CS601: Principles of Software Development

Web Architecture. HTTP. HTTPS. HTML.

Olga A. Karpenko

Parts of this presentation is based on the materials of Prof. Engle.

Announcements

- Office hours today start at 4:30
- Code Camp: Wednesday, Harney 411
  - 3:30-4:30 (for students who have cs673 at 4:45)
  - 4:30-5:30 (for everybody else)
  - Please bring scratch paper
- Midterm on Friday, in class

Internet

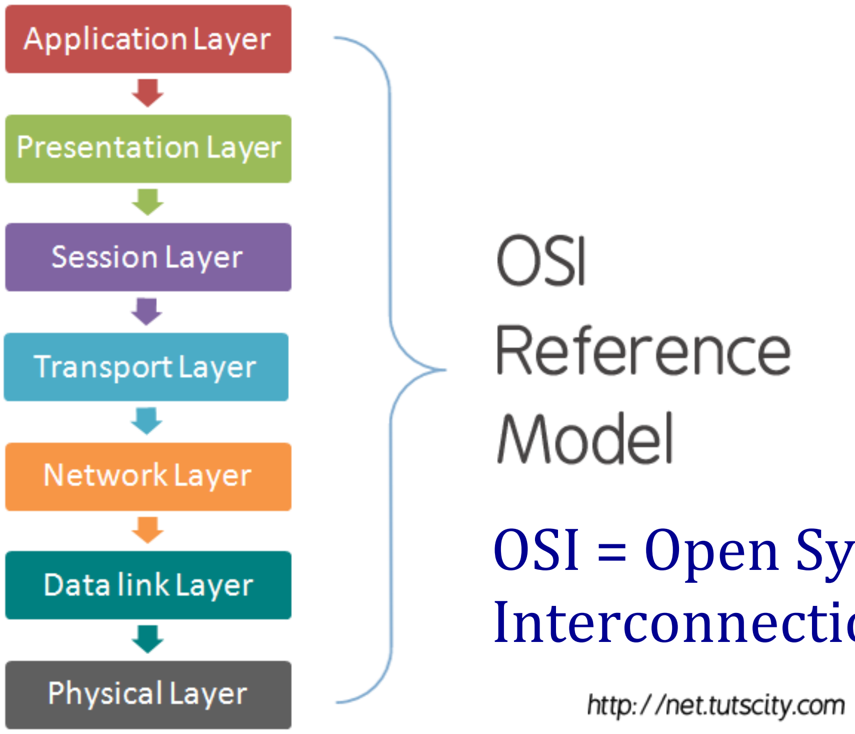
- Global network of networks
- Hardware + Software
  - servers, switches, routers, etc.
  - protocols etc.

World Wide Web

- An application that runs on the Internet
- A collection of websites "connected" via hyperlinks
- Each website:
  - is identified by URL
  - Accessed using HTTP
  - Content in HTML
- Other applications run on Internet besides Web

OSI Reference Model

- Describes how the network systems are supposed to communicate with each other



Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data link Layer

Physical Layer

OSI Reference Model

OSI = Open Systems Interconnection Model

http://net.tutscity.com

http://net.tutscity.com/wp-content/uploads/2011/01/OSI-reference-model.png

OSI Reference Model

- Application layer
  - Network applications: WWW, File transfer, Email etc..

OSI Reference Model

- Presentation layer
  - Prepares data for the application layer
    - Encryption/decryption, compression etc...

https://www.youtube.com/watch?v=sVDwG2RdJho

OSI Reference Model

- Session layer
  - Determines how two devices
    - Establish, maintain and manage a connection
    - Ex: a session between your browser and the bank server
  - Unique **session id** for each session
    - data streams from different clients will be separate

https://www.youtube.com/watch?v=sVDwG2RdJho

OSI Reference Model

- Transport layer
  - Breaks data into segments
  - Establishes logical connection between computers
  - Congestion/Flow control – regulates the number of packets sent
  - Protocols
    - TCP (reliable connections)
    - UDP (unreliable but less latency)

https://www.youtube.com/watch?v=sVDwG2RdJho

http://www.bpsharma.in/eLearning/Networking/OSI_Reference_Model.htm

OSI Reference Model

- Network layer
  - Provides connections between hosts on dif. networks
  - Determines how to route data
    - IP protocol
  - Routers operate on this level
  - Data at this level is called “packets”

