Tutorial questions for Chapter 2.

Expected time to complete: 1 or 2 week(s).

Simple questions: Pease answer the following questions:

1. Define the following terms:
    a. process in a server or client: **program running within a host**. More precisely, a process is the instance of a computer program that is being executed. For example, Firefox is a web browser program. There can be multiple instances of Firefox running in a computer. Each of them is an independent process. Stopping one process will not affect the other.
    b. sockets: **A socket is one endpoint of a two-way communication link between two processes running on the network.** A socket is bound to a port number (0 to 64k) so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.
    c. HTTP: **hypertext transfer protocol**. A set of rules (protocol) for transmitting hypermedia (for example graphics, audio, video, plain text and hyperlinks) documents, such as HTML. It was designed for communication between web browsers and web servers, but it can also be used for other purposes (API, SOAP messages)
    d. DNS: **Domain Name System**. A hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network. It maps easy to remember (domain) names to the numerical IP addresses.
    e. CDN: **Content distribution networks.** A geographically distributed network of proxy servers that provides high availability and performance. They allow faster web objects (text, graphics and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social media site content access.

2. What are the differences between the following?
    a. Client-server architecture and P2P

        Client server: dedicated server and specific clients
        P2P: each node can act as both server and client

    b. TCP and UDP

        TCP: connection-oriented protocol. It establishes a connection between a sender and receiver before data can be sent. This takes time but has good error and flow control.
        UDP: connectionless protocol. It's simple but fast as packets are simply "sent"

    c. Persistent HTTP and non-persistent HTTP

        Persistent HTTP: multiple objects can be sent over single TCP connection between client, server. This means there is only one handshake overhead even if multiple images and files are contained in a HTML page.
        non-persistent HTTP: Each object (image, file) results in a new connection with a handshake.

d. HTTP/1.0, HTTP/1.1 (**You may compare them with HTTP/2**)

HTTP/1.0	HTTP/1.1	HTTP/2
GET, POST, HEAD	GET, POST, HEAD PUT, DELETE	
Stateless	Stateful	
Connectionless	persistent and pipelined connections	
Not supported	chunked transfers, compression/decompression	
	multiple languages	
	textual	binary
	fully multiplexed	fully multiplexed
		one connection for parallelism
		header compression
		Server Pushing

Source and more explanation:
HTTP/1 vs HTTP/1.1 : http://www.ra.ethz.ch/cdstore/www8/data/2136/pdf/pd1.pdf
HTTP/2 : https://www.thewebmaster.com/hosting/2015/dec/14/what-is-http2-and-how-does-it-compare-to-http1-1/

e. SMTP, POP3 and IMAP

SMTP: Simple Mail Transfer Protocol: Protocol used by the sender to **send** an email to an email server (SMTP server).

POP3 and IMAP are protocols for receiving emails at the client side.
POP3: Post Office Protocol – version 3: downloads the email from a server to a single computer, then deletes the email from the server
IMAP: Internet Message Access Protocol: stores the message on a server and synchronizes the message across multiple devices

f. Iterative query and (all) recursive query in DNS

Iterative query: must be supported by all DNS. May give the answer (IP) or a referral to another DNS that can give an answer.
recursive query: Will give a final answer (IP). It will recursively follow up and query other DNS server's in the internet on your behalf for the answer.

3. List at least three protocols that adopt the following protocols, which are not listed on the lecture slides:

a. TCP

b. UDP

Refer to https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers for the complete list of protocols and associated port numbers.

4. Is it reasonable to use UDP for the following protocols; briefly discuss.

a. HTTP

- HTTP, as an application protocol, can be transferred over UDP transport protocol. services that use UDP and an underlying protocol for transferring HTTP data and streaming it to the end-user:
    - XMPP's Jingle Raw UDP Transport Method
    - A number for services that use UDT --- UDP-based Data Transfer Protocol, which is a superset of UDP protocol.
    - The Transport Layer Security (TLS) protocol encapsulating HTTP as well as the above mentioned XMPP and other application protocols does have an implementation that uses UDP in its transport layer; this implementation is called Datagram Transport Layer Security (DTLS).
    - Push notifications in GNUTella are HTTP requests sent over UDP transport.

- QUIC protocol (which is more strictly a pseudo-transport or a session layer protocol) does use UDP for carrying HTTP/2.0 traffic and much of Google's traffic already uses this protocol. It's currently progressing towards standardization as HTTP/3.

- Also see : https://thenewstack.io/http-3-replaces-tcp-with-udp-to-boost-network-speed-reliability/

- There are more in favor in using UDP

b. telnet

- The standard telnet allows to telnet to services running on TCP ports only.
- However, the utility Netcat can be used for working with UDP ports in a very similar manner.
    - https://en.wikipedia.org/wiki/Netcat
    - Netcat site: https://nc110.sourceforge.io/
- Telnet uses small packets, typically interactively
- If the higher-level protocol can handle the errors, packet loss and security UDP can be used.
- There would be no “connection” with the end point established. This means data needs to be sent to know the end point exists.
- It is possible to use UDP.

5. How can a host (e.g., PC, laptop) be uniquely identified in a network? IP address is a 32-bit number that uniquely identifies a host (computer or another device, such as a printer or router) on a TCP/IP network

6. How a process (e.g., a web browser) is uniquely identified in a host?

Process generally have a unique process id for the OS to keep track, schedule and manage.

