OpenLCB Standard	
OpenLCB-CAN Datagram Transport	
Feb 2, 2013	Preliminary

1 Introduction (Informative)

This specification defines the protocol for transporting OpenLCB datagrams via CAN segments.

2 Intended Use (Informative)

The datagram transport protocol is intended to efficiently transfer small amounts (0-72 bytes) of data reliably between two OpenLCB nodes. It allows for management of overlapping independent transmissions.

The datagram transport protocol relies on the underlying OpenLCB message transport protocol for reliable sequenced communications.

This document describes the required message formats for datagram transport. §4 gives an overview of the message types with an abstract numeric description intended as a normative guide to the construction of concrete message types over specific physical transport media. §§7–ff. describe in concrete detail the implementation of the datagram transport message formats for the specific physical transport media that have been adopted as normative standards.

3 References and Context (Normative)

This specification is in the context of the following OpenLCB-CAN Standards:

- The OpenLCB Message Network Standard, which defines the basic messages and how they interact. Higher-level protocols are based on this message network, but are defined elsewhere. The Message Network Standard defines the Initialized node state which is referenced here.
- The OpenLCB-CAN Frame Transport Standard, which specifies the use and format of CAN frames for OpenLCB communications.

4 Message Formats (Normative)

In the following, the “Common MTI” column specifies the the MTI value to be used when communicating in OpenLCB common format. The Common MTI is an abstract numeric description intended as a normative guide to the construction of concrete message formats over specific physical transport media.

4.1 Datagram Content

Name	Dest ID	Event ID	Common MTI	Data Content
Datagram Content	Y	N	0x1C48	0-72 bytes

4.2 Datagram Received OK

Name	Dest ID	Event ID	Common MTI	Data Content
Datagram Received OK	Y	N	0x0A28	

4.3 Datagram Rejected

Name	Dest ID	Event ID	Common MTI	Data Content
Datagram Rejected	Y	N	0x0A48	Error Code

Nodes shall accept and process Datagram Rejected messages that do not contain a full data code. Missing error code bits are to be interpreted as zero.

4.3.1 Error Codes

The Error Code field contains 16 bits.

Zero or one of the two following bits shall be set:

- 0x1000 Permanent error – The received datagram will always invoke the same error.
- 0x2000 Resend OK – The error condition is likely to be temporary, so a resend of the original datagram may be attempted.

When the “Resend OK” bit is set, the following bits are defined:

- 0x4000 Transport error – The error could have been due to a failure in message/frame transport.
- 0x0020 Buffer unavailable - The node wasn't able to receive the datagram because of a shortage of buffers.
- 0x0040 Out of order - An internal inconsistency was found in the OpenLCB message sequence and/or in the CAN frames making up a datagram.

When the “Permanent error” bit is set, the following bits are defined:

- 0x0020 – Source not permitted – this node will not accept datagrams of this type from the transmitting node

- 0x0040 – Datagrams not accepted – this node will not accept datagrams of this type from any node

Nodes may, but are not required to, use the low five bits of the error code field to define specific error codes in concert with the flag bits defined above.

All other bits and bit combinations are reserved.

5 States (Normative)

The common OpenLCB datagram protocol has no formal states.

6 Interactions (Normative)

A node that receives a valid Datagram Content message shall send either a Datagram Received OK or Datagram Rejected message in reply. A node the receives a Datagram Content message that does not comply with this Standard may, but is not required to, reply with a Datagram Rejected message.

6.1 Normal Transmission

Normal transmission consists of the transmitting node sending a Datagram Content message to the receiving node, followed by the receiving node sending a Datagram Received OK message to the transmitting node.

6.2 Rejected Transmission

After the transmitting node sends a Datagram Content message to the receiving node, the receiving node may send a Datagram Rejected message to the transmitting node.

If a receiving node receives a 2nd Datagram Content message before sending a reply to the the 1st Datagram Content message, it may, but is not required to, reject the 2nd Datagram by sending a Datagram Rejected message with the Transport error and Resend OK error flag bits set.

Upon receipt of a Datagram Rejected message with the Resend OK bit sent, the original transmitting node may resend the same Datagram Content message, or may abandon the transmission attempt.

Upon receipt of a Datagram Rejected message with the Resend OK bit resent, the original transmitting node shall abandon the transmission attempt and not resend the original Datagram Content message.

7 Adaptation to CAN Transport (Normative)

This section describes the CAN implementation of the datagram transport message formats.

7.1 CAN Message Formats

The OpenLCB-CAN Frame Transport Standard and OpenLCB Message Network Standard define how OpenLCB messages are carried across CAN networks. Following those specifications, the Datagram Transport messages used on CAN are as defined in the following table.

Name	Dest ID	Event ID	Header Format	Data-part Content
Datagram Content	Y	N	0x1Add,dsss[1] — Single[2] 0x1Bdd,dsss — First 0x1Cdd,dsss — Middle 0x1Ddd,dsss — Last	0–8 bytes
Datagram Received OK	Y	N	0x19A2,8sss	0xfddd[3]
Datagram Rejected	Y	N	0x19A4,8sss	0xfddd, Error Code

7.2 CAN States

A node implementing the OpenLCB-CAN protocol shall maintain a Datagram-started state for each datagram that it is receiving as a sequence of frames. If the node receives multiple overlapping datagrams, the states shall be independent.

7.3 CAN Interactions

7.3.1 Normal Transmission

Normal transmission of a datagram over CAN consists of the transmitting node sending the Datagram Content message using one of two sequences of Datagram frames:

- One Datagram Content Single Frame
- One Datagram Content First Frame, followed by zero or more Datagram Content Middle Frame, followed by one Datagram Content Last Frame

A node shall not interleave transmission of frames from more than one datagram. A node shall not transmit frames with lower CAN priority between the frames making up a datagram. A node may, but is not required to, transmit frames with higher CAN priority between the frames making up a datagram.

A receiving node receiving either of the above sequences shall send either a Datagram Received OK or Datagram Rejected message in reply.

7.3.2 Rejected Transmission

If a receiving node receives a sequence of Datagram frames other than one of

- One Datagram Content Single Frame
- One Datagram Content First Frame, followed by zero or more Datagram Content Middle Frame, followed by one Datagram Content Last Frame

the receiving node shall send a Datagram Rejected message with the Transport error and Resend OK bits set.

---

[1]sss —The 12-bit source alias field

[2]Because CAN frames are limited to 8 bytes, datagrams larger than 8 bytes must be broken up among multiple messages. Thus, four distinct message types are defined to aid in flow control.

[3]fddd — First two bytes of the data-part, representing the 4-bit flag field and 12-bit destination Alias. See the OpenLCB-CAN Frame Transport Standard.

Table of Contents