Csci 4131
Internet Programming:
TCP/IP
HTTP

Lecture 12, October 15th

Fall 18

Dr. Dan Challou

Logistics

- HW 4 – building a small HTTP server using Python is out: Due date is Friday, **October 26th** at **2:00 PM**
- You'll need to understand the protocol – which both the readings (on the schedule and in the assignment) and lecture will cover
- Readings on the class schedule (from last week and this week):
  - http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
  - https://tools.ietf.org/html/rfc2616
  - Additional
  - http://www.w3c.org/Protocols/

Last Time

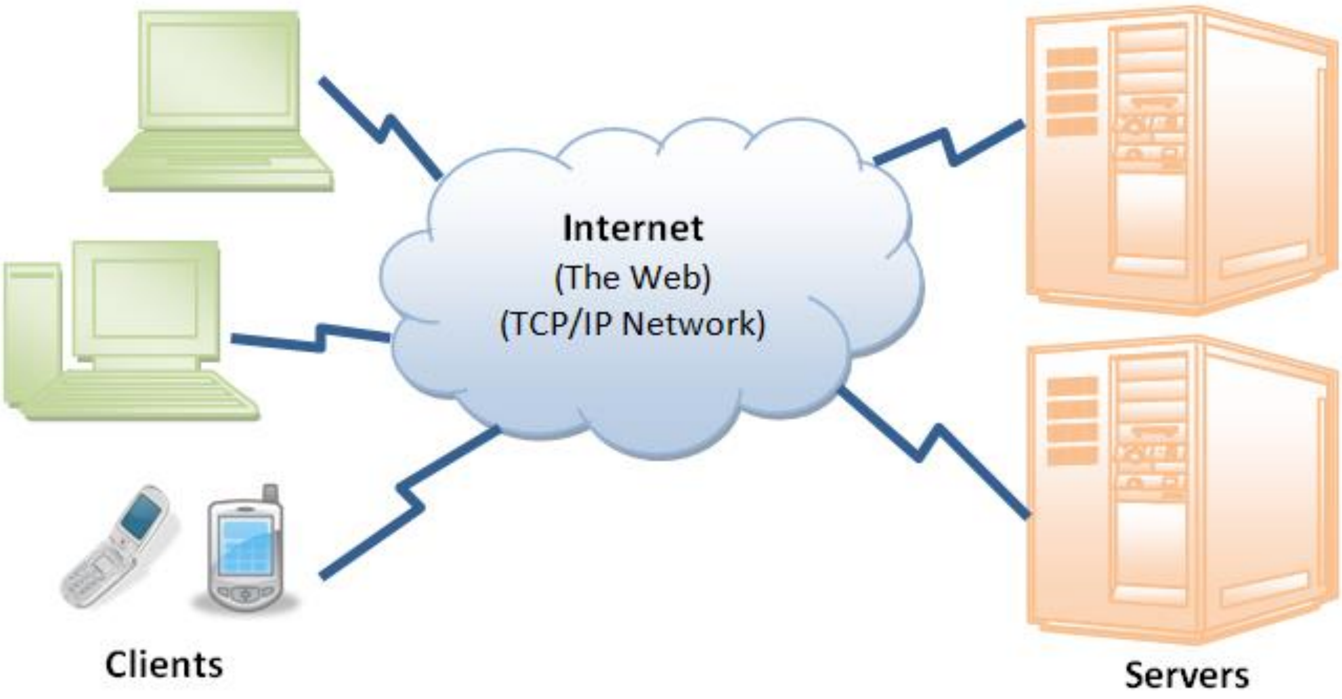
- Wrapped up a detailed look at DOM events and interaction with JavaScript
- Started discussion on another Key aspect/technology of the WWW:
  - Revisted URLs (and URIs and URNs)
  - Protocols and Servers (high level)
  - The HTTP Protocol

Questions?

