



COMP 4021

Internet Computing

Internet Models

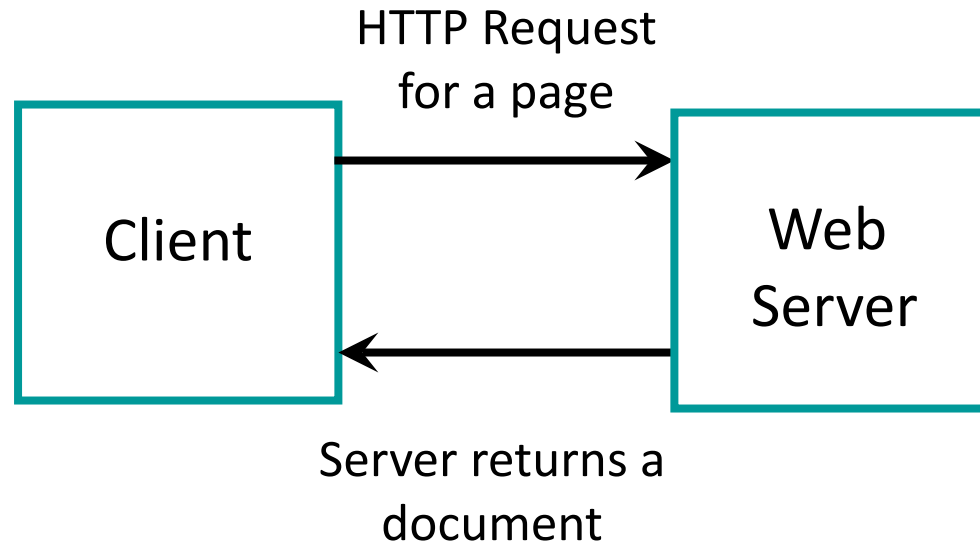
Dr. Kenneth LEUNG

Slides created by Dr. David ROSSITER

Internet Models

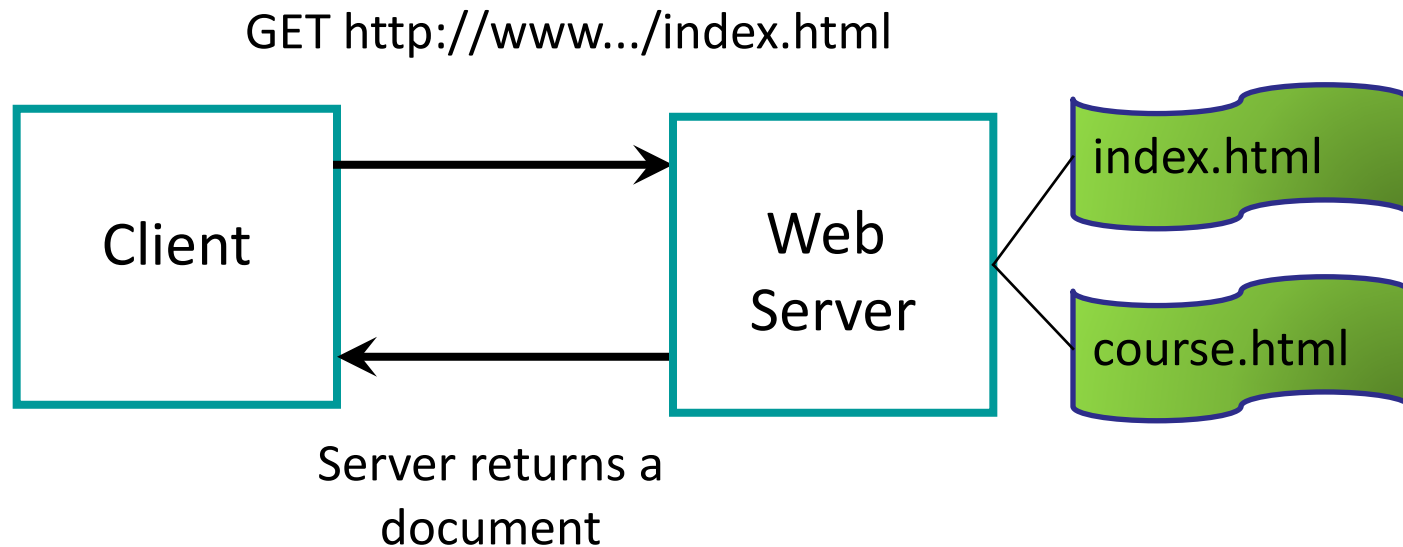
- This presentation gives you an idea of the most common internet models
- ‘model’ = the way in which the system is structured
- Discussed in this presentation:
 - Client-server
 - Server based sharing
 - P2P
 - BitTorrent

