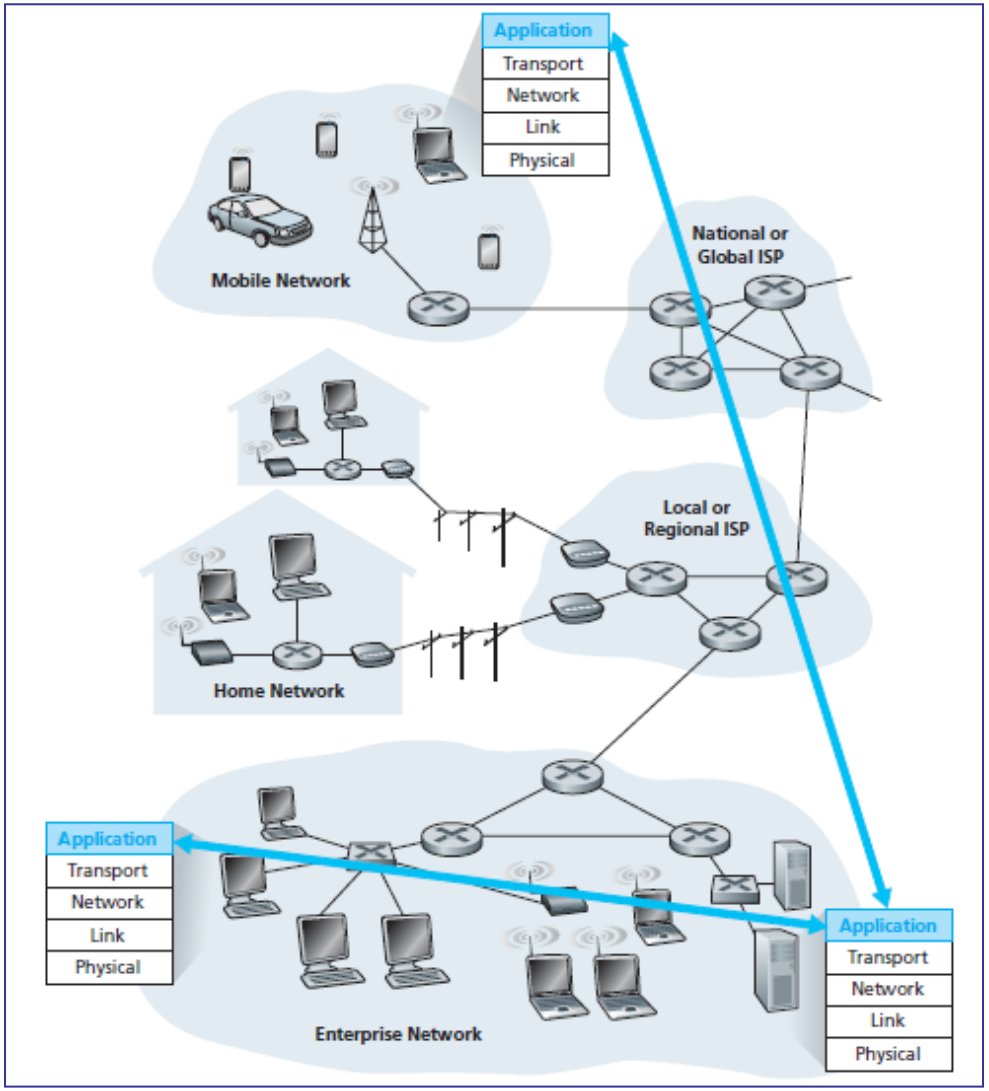
Application Layer

- **Overview:**
  - Applications have been the **driving force behind the Internet's success.**
  - In this chapter, we study the **conceptual** and **implementation** aspects of **network applications**.
  - **Key application-layer** concepts, including **network services** required by applications, **clients and servers**, **processes** and **transport-layer interfaces.**
  - **Web, e-mail, DNS** and **peer-to-peer (P2P) file** distribution.
  - **Network application development** over both **TCP** and **UDP**.
  - Study of **socket API** and walk through some simple client-server applications
  - **Network application development** is writing programs that run on different end systems and communicate with each other over the network.
  - Ex. **Web application, P2P file-sharing system**

Principles of Network Applications

Communication for a network application takes place between end systems at the application layer



Principles of Network Applications

Addressing Processes:

- To identify the receiving process, **two pieces of information** need to be specified:
    1. **The address of the host**
    2. **An identifier that specifies the receiving process in the destination host.**
- The host is identified by its **IP address, 32-bit quantity** that we can think of as uniquely identifying the host
- The sending process must also identify the receiving process, A **destination port number** serves this purpose.
- Popular applications have been assigned **specific port numbers**. For example, a Web server is identified by port number 80.

Principles of Network Applications

Application-Layer Protocols

- Defines how an **application's processes, running on different end systems, pass messages to each other.**
- It defines;
  - *The **Types** of messages exchanged*
  - *The **Syntax** of the various message types*
  - *The **Semantics** of the fields*
  - ***Rules** for determining **when** and **how** a process sends messages and responds to messages*
- **An application-layer protocol is only one piece of a network application. Ex. Web Application, E-mail Application.**
- **Few important Network Applications;** the Web, File transfer, Electronic mail, Directory service

The Web and HTTP

- **World Wide Web** dramatically changed, and continues to change, how people interact inside and outside their work environments. Web operates **on demand.**

Overview of HTTP

- The **Hyper Text Transfer Protocol (HTTP)**, the Web's application-layer protocol, is at the heart of the Web.
- HTTP is implemented in two programs: **Client program** and **Server program**.
- A **Web page** (also called a **document**) consists of objects. An **Object** is simply a file.
- **Web pages** consist of a **base HTML file** and **several referenced objects.**
- Each **URL** has two components: **the hostname of the server** that houses the object and **the object's path name.**

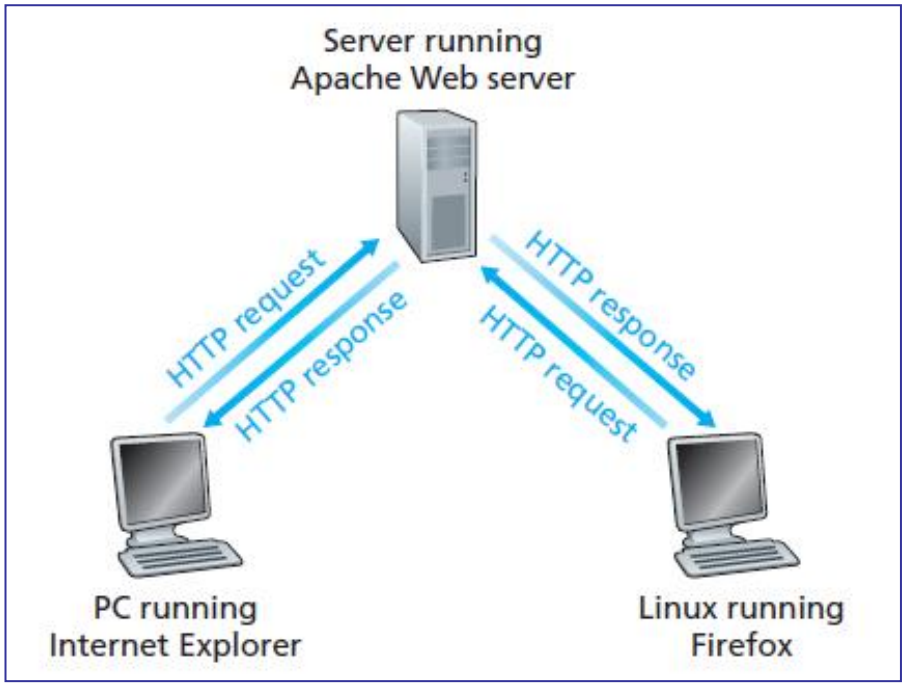
http://www.someSchool.edu/someDepartment/picture.gif

