Lecture-4

Circuit Switching and Packet Switching, Physical Media

- Circuit Switching
- Packet Switching
- Physical Media

Circuit Switching

- Circuit switching is a communication method where a dedicated communication path, or circuit is established between two devices before data transmission begins.
- The circuit remains dedicated to the communication for the duration of the session, and no other devices can use it while the session is in progress.
- Circuit switching is commonly used in voice communication and some types of data communication.

Circuit Switching : FDM and TDM

Example : 4 users

Reliable
Data Liability

• Guaranteed
High bandwidth

Numerical Problem.

Ques How long does it take to send a file of 640,000 bits from host A to host B over a circuit-switched network?

- All links are 1.536 Mbps
- Each link uses TDM with 24 slots/sec
- 500 msec to establish end-to-end circuit

$$\text{Each circuit data rate} = \frac{1.536 \text{ Mbps}}{24}$$

$$= 64 \text{ Kbps}$$

$$\text{time} = \frac{6,40,000}{64 \text{ Kbps}}$$

$$= \frac{10}{64000} \text{ sec}$$

$$= 10 \text{ sec}$$

$$\text{Total time} = 10 + 0.5$$

$$= 10.5 \text{ sec}$$

Advantages of Circuit Switching

- Guaranteed bandwidth

Provides a dedicated path for communication, ensuring that bandwidth is guaranteed for the duration of the call.

- Low latency

Provides low latency because the path is predetermined, and there is no need to establish a connection for each packet.

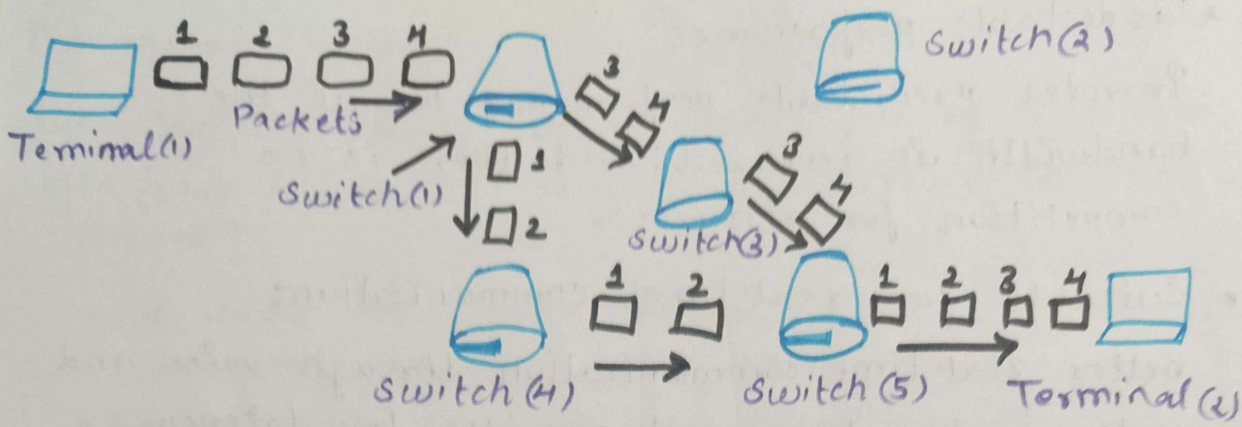
- Predictable performance
  Provides predictable performance because the bandwidth is reserved, and there is no competition for resources.
- Suitable for real-time communication:
  Better real-time communication through voice and video media because it provides low latency and predictable performance.

Disadvantages

- Inefficient use of bandwidth
  Because the bandwidth is reserved for the entire duration of the call, even when no data is being transmitted.
- High cost:
  Because it requires dedicated resources, such as hardware and bandwidth, for the duration of the call.

Packet Switching

It is a communication method where data is divided into smaller units called packets and transmitted over the network.