Client-Server (2-Tier)



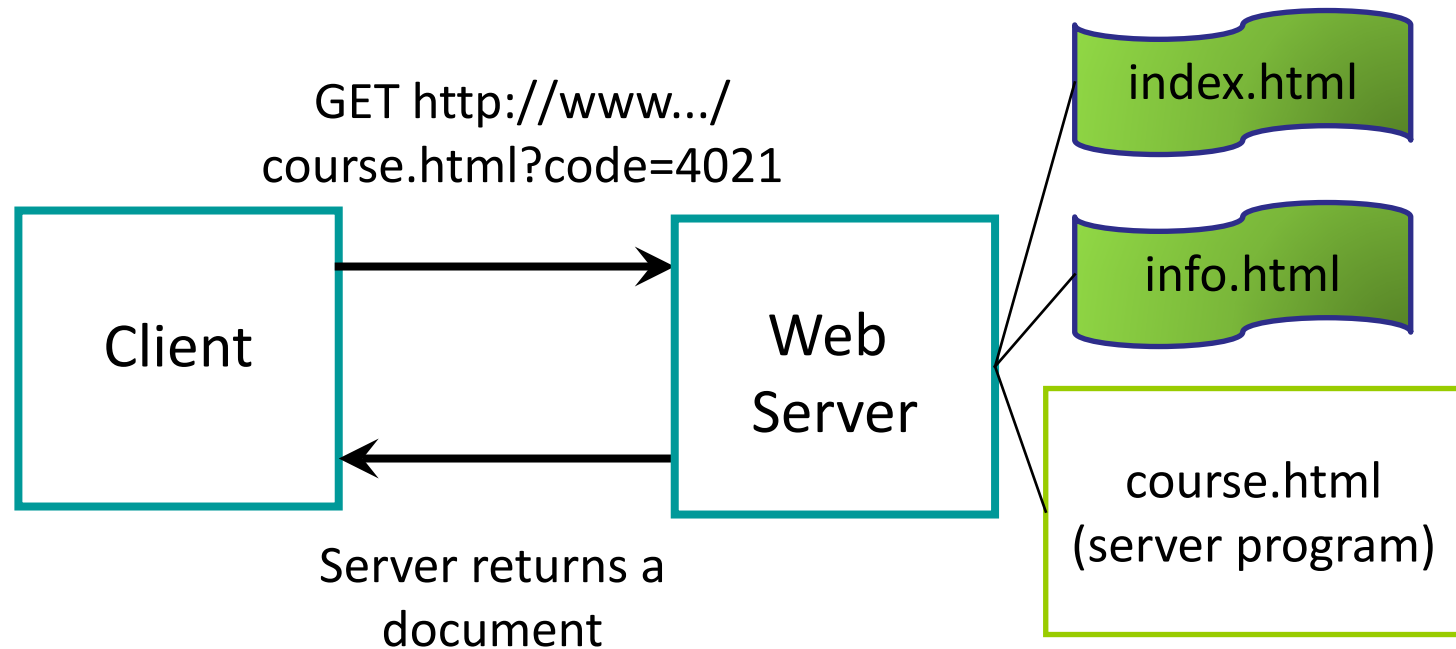
- Client-server is a general term, which include non-http (non-web) clients and servers
- Popular web servers: Apache, IIS, WebSphere, etc.