The Web and HTTP

Overview of HTTP

- **Web browsers** implement the client side of HTTP, **Web Servers** implements the server side of HTTP.
- **HTTP** defines how Web clients request Web pages from Web servers and how servers transfer Web pages to clients.

HTTP request-response behaviour



The Web and HTTP

Overview of HTTP

- HTTP uses **TCP** as its underlying transport protocol.
- The browser and the server processes access TCP through their **socket interfaces.**
- TCP provides a **reliable data transfer** service to HTTP.
- Server sends requested files to clients without storing any **state information** about the client. HTTP is said to be a **stateless protocol.**

The Web and HTTP

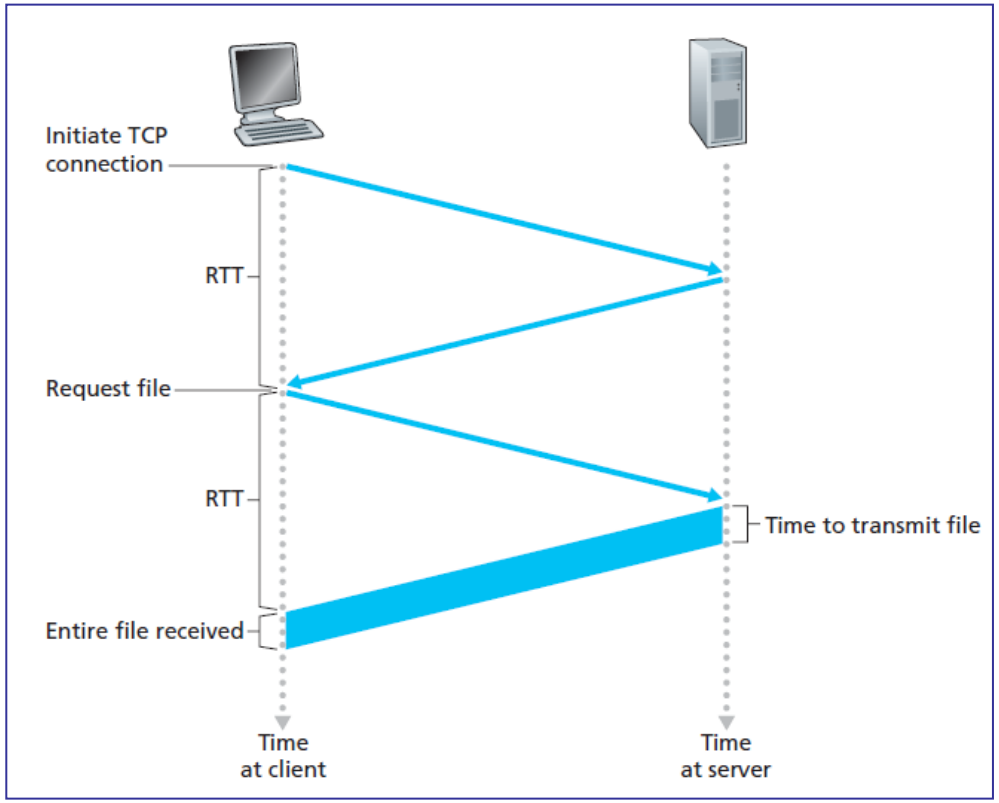
Non-Persistent and Persistent Connections

- In many Internet applications, the client and server communicate for an extended period of time.
- **Non-persistent connections** sends each request/response pair over a separate TCP connection. **Persistent connections** sends requests and their corresponding responses be over the same TCP connection. **HTTP** uses persistent connections in its default mode.
- **HTTP with Non-Persistent Connections:**
  - **Example -** The page consists of a base HTML file and 10 JPEG images.
  - URL: *http://www.someSchool.edu/someDepartment/home.index*
  - TCP connection transports exactly one request message and one response message. Thus, in this example, when a user requests the Web page, 11 TCP connections are generated.

The Web and HTTP

Non-Persistent and Persistent Connections

Back-of-the-envelope calculation for the time needed to request and receive an HTML file



Total Response Time = Two RTTs + The Transmission Time at the server of the HTML file

The Web and HTTP

Non-Persistent and Persistent Connections

- **HTTP with Persistent Connections:**
  - Non-persistent connections place a **significant burden on the Web server**, which may be serving requests from hundreds of different clients simultaneously.
  - Each object suffers a delivery **delay of two RTTs.**
  - With **persistent connections**, the server leaves the TCP connection open after sending a response.
  - Subsequent requests and responses between the same client and server can be sent over the same connection.

The Web and HTTP

HTTP Message Format

There are two types of HTTP messages, request messages and response messages.

HTTP Request Message:

GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr

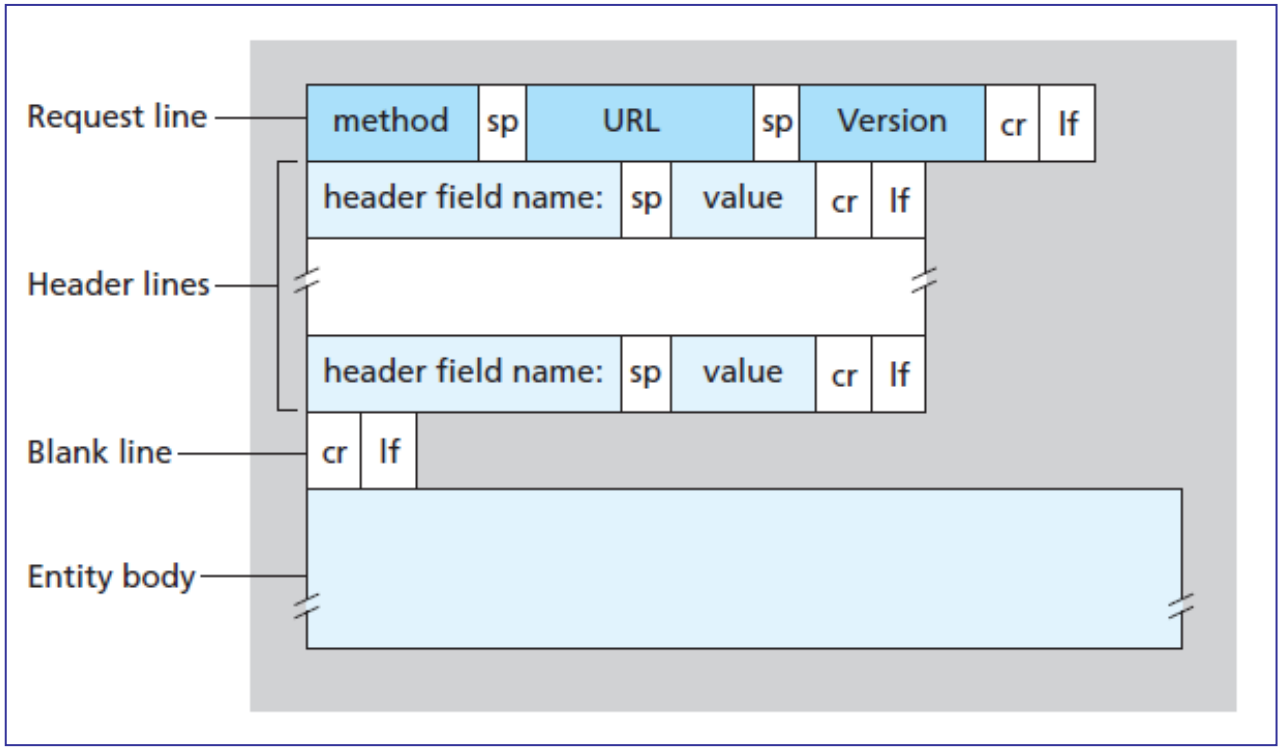
- **Request line** and **header line.**
- The request line has three fields: the **method field**, the **URL field**, and the **HTTP version field**.

The Web and HTTP

HTTP Message Format

There are two types of HTTP messages, request messages and response messages.

HTTP Request Message:



General format of an HTTP Request Message

The Web and HTTP

HTTP Message Format

There are two types of HTTP messages, request messages and response messages.

HTTP Request Message:

Different Methods

Get: Used when the browser requests an object, with the requested object identified in the URL field

Post: Used when the user fills out a form

Head: Used for debugging

Put: Used by applications that need to upload objects to Web servers.

Delete: Used to delete an object on a Web server.

The Web and HTTP

HTTP Message Format

HTTP Response Message:

HTTP/1.1 200 OK

Connection: close

Date: Tue, 09 Aug 2011 15:44:04 GMT

Server: Apache/2.2.3 (CentOS)

Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT

Content-Length: 6821

Content-Type: text/html

(data data data data data ...)

- It has three sections: an **initial status line**, **six header lines**, and then the **entity body.**
- The **status line** has three fields: the **protocol version field**, a **status code**, and a **corresponding status message**.

The Web and HTTP

HTTP Message Format

HTTP Response Message:

Header lines:

Connection: close the TCP connection after sending the message.

Date: indicates the time and date when the HTTP response was created.

Server: indicates that the message was generated by an Apache Web server;

User-agent: header line in the HTTP request message.

Last-Modified: indicates the time and date when the object was created or last modified.

Content-Length: Number of bytes in the object being sent.

Content-Type: object in the entity body is HTML text

The Web and HTTP

HTTP Message Format

HTTP Response Message:

Some common status codes:

200 (OK): Request succeeded and the information is returned in the response.

301 (Moved Permanently): Requested object has been permanently moved

400 (Bad Request): This is a generic error code indicating that the request could not be understood by the server.

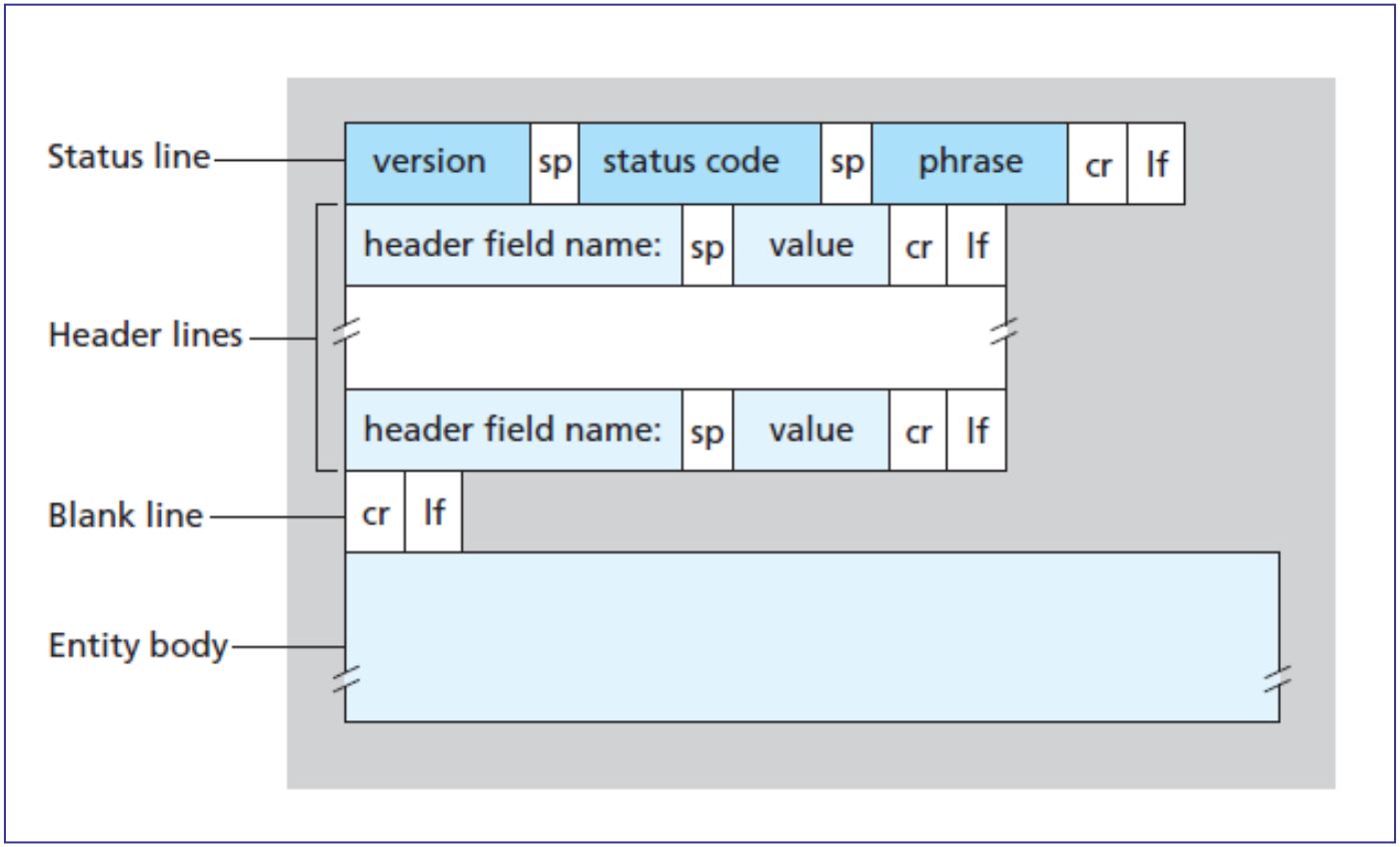
404 (Not Found): The requested document does not exist on this server.

