Computer Networks (CS F303)

BITS Pilani
Pilani Campus

Virendra Singh Shekhawat
Department of Computer Science and Information Systems

innovate achieve lead

BITS Pilani

Pilani Campus

Second Semester 2020-2021

Module-2 Application Layer

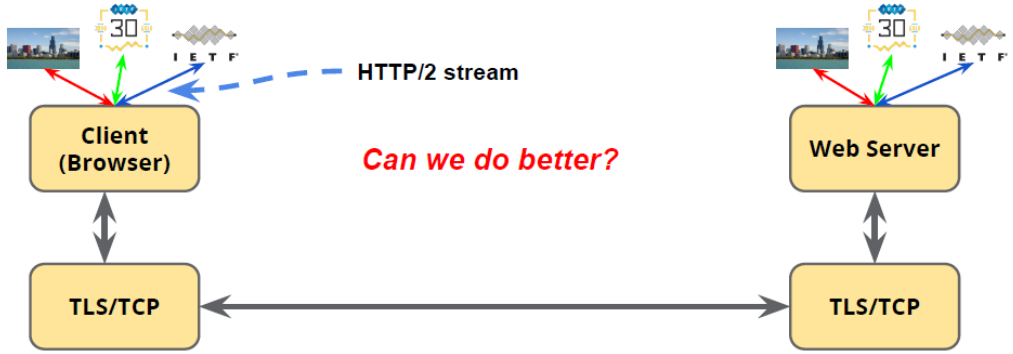
Today's Agenda

- HTTP/2 Protocol
  - Stream oriented protocol
- Domain Name System (DNS)
  - Hostname to IP address mapping system

HTTP/2Feature: Stream Multiplexing

- What is ***stream***?
  - Bi-directional sequence of text format frames sent over the HTTP/2 protocol exchanged between the server and client

- HTTP/1 is capable of transmitting only one stream at a time
  - Receiving large amount of media content via individual streams sent one by one is inefficient and resource consuming



- HTTP/2 allows transmission of parallel multiplexed requests and responses
  - A binary framing layer is created
  - This layer allows client and server to disintegrate the HTTP payload into small, independent and manageable interleaved sequence of frames
  - This information is then reassembled at the other end

HTTP/2 Feature: Server PUSH

- It allows the server to send additional cacheable information to the client that isn't requested but is anticipated in future requests.

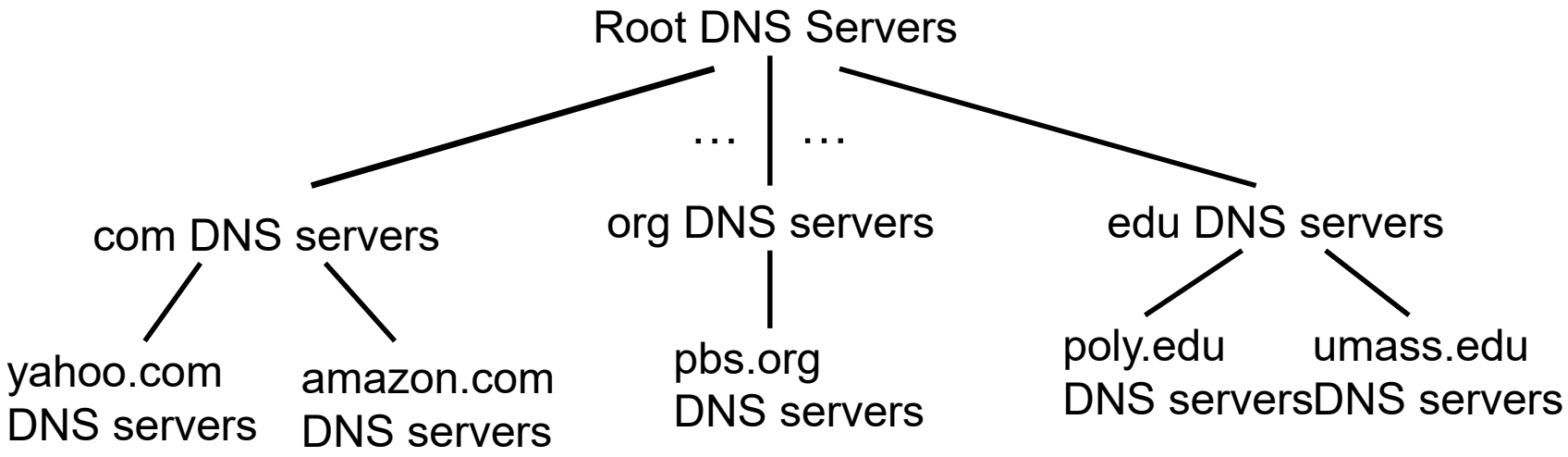
- This mechanism saves a request-respond round trip and reduces network latency.