Store and Forward

- Takes L/R seconds to transmit (push out) packet of L bits on to link or R bps
- Entire packet must arrive at router before it can be transmitted on next link: store & forward
- delay = $3L/R$ (assuming zero propagation delay)
  ↓
  more on delay shortly

Delay time - Packet switching → ~~slow~~ High

Per packet processing time → High

Circuit Switching	Packet Switching

Physical Media

- Bit: propagates b/w transmitter/receiver pairs.
- Physical link is what lies b/w transmitter and receiver

Guided Media:

Signals propagate in solid media: copper, fiber, coaxial cable.

Unguided Media:

Signals propagate freely e.g. ratio.

Twisted Pair (TP)

Two insulated copper wires

- Category 3: traditional phone wires, 10 Mbps Ethernet
- Category 5: 100Mbps Ethernet

Coaxial cable

- Two concentric copper conductors
- Bidirectional
- Baseband
  - Single channel on cable
  - legacy ethernet
- Broadband
  - multiple channels on cable
  - HFC

Fiber optic cable:

Glass fiber carrying light pulses, each pulse a bit.

High speed operation

High speed point-to-point transmission (e.g. 10's - 100's Gbps)

Low error rate

- repeaters spaced far apart
- immune to electromagnetic noise.

Radio

- Signal carried in electromagnetic spectrum
- No physical 'wire'
- Bidirectional
- Propagation environment effects
  - reflection
  - obstruction by objects
  - interference

Radio link types:

- Terrestial microwave
  - e.g up to 45 Mbps channels
- LAN (e.g. Wi-Fi)
  - 11Mbps, 54 Mbps
- Wide-area (e.g. cellular)
  - e.g. 3G: hundreds of Kbps
- Satellite
  - Kbps to 45Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
  - Geosynchronous versus low-altitude

Lecture 5

Application Layer Protocols HTTP, FTP

- Web and HTTP
- HTTP overview
- Non-persistent HTTP
- Persistent HTTP
- HTTP Request Message
- User-server State: Cookies
- Web caches (Proxy server)
- FTP

Web and HTTP

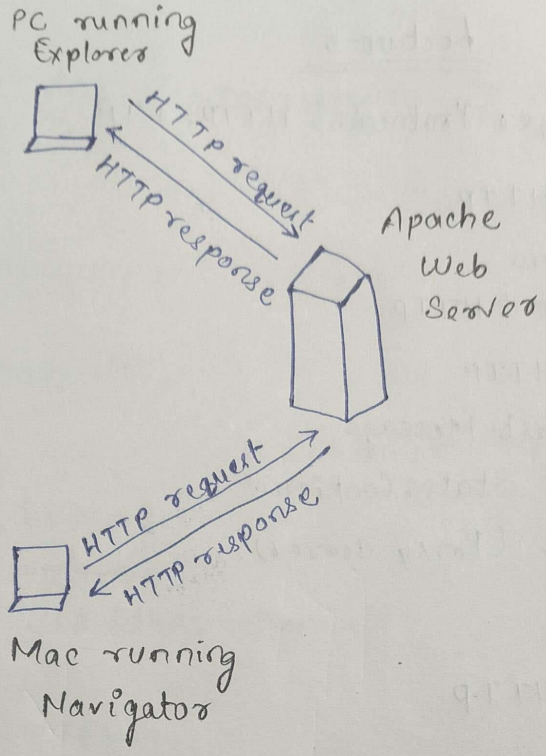
- Web page consists of objects
- Object can be HTML file, JPEG image, Java applet, audio file.
- Web page consists of base HTML-file which includes several referenced objects.
- Each object is addressable by a URL
- Example URL.

www.someschool.edu/someDept/pic.gif

host name | path name

- HTTP: hypertext transfer protocal
- Web's application layer protocal
- client/server model
  - client: browser that requests, receives, "displays" Web objects.
  - server: Web server sends objects in response to requests.
- HTTP 1.0: RFC 1945

- HTTP1.1: RFC 2068



- Uses TCP:
  - Client initiates TCP connection (creates socket) to server, port 80.
  - Server accepts TCP connection from client.
  - HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server).
  - TCP connection closed

HTTP is 'stateless'

- server maintains no information about past client requests

Protocols that maintain 'state' are complex!

