OpenLCB Technical Note	
Simple Node Information Protocol	
Jun 15, 2012	Preliminary

1 Introduction

There is a need for something for tiny 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.

See the larger description in the ACDI TN, which should be copied here too. But note that SNII they can exist independently, which needs to be handled too.

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 Node Information Request	Y		0x3520	0x1Edd,dsss 52	

Name	Dest ID		Common MTI	CAN format	Data Content
Simple Node Information Reply	Y		0x3530	0x1Edd,dsss 53	Content, see below

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

- The byte 1 to indicate a version
- A null-terminated string for manufacturer name
- A null-terminated string for node model name
- A null-terminated string for node hardware version
- A null-terminated string for node software version
- The byte 1 to indicate a version
- A null-terminated string for user-provided node name
- A null-terminated string for user-provided node description

2.5 Interactions

When a node gets a Simple Node Information Request, if possible it shall reply with one or more Simple Node Information Reply messages containing the node's 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. Directed, so CAN priority doesn't really help. Just reject (which kind) and have upstream resend the request, if you can't hold onto the request until it happens again.

Table of Contents

