OpenLCB Standard	
OpenLCB-CAN Datagram Transport	
Jan 25, 2013	Preliminary

1 Introduction (Informative)

This specification describes 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 ~~Specifications~~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 Frame Transport Standard, which specifies the use and format of CAN frames for OpenLCB communications.
- ~~The OpenLCB Message Network Standard, which specifies the OpenLCB message transport mechanism.~~

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. ~~The "CAN format" column specifies the format of CAN frames when communicating in CAN format.~~

4.1 Datagram Content

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

4.1.1 ~~CAN-Datagram Content Single Frame~~

~~Name~~	~~Dest ID~~	~~Event ID~~	~~Common MTI~~	~~CAN format~~	~~Data Content~~
~~Datagram Content Single Frame~~	~~N~~	~~Y~~	~~N/A~~	~~0x1Add,dsss~~	~~0-8 bytes~~

4.1.2 ~~CAN-Datagram Content First Frame~~

~~Name~~	~~Dest ID~~	~~Event ID~~	~~Common MTI~~	~~CAN format~~	~~Data Content~~
~~Datagram Content First Frame~~	~~Y~~	~~N~~	~~N/A~~	~~0x1Bdd,dsss~~	~~0-8 bytes~~

4.1.3 ~~CAN-Datagram Content Middle Frame~~

~~Name~~	~~Dest ID~~	~~Event ID~~	~~Common MTI~~	~~CAN format~~	~~Data Content~~
~~Datagram Content Middle Frame~~	~~Y~~	~~N~~	~~N/A~~	~~0x1Cdd,dsss~~	~~0-8 bytes~~

4.1.4 ~~CAN Datagram Content Last Frame~~

~~Name~~	~~Dest ID~~	~~Event ID~~	~~Common MTI~~	~~CAN format~~	~~Data Content~~
~~Datagram Content Last Frame~~	~~Y~~	~~N~~	~~N/A~~	~~0x1Ddd,dsss~~	~~0-8 bytes~~

4.2 Datagram Received OK

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

4.3 Datagram Rejected

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

Nodes must 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. The following bits are independent flags:

- 0x1000 Permanent error – The received datagram will always invoke the same error
- 0x2000 Resend OK – The error condition may be cleared, so a resend of the original datagram can be attempted
- 0x4000 Transport error – The error could have been due to a failure in message/frame transport

Specific values for the 16-bit field:

- 0x1020 – Permanent error, source not permitted – this node will not accept datagrams of this type from the transmitting node
- 0x1040 – Permanent error, datagrams not accepted – this node will not accept datagrams of this type from any node

- 0x2000 – Temporary condition, please resend – some temporary condition in the receiving node prevented receiving the datagram, but there's reason to believe a resend will succeed.
- 0x6000 – Transport error, please resend – some temporary condition, possibly in the transport layer, prevented receiving the datagram, but there's reason to believe a resend will succeed.

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 independent flag bits defined above.

All other bits and bit combinations are reserved.

5 States (Normative)

The common OpenLCB datagram protocol has no formal states.

A node implementing the OpenLCB-CAN protocol must 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 must be independent.

6 Interactions (Normative)

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. The receiving node shall send either a Datagram Received OK or Datagram Rejected message in reply.

6.1.1 ~~CAN Protocol~~

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

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. The receiving node shall send either a Datagram Received OK or Datagram Rejected message in reply.

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 ~~Out-of-Sequence~~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.

6.2.1 ~~CAN Protocol~~

~~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 Out-of-Sequence error bit set.~~

7 Adaptation to CAN Transport

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

Due to the limitations of CAN, namely a 29-bit header and 8-byte data-part, the format of CAN Event messages have been adapted, as per 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

	~~Dest ID~~	~~Event ID~~	~~Simple Node~~	~~Header Format~~	~~Data-part Content~~
~~Datagram Content~~	~~Y~~	~~N~~	~~N~~	~~0x195B,4sss 1~~	~~EventID~~

[1] sss —The 12-bit source alias field

[2] Quality — 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] 0ddd — First two bytes of the data-part, representing the 12-bit destination Alias. See the OpenLCB-CAN Frame Transport Standard.

Identify Consumer	N	Y	Y	0x198F,4sss	Event ID
Consumer Identified	N	Y	N	0x194C,4sss - Valid[2] 0x194C,5sss - Invalid 0x194C,6sss - Reserved 0x194C,7sss - Unknown	Event ID
Consumer Range Identified	N	Y	N	0x194A,4sss	EventID Range
Identify Producer	N	Y	Y	0x1991,4sss	Event ID
Producer Identified	N	Y	N	0x1954,4sss - Valid 0x1954,5sss - Invalid 0x1954,6sss - Reserved 0x1954,7sss - Unknown	Event ID
Producer Identified Range	N	Y	N	0x1952,4sss	EventID Range
Identify Events	N	N	Y	0x1997,0sss	
	Y	N	N	0x1996,8sss	0ddd[3]

Footnotes:
[1] sss - The 12-bit source alias field.
[2] Quality - Producer and Consumer Identified messages are flagged with the validity of the reply.
[3] 0ddd - First two bytes of the data-part, representing the 12-bit destination Alias, the top 4 bits are zero.

7.1 Interactions

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

7.2 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.

Table of Contents