- past history (state) must be maintained.
- if server/client crashes, their views of 'state' may be inconsistent, must be reconciled.

Nonpersistent HTTP

- At most one object is server over a TCP connection.
- HTTP/1.0 uses nonpersistent HTTP.

Persistent HTTP

- Multiple objects can be sent over single TCP connection blw client & server.
- HTTP/1.1 uses persistent connections in default mode.

Nonpersistent HTTP

Suppose users enters URL
www.someSchool.edu/someDepartment/home.index

1(a) HTTP client initiates TCP connection to HTTP server (process) at www.someSchool.edu on port 80

1b. HTTP server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client.

time ↓

Suppose user enters URL ... (contains text, references to 10 jpeg images)

2. HTTP client sends HTTP request message (containing URL) into TCP connection socket. Message indicates that client wants object someDepartment/home.index

3. HTTP server receives request messages, forms response message containing requested objects, and sends message into its socket.

time ↓

4. HTTP server closes TCP connection.

5. HTTP client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects.

time ↓

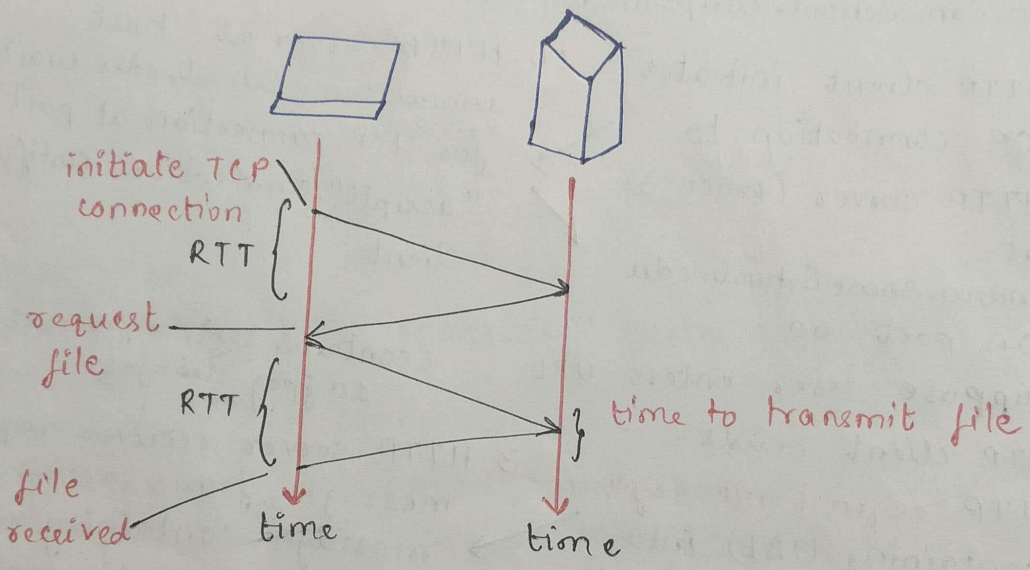
6. Steps 1-4 repeated for each of 10 jpeg objects.

Non-persistent HTTP: Response Time

Definition of RTT: time to send a small packet to travel from client to server & back.

Response time:

- one RTT to initiate TCP connection.
- one RTT for HTTP request and first few bytes of HTTP response to return.
- file transmission time. total = 2RTT + transmit time



Persistent HTTP

- server leaves connection open after sending response.
- subsequent HTTP messages blw same client/server sent over open connection.

Persistent with pipelining

- default in HTTP/1.1
- client sends requests as soon as it encounters a referenced object
- as little as one RTT for all the referenced objects.

Non Persistent HTTP issues

- requires 2 RTTs per object
- OS overhead for each TCP connection
- browsers often open parallel TCP connections to fetch referenced objects.

Persistent w/o pipelining:

- client issues new request only when previous response has been received
- one RTT for each referenced object.

