Networks (Tutorial). Week 3

Shinnazar Seytnazarov, PhD

Innopolis University

s.seytnazarov@innopolis.ru

January 24, 2022

Topic of the lecture

- The Application Layer
- Application Architecture
- Principles of Network Applications
- Web and HTTP
- FTP

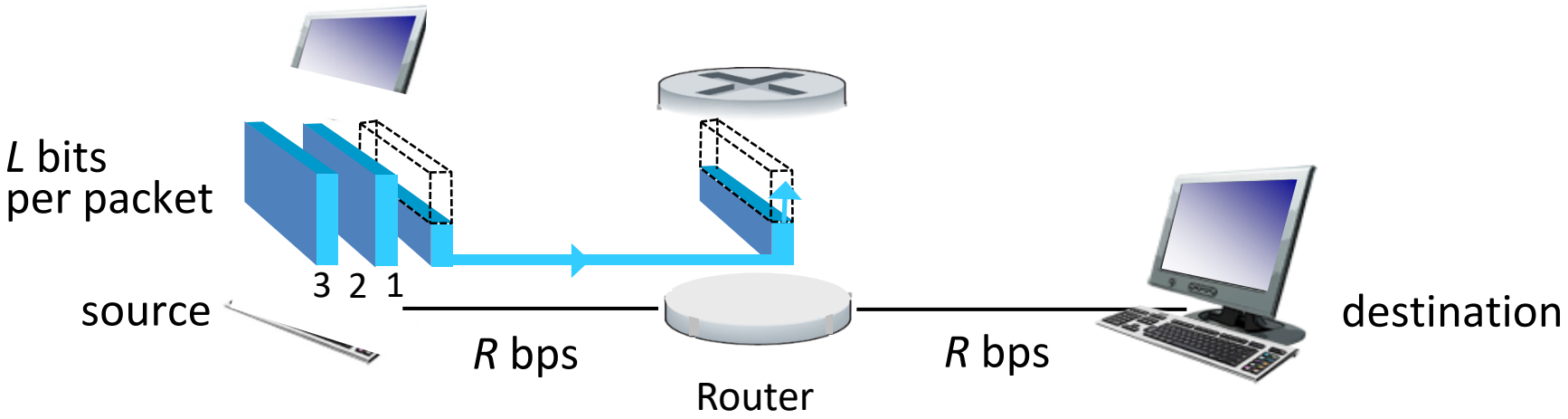
Topic of the tutorial

- The application layer protocol (FTP)

Topic of the lab

- The application layer protocol (FTP)

