacons_received)
metrics set 0x0a 0x06 65535 (total_beacons_missed)
metrics set 0x0a 0x07 65535 (total_sync_losses_due_to_beacon_misses)metrics set 0x0a 0x08 65535 (times_gw_discovery_invoked)

Registration and Key Refresh Metrics

metrics set 0x0b 0x01 255 (key_refresh_att)
metrics set 0x0b 0x02 255 (key_refresh_err)
metrics set 0x0b 0x03 255 (reg_att)metrics set 0x0b 0x04 255 (reg_err)

FFS/FFN Metrics

metrics set 0x0c 0x01 255 (reg_retries)

Sidewalk Bulk Data Transfer Metrics

metrics set 0x0d 0x01 255 (attempts)metrics set 0x0d 0x02 255 (errors)

Multi-Link Manager Metrics

metrics set 0x0e 0x01 65535 (conn_att_ble)
metrics set 0x0e 0x02 65535 (conn_att_fsk)
metrics set 0x0e 0x03 65535 (conn_att_lora)
metrics set 0x0e 0x04 65535 (conn_timeout_ble)
metrics set 0x0e 0x05 65535 (conn_timeout_fsk)
metrics set 0x0e 0x06 65535 (conn_timeout_lora)
metrics set 0x0e 0x07 255 (avg_msg_roundtrip_time)

metrics set 0x0e 0x08 65535 (max_time_to_succ_conn)
metrics set 0x0e 0x09 255 (num_links_travrsd_to_succ_conn)metrics set 0x0e 0x0a 255 (last_selected_policy)

Autoconnect Manager Metrics

metrics set 0x0f 0x01 65535 (conn_att_ble)
metrics set 0x0f 0x02 65535 (conn_att_fsk)
metrics set 0x0f 0x03 65535 (conn_att_lora)
metrics set 0x0f 0x04 65535 (conn_timeout_ble)
metrics set 0x0f 0x05 65535 (conn_timeout_fsk)
metrics set 0x0f 0x06 65535 (conn_timeout_lora)
metrics set 0x0f 0x07 255 (all_links_last_priority_cfg)
metrics set 0x0f 0x08 65535 (last_timeout_cfg_ble)
metrics set 0x0f 0x09 65535 (last_timeout_cfg_fsk)
metrics set 0x0f 0x0a 65535 (last_timeout_cfg_lora)
metrics set 0x0f 0x0b 65535 (min_time_to_succ_conn)metrics set 0x0f 0x0c 65535 (max_time_to_succ_conn)

Gateway Scanning and Selection Metrics

metrics set 0x10 0x01 255 (uniq_pan_gw_found)
metrics set 0x10 0x02 255 (uniq_wan_gw_found)
metrics set 0x10 0x03 255 (total_numb_scan_fail)
metrics set 0x10 0x04 255 (total_numb_sync_lost_low_rssi)
metrics set 0x10 0x05 255 (total_numb_sync_lost_gw_load)
metrics set 0x10 0x06 255 (numb_switch_policies)metrics set 0x10 0x07 255 (last_select_policy)

Expected Result:

CMD: ERR: 0 for each command

Step 4

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows all metrics data

Step 5

Description:

Trigger metrics send:
metrics send
sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
MET: NDB5 st:0 (remaining metrics piggybacked on UL)
When complete: No MET: NDB5 st:0 after sid send
Key check: EN sends metrics via FSK, remaining via piggybacking

Step 6

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

FSK/EP/EPMETRICS/REPORT/BV/03: Verify Endpoint Metrics are uploaded via Piggybacking over FSK if the explicit reporting is disabled

Case ID: C103619334

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/86144603

Test Steps:

Step 1

Description:

Initialize FSK stack and wait for time sync and join:
sid init 2 (or 5, 6, 7)
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6

Step 2

Description:

Disable metrics reporting via API:

{"deviceId": "{<DEVICE_ID>}", "enableMetricsReport": false}

Expected Result:

Received write capability and config message:
[Rx][0:2:1:2] err:0

ECPFG(8) st 10->2
ECPFG(2) st 2->5

Response write capability and config message:
[Tx][0:2:3:2] st:0

ECPFG(5) st 5->10

Key check: EN received configuration and sent response

Step 3

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows all metrics data

Step 4

Description:

Trigger metrics send and verify piggybacking:

metrics send

sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

MET: NDB5 st:0 (remaining metrics piggybacked on UL)

When complete: No MET: NDB5 st:0 after sid send

Key check: All metrics uploaded via piggybacking even though explicit reporting is disabled

Step 5

Description:

Recover metrics reporting setting:

print clear_cap 1

Expected Result:

CMD: ERR: 0

Step 6

Description:

Check on Splunk by TX CRID:

index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

FSK/EP/EPMETRICS/UL/BV/01: Verify Uplink messages Metrics and Total number of TX unique packets Metrics

Case ID: C103619335

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972902

Test Steps:

Step 1

Description:

Initialize FSK stack, wait for time sync and join, clear metrics:
sid init 2 (or 5, 6, 7)
sid start
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
FSK_MAC: fsk_tx_pkts = 0

Step 2

Description:

Send UL and verify fsk_tx_pkts increments:
sid send TEST
print metrics 0x03 (check FSK_MAC: fsk_tx_pkts = 1)

Expected Result:

FskTx:TxDone sqn:<sqn> dst_id:0x<id> ch:<ch> pktlen:<len> tx_st:0
FSK_MAC: fsk_tx_pkts = 1
Key check: fsk_tx_pkts increments by 1 after each TX

Step 3

Description:

Upload metrics to Cloud via S360:
gamma: https://gamma.360.sidewalk.amazon/jobs/create
prod: https://360.sidewalk.amazon/jobs/create

Job Type: Periodic Metrics Upload

Expected Result:

ENC: M:15 [B:E:3]
TX CRID: <crid>
Key check: Metrics uploaded, note the TX CRID

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

FSK/EP/EPMETRICS/DL/BV/01: Verify Downlink messages Metrics and Total number of RX unique packets Metrics

Case ID: C103619336

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972905

Test Steps:

Step 1

Description:

Initialize FSK stack, wait for time sync and join, clear metrics:
sid init 2 (or 5, 6, 7)
sid start
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
FSK_MAC: fsk_rx_pkts = 0

Step 2

Description:

Send DL from AWS IoT and verify fsk_rx_pkts increments:
Send DL from AWS IoT
print metrics 0x03 (check FSK_MAC: fsk_rx_pkts = 1)

Expected Result:

FskRx:RxDone sqn:<sqn> src_id:0x<id> dst_id:0x<id> ch:<ch> pktlen:<len> rx_st:0
FSK_MAC: fsk_rx_pkts = 1
Key check: fsk_rx_pkts increments by 1 after each RX

Step 3

Description:

Upload metrics to Cloud via S360:
gamma: https://gamma.360.sidewalk.amazon/jobs/create
prod: https://360.sidewalk.amazon/jobs/create
Job Type: Periodic Metrics Upload

Expected Result:

ENC: M:15 [B:E:3]
TX CRID: <crid>
Key check: Metrics uploaded, note the TX CRID

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

LORA/EP/EPMETRICS/REPORT/BV/01: Verify Endpoint sends Endpoint Metrics to Cloud over LoRa in a specific time interval

Case ID: C103619339

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972857

Test Steps:

Step 1

Description:

Initialize LoRa stack and wait for time sync and join:
sid init 3 (or 4, 6, 7)
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6
Key check: Time synced and joined

Step 2

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows metrics data

Step 3

Description:

Wait uploading interval for periodic reporting or trigger by command:
metrics send

Default uploading interval is 24 hours
Check current interval: print get_cap 3 -> metrics periodicity

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
Key check: EN sends metrics to cloud via LoRa

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

LORA/EP/EPMETRICS/REPORT/BV/02: Verify Endpoint sends remaining Endpoint Metrics to Cloud over LoRa via piggybacking on a periodic reporting

Case ID: C103619340

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/87023644

Test Steps:

Step 1

Description:

Initialize BLE+FSK+LoRa stack and wait for time sync and join:
sid init 7
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6

Step 2

Description:

Set all metrics to boundary value (see C87023642 step 3 for full command list)

Expected Result:

CMD: ERR: 0 for each command

Step 3

Description:

Print all metrics:
print metrics 0x36

Expected Result:

EN shows all metrics data

Step 4

Description:

Trigger metrics send:
metrics send
sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

ENC: M:15 [B:E:2]
TX CRID: <crid>
MET: NDB5 st:0 (remaining metrics piggybacked on UL)
When complete: No MET: NDB5 st:0 after sid send

Key check: EN sends metrics via LoRa, remaining via piggybacking

Step 5

Description:

Check on Splunk by TX CRIDs:

index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PIGGYBACK"

Expected Result:

Metrics data found on Splunk (both periodic and piggybacked)

LORA/EP/EPMETRICS/REPORT/BV/03: Verify Endpoint Metrics are uploaded via Piggybacking over LoRa if the explicit reporting is disabled

Case ID: C103619341

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/86144604

Test Steps:

Step 1

Description:

Initialize LoRa stack and wait for time sync and join:
sid init 3 (or 4, 6, 7)
sid start

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6

Step 2

Description:

Disable metrics reporting via API:

{"deviceId": "{<DEVICE_ID>}", "enableMetricsReport": false}

Expected Result:

Received write capability and config message:
[Rx][0:2:1:4] err:0

ECPFG(8) st 10->2
ECPFG(2) st 2->5

Response write capability and config message:
[Tx][0:2:3:4] st:0

ECPFG(5) st 5->10

Key check: EN received configuration and sent response

Step 3

Description:

Verify metrics reporting is disabled:
print get_cap 1

Expected Result:

metrics enabled 0

Step 4

Description:

Print all metrics:

print metrics 0x36

Expected Result:

EN shows all metrics data

Step 5

Description:

Trigger metrics send and verify piggybacking:

metrics send

sid send TEST (repeat until MET: NDB5 st:0 no longer appears)

Expected Result:

MET: NDB5 st:0 (remaining metrics piggybacked on UL)

When complete: No MET: NDB5 st:0 after sid send

Key check: All metrics uploaded via piggybacking even though explicit reporting is disabled

Step 6

Description:

Recover metrics reporting setting:

print clear_cap 1

Expected Result:

CMD: ERR: 0

Step 7

Description:

Check on Splunk by TX CRIDs:

index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PIGGYBACK"

Expected Result:

Metrics data found on Splunk (piggybacked only, no periodic)

LORA/EP/EPMETRICS/UL/BV/01: Verify Uplink messages Metrics and Total number of TX normal packets Metrics

Case ID: C103619342

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972922

Test Steps:

Step 1

Description:

Initialize LoRa stack, wait for time sync and join, clear metrics:
sid init 3 (or 4, 6, 7)
sid start
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6
LORA_MAC: lora_tx_normal_pkts = 0

Step 2

Description:

Send UL and verify lora_tx_normal_pkts increments:
sid send TEST
print metrics 0x01 (check LORA_MAC: lora_tx_normal_pkts = 1)

Expected Result:

LoraTx:TxDone sqn:<sqn> dst_id:0x<id> ch:<ch> pktlen:<len> tx_st:0
LORA_MAC: lora_tx_normal_pkts = 1

Key check: lora_tx_normal_pkts increments by 1 after each TX

Step 3

Description:

Upload metrics to Cloud via S360:
https://gamma.360.sidewalk.amazon/jobs/create
Job Type: Periodic Metrics Upload

Expected Result:

ENC: M:15 [B:E:3]
TX CRID: <crid>

Key check: Metrics uploaded, note the TX CRID

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

LORA/EP/EPMETRICS/DL/BV/01: Verify Downlink messages Metrics and Total number of RX packets Metrics

Case ID: C103619343

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/83972921

Test Steps:

Step 1

Description:

Initialize LoRa stack, wait for time sync and join, clear metrics:
sid init 3 (or 4, 6, 7)
sid start
print clear_metrics 0x36

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6
LORA_MAC: lora_rx_pkts = 0

Step 2

Description:

Send DL from AWS IoT and verify lora_rx_pkts increments:
Send DL from AWS IoT
print metrics 0x01 (check LORA_MAC: lora_rx_pkts = 1)

Expected Result:

LoraRx:RxDone sqn:<sqn> src_id:0x<id> dst_id:0x<id> ch:<ch> pktlen:<len> rx_st:0
LORA_MAC: lora_rx_pkts = 1

Key check: lora_rx_pkts increments by 1 after each RX

Step 3

Description:

Upload metrics to Cloud via S360:
https://gamma.360.sidewalk.amazon/jobs/create
Job Type: Periodic Metrics Upload

Expected Result:

ENC: M:15 [B:E:3]
TX CRID: <crid>

Key check: Metrics uploaded, note the TX CRID

Step 4

Description:

Check on Splunk by TX CRID:
index=halo_operations* <sidewalkID> methodIdentifier = "ENDPOINT_METRICS_PERIODIC"

Expected Result:

Metrics data found on Splunk

EP Capability Test Cases

CMN-EP-CAPABILITY-REPORT-BV-01: When KV store does not have the capability report saved, the event to send capability is scheduled

Case ID: C103619346

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497750

Test Steps:

Step 1

Description:

Clear capability KV store to simulate first-time setup (no capability report saved):
print clear_cap 3

Expected Result:

CMD: ERR: 0
Key check: KV store cleared, EN will send capability report on next startup

Step 2

Description:

Initialize stack and start:
BLE: sid init 1 / sid start
FSK: sid init 2 / sid start
LoRa: sid init 3 / sid start

Expected Result:

CMD: ERR: 0

Step 3

Description:

Wait for time sync and join:
BLE: Wait for time sync only
FSK/LoRa: Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 4

Description:

Check EN sends capability report and receives ACK
BLE: establish BLE connection first: sid conn_req 1

Expected Result:

ECPFG(2) st 2->5 (Ready to send)
Sent capability and config message:
[Tx][0:2:2:-] st:0

ECPFG(5) st 5->6 (Waiting for ACK)
ECPFG(2) st 6->2 (ACK received, back to check state)
ECPFG(6) st 2->7 (Updating KV store)

Received capability and config message:
[Rx][0:2:3:-] err:0

ECPFG(2) st 7->10 (Done, idle)

Key check: Capability report sent and ACK received

CMN-EP-CAPABILITY-REPORT-BV-02: When Capability stored in the KV store is equal to the capability of the Endpoint, the Capability event report is not generated.

Case ID: C103619347

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497752

Test Steps:

Step 1

Description:

Clear capability KV store to simulate first-time setup (no capability report saved):
print clear_cap 3

Expected Result:

CMD: ERR: 0
Key check: KV store cleared, EN will send capability report on next startup

Step 2

Description:

Initialize stack and start:
BLE: sid init 1 / sid start
FSK: sid init 2 / sid start
LoRa: sid init 3 / sid start

Expected Result:

CMD: ERR: 0

Step 3

Description:

Wait for time sync and join:
BLE: Wait for time sync only
FSK/LoRa: Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 4

Description:

Check EN sends capability report and receives ACK
BLE: establish BLE connection first: sid conn_req 1

Expected Result:

ECPFG(2) st 2->5 (Ready to send)
Sent capability and config message:
[Tx][0:2:2:-] st:0

ECPFG(5) st 5->6 (Waiting for ACK)
ECPFG(2) st 6->2 (ACK received, back to check state)
ECPFG(6) st 2->7 (Updating KV store)

Received capability and config message:
[Rx][0:2:3:-] err:0

ECPFG(2) st 7->10 (Done, idle)

Key check: Capability report sent and ACK received

Step 5

Description:

Wait 5 seconds, then reinitialize stack and wait for time sync/join:
sid deinit
BLE: sid init 1 / sid start
FSK: sid init 2 / sid start
LoRa: sid init 3 / sid start
BLE: Wait for time sync only, then: sid conn_req 1
FSK/LoRa: Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 6

Description:

Verify EN does NOT send capability report again

Expected Result:

No [Tx][0:2:2:-] within 20 seconds

Key check: Capability report NOT sent (KV store already synced with cloud)

CMN-EP-CAPABILITY-REPORT-BV-03: Endpoint default capability report

Case ID: C103619348

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/86756857

Test Steps:

Step 1

Description:

Init and start stack: sid init <stack>, sid start

Step 2

Description:

Wait until time sync and join complete, for BLE, wait time sync complete only

Expected Result:

time sync complete:

[Rx][0:108:3:4] err:0

join complete:

[Rx][0:22F:3:4] err:0

Step 3

Description:

Check if EN send capability report and recieve ACK, for ble, need to establish BLE connection first
(if KV store of EN synced with cloud, ignore this step)

Expected Result:

For BLE, check ECPFG only

Event: SID_EP_CAP_CFG_EVENT_INTERNAL
State: SID_EP_CAP_CFG_STATE_CHECK_FOR_NTFY_STATUS_READY -> SID_EP_CAP_CFG_STATE_SEND_NTFY_READY
<info> app: ECPFG(2) st 2->5

Sent capability and config message:
<info> app: [Tx][0:2:2:<stack>] st:0, FSK stack is 2, lora is 4

Event: SID_EP_CAP_CFG_EVENT_ON_SENT_STATUS_RECVD
State: SID_EP_CAP_CFG_STATE_SEND_NTFY_READY -> SID_EP_CAP_CFG_STATE_WAIT_FOR_ACK_FOR_NOTIFFICATION
<info> app: ECPFG(5) st 5->6

Event: SID_EP_CAP_CFG_EVENT_INTERNAL
State: SID_EP_CAP_CFG_STATE_WAIT_FOR_ACK_FOR_NOTIFFICATION ->
SID_EP_CAP_CFG_STATE_CHECK_FOR_NTFY_STATUS_READY
<info> app: ECPFG(2) st 6->2

Event: SID_EP_CAP_CFG_EVENT_ACK_FOR_NOTIFICAITON_RECVD

State: SID_EP_CAP_CFG_STATE_CHECK_FOR_NTFY_STATUS_READY -> SID_EP_CAP_CFG_STATE_UPDATE_KV_STORE
<info> app: ECPFG(6) st 2->7

Received capability and config message:
<info> app: [Rx][0:2:3:<stack>] err:0, FSK stack is 2, lora is 4

Event: SID_EP_CAP_CFG_EVENT_INTERNAL
State: SID_EP_CAP_CFG_STATE_UPDATE_KV_STORE -> SID_EP_CAP_CFG_STATE_NTFY_IDLE
<info> app: ECPFG(2) st 7->10

Step 4

Description:

Print the current capabilities:

print get_cap 3

Expected Result:

Should contain EP capability content:

EX:

--> print get_cap 3
[00026911] <info> app: Capability version 0
[00026911] <info> app: SDK version 40.32.0
[00026911] <info> app: links enabled 0x07
[00026911] <info> app: Links enabled BLE 1 FSK 1 LoRa 1
[00026911] <info> app: features_support 0x07F3
[00026911] <info> app: Features support Static device 1 Mobile Device 0 Battery powered 0 Line Powered 1 ffs over fsk 1 metrics enabled 1
[00026912] <info> app: Features support Coverage test 1 Lora low latency 1 Auto connect 1 MLM 1 SBDT 1 Traffic throttling enabled 0
[00026917] <info> app: Qualification id 0x07AB
[00026917] <info> app: Traffic threshold table id 0
[00026917] <info> app: metrics periodicity 24 hours
[00026917] <info> app: BLE Tx power 0 FSK Tx power 20 LoRa Tx power 20
[00026917] <info> app: threshold table id 0
[00026917] <info> app: LoRa static normal rate 1 message every 15 minutes
[00026917] <info> app: LoRa mobile normal rate 1 message every 6 minutes
[00026917] <info> app: LoRa static burst rate 1 message every 5 minutes
[00026917] <info> app: LoRa mobile burst rate 1 message every 1 minutes
[00026917] <info> app: LoRa static max packets per day 96
[00026917] <info> app: LoRa mobile max packets per day 250
[00026917] <info> app: FSK max packets per min 700
[00026918] <info> app: FSK max packets per day 1008000
[00026918] <info> app: BLE max packets per min 600
[00026918] <info> app: BLE max packets per day 864000

Step 5

Description:

Get capability via API

Expected Result:

Check the capability is same as step 4

EX:


```
{
   "capabilities": {
      "enableMetricsReport": true,
      "enableTrafficThrottling": false,
      "featuresSupported": [
         "FFN_OVER_FSK",
         "LORA_LOW_LATENCY",
         "COVERAGE_SUPPORT",
         "MULTI_LINK",
         "STATIC_DEVICE",
         "LINE_POWERED",
         "SBDT",
         "AUTO_CONNECT",
         "LINE_POWERED",
         "LINK_METRICS"
      ],
      "linksSupported": [
         "LORA",
         "BLE",
         "FSK"
      ],
      "manufacturerType": "REFERENCE_BOARDS",
      "maxTxPower": {
         "bleTxPower": 0,
         "fskTxPower": 20,
         "loraTxPower": 20
      },
      "metricsPeriodicity": 4,
      "qualificationId": "1",
      "sdkVersion": "40.32.0",
      "trafficThresholdTable": {
         "ble": {
            "packetsPerMin": 600
         },
         "fsk": {
            "packetsPerMin": 700
         },
         "lora": {
            "mobileBurstRate": 1,
            "mobileBurstUnit": "MINUTES",
            "mobileMaxPacketPerDay": 250,
            "mobileNormalRate": 6,
            "mobileNormalUnit": "MINUTES",
            "staticBurstRate": 5,
            "staticBurstUnit": "MINUTES",
            "staticMaxPacketPerDay": 96,
            "staticNormalRate": 15,
            "staticNormalUnit": "MINUTES"
         },
         "tableId": 0
      },
```

```
      "version": 0
    },
    "deviceId": "A1234567890",
    "intended": {},
    "updatedAt": 1727330209771
}
```


PS.

- The unit of metricsPeriodicity is 6hr, so here "metricsPeriodicity": 4 which means 4*6 = 24 hours show on EP

Step 6

Description:

Search on splunk, if KV store of EN already synced with cloud on step 3, initiateRequestCapabilities via API then search on splunk:

index=halo_operations* <sidewalkID> methodIdentifier = "NotifyCapabilityHandler"

Expected Result:

All values should same as step 4

CMN-EP-CAPABILITY-REQUEST-BV-01: Endpoint is online, send request to get endpoint capability

Case ID: C103619350

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497758

Test Steps:

Step 1

Description:

Initialize stack, start, and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 2

Description:

Initiate request capability via API, then check EN sends capability report and ACK received

Expected Result:

API response: {"requestId": "<id>", "status": "SUCCESS"}

On EN:
[Rx][0:2:0:-] err:0 (Read request received)
ECPFG(2) st 2->5 (Ready to send)
Sent capability and config message:
[Tx][0:2:3:-] st:0
ECPFG(5) st 5->10 (Done, idle)

Key check: EN received capability request and sent response

Step 3

Description:

Wait 20 seconds for cloud to update, then get capability via API and verify it matches EN:
print get_cap 3

Expected Result:

API response capability matches EN capability values

Key check: Cloud capability same as EN capability

CMN-EP-CAPABILITY-CONFIGURE-BV-01: Endpoint configuraton shall be updated on user setting

Case ID: C103619351

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497763

Test Steps:

Step 1

Description:

Initialize stack, start, and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 2

Description:

Set configuration via API with the following values:
{ "deviceId": "{<DEVICE_ID>}", "trafficThresholdTable": { "tableId": 1, "lora": { "staticNormalRate": 1, "staticNormalUnit": "MINUTES", "staticBurstRate": 2, "staticBurstUnit": "MINUTES", "mobileNormalRate": 3, "mobileNormalUnit": "MINUTES", "mobileBurstRate": 4, "mobileBurstUnit": "MINUTES", "staticMaxPacketPerDay": 5, "mobileMaxPacketPerDay": 6 }, "fsk": { "packetsPerMin": 7 }, "ble": { "packetsPerMin": 8 } }, "enableTrafficThrottling": { "tableId": 1, "enableThrottling": false }, "enableMetricsReport": false, "metricsPeriodicity": 9}

Expected Result:

- Received write capability and config message: - [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5 (INTERNAL event, CHECK -> SEND_NTFY_READY)

- Response write capability and config message: - [Tx][0:2:3:-] st:0 ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)

Key check: EN received configuration and sent response

Step 3

Description:

Wait 20 seconds for cloud to update, then verify cloud capability matches EN:
print get_cap 3

Expected Result:

API response capability matches EN capability values Key check: Cloud capability same as EN capability after configuration update

CMN-EP-CAPABILITY-REGISTER-BV-01: Endpoint capability is set to default after re-registration

Case ID: C103619352

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84864409

Test Steps:

Step 1

Description:

Initialize stack, start, and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join
(FSK): Njoin(1, 2) st 2->6

Step 2

Description:

Set configuration via API with the following values:


```
{
  "deviceId": "{<DEVICE_ID>}",
  "trafficThresholdTable": {
    "tableId": 3,
    "lora": {
      "staticNormalRate": 1,
      "staticNormalUnit": "MINUTES",
      "staticBurstRate": 1,
      "staticBurstUnit": "MINUTES",
      "mobileNormalRate": 1,
      "mobileNormalUnit": "MINUTES",
      "mobileBurstRate": 1,
      "mobileBurstUnit": "MINUTES",
      "staticMaxPacketPerDay": 1,
      "mobileMaxPacketPerDay": 1
    },
    "fsk": {
      "packetsPerMin": 1
    },
    "ble": {
      "packetsPerMin": 1
    }
  },
  "enableTrafficThrottling": {
    "tableId": 3,
    "enableThrottling": false
  },
```

"enableMetricsReport": false,
"metricsPeriodicity": 1}

Expected Result:

- Received write capability and config message: - [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5 (INTERNAL event, CHECK -> SEND_NTFY_READY)

- Response write capability and config message: - [Tx][0:2:3:-] st:0 ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)

Key check: EN received configuration and sent response

Step 3

Description:

Deregister EN via API

Expected Result:

EVENT SID STATUS: State: 1, Reg: 1, Time: 1, Link_Mask: 0 Key check: EN deregistered (Reg: 1)

Step 4

Description:

Re-register EN, then reinitialize stack and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync / sid conn_req 1
FSK: sid init 2 / sid start / Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: - Join (FSK): Njoin(1, 2) st 2->6 Key check: EN re-registered (Reg: 0)

Step 5

Description:

Check EN sends capability report and receives ACK
BLE: establish BLE connection first: sid conn_req 1

Expected Result:

- ECPFG(2) st 2->5 (Ready to send) - Sent capability and config message:
- [Tx][0:2:2:-] st:0
- ECPFG(5) st 5->6 (Waiting for ACK)
- ECPFG(2) st 6->2 (ACK received, back to check state)
- ECPFG(6) st 2->7 (Updating KV store)

- Received capability and config message: - [Rx][0:2:3:-] err:0 ECPFG(2) st 7->10 (Done, idle)

Key check: Capability report sent and ACK received

Step 6

Description:

Verify capability back to default and matches cloud:
print get_cap 3

Expected Result:

All capabilities should return to default values API response capability matches EN default capability Key check: Capability reset to default after re-registration

CMN-EP-CAPABILITY-THRESHOLD-BV-01: Traffic threshold reset mechanism

Case ID: C103619353

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84746524

Test Steps:

Step 1

Description:

Initialize stack, start, and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 2

Description:

Set configuration via API (low threshold, enable throttling):

```
{
  "deviceId": "{<DEVICE_ID>}",
  "trafficThresholdTable": {
    "tableId": 1,
    "lora": {
      "staticNormalRate": 10, "staticNormalUnit": "MINUTES",
      "staticBurstRate": 1, "staticBurstUnit": "SECONDS",
      "mobileNormalRate": 10, "mobileNormalUnit": "MINUTES",
      "mobileBurstRate": 1, "mobileBurstUnit": "SECONDS",
      "staticMaxPacketPerDay": 3,
      "mobileMaxPacketPerDay": 3
    },
    "fsk": { "packetsPerMin": 3 },
    "ble": { "packetsPerMin": 3 }
  },
  "enableTrafficThrottling": { "tableId": 1, "enableThrottling": true },
  "enableMetricsReport": true,
  "metricsPeriodicity": 6
}
```