Client-Server: Static Pages



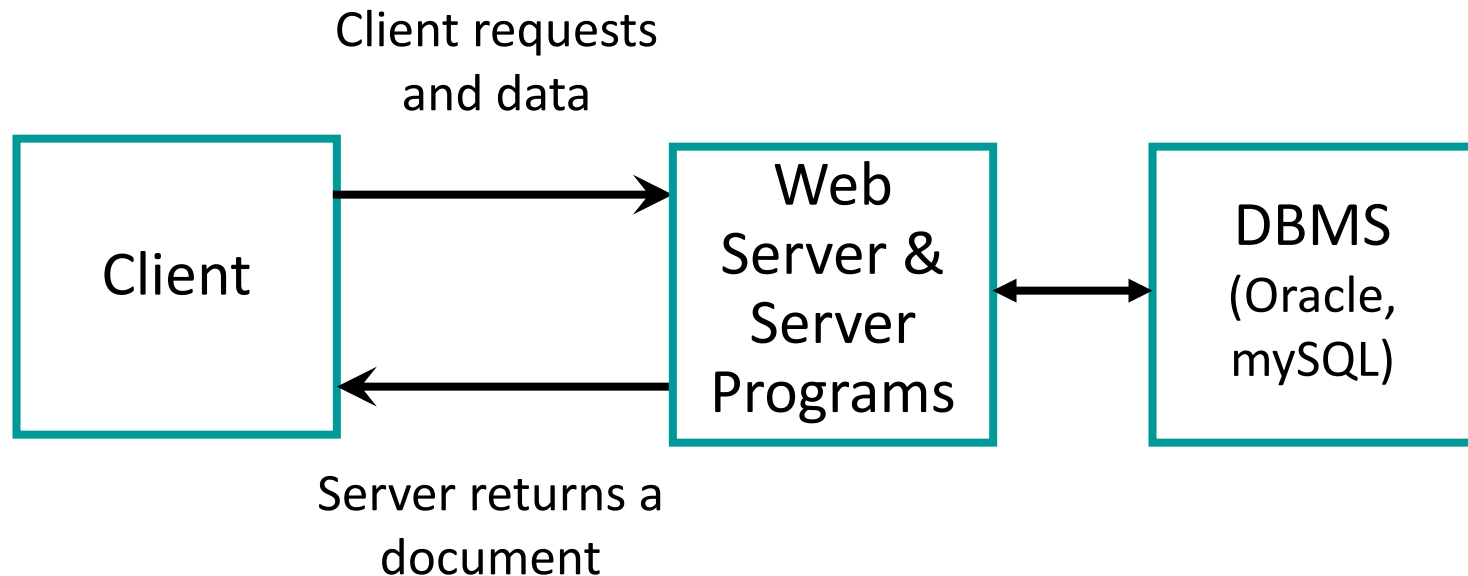
- Pages are static HTML pages with specific URLs (non-parameterized)

Client-Server: Dynamic Pages



- Contents are assembled dynamically
- Request URL can contain parameters or be form data
- Server programs can be written in PHP, JSP, Python, Perl, C, etc. and accept parameters on the URL string

3-Tier Architecture



- Contents are assembled dynamically
- Request URL can contain parameters or be form data
- Server programs can be written in PHP, Python, Perl, C, etc. and accept parameters on the URL string

Server Based Sharing

- Can use the simple client-server model to share data
- I.e. sharing text data makes a basic text chat system for multiple users
- Example on next two pages
 - Structural outline on next page
 - Corresponding screen display (using HTML) on the following page

Client (Browser)

Top frame

Enter \$name,
\$image, \$message

print \$name, \$image

Bottom frame

SH says **Hello...hee hee...**
Davy says **Testing from**
davy . . .

