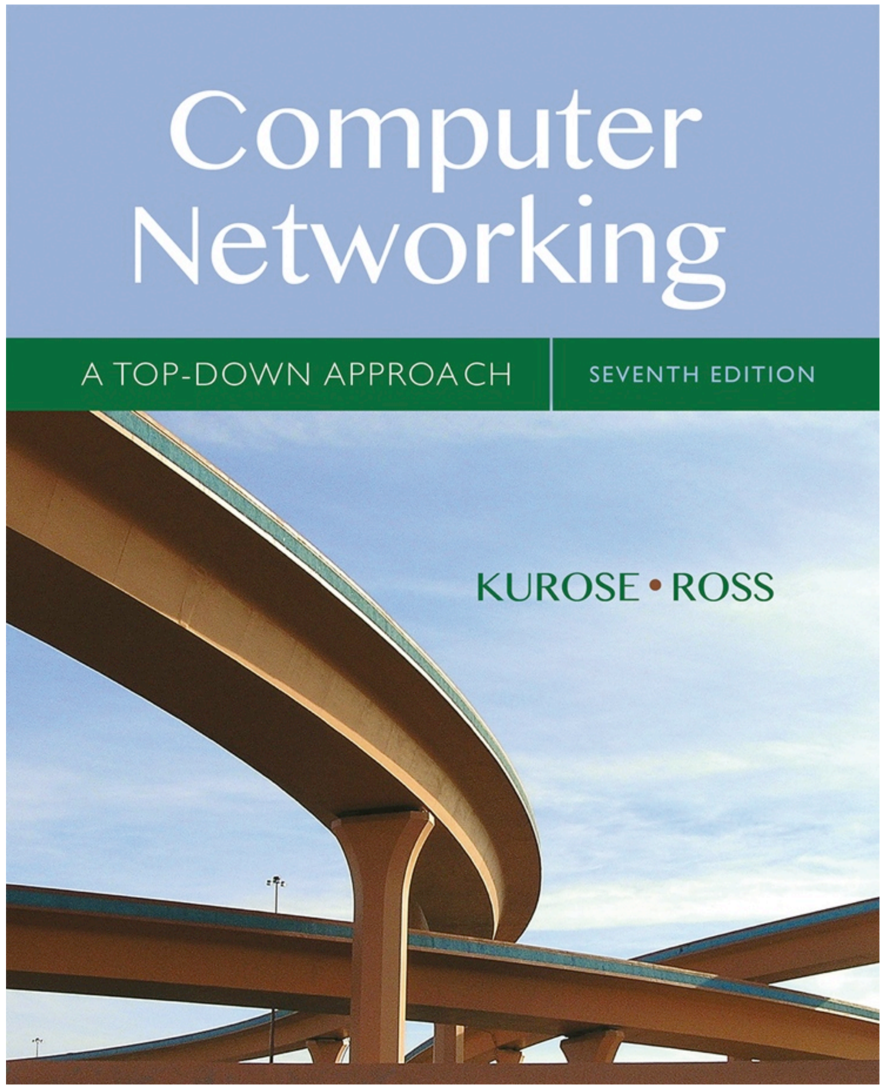
CN-Basic
L17

File Transfer Protocol

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

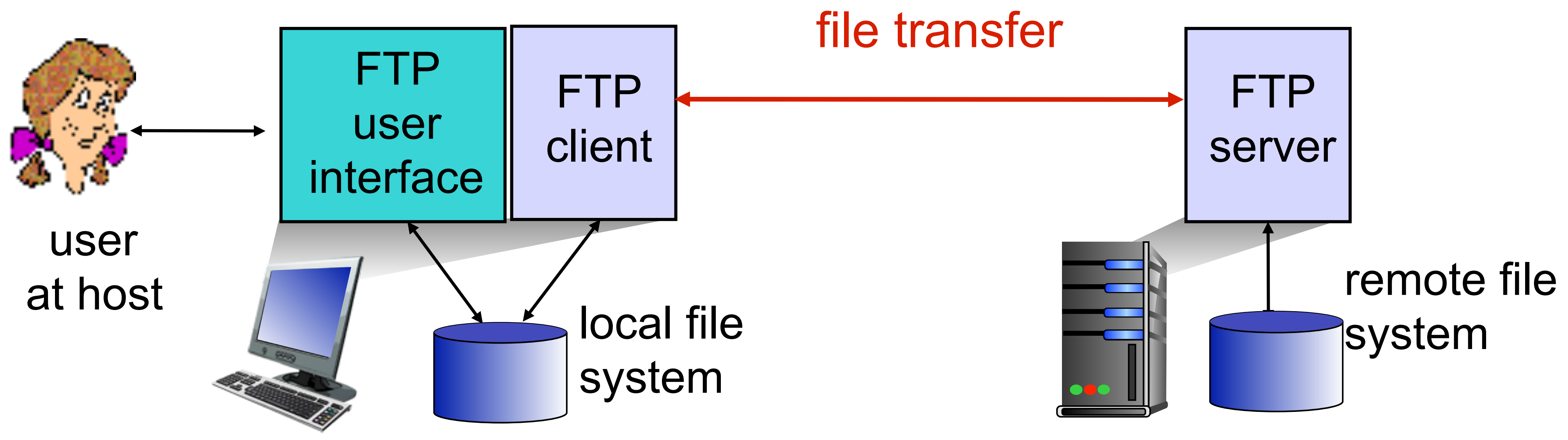
Chapter 2
Application Layer



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

FTP: the file transfer protocol



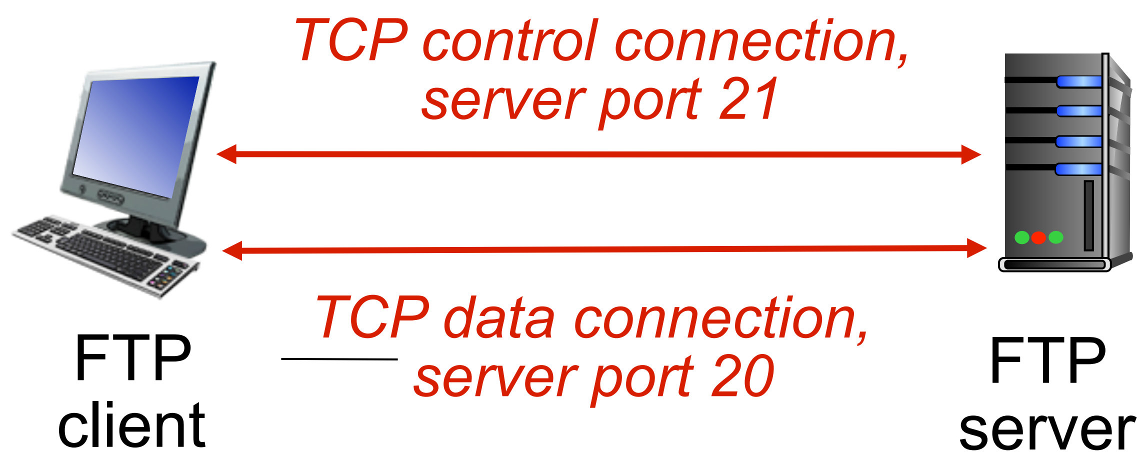
- Transfer file to/from remote host
- Client/server model
- *Client:* side that initiates transfer (either to/from remote)
- *Server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: the file transfer protocol

- Challenges
  - Different filename conventions on two systems
  - Different ways to represent text and data
  - Different directory structures

FTP: separate control, data connections

- FTP client contacts FTP server at port 21, using TCP
- Client gets authorized over control connection
- Client browses remote directory, sends commands over control connection
- When server receives file transfer command, server opens 2nd TCP data connection (for file) *to* client
- After transferring one file, server closes data connection
- Control connection remains open for entire session


TCP control connection, server port 21
TCP data connection, server port 20
FTP client
FTP server

- Server opens another TCP data connection to transfer another file
- Control connection: *"out of band"*
- FTP server maintains "state": current directory, earlier authentication

FTP: data connections

- Opened and closed for each file transfer
- Client issues a passive open using ephemeral port
  - Sends the port number to server
    - Using `PORT a,b,c,d,mm,nn` command
      - a.b.c.d: IP address
      - mm * 256 + nn : Port number
