OpenLCB Technical Note

Glossary

Jan 31, 2013

~~Preliminary~~ ~~Updated~~Draft

# 1 Introduction

This glossary provides working definitions and commentary on definitions for OpenLCB. It is not normative in any way.

We provide a separate section of “Key Terms” for initial reading.

# 2 Key Terms

Event

OpenLCB allows nodes to notify each other when specific "events" occur on the layout. These in turn can cause nodes to take particular actions. Events are not necessarily attached to a producer ('Button 2 pressed') or attached to a consumer ('Turn off light 4'), but rather to an overall state change ('Set for nighttime operation'). This is called a "Producer/Consumer model".

Node

A “node” is the basic unit of addressability in OpenLCB. It can be a single board, or a process in a larger computer, or any other independent actor on the OpenLCB network. Every OpenLCB interaction originates in a node. Every board that connects to OpenLCB is at least one node. For example, a simple turnout controller board is one node, while a PC with multiple programs running may contain several nodes.

Message

The basic unit of OpenLCB communication. OpenLCB nodes exchange messages to control communications and to move information. It may take several messages to move a single datagram, or to start a stream, for example.

Protocol

An OpenLCB protocol defines how messages are exchanged to do some particular thing. There are low level protocols for things like sending datagrams, exchanging events, and doing link start-up. Higher-level protocols then define how configuration information is exchanged, or a nodes memory can be read, or a message can be put on a display.

# 3 General Glossary

Addressed

An “addressed” message is meant for only a single node. Contrast “global”.

Alias

Short form of a Node ID (NID) number which can be mapped back and forth to the full number. Often used with a modifier, as "NID Alias", "DID Alias" or "SID Alias"", where

DID and SID are specific types of Node ID (see below) and which are then written NIDa, DIDa and SIDa. Only used in certain forms of the OpenLCB protocols, e.g. on CAN links.

Automatically Routed Event

The Event transfer protocol defines a series of messages that determine where Event messages must be routed. Automatically Routed Events are events within a particular part of the EventID space that are routed to all nodes, without using those messages. See the Event transfer protocol documentation for more information.

Board

“Board” has no precise meaning in OpenLCB. It generally refers to a single piece of electronics. Sometimes used to refer to “the piece of electronics that implements a Node”. ~~Not really something that occurs in OpenLCB itself, we need to talk about how the common term "board" maps onto OpenLCB. E.g. A node (board) may connect to several things (devices) on the layout.~~

Bridge

~~Connects two OpenLCB segments with minimal changes to the content of the messages. For example, a bridge between two CAN segments would allow more nodes to be attach to the combined segments as if they were one; a bridge between a CAN segment and Ethernet segment would transform message format, but transfer every message.~~ Usually a bridge works on the data link layer of the OSI model, and therefore has knowledge only about those CAN frames transferred, but not about the meaning of the messages as they are defined in a higher layer of the OSI model, such as the transport layer. Similarly to a repeater, a bridge may be used to allow the connection of more nodes than is possible on one CAN segment, but this is not its main purpose. Usually a bridge does not translate between different wire protocols, and for that a gateway is used. A bridge differs from a repeater in that a bridge can forward CAN frames to different segments based on Node ID. To do this, the bridge listens to, and interprets, the frames transferred in order to learn which nodes are connected to which segment. For OpenLCB the bridge may also support different node aliases on the different segments to which it is connected. It this case, it would translate the node alias before forwarding that frame to another segment.

CDI

“Configuration Definition Information”, the information provided by a node so that other nodes can understand what the configuration information in the first node includes, and how it's organized.

Datagram

The actual ~~message~~data content transported by the Datagram Protocol. This can be up to 72~~70~~ bytes.

Datagram Protocol

The datagram protocol is a way for a OpenLCB node to efficiently send a short, definite-length message to another specific node. In that, it lies in between the event exchange protocol, which efficiently sends very short event IDs to all interested nodes, and the streaming protocol, which sends long messages between specific nodes, at some cost in efficiency. Sending a datagram takes multiple messages.

Datagram Protocol ID (DPID or DPI)

The first byte of data transferred by a datagram is used to indicate the protocol being transported.

Device

"Device" has no precise meaning in OpenLCB. It generally refers to a single piece of equipment on a layout connected to a Node.

~~Not really something that occurs in OpenLCB itself, we need to talk about how the common term~~

~~"device" maps onto OpenLCB. E.g. A node (board) may connect to several things (devices) on the layout..~~

Destination Node ID (DID)

Node ID of the node to which a specific message is addressed.

Event Exchange Protocol

Using PCER messages to propagate Event IDs from multiple source nodes to multiple destination nodes. These are typically used to announce that something of interest has happened on the layouts so that controlled devices can respond. Also includes inquiry and response messages for examining configuration and status.

Event ID (EID)

The number that identifies a specific event. OpenLCB event IDs must be globally unique.

Events are not associated with any particular node. It may be convenient to use e.g. Node ID as a way of numbering them uniquely, but Node IDs and Event IDs are not related.

