OpenLCB Technical Note	
Simple Node Information Protocol	
Aug 19, 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 of whether ACDI is present in the node or not.

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 the version of the following content
- A null-terminated string for manufacturer name of no more than 41 bytes including terminating null.
- A null-terminated string for node model name of no more than 41 bytes including terminating null.
- A null-terminated string for node hardware version of no more than 21 bytes including terminating null.
- A null-terminated string for node software version of no more than 21 bytes including terminating null.
- The byte 1 to indicate the version of the following content
- A null-terminated string for user-provided node name of no more than 63 bytes including terminating null.
- A null-terminated string for user-provided node description of no more than 64 bytes including terminating null.

Note that the total length, including version numbers and nulls, must be less than 253 bytes.

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: 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 SNII 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 SNII users re-request the info when they see the reset.)

Table of Contents

