

# **CSE2029: Data Communication & Computer Networks**

## **Lecture-8: DNS and P2P Applications**

**Faculty: Dr. Sandeep Kumar**

# Outline

- ❖ *DNS:*
  - ❖ *DNS — The Internet's Directory Service*
  - ❖ *Load distribution through DNS*
  - ❖ *Hierarchy of DNS Database*
  - ❖ *Interaction of the various DNS servers*
  - ❖ *DNS Caching*
- ❖ *Peer-to-Peer File Distribution*
  - ❖ *Peer-to-Peer File Distribution with BitTorrent*

## DNS — The Internet's Directory Service

- Internet hosts can be identified in many ways:
- One identifier for a host is its **hostname**. ✓ Hostnames—such as www.facebook.com, www.google.com, gaia.cs.umass.edu—are mnemonic and are therefore appreciated by humans.
- However, hostnames provide little, if any, information about the **location** within the Internet of the host. For example: A hostname such as www.eurecom.fr, which ends with the country code .fr, tells us that the host is probably in France, but doesn't say much more. Furthermore, because hostnames can consist of variable-length alphanumeric characters, they would be difficult to process by routers.
- For these reasons, hosts are also identified by so-called **IP addresses**. ✓ An IP address is hierarchical because as we scan the address from left to right, we obtain more and more specific information about where the host is located in the Internet.

## DNS — The Internet's Directory Service

- We have just seen that there are two ways to identify a host—by a hostname and by an IP address. People prefer the more mnemonic hostname identifier, while routers prefer fixed-length, hierarchically structured IP addresses.
- In order to reconcile these preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the Internet's domain name system (DNS).
- The DNS is (1) a distributed database implemented in a hierarchy of DNS servers, and (2) an application-layer protocol that allows hosts to query the distributed database.
- The DNS servers are often UNIX machines running the Berkeley Internet Name Domain (BIND) software. The DNS protocol runs over UDP and uses port 53.
- DNS is commonly employed by other application-layer protocols, including HTTP and SMTP, to translate user-supplied hostnames to IP addresses.

## DNS — The Internet's Directory Service

- As an example, consider what happens when a browser running on some user's host, requests the URL [www.someschool.edu/index.html](http://www.someschool.edu/index.html). In order for the user's host to be able to send an HTTP request message to the Web server www.someschool.edu, the user's host must first obtain the IP address of www.someschool.edu. This is done as follows:
  - The same user machine runs the client side of the DNS application.
  - The browser extracts the hostname, www.someschool.edu, from the URL and passes the hostname to the client side of the DNS application.
  - The DNS client sends a query containing the hostname to a DNS server.
  - The DNS client eventually receives a reply, which includes the IP address for the hostname.  
[All DNS query and reply messages are sent within UDP datagrams to port 53.]
  - Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.

