

# Web Programming (CS-A)

## CS 406

Shamsa Abid

Lecture 1

# Course Details

- Room CS-11
- Mon-Wed 8:00 – 9:20 am
- Pre-Req: Databases

# Web Programming Course Outline

Lecture#	Topic	Reference	Assessment
1	Intro to course, Web Essentials	A: 1,2,3, B: 1	
2	Web Technologies and Architecture Overview		
3	HTML	A: 4, B:2	
4	CSS	B: 3	Assignment 1
5,6	JavaScript and AJAX	A: 8, B: 4	
7	DOM	B: 5	
8	XML, JSON	B: 7, C: 8	Assignment 2
9	Introducing MERN full stack development		
10	Writing web servers in Node.js		
11	Basic Express setup		
12	Routes and JWT Authentication		

# Web Programming Course Outline

L#	Topic	Ref	Assessment
13	More Routes, Models and Validations		
14	React, Bootstrap - FrontEnd		
15	Redux authentication		
16	Building the frontend - part 1		
17	Building the frontend - part 2		
18	Front end unit testing		
19	Deploy		Assign. 3
20	Responsive design - Flex box, styled components		
21	Guest lecture		
22	Performance and scalability issues		
23	Prevention from Sql Injection and cross-site attacks		

# Web Programming Course Outline

Lecture#	Topic	Reference	Assessment
24	Web Services	B: 9	
25	Search Technologies	A: 12	
26	Trends and Future Directions	A: 13	
27	Project Presentations		Project
28	Course Wrap Up		
	<b>Assessment</b>		
	Assignments	35%	
	Mid Term	25%	
	Project	40%	
	Total	100%	

# Questions about you

- Databases?
- Networking? Data communications?
- HTML?
- Blogging?
- Programming courses?
- Languages?

# Agenda

- The Internet
- Basic Internet Protocols
- The World Wide Web
- HTTP Request Message
- HTTP Response Message
- Web Clients
- Web Servers

# The Internet

A circular phylogenetic tree diagram illustrating the evolutionary relationships among various accessions. The tree is rooted at the center, with numerous branches extending outwards to terminal taxa. Each branch is labeled with an accession number, typically following the format 207.205.XXX.YYY. The labels are color-coded: green for most tips, red for some internal nodes, and blue for others. The tree shows a complex branching pattern, indicating multiple lineages and potential gene flow or hybridization events.