Gateway

~~Connects two segments of the OpenLCB while translating to a different wire protocol. For example, a gateway can change NodeIDs on one side of the Router into aliases on the other side. The use of unique identifiers in OpenLCB is meant to reduce the need for gateways. They generally only appear to connect the wider network to a CAN link.~~ A Gateway translates between different wire protocols (e.g. CAN and ethernet), or even different network architectures (e.g. OpenLCB and Loconet). They repackage and convert data going from one environment to another so that each environment can understand the other's data. Since OpenLCB uses unique identifiers, and these are meant to be used unchanged on most wire protocols, this should reduce the necessity of Gateways.

Global

A “global” message is meant for all OpenLCB nodes that are interested in it. Contrast “Addressed”.

Installation

An OpenLCB installation is the complete set of OpenLCB hardware, nodes, etc, that can be reached from any one of them.

Message Type Indicator (MTI)

Every OpenLCB message contains a Message Type Indicator field that identifies the type of that particular message. Also used to refer to a particular value for that field.

Node ID (NID)

Number identifying a specific node. OpenLCB node IDs must be globally unique, so they form a one-to-one mapping to the nodes themselves.

Producer/Consumer Model

A way of thinking about communications on a layout network. Events (unique messages on the layout network) are “produced” by some node(s), and “consumed” by other node(s). An event indicates something happened, but the corresponding message isn't necessarily a status report or a command for something to change; it's just an announcement of the event. That event can be produced by many different sources (e.g. the “use passing siding” event could be produced by a push button at either end of the siding, or on a dispatcher panel), and can be consumed by several sources (it can turn on lamps on a panel, set turnouts in multiple places, etc).

Producer-Consumer Event Report (PCER)

OpenLCB message type sent globally by producers of a particular Event to indicate that it has occurred, so that the corresponding consumers of that Event are notified.

Repeater

~~Connects two segments of the same type at the physical level, without transformation of message format.~~ Connects two segments of the same type at the pysical layer of the OSI model. It regenerates the received signals and then retransmits the regenerated (or conditioned) signals on the other segment. It has no knowledge of the information transfered. It is usually used to connect two CAN segments to allow more nodes to be attach to the combined segments as if they were one.

Router

~~OpenLCB routers forward traffic from one link to another without reformatting. This can be a simple as receiving on one link and forwarding on N, or it could involve tracking e.g. event usage and only forwarding needed messages.~~ Routers work at the Network layer of the OSI model, meaning that they can route frames across multiple segments based only on information contained in messages. They do not have knowledge about protocols of (higher) transport layer. Routers are usually used in complex network situations because they provide better traffic management than bridges, and do not pass broadcast traffic. Routers can share status and routing information with one another, and can use this information to bypass slow or malfunctioning segments.

Source Node ID (SID)

The Node ID of the node which originated a specific message.

Segment

~~Subset of an overall OpenLCB installation which is reached via path or gateway. A segment typically uses a specific wire protocol.~~ Subset of an overall OpenLCB installation which uses a specific wire protocol.  A segment never spans over Bridges, Gateways, Repeaters or Routers.

Simple Node

A “simple node” is typically an individual board with a small processor on it which is acting a as leaf node in an OpenLCB network. A simple node doesn't initiate complex communications with other nodes, but may be the target of e.g. configuration operations.

Stream

Streams are a method of moving a large number of bytes between two nodes. The streaming protocol defines a way to do this in OpenLCB.

Streaming Protocol

Protocol that defines the messages and interactions to setup a stream transfer, move the data along with the necessary buffer management, and then take down the transfer at the end.  Streams are efficient ways to move large amounts of data when the setup and take-down process can be considered a negligible part of the total transfer.

Virtual Node

One of one or more nodes co-resident within a single board or computer. Often a board is a single node, but it's possible for a board or computer to behave on the OpenLCB as if it contained multiple nodes. Those are then referred to as virtual nodes.

Well Known Events

Certain Events are globally defined to have specific meanings. Their Event IDs are documented as part of the event protocol. These are called “well known events”.

Wire protocol

Version of the OpenLCB common messages, interactions, etc adapted to a particular transport mechanism. Examples are the wire protocols for CAN bus segments and TCP/IP links.

## 4 CAN Glossary

CAN

Controller Area Network, an ISO-standardized communication network type. The CAN specification includes aspects of the physical- and link-layer definitions for OpenLCB.

Frame

A Frame is packet as it is defined on the CAN bus. It consists of a 11 or 29 bit CAN header and zero through 8 bytes of data. An OpenLCB message may become one or more frames when transmitted over CAN. Some frames, called Control Frames, are used for link-layer control and do not correspond to any particular OpenLCB message.

OpenLCB-CAN

Short form for “OpenLCB as implemented on a CAN link”. For example, “OpenLCB-CAN message” refers to an OpenLCB message formatted for transmission over a CAN link.

Segment

See general definition.  Note that a nodes on a single CAN segment share CAN arbitration and a set of aliases.   ~~A single CAN segment is the set of CAN nodes that share CAN arbitration.~~

## Table of Contents