Domain Name System (DNS)

- When a browser (HTTP client) requests the URL www.bits-pilani.ac.in/index.html
- In order to send an HTTP request msg to the destination host the client's host must obtain the IP address of the destination host

- The **domain name system** maps the **name** people use to locate a website to the IP address that a computer uses to locate a website.
- Why do we need the mapping between host name and IP address?
- *Application-layer protocol:* hosts, name servers communicate to *resolve* names (address/name translation)

DNS Structure – Distributed Hierarchical Database



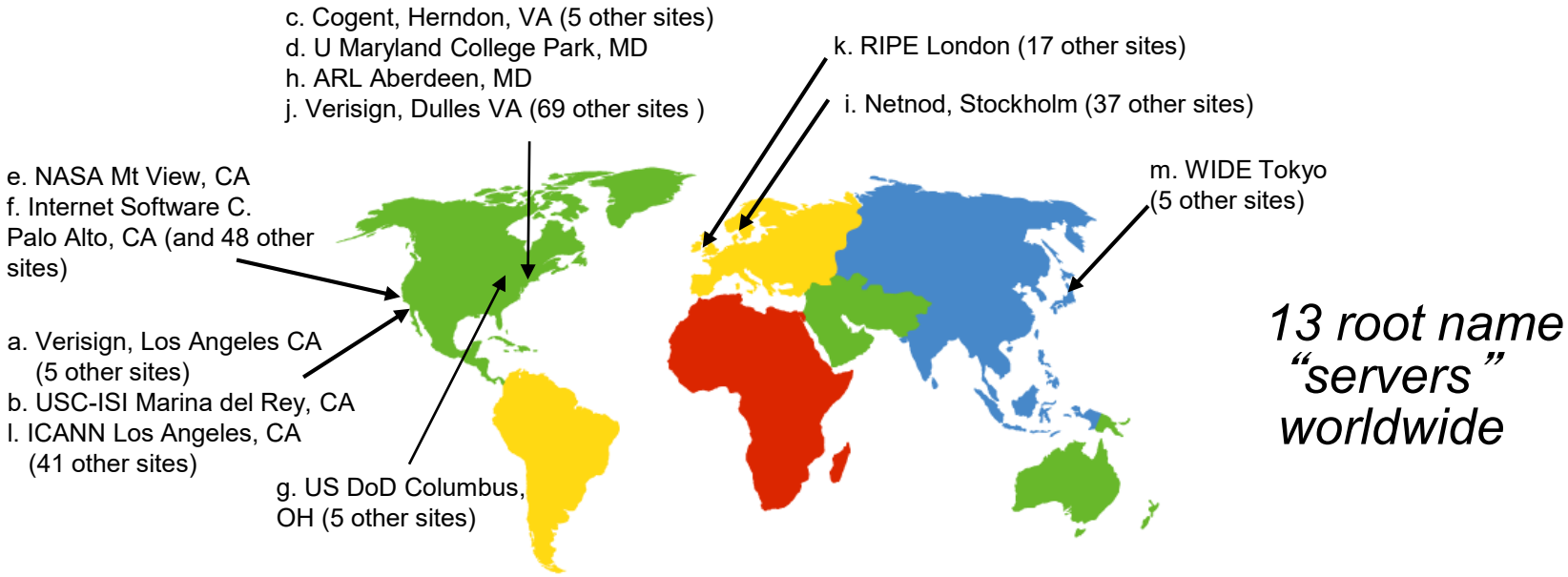
Client wants IP for www.amazon.com; 1st approx:

- Client queries root server to find com DNS server
- Client queries .com DNS server to get amazon.com DNS server
- Client queries amazon.com DNS server to get IP address for www.amazon.com

