OpenLCB Working Note	
Location Service	
Sep 11, 2023	Preliminary

1 Introduction

A Working Note is an intermediate step in the documentation process. It gathers together the content from various informal development documents, discussions, etc into a single place. One or more Working Notes form the basic for the next step, which is one or more Standard/TechNote pairs.

This working note is the start of effort toward a general location service protocol which can be used by e.g. RFID and QR scanners and other detectors to publish information about the detected location of rolling stock.

1.1 Served Use Cases

- A signal needs to determine occupancy in the block before it. It is not concerned with identity, but it does need to know location and entry/exit.
- A train enters a current-detected block. A detector recognizes this, successfully queries the decoder for its address, and reports that information on the OpenLCB bus. That information results in a signal sending a speed command such as Slow or Stop to that particular detector.
- A train crosses an RFID reader which determines the train's DCC address and reports that address on the OpenLCB bus.
- A train exits a current-detected block. A detector recognizes this, and reports that interaction on the OpenLCB bus.
- A train enters a current detected block. A detector recognizes this, and attempts unsuccessfully to query the decoder for its address. The detector emits a report without decoder address information.
- A train-announcement unit needs to determine the identity and location of approaching trains so it can announce their arrival.
- A train display needs to gather all train locations for display to the users.
- A train-display needs to gather the location and identity of all trains within a specified area.
- An application needs to gather the location and identity of all trains.
- A node powers up and emits a query which generates messages carrying all available location information.

- A detector observes that a piece of rolling stock has been added to the layout. It reports this. A throttle or command station provides that rolling stock in its roster display.
- A detector observes that a piece of rolling stock has been removed from the layout. A throttle or command station removes that rolling stock from its roster display.

1.2 Unserved Use Cases

(To be determined as the proposal is developed)

1.3 Requirements

1) The protocol should cover multiple hardware types, including RFID and QR scanners, RailCom and other DCC detectors, and follow-on hardware that can detect and identify rolling stock at a point, on a track section, or in an area.

2) To the extent possible, the protocol should identify standard forms for common information. For example, the Unique ID of the identified rolling stock should be in a common location in the protocol's message(s) so that it can be extracted regardless of the form of the sensor that generated it. For example, a RFID detector and a QR detector on the same layout should both provide the same information on the identity of the detected rolling stock.

3) To the extent possible, the protocol should identify methods for its message(s) to carry detector-specific information. For example, a RFID detector and a QR detector on the same layout should both be able to publish their tag-specific information.

4) The identification of the scanning hardware should be guaranteed unique. This lets hardware be added to a layout without having to resolve conflicts.

5) The messages should not rely on the transmitting-node-address field of the message for decoding. I.e. the protocol doesn't say "You identify who made the scan by looking at who sent the message". This allows messages to be created by e.g. a debugging node during development.

6) It should be possible to look at the message(s) carrying a specific scan and know that they contain a scan without additional information. I.e. you don't have to configure a layout monitor with all possible scanner IDs before it can start processing or displaying scanning results.

7) The identification of which scanner is reporting should be early in the message(s). This allows nodes to drop the receipt of the message(s) early, without having to buffer them completely. (Note that the relative priority of this requirement and the one below has not yet been determined)

8) The identification of the detected rolling stock should be early in the message(s). This allows nodes to drop the receipt of the message(s) early, without having to buffer them completely. (Note that the relative priority of this requirement and the one above has not yet been determined)

9) There should be a standard way to find the UniqueID of the scanned piece of rolling stock, when the scanner can provide one. For some scanners, e.g some types of RFID scanner, the

scanner might not be able to map the scanned value to a specific piece of rolling stock, in which case this information field would be blank.

10) Note that the item directly above is referring to a Unique ID (node ID), not a DCC address. Someday, there will be LCC-native trains and this protocol should be ready for them. The alternative is inevitably a messy migration at some point in the future.

11) Ideally, the messages for this protocol could be a super-set of those from the DCC Detection Protocol.

