



FTP: File Transfer Protocol

EE 122: Intro to Communication Networks

Fall 2007 (WF 4-5:30 in Cory 277)

Vern Paxson

TAs: Lisa Fowler, Daniel Killebrew & Jorge Ortiz

<http://inst.eecs.berkeley.edu/~ee122/>

Materials with thanks to Jennifer Rexford, Ion Stoica,
and colleagues at Princeton and UC Berkeley

1

File Transfer Protocol (FTP)

- Allows a user to copy files to/from remote hosts
- Usage:
 - Client connects to the FTP server
 - User provides a login id and password to become authenticated
 - User can explore the directories
 - User can download files from and upload files to the server
- A predecessor of the Web (RFC 959 in 1985)
- No URL, hypertext, and helper applications to assist user
 - Requirements for the user:
 - o Know the name of the server machine
 - o Have an account on the machine
 - o Can find the directory where the files are stored
 - o Know whether the file is text or binary
 - o Know what tool to run to render and edit the file

2

How Do You Transfer Files Today?

- HTTP - the usual Web transfer mechanism (`http://`)
- FTP
 - You may not have realized that you use it
 - `ftp://` links in web pages (e.g.: `www.kernel.org`)
- sftp
 - E.g.: to upload your project files to EECS inst. machines
- BitTorrent and other file-sharing software
- scp
- Any others?

3

Why Study FTP?

- Helps cement familiarity with text/status-code based protocols (similar to SMTP)
- Illustrates use of **multiple concurrent connections**
 - One for control (commands & replies)
 - Another connection for data (depending on the command)
- Illustrates **reversal of roles**
 - For data connection, FTP user's process plays the server role, FTP server plays the client role
- We'll later use FTP as an example when looking at issues with Network Address Translation (**NAT**)

4

Example Commands

- Authentication
 - USER: specify the user name for login
 - PASS: specify the user's password
- Exploring the files
 - LIST: list the files for the given file specification
 - CWD: change to the given directory
- Downloading and uploading files
 - TYPE: set type to ASCII (A) or binary image (I)
 - RETR: retrieve the given file
 - STOR: upload the given file
- Closing the connection
 - QUIT: close the FTP connection

5

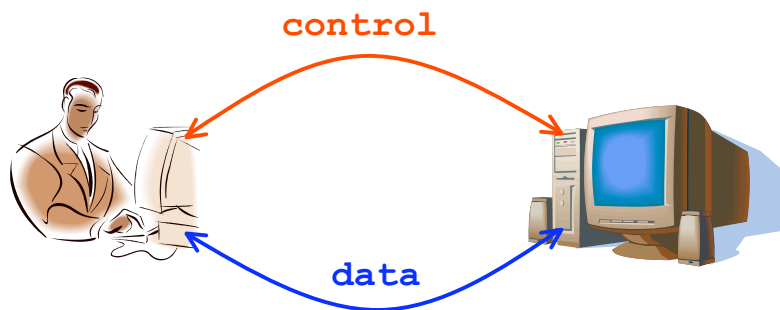
Server Response Codes

- 1xx: positive preliminary reply
 - The action is being started, but expect another reply before sending the next command.
- 2xx: positive completion reply
 - The action succeeded and a new command can be sent.
- 3xx: positive intermediate reply
 - The command was accepted but another command is now required.
- 4xx: transient negative completion reply
 - The command failed and should be retried later.
- 5xx: permanent negative completion reply
 - The command failed and should not be retried.

6

FTP Data Transfer

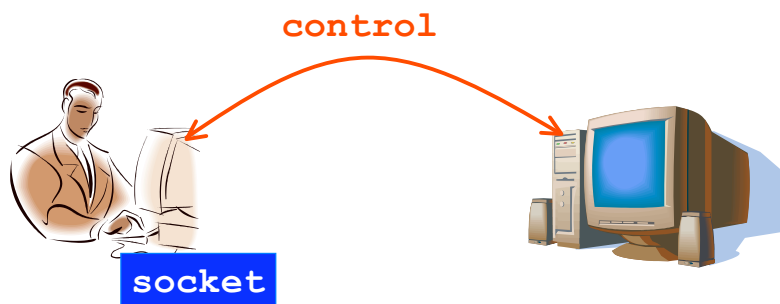
- Use separate data connection to:
 - Send lists of files (LIST)
 - Retrieve a file (RETR)
 - Upload a file (STOR)



7

Creating the Data Connection

- Client acts like a server
 - Creates a socket
 - o Assigned an ephemeral port number by the kernel
 - Listens on socket
 - Waits to hear from FTP server



8

Creating Data Connection (cont.)

