**CSE-5344 Project1**

Web Proxy Server

**Code name:** server.py

Flow of the program:

1. Function **start\_server()** is called from the **WebProxy** class
2. In **start\_Server()**, a server socket **ssoc** is created. The port number i.e. **sport** is **8080.** Function **create\_thread(ssoc)** is called.
3. **create\_thread(ssoc) ,** on receiving request from client i.e. Web browser, creates a new thread **thread1** and starts it.
4. In the **run()** function of class **myThread(threading.Thread)**, the connection socket **connsoc** is created for the request from client. Current time is stored in variable **start\_time** which will be used to calculate the RTT.
5. The client details, request details and socket details are logged in text file **log.txt** and are printed as program output. In case of exceptions generated, the error code and error is sent to client and also logged in the **log.txt**. After exceptions, **create\_thread(ssoc)** so that new request can be made from browser.
6. Function **cache(data,url,connsoc,self.socc,start\_time)** is called. For the request made, it tries to open the cache file. In case of New requests, since there is no cached file, control goes to the **except** block. In the **except** block it is *Cache Miss****.*** A text file is created for caching. A new socket **proxysoc** is created between web proxy server and the actual web server at **port 80**. The request **req** is sent and response **res** is received from the actual web server. Response **res**  is sent from proxy server to the client and also the complete response is stored in **fresp**. Current time is taken as **end\_time.** *RTT* is calculated as (**end\_time - start\_time**). **fresp** is written into the text file created for caching.
7. **connsoc** and **proxysoc** are closed. **create\_thread(ssoc)** is called for accepting new requests from client.
8. For the requests that are made again, it’s a case of *Cache Hit*. In this case, instead of getting response from actual web server, the cached file is opened, the response is read and sent to the client directly. **end\_time** is calculated and *RTT* is calculated as (**end\_time - start\_time**). **connsoc** and **proxysoc** are closed. **create\_thread(ssoc)** is called for accepting new requests from client.

Q) Do you see variation in time when object was cached? Yes/No give reasons

Ans) **Yes.** The time taken for packets to travel back and forth while sending request from proxy server to web server and receiving response from web server to proxy server is reduced if the response is already cached. Thus, there is a variation in time.

References:

1. J. Kurose and K. Ross, “Computer Networking: A Top-Down Approach,” 6th edition
2. <http://heather.cs.ucdavis.edu/~matloff/Python/PLN/FastLanePython.pdf>
3. <http://null-byte.wonderhowto.com/how-to/sploit-make-proxy-server-python-0161232/>
4. <https://www.tutorialspoint.com/python/python_multithreading.htm>
5. <http://stackoverflow.com/questions/2757887/file-mode-for-creatingreadingappendingbinary>
6. <https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html>
7. <http://stackoverflow.com/questions/449560/how-do-i-determine-the-size-of-an-object-in-python>
8. <https://docs.python.org/3.3/library/socket.html>
9. <https://www.tutorialspoint.com/python/python_date_time.htm>
10. <http://stackoverflow.com/questions/1035340/reading-binary-file-in-python-and-looping-over-each-byte>
11. <https://recalll.co/app/?q=python%20-%20Use%20HTTP/1.1%20with%20SimpleHTTPRequestHandler>
12. <https://blog.packet-foo.com/2014/07/determining-tcp-initial-round-trip-time/>