https://www.youtube.com/watch?v=sVDwG2RdJho

OSI Reference Model

- Data Link layer
  - Connections between hosts on the same network
  - Ethernet

https://www.youtube.com/watch?v=sVDwG2RdJho

OSI Reference Model

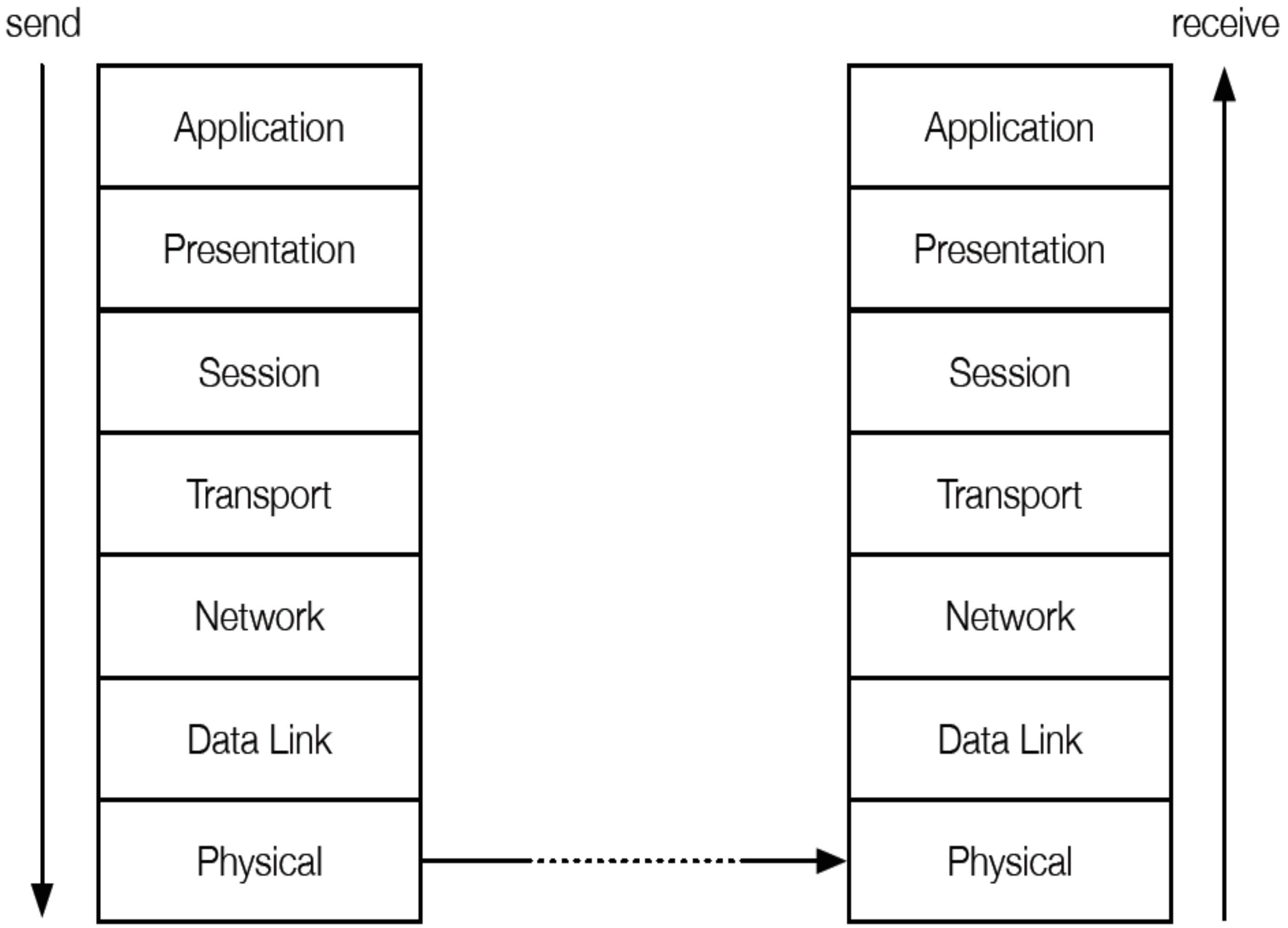
- Physical layer specifies
  - How data is processed into bits
  - How data is physically transferred over cables, etc..

https://www.youtube.com/watch?v=sVDwG2RdJho

Remembering Layers

- A-P-S-T-N-D-P
- "All People Seem to Need Data Processing"
- Backwards: "Programmers do not throw sausage pizza away"