List of all top level domain servers is available at: ***https://www.icann.org/resources/pages/tlds-2012-02-25-en***

Root Name Servers

- Root name server:
  - Total 13 server, mostly located in North America.
  - Each server is actually a network of replicated servers



13 root name "servers" worldwide

DNS Services

- Hostname to IP address translation
  - Host name to IP address mapping
- Host aliasing
  - Canonical name to alias name(s) mapping
- Mail server aliasing
  - Host name to mail server mapping
- Load distribution
  - Replicated Web servers: many IP addresses correspond to one name

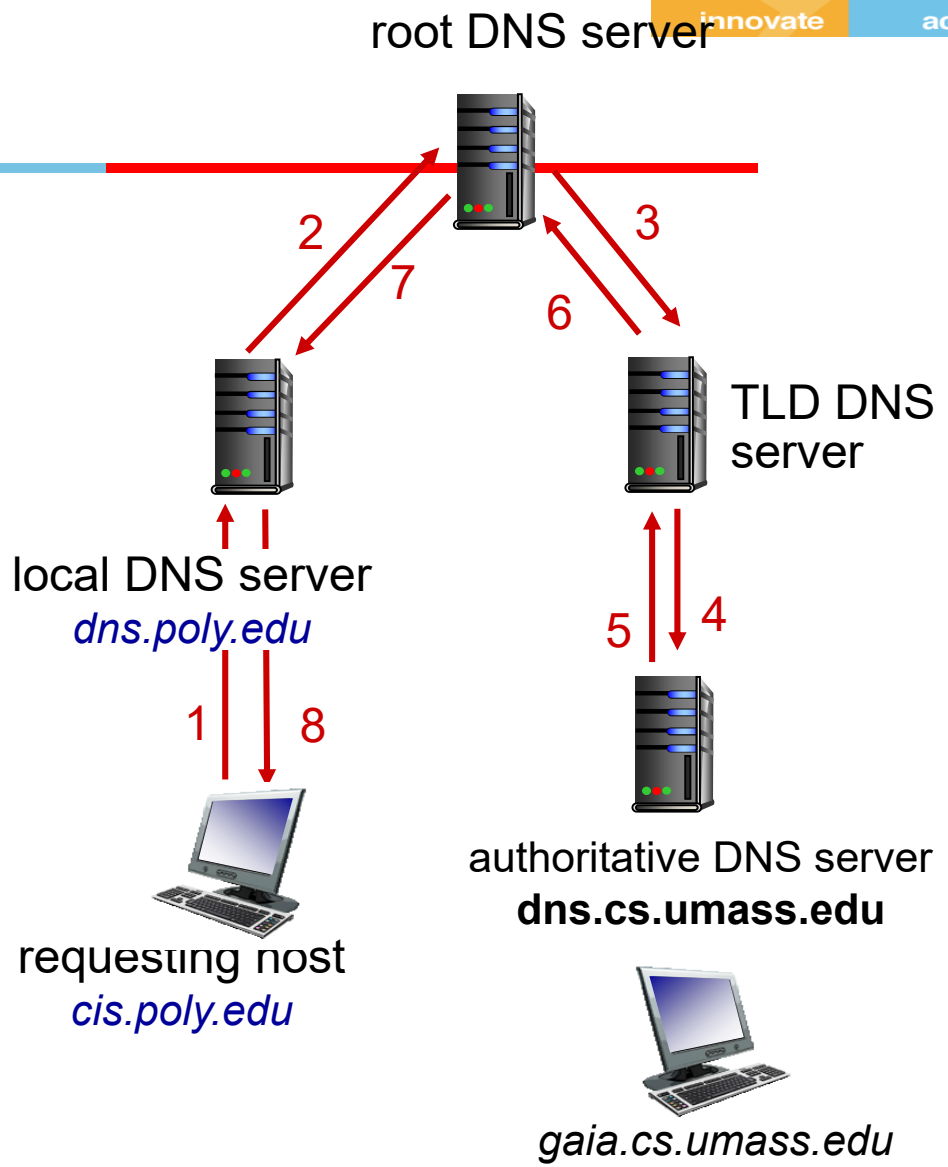
c) Let us assume that www.abc.com is a very popular web site and the many client requests cannot be handled by a single server but rather by a cluster of web servers (each having a different IP address). Describe the process that DNS offers for load balancing.