Expected Result:

- Received write capability and config message:- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5   (INTERNAL event, CHECK -> SEND_NTFY_READY)

- Response write capability and config message:- [Tx][0:2:3:-] st:0
- ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)

Key check: EN received configuration and sent response

Step 3

Description:

Verify EN applied new traffic config:

print get_cap 3

Expected Result:

Features support ... Traffic throttling enabled 1

Traffic threshold table id 1

Key check: New traffic config applied

Step 4

Description:

Send 3 ULs (BLE: establish connection first):

BLE: sid conn_req 1

sid send TEST (repeat 3 times)

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: All 3 ULs sent successfully

Step 5

Description:

sid stop / sid start, wait for joined (BLE: connect again):

sid stop

sid start

BLE: sid conn_req 1

FSK/LoRa: Wait for join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -

Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 6

Description:

Send 1 UL (should be dropped - threshold not reset by sid stop/start):

sid send TEST

Expected Result:

TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0

CMD: ERR: -20

Key check: UL dropped (threshold not reset by sid stop/start)

Step 7

Description:

Reinitialize stack to reset threshold:

sid deinit

BLE: sid init 1 / sid start / Wait for time sync / sid conn_req 1

FSK: sid init 2 / sid start / Wait for time sync and join

LoRa: sid init 3 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -

Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 8

Description:

Send 3 ULs (threshold reset after reinit):

sid send TEST (repeat 3 times)

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: All 3 ULs sent successfully (threshold reset)

CMN-EP-CAPABILITY-PERIODICITY-BV-01: Set metrics periodicity.

Case ID: C103619354

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497782

Test Steps:

Step 1

Description:

Initialize stack and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join
Please choose either one

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 2

Description:

Check EN sends capability report and receives ACK
BLE: establish BLE connection first: sid conn_req 1

Expected Result:

ECPFG(2) st 2->5 (Ready to send)
Sent capability and config message:
[Tx][0:2:2:-] st:0
ECPFG(5) st 5->6 (Waiting for ACK)
ECPFG(2) st 6->2 (ACK received, back to check state)
ECPFG(6) st 2->7 (Updating KV store)
Received capability and config message:
[Rx][0:2:3:-] err:0
ECPFG(2) st 7->10 (Done, idle)
Key check: Capability report sent and ACK received

Step 3

Description:

Set metricsPeriodicity to 1 and enableMetricsReport to true via API:

```
{
  "deviceId": "{<DEVICE_ID>}",
  "enableMetricsReport": true,
  "metricsPeriodicity": 1
}
```

Expected Result:

- Received write capability and config message:- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event)
ECPFG(2) st 2->5 (Ready to send)

- Response write capability and config message:- [Tx][0:2:3:-] st:0
- ECPFG(5) st 5->10 (Done, idle)

Key check: EN received configuration and sent response

Step 4

Description:

Get capability via API, verify metricsPeriodicity and enableMetricsReport

Expected Result:

enableMetricsReport: true
metricsPeriodicity: 1
Key check: metricsPeriodicity = 1 (6 hours), enableMetricsReport = true

Step 5

Description:

Print capabilities on EN:
print get_cap 3

Expected Result:

metrics periodicity 6 hours
metrics enabled 1
Key check: EN shows correct metrics periodicity

Step 6

Description:

Reboot EN and restart stack, then wait 6 hours:
reboot
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join
Wait 6 hours

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -

After 6 hours: metrics send to cloud
<info> app: ENC-S: M:15 [B:E:2] err 0

Step 7

Description:

Search on Splunk to verify metrics report sent:
index=halo_operations* <sidewalkID> methodIdentifier = "NotifyCapabilityHandler"

Expected Result:

Metrics report found on Splunk
Key check: EN sent metrics report after 6 hours

CMN-EP-CAPABILITY-PERIODICITY-BV-02: Set metrics periodicity when link metrics is disabled

Case ID: C103619355

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497784

Test Steps:

Step 1

Description:

Initialize stack and wait for time sync/join:
BLE: sid init 1 / sid start / Wait for time sync
FSK: sid init 2 / sid start / Wait for time sync and join
LoRa: sid init 3 / sid start / Wait for time sync and join

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -
Join (FSK/LoRa): Njoin(1, 2) st 2->6

Step 2

Description:

Check EN sends capability report and receives ACK
BLE: establish BLE connection first: sid conn_req 1

Expected Result:

ECPFG(2) st 2->5 (Ready to send)
- Sent capability and config message:- [Tx][0:2:2:-] st:0
- ECPFG(5) st 5->6 (Waiting for ACK)
- ECPFG(2) st 6->2 (ACK received, back to check state)
- ECPFG(6) st 2->7 (Updating KV store)

- Received capability and config message:- [Rx][0:2:3:-] err:0
- ECPFG(2) st 7->10 (Done, idle)

Key check: Capability report sent and ACK received

Step 3

Description:

Set metricsPeriodicity to 1 and enableMetricsReport to false via API:

```
{
  "deviceId": "{<DEVICE_ID>}",
  "enableMetricsReport": false,
  "metricsPeriodicity": 1
}
```

Expected Result:

- Received write capability and config message:- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event)
ECPFG(2) st 2->5   (Ready to send)

- Response write capability and config message:- [Tx][0:2:3:-] st:0
- ECPFG(5) st 5->10 (Done, idle)

Key check: EN received configuration and sent response

Step 4

Description:

Get capability via API, verify metricsPeriodicity and enableMetricsReport

Expected Result:

enableMetricsReport: false

metricsPeriodicity: 1

Key check: metricsPeriodicity = 1 (6 hours), enableMetricsReport = false

Step 5

Description:

Print capabilities on EN:

print get_cap 3

Expected Result:

metrics periodicity 6 hours

metrics enabled 0

Key check: EN shows metrics disabled

Step 6

Description:

Reboot EN and restart stack, then wait 6 hours:

reboot

BLE: sid init 1 / sid start / Wait for time sync

FSK: sid init 2 / sid start / Wait for time sync and join

LoRa: sid init 3 / sid start / Wait for time sync and join

Wait 6 hours

Expected Result:

Time sync: EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: -

Step 7

Description:

Search on Splunk to verify metrics report NOT sent:

index=halo_operations* <sidewalkID> methodIdentifier = "NotifyCapabilityHandler"

Expected Result:

No metrics report found on Splunk

Key check: EN did NOT send metrics report (enableMetricsReport = false)

BLE-EP-CAPABILITY-THRESHOLD-BV-01: Verify traffic thresholds for BLE stack

Case ID: C103619361

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497780

Test Steps:

Step 1

Description:

Initialize BLE stack and wait for time sync:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

Step 2

Description:

Establish BLE connection:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
Key check: BLE connected

Step 3

Description:

Check EN capability synced with cloud (KV_STORE updated)

Expected Result:

ECPFG(2) st 7->10 (Done, idle)
Key check: Capability synced

Step 4

Description:

Set configuration via API (low threshold, enable throttling):

```
{
  "deviceId": "{<DEVICE_ID>}",
  "trafficThresholdTable": {
    "tableId": 1,
    "lora": { "staticNormalRate": 10, "staticBurstRate": 1, "mobileNormalRate": 10, "mobileBurstRate": 1,
"staticMaxPacketPerDay": 3, "mobileMaxPacketPerDay": 3 },
    "fsk": { "packetsPerMin": 1 },
    "ble": { "packetsPerMin": 1 }
  },
  "enableTrafficThrottling": { "tableId": 1, "enableThrottling": true }
}
```

Expected Result:

- Received write capability and config message:- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5 (INTERNAL event, CHECK -> SEND_NTFY_READY)

- Response write capability and config message:- [Tx][0:2:3:-] st:0
- ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)

Key check: EN received configuration and sent response

Step 5

Description:

Verify EN applied new traffic config:
print get_cap 3

Expected Result:

Traffic throttling enabled 1
BLE max packets per min 1
Key check: New traffic config applied

Step 6

Description:

Send 4 ULs with BLE connection:
sid conn_req 1
T=0s: sid send TEST (1st UL)
T=25s: sid send TEST (2nd UL)
T=47s: sid send TEST (3rd UL)
T=65s: sid send TEST (4th UL)

Expected Result:

1st UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0
2nd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0
CMD: ERR: -20
3rd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0
CMD: ERR: -20
4th UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0
Key check: UL 2~3 dropped (threshold exceeded), UL 4 success after threshold reset

FSK-EP-CAPABILITY-THRESHOLD-BV-01: Verify traffic threshold for FSK stack.