HTTP Request Message

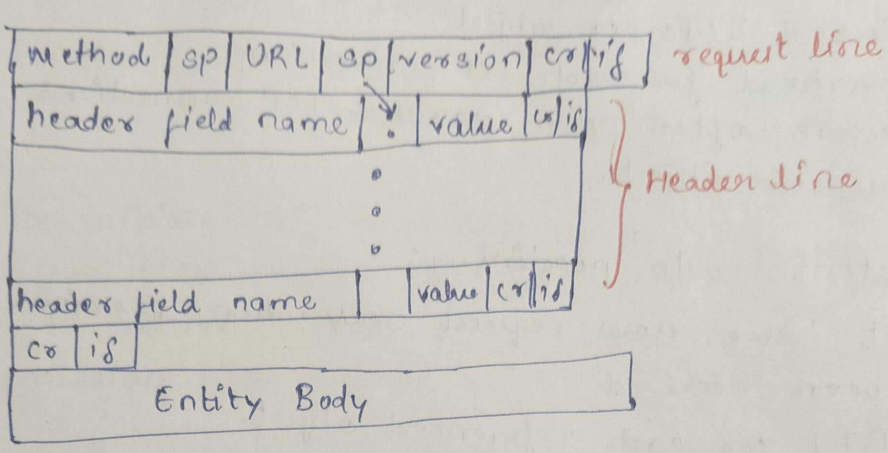
- two types of HTTP messages: request, response
- HTTP request message:
  - ASCII (Human-readable format)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language: fr
```

Header lines

Carriage return, line feed → (extra carriage return, line feed) indicate end of message

HTTP Request Message: General format



Method Types

The first line of an HTTP request message is called the request line; the subsequent lines are called the header lines. The request has three fields: the method field, the URL field, and the HTTP version field.

HTTP/1.0

- GET
- POST
- HEAD
- asks server to leave requested object out of response

HTTP/1.1

- Get, POST, HEAD
- PUT
  - uploads file in entity body to path specified in URL field
- DELETE
  - deletes file specified in the URL field.

HTTP Response Message

status line (protocol status code status phrase)

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-modified: Mon, 22 Jun 1998.....
Content-Length: 6821
Content-Type: text/html
```

header lines

data, e.g., requested HTML file

data data data data.....

HTTP Response Status Codes

In first line in server → client response message.

A few sample codes:

200 OK
- requested succeeded requested object later in this message.

301 Moved Permanently
- requested object moved, new location specified later in this message (Location)

400 Bad Request
- request message not understood by server

404 Not Found
- requested document not found on this server

505 HTTP Version Not Supported

User-server State: Cookies

Many major Websites use cookies

Four components:

1] cookie header line of HTTP response message.

2] cookie header line in HTTP request message.

3] cookie file kept on user's host, managed by user's browser

4] back-end database at web-site.

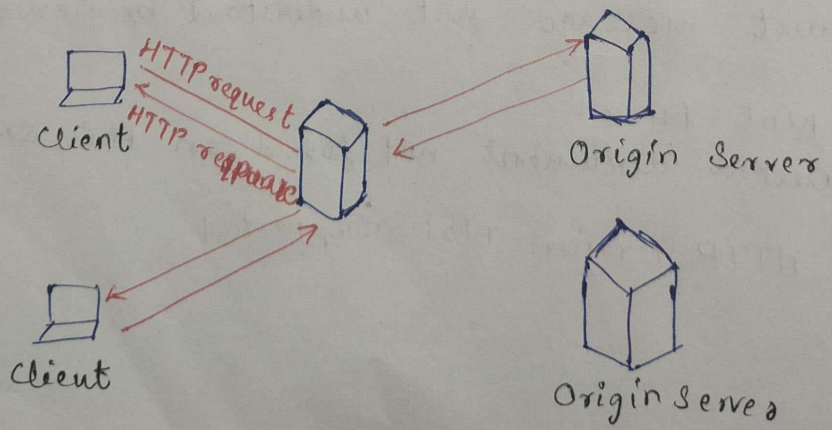
Example:

- Susan access Internet always from same PC.
- She visits a specific e-commerce site for first time.
- When initial HTTP requests arrives at site, site creates a unique ID and creates an entry in backend database for ID.