OSI Reference Model



send

Application

Presentation

Session

Transport

Network

Data Link

Physical

receive

Application

Presentation

Session

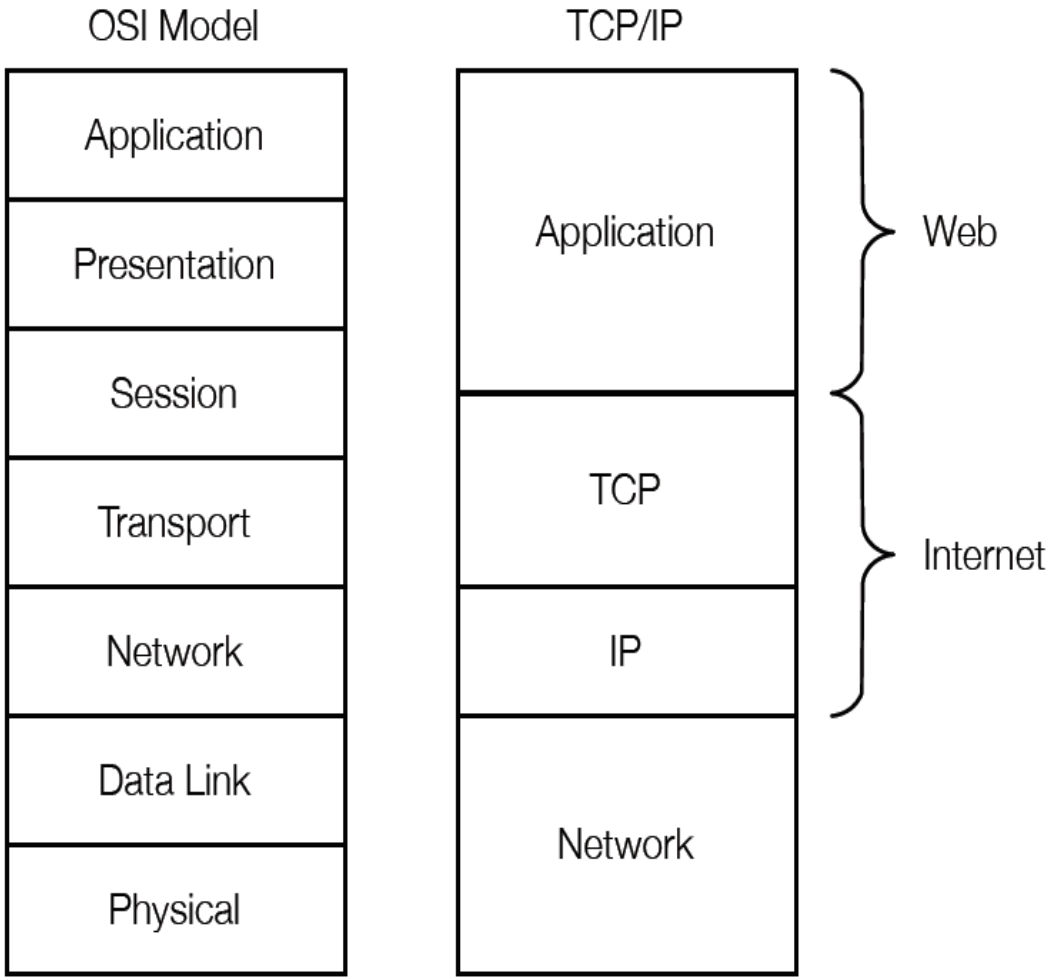
Transport

Network

Data Link

Physical

TCP/IP Network Model



OSI Model

Application

Presentation

Session

Transport

Network

Data Link

Physical

TCP/IP

Application

TCP

IP

Network

Web

Internet

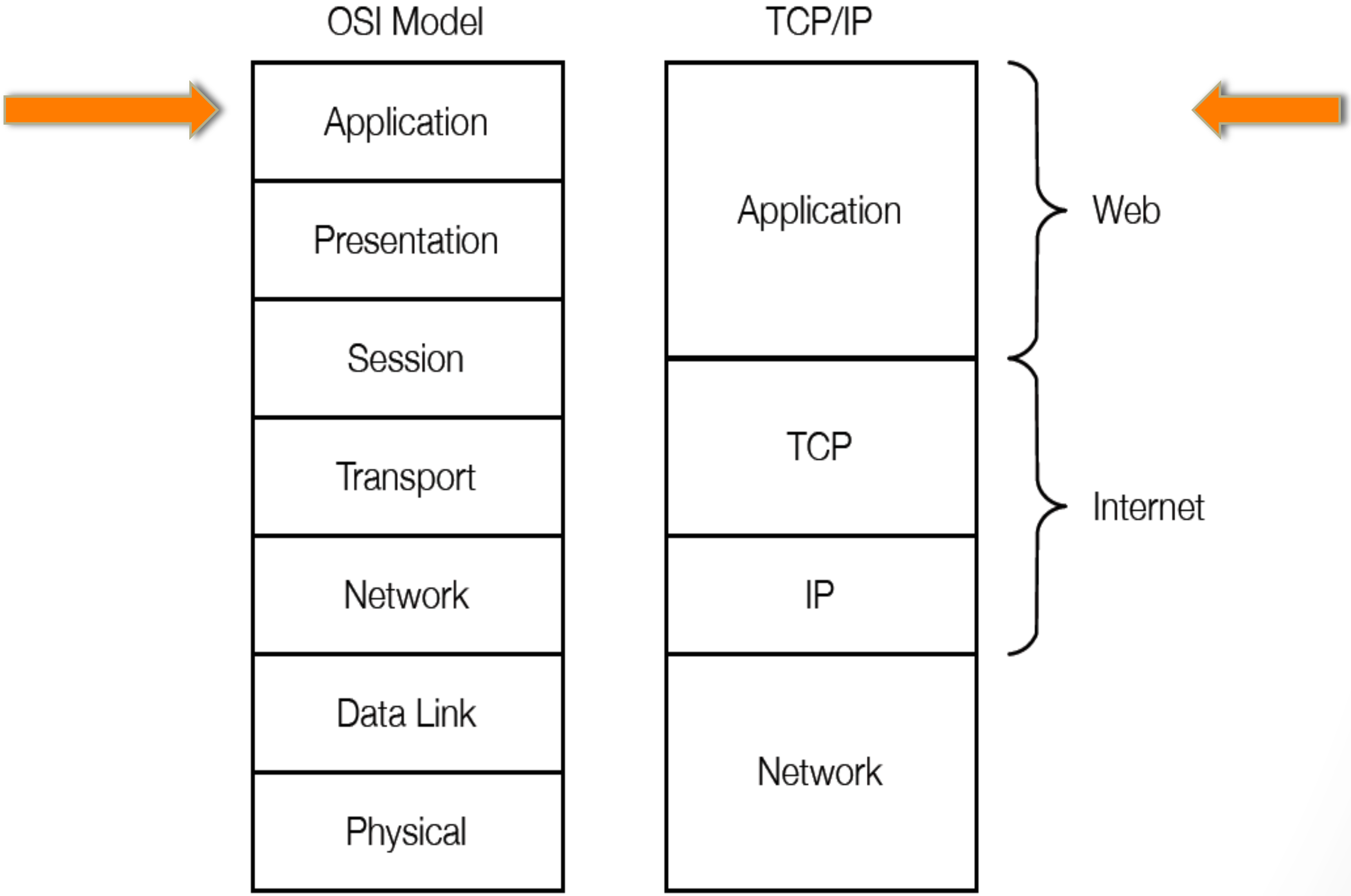
Client-server model

- Standard model for developing network applications
- Server: process that is trying to provide service to clients
  - Waits for requests from clients
- Client: requesting some service
- May run on different machines