505 (HTTP Version Not Supported): The requested HTTP protocol version is not supported by the server.

The Web and HTTP

HTTP Message Format

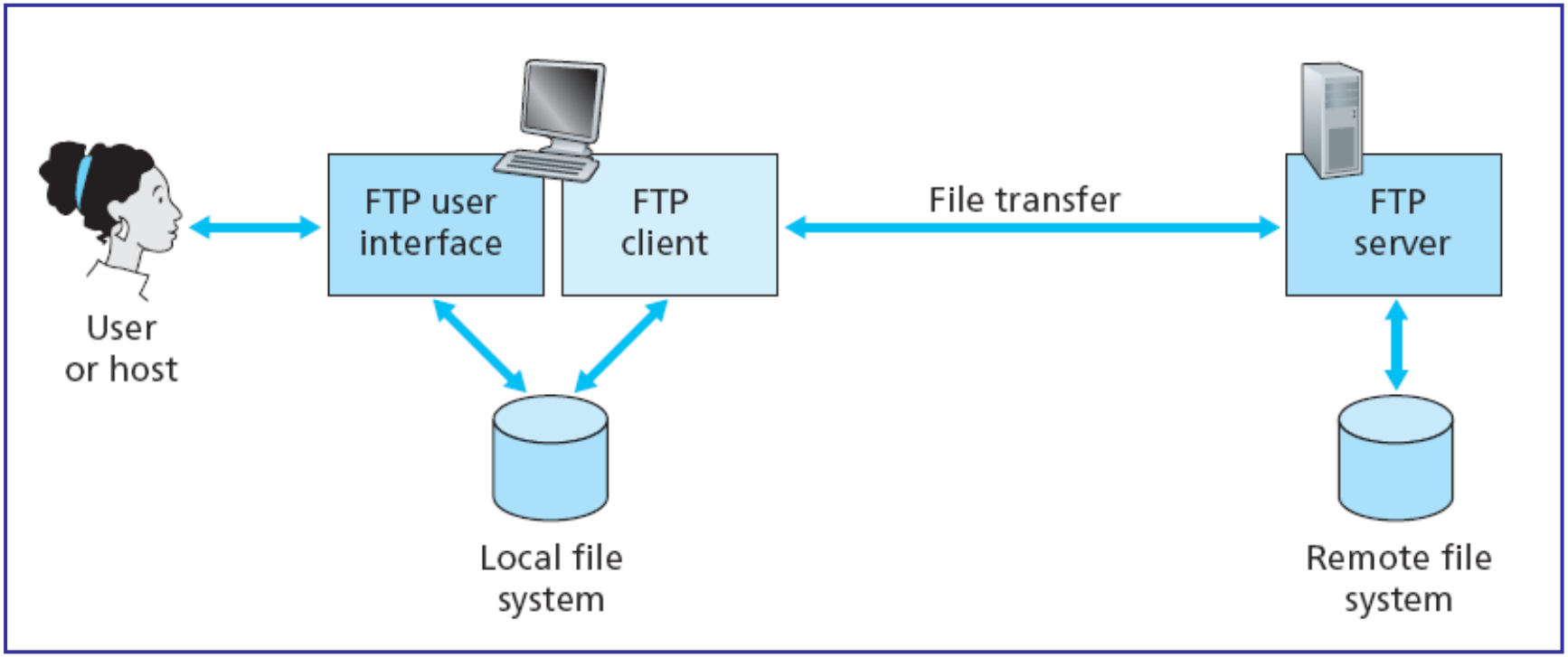
HTTP Response Message:



General format of an HTTP Response Message

File Transfer: FTP

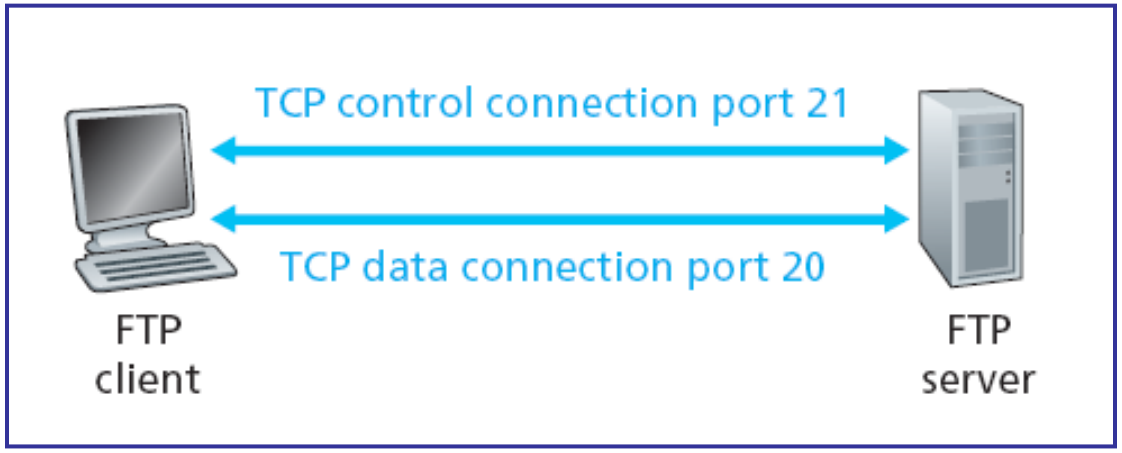
- In a typical **FTP session**, the user is sitting in front of one host (the local host) and wants to **transfer files to or from a remote host.**



FTP moves files between local and remote file systems

File Transfer: FTP

- FTP uses **two parallel TCP connections** to transfer a file, a **control connection** and a **data connection**.
- **FTP** is said to send its control information **out-of-band. HTTP** is said to send its control information **in-band.**
- Data connections are **non-persistent.**
- Throughout a session, the FTP server must maintain **state** about the user.



Control and Data connections

File Transfer: FTP

FTP Commands

- **USER username:** Used to send the user identification to the server.
- **PASS password:** Used to send the user password to the server.
- **LIST:** Used to ask the server to send back a list of all the files in the current remote directory.
- **RETR filename:** Used to retrieve a file from the current directory of the remote host.
- **STOR filename:** Used to store a file into the current directory of the remote host.

