

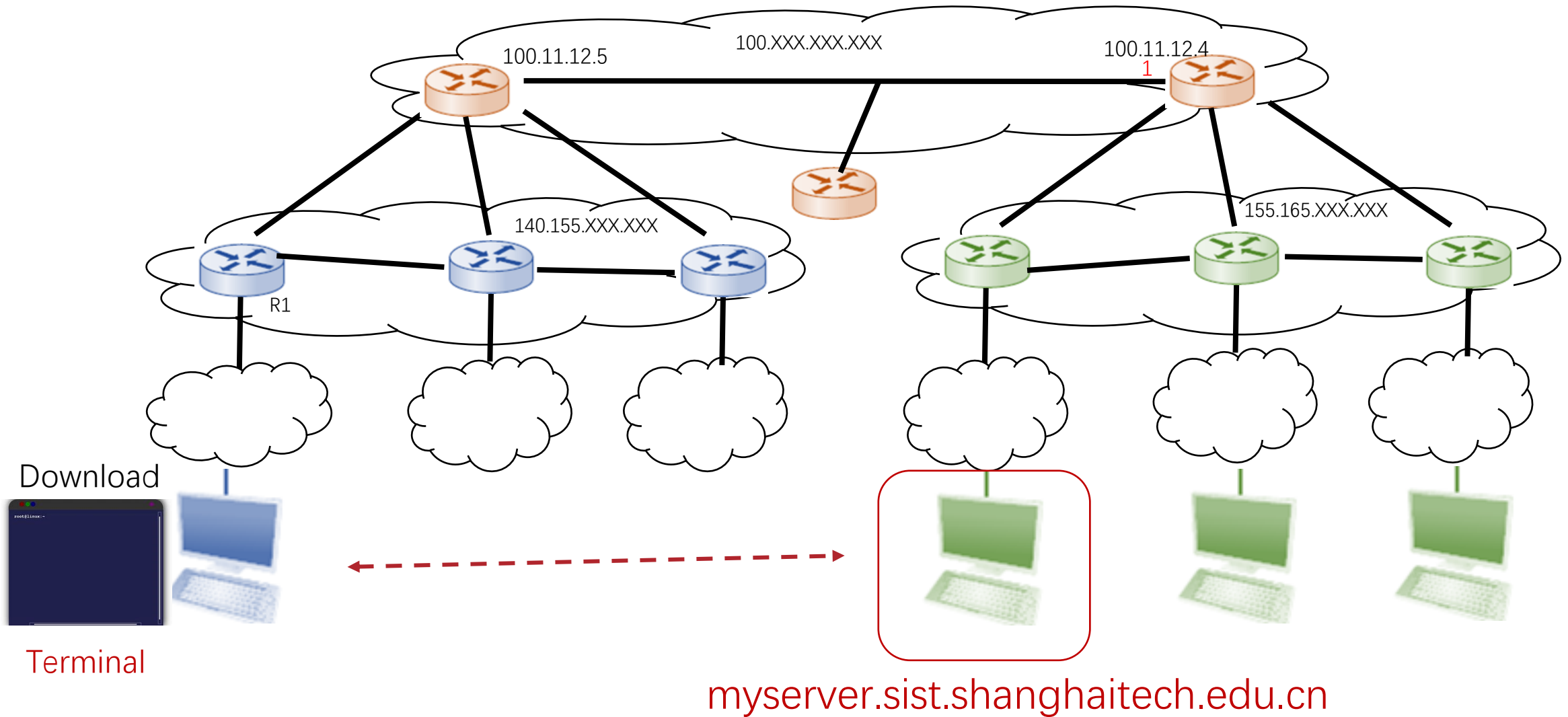


CS120: Computer Networks

Lecture 26. FTP & P2P

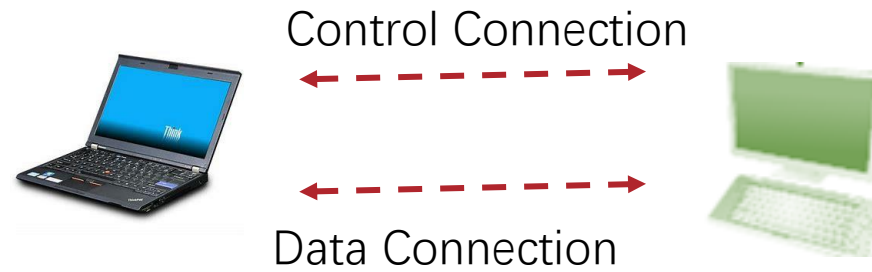
Zhice Yang

File Service



File Transfer Protocol

- FTP: RFC 959
- Use TCP
- Two Connections
 - Control Connection
 - Server Port 21
 - Control Command
 - Authentication
 - Show Directory
 - Data Connection
 - Open one TCP connection for transferring a data stream
 - One data stream one data connection
- Two Working Mode
 - Passive Mode: client connects to server for data connection
 - Active Mode: server connects to client for data connection



