

Csci 4131 Internet Programming

Fall 2021

Lecture 10

October 11th

Instructor: Dr. C

Logistics – Csci 4131 Lecture 10, Oct 11th

- **HW 4 is out – and is due Friday 10/22**

Reading/Tutorials – This Week

- HTTP Protocol
 - https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
 - **And see course Schedule on Canvas (In the Resources Module)**

Upcoming

Necessary Preparations for enabling your understanding of HTTP:

Getting Setup for Python 3.x – which you will use to do HW4

- <https://www.python.org> – to download python to your machine so you can develop and run HW Assignment 4 – note, our target machines are ubuntu, and our examples have been developed and tested on the cse labs machines. That is where we will be testing **and GRADING** your programs.
- <https://docs.python.org/3/> - documentation on newest version of Python
- <https://docs.python.org/3/howto/sockets.html>
- <https://docs.python.org/3/howto/sockets.html>
- - documentation on the socket library – which will help you understand EchoClient and EchoServer python programs (which I'll post on Canvas for your review and refactoring for HW Assignment 4)

Learning or Refreshing your Python

- <https://docs.python.org/3/tutorial/>
- <https://www.learnpython.org/>
- <https://www.linkedin.com/learning/topics/python?u=42740356>
- (need your x500 and umn password to login to linked in learning – lots of Python Tutorials there)

Upcoming: Node.js

Introduction to Node.js (Building a Webserver in JavaScript)

<https://www.w3schools.com/nodejs/>

<https://codeburst.io/the-only-nodejs-introduction-youll-ever-need-d969a47ef219>

And, we there will by a zybooks Prep and/or Homework on Node.js

Questions?

Review Lecture 9, Exercise 1:

- Refactor (fix/rewrite) the HTML/JavaScript below so the text:

Hello World

is displayed in the text box when the browser loads the page

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Race Condition</title>
  <script language=javascript>
    var textobj = document.getElementById("stuff");
    textobj.value = "Hello World";
  </script>
</head>
<body>
  <input type="text" id="stuff"/>
</body>
</html>
```

See file rfix.html on Canvas for one possible answer!

Questions?

Recall URL's – used by HTTP Protocol

- Recall, a URL has the following syntax:

protocol://hostname:port/path-and-file-name

- There are 4 parts in a URL:
 - *Protocol*: The application-level protocol used by the client and server, e.g., **HTTP**, FTP, and telnet.
 - *Hostname*: The DNS domain name (e.g., www.test101.com) or IP address (e.g., 192.128.1.2) of the server.
 - *Port*: The TCP port number that the server is listening for incoming requests from the clients.
 - *Path-and-file-name*: The name and location of the requested resource, under the server document base directory.

URL Port Numbers

- Recall, a URL has the following syntax:

*protocol://hostname:**port**/path-and-file-name*

BUT the Port number can be omitted if the server is running on the default port for that service.

- For example,:

HTTP servers (port 80), (also default port for localhost)

HTTPS (port 443),

FTP (port 21),

Telnet (port 23).

- Protocol can be:

–http, telnet, ftp, mailto, others

URIs, URL's, URN's

- [https://wiki.digitalclassicist.org/URIs URNs and URLs: What's the difference?](https://wiki.digitalclassicist.org/URIs%20URNs%20and%20URLs%3F)

Lecture 10, Exercise 1: **Submit your answers to Lecture 10, Exercise 1 Submission item in week 6**

Specify which of the following items below is a URI, URL, *and/or* URN (URN is defined as Universal Resource Name).

- a) isbn:0451450523
- b) <http://www.cs.umn.edu/academics/classes.php>
- c) telnet 195.0.1.12:80
- d) <mailto:Jim.Smith@acme.com>
- e) tel: 19146286389

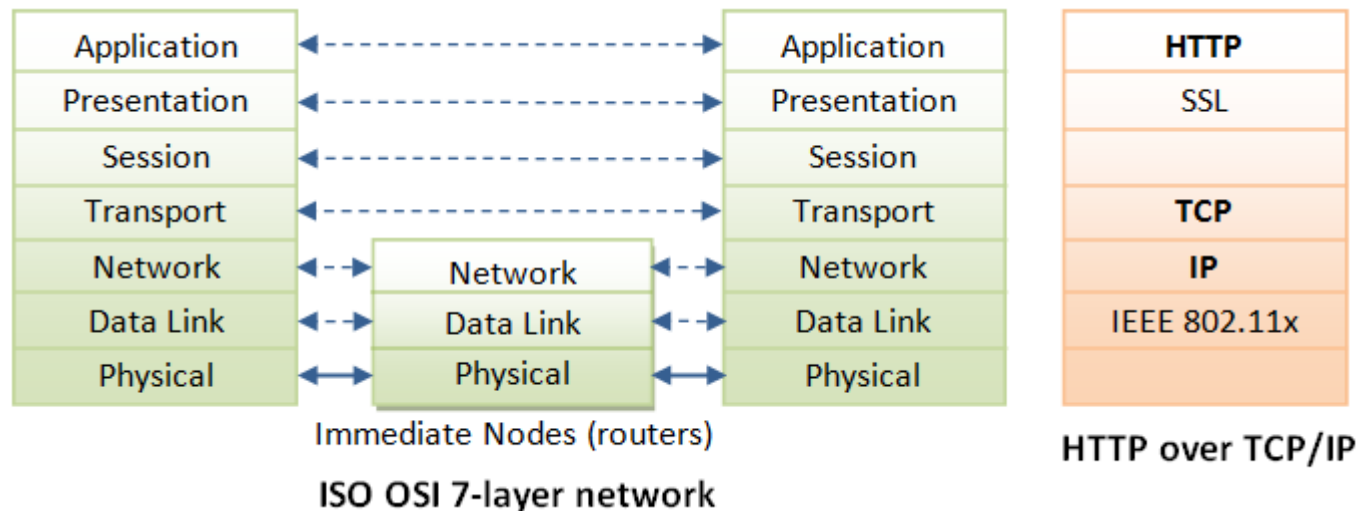
PLEASE RAISE HAND AND CLOSE COMPTUER (slightly) WHEN DONE...

Holy Buckets! Enough with URLs Already!!!

- **What's the point ?**
 - URLs specify the exact location of every host (running SOFTWARE CLIENT AND/OR SERVER PROGRAMS) on the Internet Network, *and*
 - Are used by the HTTP Protocol and other protocols to function!

HTTP relies on protocols below it to function - which are usually built on TCP/IP

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



- ***(HTTP need not run on TCP/IP. It only presumes a reliable transport. Any transport protocols that provide such guarantees can be used.)***
- See: https://en.wikipedia.org/wiki/Transport_layer for a nice discussion of other possible protocols