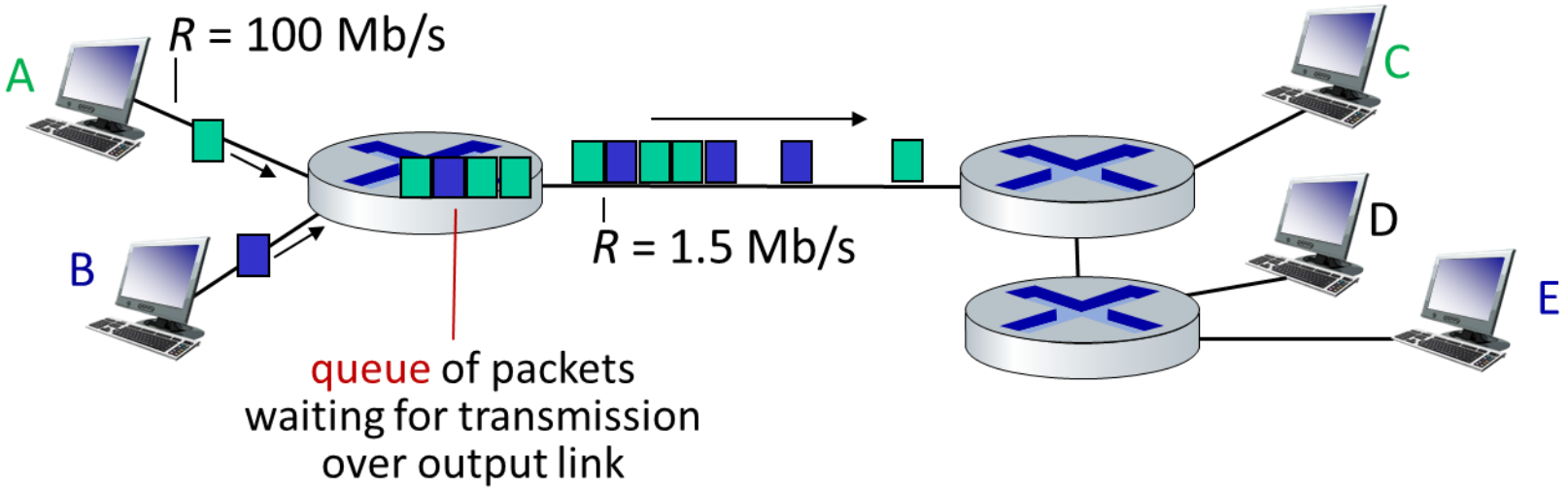
Last class exercise



Source starts transmitting the 1st packet at time 0 and R is link bandwidth

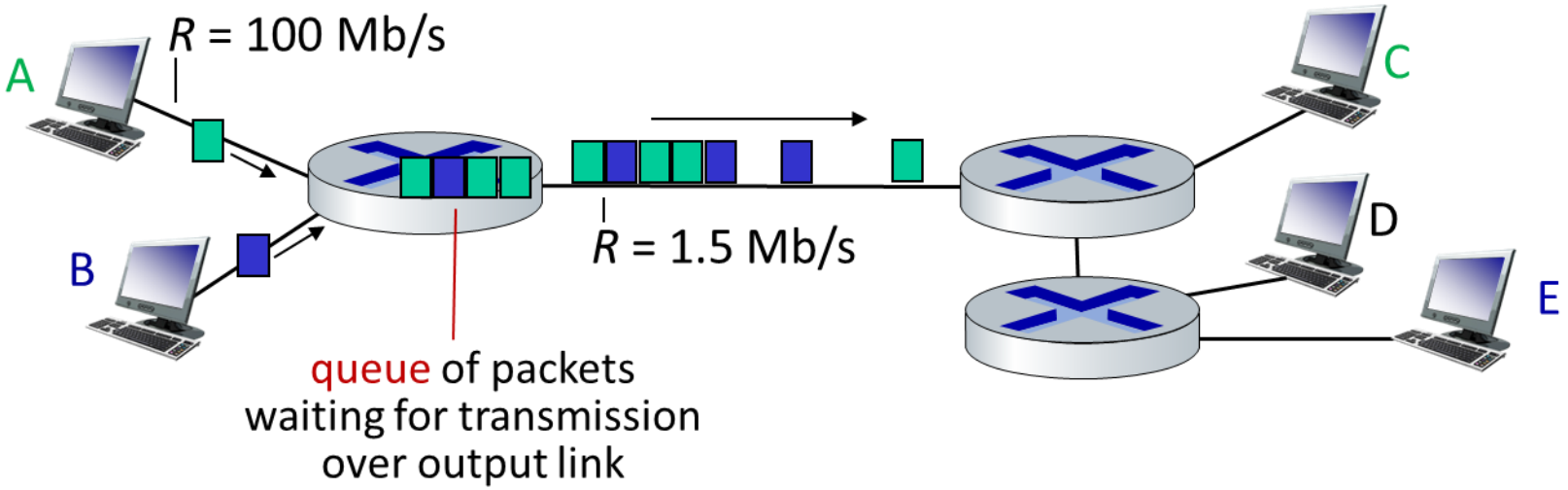
- Q1) assuming zero propagation delay, when the destination receives all three packets? Answer: 4L/R
- Q2) How about if each link has propagation delay of t seconds?
- Q3) What if there were N routers?