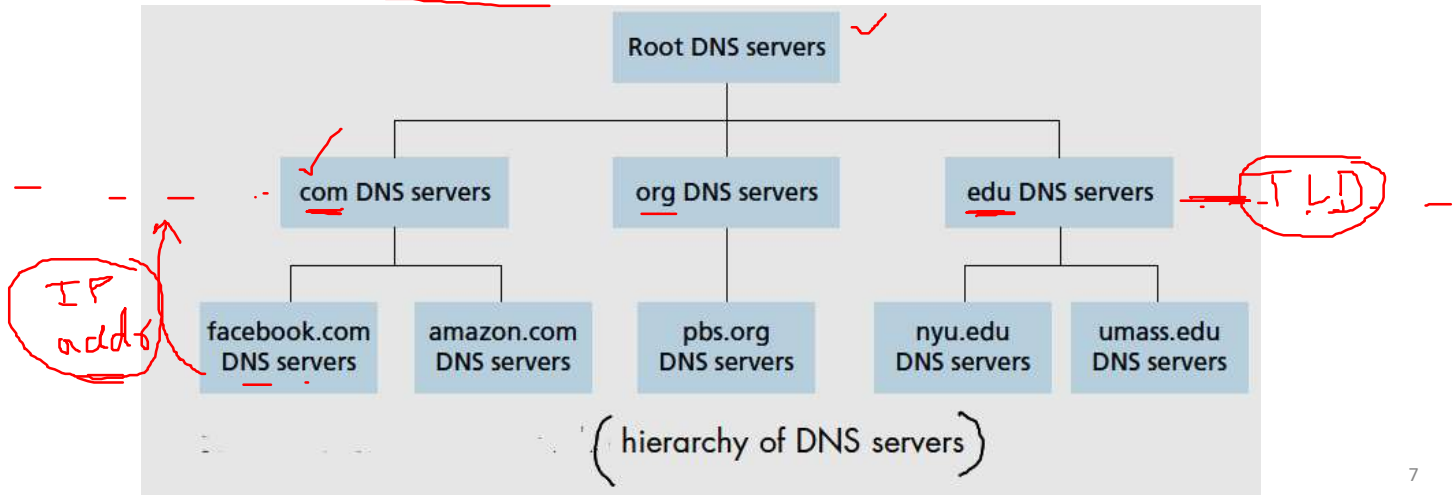
## Load distribution through DNS

DNS provides a **Load distribution services** in addition to translating hostnames to IP addresses as follows:

- Busy sites, such as cnn.com are replicated over multiple servers, with each server running on a different end system and each having a different IP address. For replicated Web servers, a set of IP addresses is thus associated with one hostname. The DNS database contains this set of IP addresses.
- When clients make a DNS query for a name mapped to a set of addresses, the server responds with the entire set of IP addresses, but rotates the ordering of the addresses within each reply. Because a client typically sends its HTTP request message to the IP address that is listed first in the set, DNS rotation distributes the traffic among the replicated servers.

## Hierarchy of DNS Database

- DNS uses a large number of servers, organized in a hierarchical fashion and distributed around the world.
- No single DNS server has all of the mappings for all of the hosts in the Internet. Instead, the mappings are distributed across the DNS servers.
- To a first approximation, there are three classes of DNS servers— root DNS servers, top-level domain (TLD) DNS servers, and authoritative DNS servers—organized in a hierarchy as shown in Figure below:



## Hierarchy of DNS Database

let's first take a closer look at these three classes of DNS servers:

- Root DNS servers: There are more than 1000 root servers instances scattered all over the world. These root servers are copies of 13 different root servers, managed by 12 different organizations, and coordinated through the Internet Assigned Numbers Authority [IANA]. Root name servers provide the IP addresses of the TLD servers.
- Top-level domain (TLD) servers: For each of the top-level domains—top-level domains such as com, org, net, edu, and gov, and all of the country top-level domains such as uk, fr, ca, and jp—there is TLD server (or server cluster). The network infrastructure supporting a TLD can be large and complex. TLD servers provide the IP addresses for authoritative DNS servers.