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 is the correct version of the sentence below? **(Please answer)**

- 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 (Network News Transfer Protocol) at 119,
 - and DNS at 53.
 - Port 1024 and above are available to the users.

TCP / IP Final Thoughts / Summary

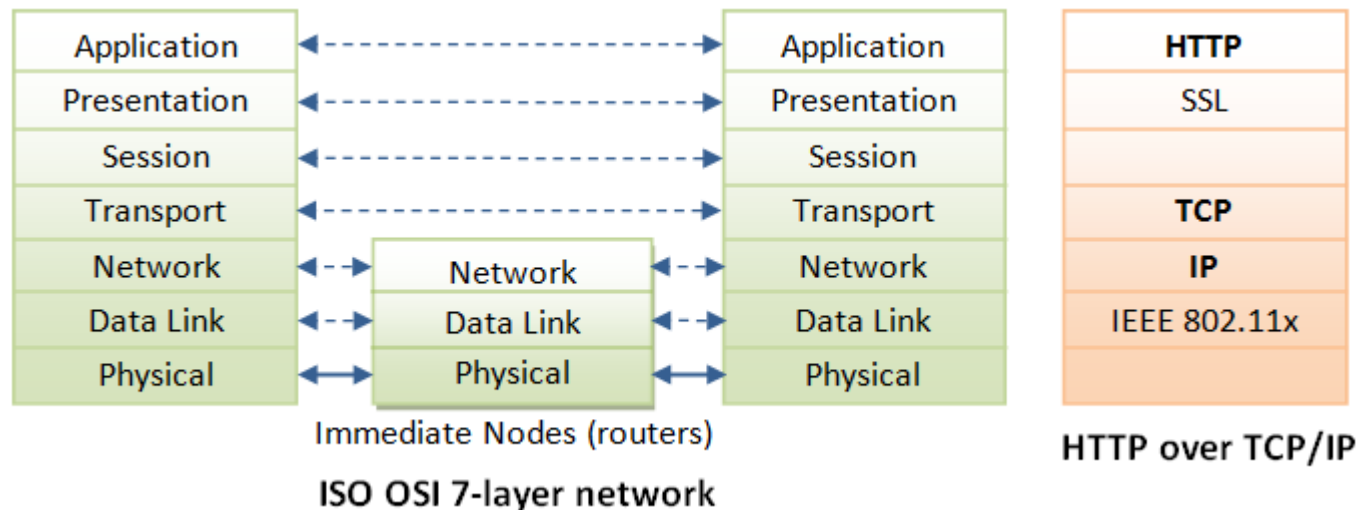
- TCP port 80 is pre-assigned to HTTP, as the default HTTP port number.
- This does not prohibit you from running an HTTP server at other user-assigned port number (1024-65535) such as 8000, 8080 (e.g., when testing a new server).
- You can also run multiple HTTP servers in the same machine on different 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:**
For TCP/IP to function, it needs to know:
(a) IP address or hostname, (b) Port number.

HTTP can be built on an Interface to Presentation Layer

- HTTP is a client-server application-level protocol.
- It typically uses the protocols below it in the 7 layer model below, with TCP/IP underneath (*but any guaranteed Transport Layer Protocol will work*), as illustrated below.



- ***What mechanism can you name that is an interface to the Presentation Layer in the 7-layer ISO OSI 7 – layer network Model??***
- **Your thoughts?**

HW 4 Concept

- YOU Build a limited-functionality HTTP server in Python 3 by layering the functionality to deal with the HTTP protocol on top of the presentation layer
