Programming Assignment - Proxy Server

Instructions

- Follow the instructions in this document and complete both sections.
- This assignment has two components. Section 1 is a programming task and Section 2 is a number of questions to test your understanding.
- There are a total of 55 marks available.
- Each student will make one submission consisting of two files - a python file for Section 1 and a PDF file for Section 2.
- Submit via Canvas by **11:59pm Friday, Week 11 (18th October 2024)**.
- This Assignment is worth 15% of your total grade.

1 Programming Assignment (35 marks)

In this section, you will develop a proxy server (hosted on your computer) which is able to cache web pages. It should be a very simple proxy server which only understands GET requests, but is able to handle all kinds of objects including HTML pages and images.

A proxy server is essentially a gateway between a user and the internet. They are often used as an additional layer of security separating a private network from the internet. One benefit is that the user's IP address will be hidden, so that the website only sees the address of the proxy server.

Consider Figure 1. The client wants to get data from a webpage, it first sends the request to the proxy server. The proxy receives the request and processes it (performing security checks or checking the request against a 'blacklist' of forbidden sites). The request is then forwarded to its intended address and the reply is sent back to the proxy. Again, the proxy will process this and then forward the data on to the client.

1.1 What you need to do

In this assignment, you will take the template python script provided on Canvas and you will fill in the code where indicated. This script uses the socket module for establishing network connections. For more information on this module and the functions it provides, see the documentation here: https://docs.python.org/3/library/socket.html

Sections that need to be completed are marked with `# TODO: Start` and `# TODO: End`. Follow the guidance given by the 'Start' comments.

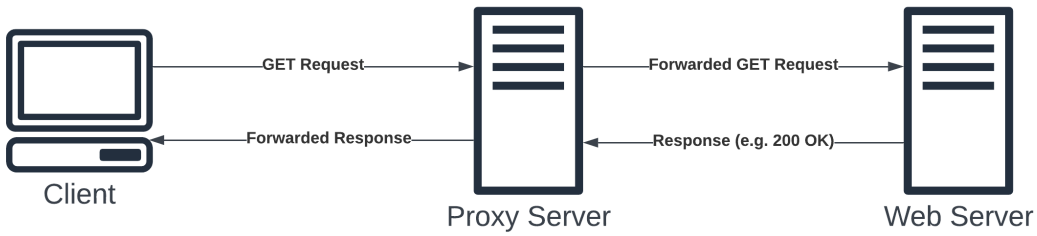

Figure 1: The client-proxy-web process

1.2 Hints

You will need to look through the documentation carefully to determine the correct functions to implement the proxy server.

A couple useful hints:

- The HTTP messages will need to be sent/received in UTF-8 format. So the string `.encode()` and `.decode()` methods will be useful for handling these.
- The `socket.sendall()` method will work more reliably than `socket.send()`.
- Use print statements to check the HTTP messages being received/sent are as expected.
- Pay attention to how HTTP messages are formatted, including whitespaces, `'\r'` and `'\n'`.
- The 'If-Modified-Since' time should be given in GMT, not local time.
- There are plenty of free online resources that give good examples of how to correctly use the socket module.

1.3 Testing the proxy server

It will be best to test your proxy server on simple HTML pages, such as those used in the first Wireshark lab. You are welcome to test on any web page you like, but the results will be easiest to inspect on simple pages. To test the proxy server:

- Ensure the python script is running.
- Open your browser and disable the browser's cache for testing. To disable the cache in Chrome or Firefox:
  - Right click in the browser window and select 'Inspect'.
  - Select the 'Network' tab and ensure the 'Disable cache' box is ticked.
  - Keep this 'Network' box open while you do your testing.
- In the URL bar, type `localhost:8888/` followed by the website URL. For example: localhost:8888/www.example.com.
- If the proxy is working correctly, the webpage will load as normal and a file will be created in the `PROXY_CACHE` folder in the same location as your python script.

– Don't forget to clear this `PROXY_CACHE` folder in between tests, especially if you have errors.

Note that this assumes the proxy server is hosted on the same machine as the browser. The default host name (`localhost`) can be replaced by the IP address or host name for a proxy hosted on another machine, but this is not required for this assignment. The port number (8888) can also be changed, as long as it matches the port number used in the python script.

1.4 Criteria

The first section of the assignment will be assessed according to the following:

Requirement	Not Implemented	Partially Implemented	Implemented
Bind socket, begin listening and receive request from client	0 marks	2.5 marks	5 marks
Create web socket and connect to web server	0 marks	2.5 marks	5 marks
Request is sent to web server and response is received successfully	0 marks	2.5 marks	5 marks
Files are cached and retrieved from cache successfully	0 marks	2.5 marks	5 marks
Files only retrieved from cache if 304 status returned	0 marks	2.5 marks	5 marks
Catch errors and send the client an error code	0 marks	2.5 marks	5 marks
Template used, code is efficient, well structured and clear	0 marks	2.5 marks	5 marks

2 Questions (20 marks)

Along with the completed code from the previous sections, you are required to answer the following questions:

1. Considering the functionality of this simple proxy server, identify and discuss at least three significant limitations of your proxy server. **(6 Marks)**
2. How could you implement potential improvements into this script to overcome these limitations? A flowchart or pseudocode may help to illustrate your answer. **(6 marks)**
3. Is the proxy server using UDP or TCP sockets? How can you tell? What other protocols are involved?**(2 marks)**

4. In this question, you will put together much of what you have learned about Internet protocols. Suppose you buy a brand new computer, connect it to Ethernet, and want to download a Web page. What are all the protocol steps that take place, starting from powering on your PC to getting the Web page? Assume there is nothing in our DNS or browser caches when you power on your PC. (Hint: the steps include the use of Ethernet, DHCP, ARP, DNS, TCP, and HTTP protocols.) Explicitly indicate in your steps how you obtain the IP and MAC addresses of a gateway router. **(6 marks)**

3 Submission

For this assignment, you are required to submit **two** files - Canvas accepts multiple files for a single submission. Please submit your code from Section 1 as a single python file (based upon the given template). As a separate file, please submit your answers to the questions in Section 2 as a PDF file.