File Transfer Protocol

- Control Connection
 - Like HTTP, Messages are Text-oriented

ABOR - **abort** a file transfer

CWD - **change working directory**

DELE - **delete** a remote file

LIST - **list** remote files

MDTM - return the **modification time** of a file

MKD - **make** a remote **directory**

NLST - **name list** of remote directory

PASS - send **password**

PASV - enter **passive** mode

PORT - open a data **port**

PWD - **print working directory**

QUIT - terminate the connection

RETR - **retrieve** a remote file

RMD - **remove** a remote **directory**

RNFR - **rename from**

RNTO - **rename to**

SITE - **site**-specific commands

SIZE - return the **size** of a file

STOR - **store** a file on the remote host

TYPE - set transfer **type**

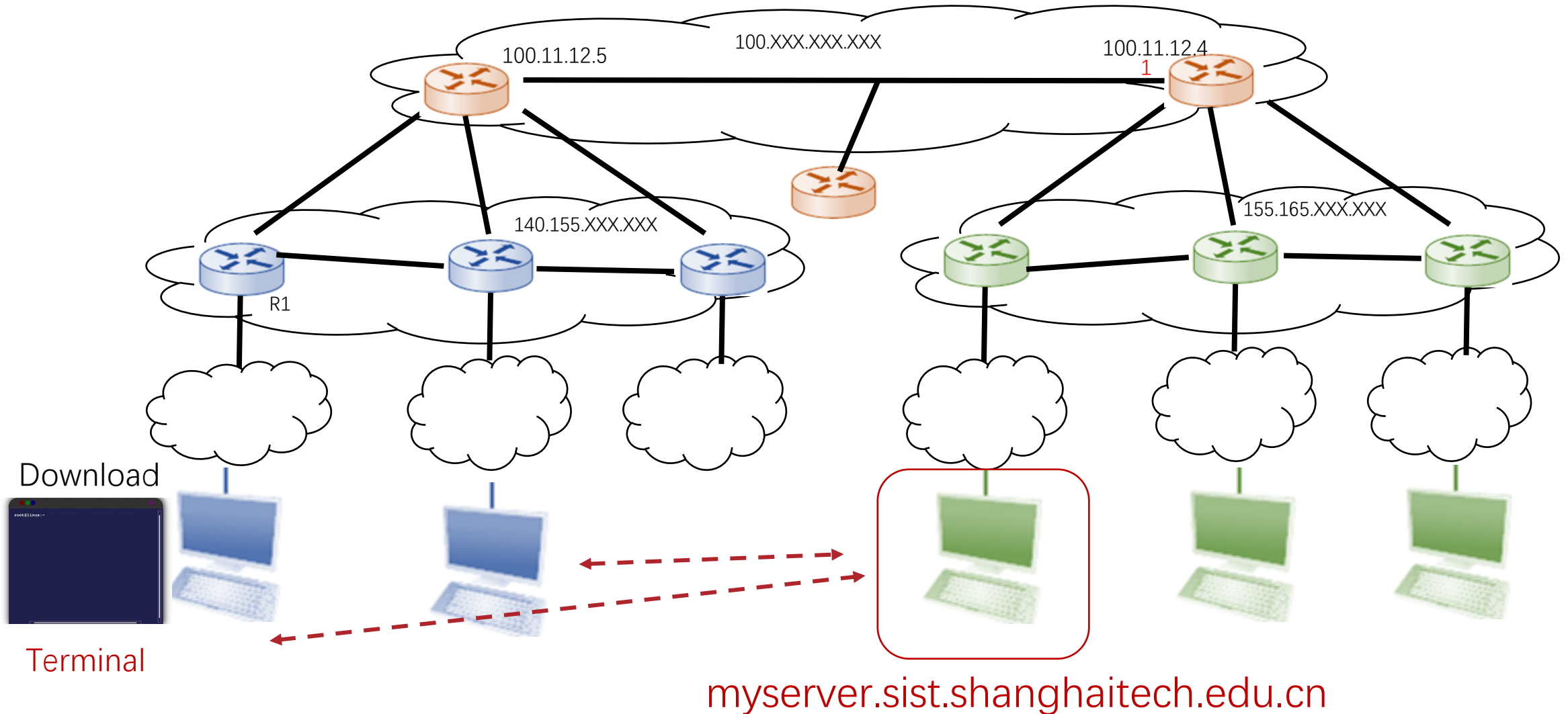
USER - send **username**

Demo

- Telnet
- FileZilla

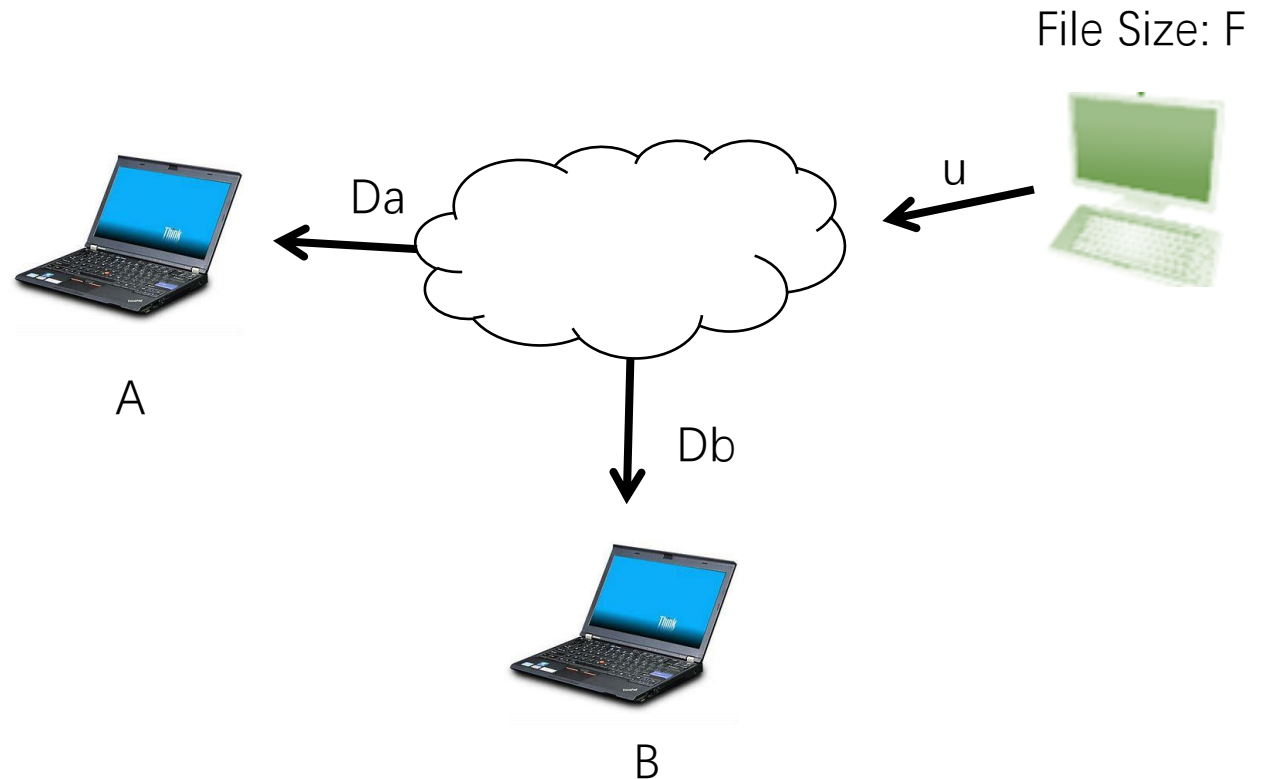
```
Status:    Connecting to 163.22.12.51:21...
Status:    Connection established, waiting for welcome message...
Response:  220- *- National Chi Nan University FTP Service *-
Response:  220
Command:   AUTH TLS
Response:  530 Please login with USER and PASS.
Command:   AUTH SSL
Response:  530 Please login with USER and PASS.
Status:    Insecure server, it does not support FTP over TLS.
Command:   USER anonymous
Response:  331 Please specify the password.
Command:   PASS *****
Response:  230 Login successful.
Command:   OPTS UTF8 ON
Response:  200 Always in UTF8 mode.
Status:    Logged in
Status:    Retrieving directory listing...
Command:   PWD
```