Web Caches (Proxy Server)

Goal: satisfy client request w/o involving origin server

- user sets browser: Web accesses via cache
- browser sends all HTTP requests to cache
  - object cache: cache ~~req~~ returns object
  - else cache requests object from origin server, then returns object to client



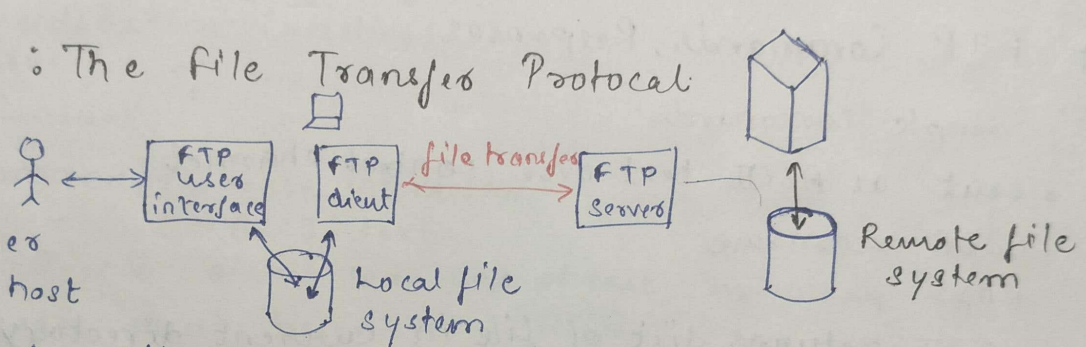
More about Web Caching

- Cache acts as both client and server
- Typically cache is installed by ISP (university, company, residential ISP)

Why web caching?

- Reduce response time for client request.
- Reduce traffic on an institution's access link.
- Internet dense with caches: enables "poor" content providers to effectively deliver content

FTP : The File Transfer Protocal

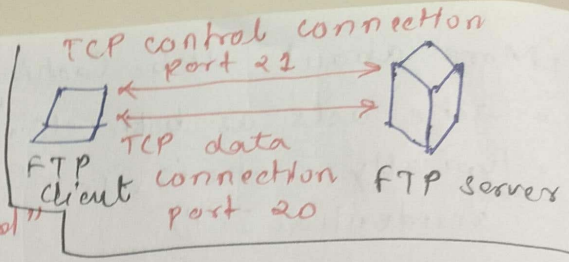


- transfer file to/from remote host
- client/server model
  - client: side that initiates transfer (either to/from remote)
  - server: remote host
- ftp: RFC 959
- ftp server: port 21

FTP: Separate Control, Data Connections

- FTP client contacts FTP server at port 21, specifying TCP as transport protocol.
- Client obtains authorization over control connection.
- Client browses remote directory by sending commands over control connection.
- When server receives file transfer command, server opens 2nd TCP connection (for file) to client.
- After transferring one file, server closes data connection.

- Server opens another TCP data connection to transfer another file.
- Control connection: "out of band"
- FTP server maintains "state": current directory, earlier authentication



- Server opens another TCP data connection to transfer another file.
- Control connection: "out of band"

FTP Commands, Responses

Sample commands:

- sent as ASCII text over control channel
- USER username
  PASS password
- LIST returns 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

Lecture-6

SMTP, POP, IMAP, DNS

→ Electronic Mail: SMTP [RFC 2821]

- uses TCP to reliably transfer email message from client to server, port 25
- direct transfer: sending server to receiving server.
- three phases of transfer
  - handshaking (greeting)
  - transfer of messages
  - Closure
- command/response interaction
  - Commands: ASCII text
  - response: status code and phrase
- messages must be in 7-bit ASCII

Scenario: Ravi Sends Message to Kamal

1. Ravi uses UA to compose message and "to" Kamal@someschool.edu
2. Ravi's UA sends message to her mail Server; message placed in message queue
3. Client side of SMTP opens TCP connection with Kamal's mail server
4. SMTP client sends Ravi's message over the TCP connection.
5. Kamal's mail server places the message in Kamal's mailbox,
6. Kamal invokes his user agent to read message.

