

# CS:4980 Advanced Computer Networks

## Programming Project # 1

---

**Due Date: Friday October 9, 2015 at noon in ICON Dropbox.**

Submission Instructions: Submit all required files in a single .zip file.

*Late submissions are not allowed unless you have prior permission to do so.*

*You are not allowed to copy code/data from the Internet. You cannot discuss your code/data with each other (except your project partner).*

---

You are asked to implement an HTTP client and server running a pared down version of HTTP/1.1 using sockets and threads/processes. This project can be completed in Python (used in textbook), C/C++, or Java.

Your server should take command line arguments specifying the server port and path to the directory containing index.html and other files provided to you (<https://www.dropbox.com/sh/2jhfln8yvanraj6/AAD66xUk8E6n2NYCgLQkzhU-a>). The format of the command line is:  
`myserver server_port directory_path`

Your client should take command line arguments specifying a server name or IP address, the port on which to contact the server, and an input text file containing HTTP GET request for index.html. The format of the command line is:  
`myclient server_ip server_port file_name`

After receiving the index.html from the HTTP server, your client should parse it and issue subsequent requests to fetch the images referenced in the html file (consider only those with `<img>` tag and `src` attribute pointing to files in `./index_files/` path). There are > 300 such references in the index.html provided to you. Some references might be missing, in which the server should return

appropriate HTTP response code. You can use HTML parsers (e.g., HTMLParser in Python) to parse index.html.

You will implement the following flavors of HTTP.

1. Non-persistent HTTP with 1 connection
2. Persistent HTTP with 1 connection
3. Non-persistent HTTP with 5 parallel connections
4. Persistent HTTP with 5 parallel connections

For each of these HTTP flavors, you are required to report HTTP page load time as a function of varying delay (e.g., 10 ms, 50 ms, 100 ms, 250 ms, 500 ms, 1 sec) and loss (e.g., 0%, 0.1%, 0.5%, 1%, 5%, 10%, 50%). You should use netem at the HTTP server for this purpose. You can find more details about how to use netem at: <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>

Analyze and plot page load times for each HTTP flavor and for varying delay and loss. Answer the following questions and include detailed discussion and justification for your answers.

**Q1.** How does the performance of HTTP with non-persistent connections and HTTP with persistent connections compare?

**Q2.** How does the performance of HTTP with 1 connection and HTTP with 5 parallel connections compare?

**Q3.** How does page load time degrade for increasing delay and loss?

Extra Credit:

**Q4.** What is the impact of varying number of parallel TCP connections on page load time? Can increasing parallel TCP connections degrade page load time?

What to submit:

1. Submit client and server **source codes** and **binaries** for each of the 4 HTTP flavors. Include a **readme file**. Your code will be tested on Ubuntu 14.04.3 LTS.
2. Submit a **detailed report** (in PDF format) showing sample outputs and containing detailed answers to the aforementioned problems.
3. Include the following **signed plagiarism statement** as a scanned PDF. Your submission will be rejected without the inclusion of this signed statement.

I certify that all work submitted in this document is my own work. I have completed all of the assignments on my own without assistance from others except as indicated by appropriate citation. I have read and understand the university policy on plagiarism and academic dishonesty. I further understand that official sanctions will be imposed if there is any evidence of academic dishonesty in this work. Please provide your identity to certify that the above statements are true.

Group Member 1:

Name (Printed)	Signature	Date
Jianshen He	Jianshen He	Oct 11, 2015

---

Group Member 2:

Name (Printed)	Signature	Date

---