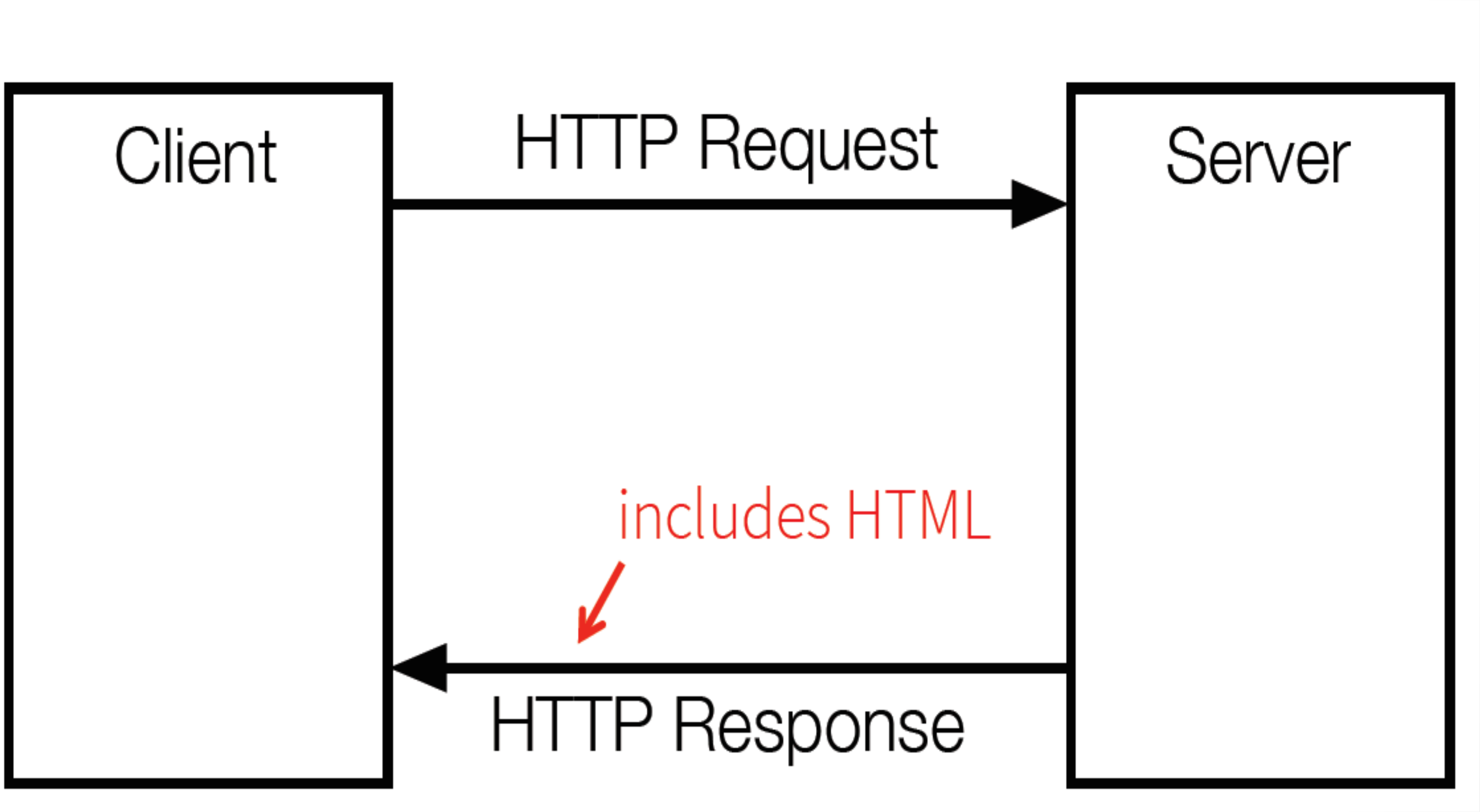
HTTP

- Stands for Hypertext Transfer Protocol
- Webpages are transferred from server to browser using HTTP