Packet-switching: queueing



- Queueing occurs when work arrives faster than it can be serviced:

Packet-switching: queueing



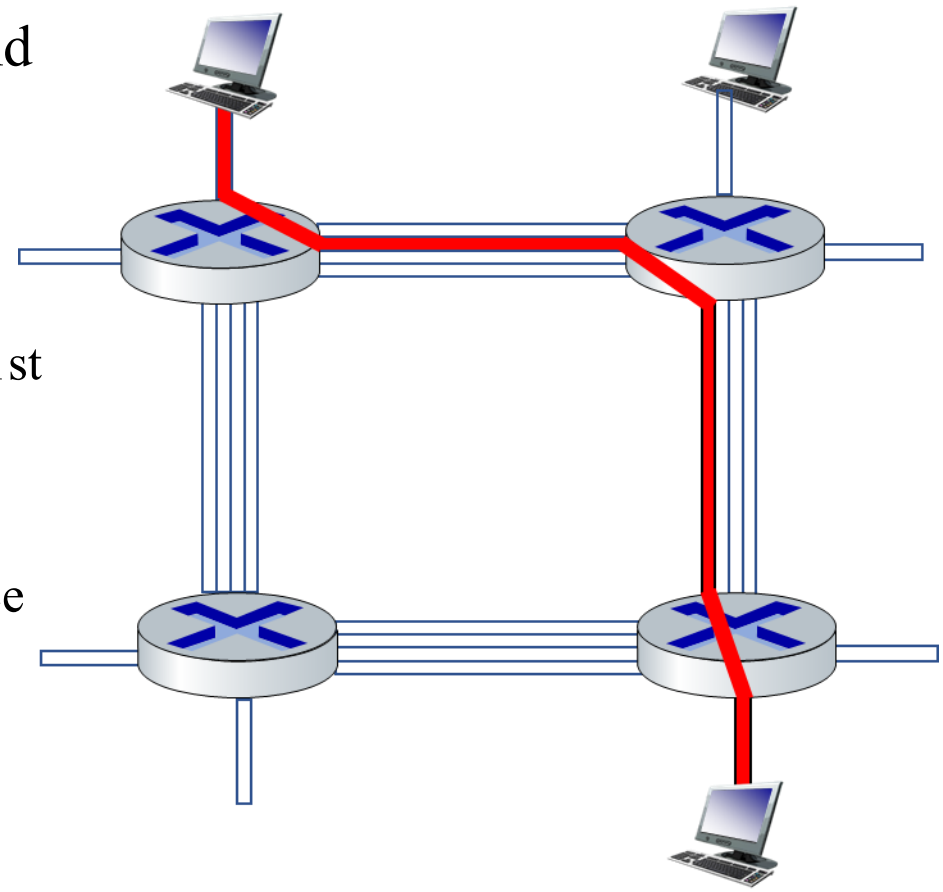
Packet queuing and loss: if arrival rate (in bps) to link exceeds transmission rate (bps) of link for some period of time:

- packets will queue, waiting to be transmitted on output link
- packets can be dropped (lost) if memory (buffer) in router fills up

Alternative to packet switching: circuit switching

End-to-end resources allocated to, reserved for "call" between source and destination

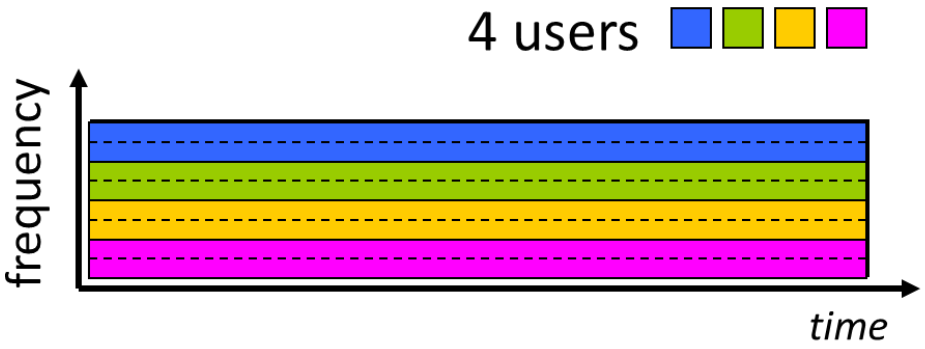
- in diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call (no sharing)
- commonly used in traditional telephone networks