File Service for Multiple Clients



File Service for Multiple Clients

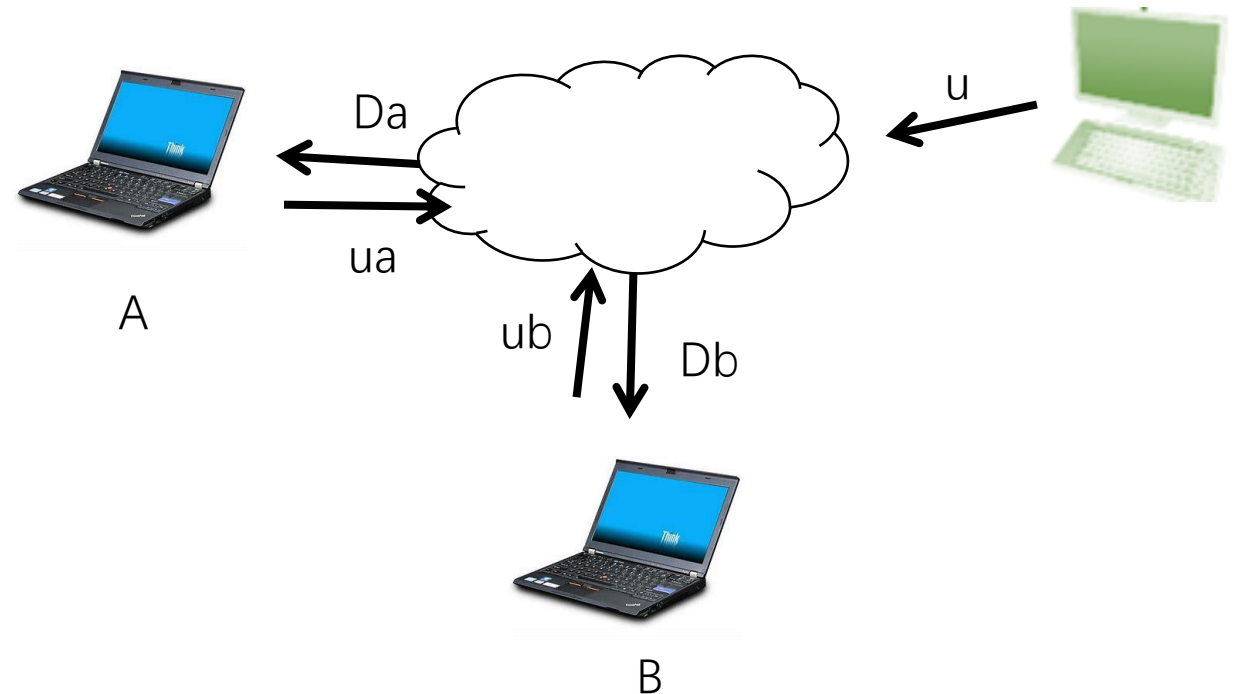
- Simple Approach
 - Server: sequentially send (upload) file copies
 - Client: download file copy
 - Total Time
 - $\text{Max} \{2 \cdot F/u, F/D_a, F/D_b\}$



File Service for Multiple Clients

- Peer to Peer (P2P) Approach

- Server: transmit (upload) file copies to clients
- Client: download file copies and transmit file copies to other clients
- Total Time
 - $\text{Max} \{F/D_a, F/D_b, 2F/(u+u_a+u_b)\}$