Server Programs

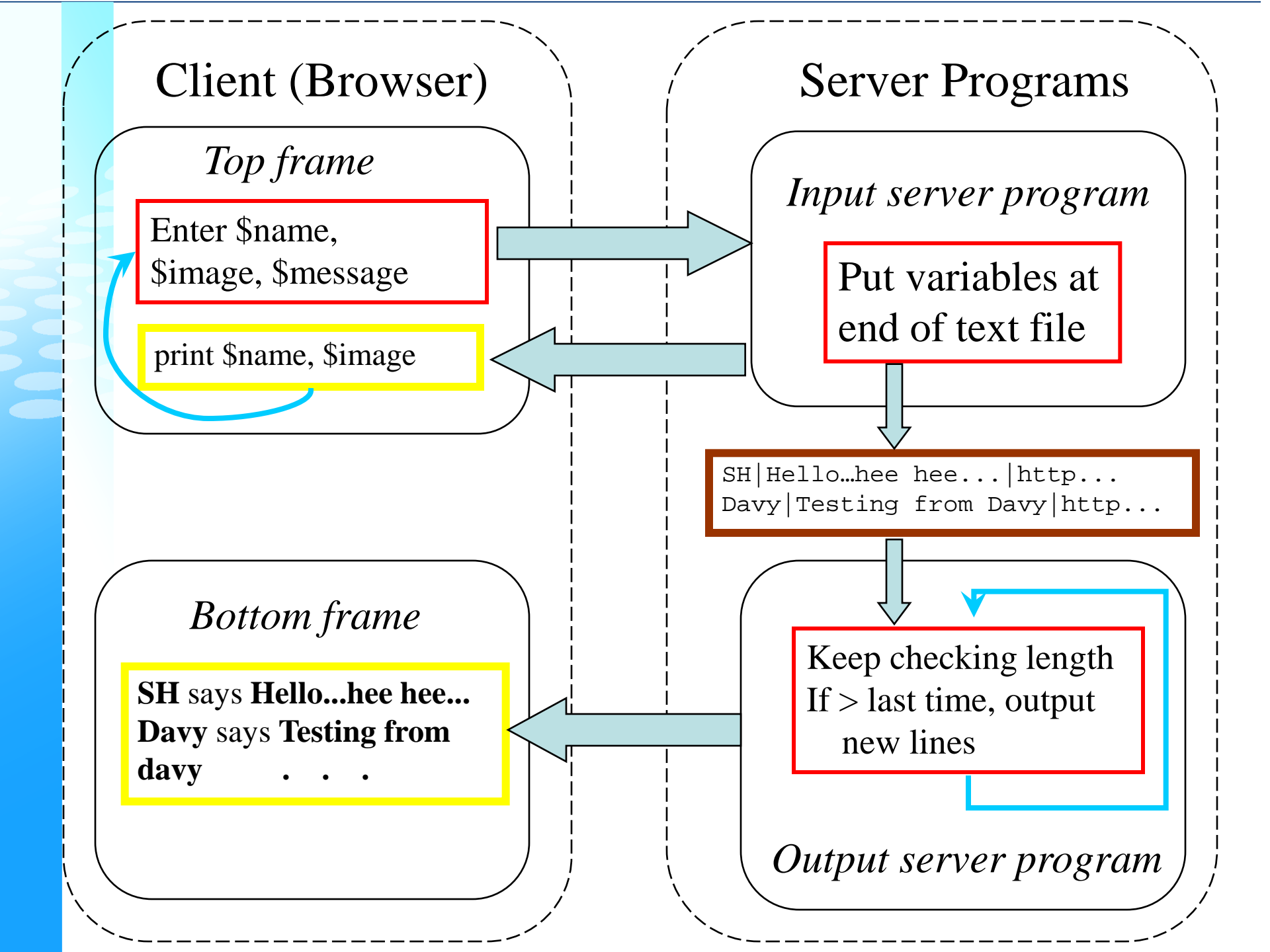
Input server program

Put variables at
end of text file

```
SH|Hello...hee hee...|http...  
Davy|Testing from Davy|http...
```