Circuit switching: FDM and TDM

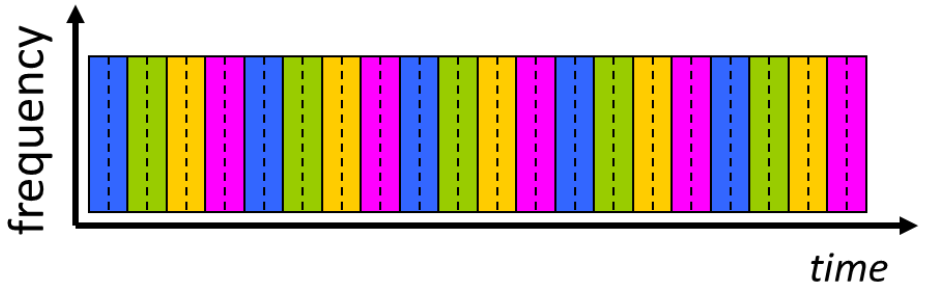
Frequency Division Multiplexing (FDM)

- optical, electromagnetic frequencies divided into (narrow) frequency bands
- each call allocated its own band, can transmit at max rate of that narrow band



Time Division Multiplexing (TDM)

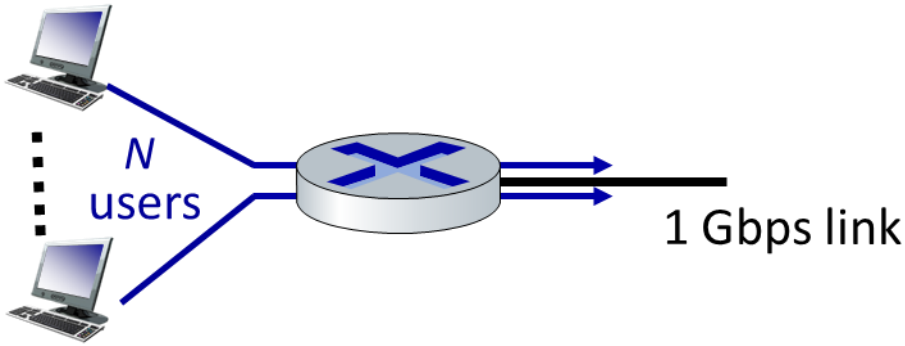
- time divided into slots
- each call allocated periodic slot(s), can transmit at maximum rate of (wider) frequency band (only) during its time slot(s)



Packet switching versus circuit switching

example:

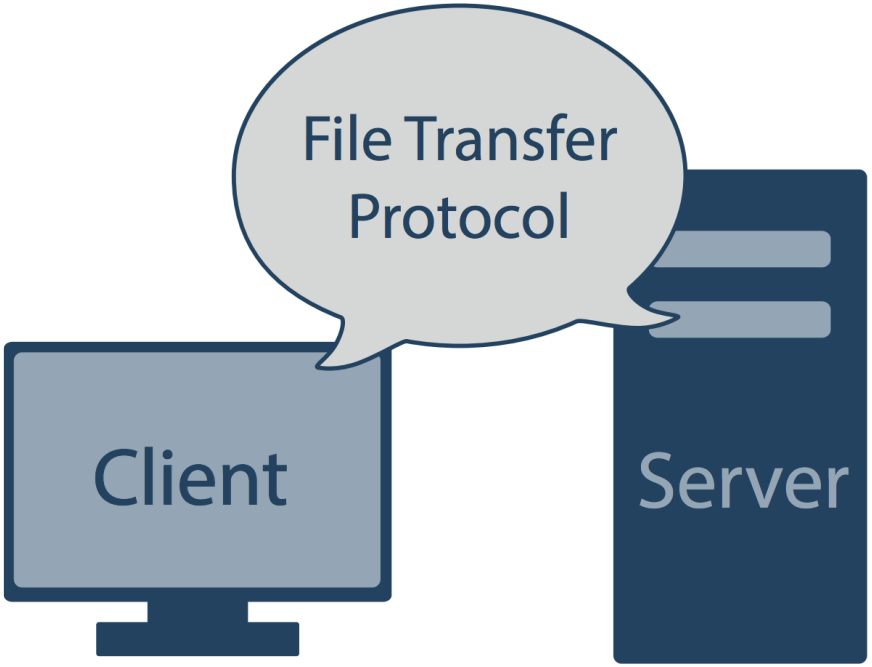
- 1 Gb/s link
- each user:
- 100 Mb/s when “active”
- active 10% of time