Sol: The authoritative DNS server keeps a list of IP addresses (one corresponding to each replicated web server) for canonical name of the web server. The DNS server sends an entire list of IP addresses in response to the DNS query but rotates ordering within each reply. Client typically sends request to first IP in the list.

OR

Sometimes DNS server responds the query with a single IP address from the list in round robin fashion. Eventually, load balancing is achieved at request level; i.e. number of requests (http requests) are distributed equally among multiple replicas of web server. <both answers will be considered for full credit of 2 marks>

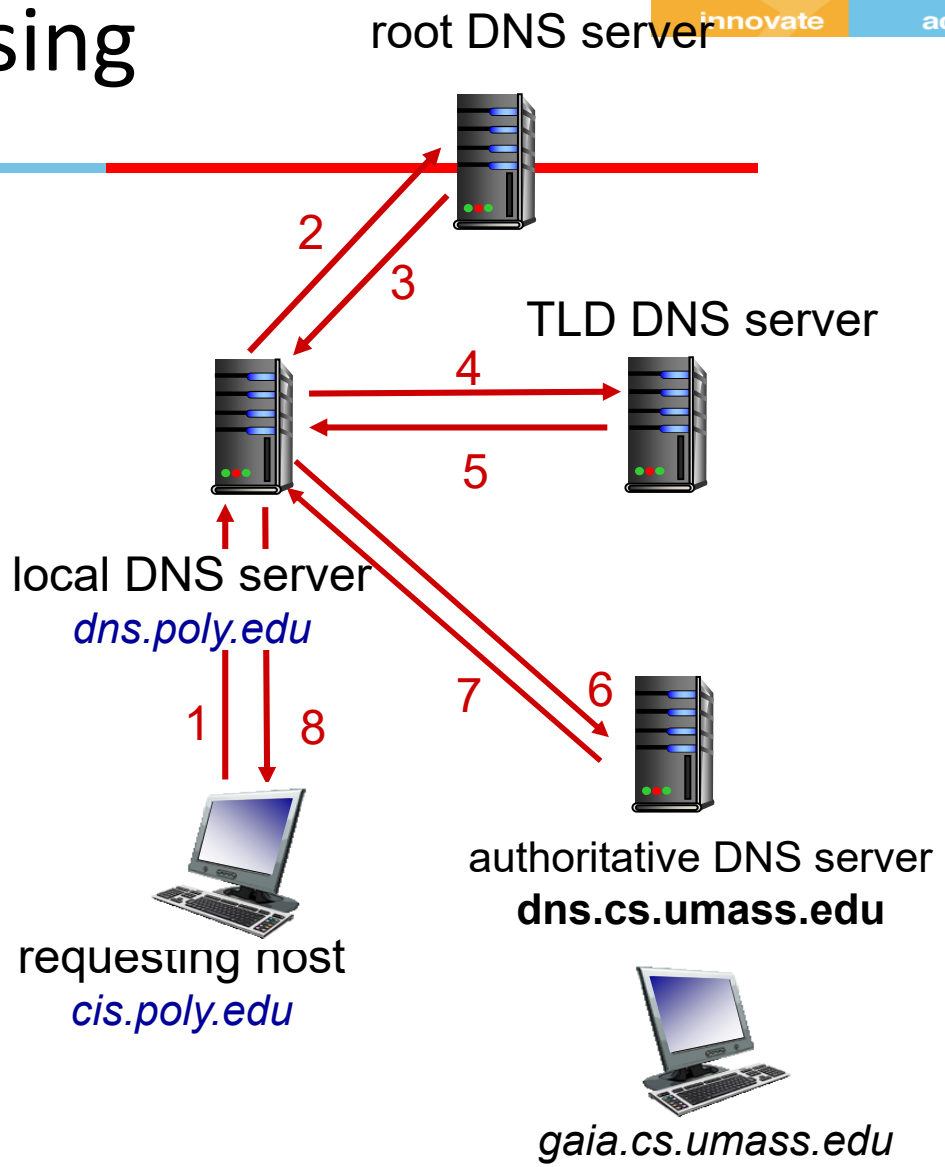
Recursive Query



Practical DNS Query Processing

- TLD server may know only of an intermediate DNS server for the hostname, which in turn knows the authoritative DNS server for the hostname.