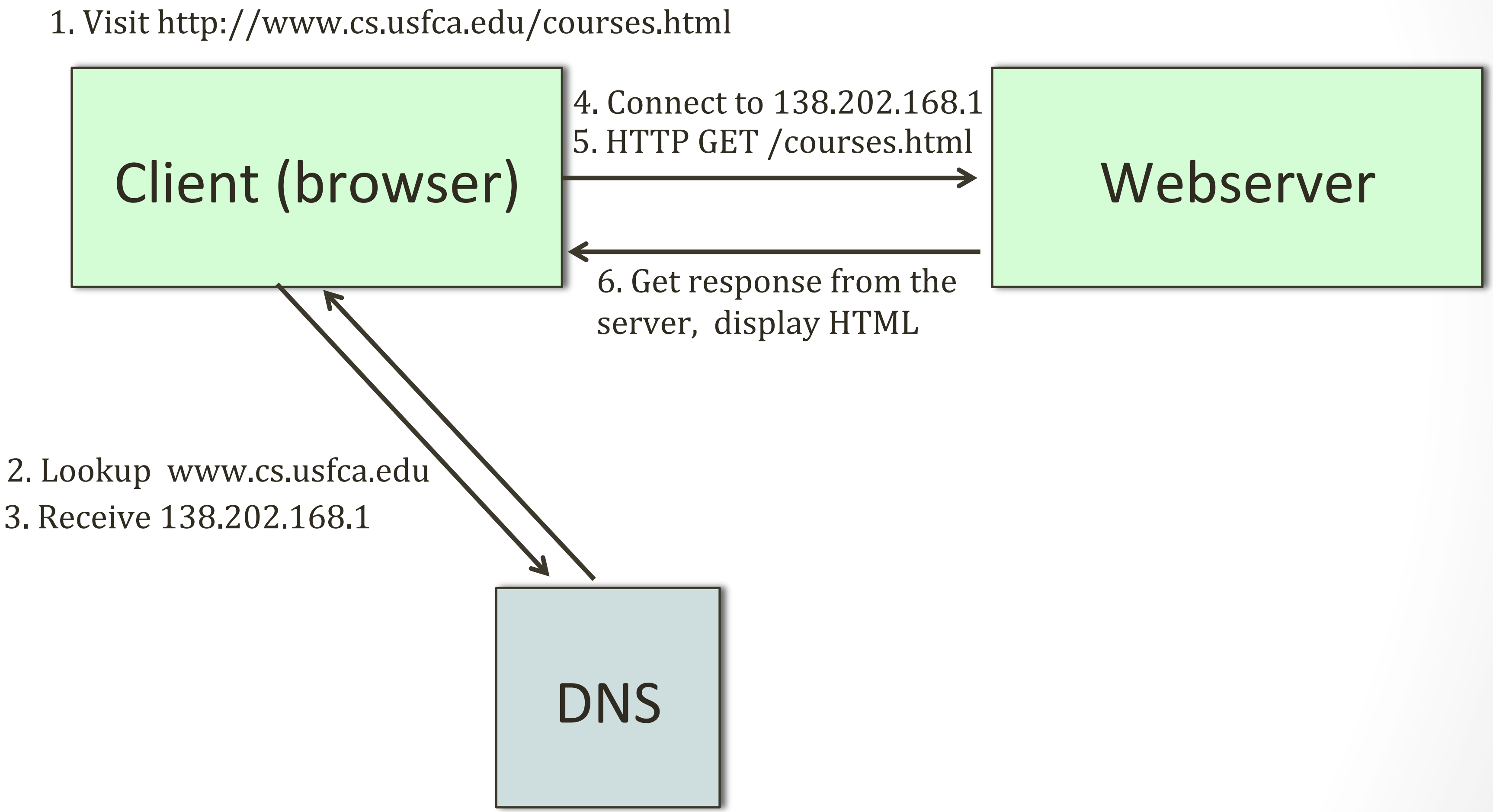
Application-level Protocol



Basic Client/Server Architecture



Basic Client/Server Architecture



1. Visit http://www.cs.usfca.edu/courses.html

Client (browser)

4. Connect to 138.202.168.1
5. HTTP GET /courses.html

Webserver

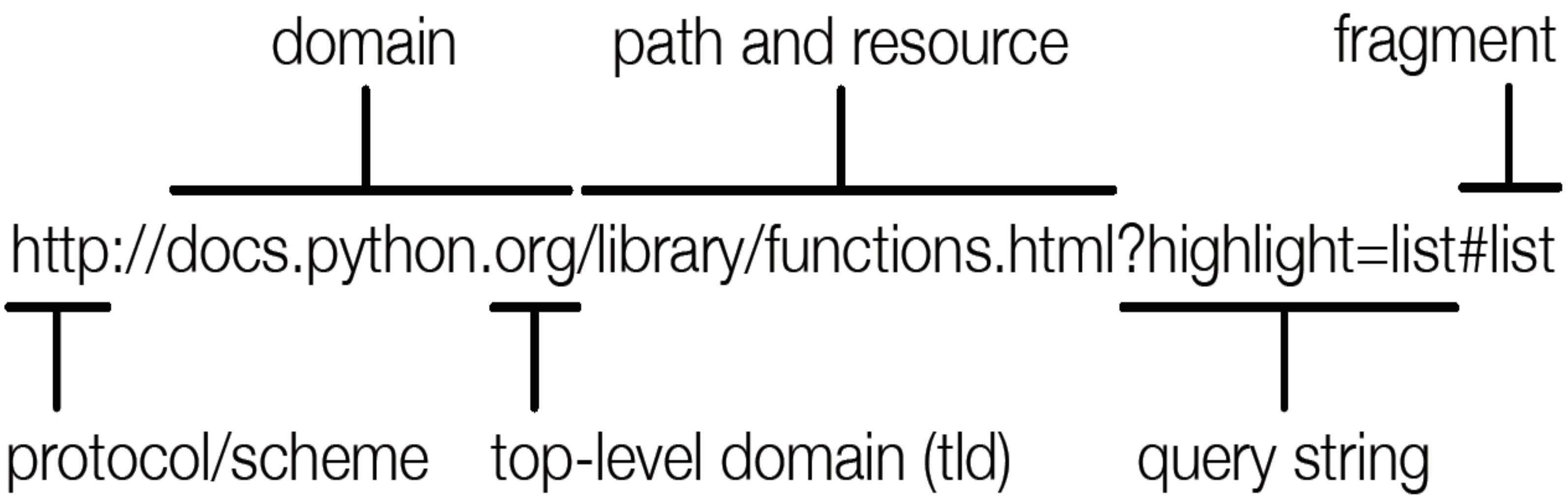
6. Get response from the server, display HTML

2. Lookup www.cs.usfca.edu
3. Receive 138.202.168.1

DNS

Diagram is based on the materials of Prof. Rollins.