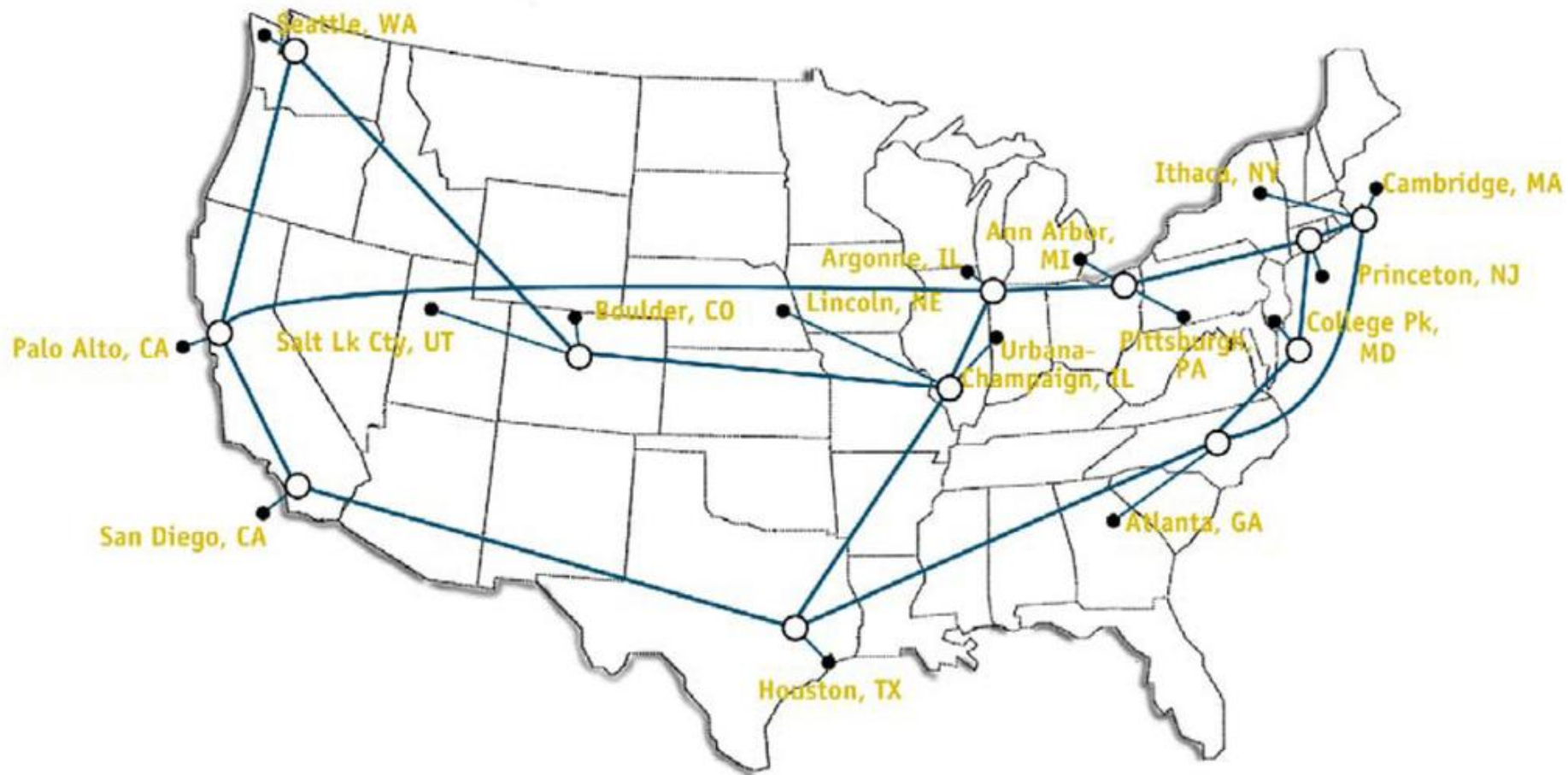
# The Internet

- What it is? Where it came from?
- US DoD's ARPANET project
  - Began late 1960s
  - Purpose: to connect researchers and develop tools for heterogeneous networking
- ARPANet 4 systems 4 OS

# The Internet

- Benefits of networking
  - Email in 1972
- Internet Protocols
  - FTP, SMTP
  - TCP/IP
    - Host-to-host communication with LANs and between networks
    - 1982 ARPANET adopted TCP/IP
- 1985 NSFNET
  - Goal to connect regional supercomputers
  - Backbone through which other networks could interconnect

# NSFNET T3 Network 1992



# The Internet

- Original backbone 56 kbits/s
  - Textual traffic
  - 1988 1.5 Mbits/s
  - 1991 45 Mbits/s
  - Kept growing
- 1990
  - NSFNET center of the internet
    - The collection of computer networks connected via the public backbone and communicating using TCP/IP
  - Commercial dial-up offered
- Initial reservations on conducting commerce
- 1995 private firms and ISPs

# The Internet

- In summary, the Internet is a collection of computers that can communicate with one another using TCP/IP over an open, global communications network.

# Internet Governance

- The technical underpinning and standardization of the core protocols (IPv4 and IPv6) is an activity of the Internet Engineering Task Force (IETF)
- The principal name spaces of the Internet are administered by the Internet Corporation for Assigned Names and Numbers (ICANN).
- ICANN coordinates the assignment of unique identifiers for use on the Internet, including domain names, Internet Protocol (IP) addresses, application port numbers in the transport protocols, and many other parameters



ICANN headquarters in the Playa Vista neighborhood of Los Angeles, California, United States.