12) Ideally, information about short circuit and current overload status could be conveyed by this protocol. These are often generated by the same device as the transponder messages e.g. LocoNet devices. These are providing location-specific information. It would be good to have some commonality between all location based messages. Another example would be current or voltage measurements at a point.

13) An RFID reader reads a tag and uses this protocol to distribute the raw tag contents. A separate database node receives this content, does an internal lookup to locate the rolling stocks Unique ID, and then uses this protocol to distribute a version of the original distribution that also carries the Unique ID.

14) The user has a throttle, which has a train on it. There is a location report coming in for that train. The user would like to display on the throttle the name of the location where that train was seen. (This has to be a user-visible name, so coming out of some configuration.) This raises two indirection issues that need to be solvable:

    A) Indirection of the OpenLCB train and the detector address. Say this is a DCC train from a command station, then there is a known DCC address, which is recognized by RailCom detectors, and reported. The harder case of OpenLCB trains needs to be handled too.

    B) Indirection from the detector address to a user-visible (configured) block name.

15) Cab signaling: display on the throttle what the next signal aspect is for the selected train. This raises indirection issues:

    A) Going from train to block as above

    B) Going from physical orientation plus reverser setting on the logical locomotive to an absolute direction inside the block

    C) Going from block + absolute direction to a next signal (which is a multi-state state machine)

    D) Aspect interpretation (state to human readable text)

2 Specified Sections

This is the usual section organization for a Technical Note, to accumulate the Standard and Technical Note content in its eventual order.

2.1 Introduction

Note that this section of the Standard is informative, not normative.

2.2 Intended Use

Note that this section of the Standard is informative, not normative.

2.3 Reference and Context

See also the DCC Detection working note for a simpler form specific to DCC.

This working note assumes adoption of the Event With Payload working note as a Standard.

2.4 Message Formats

2.5 States

2.6 Interactions

3 Background Information

RFID readers and tags come in multiple forms. With 13.56MHz RFID, be it ISO14443 (as used in the RC522 readers), or ISO15693 as used in StaRFIshrail, the following points apply:

- Up to 8 bytes can be read "on the fly", i.e. as the tag on a train passes over a reader at speed. Depending on how the reader is set up, these 8 bytes can either be the UID of the tag, programmed in at time of manufacture, or 8 bytes of EPROM, which can be programmed into the tag by the user. The usage of these 8 bytes would be determined by the user, but could, say, be 2 bytes for Stock ID, 2 bytes for DCC address, and 4 bytes for other purposes.
- In addition, there at least another 96 bytes of user programmable EPROM, but these need to be read when the tag is essentially static over a reader, due to the time taken to read all the bytes.

The tags used by the MERG CAN RFID readers provide five bytes of programmable payload. Multiple conventions have been proposed for the contents of those bytes.

QR codes come in multiple forms. Depending on version, size and error correction level, they can contain thousands of characters.

Some detectors can report the direction of the detected train. For some, this is whether the train is moving in forward or reverse. For others, it's whether the train is moving in one direction on the track vs the other (North vs South, East vs West). ProTrak Grapevine systems can report moving North/East, not moving, and moving South/West.

RPS detectors report absolute X, Y and Z coordinates of the detected rolling stock. The report is emitted at specified intervals. Although these are no longer being produced, something similar may emerge in the future with developments in e.g. video monitoring of a layout.

Traditional block detectors and point detectors are an interesting special case. They're simple, low-cost and common on layouts. They can't sense any address information from the detected rolling stock. The protocol should be able to convey their information too.

Table of Contents

1 Introduction....1
1.1 Served Use Cases....1
1.2 Unserved Use Cases....2
1.3 Requirements....2
2 Specified Sections....3
2.1 Introduction....3
2.2 Intended Use....3
2.3 Reference and Context....3
2.4 Message Formats....3
2.5 States....3
2.6 Interactions....3
3 Background Information....3

140