Uniform Resource Locator



domain

path and resource

fragment

http://docs.python.org/library/functions.html?highlight=list#list

protocol/scheme

top-level domain (tld)

query string

Uniform Resource Locator

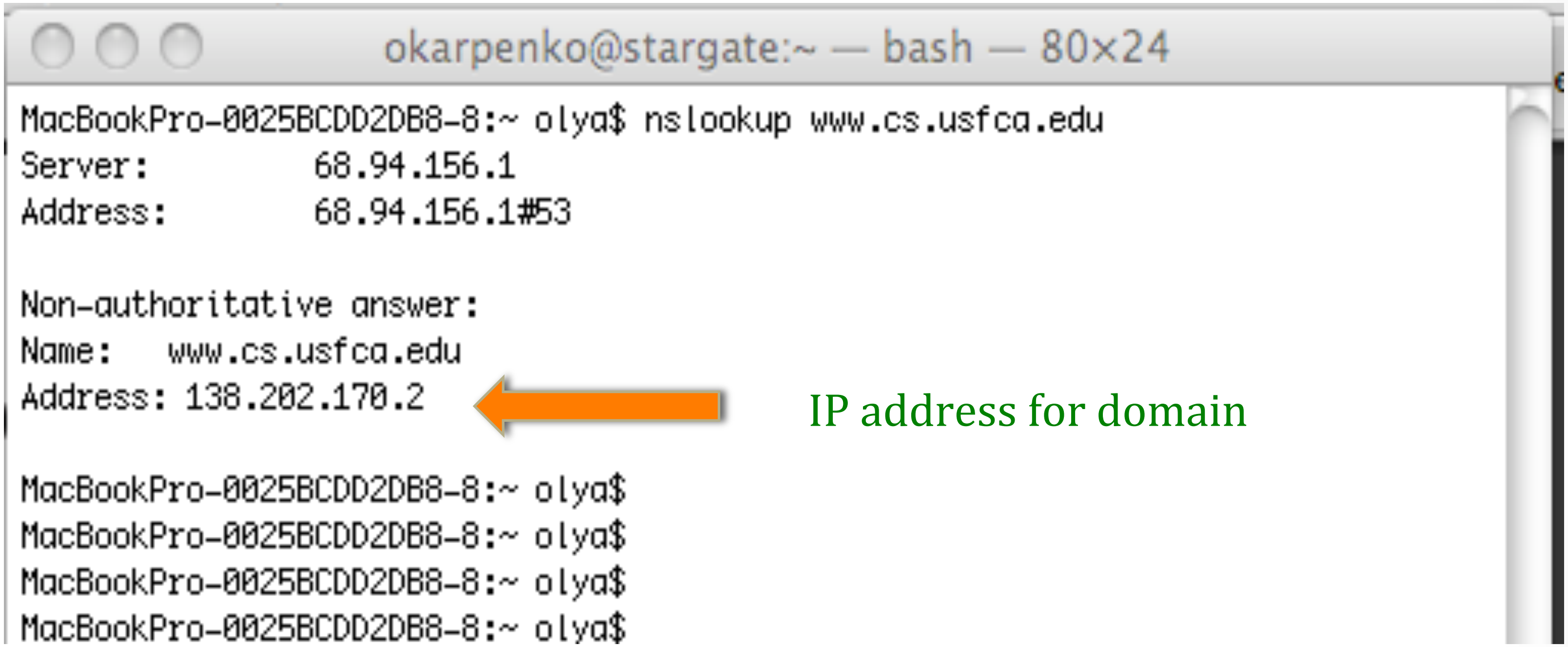
- Required
  - – Protocol used to transfer resource
  - – Domain (gets converted to IP address)
  - – Path and resource converted to actual file on webserver (defaults to / if missing)
- Optional
  - – Query string passes data via GET to webserver
  - – Fragment marks specific location on page

URI, URL, URN

- URI (Uniform Resource Identifier) - a string that identifies a resource
  - abstract or physical
  - Can be a locator, a name or both
  - Ex: `files.hp.com` – may respond to dif. protocols
- URL – A type of URI, Identifier + Locator
  - Ex: `http://www.cs.usfca.edu/courses.html`
- URN – unique name

https://danielmiessler.com/study/url-uri/

URL



```
MacBookPro-0025BCDD2DB8-8:~ olya$ nslookup www.cs.usfca.edu
Server:         68.94.156.1
Address:        68.94.156.1#53

Non-authoritative answer:
Name:   www.cs.usfca.edu
Address: 138.202.170.2

MacBookPro-0025BCDD2DB8-8:~ olya$
MacBookPro-0025BCDD2DB8-8:~ olya$
MacBookPro-0025BCDD2DB8-8:~ olya$
MacBookPro-0025BCDD2DB8-8:~ olya$
```

IP address for domain

IP address

- A 32-bit number
- Assigned to each device participating in a computer network
    - That uses IP protocol for communication
- Two functions
    - Host identification
    - Location addressing
- Can use it in the browser

DNS

- Domain Name System
  - DNS maps a name ->an IP address
- DNS Lookup:
  - Browser cache
  - OS cache
  - Router cache
  - ISP DNS Server Cache
  - Recursive search

HTTP Requests

- Use services of TCP on port 80
- Versions
  - HTTP/1.1 and HTTP/1.0
- Request Types
  - GET
  - POST
  - HEAD

…

GET Request

```
GET /~okarpenko/index.html HTTP/1.1
Host: www.cs.usfca.edu
Connection: close
```