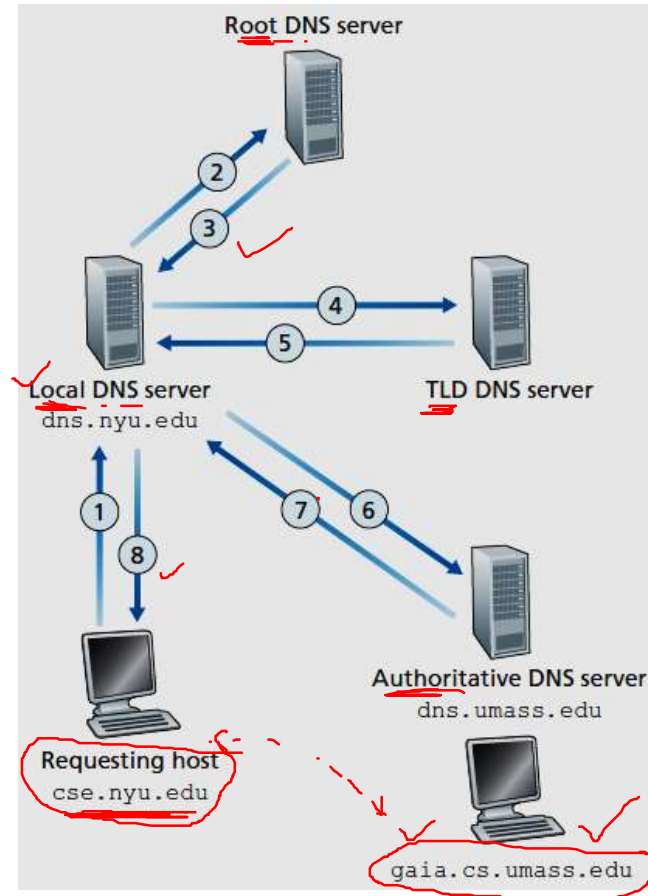
## Hierarchy of DNS Database

- Authoritative DNS servers: Every organization with publicly accessible hosts (such as Web servers and mail servers) on the Internet must provide publicly accessible DNS records that map the names of those hosts to IP addresses. An organization's authoritative DNS server houses these DNS records. An organization can choose to implement its own authoritative DNS server to hold these records; alternatively, the organization can pay to have these records stored in an authoritative DNS server of some service provider. Most universities and large companies implement and maintain their own primary and secondary (backup) authoritative DNS server.
- Local DNS server: There is another important type of DNS server called the local DNS server. A local DNS server does not strictly belong to the hierarchy of servers but is considered as central to the DNS architecture. Each ISP—such as a residential ISP or an institutional ISP—has a local DNS server (also called a default name server). When a host makes a DNS query, the query is sent to the local DNS server, which acts a proxy, forwarding the query into the DNS server hierarchy.

## Interaction of the various DNS servers

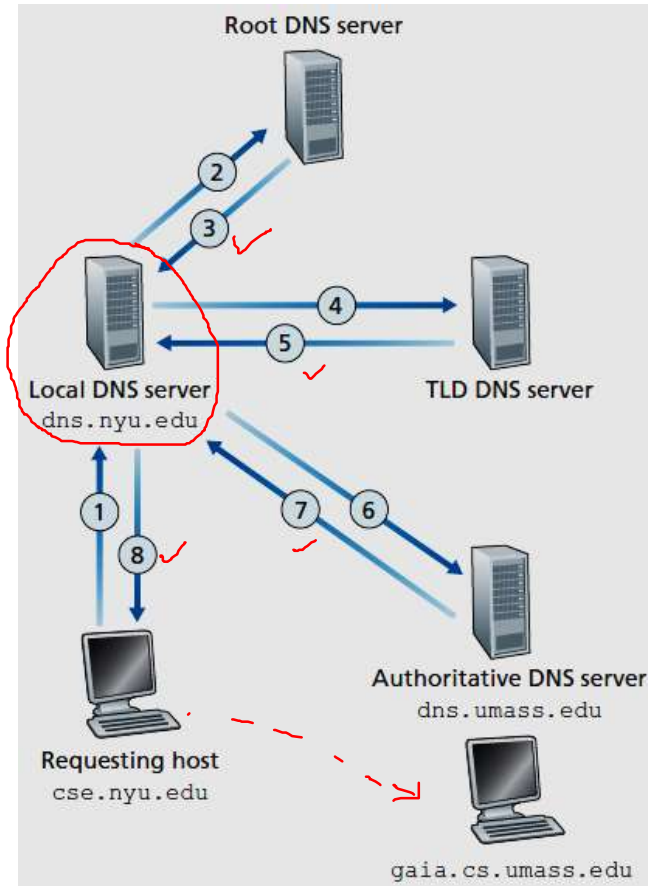
### EXAMPLE:

- Suppose the host cse.nyu.edu desires the IP address of gaia.cs.umass.edu.
- Also suppose that NYU's local DNS server for *cse.nyu.edu* is called **dns.nyu.edu** and that an authoritative DNS server for *gaia.cs.umass.edu* is called **dns.umass.edu**.
- First, the host *cse.nyu.edu* sends a DNS query message to its local DNS server, *dns.nyu.edu*. The query message contains the hostname to be translated, namely, *gaia.cs.umass.edu*.



