Checking the OpenLCB Train Search Protocol Standard

July 22, 2024

1 Introduction

This note documents the procedure for checking an OpenLCB implementation against the Train Search Protocol 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.

The Train Search Protocol specifies the behavior of both providers and requestors of train search information. Only the provider is checked here.

2 Train Search Protocol Provider Procedure

These checks require the attached node to, when appropriately commanded, to create a virtual train node with address 12.

2.1 Virtual Train Node creation check

This checks the interaction in section 6.2 of the Standard followed by checking the interaction in section 6.1 of the Standard.

This check starts with the device under test having not created a virtual node with address 12. For some devices, doing a power cycle will drop all virtual nodes and prepare the device for this test.

1. Send a search Identify Producer with search nibbles 0x12.FF.FF and not-Allocate, Exact, Address Only, Any/Default protocol 0x60.
2. Wait one second for a reply. There should not be one.

3. Send a search Identify Producer with search nibbles 0x12.FF.FF and Allocate, Exact, Address Only, Any/Default protocol 0xE0.
4. Wait up to one second for a reply. A Producer Identified Valid for the same event ID should be received.
5. Send a search Identify Producer with search nibbles 00x12.FF.FF and not Allocate, Exact, Address Only, Any/Default protocol 0x60.
6. Wait up to one second for a reply. A Producer Identified Valid for the same event ID should be received.

2.2 Match partial values check

This is checking aspects of the Event Identifier matching algorithm described in section 6.3 of the Standard.

1. Ensure node with address 12 exists; may have been created in the previous check.
   (a) Send a search Identify Producer with search nibbles 0x12.FF.FF and Allocate, Exact, Address Only, Any/Default protocol 0xE0.
   (b) Wait up to one second for a reply. A Producer Identified Valid for the same event ID should be received.
2. Search not-allocate for partial match on 2; should exist
   (a) Send a search Identify Producer with search nibbles 0x2F.FF.FF and not-Allocate, Exact, Address Only, Any/Default protocol 0x60.
   (b) Wait up to one second for a reply. A Producer Identified Valid for the same event ID should be received.
3. Search not-allocate for partial match on 1; should exist
   (a) Send a search Identify Producer with search nibbles 0x1F.FF.FF and not-Allocate, Exact, Address Only, Any/Default protocol 0x60.
   (b) Wait up to one second for a reply. A Producer Identified Valid for the same event ID should be received.

(Should partial match on 0x1F2FFF exist? 0x2F1FFF?)

2.3 Reserved address values check

This is checking the reserved values found in Table 2 of section 5.2 of the Standard.

Having 0xA through 0xE in a search nibble should result in no matches.

1. Send a search Identify Producer with search nibbles 0x0B.FF.FF and Allocate, Exact, Address Only, Any/Default protocol 0xE0.
2. Wait one second for a reply. There should not be one.

2.4 Reserved address values check

This is checking the reserved values found in Table 4 of section 5.2 of the Standard.

Having reserved values in the protocol nibble should result in no matches.

1. Send a search Identify Producer with search nibbles 0x12.FF.FF and protocol byte Allocate plus reserved values 0xF0.
2. Wait one second for a reply. There should not be one.
3. Send a search Identify Producer with search nibbles 0x12.FF.FF and protocol byte Allocate plus reserved values 0xF8.
4. Wait one second for a reply. There should not be one.
5. Send a search Identify Producer with search nibbles 0x12.FF.FF and protocol byte Allocate plus reserved values 0xE3.
6. Wait one second for a reply. There should not be one.