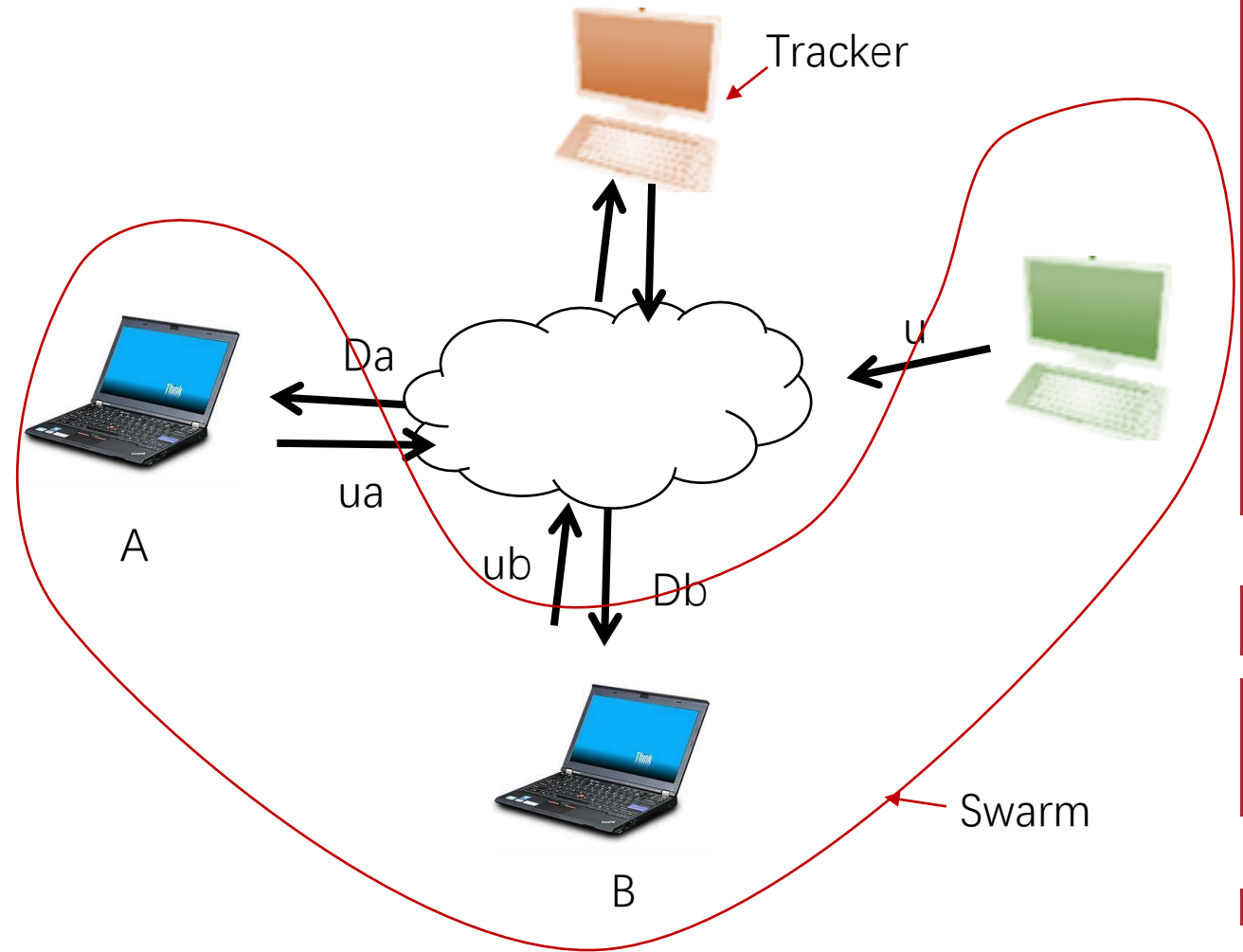
P2P File Distribution: BitTorrent

- BitTorrent is a P2P file sharing system
 - Client: BitTorrent, uTorrent, Thunder, etc



BitTorrent

- The tracker is a central server keeping a list of all peers participating in the swarm
- A swarm is the set of peers that are participating in distributing the same files
- Peer joins a swarm by asking the tracker for a peer list and connects to those peers