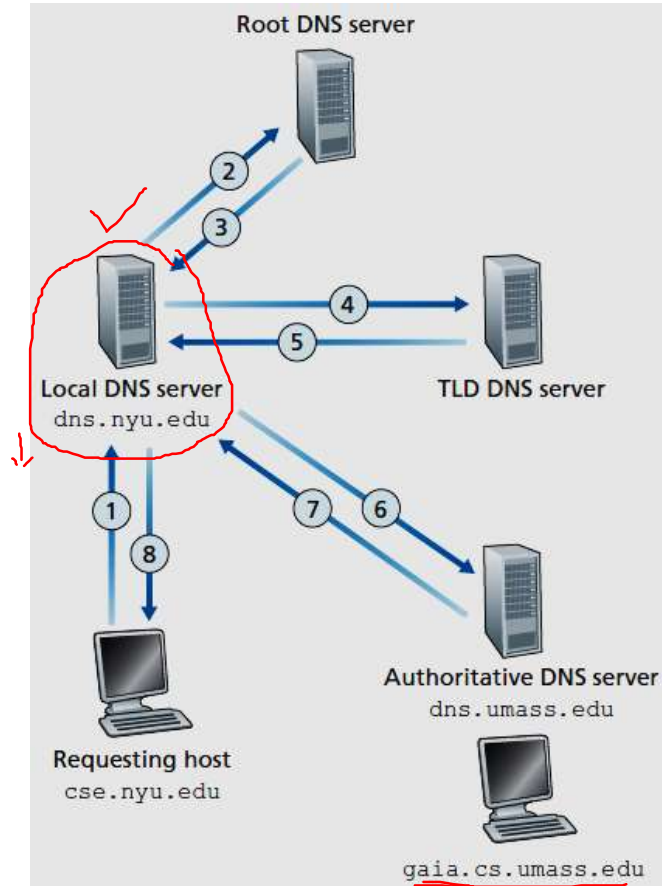
## Interaction of the various DNS servers

- The local DNS server forwards the query message to a root DNS server. The root DNS server takes note of the **edu** suffix and returns to the local DNS server a list of IP addresses for TLD servers responsible for **edu**.
- The local DNS server then resends the query message to one of these TLD servers. The TLD server takes note of the **umass.edu** suffix and responds with the IP address of the authoritative DNS server for the University of Massachusetts, namely, `dns.umass.edu`.
- Finally, the local DNS server resends the query message directly to `dns.umass.edu`, which responds with the IP address of `gaia.cs.umass.edu`.



## Interaction of the various DNS servers


- Note that in this example, in order to obtain the mapping for one hostname, eight DNS messages were sent: four query messages and four reply messages!
- The implementation of DNS caching reduces this query traffic.*