Keep checking length
If > last time, output
new lines

Output server program

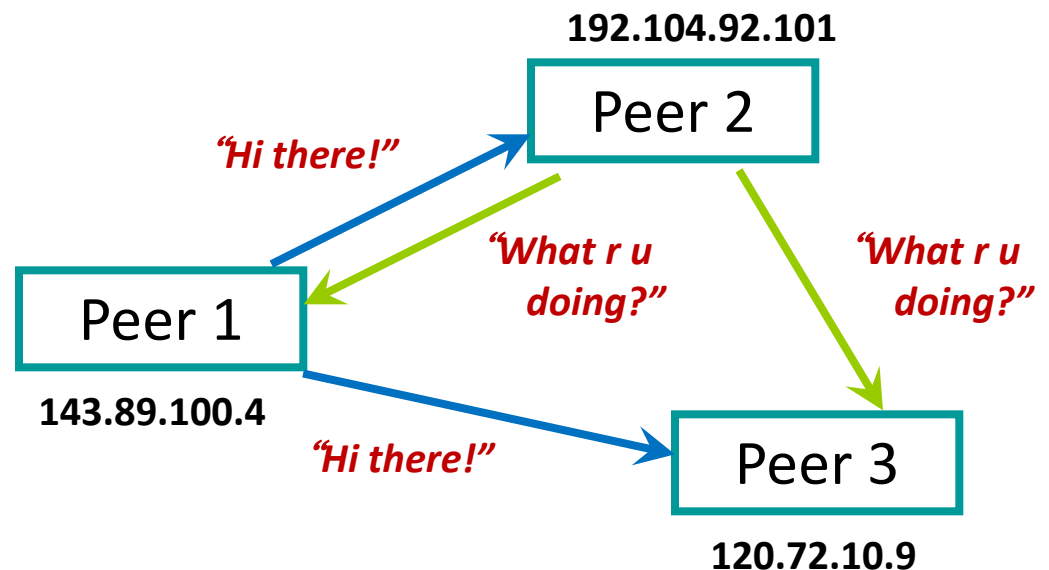


Peer-to-Peer (P2P) Systems

- Peers can communicate with each other without a server
 - A machine can be a client (sending requests) at one time and a server (sending responses) at another time
- P2P is a natural in the real world, e.g., conversation between two persons , and is resilient

P2P chat system

- Each message is broadcast to every connected node
- The same program runs on each node



Peer-to-Peer (P2P) Problems

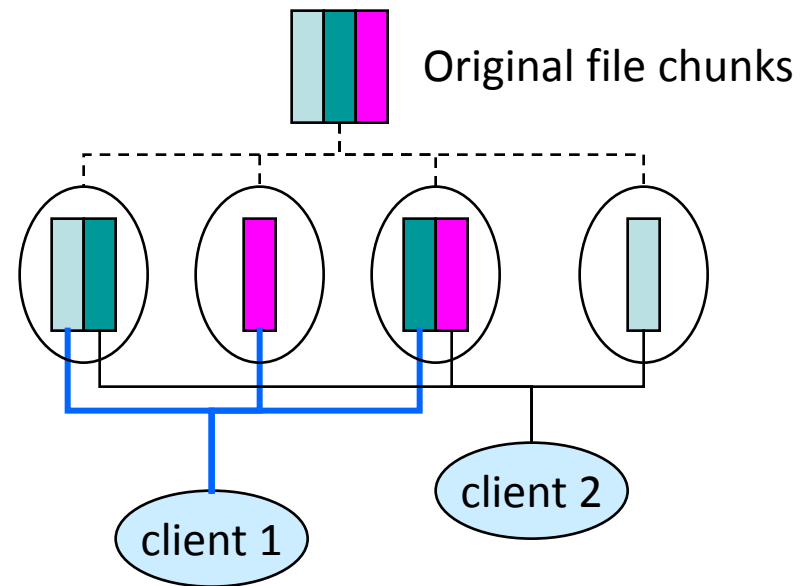
- Peers may come and go; need to know who is online
- Peers need to find each other in order to connect
- Peers need to be willing to help each other (avoid free riders)

File Sharing on Different Models (1)

- Client Server with no replication
 - Slow due to heavy workload on a single server
 - Bandwidth limited by the worst network link between client and server
- Client Server with replication
 - Load on each replica is reduced
 - May still be slow due to limitations on individual server's speed and worst network link to the client
 - Need to maintain replica update
 - Akamai's content delivery network pushes content to “edge servers” one or few hops from the clients

File Sharing on Different Models (2)

- Client-Server with file splitting
 - Split a file into **n** chunks with unique sequence number, distribute (or even replicate) them on multiple server
 - Client download chunks from multiple servers and assemble the chunks into the complete file
 - Speed up **n** times
 - Maximize use of local bandwidth
 - Client needs to do assembling
 - Does it sound like Bit Torrent?
 - How to keep track of the chunks?