# Basic Internet Protocols

- Def
- TCP/IP
- HTTP

# Basic Internet Protocols

- Def
  - A communication protocol is a detailed spec of how a communication between two computers will be carried out in order to serve some purpose



# TCP/IP

- Developed in 1980s
- Outgrowth of ARPANET
- Design goal
  - Establishing a decentralized and distributed network topology
  - Packet switching technology
    - Message split up, different route on network and reassemble by recipient
    - INWG- Internet working Group- connecting heterogeneous networks

# TCP/IP

- Bodies that define standards or manage
  - *IAB (Activities Board)*
  - *IRTF (Internet Research Task Force)*
  - *IETF (Engg Task Force)*
  - *IESG (Engg Steering Group)*
- Proposals for new protocols are provided in the form of RFCs Requests for Comments. Once approved the RFCs are treated as the standard documentation for the new or updated protocol.

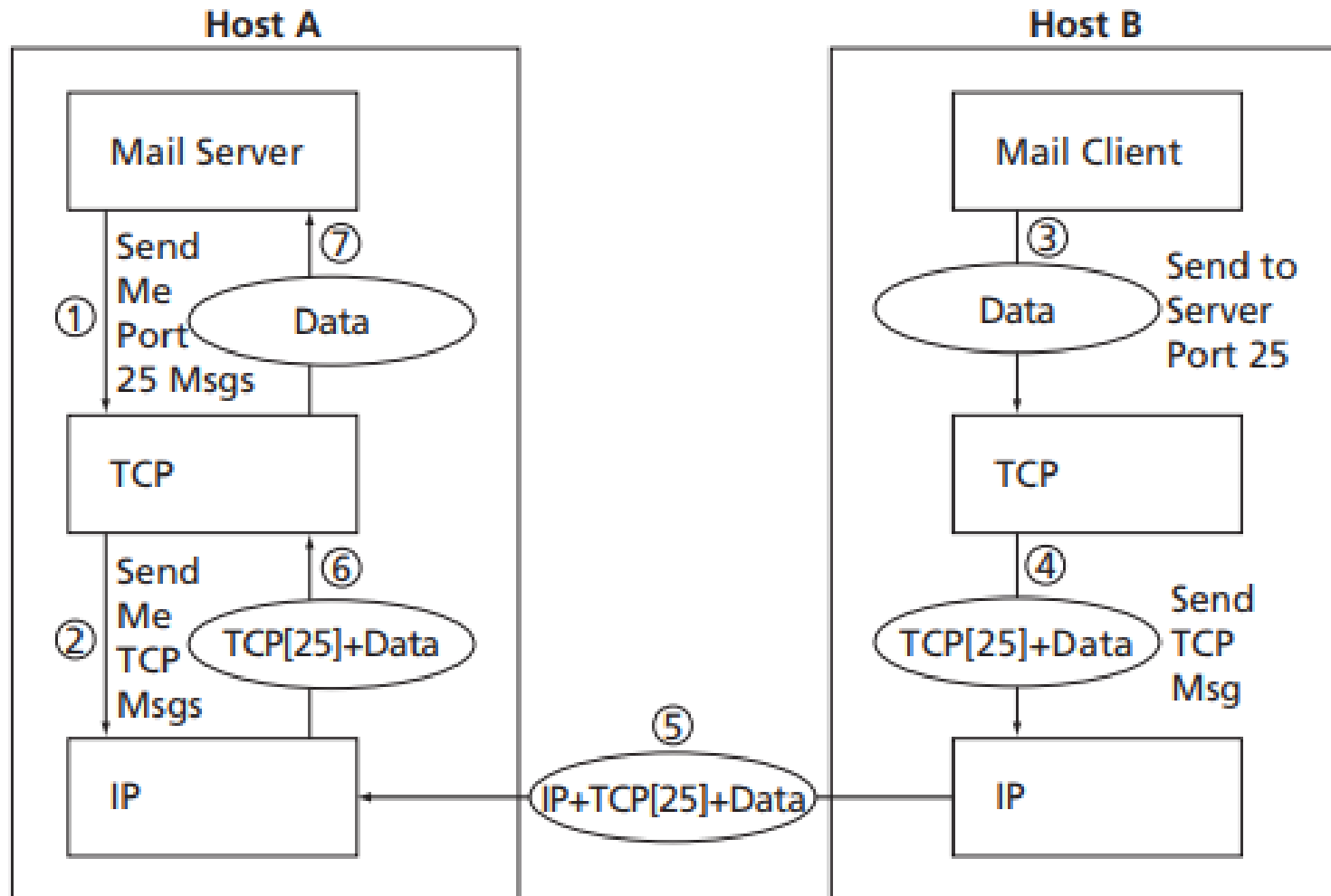
# TCP/IP

- TCP and IP
- Services
  - email, browsing, file downloads, accessing remote databases
- IP address
  - 32 bit
  - 192.0.34.166
- Function of IP software
  - Transfer data from source to destination
  - Creates a packet with header info
  - Gateway
  - Hops
  - Routes and BGP
  - checksum