- Your server will have to receive and respond to a subset of HTTP request messages
- Figure out what resource to find and return to the requesting client
 - Find the Resource
 - Return the resource (Compose/return a Response Message)
- OR
 - Return an error! (In a Response Message)

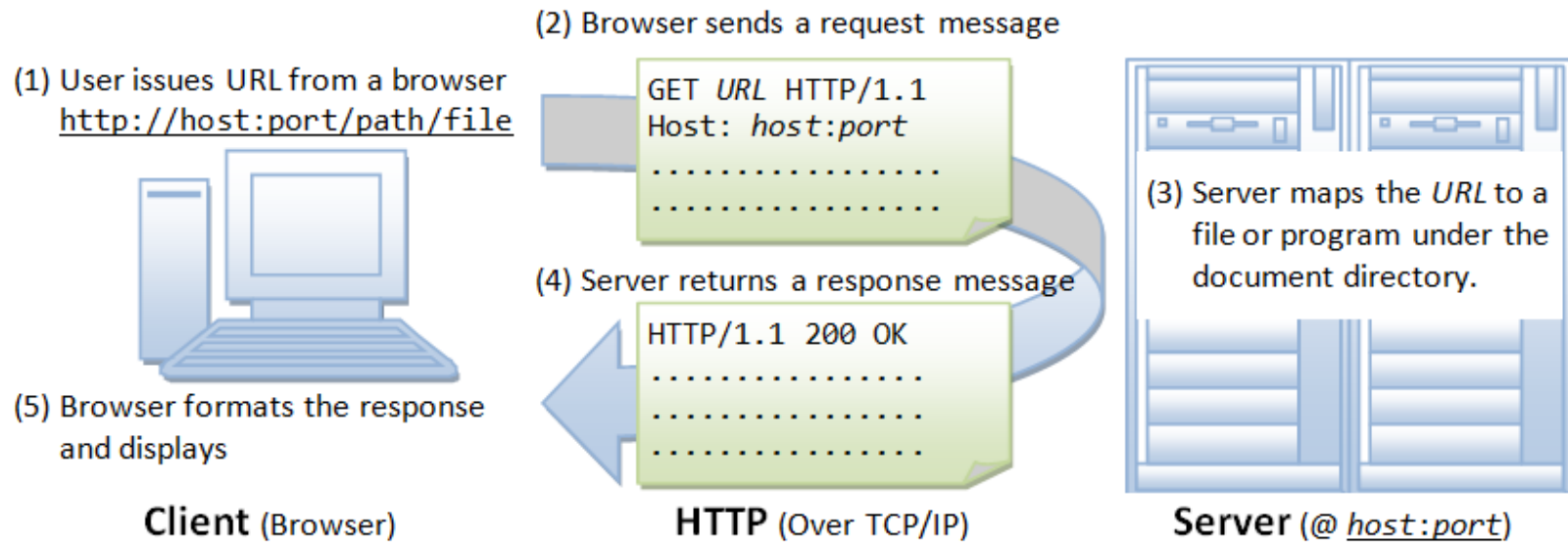
Lets have a look at Python Programs: **EchoClient.py** and **EchoServer.py** (posted on Canvas!)

- Recall:
- ***HTTP is an ISO-OSI Level 7 protocol –that is an Application-level protocol***
- HTTP is built on ISO-OSI level 6 (and below) applications – ***and level 6 and below can be accessed via sockets***
- You are free to use this code to construct the Web Server you will build for HW 4

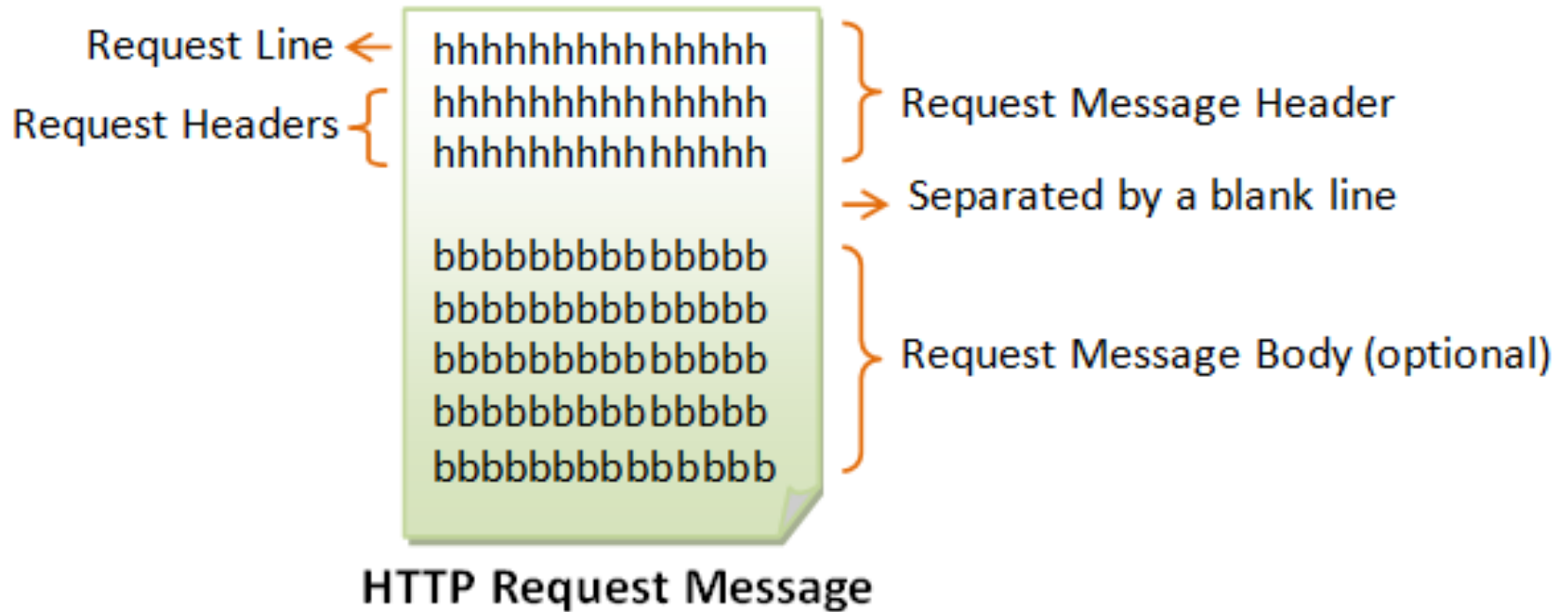
Details, Details – You'll need to understand them to build your Server

- https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
- <http://www.w3c.org/Protocols/>
- Do at least the first reading,
- Refer to the second official Reference for clarification on the details of HTTP 1.1

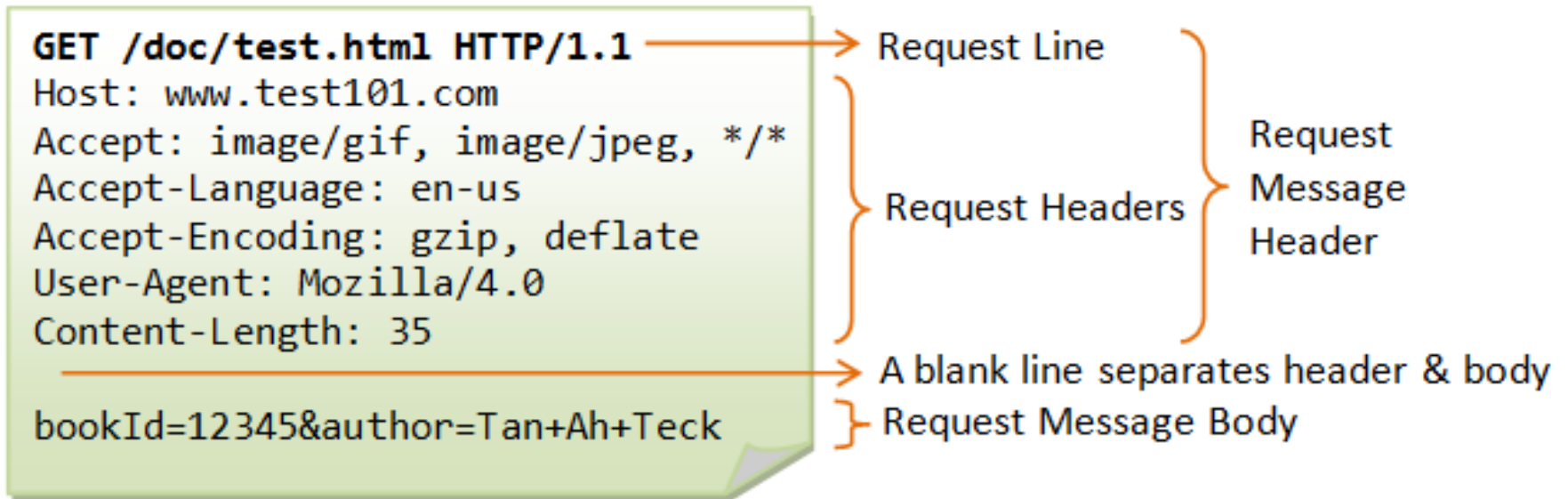
Recall, yet again, how the HTTP Protocol Works



Format of HTTP Request Message



HTTP Request Message Example



THAT's ALL FOLKS: Next Time

- HW 4 overview (Implement/Test a very simple HTTP server in Python)
- Revisit HTTP Protocol in very specific detail
- Introduction to Node.js?