

Re: Suggestions for custom application-layer protocol?

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.embedded/2005-05/0112.html>

From: Gordon Burditt (gordonb.ir1o9_at_burditt.org)

Date: 05/25/05

Date: Wed, 25 May 2005 18:03:17 -0000

*>I want the protocol to be text based rather than binary since the data
>throughput is low. I also want the protocol to be based on TCP/IP. Does
>anyone here have any suggestions on the design of a simple protocol? Are
>there simple, standard ways of formatting text messages to be sent over a
>TCP socket (such as comma-separated)? I would prefer to avoid the
>complexities of XML. Since either the embedded device or the monitoring
>terminal can initiate a message, is it preferable to have TCP servers
>running on both sides? Or is it better to simply leave a TCP connection
>open between client and server? Any suggestions on ultra-simple standard
>protocols that do something similar to this?*

Take a look at protocols like SMTP, NNTP, POP3, IMAP, etc. Another example to look at is the "AT" command set of modems (although this doesn't use TCP). A command is one line of text. It starts with some kind of command verb that identifies what follows. The response is usually one line of text, and something at the beginning of the line identifies what kind of response it is (status code) and may identify whether it's the last line or not. The status codes are in groups so a client doesn't have to be aware of all the status codes (e.g. 4NN is a temporary failure of some kind, 5NN is a permanent failure).

Spontaneous responses make the protocol a bit more complicated. You need to be able to identify a spontaneous response (status code) and perhaps the spontaneous response only says there IS data, and the client needs to ask for it to actually get it sent. IMAP uses tags on requests and replies so you can match them up, and deals with spontaneous responses.

Use any kind of field separator that's appropriate for the data you are sending. POP3 and IMAP use spaces. Commas might be more appropriate for certain kinds of data.

Testing is generally easy: telnet to the appropriate port and manually type commands. Read the responses.

comp.os.linux.embedded: Re: Suggestions for custom application-layer protocol?

Gordon L. Burditt