Case ID: C103619362

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497779

Test Steps:

Step 1

Description:

Initialize FSK stack and wait for time sync and join:
sid init 2
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6

Step 2

Description:

Check EN capability synced with cloud (KV_STORE updated)

Expected Result:

ECPFG(2) st 7->10 (Done, idle)
Key check: Capability synced

Step 3

Description:

Set configuration via API (low threshold, enable throttling):

```
{
   "deviceId": "{<DEVICE_ID>}",
   "trafficThresholdTable": {
      "tableId": 1,
      "lora": { "staticNormalRate": 10, "staticBurstRate": 1, "mobileNormalRate": 10, "mobileBurstRate": 1,
"staticMaxPacketPerDay": 3, "mobileMaxPacketPerDay": 3 },
      "fsk": { "packetsPerMin": 1 },
      "ble": { "packetsPerMin": 1 }
   },
   "enableTrafficThrottling": { "tableId": 1, "enableThrottling": true }
}
```

Expected Result:

- Received write capability and config message:- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5   (INTERNAL event, CHECK -> SEND_NTFY_READY)

- Response write capability and config message:- [Tx][0:2:3:-] st:0
- ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)

Key check: EN received configuration and sent response

Step 4

Description:

Verify EN applied new traffic config:

print get_cap 3

Expected Result:

Traffic throttling enabled 1

FSK max packets per min 1

Key check: New traffic config applied

Step 5

Description:

Send 4 ULs:

T=0s: sid send TEST (1st UL)

T=25s: sid send TEST (2nd UL)

T=47s: sid send TEST (3rd UL)

T=65s: sid send TEST (4th UL)

Expected Result:

1st UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0

2nd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0

CMD: ERR: -20

3rd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0

CMD: ERR: -20

4th UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0

Key check: UL 2~3 dropped (threshold exceeded), UL 4 success after threshold reset

LORA-EP-CAPABILITY-THRESHOLD-BV-01: Verify traffic threshold on static endpoint of LoRa

Case ID: C103619363

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/84497777

Test Steps:

Step 1

Description:

Initialize LoRa stack and wait for time sync and join:
sid init 3
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 2) st 2->6

Step 2

Description:

Check EN capability synced with cloud (KV_STORE updated)

Expected Result:

ECPFG(2) st 7->10 (Done, idle) Key check: Capability synced

Step 3

Description:

Set configuration via API (low threshold, enable throttling): { "deviceId": "{<DEVICE_ID>}", "trafficThresholdTable": { "tableId": 1, "lora": { "staticNormalRate": 10, "staticBurstRate": 1, "mobileNormalRate": 10, "mobileBurstRate": 1, "staticMaxPacketPerDay": 3, "mobileMaxPacketPerDay": 3 }, "fsk": { "packetsPerMin": 1 }, "ble": { "packetsPerMin": 1 } }, "enableTrafficThrottling": { "tableId": 1, "enableThrottling": true } }

Expected Result:

Received write capability and config message:
- [Rx][0:2:1:-] err:0
- ECPFG(8) st 10->2 (SET_CFG event, NTFY_IDLE -> CHECK_FOR_NTFY_STATUS_READY)
- ECPFG(2) st 2->5 (INTERNAL event, CHECK -> SEND_NTFY_READY)
Response write capability and config message:
- [Tx][0:2:3:-] st:0 ECPFG(5) st 5->10 (ON_SENT_STATUS_RECVD, SEND_NTFY_READY -> NTFY_IDLE)
Key check: EN received configuration and sent response

Step 4

Description:

Verify EN applied new traffic config:
print get_cap 3

Expected Result:

Traffic throttling enabled 1
LoRa static burst rate 1 message every 1 seconds

LoRa static max packets per day 3
Key check: New traffic config applied

Step 5

Description:

Send 4 ULs (verify burst rate): T=0s: sid send TEST (1st UL) T=25s: sid send TEST (2nd UL) T=47s: sid send TEST (3rd UL) T=65s: sid send TEST (4th UL)

Expected Result:

1st UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0
2nd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0 CMD: ERR: -20
3rd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0 CMD: ERR: -20
4th UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0
Key check: UL 2~3 dropped (threshold exceeded), UL 4 success after threshold reset

Step 6

Description:

Wait 61 seconds, then send 3 more ULs (verify max packets per day): T=0s: sid send TEST (1st UL) T=65s: sid send TEST (2nd UL) T=75s: sid send TEST (3rd UL)

Expected Result:

1st UL: EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: - LINK_MODE: 0 ADDITIONAL ATTRS: 0
2nd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0 CMD: ERR: -20
3rd UL: TRAFFIC THRESHOLD RATE EXCEEDED on LINK - FOR LIMIT TYPE 0 CMD: ERR: -20
Key check: Max packets per day (3) reached, subsequent ULs dropped

MAC Address Rotation Test Cases

BLE/EP/BCN/MAC/BV/01: Verify MAC address rotation after registration

Case ID: C103619309

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/68193723

Test Steps:

Step 1

Description:

Initialize BLE stack and collect MAC address in beacon:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Step 2

Description:

Wait for EN to register and collect MAC address again

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Key check: EN registered (Reg: 0)

Step 3

Description:

Check MAC address rotation

Expected Result:

mac_addr_1 != mac_addr_2

Key check: MAC address rotated after registration

BLE/EP/BCN/MAC/BV/02: Verify MAC address rotation after time sync

Case ID: C103619310

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69384284

Test Steps:

Step 1

Description:

Initialize BLE stack and collect MAC address before time sync:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 1, Link_Mask: 1
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Key check: EN registered (Reg: 0), time not yet synced (Time: 1)

Step 2

Description:

Wait for time sync, wait for BLE disconnected, then collect MAC address again

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Key check: Time synced (Time: 0), BLE disconnected

Step 3

Description:

Check MAC address rotation

Expected Result:

mac_addr_1 != mac_addr_2

Key check: MAC address rotated after time sync

BLE/EP/BCN/MAC/BV/03: Verify MAC address rotates every 15 minutes

Case ID: C103619311

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69385013

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait another 30s to confirm BLE disconnected:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

Key check: Time synced (Time: 0), BLE disconnected

Step 2

Description:

Monitor 2 MAC address rotations (check every 30s, expect rotation ~15 min)

Expected Result:

Round 1: MAC rotated after ~900s (<mac_addr_1> -> <mac_addr_2>)
Round 2: MAC rotated after ~900s (<mac_addr_2> -> <mac_addr_3>)

Key check: Each rotation interval is between 840s~960s (14~16 min)

Step 3

Description:

Check MAC address rotation interval

Expected Result:

All 3 MAC addresses are unique

Key check: MAC rotates every ~15 minutes

BLE/EP/BCN/MAC/BV/04: Verify MAC address rotation after connection request sent to cloud

Case ID: C103619312

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69385028

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait another 30s to confirm BLE disconnected, collect MAC address:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Step 2

Description:

Establish BLE connection and collect MAC address:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Key check: BLE connection established (BLE: 1)

Step 3

Description:

Check MAC address rotation

Expected Result:

mac_addr_1 != mac_addr_2

Key check: MAC address rotated after connection request

BLE/EP/BCN/MAC/BV/05: Verify MAC address rotation on disconnection from Sidewalk cloud

Case ID: C103619313

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69385032

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait for BLE disconnected, collect MAC address:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Step 2

Description:

Establish BLE connection:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1
BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Key check: BLE connection established (BLE: 1), MAC rotated on connection

Step 3

Description:

Wait for BLE to disconnect, collect MAC address

Expected Result:

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_3>

Key check: BLE disconnected

Step 4

Description:

Check MAC address rotation after disconnection

Expected Result:

mac_addr_2 != mac_addr_3

Key check: MAC address rotated after disconnection

BLE/EP/BCN/MAC/BV/06: Verify MAC address rotation on downlink reception

Case ID: C103619314

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/92223036

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait for BLE disconnected, collect MAC address:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Step 2

Description:

Send DL from AWS IoT

Expected Result:

EVENT SID RECEIVED: TYPE: 2, ID: <id>, LEN: <len>, LINK: 1, LINK_MODE: 1, ACK_REQUESTED: 0, ACK: 0, DUP: 0, RSSI: <rssi>, SNR: <snr>

Key check: DL received through BLE (LINK: 1)

Step 3

Description:

Wait for BLE to disconnect, collect MAC address

Expected Result:

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Step 4

Description:

Check MAC address rotation

Expected Result:

mac_addr_1 != mac_addr_2

Key check: MAC address rotated after downlink reception

BLE/EP/BCN/MAC/BV/07: Verify MAC address rotation on key refresh

Case ID: C103619315

Executable by Customer: No

Preconditions:

https://ring.testrail.net/index.php?/cases/view/92223173

Test Steps:

Step 1

Description:

Setup Nordic-based BT sniffer, connect it to PC and start sniffing packets.

Expected Result:

Sniffer gives possibility to capture BT traffic (beacons) including MAC address sent over the air.

Step 2

Description:

Trigger key refresh

Expected Result:

BLE connection is established and key refresh is completed

Step 3

Description:

Wait until BLE connection is disconnected

Expected Result:

MAC address has rotated and is different than in previous step.