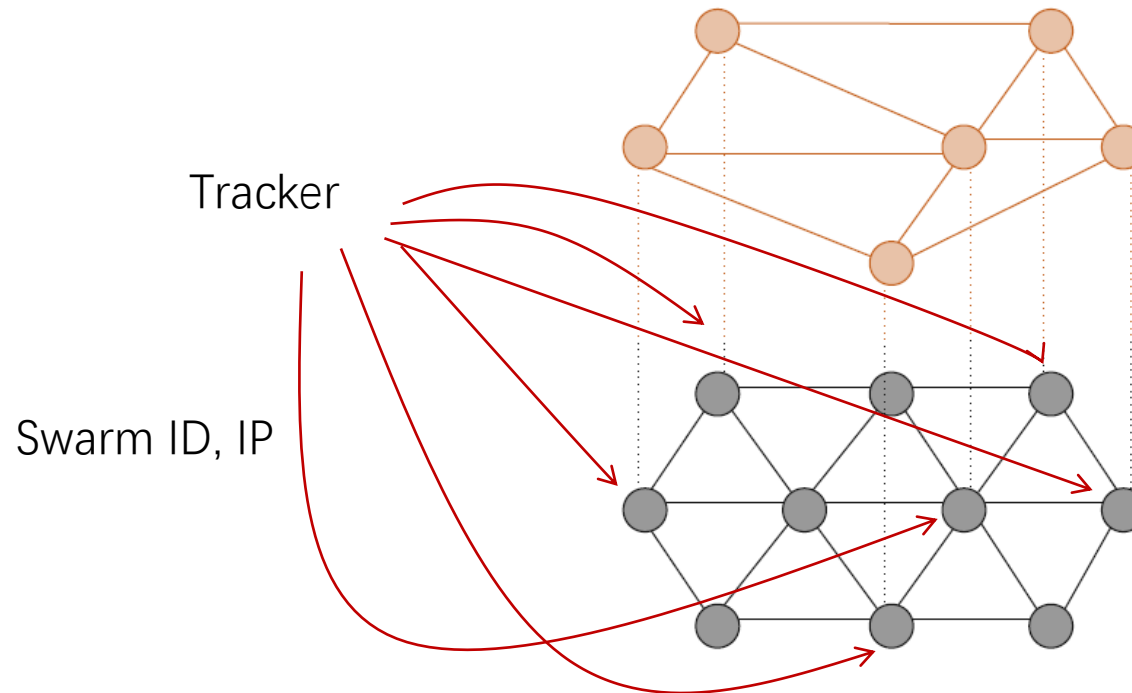


BitTorrent

- A metadata file (**.torrent**) is distributed to all peers
 - Usually via HTTP
 - **.torrent** is encoded with “B-encode”
 - Tools to decode:
https://www.tools4noobs.com/online_tools/torrent_decode/
 - Contains
 - File names
 - SHA-1 hashes of all pieces of the file
 - <http://www.sha1-online.com/>
 - Tracker's url
 - Tracker list
 - Info-hash
 - etc.