Sample SMTP Interaction

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <ravi@crepes.fr>
S: 250 ravi@crepes.fr.... Sender ok
C: RCPT TO: <Kamal@hamburger.edu>
S: 250 Kamal@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

Try SMTP Interaction for Yourself

- telnet servername 25
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands.

above lets you send email without using email client (reader)

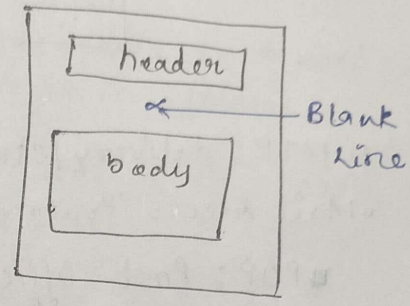
Mail Message Format

SMTP: protocal for exchanging email msgs

RFC 822: standard for text message format

- header lines e.g.,
  - To:
  - From:
  - Subject:

different from SMTP commands!



- body
  - the "message", ASCII characters only

Message Format: Multimedia Extensions

- MIME: multimedia mail extension, RFC 2045, 2056
- additional lines in msg header declare MIME content type.

MIME version

method used to encode data

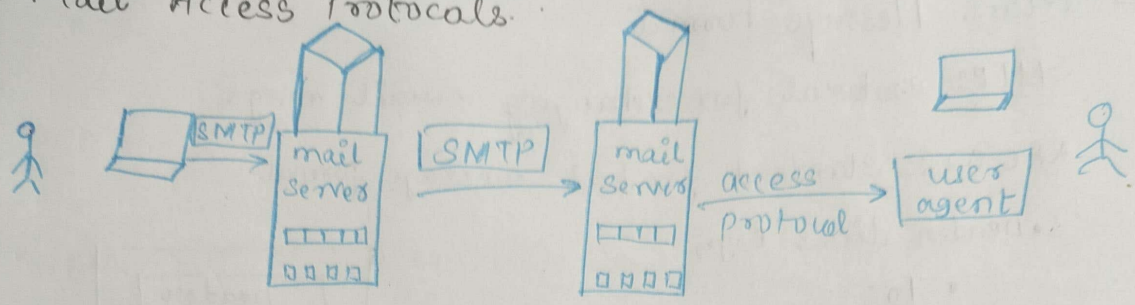
multimedia data type, subtype, parameter declaration

encoded data

```
From: ravi@crepes.fr
To: Kamal@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding:
        base64
Content-Type: image/jpeg

base64 encoded data ----
.....
---- base64 encoded data
```

Mail Access Protocols



- SMTP: delivery/storage to receiver's server
- Mail Access Protocol: retrieval from server
  - POP: Post Office Protocol [RFC 1939]
    - authorization (agent <--> server) and download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - more features (more complex)
    - manipulation of stored msgs on server
- HTTP: Hotmail, Yahoo! Mail, etc

POP3 Protocol

authorization phase
- client commands:
  - **user:** declare username
  - **pass:** password
- server response
  - **+OK**
  - **-ERR**

```
S: +OK POP3 server ready
C: user kamal
S: +OK
C: pass hungry
S: +OK user succesfully logged on
```

transaction phase; client:
- **list:** list message numbers
- **retr:** retrieve message by number
- **dele:** delete

```
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
```

POP3 (More) and IMAP

More about POP3

- Previous example uses "download and delete" mode
- Kamal cannot re-read email if he manages client.
- "Download-and-keep" copies of messages on different clients
- POP3 is stateless across sessions.