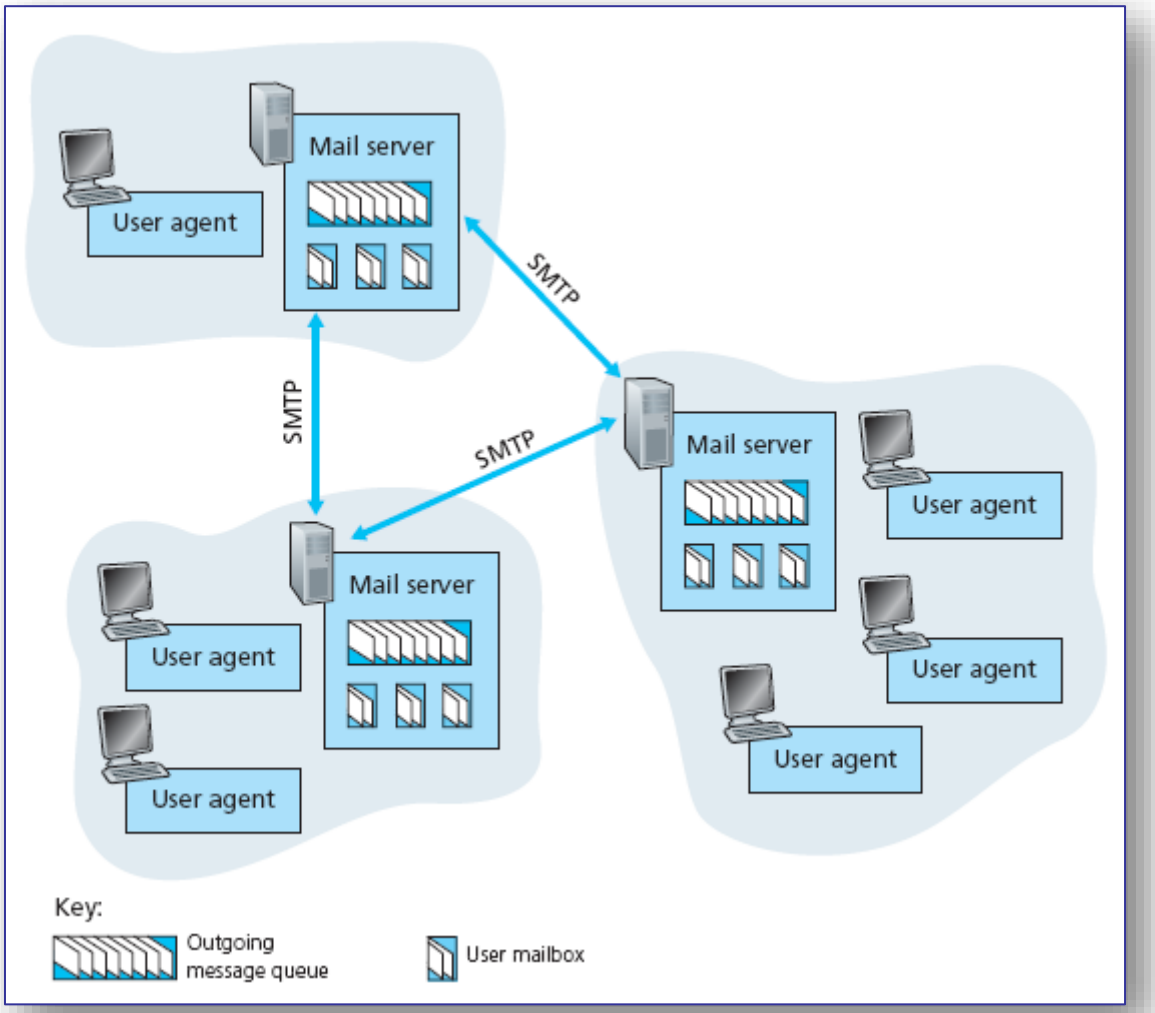
FTP Replies

- **331** Username OK, password required
- **125** Data connection already open; transfer starting
- **425** Can’t open data connection
- **452** Error writing file

Electronic Mail in the Internet

- Internet's **most important and utilized applications.**
- E-mail is an **asynchronous** communication medium.
- three major components: **user agents**, **mail servers**, and the **Simple Mail Transfer Protocol (SMTP).**
- **User agents** allow users to read, reply to, forward, save, and compose messages.
- **User agent** sends the message to sender's mail server, where the message is placed in the mail server's **outgoing message queue.**
- **Receiver's mailbox** manages and maintains the messages that have been sent to receiver.
- **Sender's mail server** must also deal with **failures** in Receiver's mail server.
- SMTP uses the **reliable data transfer** service of **TCP** to transfer mail from the sender's mail server to the recipient's mail server.
- SMTP has two sides: **a client side** and **a server side.**

