Comp 4621
Project Report

Kee Pak Yiu
20512528

Reference:
I googled a lot to understand the concept involved in the project, https://gist.github.com/darkwave/52842722c0c451807df4 this is one of the project I studied a lot to understand forward proxy, but end up I find these code not working. However, it still help me a lot on understanding the concept.

Project Implementation:

I implement the project on python, as I don't need to care about the memory allocation on python.

In the python file, the main function will create a Server object and run server.loop() from it. And this is the on going loop to function the server.

in the loop() function, it run an infinite while loop to accept incoming connections. When there is a new incoming connection, it create a new thread running thread_function() to handle it, while loop() will continue it loop to accept new incoming connection. By this it can fulfil the requirement of supporting multiple threads.

In the thread_function(), it will create a channel to pass client and target address' message to each other.

First, it will check if the target URL is blocked or not. And send a 404 message to the client if it is the case, fulfilling the requirement of supporting access control. While the list of blocked URL is initialise as global variable in BLOCKED_URL, it can be changed by changing it content before running the python file.

Then, it will check if the request is in HTTP or in HTTPS. If it is in HTTPS, it will send a 200 message to the client after the connection is set up with the target server, and then it will run two thread calling two forward_function() to handle incoming message from both side and forward them to their corresponding destination.

If the request is in HTTP, thread_function() will first check if the request file exist in the cache, and reply with the cache if it exist. Else, it creates two thread and run forward_function() very much like handling HTTPS, except some different in the flag when calling the function.

In forward_Function(), if the request is in HTTP and if the thread is the thread receiving data from the target server side, it first create an array to ready to cache the file.

In this project, cache is implemented by using dictionary. Using the URL of the request as the key, and an array of int as the data. And all caches are stored in an increasing integer as file name. An array is used to store the data, as there will be files there need more than one packet to be sent.

After creating the entry to store future cache, forward_Function() enter an infinite while loop to forward messages, it return when there is no more data received. While if the request is HTTP, it also store the incoming response to the cache.