Q: how many users can use this network under circuit-switching and packet switching?

- circuit-switching: 10 users
- packet switching: with 35 users, probability > 10 active at same time is less than .0004
- Q: how did we get value 0.0004?

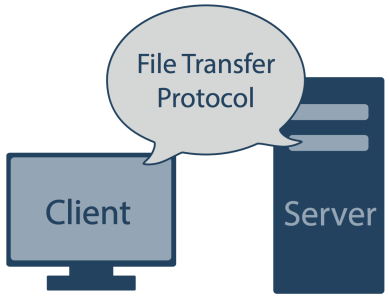
File Transfer Protocol (FTP)



Source: https://southrivertech.com/wp-content/uploads/FTP_Explained1.pdf

History of FTP

- In the early days of computing, complex sets of commands had to be learned to use the Internet.
- FTP, invented in the early 1970s, established a standard protocol for transferring files between systems.
- FTP is a widely-accepted Internet Standard, created and made available through the Internet Engineering Task Force (IETF)

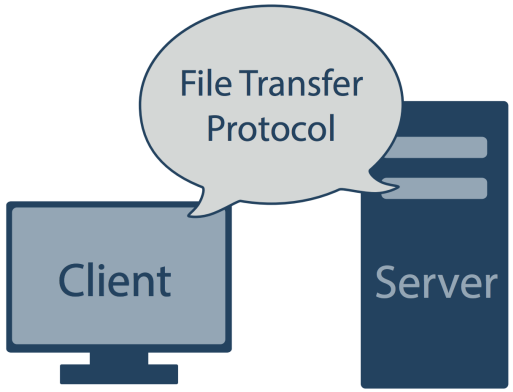


Usage of FTP

- Uploading webpages to web servers for publishing on the Internet
- Browsing and downloading files from public software sites
- Transferring large files among two parties that are too large for email attachments
- Downloading and uploading content like university's assignments via an FTP server
- Distributing the latest revisions of programs by software developers

Basics Functions and Terms of FTP

- To use FTP, we need
  - FTP client
  - FTP server
- You also need to know
  - Server address
  - Username and password
  - Port number



Essential FTP Terms

- **Anonymous FTP**
  - Various public servers allow anonymous login.
  - Users can log in to servers without an account to download files.
  - Uploading is not allowed for anonymous login.
  - Take note that your IP address is tracked even though it is an anonymous session.

Essential FTP Terms

- **Get:** also known as "Download".
  - Copy files from the FTP site to the FTP client's system.

- **Put:** also known as "Upload".
  - Copy files from the client's system to the FTP site.
  - Uploading is restricted to authorized users only.

- **FTP Site:**
  - A hosting server that contains files for download and upload.
  - To access the FTP site, you need to type in the address, which begins with ftp:// (instead of http://)

