

Re: How to make ftp server less verbose

Source:

<http://www.tech-archive.net/Archive/Internet-Server/microsoft.public.inetserver.iis.ftp/2004-06/0233.html>

From: Alun Jones [MSFT] (alunj_at_online.microsoft.com)

Date: 06/25/04

Date: Fri, 25 Jun 2004 10:40:05 -0700

"io" <iolshansky@agrilink-int.com> wrote in message
news:uO6moBmWEHA.2816@TK2MSFTNGP11.phx.gbl...
> *Thanks for info Alun, the reason to ask in the first place is CDMA 1x
price
> plans in Australia where they charge on transmitted data volume basis. It
> happened that we need to transmit small files (~0.5K) every hour from CDMA
> modem-equipped devices to our FTP server and we found that FTP protocol
> overhead was forbiddingly high being ~1400 bytes in average for each
> transfer. I think we need to look for some other solution, prob. come up
> with proprietary protocol.*

I'd be interested to see your analysis of the extra data that leads you to
suspect the response messages as being a significant portion of that.
Here's an example of the traffic required for an upload:

```
{Connection}
220 Microsoft FTP Service [27 bytes - 21 bytes text]
USER eric [11 bytes]
331 Password required for eric. [33 bytes - 27 bytes text]
PASS flimflam [15 bytes]
230 User eric logged in. [26 bytes - 20 bytes text]
TYPE I [8 bytes]
200 Type set to I. [20 bytes - 14 bytes text]
PORT 127,0,0,1,9,79 [21 bytes]
200 PORT command successful. [30 bytes - 24 bytes text]
{Server connects to 127.0.0.1:9*256+79}
STOR foo.txt [12 bytes]
150 Opening BINARY mode data connection for foo.txt. [54 bytes - 48 bytes
text]
{Client sends data to server}
{Client closes data connection}
226 Transfer complete. [24 bytes - 18 bytes text]
QUIT [6 bytes]
221 [6 bytes - 0 bytes text]
{Closure}
```

So, ignoring TCP framing and negotiation, this shows a simple transfer as taking up 293 bytes of commands and text. If we strip all the text out of every response there, and reduce it to the bare minimum of numbers and spaces (assuming that the FTP client will accept that), we see that there's 172 bytes of text, and 121 bytes of absolutely essential material.

Where is the rest of your ~1400 bytes coming from? If it's from a minimum unit size on each packet, ripping the text out will have no effect. Note that our longest message was 54 bytes – if your minimum unit size is above that (plus headers and other framing), there's no savings possible.

But if you think that removing the text will achieve this, why not try writing a relatively simple proxy in