- DNS responses are usually cached to improve the delay performance and to reduce the number of DNS messages
  - e.g., Local DNS server caches the TLD server information



DNS Records

DNS: distributed database for storing resource records (RR)

RR format: `(name, value, type, ttl)`

type=A

- **`name`** is hostname
- **`value`** is IP address

type=NS

- **`name`** is domain (e.g., foo.com)
- **`value`** is hostname of authoritative name server for this domain

type=CNAME

- **`name`** is alias name for some "canonical" (the real) name
- **`www.ibm.com`** is really **`servereast.backup2.ibm.com`**
- **`value`** is canonical name

type=MX

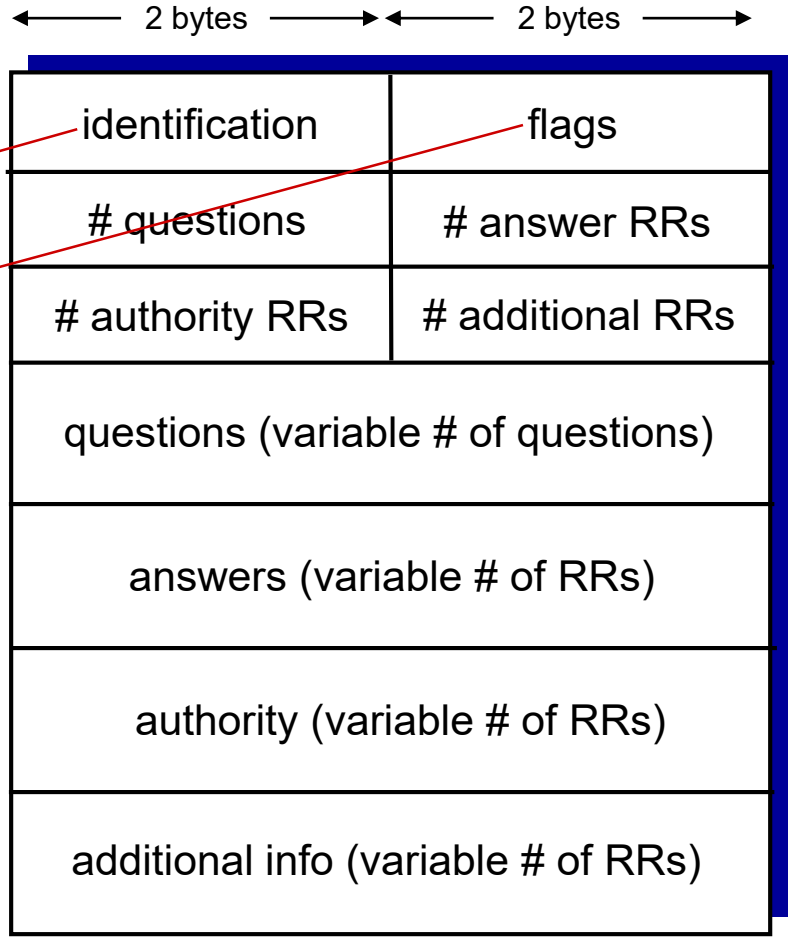
- **`value`** is name of mailserver associated with **`name (host name, i.e., mailserver alias)`**

DNS Messages

- *Query* and *reply* messages, both with same *message format*
- *Explore DNS protocol in Lab Session #2*

