

Homework 2

Topic: Application Layer Protocols

Posted on Tue Oct. 17th, 2017

Due on Tue Oct. 31st 2017, 3pm, *online only* (on Canvas + EECS Server)

You are strongly advised to work in groups of two.

1. (50 Points) Socket Programming (in Python).

Use **Python 3** to implement the following:

- (a) (30 Points) **Assignment 1: HTTP Server & Client:** Implement the web servers and the client as described here: <https://canvas.eee.uci.edu/files/2246990>.
 - i. (20 Points) Single-threaded web server, including error handling.
 - ii. (5 Points) Multi-threaded web server.
 - iii. (5 Points) HTTP client.
- (b) (20 Points) **Assignment 2: Mail Client:** Implement the mail client as described here: <https://canvas.eee.uci.edu/files/2246989>.

Note: If you need to familiarize yourself with python, read the tutorial “Getting Started with Python”, available at the companion website and also re-posted here for your convenience: <https://canvas.eee.uci.edu/files/2246988>. *Just make sure to use Python 3 instead of 2.*

Turning the Code in: This part (the coding part) of the homework is to be turned in using one of the EECS servers. See the following file for instructions on how to submit the coding part of the assignment: <https://canvas.eee.uci.edu/files/2247007>. *The rest of the homework is to be turned in via Canvas.*

2. (35 Points) Wireshark Labs

- (20 Points) **HTTP.** See <https://canvas.eee.uci.edu/files/2246996> and complete only two parts of this assignment:
 - (14 points) Part 1: Basic HTTP GET/Response interaction, i.e., answer questions 1-7, using <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html> as the test file.
 - (6 points) Part 4: HTML Documents with Embedded Objects, i.e., answer questions 16-17, using <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html> as the test file.
- (15 Points) **DNS.** See <https://canvas.eee.uci.edu/files/2246994> and answer questions 11-19 only. It is preferred that you capture actual network traffic. If this is not possible, then you can use some of the traces provided.

Note: If you need to familiarize with Wireshark first, see the Discussion of Week 1, or go to the companion website, under “Wireshark Labs”, (“Getting Started”). Here is also the user guide for wireshark https://www.wireshark.org/docs/wsug_html_chunked/.

3. (15 Points) **HTTP (from 2015 Midterm, similar to problems 7-8 in Ch.2)**

You open your browser and you click on a URL. Assume that the webpage associated with the URL consists of a small HTML file and 2 small images, all stored on the same webserver. Let RTT_{cs} be the RTT from the client host to the webserver containing the URL. The IP address of the webserver is not cached at the local host but is cached at your local DNS server. Let RTT_{DNS} be the RTT from the client host to the local DNS server.

What is the total delay from the time you click on the webpage until the time that the entire webpage is displayed on your browser? Take into account *all* messages sent/received by the client host, including DNS, HTTP and TCP-related messages. Consider two cases:

- (a) Persistent HTTP without pipelining.
- (b) Non-persistent HTTP with 2 parallel connections.