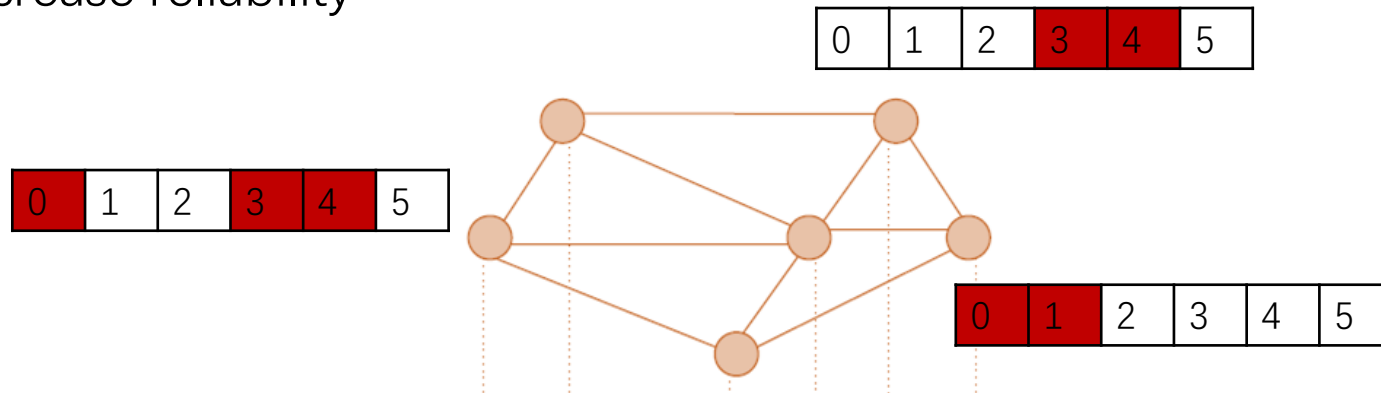
BitTorrent

- The Overlay Networks in P2P
 - Tracker tracks peer information
 - New peer registers with tracker to get list of peers
 - Download files from peers through TCP



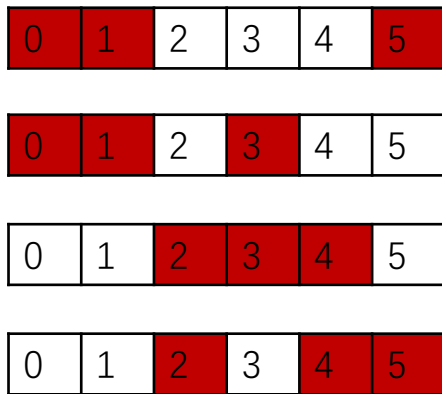
BitTorrent

- File Distribution
 - Peers may have different pieces of file
 - Upload pieces while downloading
 - New peer has no pieces
 - But will accumulate over time
 - Peers exchange information of the pieces they have
 - To maximize throughput
 - To Increase reliability

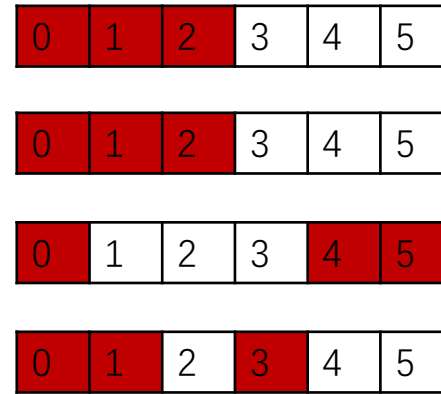


BitTorrent

- Piece Overlap
 - Big overlap -> Only a few peers can exchange pieces
 - Minimize piece overlap
 - Download random pieces
 - Priorities the rarest pieces, aiming towards uniform piece distribution



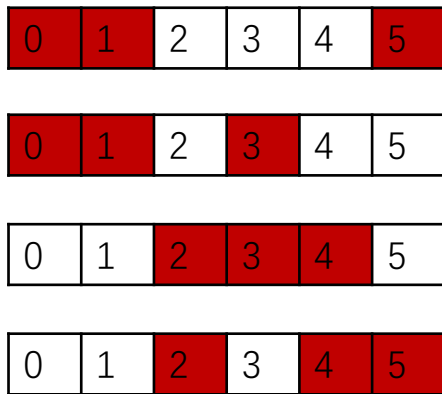
Small overlap



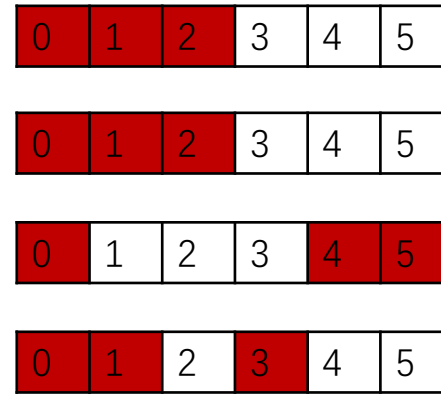
Big overlap

BitTorrent

- Piece Redundancy
 - Be tolerant against dropping peers
 - Maximize piece redundancy
 - Maximize the number of distributed copies (the rarest pieces)
 - Download the rarest pieces first



Distributed copies = 2



Distributed copies = 1

BitTorrent

- The Last Piece
 - The download time of the last piece could be longer than other pieces
 - Pieces with fast download speed have been finished
 - Increase download choices for the last piece
 - Assign more peers to transmit