Today

- HTTP Message overview revisited
- TCP-IP
- HTTP, and HW 4

The Web



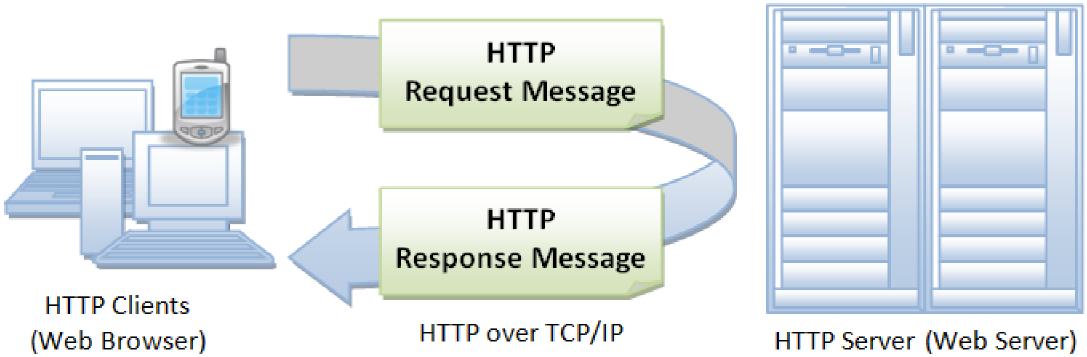
Massive distributed Client/Server Information System

HTTP – how info moves over the web

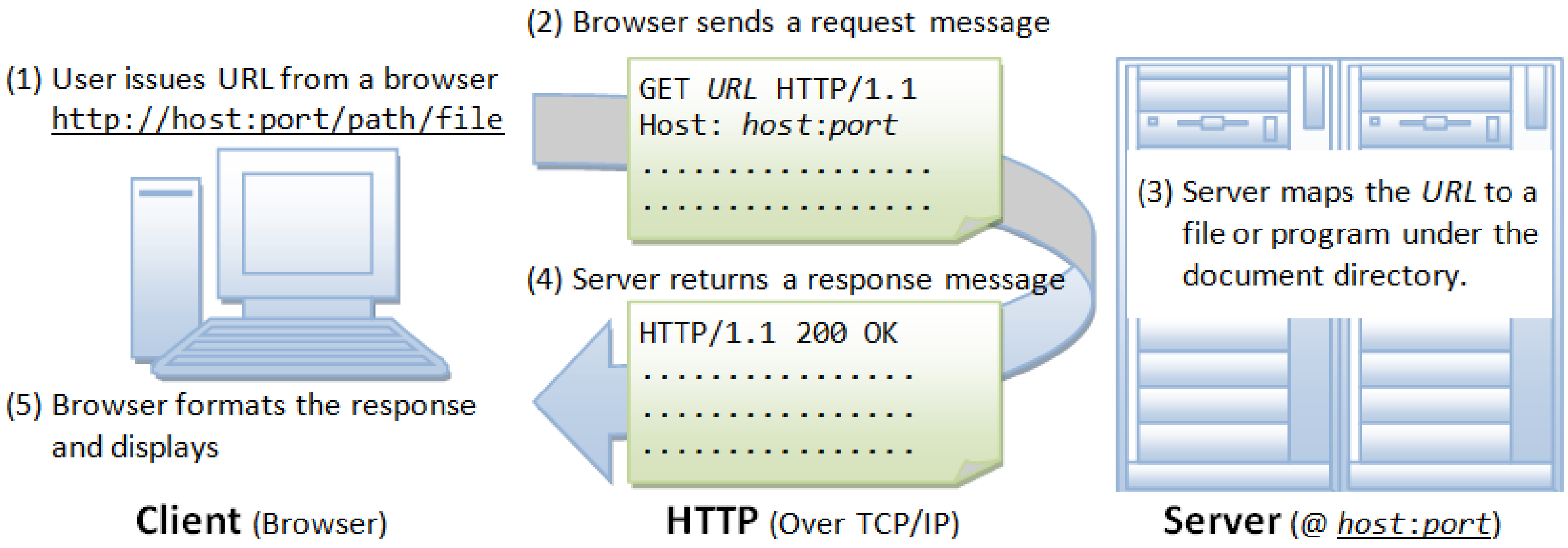
- HTTP (Hypertext Transfer Protocol) is perhaps the most popular application protocol used in the Internet (or The WEB).

- HTTP is an *asymmetric request-response client-server* protocol as illustrated.
  - An HTTP client sends a request message to an HTTP server.
  - The server, in turn, returns a response message.

- HTTP is a *pull protocol*: the client *pulls* information from the server (instead of server *pushing* information to the client).



How the HTTP Procol Works



(1) User issues URL from a browser
http://host:port/path/file

(2) Browser sends a request message

```
GET URL HTTP/1.1
Host: host:port
..................
..................
```

(3) Server maps the *URL* to a file or program under the document directory.