## DNS Caching

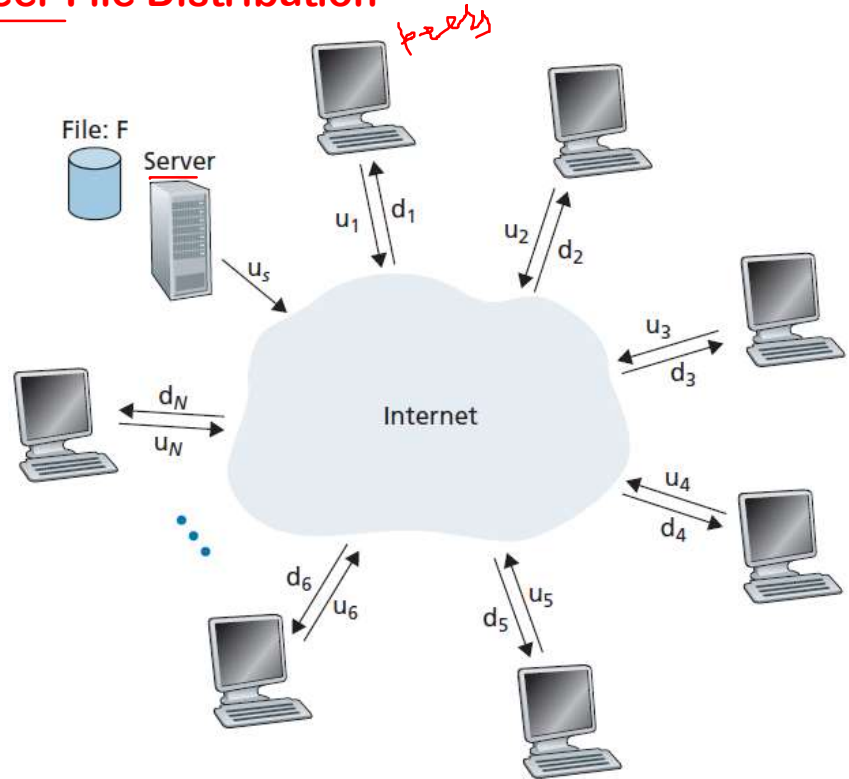
- Our discussion thus far has ignored DNS caching, a critically important feature of the DNS system. In truth, DNS extensively exploits **DNS caching** in order to improve the delay performance and to reduce the number of DNS messages ricocheting around the Internet.
- The idea behind DNS caching is very simple. In a query chain, when a **DNS server (say Local DNS Server)** receives a DNS reply containing a mapping from a hostname to an IP address, it can cache the mapping in its local memory.
- If a hostname/IP address pair is **cached in a DNS server** and another query arrives to the DNS server for the same hostname, the DNS server can provide the desired IP address.
- Because hosts and mappings between hostnames and IP addresses are by no means permanent, DNS servers discard cached information after a period of time (often set to two days).

## Peer-to-Peer File Distribution

- Most of the applications including the Web, e-mail, and DNS—all employ client-server architectures with significant reliance on the **always-on infrastructure servers**.
- Recall that with a P2P architecture, there is no (or minimum) reliance on always-on infrastructure servers. Instead, pairs of intermittently connected hosts, called peers, communicate directly with each other. These peers are not owned by a service provider, but are instead PCs, laptops, and smartphones controlled by users.
- Consider a very natural P2P application, namely, distributing a large file from a single server to a large number of hosts (called peers). The file might be a new version of the Linux operating system, a software patch for an existing operating system or an MPEG video file. 