- But, the server doesn't know the port number
- After listening, client tells it to the server
 - Via the PORT command on the control connection

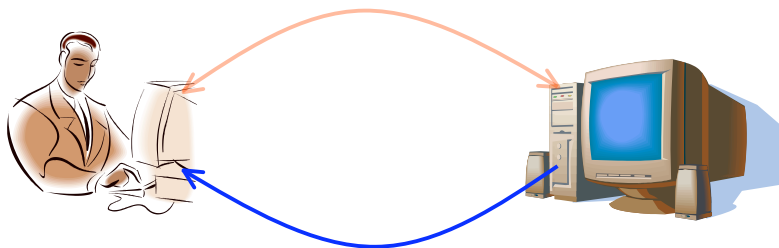
PORT <IP address, port #>



9

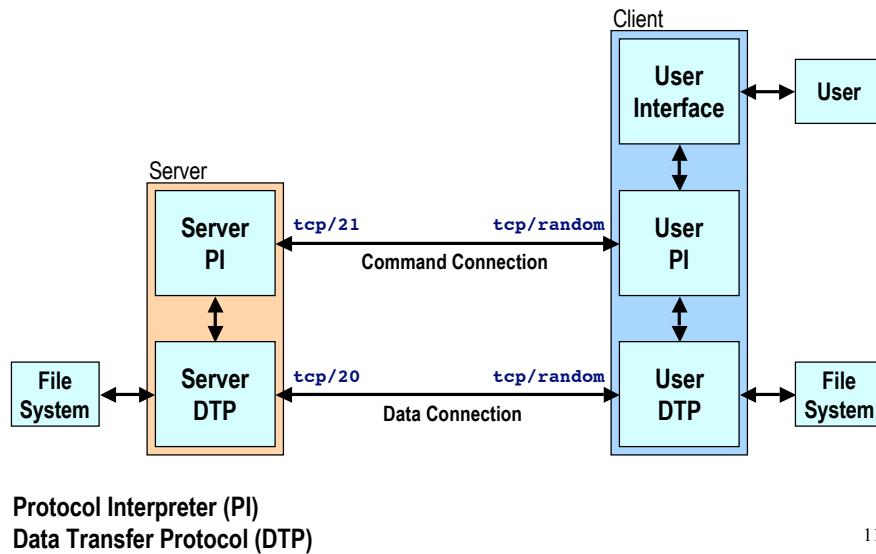
Creating Data Connection (cont)

- Then, the **server** initiates the data connection
 - Connects to the socket on the client machine
 - Client accepts to complete the connection
- Data now flows along **second** connection; first connection remains open for more commands/replies



10

FTP Communication Overview



Why Out-of-Band Control?

- Avoids need to mark end of data transfer
 - Data transfer ends by closing data connection
 - Control connection stays up
- Can abort a transfer without killing control connection
 - Avoids requiring the user to log in again
 - Done with an ABOR on the control connection
- Third-party file transfer between two hosts
 - Data connection could go to a different host
 - o Send a different client IP address to the server
 - o e.g., user coordinates transfer between two servers
 - However: Rarely needed and presents security issues

12

Example

- Traffic captured using
`tcpdump -s 0 -w ftp.trace host ftp.ee.lbl.gov`
- Issue command
`ftp ftp.ee.lbl.gov`

13

What It Looks Like to the User

```
% ftp ftp.ee.lbl.gov
Connected to ee.lbl.gov.
220-
                NOTICE TO USERS

                This computer is a Federal computer system and is the property
                of the United States Government. It is for authorized use only.
                Users (authorized or unauthorized) have no explicit or implicit
                expectation of privacy.

[...]
                LOG OFF IMMEDIATELY if you do not agree to the conditions stated
                in this warning.

                Your ip address is 128.32.48.169
                The local time is Wed Sep 27 15:04:44 2006

220 131.243.1.10 FTP server ready
Name (ftp.ee.lbl.gov:ee122): anonymous
331 Anonymous login ok, send your complete email address as your passw
```

```
Password:  
230 Anonymous access granted, restrictions apply.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> pwd  
257 "/" is current directory.  
ftp>
```

```
ftp> ls bro*  
200 PORT command successful  
150 Opening ASCII mode data connection for file list  
bro-0.9-current.tar.gz.OLD  
bro-0.9a7.tar.gz  
bro-0.9a8.tar.gz  
bro-0.9a9.tar.gz  
bro-change-log.txt  
bro-libidmef-0.7.2-modified.tar.gz  
bro-pub-0.7-stable.tar.gz  
bro-pub-0.8-stable.tar.gz  
bro-pub-0.8a87.tar.gz  
bro-pub-0.8a88.tar.gz  
bro-pub-0.9a4.tar.gz  
bro-pub-0.9a4a.tar.gz  
226 Transfer complete.  
remote: bro*  
283 bytes received in 0.0013 seconds (218.37 Kbytes/s)  
ftp>
```

```

ftp> get bro-0.9a9.tar.gz.TYPO
200 PORT command successful
550 bro-0.9a9.tar.gz.TYPO: No such file or directory
ftp> get bro-0.9a9.tar.gz
200 PORT command successful
150 Opening BINARY mode data connection for bro-0.9a9.tar.gz (3440652
226 Transfer complete.
local: bro-0.9a9.tar.gz remote: bro-0.9a9.tar.gz
3440652 bytes received in 0.81 seconds (4161.86 Kbytes/s)
ftp> cd ..
250 CWD command successful
ftp> pwd
257 "/" is current directory.
ftp> cd secret-files
550 secret-files: No such file or directory
ftp> quit
221 Goodbye.