HEAD Request

- Gets the header of the file (not the document)

```
HEAD /~okarpenko/index.html HTTP/1.1
Host: www.cs.usfca.edu
```

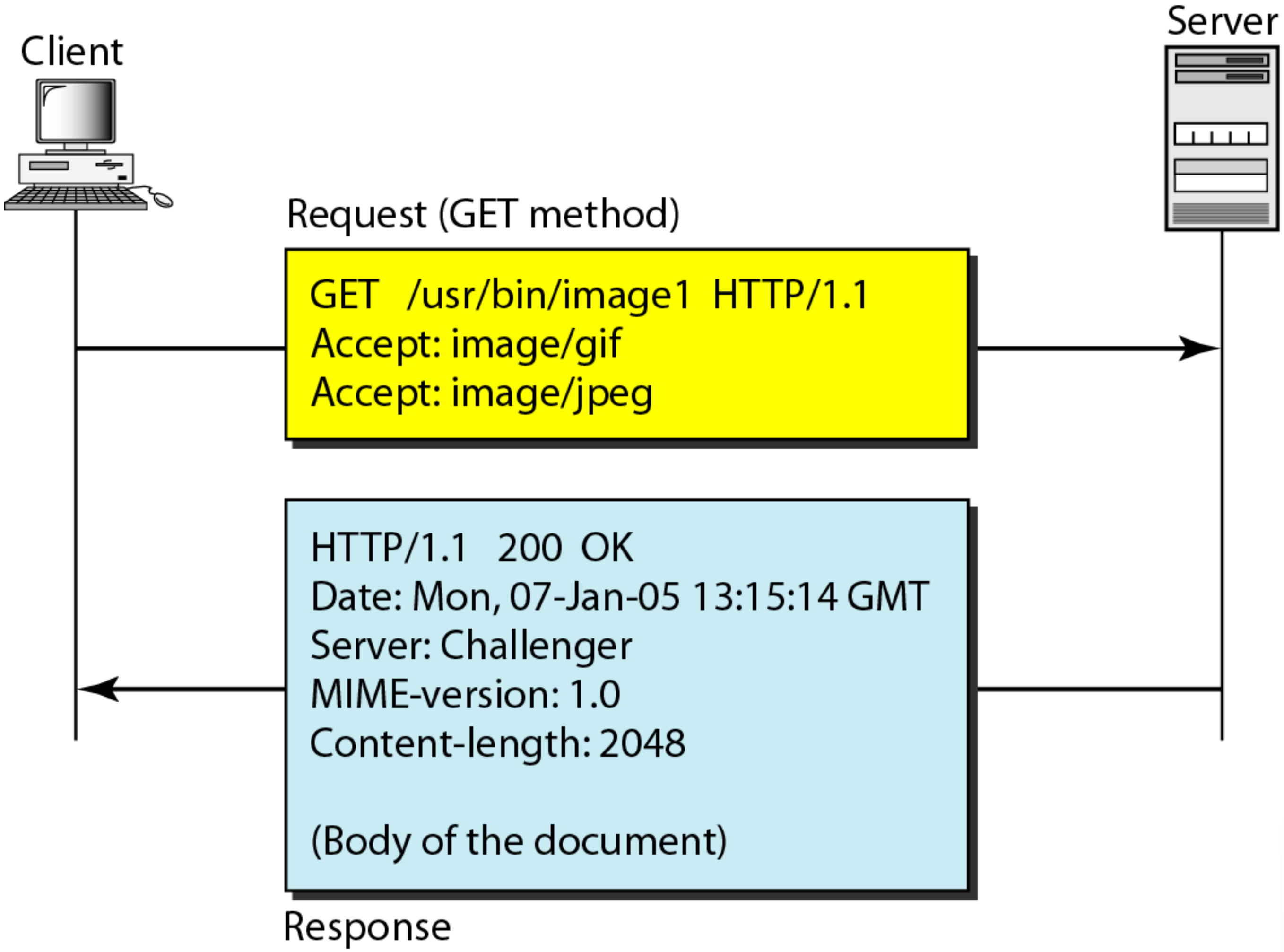
Example

- Use telnet to connect to a remote server
- Type HTTP requests and see the responses

Status Codes

200	OK
202	Accepted
204	No content
301	Move permanently
400	Bad request
401	Service unauthorized
404	Not found
500	Internal server error
…	

HTTP Request / Response



Client

Server

Request (GET method)

```
GET   /usr/bin/image1  HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

```
HTTP/1.1   200  OK
Date: Mon, 07-Jan-05 13:15:14 GMT
Server: Challenger
MIME-version: 1.0
Content-length: 2048

(Body of the document)
```

Response

Submitting Data

- Both GET and POST can be used
- GET method
- – Encodes data in URL as query string
- E.g. http://www.google.com/search?q=usfca
  - Good for getting or retrieving data
- POST method
  - Encodes data in request message body
  - Good for sending data to be stored
  - Used often for surveys, login forms, etc.

URL Class

- In java.net package