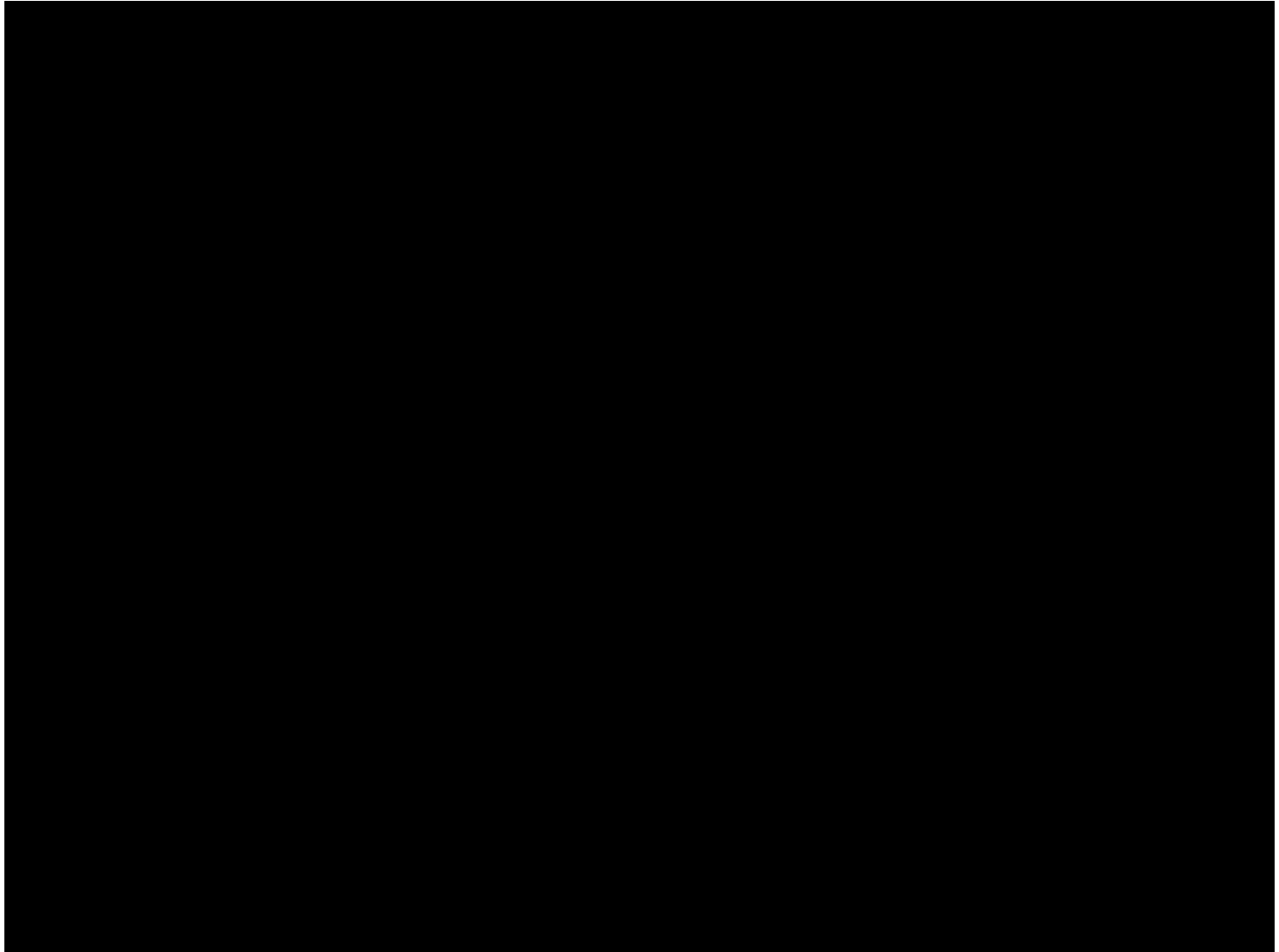
# TCP

- Reliable communication
- Full duplex communication
- Acknowledgement

# Communication using TCP/IP



# Protocol Layers of TCP/IP



# Protocol Layers of TCP/IP

- Protocol stack
  - Network interface layer
    - Lowest level of data transmission
  - Internet layer (IP)
    - Intersystems comm mechanisms, message routing, validity checking, composition of message headers, transmission of error and control msgs
  - Transport layer (TCP)
    - Reliable, connection oriented message transport.
    - UDP – audio video streaming
  - Application layer
    - Session mgmt, presentation

# Stateful Protocol

- Allows client to support sequence of commands within a session
- Server is required to maintain the state of connection until terminated
- HTTP is *stateless*.
  - Does not rely on persistent connection for communication
  - AN HTTP request consists of single request command from client and single response from server



# How do TCP/IP clients and servers communicate?

- TCP/IP client programs open a socket, which is simply a TCP connection between the client machine and the server machine.
- Servers listen for connection requests that come in through specific ports. A port indicates which server program is its intended recipient.
- Servers listen for requests on well-known port numbers.
- For example, Telnet servers normally listen for connection requests on port 23, SMTP servers listen to port 25, and web servers listen to port 80.

# DNS

- Domain Name Service
- Human friendly names
  - [www.example.org](http://www.example.org)
- IP address and host name resolution
- Nslookup
- Reverse lookup

# TCP/IP Application services

- Telnet
- E-mail
  - SMTP
  - POP
  - IMAP
- Security and encryption
- File server protocols

# TCP/IP Application services

- Telnet
  - to 'log in' to a remote machine over the Internet.
  - you can use a Telnet client program to connect and 'talk' to *any TCP* server by knowing its address and its port number.

# TCP/IP Application services

- E-mail
