1 Introduction (Informative)

This specification defines the protocol for transmitting unidirectional streams of data between two nodes on an OpenLCB bus.

2 Intended Use (Informative)

5 Streaming can be used to move large amounts of data (kilobytes and up) across a OpenLCB implementation. Streaming transport can also be used when the data transfer is unknown or indefinite in its volume or persistence.

3 References and Context (Normative)

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

10 • The OpenLCB Unique Identifiers Standard, which defines the method for allocating unique identifiers that form the basis of Stream Content UIDs, defined below.

• 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 encoding of stream data messages
15 on the CAN physical layer, which is referenced here. The Message Network Standard also defines the error codes that are used here.

• The OpenLCB Memory Access Configuration Protocol Standard, which will be used as an example of an application-level protocol that interacts with the Streaming protocol for data transport.

20 # 4 Definitions (Normative)

Stream is a uni-directional sequence of bytes transmitted from one OpenLCB node to another OpenLCB node. There is no out-of-band information that can be sent along with the payload using the Streaming protocol. There is no restriction on the total number of bytes sent, or the time that the stream can be open for sending data.

25 **Source node** is the OpenLCB node which is sending the data (originating the stream).

Destination node is the OpenLCB node which is receiving the data.

Source Stream ID (SID) is a one-byte (8-bit) identifier (range [0..254], reserved [255]), assigned by the source node at its discretion to identify the stream.

30 **Destination Stream ID (DID)** is a one-byte (8-bit) identifier (range [0..254], reserved [255]), assigned by the destination node at its discretion to identify the stream.

Stream Content UID is a non-zero 6-byte globally unique identifier assigned to specify the type (format and purpose) of the data being transmitted. Stream Content UIDs are assigned using Unique Identifiers in the same fashion as Node IDs.

(Max) Buffer size is an unsigned 2-byte value (range [1..65535]) specifying the maximum number of bytes in-flight; that is, the maximum number of bytes the source node may send without receiving a reception acknowledgement from the destination node.

5 Message Formats

35 In the following, the "common MTI" column specifies 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.

5.1 Stream Initiate Request

40 Stream Initiate Request is an addressed message sent by the Source node to the destination node to request creating the stream. This is the only method to create a stream. The Destination Stream ID is only to be suggested if previously negotiated with the destination node using some other prerequisite protocol.

Data content:

Name	Dest ID	Event ID	Common MTI	Data Content	
Stream Initiate Request	Y	N	0x0CC8	5, 6 or 12 bytes	
				Byte 0-1	Max Buffer Size (proposed by source).
				Byte 2-3	Flags: LSB (bit 0): Deprecated, send as 0, ignore on receipt. bit 1-7: Reserved, send as 0, ignore on receipt.
				Byte 4	Source Stream ID, value 0xFF is reserved.
				Byte 5	Optional: Proposed Destination Stream ID, value 0xFF is reserved.
				Bytes 6-11	Optional: Stream Content UID.

45

5.2 Stream Initiate Reply

Name	Dest ID	Event ID	Common MTI	Data Content	
Stream Initiate Reply	Y	N	0x0868	6 bytes + optional error string	
				Bytes 0-1	Max Buffer Size (final value), or zero if the request is rejected.
				Bytes 2-3	Flags and Error Codes: 0x8000 Accept, no errors 0x0000 Accept, no errors 0x1000 Permanent error, specifically: • 0x1010 streams not supported • 0x1020 source not permitted • 0x1040 unimplemented • others reserved, do not send 0x2000 temporary error, specifically: • 0x2020 buffer unavailable • 0x2040 unexpected, out of order, or inconsistency in the message or frame sequence received. • others reserved, do not send
				Byte 4	Source Stream ID, value 0xFF is reserved.
				Byte 5	Destination Stream ID, value 0xFF is reserved.
				Bytes 6...	Optional zero-terminated human readable error string.

5.3 Stream Data Send

Name	Dest ID	Event ID	Common MTI	Data Content	
Stream Data Send	Y	N	0x1F88	1 – <Max Buffer Size + 1> Bytes	
				Byte 0	Destination Stream ID, value 0xFF is reserved.
				Byte 1 ...	Stream payload bytes.