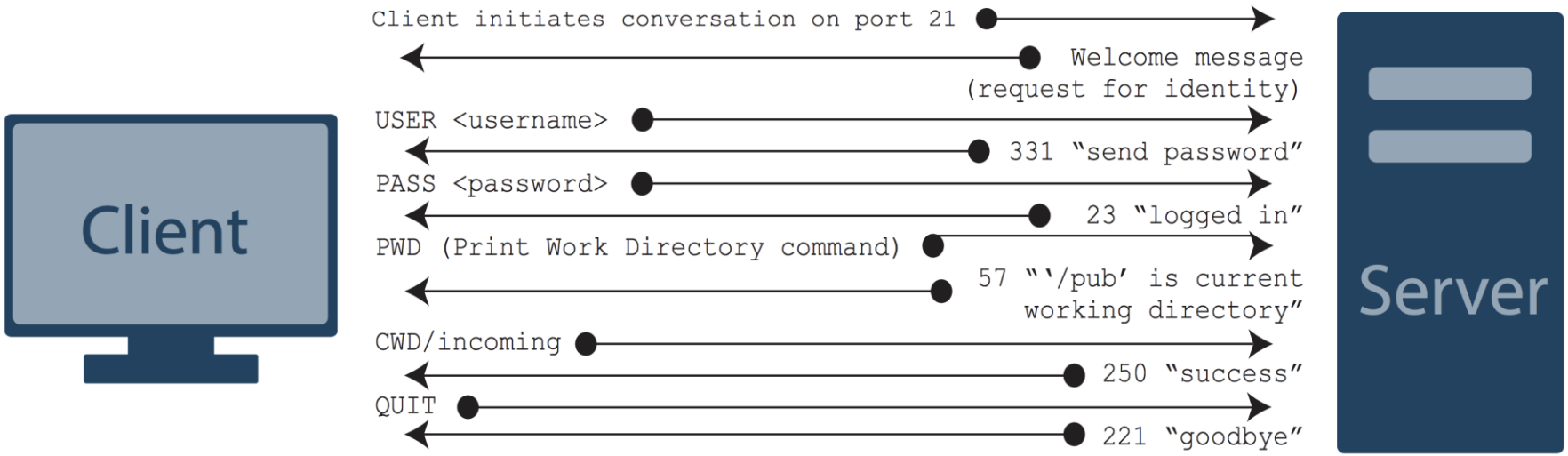
FTP Connections

- An important concept to remember is that FTP connects using two TCP ports for all communications between the server and user.
  - Control Connections
  - Data Connections

Control Connections

- Generally control connection is established on port 21
- It is the primary connection and is used to send commands back and forth between the client and server.
- After establishing a connection or "handshake,"
  - The client issues the retrieve command – RETR, to initiate the file transfer
  - RETR is followed by the name of the file to be retrieved.
- If the file exists and if the client has rights to access the file, the server will issue a reply indicating that everything is OK and that the file transfer will begin.