- Server issues the active open using port 20

FTP commands, responses

- *Sample commands:*
- Sent as ASCII text over control channel
- **`USER`** ***`username`***
- **`PASS`** ***`password`***
- **`LIST`** return list of file in current directory
- **`RETR filename`** retrieves (gets) file
- **`STOR filename`** stores (puts) file onto remote host

- Note: These are different from ftp application commands

- *sample return codes*
- Status code and phrase (as in HTTP)
- **`331 Username OK, password required`**
- **`125 data connection already open; transfer starting`**
- **`425 Can't open data connection`**
- **`452 Error writing file`**

FTP - Command Responses

- Each response has two parts
  - Three digit number (xyz) followed by text
- First digit defines the status of the command
  - 1yz: positive preliminary reply
  - 2yz: (positive completion reply)
  - 3yz: (positive intermediate reply)
  - 4yz: (transient negative reply)
  - 5yz: (permanent negative reply)
- 2nd digits provides further information about status
  - 0 (syntax), 1(information) 2(connection), 3(auth/acct)
  - 4 (unspecified), 5(file system)

FTP Example

```
• $ ftp 10.211.55.10
• Connected to 10.211.55.10.
• 220 Welcome to KSIT FTP service.
• Name (10.211.55.10:rprustagi): dummy
• 331 Please specify the password.
• Password:
• 230 Login successful.
• Remote system type is UNIX.
• Using binary mode to transfer files.
• ftp> pwd
• 257 "/home/dummy" is the current directory
• ftp> ls
• 200 PORT command successful. Consider using PASV.
• 150 Here comes the directory listing.
• -rw-r--r--    1 1002     1002         8980 Apr 20  2016 examples.desktop
• 226 Directory send OK.
• ftp>
```

FTP Example...

