

OpenLCB Test plan for the Datagram Transport Protocol

The OpenLCB Group

February 9, 2024

1 Introduction

This note documents the procedure for testing an OpenLCB implementation against the Datagram Transport Standard.

The tests are traceable to specific sections of the Standard.

The testing assumes that the Device Under Test (DUT) is being exercised by other nodes on the message network, e.g. is responding to enquiries from other parts of the message network.

1.1 Required Equipment

See the separate “Installing the OpenLCB Test Software” document for initial installation and set up of the test program.

If a direct CAN connection will be used, a supported USB-CAN adapter ¹ is required. Connect the adapter to the DUT using a single UTP cable and connect two CAN terminators.

Provide power to the DUT using its recommended method.

2 Set Up

The following steps need to be done once to configure the test program:.

1. Start the test configuration program.
2. Select “Set Up DUT”.
3. Get the Node ID from the DUT²

¹See “Installing the OpenLCB Test Software”

²Where do we require this to be marked on a node?

4. Enter that Node ID into the program.
5. Configure the test program for the USB-CAN adapter's device address or the TCP hostname and port.
6. Quit the test program and reply "Y" to "Save configuration?" when prompted.

The following steps need to be done at the start of each testing session.

1. Check that the DUT is ready for operation.
2. Start the test program.

3 Datagram Transport Procedure

Select "Datagram testing" in the test program, then select each section below in turn. Follow the prompts for when to check outputs against the node documentation.

Note that this process is unable to test datagrams from the DUT to the tester. There is no standard way to elicit those. They will be tested as part of the Memory Configuration Protocol testing, if applicable.

A node which does not self-identify in PIP that it supports datagram transfer will be deemed to have passed these tests. ³

3.1 Datagram Reception

This tests the messages in Standard sections 4.1 and 4.2 or 4.3, including the error code constraints in Standard section 4.3.1 ⁴ and the interactions in sections 6.1 or 6.2.

The tester will send a datagram to the DUT. The DUT in turn will either accept the datagram (sections 4.2 and 6.1) or reject it (sections 4.3 and 6.2). Either response is acceptable.

This section's tests cover:

1. If the datagram is accepted,
 - (a) That the Datagram OK message is received,
 - (b) That the Datagram OK message has proper source and destination node IDs,
 - (c) that the Datagram OK message contains exactly 1 byte of flags,
 - (d) that the 0x70 bits of the flags are zeros.

³Using the -p option or setting the checkpip default value False will skip this check.

⁴This section references the Message Transport Standard, section 3.5.5

2. if the datagram is rejected,
 - (a) That the Datagram Rejected message is received,
 - (b) That the Datagram Rejected message has proper source and destination node IDs,
 - (c) that the Datagram Rejected message contains at least two bytes of error code,
 - (d) that the 0xF000 bits of the error code are either 0x1000 or 0x2000,
 - (e) that the 0x0F00 bits of the error code are zeros.

This process is repeated for datagrams of length 1, 10 and 72 bytes to exercise the full range of datagram framing.