Client-Server Conversation to Establish FTP Transfer



Source: https://southrivertech.com/wp-content/uploads/FTP_Explained1.pdf

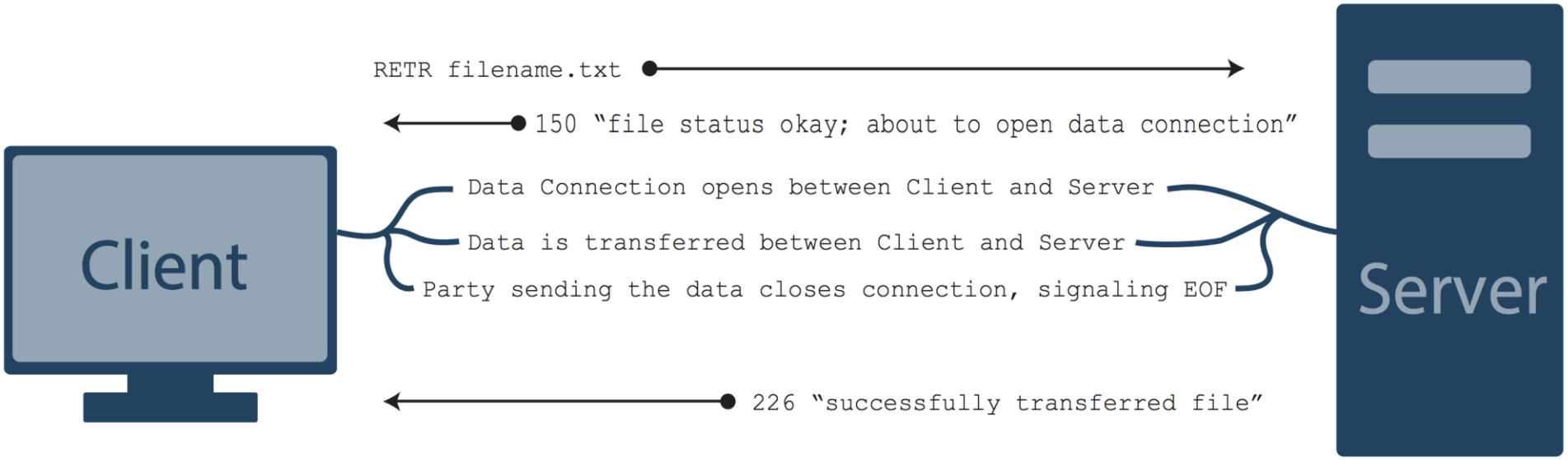
Data Connections

- Using the established control connection, the client and server will create a separate data connection
- It is used solely to transfer the requested data.
- The data connection stays open until the transfer is complete, after which the Data Connection is closed.

Data Connections

- Data connections are opened on a port negotiated by the client and server prior to the RETR command.
- Data connections are closed by either the client or the server, depending on which party is sending the information.
- When a client is retrieving data from a server, the server will close the connection once all data has been transferred.
- When the client is transferring data to the server, the client will terminate the connection when all information is transferred.

File Transfer after Handshake



Source: https://southrivertech.com/wp-content/uploads/FTP_Explained1.pdf

Negotiation of Data Connection

- When a client and server intend to transfer data, they usually negotiate the details of the **Data Connection** prior to opening it.
- Though RFC-959 defines a mechanism for pre-negotiated details
- FTP clients rarely rely on the default values; if the client fails to issue a PASV or PORT command, the data connection defaults to port 20.

Passive and Active Modes

- Nearly all FTP clients will explicitly specify the IP address and port for the data connection each time a file is transferred.

- The IP address used for the original **Control Connection** must be combined with an unused port
- Usually a port numbered higher than 1024 and and lower than 65535.
- Ports below 1024, other than port 20, are reserved for system services.

Passive and Active Modes

- During the address/port negotiation phase, the client will issue either the PORT command (when in Active Mode) or the PASV command (when in Passive Mode).

- **Active Mode** – The client issues a PORT command to the server signaling that the client will "actively" provide an IP and port number to open the **Data Connection** back to the client.

- **Passive Mode** – The client issues a PASV command to indicate that the client will wait "passively" for the server to supply an IP and port number, after which the client will create a Data Connection to the server

Why Active or Passive?

- The ability to choose between active and passive mode data connections primarily comes in handy when navigating firewalls.

- For instance:
  - You have an FTP server sitting behind a firewall.
  - The firewall blocks all incoming traffic except for traffic bound for port 21 (the default for FTP traffic), but the server is able to send information out through the firewall freely.
  - If the FTP client were to issue the **PASV** command to the FTP server, the server would respond with an IP address and port that the client should use to connect back to the server.
  - However, since the firewall is blocking access to all ports except 21, the FTP client will not be able to connect to server's chosen port.
  - To correct this problem, the FTP client would need to issue the **PORT** command to the server.

Why Active or Passive?

- Since the client is now the active entity in establishing the connection, the server would be able to open an outbound connection of its choice through the firewall to the FTP client's designated port to make a connection only long enough to make a secure transaction.

Handling Firewalls

- However, due to today's security-conscious environment:
- Clients usually have a firewall as well, which would prevent the server from opening a connection back to the client in response to the PORT command.

Handling Firewalls

- Many FTP servers now specify a Passive Port Range/Block, which defines a pool of ports set aside for connections with the FTP server, which would be applied to the firewall.

- The firewall administrator would then set up routing rules on the corporate Firewall so that any connections from clients on these ports would be automatically forwarded to the FTP Server.

- The corporate FTP Server could then use PASV mode to instruct the client to use one of the ports within the allowed server port range.

File Transfer Modes

- FTP transfers files between systems by using one of these two modes
  - ASCII
  - Binary

- The mode is determined at the initial stage of all FTP transactions by the server.