(4) Server returns a response message

```
HTTP/1.1 200 OK
.................
.................
.................
```

(5) Browser formats the response and displays

Client (Browser)

HTTP (Over TCP/IP)

Server (@ *host*:*port*)

HTTP Protocol, Revisited, in detail

HyperText Transfer Protocol is a simple protocol for request/reply based interactions between a client and a server.

This serves as the foundation of the World Wide Web.

- Request messages contain a very small number methods that can be invoked on the server.
  - It is extensible so new methods can be added.
  - Parameter data can be included in a request.

- Response messages composed and sent by an HTTP server to a client indicate the status of the request processing and possibly some response data.

Details of the HTTP Protocol

- When you enter a URL in the address box of the browser, the browser translates the URL into a request message according to the specified protocol; and sends the request message to the server.

- For example, the browser translated the URL:

  http://www.test101.com/docs/index.html

  into the following request message:

```
GET /docs/index.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

HTTP Protocol Continued

- When the previous request message reaches the server, the server can take one of these three actions:

1. The server interprets the request received, maps the request into a *file* under the server's documents directory, and returns the file requested to the client.

2. The server interprets the request received, maps the request into a *program* kept in the server, executes the program, and returns the output of the program to the client.

   (What examples have we seen of this???)

3. The request cannot be satisfied, the server returns an error message.

Assume the following

- On the host:
  - www.test101.com
- In the file:
  - /docs/index.html

```
<!DOCTYPE html>
<html>
  <body>
      <h1>It works!</h1>
  </body>
</html>
```

Example of an HTTP response message:

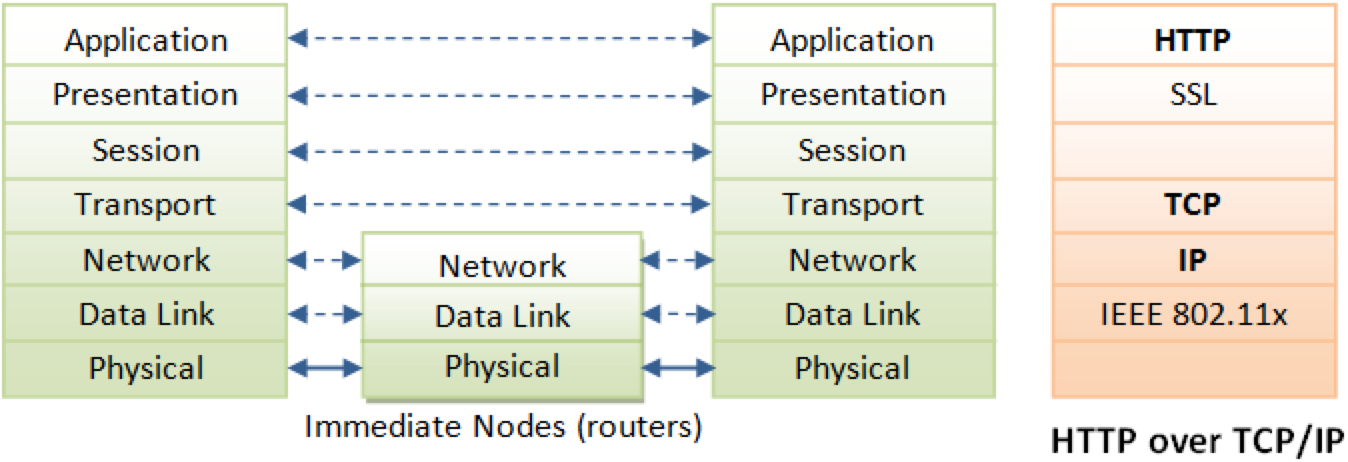
```
HTTP/1.1 200 OK
Date: Sun, 18 Oct 2009 08:56:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Sat, 20 Nov 2004 07:16:26 GMT
ETag: "10000000565a5-2c-3e94b66c2e680"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html
X-Pad: avoid browser bug