```

What It Looks Like “On The Wire”

```
% ftp ftp.ee.lbl.gov
```

Server sends exactly this text:

```

< 220-
<
<          NOTICE TO USERS
<
<   This computer is a Federal computer system and is the property
[...]
```

```

< LOG OFF IMMEDIATELY if you do not agree to the conditions stated
< in this warning.
<
<   Your ip address is 128.32.48.169
<   The local time is Wed Sep 27 15:04:44 2006
<
< 220 131.243.1.10 FTP server ready

```

```
Name (ftp.ee.lbl.gov:ee122): anonymous
```

Client sends:

```
> USER anonymous
```

Server replies with exactly this text:

```
< 331 Anonymous login ok, send your complete email address as your pas
```

```
Password:
> PASS ee122@c199.eecs.berkeley.edu
< 230 Anonymous access granted, restrictions apply.
> SYST
< 215 UNIX Type: L8
Remote system type is UNIX.
> TYPE I
< 200 Type set to I
Using binary mode to transfer files.
ftp> pwd
> PWD
< 257 "/" is current directory.
ftp>
```

```
ftp> ls bro*
> PORT 128,32,48,169,189,39
< 200 PORT command successful
> TYPE A
< 200 TYPE set to A
> NLST bro*
< 150 Opening ASCII mode data connection for file list
```

The server sends the following on a separate connection
to 128.32.48.169, port $189 \times 256 + 39 = 48423$

```
< bro-0.9-current.tar.gz.OLD
< bro-0.9a7.tar.gz
< ... etc ...
< bro-pub-0.9a4a.tar.gz
```

Here the server closes the separate connection.

The server sends this using the control connection again:

```
< 226 Transfer complete.
remote: bro*
283 bytes received in 0.0013 seconds (218.37 Kbytes/s)
ftp>
```

```
ftp> get bro-0.9a9.tar.gz.TYPO
> TYPE I
< 200 TYPE set to I
> PORT 128,32,48,169,189,41
< 200 PORT command successful
> RETR bro-0.9a9.tar.gz.TYPO
< 550 bro-0.9a9.tar.gz.TYPO: No such file or directory
ftp> get bro-0.9a9.tar.gz
> PORT 128,32,48,169,189,42
< 200 PORT command successful
> RETR bro-0.9a9.tar.gz
< 150 Opening BINARY mode data connection for bro-0.9a9.tar.gz (344065
```

The server now transfers the 3MB+ file using a separate connection
To 128.32.48.169, port 189*256+42 = 48426.

When done, it closes the separate connection and continues on the control channel:

```
< 226 Transfer complete.
local: bro-0.9a9.tar.gz remote: bro-0.9a9.tar.gz
3440652 bytes received in 0.81 seconds (4161.86 Kbytes/s)
ftp>
```

```
ftp> cd ..
> CWD ..
< 250 CWD command successful
ftp> pwd
> PWD
< 257 "/" is current directory.
ftp> cd secret-files
> CWD secret-files
< 550 secret-files: No such file or directory
ftp> quit
> QUIT
< 221 Goodbye.
```

PASV Mode

- Command issued by client
- Process:
 - Client issues PASV command
 - Server picks and returns an IP and port number
 - Server must listen on that port number
 - o Rather than initiating a connection upon receipt of a command
 - Client establishes a data connection to the above

23

PASV Mode Example

```
$ telnet 192.168.0.1
21 Trying 192.168.0.1...
Connected to localhost. Escape character is '^]'.
220 FTP Server Ready
USER ftp
331 Please specify the password.
PASS password
230 Login successful.
SYST
215 UNIX Type: L8
PASV
227 Entering Passive Mode (192,168,0,1,84,149)
LIST
150 Here comes the directory listing.
226 Directory send OK.
```



Is there anything here that might cause concern?

24