

File Transfer Protocol (FTP)

Kameswari Chebrolu

FTP Protocol (RFC 959)

- Helps transfer files from one host to another
- Based on client-server architecture
- Can connect anonymously, also supports clear-text sign-in
- Can be secured via SSL/TLS as well (FTPS)
 - FTPS not same as SFTP or SCP
- Clients: Command-line, GUI based, web-browsers (ftp://)

Protocol Details

- Employs TCP and server listens on port 21
- Separate control and data channel
 - Control channel used for authorization, browsing directory listing; kept open during a session
 - Data channel supports file transfer; closed after each transfer (one file per connection)
- Through a session, FTP protocol maintains state
 - E..g for a given user, current directory as user explores directory tree

Commands

- ASCII based, sent over control channel
- USER *username*
- PASS *password*
- LIST (return list of files in current directory)
- RETR *filename* (retrieves file)
- STOR *filename* (stores file onto remote host)

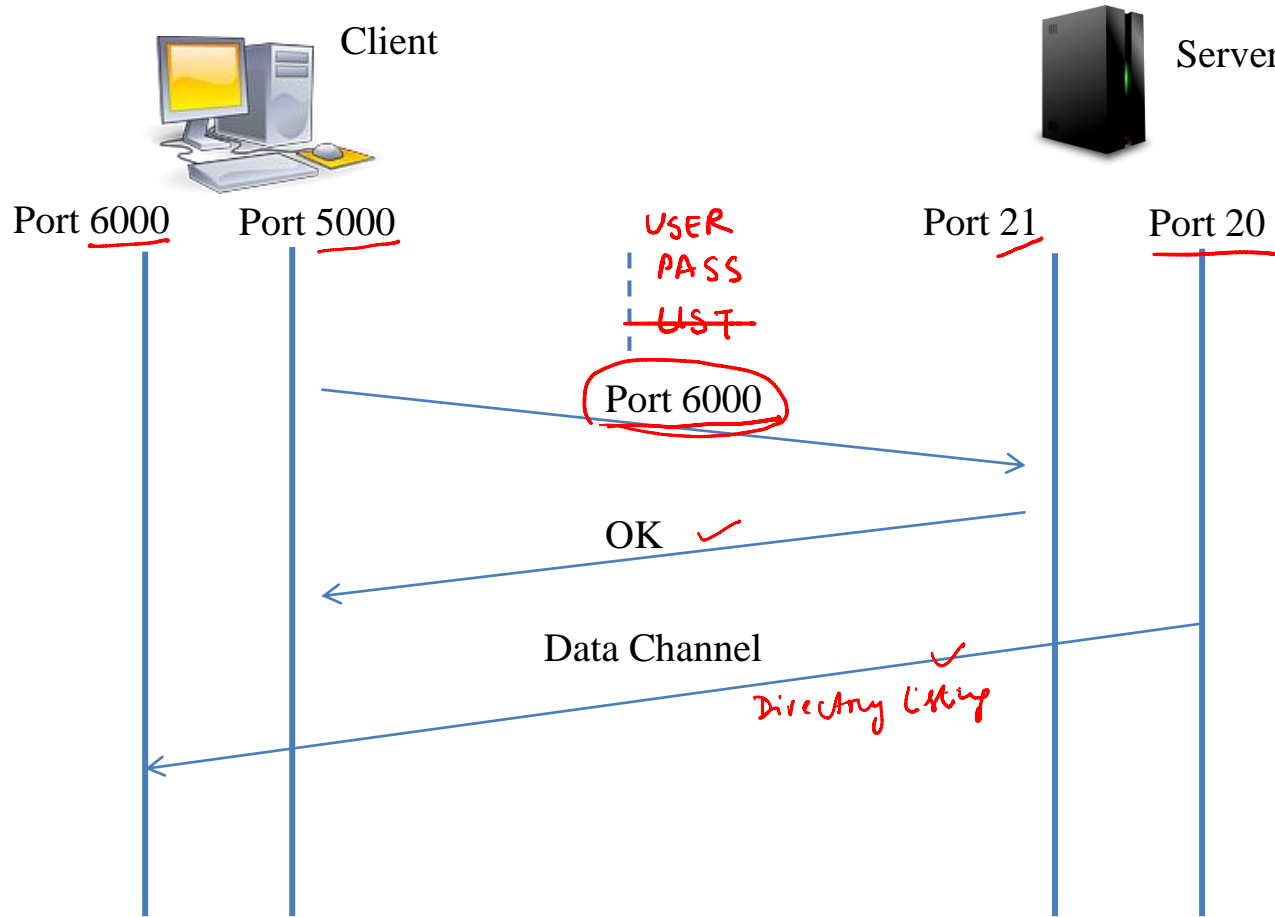
Sample Status Codes and Phrases

- 125 data connection already open; transfer starting
- 220 Service ready for new user
- 331 Username OK, need password
- 425 Can't open data connection
- 553 Requested action not taken. File name not allowed