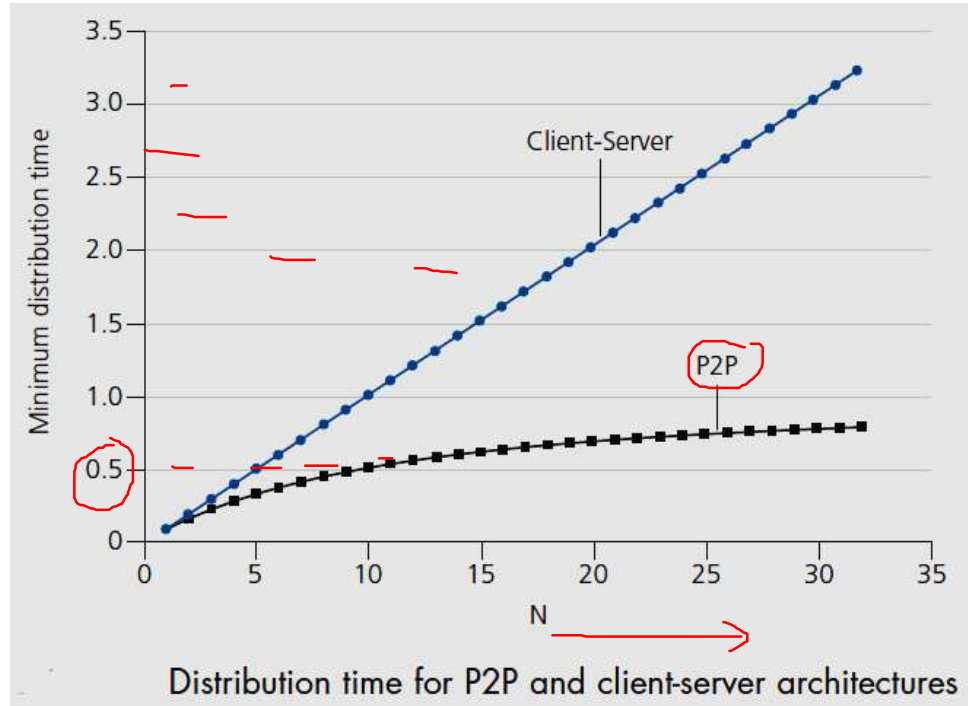
## Peer-to-Peer File Distribution

- In client-server file distribution, the server must send a copy of the file to each of the peers—placing an enormous burden on the server and consuming a large amount of server bandwidth.
- In P2P file distribution, each peer can redistribute any portion of the file it has received to any other peers, thereby assisting the server in the distribution process.



## Peer-to-Peer File Distribution

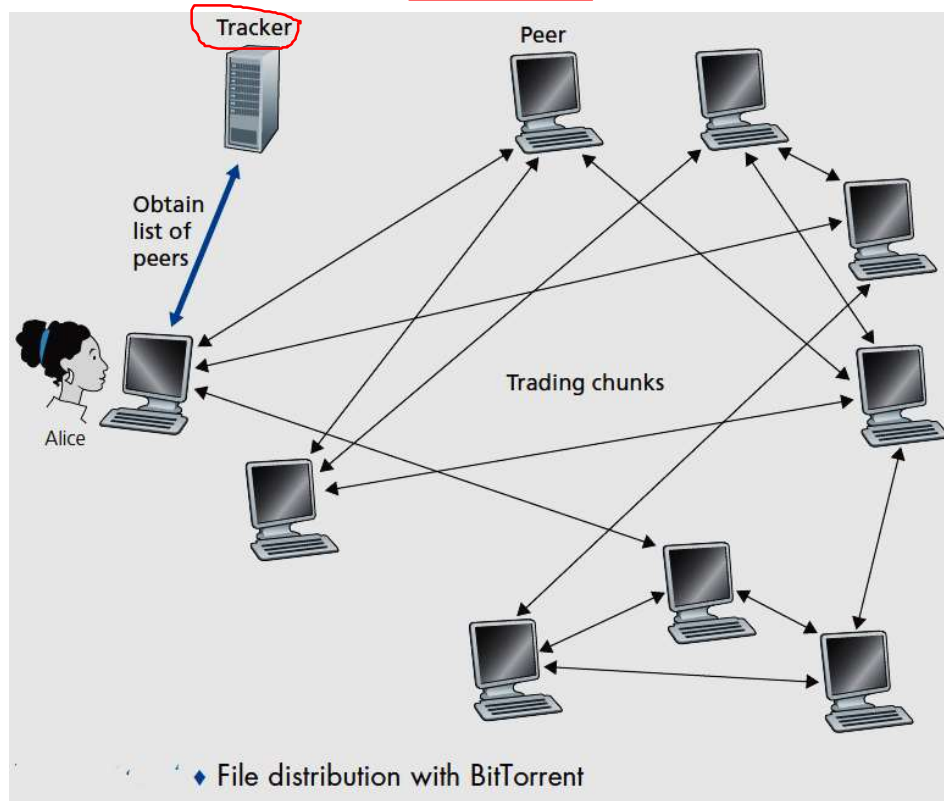
- As of 2020, the most popular P2P file distribution protocol is BitTorrent
- Originally developed by Bram Cohen, there are now many different independent BitTorrent clients conforming to the BitTorrent protocol, just as there are a number of Web browser clients that conform to the HTTP protocol.