IMAP

- Keeps all messages in one place: the server
- Allows user to organize messages in folders
- Keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

DNS: Domain Name System

People: many identifiers:
- SSN, name, passport #

Internet hosts routers
- IP address (32 bit) used for addressing datagrams.
- "name, e.g., www.yahoo.com - used by humans.

Domain Name System
- distributed database implemented in hierarchy of many name servers
- application-layer protocol host, routers, name servers to communicate to resolve names (address/name translation)
  - note: core Internet f", implemented as application-layer protocol.

DNS Services

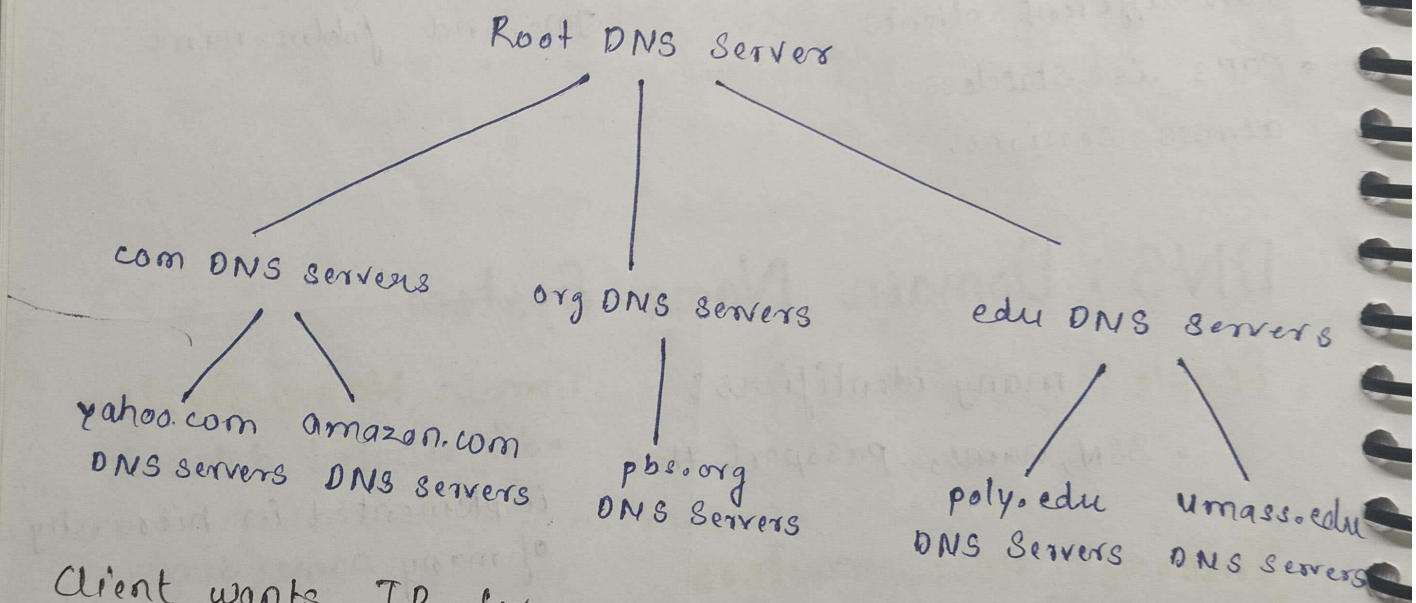
- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

Why not centralize DNS?

- Single point of failure
- traffic volume
- distant centralized database
- maintenance

doesn't ~~scale~~ scale!

Distributed, Hierarchical Database



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

- Client queries a 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

Local Name Server

- Does not strictly belong to hierarchy.
- Each ISP (residential ISP, company, university) has one.
  - Also called "default name server"
- When a host makes a DNS query, query is sent to its local DNS server.
  - Acts as a proxy, forwards query into hierarchy.

Example Iterative Query

- Host at cis.poly.edu wants IP address for gaia.cs.umass.edu
- iterated query
  - contacted server replies with name of server to contact