50 ## 5.4 Stream Data Proceed

Name	Dest ID	Event ID	Common MTI	Data Content	
Stream Data Proceed	Y	N	0x0888	2 or 4 Bytes	
				Byte 0	Destination Stream ID, value 0xFF is reserved.
				Byte 1	Source Stream ID, value 0xFF is reserved.
				Byte 2-3	Optional Flags

5.5 Stream Data Complete

Name	Dest ID	Event ID	Common MTI	Data Content	
Stream Data Control	Y	N	0x08A8	2 or 6 Bytes	
				Byte 0	Destination Stream ID, value 0xFF is reserved.
				Byte 1	Source Stream ID, value 0xFF is reserved.
				Bytes 2-3	Optional Flags
				Bytes 4-9	Number of payload bytes transferred in total (sum of all data segments) % 2^{24}.

6 States (Normative)

55 A stream can be in one of the following states:

Non-existent or **closed**. Between the nodes <Source Node ID> → <Destination Node ID> both the Source Stream ID and the Destination Stream ID have never been used or the Stream according to the last use was completed or terminated.

Announced. The Source node has sent the Source Stream ID to the Destination node via some higher-
60 level protocol, such as in a Memory Configuration Protocol datagram 'Read Stream Reply', or 'Write Stream Command'.

Initiated. The Source node has sent a Stream Initiate Request to the Destination node with the Source Stream ID, and is waiting for a response.

Open. The Destination node has sent a Stream Initiate Reply to the Source node with the Source
65 Stream ID and the Destination Stream ID and the error code set to Accepted.

7 Interactions (Normative)
7.1 Opening a stream with announcement

In this method the streaming protocol works together with a higher-level protocol. It is the higher-level protocol that requests the transfer, and defines the format, contents and possibly the length of the
70 stream payload. As part of handling the higher-level protocol, the Source node has to know the Destination node ID, and allocate a Source Stream ID for that Destination node that is in the Closed / Non-existent state. The Source node transmits the Source Stream ID to the Destination node in an addressed message of the higher-level protocol. The Source node then waits for a reception acknowledgement of the higher-level protocol (such as a Datagram Received OK) from the Destination
75 node. If this acknowledgement is positive, the Stream transitions to the Announced state; if it returns an error, the Stream is considered to be in the Closed state. If the Destination node can not handle the streaming protocol, it should return an error in the higher-level protocol.

After the acknowledgement of the reception of the announcement message, the Source node sends a Stream Initiate Request to the Destination node, with the same Source Stream ID as in the
80 announcement message. The Source node may pass or omit the Stream Content Type in the Stream Initiate Request, as the higher-level protocol specification requires. The Source node proposes a Max Buffer Size for the transmission. Sending this message transitions the stream to the Initiated state.

The Destination node is then required to send a Stream Initiate Reply, or one of the general error messages defined in the Message Network Standard, such as Optional Interaction rejected, or
85 Terminate Due To Error. When possible, a Stream Initiate Reply with an error code should be sent instead of a general error message.

To accept the stream and transition to Open state, the Destination node must allocate a Destination Stream ID that's not currently in use with the same Source node, reply with a Stream Initiate Reply message, with the error code having the Accept bit set, the Source Stream ID set to the stream ID sent

90 by the Source node, the Destination Stream ID filled in, and the Max Buffer Size set to a non-zero value that is at less than or equal to what was proposed in the Stream Initiate Request. Sending this message transitions the stream to Open state, with the Max Buffer Size as entered in the Stream Initiate Reply message.

To reject the stream, the Destination node must send a general error message, or a Stream Initiate
95 Reply message with the appropriate Source Stream ID, zero Max Buffer Size and the error code set to a value with the Accept bit clear and one of the Temporary error or Permanent error bits set. When possible, a Stream Initiate Reply should be sent instead of a general error message. When sending a general error message, the MTI value must be set to the MTI of Stream Initiate Request. A rejected stream is transitioned to the Closed state. If needed, the interaction may be re-started using the higher-
100 level protocol.