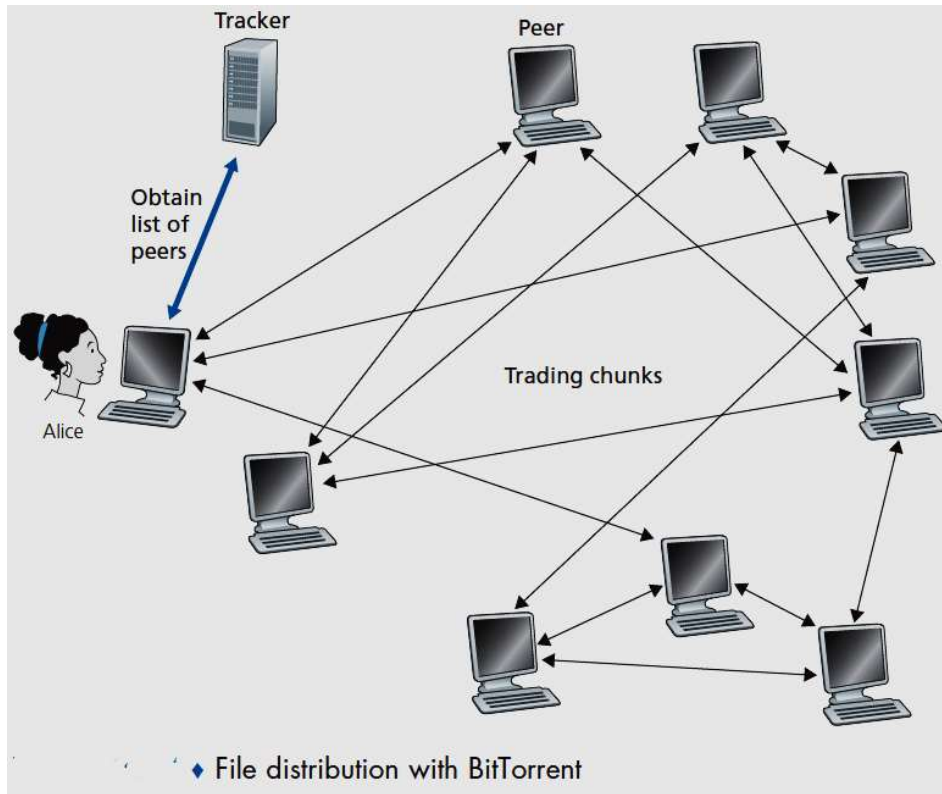
## Peer-to-Peer File Distribution with BitTorrent

- The collection of all peers participating in the distribution of a particular file is called a torrent.
- Peers in a torrent download equal-size chunks of the file from one another, with a typical chunk size of 256 KBytes.
- When a peer first joins a torrent, it has no chunks. Over time it accumulates more and more chunks. While it downloads chunks it also uploads chunks to other peers.
- Once a peer has acquired the entire file, it may (selfishly) leave the torrent, or (altruistically) remain in the torrent and continue to upload chunks to other peers.



## Peer-to-Peer File Distribution with BitTorrent

- Any peer may leave the torrent at any time with only a subset of chunks, and later rejoin the torrent.
- Each torrent has an infrastructure node called a tracker. When a peer joins a torrent, it registers itself with the tracker and periodically informs the tracker that it is still in the torrent.
- In this manner, the tracker keeps track of the peers that are participating in the torrent. A given torrent may have fewer than ten or more than a thousand peers participating at any instant of time.



***Thank you.***