- The FTP client will automatically switch to the mode.

ASCII Mode

- These file types are safe for ASCII transfer
  - Text files
  - HTML files
  - CGI scripts

Binary Mode

- These file types must be transferred in binary mode
  - Images
  - Applications
  - .zip, .sit or .tar packages
  - Proprietary file formats such as .doc, .xls, .fla, .swf
  - Anything that's not made entirely of text characters!

FTP Features

- **Interactive Access**
  - FTP provides an interactive interface to allow humans to interact with remote servers.
- **Format Specification**
  - FTP allows the client to specify the type and representation of stored data.
    - The user can specify whether a file contains text or binary data.
- **Authentication Control**
  - FTP requires clients to authorize themselves by sending a login name and password to the server before requesting file transfers.
    - The server refuses access to clients that cannot provide a valid login and password.

FTP Process Model

- FTP server implementations allow simultaneous access by multiple clients
- Clients use TCP to connect to a server.
- The FTP server process awaits connections and creates a slave process to handle each connection.
- The slave process accepts and handles a control connection from the client.

Security

- An encrypted connection secures the data while it is transferred between systems.

- FTP connections are usually not encrypted but some FTP servers may require or offer an encrypted connection.

- The types of encryption are:
  - Implicit SSL
  - Explicit SSL
  - SFTP

Security – Implicit SSL

- Only SSL supported clients are allowed access.
- Secured communication is setup from the beginning of the connection.
- Server and client do not transmit clear text during the session.
- Default SSL port is 990.

Security – Explicit SSL

- A mix of non-secure and secure clients are allowed.

- Unencrypted FTP connection are established but can be upgraded to a secure connection when sensitive data are requested for sending.

Security – SFTP

- SFTP stands for Secure FTP.
- It uses secure shell connection (SSH) and requires encrypted public key authentication.
- Files are transferred between computers over a SSH secure data stream.

Console FTP clients

- **NcFTP** is the all−time favorite ftp client of many Unix users. It comes bundled with most Linux distributions

- **lftp** is a sophisticated command line based FTP client. Like bash, it has job control. It uses the GNU readline library for input, so you have command line completion and editing.

- **Comfortable FTP (cftp)** is a full screen mode client. What it lacks in features, it makes up for in ease of use. You browse through the directories using the arrow keys and enter.

Graphical FTP clients

- **gFTP** is an FTP client for X Windows. It feature
  - Features simultaneous downloads, resuming of interrupted file transfers, file transfer queues, downloading of entire directories, ftp proxy support, remote directory caching, passive and non−passive file transfers, drag−n−drop support, a very nice connection manager and more.

- **LLNL XFTP** was one of the first graphical FTP clients for Linux. It supports FXP (file transfer between two remote hosts), and has a Motif based interface.

- **Guiftp** is a simple ftp client written with the GTK+ toolkit. It's good if you don't need many features and want a simple, clean look.

FTP - Servers

- **ProFTPD** is a powerful FTP server that includes Apache−style configuration, extensive support for virtual hosts, and internal ls.

- **WU−FTPD** is the ftp daemon included with many Linux distributions, including Red Hat and Caldera. You can learn more about WU−FTPD at http://www.wu−ftpd.org.

Summary

- FTP has been around for a long time and while its popularity has decreased since the introduction of cloud services

- It is still commonly used by administrators for file uploads to the web server and file data transfer/backups to FTP servers.

Acknowledgements

- Most part of this tutorial was prepared by M.Fahim, G.Succi, and A.Tormasov

Reference

- This tutorial is based on the following documents as well as lecture materials over the internet.
  - http://users.cs.cf.ac.uk/Dave.Marshall/Internet/Lectures/Handout_4_PF.pdf
  - https://www.tldp.org/HOWTO/pdf/FTP.pdf
  - https://www.2brightsparks.com/resources/articles/an-introduction-to-ftp.pdf
  - https://southrivertech.com/wp-content/uploads/FTP_Explained1.pdf