Electronic Mail in the Internet

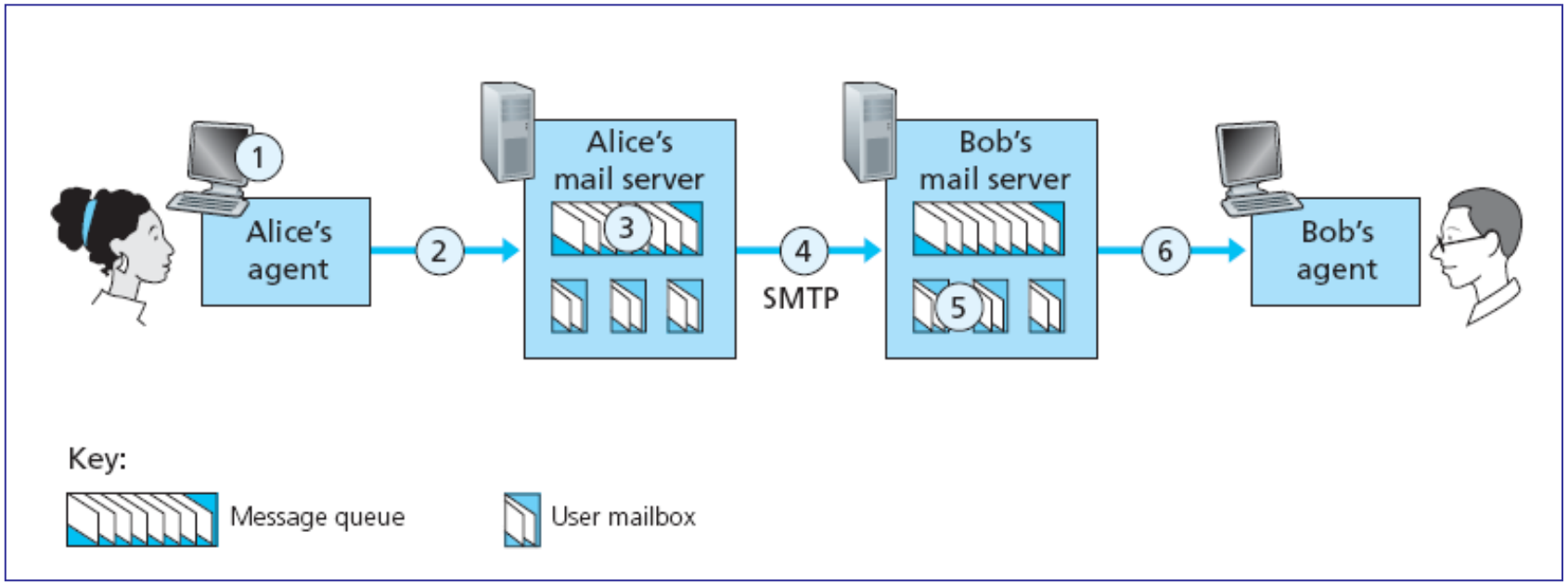


A high-level view of the Internet e-mail system

Electronic Mail in the Internet

SMTP

- Transfers messages from **senders' mail servers** to the **recipients' mail servers.**
- SMTP does not normally use **intermediate mail servers** for sending mail.
- SMTP uses **persistent connections.**



Alice sends a message to Bob

Electronic Mail in the Internet

SMTP

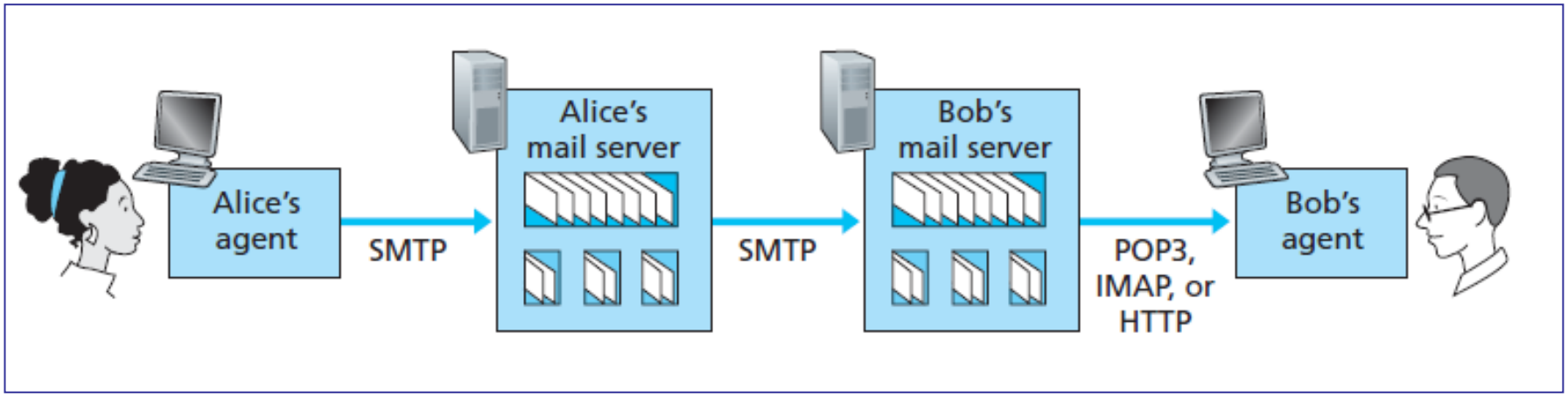
- The client SMTP has TCP establish a connection to **port 25** at the server SMTP.
- The server and client perform some **application-layer handshaking.**

S: 220 hamburger.edu
C: ***HELO*** crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: ***MAIL FROM***: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: ***RCPT TO***: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: ***DATA***
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: ***QUIT***
S: 221 hamburger.edu closing connection

Electronic Mail in the Internet

Mail Access Protocols

- We have tacitly assumed that Receiver reads his mail by logging onto the server host and then executing a mail reader that runs on that host and receiver.



E-mail protocols and their communicating entities

- User agent can't use SMTP to obtain the messages.
- Popular mail access protocols are **Post Office Protocol—Version 3 (POP3),** and **Internet Mail Access Protocol (IMAP).**

Electronic Mail in the Internet

POP3

- Extremely simple mail access protocol.
- POP3 begins when the user agent opens a **TCP connection** to the mail server on port **110.**
- Three phases: **Authorization**, **Transaction** and **Update**.
- User agent issues commands, and the server responds to each command with a reply.
- Two possible responses during **Transaction phase**: +OK and –ERR
- Two important commands during **Authorization phase**: user <username> and pass <password>.
- A user agent can often be configured to **"download and delete"** or to **"download and keep."**

Electronic Mail in the Internet

POP3

C: **list**
S: 1 498
S: 2 912
S: .
C: **retr** 1
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: **dele** 1
C: **retr** 2
S: (blah blah ...
S: .....
S: .....blah)
S: .
C: **dele** 2
C: **quit**
S: +OK POP3 server signing off

Download and delete mode

Electronic Mail in the Internet

IMAP

- The **POP3 protocol** does not provide any means for a user to create remote folders and assign messages to folders.
- IMAP has many more features than POP3, but it is also significantly more **complex.**
- An IMAP server will associate each message with a folder.
- INBOX is the first folder.
- Provides commands to allow users to create folders and move messages from one folder to another.
- Unlike POP3, an IMAP server maintains user **state information** across IMAP sessions.
- It has commands that permit a user agent to obtain components of messages.

DNS—The Internet's Directory Service

- One identifier for a host is its **hostname.**
- hostnames provide little, if any, information about the **location within the Internet of the host.**
- They would be difficult to process by **routers.**
- **People** prefer the more mnemonic hostname identifier, while **routers** prefer fixed-length, hierarchically structured IP addresses.
- 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.
    2. An **Application-Layer Protocol** that allows hosts to query the distributed database.
- DNS is commonly employed by other application-layer protocols.
- **Example:** www.someschool.edu/index.html

DNS—The Internet's Directory Service

DNS Services:

- **Translating hostnames to IP addresses.**
- **Host aliasing:** Can be invoked by an application to obtain the **canonical hostname** for a supplied **alias hostname** as well as the **IP address of the host.**
- **Mail server aliasing:** Can be invoked by a mail application to obtain the **canonical hostname** for a supplied **alias hostname** as well as the **IP address of the host.**
- **Load distribution:** Used to perform load distribution among **replicated servers**, such as replicated Web servers. A set of IP addresses is thus associated with one canonical hostname.

