OpenLCB Technical Note	
Simple Train Information Protocol	
Mar 9, 2014	Preliminary

1 Introduction

There is a need for something for train nodes, that uses a single message instead of datagrams and configuration. This allows getting getting the information with a bare minimum of support at the requested node end, so long as the requesting node can be sure to handle it.

2 Annotations to the Standard

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

Cross-link to ACDI is important.

2.4 Messages

Name	Dest ID		Common MTI	CAN format	Data Content
Simple Train Information Request	Y		0x0DA8	0x19DA,8sss	fddd

Name	Dest ID		Common MTI	CAN format	Data Content
Simple Train Information Reply	Y		0x09C8	0x199C,8sss	fddd, Content, see below

The Simple Train Information Reply message(s) carry a stream of bytes:

- The byte 1 to indicate the version of the following content
- A null-terminated string for the road name (i.e. “Rio Grande Southern”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.
- A null-terminated string for the train class (i.e. “K-27”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.
- A null-terminated string for the road number (i.e. “455”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.
- A null-terminated string for the train name (i.e “San Juan Express”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.
- A null-terminated string for the manufactures name (i.e “Blackstone”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.
- A null-terminated string for the owners name (i.e “John Smith”) is recommended, but not required that this string be 32 characters or less based on a single line display of a modern smart phone.

Note that the total length, including version numbers and nulls, must be less than [253] bytes. This is based on the length of SNIP. There is discussion as to if this limit is needed. The argument is due to buffer management in a bridge between a fast and slow physical layer (Ethernet to CAN for instance). The bridge must buffer the entire Ethernet message then split it into CAN 8 byte messages. Having a fixed max length allows fixed buffers in the software/firmware development.

The data returned by STIP must equivalent that is returned in the corresponding elements in the TDI Protocol.

2.5 Interactions

When a node gets a Simple Train Information Request, if possible it shall reply with one or more Simple Train Information Reply messages containing the train information. If it's not able to process the information, it shall send an Optional Interaction Rejected with an appropriate error code.

It's recommended that the rejection message have the temporary-error bit set, so that the node sending the original request will retry it.

3 Draft notes

Handling a second request while sending the first: If they can't be queued for later (immediate) processing, the node can reply with OptionalInteractionRejected message that indicates the issue is temporary, and that the request should be retried if the information is still desired.

(Setting of the first/last bits to make it a message; “because this version of the specification predates the use of first/last bits”, nodes initiating a transfer must be able to handle a series of reply messages)

Do nodes sending the response need to be able to resend it in response to a TerminateDueToError? This might be needed is e.g. a gateway gets congested while merging frames.

The information returned is intended to be considered static: A node may request it and never have to request it again, because it won't change. But the user might change the user-provided node name. How will other nodes know to do a STIP request? They don't. Configuration tools could force themselves to reacquire the data, but there's no global notification of the change. (Allowing the node to only change on reset would fix that, because we could have SPIP users re-request the info when they see the reset.)

Table of Contents

1 Introduction....1
2 Annotations to the Standard....1
2.1 Introduction....1
2.2 Intended Use....1
2.3 Reference and Context....1
2.4 Messages....1
2.5 Interactions....2
3 Draft notes....2