msg header

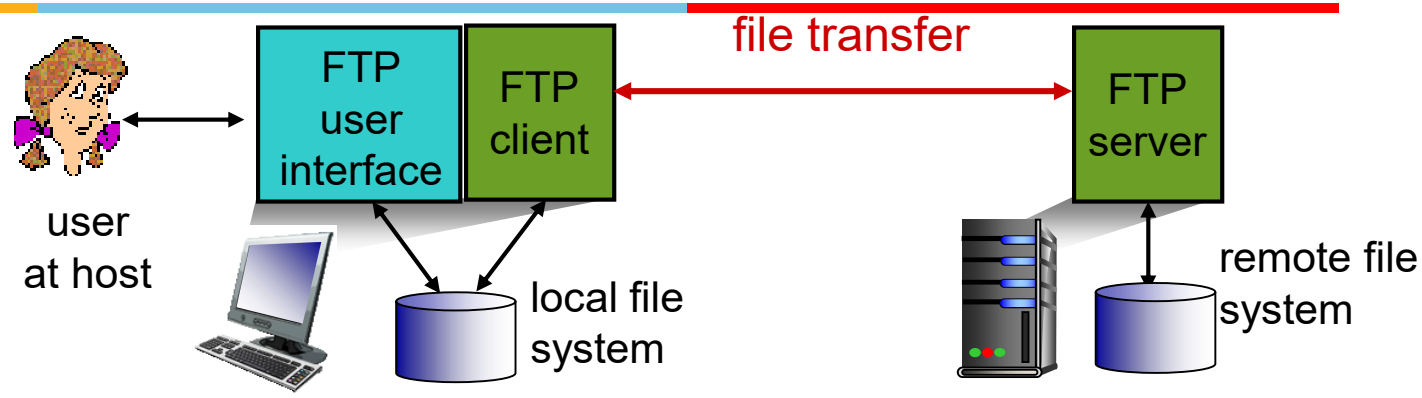
- identification: 16 bit # for query, reply to query uses same #
- flags:
  - query or reply
  - recursion desired
  - recursion available
  - reply is authoritative



Inserting Records into DNS

- A newly created domain name should be first registered at a registrar
  - Internet Cooperation of Assigned Names and Numbers (ICANN) accredits the registrars
  - Accredited registrar list is available at www.internic.net

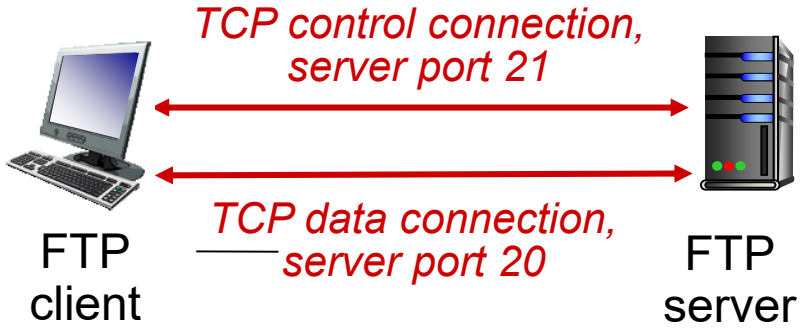
FTP: File Transfer Protocol



- Transfer file to/from remote host
- Client/server model
  - *Client:* side that initiates transfer (either to/from remote)
  - *Server:* remote host
- ftp: RFC 959
- ftp server: port 21

FTP: Connections

- Control connection
  - Authorization, directory listing etc.
- When server receives file transfer command,
  - *Server* opens *2nd* TCP data connection (for file) *to* client
- After transferring one file, server closes data connection



FTP Commands and Responses

Sample commands:

- Sent as ASCII text over control channel
- **USER *username***
- **PASS *password***
- **LIST** return list of file in current directory
- **RETR filename** retrieves (gets) file
- **STOR filename** stores (puts) file onto remote host