BLE/EP/BCN/MAC/BV/08: Verify MAC address rotation on long lived connection

Case ID: C103619316

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/92419860

Test Steps:

Step 1

Description:

Initialize BLE stack, wait for time sync, wait for BLE disconnected, collect MAC address:
sid init 1
sid start

Expected Result:

CMD: ERR: 0
EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0
BT sniffer captures BLE beacon with MAC address: <mac_addr_1>

Step 2

Description:

Set long lived connection policy:
ble_policy 1 1 1

Expected Result:

CMD: ERR: 0 BLE conn policy 1

Key check: Long lived policy set

Step 3

Description:

Establish BLE connection:
sid conn_req 1

Expected Result:

CMD: ERR: 0
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 1

Key check: BLE connection established (BLE: 1)

Step 4

Description:

Wait 15 minutes and verify BLE is still connected

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 1

Key check: BLE still connected after 15 min (Link_Mask: 1)

Step 5

Description:

Set default policy and wait for BLE to disconnect, collect MAC address:

ble_policy 1 0

Expected Result:

CMD: ERR: 0

EVENT SID STATUS LINK MODE: LORA: 0, FSK: 0, BLE: 0

BT sniffer captures BLE beacon with MAC address: <mac_addr_2>

Step 6

Description:

Check MAC address rotation

Expected Result:

mac_addr_1 != mac_addr_2

Key check: MAC address rotated after long lived connection

Power Optimization Test Cases

FSK/EP/PWR/OPT/BV/01: Verify RX duration can be adjusted

Case ID: C103619364

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672352

Test Steps:

Step 1

Description:

Initialize FSK stack with power optimization profile (rx_duration=6s):
sid init 2
sid option -lp_set 2 630 1 6
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN joined via FSK

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window opening event
Key check: RX windows closed after idle period

Step 3

Description:

Wait for beacon received, then send UL (T=0s):
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 2
Key check: UL sent, T=0

Step 4

Description:

T < 5s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ..., LINK: 2
Key check: DL received within RX duration window

Step 5

Description:

Wait until T >= 7s, then send DL from cloud to EN

Expected Result:

No DL received

Key check: DL not received after RX duration expired (rx_duration=6s + rx_window_interval=0.63s)

FSK/EP/PWR/OPT/BV/02: Verify RX windows after control message sent should not be closed.

Case ID: C103619365

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672354

Test Steps:

Step 1

Description:

Initialize FSK stack with power optimization profile:
sid init 2
sid option -lp_set 2 5040 1 6
sid start
Wait for GWD scan to complete, then disable GW WiFi

Expected Result:

GWD(<n>) st <n>->6
Key check: GWD finished, GW WiFi disabled

Step 2

Description:

Wait for 2 time-sync requests from EN, then enable GW WiFi

Expected Result:

```
[Tx][0:108:0:<n>] (2 times)
Key check: GW WiFi re-enabled before 3rd time-sync request
```

Step 3

Description:

Wait for time-sync response, then disable GW WiFi

Expected Result:

```
[Rx][0:108:3:<n>]
Key check: EN time-synced, GW WiFi disabled again
```

Step 4

Description:

Wait for 2 join requests from EN, then enable GW WiFi

Expected Result:

```
[Tx][0:22F:1:<n>] (2 times)
Key check: GW WiFi re-enabled before 3rd join request
```

Step 5

Description:

Wait for join response

Expected Result:

```
[Rx][0:22F:3:<n>]
Njoin(1, 2) st 2->6
Key check: EN joined
```

Step 6

Description:

Wait 30 seconds, then send DL from cloud to EN

Expected Result:

No EVENT SID RECEIVED log

Key check: RX windows closed after 30s idle, DL not received

FSK/EP/PWR/OPT/BV/03: Verify RX window termina- tion request stop RX window opended at profile 2

Case ID: C103619366

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672357

Test Steps:

Step 1

Description:

Initialize FSK stack with power optimization profile (rx_duration=20s):
sid init 2
sid option -lp_set 2 630 1 20
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN joined via FSK

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window opening event
Key check: RX windows closed

Step 3

Description:

Send UL (T=0s):
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 2
Key check: UL sent, T=0

Step 4

Description:

T < 5s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ..., LINK: 2
Key check: DL received within RX duration window

Step 5

Description:

Execute RX terminate command:
sid option -sub_ghz_ctl 1

Expected Result:

CMD: ERR: 0

Key check: RX windows terminated

Step 6

Description:

T = 10s: Send DL from cloud to EN

Expected Result:

No EVENT SID RECEIVED log

Key check: DL not received, RX windows already terminated

FSK/EP/PWR/OPT/BV/04: Verify RX window termina- tion request should not stop RX window opened at profile 1

Case ID: C103619367

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/95516606

Test Steps:

Step 1

Description:

Initialize FSK stack with profile 1:
sid init 2
sid option -lp_set 1
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN joined via FSK with profile 1

Step 2

Description:

Wait 30 seconds, then send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ..., LINK: 2
Key check: DL received, profile 1 RX windows always open

Step 3

Description:

Execute RX terminate command:
sid option -sub_ghz_ctl 1

Expected Result:

CMD: ERR: -6
Key check: Command fails, profile 1 does not support RX terminate

Step 4

Description:

Send DL from cloud to EN again

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ..., LINK: 2
Key check: DL still received, RX windows not affected by terminate command

FSK/EP/PWR/OPT/BV/05: Verify Beacon interval can be adjusted

Case ID: C103619368

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672360

Test Steps:

Step 1

Description:

Initialize FSK stack with profile 2, beacon interval=3:
sid init 2
sid option -lp_set 2 5040 3 0
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
Njoin(1, 2) st 2->6
Key check: EN joined via FSK

Step 2

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 30240ms (3x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)
Interval ≈ 30240ms ±30ms
Key check: Beacon interval = 3x 10080 ms (30240ms)

Step 3

Description:

Change beacon interval to 2x:
sid option -lp_set 2 5040 2 0
Wait for join response

Expected Result:

```
[Rx][0:22F:3:]
Njoin(1, 2) st 2->6
Key check: Rejoined
```

Step 4

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 20160ms (2x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)
Interval ≈ 20160ms ±20ms
Key check: Beacon interval = 2x 10080 ms (20160ms)

Step 5

Description:

Change beacon interval to 1x:
sid option -lp_set 2 5040 1 0
Wait for join response

Expected Result:

```
[Rx][0:22F:3:]
Njoin(1, 2) st 2->6
Key check: Rejoined
```

Step 6

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 10080ms (1x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)
Interval ≈ 10080ms ±10ms
Key check: Beacon interval = 1x 10080 ms (10080ms)

Step 7

Description:

Switch to profile 1, beacon interval=3:
sid option -lp_set 1 3
Wait for join response

Expected Result:

```
[Rx][0:22F:3:]
Njoin(1, 2) st 2->6
Key check: Rejoined
Key check: Switched to profile 1
```

Step 8

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 30240ms (3x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)
Interval ≈ 30240ms ±30ms
Key check: Beacon interval = 3x

Step 9

Description:

Change beacon interval to 2x:
sid option -lp_set 1 2
Wait for join response

Expected Result:

```
[Rx][0:22F:3:]
Njoin(1, 2) st 2->6
Key check: Rejoined
```

Step 10

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 20160ms (2x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)
Interval ≈ 20160ms ±20ms

Step 11

Description:

Change beacon interval to 1x:

sid option -lp_set 1 1

Wait for join response

Expected Result:

```
[Rx][0:22F:3:]
Njoin(1, 2) st 2->6
Key check: Rejoined
```

Step 12

Description:

Catch 2 consecutive beacon logs, verify interval ≈ 10080ms (1x)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi: (2 times)

Interval ≈ 10080ms ±10ms

Key check: Beacon interval = 1x

FSK/EP/PWR/OPT/BV/06: Verify beacon interval revert to default after 2nd beacon misses

Case ID: C103619369

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672361

Test Steps:

Step 1

Description:

Initialize FSK stack with profile 2, beacon interval=3:
sid init 2
sid option -lp_set 2 5040 3
mac scfg C 1 <GW_Sidewalk_ID>
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 2
EVENT SID STATUS LINK MODE: LORA: 0, FSK: 1, BLE: 0
Key check: EN joined via FSK, beacon interval=3x (30240ms)

Step 2

Description:

After a beacon received, close shielding box door to block GW signal

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi:
Key check: Beacon received, shielding box closed

Step 3

Description:

Wait for 2 consecutive beacon misses:
- Beacon miss 1 at ~30s
- Beacon miss 2 at ~60s

Expected Result:

BCN:bcn miss:1
BCN:bcn miss:2
Key check: 2 beacon misses detected, EN reverts to default interval (10080ms)

Step 4

Description:

Open shielding box door to restore GW signal

Expected Result:

Key check: GW signal restored

Step 5

Description:

Check beacon received at ~70s (default interval after 2nd miss)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi:

Key check: Beacon received at ~70s using default interval (10080ms)

Step 6

Description:

Check beacon received at ~100s (reverted to user-configured interval 30240ms)

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi:

Key check: Beacon interval restored to user-configured 3x (30240ms)

Step 7

Description:

Repeat steps 1-6 with beacon interval=2:

sid option -lp_set 2 5040 2

- Beacon miss 1 at ~20s
- Beacon miss 2 at ~40s
- Beacon received at ~50s (default)
- Beacon received at ~70s (restored to 2x)

Expected Result:

BCN:bcn miss:1

BCN:bcn miss:2

BCN:pkt rcvd (at ~50s)

BCN:pkt rcvd (at ~70s)

Key check: Same behavior with beacon interval=2x (20160ms)

FSK/EP/PWR/OPT/BV/07: Verify DUT can still perform uplinks and downlinks normally after a beacon skipped

Case ID: C103619370

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/93672362

Test Steps:

Step 1

Description:

Initialize FSK stack with profile 2, beacon interval=3, rx_duration=8s:
sid init 2
sid option -lp_set 2 630 3 8
sid start
Wait for time sync and join
Wait for beacon received, record time T

Expected Result:

BCN:pkt rcvd gw_id: ch: rssi:
Njoin(1, 2) st 2->6
Key check: EN joined, beacon received at T=0

Step 2

Description:

Wait until T ≈ 11s (1 beacon skipped), then send UL and wait for DL:
sid send PING