  - had its roots in military interests
  - The transmission of electronic mail is performed through the SMTP protocol.
  - The reading of electronic mail is usually performed through either POP or IMAP.
  - SMTP (*Simple Mail Transfer Protocol*)
    - runs as a *server, service, or daemon*
    - They wait for requests to send electronic mail messages, which can come from local system users or from across the network
    - The applications responsible for transmitting e-mail messages, such as SMTP servers, are known as MTAs (Mail Transfer Agents). Likewise, the applications responsible for retrieving messages from a mailbox, including POP servers and IMAP servers, are known as MRAs (Mail Retrieval Agents).
  - POP
  - IMAP

# TCP/IP Application services

- SMTP (*Simple Mail Transfer Protocol*)
  - E-mail client programs are known as MUAs (Mail User Agents). MUAs talk to MRAs to read mail, and to MTAs to send mail.
    - First, it must *connect* to the server. It does this by opening a TCP socket to port 25 (the SMTP port) of the server.

# TCP/IP Application services

- Interaction with an SMTP server
  - the client program identifies itself to the server via the 'HELO' command.
  - If the server accepts the request, it waits for the client to send further information.
  - client transmits information about the originator of the message (using the 'MAIL' command) and each of the recipients (using a series of 'RCPT' commands).
  - the client tells the server it is about to send the actual data: by sending a command line consisting of only the word 'DATA'. Every line that follows, until the server encounters a line containing only a period, is considered part of the message body.
  - Once it has sent the body of the message, the client signals the server that it is done, and the server transmits the message to its destination (either directly or through gateways).
  - Having received confirmation that the server has transmitted the message, the client closes the socket connection using the 'QUIT' command.

