Checking the OpenLCB Event Transport Protocol Standard

March 12, 2025

1 Introduction

This note documents the procedure for checking an OpenLCB implementation against the Event Transport Standard.

The checks are traceable to specific sections of the Standard.

The checking assumes that the Device Being Checked (DBC) is being exercised by other nodes on the message network, e.g. is responding to enquiries from other parts of the message network. .

2 Event Transport Procedure

A node which does not self-identify in PIP that it supports Event Transfer will be considered to have passed these checks.

A node which does self-identify in PIP that it supports Event Transfer is expected to consume or produce at least one event. The checks are structured to check for that.

Note: Proper handling of known events should be addressed.

Note: This does not address the proper use of Unique IDs for Event IDs.

2.1 Identify Events Addressed

This section checks the addressed interaction in Standard section 6.2 and the message formats in Standard section 4.3 through 4.8.

The check starts by sending an Identify Events message addressed to the DBC. It then checks

1. That one or more Producer Identified, Producer Range Identified, Consumer Identified and/or Consumer Range Identified messages are returned,

2. That those show the DBC node as their source.

During this process, any received PCER messages with the DBC as their source are compared against previously received Producer Identified and Producer Range Identified messages. If the Event ID carried by the PCER message has not been identified by a previous message, the check fails.

2.2 Identify Events Global

This section checks the unaddressed (global) interaction in Standard section 6.2 and the message formats in Standard section 4.3 through 4.8.

The check starts by sending an Identify Events unaddressed (global) message.

While acquiring the replies to the global message, any received PCER messages with the DBC as their source are compared against previously received Producer Identified and Producer Range Identified messages. If the Event ID carried by the PCER message has not been identified by a previous message, the check fails.

It then checks

1. That one or more Producer Identified, Producer Range Identified, Consumer Identified and/or Consumer Range Identified messages are returned,
2. That those show the DBC node as their source,
3. That these identify the same events and ranges produced and consumed as the addressed form of the Identify Events message.

   While doing the comparison, any Producer Identified or Consumer Identified messages that lie within the range defined by a previous Producer Range Identified or Consumer Range Identified message, respectively, are ignored as redundant; those do not need to exactly match. [1]

2.3 Identify Producer

This section checks the interaction in Standard section 6.3, and the message formats in Standard section 4.5 through 4.7.

The check proceeds by sending an Identify Producer messages for each of the zero or more individual event IDs identified as being produced by an Identify Events message addressed to the DBC. If there are none of these, this check passes. If there are one or more, it then checks:

[1] Fast clocks, for example, may identify the full range of events they produce, then also identify specific time-varying events within those ranges.

1. That exactly one Producer Identified reply is received for each Identify Producers message sent.
2. That those show the DBC node as their source,
3. That these identify the same event ID as the corresponding Identify message.

Other messages received during this test are ignored.

2.4 Identify Consumer

This section checks the interaction in Standard section 6.4, and the message formats in Standard section 4.2 through 4.4.

The check proceeds by sending an Identify Consumers message for each of the zero or more individual event IDs identified as being consumed by an Identify Events message addressed to the DBC. If there are none of these, this check passes. If there are one or more, it then checks:

1. That exactly one Consumer Identified reply is received for each Identify Consumers message sent.
2. That those show the DBC node as their source,
3. That these identify the same event ID as the corresponding Identify message.

Other messages received during this test are ignored.

2.5 Initial Advertisement

Follow the prompts when asked to reset or otherwise initialize the DBC.

This section's checks the interaction in the preamble to Standard section 6, and the messages in Standard sections 4.1, 4.3, 4.4, 4.6 and 4.7.

Note: There's no requirement that the Identify Producer messages be sent immediately, only that they be sent before the events are produced. Nodes typically send them immediately, and that's what this checks.

This check starts by restarting the node, which causes a transition to Initialized state. That is then followed by the node identifying events that it will produce and consume by appropriate messages. This then checks:

1. That the Producer Identified, Producer Identified Range, Consumer Identified and Consumer Identified Range messages produced at node startup are the same as the ones emitted in response to an addressed Identify Events,
2. That those messages show the DBC as their source.

During this process, any received PCER messages with the DBC as their source are compared against previously received Producer Identified and Producer Range Identified messages. If the Event ID carried by the PCER message has not been identified by a previous message, the check fails.

2.6 Events With Payload

This section checks that PCER Events With Payload can successfully be processed by the node.

Because there is no standard way to force an arbitrary node to produce or consume PCER Events With Payload, this only checks that the node can coexist with PCER Events With Payload on the network.

1. Produce a PCER Event with Payload of 12 bytes, e.g the initial 8 bytes and an additional 4 in a 2nd frame.
2. Produce a PCER Event with Payload of 16 bytes.
3. Produce a PCER Event with Payload of 20 bytes.
4. Produce a PCER Event with Payload of 24 bytes.
5. Produce a PCER Event with Payload of 256 bytes.

After those have been sent, the check waits 30 seconds to see if there are any additional transmissions, including but not limited to an alias-allocation series or a Node Initialized message. If any are seen, that indicates the node has reinitialized and the check fails.

The operator is also prompted to check whether the node indicates that it has restarted, e.g. via LEDs or a screen. If it has, the check fails.

2.7 Learn Event

This section checks that the DBC is compatible with the Learn Event messages in Sections 4.9 and 6.5 of the Standard.

Because there is no standard way to force an arbitrary node to send or process Learn Event messages, this only checks that the node can coexist with Learn Event messages on the network.

1. Send a Learn Event message with an Event ID formed from the checker's Unique ID (Node ID).

After those have been sent, the check waits 30 seconds to see if there are any additional transmissions, including but not limited to an alias-allocation series or a Node Initialized message. If any are seen, that indicates the node has reinitialized and the check fails.

The operator is also prompted to check whether the node indicates that it has restarted, e.g. via LEDs or a screen. If it has, the check fails.