DNS—The Internet's Directory Service

Overview of How DNS Works:

- The application will invoke the client side of DNS, specifying the hostname that needs to be translated.
- All DNS query and reply messages are sent within **UDP datagrams** to port **53**.
- After a delay, ranging from milliseconds to seconds, **DNS in the user's host receives a DNS reply message that provides the desired mapping.**
- The **problems with a centralized design;**
  - A Single point of failure
  - Traffic volume
  - Distant centralized database
  - Maintenance

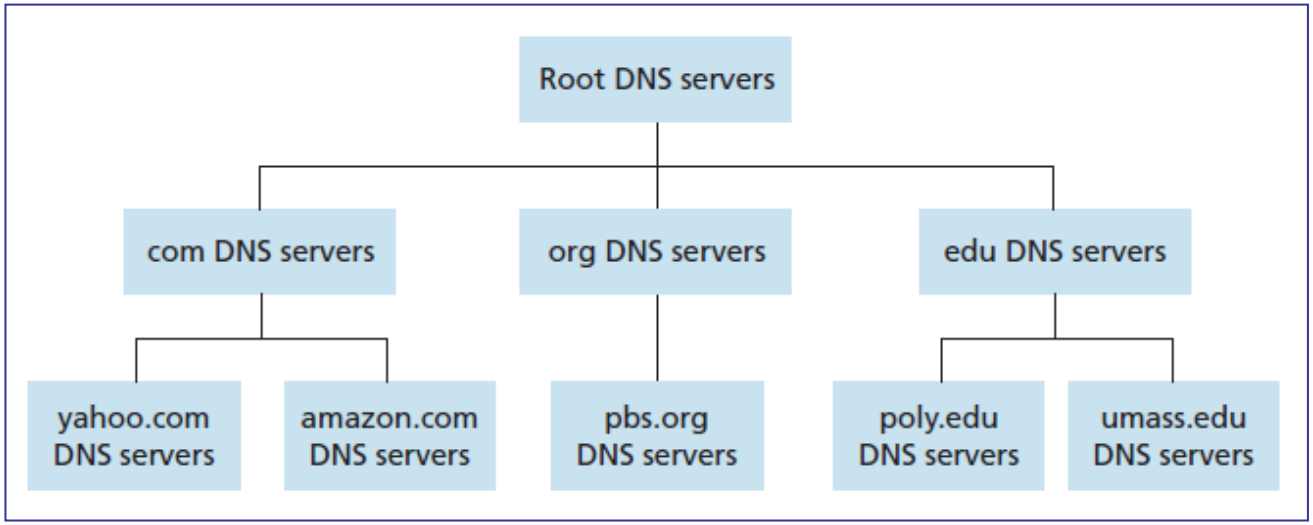
DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

- No single DNS server has all of the mappings for all of the hosts in the Internet.
- **Three classes of DNS servers**: **Root Servers**, **Top-level domain (TLD) Servers** and **Authoritative Servers**

Portion of the hierarchy of DNS servers



- **Example:** www.amazon.com

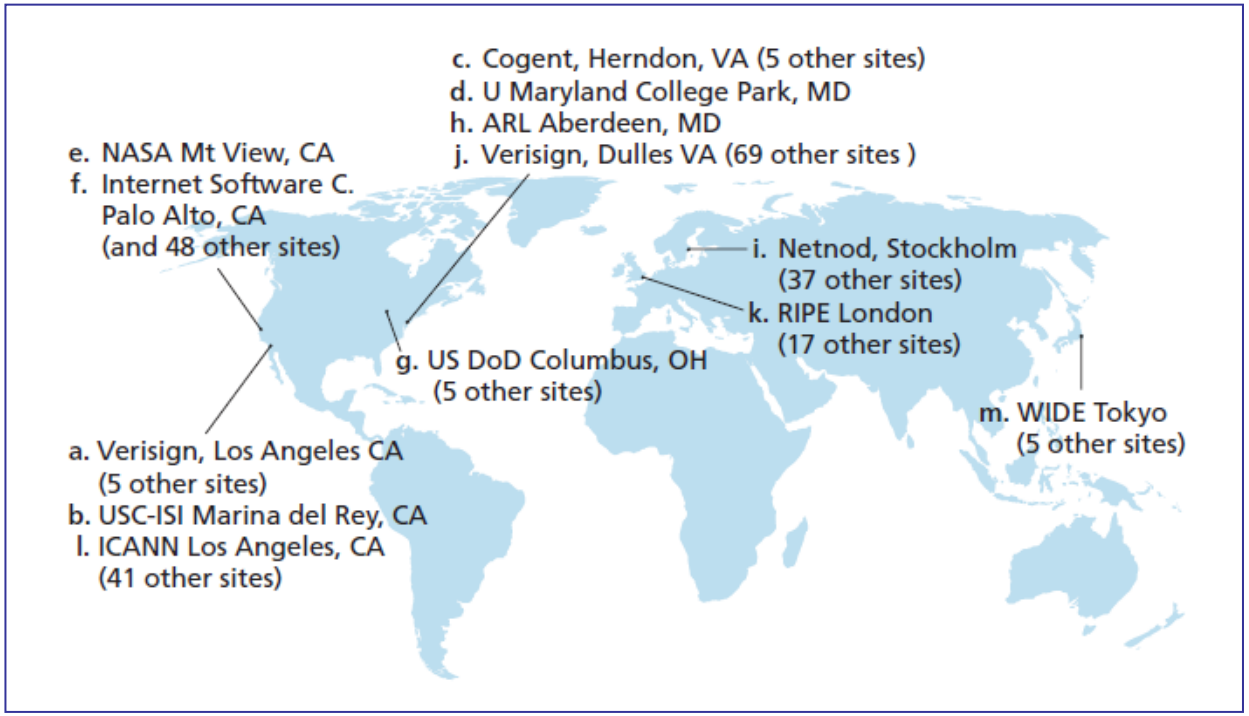
DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

- **Root DNS servers:** In the Internet there are 13 root DNS servers (labelled A through M), most of which are located in North America. All together, there are 247 root servers as of fall 2011.

DNS root servers in 2012 (name, organization, location)



DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

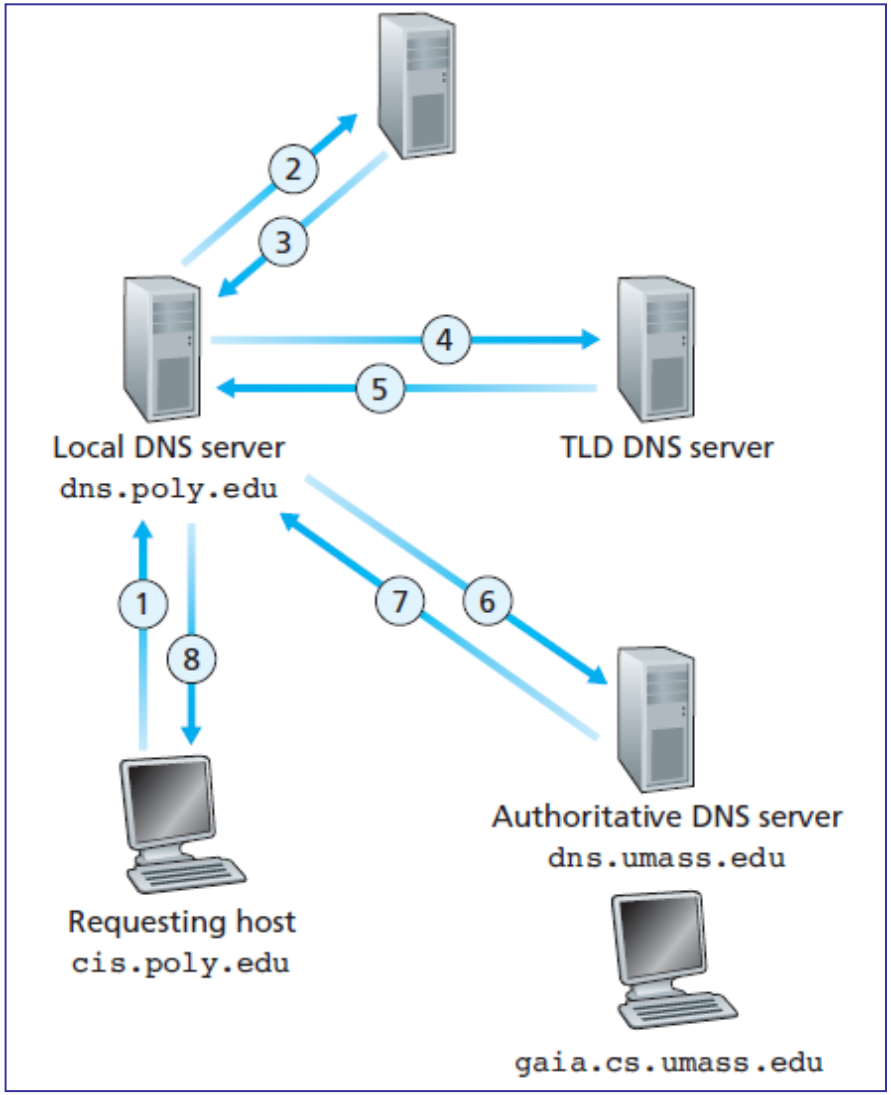
- **Top-level domain (TLD) servers:** These servers are responsible for top-level domains such as **com, org, net, edu,** and **gov**, and all of the country top-level domains such as **uk, fr, ca,** and **jp.**
- **Authoritative DNS servers:** Every organization with publicly accessible hosts on the Internet must provide **publicly accessible DNS records** that map the names of those hosts to IP addresses using organization's authoritative DNS server.
- **Local DNS 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.

DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

Interaction of the various DNS servers



DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

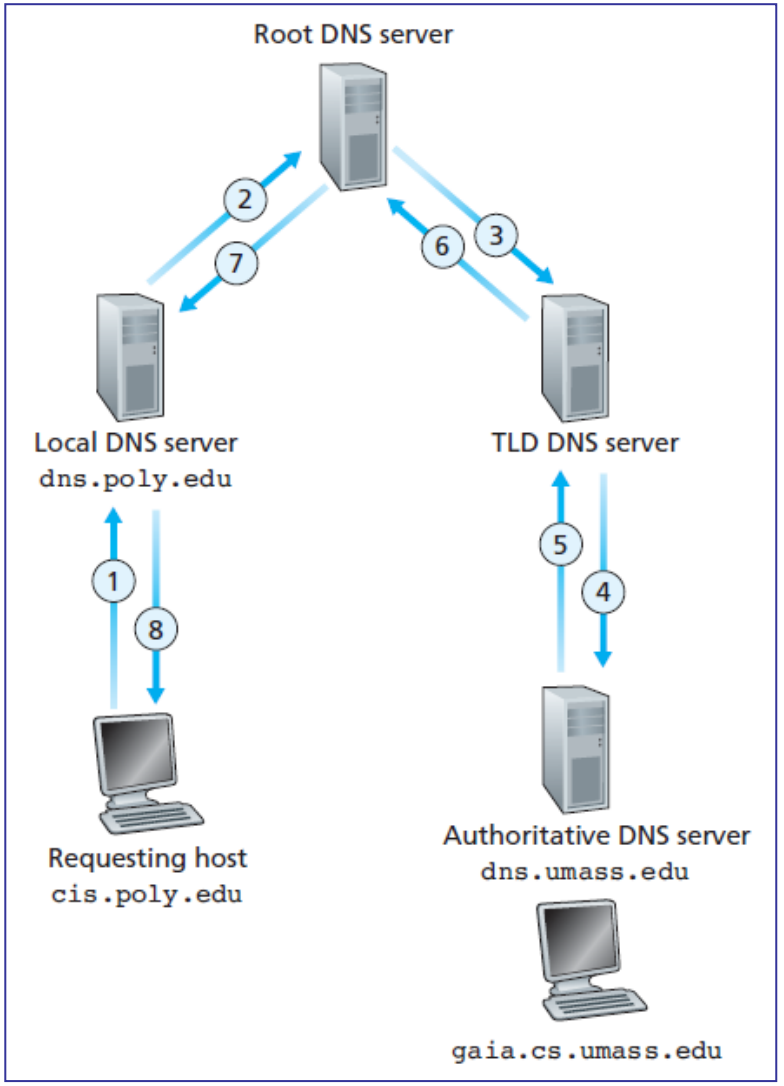
- Our previous example assumed that the TLD server knows the authoritative DNS server for the hostname. **In general this not always true.**
- **Recursive Queries** and **Iterative Queries.**
- **DNS Caching:** DNS extensively exploits DNS caching in order to improve the delay performance and to reduce the number of DNS messages moving around the internet.

DNS—The Internet's Directory Service

Overview of How DNS Works

A Distributed, Hierarchical Database:

Recursive queries in DNS



DNS—The Internet's Directory Service

DNS Records and Messages

- The DNS servers that together implement the **DNS distributed database store resource records (RRs)**, including **RRs that provide hostname-to-IP address mappings.**
- A resource record is a four-tuple that contains the following fields:

(Name, Value, Type, TTL)

- If **Type=A**, then **Name is a hostname** and **Value is the IP address** for the hostname. ***Example: (relay1.bar.foo.com, 145.37.93.126, A)***
- If **Type=NS**, then **Name is a domain** (such as foo.com) and **Value is the hostname of an authoritative DNS server** that knows how to obtain the IP addresses for hosts in the domain. ***Example: (foo.com, dns.foo.com, NS)***
- If **Type=CNAME,** then **Value is a canonical hostname** for the **alias hostname Name.** ***Example: (foo.com, relay1.bar.foo.com, CNAME)***

DNS—The Internet's Directory Service

DNS Records and Messages

- If **Type=MX**, then **Value is the canonical name of a mail server** that has an **alias hostname Name**. ***Example: (foo.com, mail.bar.foo.com, MX).***
- To obtain the **canonical name for the mail server,** a DNS client would query for an MX record.
- To obtain the **canonical name for the other server,** the DNS client would query for the CNAME record.