# TCP/IP Application services

- POP
  - *Post Office Protocol, gives users direct access to their e-mail messages* stored on remote systems.
  - *POP3 is the most recent version* used by Outlook etc
- *IMAP*
  - IMAP was intended as a successor to the POP protocol
  - IMAP servers provide support for multiple remote *mailboxes or folders*, so users can move messages from an incoming folder (the 'inbox') into other folders kept on the server



# TCP/IP Application services

- Security and encryption
  - All network traffic (email, chat msgs, etc) is available to anyone with sniffing tools
  - There has to be a key to encrypt your data
  - SSH (Secure Shell) secure replacement for Telnet
    - Uses public key authentication
    - Key pair: private key and public key
    - Messages encrypted with public key can only be decrypted with private key and vice versa
    - Keygen utility included with SSH clients
    - Private key on users computer, location known to ssh client
    - Public key on server, location known to ssh server program

# TCP/IP Application services

- File server protocols FTP
  - authenticate the connecting user, provide the user with information about available files, and allow the user to retrieve selected files.
  - FTP servers also allow users to traverse to different directories within the server's local file system, and (if authorized) to upload files into those directories.
  - FTP servers can allow open access to files without requiring explicit user authentication, using a service called anonymous FTP

# WWW

- History
  - 1989 Tim Berners Lee at CERN presented a proposal for an info mgmt system for sharing resources over computer network
  - Promoted the web as a virtual library
    - Online docs could be accessed via a unique document address, a URL (universal resource locator) and cross-referenced via hyertext links

# WWW

- Web pages, web sites and web application
  - One of the first applications that Tim Berners Lee demonstrated at CERN was designed to look up numbers in an online phone book using a web browser
- A web site simply displays content from static files. A web application can present dynamically tailored content based on request parameters, tracked user behaviors and security considerations.
  - Online shopping site

# WWW

- World Wide Web
  - the collection of machines (web servers) on the Internet that provide information via HTTP
  - particularly those that provide HTML documents.
- The protocol used by the Web is the Hypertext Transport Protocol, HTTP.
  - supports a client requesting a document from a server and the server returning the requested document.

# HTTP

- HTTP is a form of communication protocol
- HTTP communication follows a *request–response model*.
- HTTP interaction is initiated by a client sending a request message to the server; the server is then expected to generate a response message.

# HTTP

- I typed `http://www.example.org` in the Location bar
- When I pressed the Enter key, the browser
  - created a message conforming to the HTTP protocol
  - used DNS to obtain an IP address for [www.example.org](http://www.example.org)
  - created a TCP connection with the machine at the IP address obtained
  - sent the HTTP message over this TCP connection
  - received back a message containing the information that is shown displayed in the *client area of the browser*

# HTTP Request Message

- Structure
- HTTP Version
- Request-URI
- Request Method
- Header Fields and MIME Types



# HTTP Request Message

- Structure
  - Start line
  - Header field(s) (one or more)
  - Blank line
  - Message body (optional)

# HTTP Request Message

- Structure
  - Start line
    - GET / HTTP/1.1
  - Every start line consists of three parts, with a single space used to separate adjacent parts:
    - 1. Request method
    - 2. Request-URI portion of web address
    - 3. HTTP version

# HTTP Request Message

- Structure
  - Request-URI
- The concatenation of the string `http://`, the value of the Host header field (`www.example.org`, in this example), and the Request-URI (in this example) forms a string known as a *Uniform Resource Identifier* (URI).
- A URI is an identifier that is intended to be associated with a particular resource (such as a web page or graphics image) on the World Wide Web.