```
URL myURL = new URL("http://cnn.com/politics/
index.html?search=elections");
```

URL Class

- **getProtocol**
  Returns the protocol identifier component of the URL
- **getHost**
  Returns the host name component of the URL
- **getPort**
  Returns the port number component of the URL.
- **getPath**
  Returns the path component of this URL.
- **getQuery**
  Returns the query component of this URL

…

http://docs.oracle.com/javase/tutorial/networking/urls/urlInfo.html

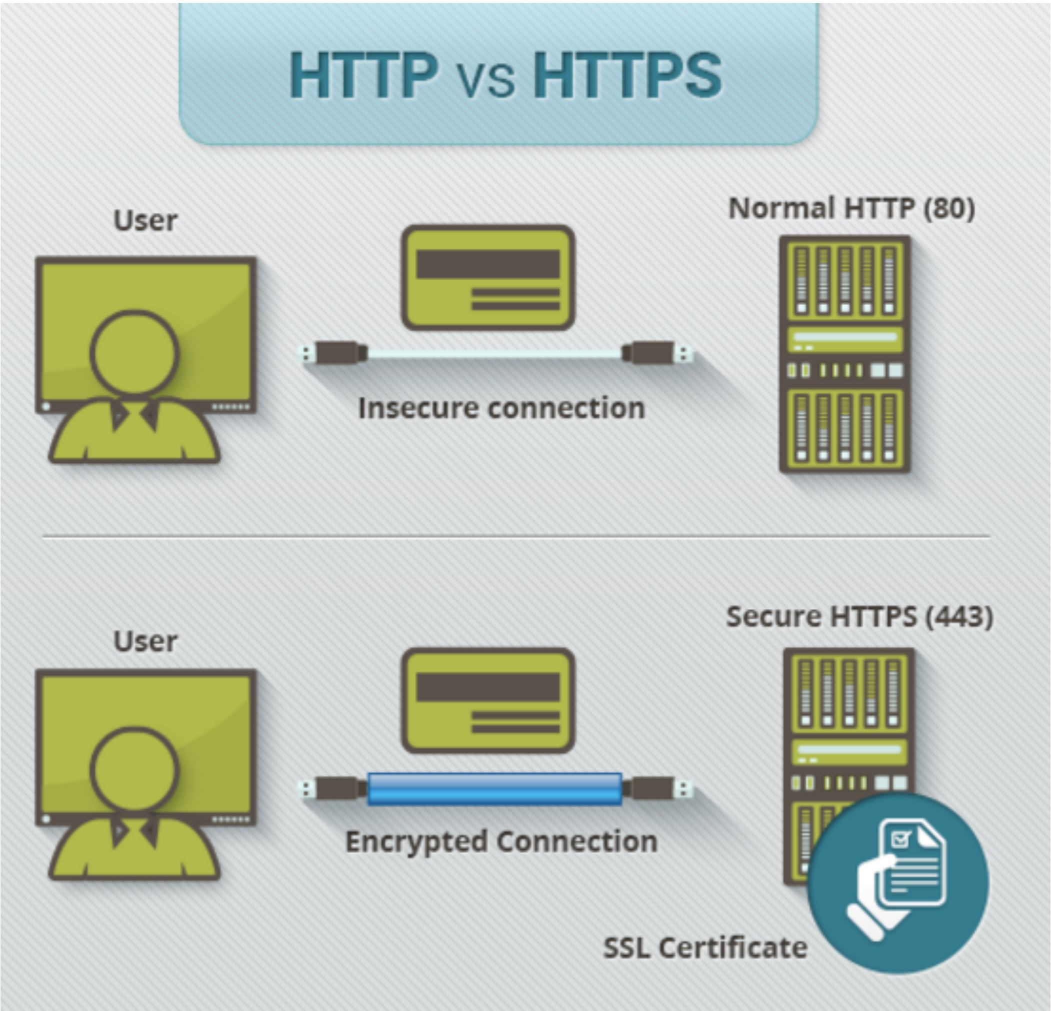
HttpFetcher Class

- Fetches a given page from the given URL
- See HttpFetcher.java

HTTPS

- A secure version of HTTP
- All communications between your browser and the webserver are encrypted
- HTTP + SSL (secure socket layer)

HTTPS



https://www.instantssl.com/ssl-certificate-products/https.html

SSL

- SSL protocol – security protocol
  - runs above TCP/IP, below HTTP
- Provides
  - Encryption (hiding what is sent)
  - Identification (making sure you trust the computer you are "talking to")

https://www.sslshopper.com/what-is-ssl.html

Port 443

- Used for websites that use SSL
  - Whenever you go to https:// you connect to port 443

SSLSocket Class in Java

- Provides a secure socket using SSL (or IETF) protocol
- Extends class Socket
  - Adds a layer for security protection (encryption/authentication)

Example

- Accessing Google's Geocode API

https://maps.googleapis.com/maps/api/geocode/json?address=University%20of%20San%20Francisco,%20US

- See WebClientSSL.java

HTML: An introduction

Terminology

- Hypertext Markup Language (HTML)
  - Describes the structure of web pages
  - Set of markup tags
  - Example:

```
<head><title>My Webpage</title></head>
```

- Cascading Style Sheet (CSS)
  - Describes the presentation of web pages
    - Color, layout, font
  - Contains no information, only style
  - Separation of content from style

HTML tags

- Keywords / tag names surrounded by **angle brackets**
- Normally **come in pairs** like <b> and </b>
- **start tag** and **end tag**
  - end tag: like the start tag, with a **forward slash** before the tag name

Html Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My Webpage</h1>

<p>This is a paragraph.</p>

<a href="http://www.cs.usfca.edu">Go USF!</a>

</body>
</html>
```

Resources: Tutorials

- W3 Schools HTML Tutorials

http://www.w3schools.com/html/default.asp

- W3 Schools CSS Tutorials

http://www.w3schools.com/css/default.asp

- http://reference.sitepoint.com/css