Peer-to-Peer File Sharing: Napster

- How to know which peers have which files?
- Napster was popular in late 90's and was out of business in 2001 due to lawsuits
- Napster supports file sharing between clients by keeping **a directory of online peers, their IP addresses and their files**
 - Napster does not copy or keep files but just facilitating peer finding
 - Perceived as legal but eventually got shut down by legal actions

Napster File Share



- Keep track of who is online and the file it can share

Napster v2.0 BETA 7

File Actions Help

Search Hot List Transfer Discover Help

Online	Filename	Filesize	Bitrate	Frequency	Length
azee_659	TEMP\Eve & Nokio - What Ya Want.mp3	4,187,243	128	44100	4:21
itshoss	TEMP\Barenaked Ladies - One Week.mp3	2,665,974	128	44100	2:48
RaztaZ1oo1	TEMP\Jessica Simpson-I Wanna Love You Forever.mp3	4,171,778	128	44100	4:20
weemswd	TEMP\Dr Dre-01-Still D_R_E_ (feat Snoop Dogg) (Radio)-RNS.mp3.mp3	4,408,343	128	44100	4:35
	TEMP\The Thong Song.mp3	4,079,744	128	44100	4:15
	TEMP\Gray - I Try.mp3	3,831,559	128	44100	3:59
	TEMP\Candy Moore - Candy.mp3	3,777,931	128	44100	3:56
Offline	TEMP\wonen g ft_ mack ten - i want it all.mp3	4,294,240	128	44100	4:28
godzilla-meow-m...	TEMP\SMASH MOUTH_09_THEN THE MORNING COMES.mp3	2,956,266	128	44100	3:06
	TEMP\CoCo Lee-01-Do You Want My Love.mp3	3,725,312	128	44100	3:53
	TEMP\Faith Hill - Breathe.mp3	4,019,223	128	44100	4:11
	TEMP\Sonique - It Feels So Good.mp3	5,429,376	128	44100	5:37
	mp3\A_Song_For_Mama.mp3	1,636,320	128	44100	1:45
	mp3\inter.mp3	3,357,352	128	44100	3:30
	mp3\Notorious B.I.G. Featuring Puff Daddy And Mase-Mo Money Mo Problem...	4,119,114	128	44100	4:17

Total Files: 329 Library Size: 1,199 megs Ping: 830

Get Selected Songs Add User to Hot List

Online (davecomp303): Sharing 1 files. Currently 551,573 files (2,278 gigabytes) available in 4,026 libraries.

Napster Text Chat



Napster v2.0 BETA 7

File Actions Help

Home Chat Library Search Hot List Transfer Discover

80's: Welcome to the 80's channel.

<scottcasty> anybody in here?
* unomom (Unknown) [sharing 7 files] has joined.
<scottcasty> anybody see the VH1 awarda?
* purciari (56K) [sharing 10 files] has joined.
<purciari> HELLO AT ALL
<scottcasty> any answers?
<scottcasty> anybody alive?
<purciari> WHERE ARE YOU FROM
<scottcasty> chicago
<purciari> I'M FROM ITALY
<scottcasty> what part?
* chillit (Unknown) [sharing 252 files] has left.
<purciari> IN ROME

User	Files	Speed
bigandstro...	99	Cable
carlosjoes	0	56K
CHEWBA...	19	56K
davecomp...	1	33.6
ecamodeo	59	56K
flolambe	4	56K
flyingcrow1	57	56K
londonmeow	50	56K
oushawn1	263	Cable
PACO-SIM...	0	T3
purciari	10	56K
rojentsch	15	Cable
scottcasty	291	ISDN-1...
sparklingbl...	259	Unknown
unomom	7	Unknown