# HTTP Request Message

## Some Non-http URL Schemes

Scheme Name	Example URL	Type of Resource
ftp	ftp://ftp.example.org/pub/afile.txt	File located on FTP server
telnet	telnet://host.example.org/	Telnet server
mailto	mailto:someone@example.org	Mailbox
https	https://secure.example.org/sec.txt	Resource on web server supporting encrypted communication
file	file:///C:/temp/localFile.txt	File accessible from machine processing this URL

# HTTP Request Method

- The primary HTTP method is GET.
  - used when you type a URL
  - used by default when you click on a link in a document displayed in your browser and when the browser downloads images for display within an HTML document.
- The POST method is typically used to send information collected from a form displayed within a browser, such as an order-entry form, back to the web server.

# HTTP Methods

---

Method	Requests server to . . .
GET	return the resource specified by the Request-URI as the body of a response message.
POST	pass the body of this request message on as data to be processed by the resource specified by the Request-URI.
HEAD	return the same HTTP header fields that would be returned if a GET method were used, but not return the message body that would be returned to a GET (this provides information about a resource without the communication overhead of transmitting the body of the response, which may be quite large).
OPTIONS	return (in Allow header field) a list of HTTP methods that may be used to access the resource specified by the Request-URI.
PUT	store the body of this message on the server and assign the specified Request-URI to the data stored so that future GET request messages containing this Request-URI will receive this data in their response messages.
DELETE	respond to future HTTP request messages that contain the specified Request-URI with a response indicating that there is no resource associated with this Request-URI.
TRACE	return a copy of the complete HTTP request message, including start line, header fields, and body, received by the server. Used primarily for test purposes.

---

# Header Fields and MIME Types

- Example HTTP Request

```
POST /servlet/EchoHttpRequest HTTP/1.1
host: www.example.org:56789
user-agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.4)
  Gecko/20030624
accept: text/xml,application/xml,application/xhtml+xml,
  text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,
  image/gif;q=0.2,*/*;q=0.1
accept-language: en-us,en;q=0.5
accept-encoding: gzip,deflate
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
connection: keep-alive
keep-alive: 300
content-type: application/x-www-form-urlencoded
content-length: 13
```

# Header Fields and MIME Types

- Field name: field value
- Header names are not case sensitive
- Header field value may wrap onto several lines
- MIME types
  - A standard that can be used to pass a variety of information including graphics and applications through email as well as other internet protocols
- Content-type
  - Top-level type and sub-type
    - Text/html, image/jpeg



# Header Fields and MIME Types

- Standard MIME types

---

Top-level Content Type	Document Content
application	Data that does not fit within another content type and that is intended to be processed by application software, or that is itself an executable binary.
audio	Audio data. Subtype defines audio format.
image	Image data, typically static. Subtype defines image format. Requires appropriate software and hardware in order to be displayed.
message	Another document that represents a MIME-style message. For example, following an HTTP TRACE request message to a server, the server sends a response with a body that is a copy of the HTTP request. The value of the Content-Type header field in the response is message/http.
model	Structured data, generally numeric, representing physical or behavioral models.
multipart	Multiple entities, each with its own header and body.
text	Displayable as text. That is, a human can read this document without the need for special software, although it may be easier to read with the assistance of other software.
video	Animated images, possibly with synchronized sound.

---

# Header Fields and MIME Types

- MIME content types

MIME Type	Description
text/html	HTML document
image/gif	Image represented using Graphics Interchange Format (GIF)
image/jpeg	Image represented using Joint Picture Expert Group (JPEG) format
text/plain	Human-readable text with no embedded formatting information
application/octet-stream	Arbitrary binary data (may be executable)
application/x-www-form-urlencoded	Data sent from a web form to a web server for processing

# Recap

- The Internet
- Basic Internet Protocols
- The World Wide Web
- HTTP Request Message
- HTTP Response Message
- Web Clients
- Web Servers

# Questions?

# Suggestions?

# Homework

# Next

- Web Technologies and Architecture Overview

# References

- A: Web Application Architecture Principles, Protocols and Practices (Leon Shklar and Rich Rosen)
- B: Web Technologies (Jeffrey Jackson)
- Wikipedia