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 2
EVENT SID RECEIVED: TYPE: 3, ID: , LEN: , LINK: 2
Key check: UL/DL successful after 1 beacon skipped

Step 3

Description:

Wait until T ≈ 22s (2 beacons skipped), then send UL and wait for DL:
sid send PING

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 2
EVENT SID RECEIVED: TYPE: 3, ID: , LEN: , LINK: 2
Key check: UL/DL successful after 2 beacons skipped

LORA/EP/PWR/OPT/BV/01: Verify number of RX windows is limited and can be changed

Case ID: C103619371

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69498341

In Device Profile A, after single transmission in Uplink, endpoint should open specified number of RX windows and go to sleep.
The number of windows is defined by 'Number of RX Window' parameter, where allowed values are as below table
<table><thead><tr><th>Field name<th>Current Implement<th>Proposed Changed<tbody><tr>RX window count5, 10, 15, 200~60 (0 is for uplink-only mode)<tr>Window validity period in second25, 50, 75, 1000~300 (0 is for uplink-only mode)EP-> Endpoint
GW -> Gateway
1) GW and EP should be flashed with debug variant build.
2) EP should be flashed with MAC v2 image (sid_dut_v2).
3) EP is registered in Sidewalk network.
4) GW is properly communicating with the Cloud (receives time sync).
5) Initialize Sidewalk stack (LoRa) on EP:
<blockquote> sid init 3
6) Set device profile to A and number of RX windows parameter equal to 5:
<blockquote> sid option -lp_set 128 5
7) Save initial configuration of the devices:
<blockquote> mac gcfg
8) Set LDR-only mode on GW:
<blockquote> mac sdev L
9) Use the same dedicated LDR Channel on both EP and GW:
<blockquote> mac scfg R <LDR_channel>
10) Start Sidewalk stack on EP:
<blockquote> sid start
Notes:
Time should be synchronized on EP, to allow change of link profile.
11) Wait for EP to complete time sync procedure:
[...] <info> app: EVENT SID STATUS: State: 1, Reg: 0, Time: 0, Link_Mask: 4
12) Wait for EP to complete Join procedure:
[...] <info> app: Njoin(1, 4) st 2->6

Test Steps:

Step 1

Description:

Initialize LoRa stack with profile A, rx_window=2:
sid init 3
sid option -lp_set 128 2
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 4) st 2->6
Key check: EN joined via LoRa, rx_window=2 (~10s validity)

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window events

Key check: RX windows closed

Step 3

Description:

Send UL (T=0s):

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4

Key check: UL sent, T=0

Step 4

Description:

T = 3s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: 4

Key check: DL received within RX window (T=3s < 10s)

Step 5

Description:

T = 15s: Send DL from cloud to EN

Expected Result:

No EVENT SID RECEIVED log

Key check: DL not received, RX windows expired (T=15s > 10s)

Step 6

Description:

Change RX windows to 60:

sid option -lp_set 128 60

Wait for join response

Expected Result:

```
[Rx][0:22F:3:<n>]
Njoin(1, 4) st 2->6
Key check: Rejoined with rx_window=60 (~300s validity)
```

Step 7

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window events

Step 8

Description:

Send UL (T=0s):

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4

Key check: UL sent, T=0

Step 9

Description:

T = 3s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: 4

Key check: DL received

Step 10

Description:

T = 293s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: 4

Key check: DL received within extended RX window (T=293s < 300s)

Step 11

Description:

T = 305s: Send DL from cloud to EN

Expected Result:

No EVENT SID RECEIVED log

Key check: DL not received, RX windows expired (T=305s > 300s)

LORA/EP/PWR/OPT/BV/02: Verify number of RX windows is reset after sending an additional uplink

Case ID: C103619372

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/69526193

Test Steps:

Step 1

Description:

Initialize LoRa stack with profile A, rx_window=30:
sid init 3
sid option -lp_set 128 30
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 4) st 2->6

Key check: EN joined via LoRa, rx_window=30 (~150s validity)

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window events

Key check: RX windows closed

Step 3

Description:

Send UL (T=0s):
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4

Key check: UL sent, T=0, RX windows open for ~150s

Step 4

Description:

T = 73s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: 4

Key check: DL received (RX window was reset by additional UL before T=73s)

Step 5

Description:

Send additional UL to reset RX window timer:

sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: <id>, LINK: 4

Key check: UL sent, RX window timer reset to ~150s

Step 6

Description:

T = 222s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: <id>, LEN: <len>, LINK: 4

Key check: DL received within reset RX window

LORA/EP/PWR/OPT/BV/03: Verify RX window termi- nation request stop RX window opened at profile A

Case ID: C103619373

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/95516616

Test Steps:

Step 1

Description:

Initialize LoRa stack with profile A, rx_window=30:
sid init 3
sid option -lp_set 128 30
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 4) st 2->6
Key check: EN joined via LoRa, profile A, rx_window=30 (~150s validity)

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window events
Key check: RX windows closed

Step 3

Description:

Send UL (T=0s):
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 4
Key check: UL sent, T=0

Step 4

Description:

T < 5s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: , LEN: , LINK: 4
Key check: DL received

Step 5

Description:

Execute RX terminate command:
sid option -sub_ghz_ctl 1

Expected Result:

CMD: ERR: 0

Key check: RX terminate command executed

Step 6

Description:

T = 18s: Send DL from cloud to EN

Expected Result:

No EVENT SID RECEIVED log

Key check: DL not received, RX windows terminated by sid option -sub_ghz_ctl 1

LORA/EP/PWR/OPT/BV/04: Verify RX window termi- nation request should not stop RX window opened at profile B

Case ID: C103619374

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/95516617

Test Steps:

Step 1

Description:

Initialize LoRa stack with profile B (infinite RX):
sid init 3
sid option -lp_set 129 0
sid start
Wait for time sync and join

Expected Result:

EVENT SID STATUS: State: 0, Reg: 0, Time: 0, Link_Mask: 4
Njoin(1, 4) st 2->6
Key check: EN joined via LoRa, profile B (infinite RX windows)

Step 2

Description:

Wait 30 seconds for RX windows to close

Expected Result:

No RX window events
Key check: RX windows closed

Step 3

Description:

Send UL (T=0s):
sid send TEST

Expected Result:

EVENT SID SEND STATUS: SID ERR: 0, TYPE: 2, ID: , LINK: 4
Key check: UL sent, T=0

Step 4

Description:

T < 5s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: , LEN: , LINK: 4
Key check: DL received

Step 5

Description:

Execute RX terminate command:
sid option -sub_ghz_ctl 1

Expected Result:

CMD: ERR: 0

Key check: RX terminate command executed

Step 6

Description:

T = 18s: Send DL from cloud to EN

Expected Result:

EVENT SID RECEIVED: TYPE: 3, ID: , LEN: , LINK: 4

Key check: DL still received, profile B not affected by terminate command

BLE Conn Policy Test Cases

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-01: Default connection policy with advertisement interval configured

Case ID: C103619378

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665110

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 0

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 0

Step 2

Description:

EN set minimum advertisement parameters

---

sid option -ble_cfg 1 0 32 0 1600 0

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0

--> [02288644] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters

---

sid option -ble_cfg 1 0 8192 65535 8192 65535

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0

--> [03256879] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-02: Default connection policy with connection parameters configured

Case ID: C103619379

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665111

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 0

Expected Result:

Check connection policy

--> sid option -ble_con 0
--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 0

Step 2

Description:

EN set minimum connection parameters

---

sid option -ble_cfg 1 1 12 12 0 10

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 1
--> [00228962] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

EN set maximum connection parameters

- sid option -ble_cfg 1 1 3200 3200 2 3200

- sid option -ble_cfg 1 1 25 25 499 3200

Expected Result:

Check connection parameters are updated

- --> sid option -ble_cfg 0 1
--> [01971481] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.

--> sid option -ble_cfg 0 1
--> [02006218] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-03: Default connection policy with both advertisement interval and con- nection parameters configured

Case ID: C103619380

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665112

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 0

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 0

Step 2

Description:

EN set minimum advertisement parameters and connection parameters

---

sid option -ble_cfg 1 2 32 0 1600 0 12 12 0 10

Expected Result:

Check connection parameters are updated

--> sid option -ble_cfg 0 2

[02687376] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)

--> [02687376] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters and connection parameters

---

- sid option -ble_cfg 1 2 8192 65535 8192 65535 3200 3200 2 3200

- sid option -ble_cfg 1 2 8192 65535 8192 65535 25 25 499 3200

Expected Result:

Check connection parameters are updated

- --> sid option -ble_cfg 0 2
[03000420] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03000421] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.
--> sid option -ble_cfg 0 2
[03168389] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03168390] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-04: Default connection policy

Case ID: C103619381

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665113

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 0

Step 2

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

- Measure time between beacons for first 30 seconds, average of the interval time should be around 160ms

Step 3

Description:

Check advertisement parameters and connection parameters

Expected Result:

Should be default value

--> sid option -ble_cfg 0 2

[00012909] <info> app: CMD: ERR: 0 BLE adv, fast_int 256(160ms), fast_to 3000(30000ms), slow_int 1600(1000ms), slow_to 0(0ms)

--> [00012909] <info> app: CMD: ERR: 0 BLE conn, min_int: 16(20ms), max_int 60(75ms), sl 0, timeout 400(4000ms)

BLE-EP-CONNECTIONPOLICY-DEFAULT-BV-05: Default connection policy with inactivity timeout

Case ID: C103619382

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665114

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 0

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 0

Step 2

Description:

EN set minimum inactivity timeout

---

sid option -ble_cfg 1 3 1

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 3

--> [00075451] <info> app: CMD: ERR: 0 BLE inactivity timeout 1s

Step 3

Description:

Establish BLE connection, after connected, wait for 2s

---

sid conn_req 1

Expected Result:

BLE is disconnected after 1s

Step 4

Description:

EN set maximum inactivity timeout

---

sid option -ble_cfg 1 3 3600

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 3

--> [00075451] <info> app: CMD: ERR: 0 BLE inactivity timeout 3600s

Step 5

Description:

Establish BLE connection, after connected, wait for 3601s

---

sid conn_req 1

Expected Result:

BLE is disconnected after 3600s

BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-01: Long lived connection policy with advertisement interval configured

Case ID: C103619385

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666786

Test Steps:

Step 1

Description:

EN set to long live policy

- sid option -ble_con 1 1 0 (Power Optimized)

- sid option -ble_con 1 1 1 (Latency Optimized)

Expected Result:

Check connection policy

Power Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 0
Latency Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 1

Step 2

Description:

EN set minimum advertisement parameters

---

sid option -ble_cfg 1 0 32 0 1600 0

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0
--> [02288644] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters

---

sid option -ble_cfg 1 0 8192 65535 8192 65535

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0

--> [03256879] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-02: Long lived connection policy with connection parameters configured

Case ID: C103619386

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666787

Test Steps:

Step 1

Description:

EN set to long live policy

---

1. sid option -ble_con 1 1 0 (Power Optimized)

2. sid option -ble_con 1 1 1 (Latency Optimized)

Expected Result:

Check connection policy

Power Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 0

Latency Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 1

Step 2

Description:

EN set minimum connection parameters

---

sid option -ble_cfg 1 1 12 12 0 10

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 1
--> [00228962] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

EN set maximum connection parameters

---

1. sid option -ble_cfg 1 1 3200 3200 2 3200

2. sid option -ble_cfg 1 1 25 25 499 3200

Expected Result:

Check connection parameters are updated

1.

--> sid option -ble_cfg 0 1

--> [01971481] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.

--> sid option -ble_cfg 0 1

--> [02006218] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-03: Long lived connection policy with both advertisement interval and connection parameters configured

Case ID: C103619387

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90665118

Test Steps:

Step 1

Description:

EN set to long live policy

- sid option -ble_con 1 1 0 (Power Optimized)

- sid option -ble_con 1 1 1 (Latency Optimized)

Expected Result:

Check connection policy

Power Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 0
Latency Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 1

Step 2

Description:

EN set minimum advertisement parameters and connection parameters

---

sid option -ble_cfg 1 2 32 0 1600 0 12 12 0 10

Expected Result:

Check connection parameters are updated

--> sid option -ble_cfg 0 2
[02687376] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)
--> [02687376] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters and connection parameters

---

- sid option -ble_cfg 1 2 8192 65535 8192 65535 3200 3200 2 3200

- sid option -ble_cfg 1 2 8192 65535 8192 65535 25 25 499 3200

Expected Result:

Check connection parameters are updated

- --> sid option -ble_cfg 0 2
[03000420] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03000421] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.
--> sid option -ble_cfg 0 2
[03168389] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03168390] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-LONGLIVED-BV-04: Long lived connection policy with inactivity timeout

Case ID: C103619388

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666797

Test Steps:

Step 1

Description:

EN set to long live policy

---

1. sid option -ble_con 1 1 0 (Power Optimized)

2. sid option -ble_con 1 1 1 (Latency Optimized)

Expected Result:

Check connection policy

Power Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 0

Latency Optimized
--> sid option -ble_con 0
[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1
--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 1

Step 2

Description:

EN set minimum inactivity timeout

---

sid option -ble_cfg 1 3 1

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 3
--> [00075451] <info> app: CMD: ERR: 0 BLE inactivity timeout 1s

Step 3

Description:

Establish BLE connection, after connected, wait for 2s

---

sid conn_req 1

Expected Result:

BLE is not disconnected

Step 4

Description:

EN set to default policy and wait for 2s

---

sid option -ble_con 1 0

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 0

BLE is disconnected after 2s

Step 5

Description:

EN set to long live policy

---

1. sid option -ble_con 1 1 0 (Power Optimized)

2. sid option -ble_con 1 1 1 (Latency Optimized)

Expected Result:

Check connection policy

Power Optimized

--> sid option -ble_con 0

[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1

--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 0

Latency Optimized

--> sid option -ble_con 0

[00062756] <info> app: CMD: ERR: 0 BLE conn policy 1

--> [00062756] <info> app: CMD: ERR: 0 Long lived connection optimized for 1

Step 6

Description:

Establish BLE connection, after connected, wait for 2s

---

sid conn_req 1

Expected Result:

BLE is not disconnected

Step 7

Description:

EN set to optimal advertising policy and wait for 2 sec

---

sid option -ble_con 1 2

Expected Result:

1. Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 2

---

2. BLE disconnected after 2 sec

BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING-BV-01: Optimal advertising connection policy with advertisement interval configured

Case ID: C103619392

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666816

Test Steps:

Step 1

Description:

EN set to optimal advertising policy

---

sid option -ble_con 1 2

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 2

Step 2

Description:

EN set minimum advertisement parameters

---

sid option -ble_cfg 1 0 32 0 1600 0

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0

--> [02288644] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters

---

sid option -ble_cfg 1 0 8192 65535 8192 65535

Expected Result:

Check advertisement parameters are updated

--> sid option -ble_cfg 0 0

--> [03256879] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-02: Optimal advertising connection policy with connec- tion parameters configured

Case ID: C103619393

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666817

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 2

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 2

Step 2

Description:

EN set minimum connection parameters

---

sid option -ble_cfg 1 1 12 12 0 10

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 1

--> [00228962] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

EN set maximum connection parameters

---

1. sid option -ble_cfg 1 1 3200 3200 2 3200

2. sid option -ble_cfg 1 1 25 25 499 3200

Expected Result:

Check connection parameters are updated

1.
--> sid option -ble_cfg 0 1
--> [01971481] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.

--> sid option -ble_cfg 0 1
--> [02006218] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-03: Optimal advertising connection policy with both advertisement interval and connection parameters configured

Case ID: C103619394

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666818

Test Steps:

Step 1

Description:

EN set to default policy

---

sid option -ble_con 1 2

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 2

Step 2

Description:

EN set minimum advertisement parameters and connection parameters

---

sid option -ble_cfg 1 2 32 0 1600 0 12 12 0 10

Expected Result:

Check connection parameters are updated

--> sid option -ble_cfg 0 2

[02687376] <info> app: CMD: ERR: 0 BLE adv, fast_int 32(20ms), fast_to 0(0ms), slow_int 1600(1000ms), slow_to 0(0ms)

--> [02687376] <info> app: CMD: ERR: 0 BLE conn, min_int: 12(15ms), max_int 12(15ms), sl 0, timeout 10(100ms)

Step 3

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 20ms

Step 4

Description:

EN set maximum advertisement parameters and connection parameters

---

- sid option -ble_cfg 1 2 8192 65535 8192 65535 3200 3200 2 3200

- sid option -ble_cfg 1 2 8192 65535 8192 65535 25 25 499 3200

Expected Result:

Check connection parameters are updated

- --> sid option -ble_cfg 0 2
[03000420] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03000421] <info> app: CMD: ERR: 0 BLE conn, min_int: 3200(4000ms), max_int 3200(4000ms), sl 2, timeout 3200(32000ms)

2.
--> sid option -ble_cfg 0 2
[03168389] <info> app: CMD: ERR: 0 BLE adv, fast_int 8192(5120ms), fast_to 65535(655350ms), slow_int 8192(5120ms), slow_to 65535(655350ms)
--> [03168390] <info> app: CMD: ERR: 0 BLE conn, min_int: 25(31ms), max_int 25(31ms), sl 499, timeout 3200(32000ms)

Step 5

Description:

Use sidewalk BLE sniffer to catch beacons

Expected Result:

Measure time between beacons for 35 seconds, average of the interval time should be around 5120ms

BLE-EP-CONNECTIONPOLICY-OPTIMALADVERTISING- BV-04: Optimal advertising connection policy with inactiv- ity timeout

Case ID: C103619395

Executable by Customer: Yes

Preconditions:

https://ring.testrail.net/index.php?/cases/view/90666819

Test Steps:

Step 1

Description:

EN set to optimal advertising connection policy

---

sid option -ble_con 1 2

Expected Result:

Check connection policy

--> sid option -ble_con 0

--> [02040569] <info> app: CMD: ERR: 0 BLE conn policy 2

Step 2

Description:

EN set minimum inactivity timeout

---

sid option -ble_cfg 1 3 1

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 3

--> [00075451] <info> app: CMD: ERR: 0 BLE inactivity timeout 1s

Step 3

Description:

Establish BLE connection, after connected, wait for 2s

---

sid conn_req 1

Expected Result:

BLE is disconnected after 1s

Step 4

Description:

EN set maximum inactivity timeout

---

sid option -ble_cfg 1 3 3600

Expected Result:

Check connection parameters are updated

---

sid option -ble_cfg 0 3

--> [00075451] <info> app: CMD: ERR: 0 BLE inactivity timeout 3600s

Step 5

Description:

Establish BLE connection, after connected, wait for 3601s

---

sid conn_req 1

Expected Result:

BLE is disconnected after 3600s