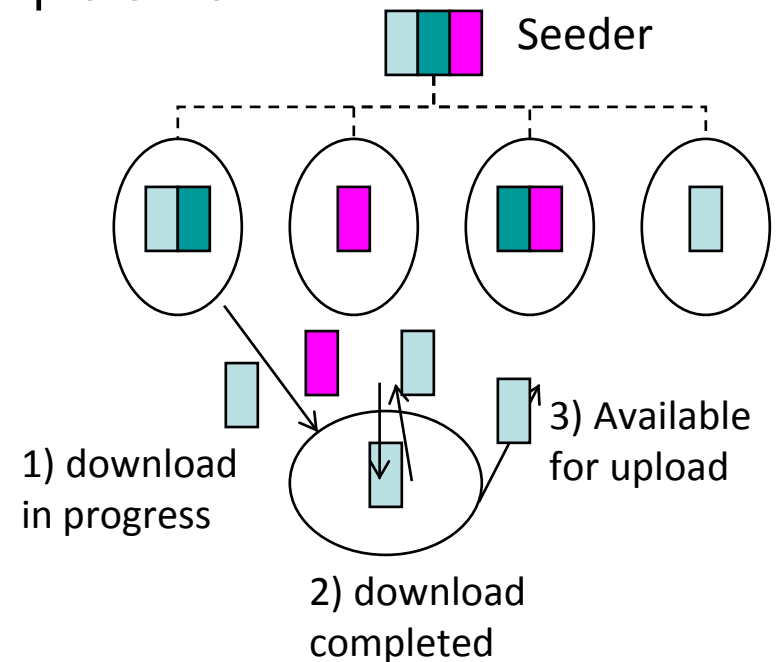
15 users in channel.

Private 80's Chat Rooms Leave Room

Online (davecomp303): Sharing 1 files. Currently 603,050 files (2,477 gigabytes) available in 4,308 libraries.

P2P BitTorrent: Downloading

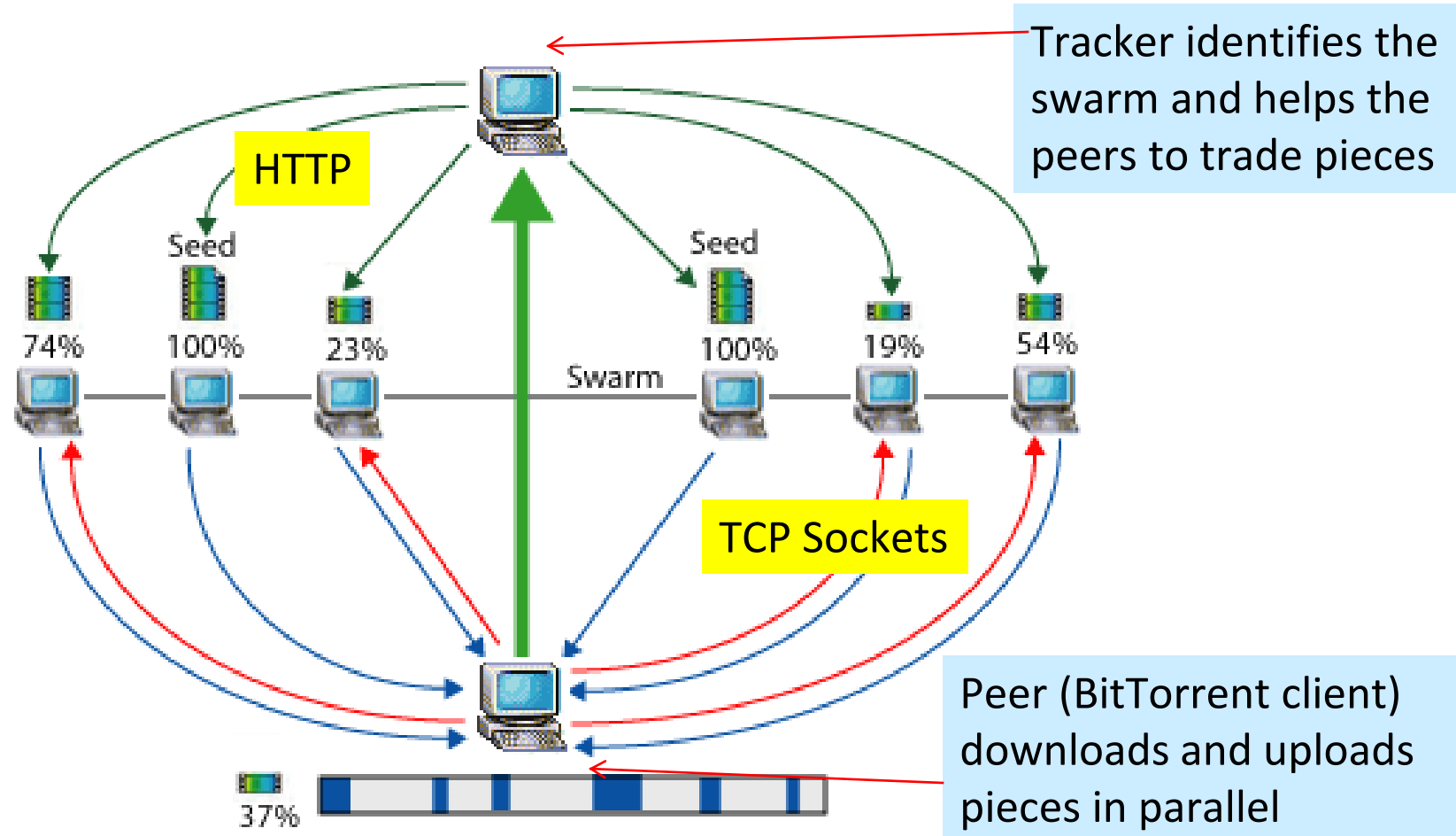
- BitTorrent is a P2P file sharing combining several ideas:
 - Files are chunked, and peers download chunks in parallel and re-assembled them to create the complete file
 - As soon as a peer has downloaded a chunk, it will upload it to other peers: more peers downloading, more peers provide upload, and more peers can download, etc.
 - After a peer completely downloads a file, it becomes a seeder