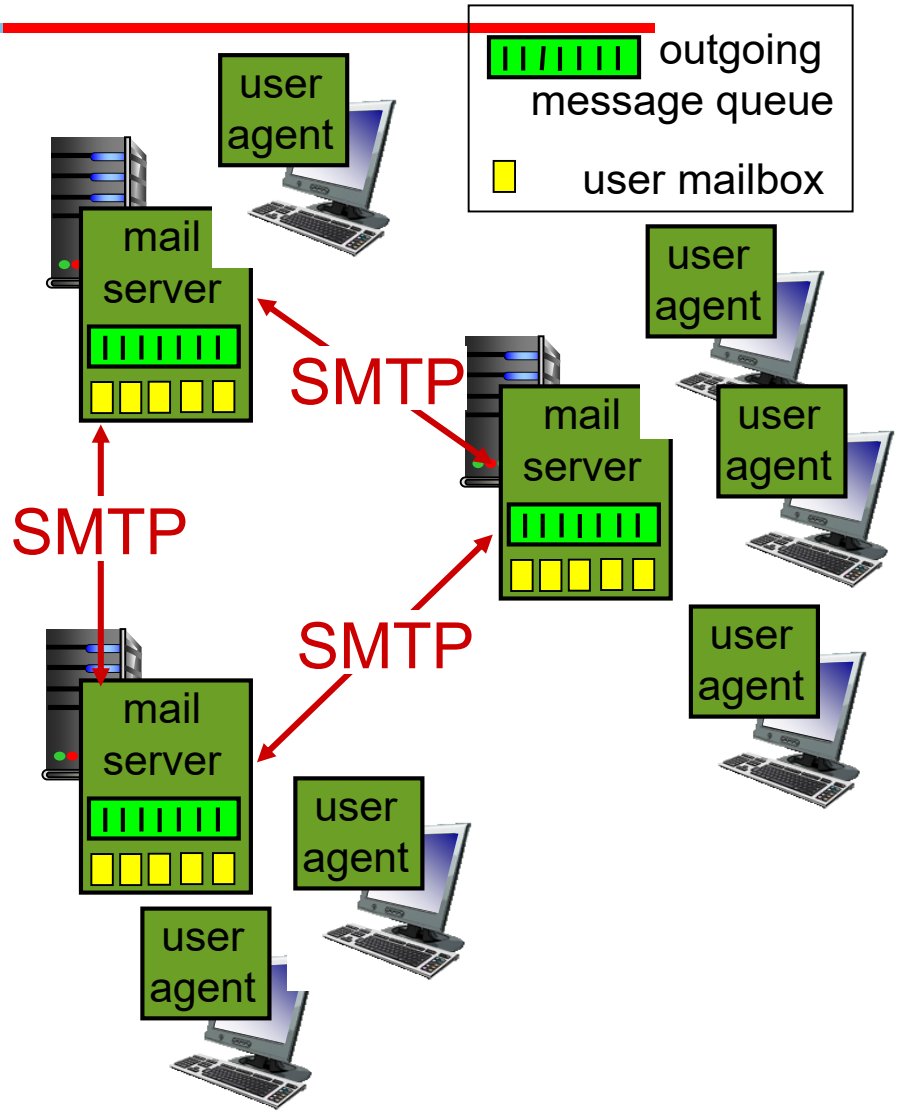
Sample return codes

- **Status code and phrase (as in HTTP)**
- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

eMail

Three major components:

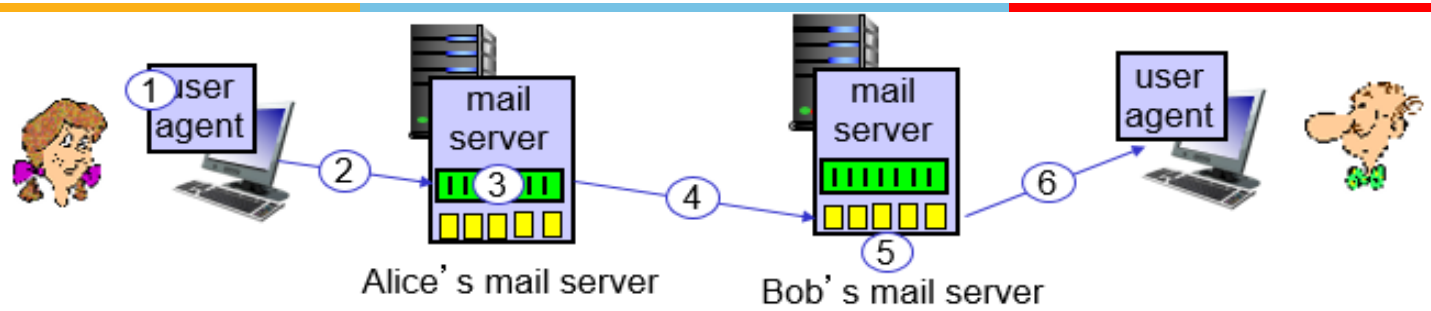
- User agents
  - e.g., Outlook, Thunderbird
- Mail servers
  - Contains incoming messages for user
- Simple mail transfer protocol:
  - SMTP



SMTP [RFC 5321, Original RFC 821]