BitTorrent

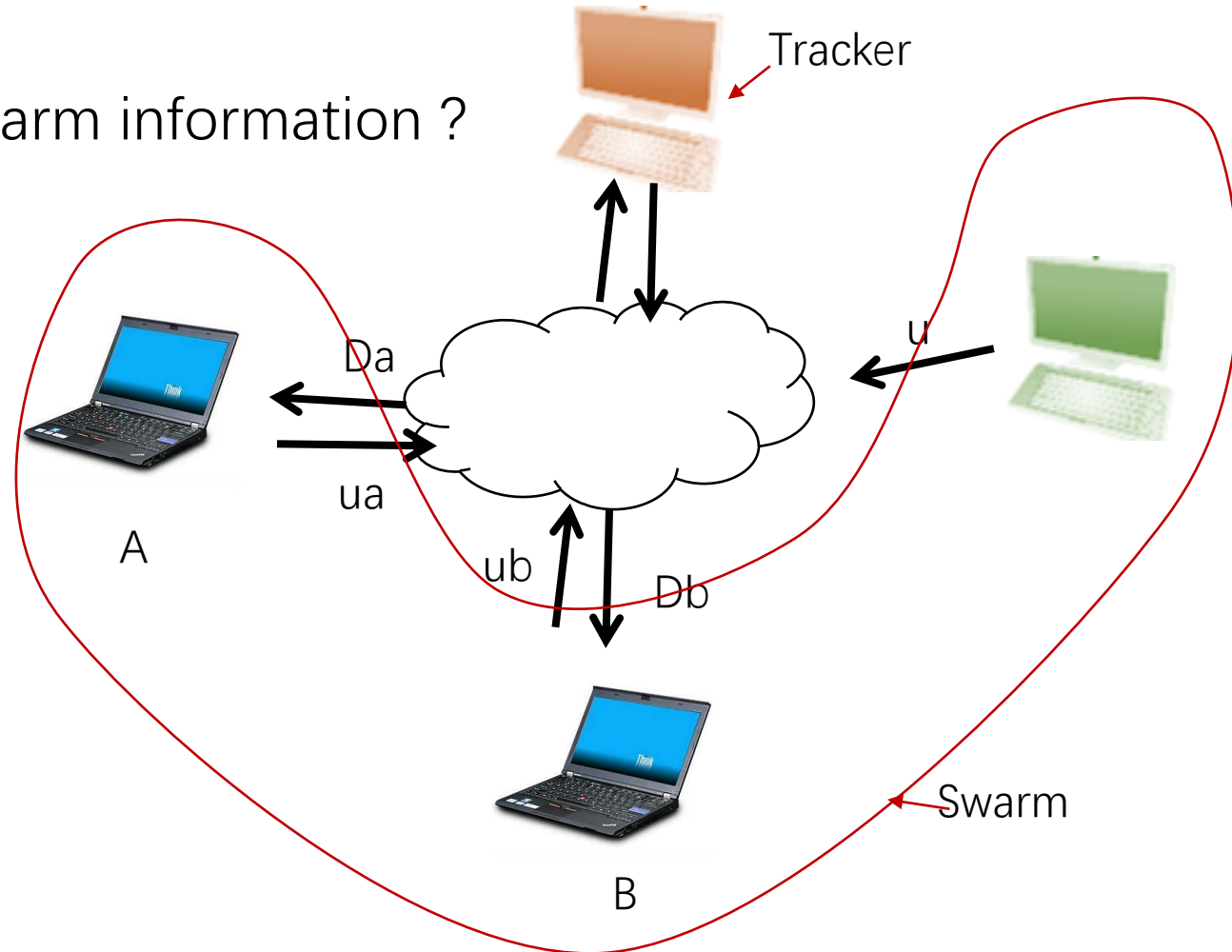
- The Piece Picking Policies
 - Random First Piece
 - Rarest Piece First
 - The End Game Mode
 - Send request to all peers to download the last piece

BitTorrent

- The Incentive to Share
 - There is a loose connection between upload and download speed
 - Each peer has an incentive to upload

BitTorrent

- Trackerless Design
 - Where to store the swarm information ?



Reference

- Textbook 9.4
- http://www.bittorrent.org/beps/bep_0005.html
- <https://www.youtube.com/watch?v=YFV908uoLPY>