7.2 Opening a stream without announcement

This method may only be used if the interaction is started by the Source node.

The interaction begins with the Source node allocating a new Source Stream ID, and sending a Stream Initiate Request to the Destination node. This message must carry a 6-byte Stream Content UID, which
105 specifies for the destination node the content and format of the data carried, and transitions the stream to the Initiated state. From this state onwards the interaction continues as in described in Section 7.1 .

If the Stream Content UID is unknown to the Destination node, it should use the Permanent Error, Unimplemented error code in the rejection message.

7.3 Data transfer

110 Once a stream transitioned to Open state, the Source node may start sending payload bytes using Stream Data Send messages. The source node may send up to Max Buffer Size bytes using one or more Stream Data Send messages, without needing to wait for acknowledgement from the destination node.

When the Destination node is ready to receive another Max Buffer Size bytes, it sends a Stream Data Proceed message to the Source node. This signals to the Source node that it may send another Max
115 Buffer Size bytes. Each Stream Data Proceed message, irrespectively of when it was sent, extends the window of allowed payload bytes to be sent by Max Buffer Size. A Stream Data Proceed may be sent before an entire buffer of data is received, but no more than one Stream Data Proceed may be provided in advance in this way.

If the Source node has no more data to send, but wants to keep the stream open it may, but is not
120 required to, periodically send Stream Data Send messages with zero payload bytes to ensure that the destination node and any gateways in the transfer path are still aware of the stream transmission.

7.4 Closing the stream

The Source node signals completion of the transfer by sending a Stream Data Complete message to the Destination node, providing the Source Stream ID and the Destination Stream ID with it. Optionally
125 the Source node may add a 4-byte value describing the total number of bytes sent in the stream. If the Source node does not want to specify this value, it may omit the bytes from the Stream Data Complete message or may send 0xFFFFFFFF to specify unknown length.

The Stream Data Complete message transitions the stream to Closed state.

130 The Stream is closed if the Destination node sends a Terminate Due To Error or Optional Interaction Rejected message with the MTI value of Stream Data Send; or if the Source node sends a Terminate Due To Error or Optional Interaction Rejected message with the MTI value of Stream Data Proceed. When possible, the specific MTI value should be used.

7.5 Special provisions for Gateways

135 A Gateway needs to be prepared for buffering up to Max Buffer Size bytes for each open stream. It may reduce the Max Buffer Size in the Stream Initiate Request message if the buffer size available is smaller than proposed. A Gateway must not modify the Max Buffer Size value of a Stream Initiate Response message.

140 A Gateway may repackage the stream payload into fewer or more Stream Data Send messages than received, as the underlying network requires. A Gateway must emit a Stream Data Send message with empty payload after its buffer is drained if it has received such a message.

A Gateway may delay forwarding Stream Data Proceed messages until its internal buffer is drained, but it must not drop any of them.

A Gateway must delay forwarding the Stream Data Complete message until its internal buffer is drained.

145 A Gateway must drop all data in its internal buffer and all delayed messages if the Stream is terminated with an error message of Terminate Due To Error or Optional Interaction Rejected.

8 Adaptation to CAN Transport (Normative)

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

8.1 Stream Data Send

150 Stream Data Send messages are sent on a CAN network using the Frame Type of 7 (Stream Data), the Destination node alias and the Source node alias in the CAN header, and the Destination Stream ID in the first data byte. The remaining data bytes can be filled with stream payload (up to 7 bytes per frame):

Name	Dest ID	Event ID	Header Format	Data-part Content
Stream Data Send	Y	N	0x1Fdd,dsss	Byte 0: Destination Stream ID Byte 1...: payload bytes

155 8.2 All Other Messages

All other messages are sent as described in the OpenLCB Message Network Standard. Note that the Stream Initiate Request message may need to be sent in two CAN frames if it contains a Stream Content UID. The format for multiple frame CAN messages is described in the OpenLCB Message Network Standard.

160

Table of Contents