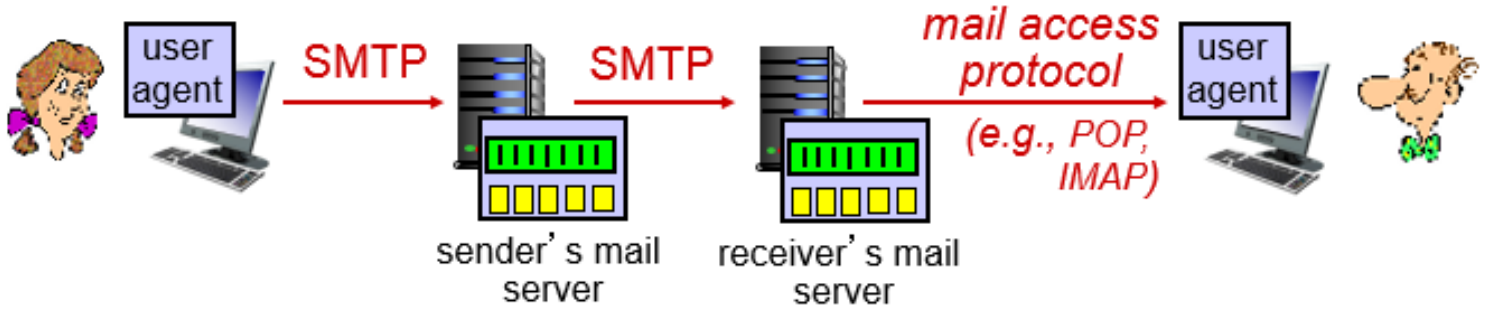
- Uses TCP to reliably transfer email message from client to server, port 25
- Direct transfer: sender's mail server to receiver's mail server
- Three phases of transfer
  - Handshaking (greeting)→Transfer of messages→Connection Closure
- Command/response interaction (like HTTP, FTP)
  - Commands: ASCII text
  - Response: status code and phrase
- Messages must be in 7-bit ASCII
  - Painful for multimedia data

Mail Transfer Process




```
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
```

Mail Access Protocols



- Mail access protocol: retrieval from server
  - POP3 [Port:110]: Post Office Protocol [RFC 1939]: authorization, download and keep, download and delete
    - User can create folders and move the messages into them locally.
    - Stateless across the sessions
  - IMAP: Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored msgs on server
    - Allows to create remote folders and maintains user state information across IMAP sessions
    - Permit a user agent to obtain components of messages. Good for low bandwidth connections.

POP3 Protocol

Authorization phase

- Client commands:
  - **`user:`** declare username
  - **`pass:`** password
- Server responses
  - **`+OK`**
  - **`-ERR`**

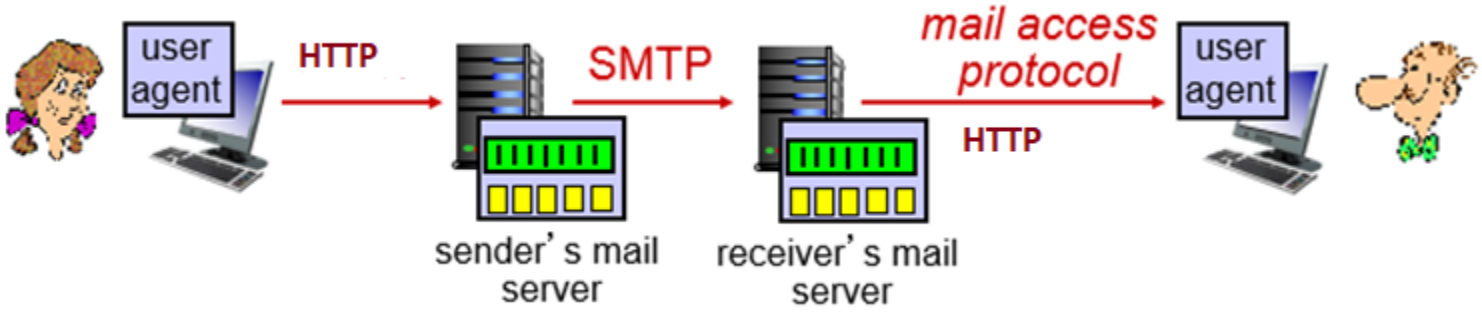
Transaction phase, client:

- **`list:`** list message numbers
- **`retr:`** retrieve message by number
- **`dele:`** delete
- **`quit`**

```
S: +OK POP3 server ready
C: user alex
S: +OK
C: pass hungry
S: +OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

Web based E-Mail

- Hotmail introduced Web-based access in the 1990s



Thank You!