Active Mode

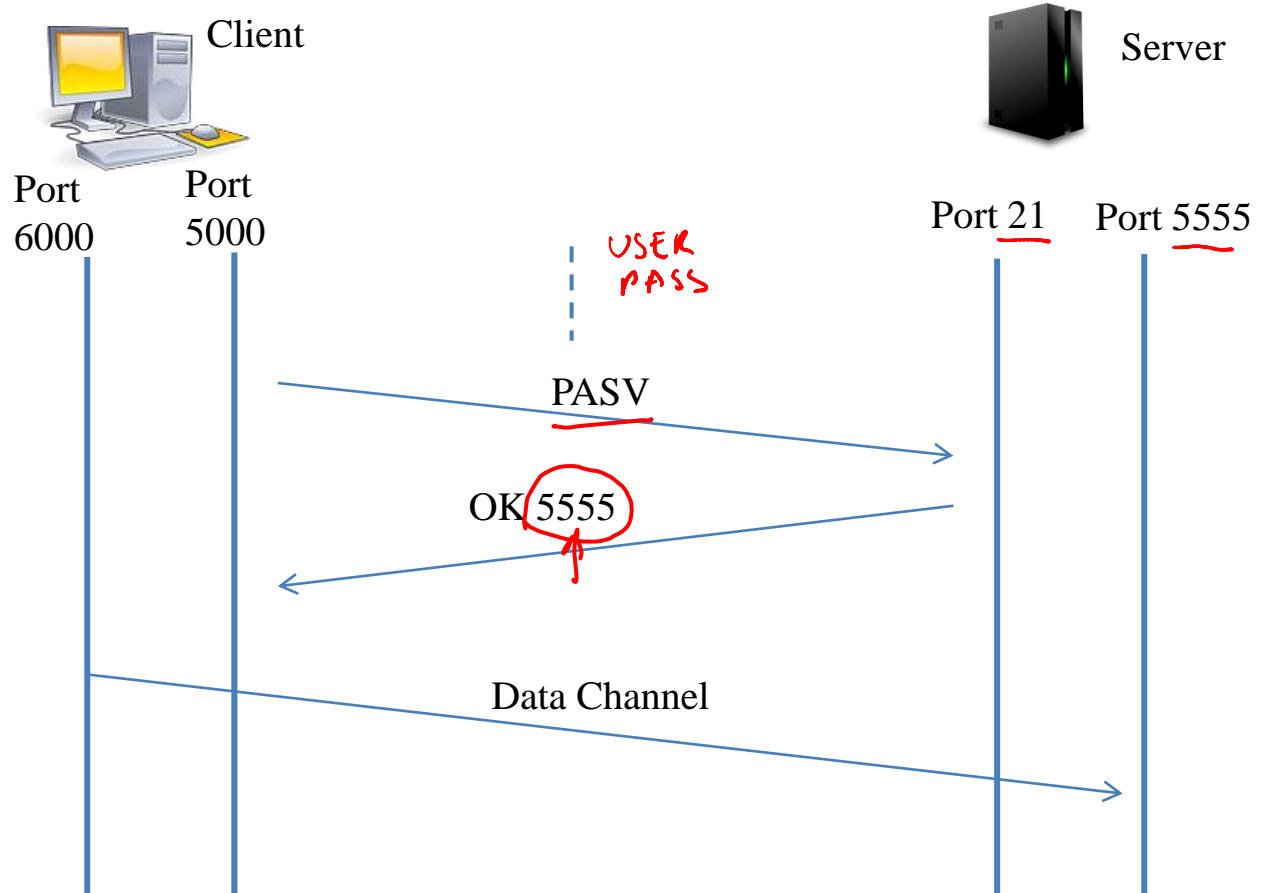
Control → TCP

Data → TCP



Passive Mode

Useful for
NAT/firewall
traversal
(basically
where
incoming
connections
from outside
are blocked)



Demo

- telnet ftp.iitb.ac.in 21
- USER/PASS/PASV
- Another terminal: telnet IP-addr 5th field *
256+6th field
- Original terminal: LIST/QUIT

Summary

- Looked at FTP that is based on client-server architecture
- Maintains separate data and control connections
- Offers two modes of operation: active and passive