COSC 4P14 — Assignment 1

This assignment is intended to act as a supplement to the course's lecture and practical components. Some questions will require a bit of research, but you still need to write your own answers (refer to the department/university's policies on plagiarism).

1. For each of the following, explain the term in sufficient detail, such that a reader would understand its meaning or role with regards to networking.
   a) Client
   b) Server
   c) Latency
   d) Datagram
   e) Encapsulation
   f) Store and Forward
   g) Protocol
   h) Segmentation

2. What are the five layers of the Internet Stack, and what is the purpose of each?

3. How does a web server request a cookie from a web browser?

4. Explain IMAP and POP3, such that I'd fully understand the distinction between them.

5. The Internet Stack doesn't really provide for things like end-to-end encryption, sessions, etc. And yet, we can still have all of these things. Why?

6. Consider two hosts, A and B, connected by a single link with a rate of **R** bps. Suppose that the two hosts are separated by **m** metres, and suppose the propagation speed along that link is **s** metres per second. Host A wants to send a packet of size **L** bits to Host B.
   a) Express the propagation delay, d_{prop}, in terms of **m** and **s**.
   b) Determine the transmission time, d_{trans}, in terms of **L** and **R**.
   c) Ignoring processing and queueing delays, formulate an expression for the total end-to-end delay.
   d) Suppose Host A beings transmitting at time $t=0$. At time $t=d_{trans}$, where is the last bit of the packet?
   e) Suppose d_{prop} is greater than d_{trans}. At time $t=d_{trans}$, where is the first bit of the packet?
   f) Suppose d_{prop} is less than d_{trans}. At time $t=d_{trans}$, where is the first bit of the packet?
   g) Suppose **s**=2.5×10^8 m/s, L=120 bits, and R=56Kbps. Find the distance, **m**, so that $d_{prop}=d_{trans}$.

7. Consider a packet of length **L** which begins at end system A, and travels over three links to end system B. These three links are connected by two packet switches. Let $\mathbf{d_i}$, $\mathbf{s_i}$, and $\mathbf{R_i}$ denote the length, propagation speed, and transmission rate of link **i**, for **i**=1, 2, 3.
   The packet switch delays each packet by d_{proc} for processing.

Assuming no queueing delays, in terms of $\mathbf{d_i}$, $\mathbf{s_i}$, $\mathbf{R_i}$, and **L**, what is the total end-to-end delay for the packet?

Suppose now the packet is 1,500 bytes, the propagation speed on all three links is 2.5×10^8 m/s, the transmission rates for all three links are 2 Mbps, the packet switch processing delay is 3 msec, the length of the first link is 5,000 km, the length of the second link is 4,000 km, and the length of the last link is 1,000 km.

For these values, what is the end-to-end delay?

8. Suppose I have the most amazingly reliable LAN ever built. All of my computers have individual fibre optic lines, made out of Crystallarium (the much-lauded and completely real, refractory line of perfect perfection). Not a single bit could possibly be lost within it.

Will I most likely still need to use TCP to browse the web, or could I hypothetically use UDP? Give a thorough explanation justifying your answer, including at least two *completely* different reasons.

9. Of course, UDP has several potential uses. One common one is for high-bandwidth multimedia transmission (e.g. live-streaming, voice chat, etc.). Name another, completely separate application for UDP, including why it still makes sense for that task.

10. Here's a fun task: you're going to use a netcat-like program (`netcat`, `nc`, `socat`, `telnet`, etc.) to send me an email at my `efoxwell@cosc.brocku.ca` address.
    - Of course, this means you must use the SMTP protocol manually
    - Include a ***small*** binary attachment (e.g. a tiny `.png`)
    - Don't forget all of the standard components you normally see in an email:
        - Proper (non-email) names for both sender and receiver
        - Subject line
        - A normal message body

In addition to having sent it, make sure to include a screenshot/text dump of the conversation as proof.

11. Write a trivial command-line tool that will retrieve the barcode image from the assignment coverpage form, for ***your*** student number, into a file named `<your username>.png` (e.g. `ef99ab.png`)
    - Don't worry about making it adaptable to other student numbers/usernames
        - The point is to retrieve ***yours***
    - Include both the script and the retrieved image with your submission
    - It doesn't matter whether it's a Bash script, a python script, a C program, etc., so long as the marker can easily verify on either one of the Kali installations or sandcastle, from the command-line
        - Include instructions if you're at all concerned about the marker understanding how to use it
        - Note that forming the HTTP requests yourself isn't required, but might be easiest
    - You can't rely on a barcode image *definitely* existing unless you've first accessed the page/form.

https://www.cosc.brocku.ca/archive/forms/cover

12. This is going to be a *slightly* more complicated web-scraping task. You'll be writing another command-line tool, but this one will report on the current prices and trends of both gold and bitcoins, each from a different online price tracker.

It doesn't matter what currency you report in (most use USD). Your tool must retrieve the current prices and changes from each page (it doesn't matter which sites, so long as they aren't both the same site), and display only:

- The current price of gold
- The change in gold (it doesn't matter what their timeframe for 'change' is)
- The current price of bitcoins
- The change in bitcoins (ditto)

Again, writing the HTTP requests yourself isn't necessary, and in this case probably more work than it's worth. Consider looking into `wget`, `curl`, or the standard python libraries for retrieving web resources.

13. It's time for another command-line tool! However, rather than beating a webpage over the head to get what you want, we're actually going to access a resource *properly*.

NASA has several publicly-available APIs, but there are a *ton*. All I want you to do is to find any public-facing API, and write a script that can retrieve (and somehow repackage/present) some information from the API of your choice. Possibilities include “the cloud cover over Brock on your birthday in 2016”, etc.

For your submission, in addition to the script (and instructions, unless it's foolproof), also include a sample output.

14. For packet-based transmissions, the queueing portion of an end-to-end transmission will typically be very small. In fact, due to the finite capacity of a router's buffer, there is an absolute upper limit on this delay for a successful communication. In spite of this, excessive queueing could potentially lead to tremendous delays in receiving data. ...how?

Submission

Type it up, and export as a pdf. If you need to include pictures that for some reason can't be embedded within the document, make it clear how the multiple files relate to each other. Bundle up everything (including your scripts and sample outputs) into a single .zip, and submit that.