- **ftp>** `put udp.tcl`
- local: udp.tcl remote: udp.tcl
- 200 PORT command successful. Consider using PASV.
- 150 Ok to send data.
- 226 Transfer complete.
- 2725 bytes sent in 0.00 secs (1.1664 MB/s)
- **ftp>**

FTP Example...

- **ftp>** `get maths.c`
- local: maths.c remote: maths.c
- 200 PORT command successful.
- 150 Opening BINARY mode data connection for 'maths.c' (738 bytes).
- 226 Transfer complete.
- 738 bytes received in 0.00 secs (15014.6 kB/s)
- **ftp>** `quit`
- 221 Goodbye.

- Analyze using wireshark capture: ftpxfer.pcap

FTP Example…

```
ftp> help
Commands may be abbreviated.  Commands are:
!             dir           mdelete       qc            site
$             disconnect    mdir          sendport      size
account       exit          mget          put           status
append        form          mkdir         pwd           struct
ascii         get           mls           quit          system
bell          glob          mode          quote         sunique
binary        hash          modtime       recv          tenex
bye           help          mput          reget         tick
case          idle          newer         rstatus       trace
cd            image         nmap          rhelp         type
cdup          ipany         nlist         rename        user
chmod         ipv4          ntrans        reset         umask
close         ipv6          open          restart       verbose
cr            lcd           prompt        rmdir         ?
delete        ls            passive       runique
debug         macdef        proxy         send
ftp>
```

FTP with firewalls

- Use PASV command
  - Client issues PASV command
  - Server responds with port number
  - Client opens new connection to new port number
  - Data xfer takes place

FTP Protocol Commands Example

```
$ telnet 10.211.55.10 21
Trying 10.211.55.10...
Connected to 10.211.55.10.
Escape character is '^]'.
220 Welcome to KSIT FTP service.
USER dummy
331 Please specify the password.
PASS dummy
230 Login successful.
PWD
257 "/home/dummy" is the current directory
CWD /home/dummyuser
250 Directory successfully changed.
```

FTP Protocol Commands Example

```
PWD
257 "/home/dummyuser" is the current directory
PORT 10,211,55,10,86,206
  # run  a server at port 86*256+206=22222
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
QUIT
```

```
$ nc -l 22222 [# after PORT command]
-rw-------    1 1003     1003       448710 Aug 11 09:07 AFCU-2017-07.pdf
-rw-------    1 1003     1003          522 Aug 11 09:09 Mem04.c
-rw-r--r--    1 1003     1003         8980 Apr 20  2016 examples.desktop
-rw-------    1 1003     1003           24 Aug 11 09:12 goodcgi.sh
-rw-------    1 1003     1003          120 Aug 11 09:17 req-normal.txt
-rw-------    1 1003     1003          118 Aug 11 09:14 shift.c
$
```

FTP Protocol Commands Example

```
230 Login successful.
PASV
227 Entering Passive Mode (10,211,55,10,33,42).
    # server listens on 8490=33*256+42
LIST
150 Here comes the directory listing.
226 Directory send OK.
QUIT
221 Goodbye
$ nc 10.211.55.10 8490 [# client connects to server]
-rw-------   1 1003    1003     448710 Aug 11 09:07 AFCU-2017-07.pdf
-rw-------   1 1003    1003        522 Aug 11 09:09 Mem04.c
-rw-r--r--   1 1003    1003       8980 Apr 20  2016 examples.desktop
-rw-------   1 1003    1003         24 Aug 11 09:12 goodcgi.sh
-rw-------   1 1003    1003        120 Aug 11 09:17 req-normal.txt
-rw-------   1 1003    1003        118 Aug 11 09:14 shift.c
$
```

FTP: Anonymous/TFTP

- Anonymous
  - To provide public access
  - Initially used for download
    - Now happens via web/HTTP
- TFTP
  - Enables file transfer w/o extensive features
  - Typically used at bootstrap time
  - Uses UDP

SFTP/SCP

- Secure FTP (SFTP)
- Secure (SCP)
  - Works on port 22 (ssh)
  - Uses SSL
- Working with certificate keys
  - No explicit login required
  - Need to generate public-private keys
  - Copy the public key (`id_rsa.pub`) to other end and copy it in the file `.ssh/authorized_keys`
    - Use `ssh-copy-id` to accomplish this step in 1 go.

File transfer over HTTP

- The current norm of downloading files
  - Header `Content-Type: application/...`
  - Client will do necessary work
    - save the file

Summary

- Client server protocol
- In band and out of band signalling
    - Data Channel
    - Control channel
- Passive mode
    - server opens data channel