7. How is a socket associated with a process? A process can bind itself to an available socket. The communication (transport) protocols will deliver any (data) packets to the appropriate port number, which then becomes available to the process which has bound to the port.

8. What is an advantage to use cookies for HTTP?

Cookies are small text files stored on the user’s computer, allowing websites to track the visitors and provide a more customized experience.

- Cookies are domain specific i.e. a domain cannot read/write cookies created by another domain.
- Cookies are browser specific.

- Cookies are profile specific.

Advantages for HTTP

- Store session state: simple to use and implement, occupies less memory, do not require any server resources. Stored on the user's computer so no extra burden on server
- Transparent: Cookies work transparently without the user being aware that information needs to be stored.
- personalized content: User preferences, themes, and other settings
- Tracking: User preferences, themes, and other settings

Additional reading : https://en.wikipedia.org/wiki/HTTP_cookie#Uses

9. What is the reason to disable cookies? (**How are cookies related to tracking?**)

Ability to track user actions and preferences.
Ability to profile a user

10. In what cases web caching is useful or not?

Useful	Not useful
Static data immutable web resources (movies, data dumps) Data accessed by many (either public or internal users) Eliminate lag time	Dynamic data

Calculation questions:

Please solve the following questions (which were the Q1, Q2 and Q3 in the 2019 final exam).

11. [HTTP GET] Suppose that a server receives the following HTTP GET message from a client browser:

```
GET /kurose_ross/interactive/quotation4.htm HTTP/1.1
Host: www.univ1.edu.au
Accept: text/plain, text/html, image/jpeg, image/gif, audio/mp4, audio/vnf.wave, video/mp4,
video/mpeg, application/*, */*
Accept-Language: en-us, en-gb;q=0.5, en;q=0.4, fr, fr-ch, zh, fi
If-Modified-Since: Thu, 25 April 2019 15:20:19 -0700
User Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11
(KHTML, like Gecko) Version/5.1.3 Safari/534.53.10
```

a. What is the name of the file that is being retrieved in this GET message?

The full path tp the resource is: kurose_ross/interactive/quotation4.htm
The file name is quotation4.htm located in the /kurose_ross/interactive subdirectory of the server web root.

b. What formats of text, images, audio, and video does the client browser prefer to receive?

The clients format lists is : text/plain, text/html, image/jpeg, image/gif, audio/mp4, audio/vnf.wave, video/mp4, video/mpeg, application/*, */*

However as */* denotes any content, and format this accepts all formats. For example, application/xhtml+xml is also accepted.

c. Does the client already have a (possibly out-of-date) copy of the requested file? Explain. If so, approximately how long ago did the client receive the file, assuming the GET request has just been issued?

Yes. The “If-Modified-Since:” implying there is a copy already in the Browser cache.

When the original copy was fetched, the Server has indicated that the file was last modified at Thu, 25 April 2019 15:20:19 -0700. This does not give a direct indication when the client downloaded the file.

However, if we assume that file was created at the moment it was originally downloaded (i.e. Thu, 25 April 2019 15:20:19 -0700) we can compute how long ago it was downloaded by
Age = current time - Thu, 25 April 2019 15:20:19 -0700

12. [HTTP Response] Suppose the server-to-client HTTP response message is the following:

```
HTTP/1.1 404 Not Found
Date: Mon, 24 Sep 2018 22:23:34 +0000
Server: Apache/2.2.3 (CentOS)
Content-Length: 74396
Keep-Alive: timeout=39, max=82
Connection: Keep-alive
Content-type: image/html
```

a. Was the server able to send the document successfully? Explain.

No.
Response was a 404, File not found

b. What is the type of file being sent by the server in response?

image/html

c. What is the default mode of connection for HTTP protocol? Is the connection in the reply persistent or non-persistent? Explain.

Default connection for HTTP

- HTTP 1.0 – Non persistent OR
- HTTP 1.1 – Persistent

Since this is HTTP 1.1. There also the keep-alive directive. Either of this implies a Persistent connection

13. [Transport layer] Suppose that nodes A and B want to establish a TCP connection via the three-way handshake. A sent the following TCP segment to B. The following is a dump (contents) of the TCP header in hexadecimal format. Ignore the space between hexadecimal numbers.

```
D201 0043 0000 2711 0000 0000 4002 06EE …
```

a. What is the destination port number? Show your working.

D201 **0043** 0000 2711 0000 0000 4002 06EE OR
0043 (hex), leading 0 optional

port number as 67 (base 10)

b. What is the sequence number? Show your working.

D201 0043 **0000 2711** 0000 0000 4002 06EE OR 0000 2711 (hex), leading 0 optional
seq number as 10001 (base 10)

c. What is the length of header? Show your working.

D201 0043 0000 2711 0000 0000 **4**002 06EE OR 4 (hex)
4 x 4 (scaling factor) = 16 bytes NOTE that the smallest header size is 20 bytes which is also valid

d. What is the window size? Show your working.

D201 0043 0000 2711 0000 0000 4002 **06EE** OR 06EE (hex), leading 0 optional
Indicating windows size as 1774 (base 10) bytes

Practice questions

14. Please try to find and run an existing socket program with both UDP and TCP. This will be useful for you to do the assignment.

You are encouraged to bring up any problems and discuss on Piazza.