<html><body><h1>It works!</h1></body></html>
```

HTTP Protocol, Continued

- The browser receives the response message, interprets the message and displays the contents of the message on the browser's window according to the media type of the response (as in the Content-Type response header).
- Common media type include "text/plain", "text/html", "image/gif", "image/jpeg", "audio/mpeg", "video/mpeg", "application/msword", and "application/pdf".
- In its idling state, an HTTP server simply listens for IP address(es) and port(s) specified in the configuration for incoming request.
- When a request arrives, the server analyzes the message header, applies rules specified in the configuration, and takes the appropriate action.
- The webmaster's main control over the action of web server is via its configuration.

HTTP over TCP/IP

- HTTP is a client-server application-level protocol.
- It typically runs over a TCP/IP connection, as illustrated below.



ISO OSI 7-layer network

HTTP over TCP/IP

- (HTTP need not run on TCP/IP. It only presumes a reliable transport. Any transport protocols that provide such guarantees can be used.)

TCP/IP

- TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of transport and network-layer protocols for machines to communicate with each other over the network.

- TCP (Transmission Control Protocol) is a transport-layer protocol, responsible for establishing a connection between two machines.

- TCP consists of 2 protocols: TCP and UDP (User Datagram Package).

- TCP is *reliable*: each packet has a sequence number, and an acknowledgement is expected. A packet will be re-transmitted if it is not received by the receiver. Packet delivery is guaranteed in TCP.

- UDP does not guarantee packet delivery => not reliable.
  - But, UDP has less network overhead and can be used for applications such as video and audio streaming, when reliability is not as critical.

What does this say?

- Teh quik brwon fox jumes over teh lzay dog

- Similarly, you will still get the meaning of an audio or video transmission if blips like this occur during the transmission

- You will notice the lack of quality in the transmission however

TCP / **IP** Continued

- IP (Internet Protocol) is a network-layer protocol (network addressing and routing).
- In an IP network, each machine is assigned an unique IP address (e.g., 165.1.2.3), and the IP software is responsible for routing a message from the source IP to the destination IP.
- In IPv4 (IP version 4), the IP address consists of 4 bytes, each ranges from 0 to 255, separated by dots, which is called a *quad-dotted form*. This numbering scheme supports up to 4G addresses on the network. IPv6 supports more addresses (16 bytes worth)
- Since memorizing a 12 digit number is difficult for most of the people, an english-like domain name, such as www.test101.com is used instead.
- The DNS (Domain Name Service) translates the domain name into the IP address (via distributed lookup tables).
- A special IP address 127.0.0.1 always refers to your own machine. It's domain name is "localhost" and can be used for *local loopback testing*

TCP / IP Continued

- TCP *multiplexes* applications within an IP machine.
- For each IP machine, TCP supports (multiplexes) up to 65536 ports (or sockets), from port number 0 to 65535.
- An application, such as HTTP or FTP, runs (or listens) at a particular port number for incoming requests.
- Port 0 to 1023 are pre-assigned to popular protocols:
  - HTTP at 80,
  - FTP at 21,
  - Telnet at 23,
  - SMTP (Simple Mail Transfer Protocol) at 25,
  - NNTP (Nework News Transfer Protocol) at 119,
  - and DNS at 53.
  - Port 1024 and above are available to the users.

TCP / IP Final Thoughts / Summary

- TCP port 80 is pre-assigned to HTTP, as the default HTTP port number.

- This does not prohibit you from running an HTTP server at other user-assigned port number (1024-65535) such as 8000, 8080 (e.g., when testing a new server).

- You can also run multiple HTTP servers in the same machine on different port numbers.

- When a client issues a URL without explicitly stating the port number, the browser will connect to the default port number 80 of the host.

- The request from the client needs to explicitly specify the port number in the URL if they want to request something from a server that is not listening to the default url.

  e.g. http://www.test101.com:8000/docs/index.html will request
  docs/index.html
  from the server listening at port 8000

- **SUMMARY:**

  To communicate over TCP/IP, you need to know:

  (a) IP address or hostname, (b) Port number.

HW Assignment 4

- Build a simple HTTP server in python
- I’ll go over Python 3 code that essentially provides the transport layer in the 7 layer ISO protocol and post it to the class Moodle page, along with these lecture slides

Idea

- Build a limited-functionality HTTP server by layering the functionality to deal with the HTTP protocol on top of the presentation layer server
- Your server will have to receive request messages (GET, HEAD, POST)
  - Figure out what resource to find and return to the requesting client
  - Find the Resource
  - Return the resource (Compose a Response Message)
- OR
  - Return an error! (In a Response Message)

Lets have a look at Echo Client and Echo Server – in Python!

- ISO-OSI Level 6 Programs – socket level
- HTTP is an ISO-OSI Level 7 protocol, the application layer
- It is built on level 6 (and below) applications
- You are free to use this code to construct the Web Server you are building for HW 4

Next Class – The Echo Client

- Will post both, along with the slides, on Moodle – please run them and review them