P2P BitTorrent: Participants (1)

- **Seeder:** Peer that provides the complete file
- **Seed:** Peer with the entire file
- **Leecher:** Peer that is downloading the file
- **Swarm:** Peers participating in distributing the same file
- **Tracker:** Central server keeping track of all peers in the swarm downloading the file
- A peer joins a swarm by asking the tracker for a peer list and connects to those peers

P2P BitTorrent: Participants (2)



P2P BitTorrent: Avoid Free Riding

- A downloader is automatically an uploader: In order to download you have to provide upload
- Reward peers who are willing to upload:
 - A peer uploads to the N neighbors which have high upload speeds; the other peers are “choked”
- Optimistic unchoking: A peer unchokes a random neighbor every 30 seconds to discover new (and better) neighbors
- Gaming: How to provide the minimum upload while maintaining a reasonable download speed?
- You can think of many problems and solutions related to peer discovery and management ...

P2P BitTorrent: Tracker

- The first uploader of a file creates a **.torrent** file containing:
 - Metadata of the file (no. of pieces, piece length, SHA-1 hash of piece)
 - URL of **tracker** that coordinates file distribution
- **Trackers** are infrastructural nodes that keep track of peers and their status in the torrent (downloading peers)
- A new peer wanting to download a file:
 - Get the torrent file (search Google for torrent-hosting sites)
 - Connect to tracker, which returns some random peers in the swarm
 - Download and update tracker with status (e.g., completed pieces)
- No central directories of peers and files to avoid legal issues
- Even centralized trackers can be avoided by selecting peers as trackers using hashing

Problems and Solutions

- What is the best order for a peer to download the pieces?
 - If every peers downloads popular pieces first, it may end up that all peers contain the popular pieces but none contain the missing ones
 - Randomly select the pieces
 - Rarest first: Pieces with fewest available copies (get them before they are gone) thus making rare piece more available
 - Random first: Request a random piece first, then use rarest-first so as to obtain complete first piece earlier and participate in upload earlier
- End game: Request last few pieces from all peers to complete download ASAP (when a piece is completed, do not forget to cancel outstanding requests on the same piece)