The University of Calgary

Department of Electrical & Computer Engineering

ENSF 462 Networked Systems

(Fall 2023)

Due: Oct 24, 2023

Lab 3 Web Proxy Server[1]

Lab Section	Section Date	Location
B01	October 19th, 2023	ENA 305
B02	October 24th, 2023	ENG 24
B03	October 18th, 2023	ICT 319

Objectives

In this lab, you will learn how web proxy servers work and one of their basic functionalities – caching. Your task is to develop a small web proxy server which is able to cache web pages. It is a very simple proxy server which only understands simple GET-requests, but is able to handle all kinds of objects not just HTML pages, but also images.

Generally, when a client makes a request, the request is sent to the web server. The web server then processes the request and sends back a response message to the requesting client. In order to improve the performance, we create a proxy server between the client and the web server. Now, both the request message sent by the client and the response message delivered by the web server pass through the proxy server. In other words, the client requests the objects via the proxy server. The proxy server will forward the client's request to the web server. The web server will then generate a response message and deliver it to the proxy server, which in turn sends it to the client.

[1] **Acknowledgement:**

The content in Lab 3 manual is modified based on the material provided on the authors' website for the textbook, *Computer Networks, A Top-down Approach, 8h ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2020.*

Proxy Behavior

Following features need to be implemented in your HTTP proxy server:

- Proxy server accepts connection from HTTP clients like web browsers.
- Reads the incoming GET HTTP request and check if the object requested is available in the local cache. If available, the object is served from the local cache, else, object is requested from the origin server where the object resides on behalf of the client and returns the same to the client once received from the origin server. Your web proxy acts both as a client and a server. It is a server when it communicates with web clients and a client when it communicates with origin servers.
- A local copy of the object is created so that any future request for the object is served from the local cache.
- When objects are locally stored, the file name should match the *hostname/pathname* of the URL in the GET request from the web browser, with the same directory structure being created locally inside the directory where your program is running.
- For requests other than GET, web proxy should return response with status code “400 Bad Request".
- For response other than “200 OK" from the origin server, send “400 Bad Request" response to the client.
- The proxy server should support non-persistent HTTP (as in HTTP 1.0). This means that once an HTTP request is served, the proxy server closes the TCP connection. To inform the client that the connection is closed, include the header line “Connection: close” in the server response.
- Your proxy server must be able to handle both text (Simple text html files) and binary (e.g., images, pdf) files.

Program Structure

You might find a skeleton code in https://gaia.cs.umass.edu/kurose_ross/programming/Python_code_only/Web_Proxy_programming_only.pdf useful in structuring your code or using specific instructions for example in reading from and writing to files or sockets. While you can use this code, it is only a guideline, and you are free to write your own code. Besides, this skeleton code might not provide all the requirements listed above, so do not depend entirely on this code.

Running the Proxy Server

Run the proxy server program first, and then request a web page from your browser. Direct the requests to the proxy server using your IP address and port number, e.g., http://localhost:8888/www.google.com can be entered in address bar. You will replace the port number used here “8888” with the port number you have used in your server code at which your proxy server is listening.

You can also directly configure your web browser to use your proxy. This depends on your browser. In Internet Explorer, you can set the proxy in Tools > Internet Options > Connections tab > LAN Settings. In Netscape (and derived browsers such as Mozilla), you can set the proxy in Tools > Options > Advanced tab > Network tab > Connection Settings. In both cases you need to give the address of the proxy and the port number that you gave when you ran the proxy server. You should be able to run the proxy and the browser on the same computer without any problem. With this approach, to get a web page using the proxy server, you simply provide the URL of the page you want. For e.g. http://www.google.com.

Make sure you clean your browser cache after each webpage access.

Assumptions

- Only well formatted GET requests are received by your proxy. But, if request is not GET, then proxy needs to send “400 Bad Request" response.
- The proxy cache always has the latest version of the object once downloaded from the origin server (i.e objects are never updated at the origin server). So, you DON’T need to implement “Conditional GET"
- All requests are for a single object in the Internet (i.e., you can IGNORE the case of base html and embedded objects in it)
- Web clients (like web browsers) and Web proxy run in the same machine.

Submit a lab report that includes the following:

- **Your name and UCID #**
- **The proxy server code and the screen shots of your client browser, verifying that you actually receive the contents of the files.**