

Design Document for P2 HTTP Proxy

Kumar Aditya

1. Socket Management

Describe 1) how many socket objects are used, 2) the variable name for each socket, and 3) the purpose of each socket object.

3 socket objects are used. The first socket is `proxy_socket`. This is used by the proxy to listen for client requests to make connections with the client (Firefox). The second socket is `client_socket`. This is the socket used for communication between the proxy and client and is provided by the `proxy_socket`'s `accept` method. Finally, the third socket is `emerald`. This socket is used for communication between the proxy and the web server and is created within the `proxy` method.

2. Multi-threading

Overall Design

Describe how multithreading is used in your program. You must include 1) the number of threads you use including the main thread, 2) what each thread is for and doing, 3) any loops you have for each thread.

2 threads are used. The main thread is used for listening to proxy requests from clients and makes new client sockets to be used for within the sub threads that communicate with the client. The sub threads are used to create sockets that communicate with the server by creating a connection and then tunneling information back to the client.

Justification

Justify why your design is efficient and show evidence it can handle multiple client requests efficiently. Specify your testing scenario (how many requests were made, which websites were being used, etc

The design should be efficient as it has one main thread handling the distribution of client requests and sub threads that handle each client request and stop dedicating resources once those requests are complete. This means that simultaneous requests should not be interfering with one another as each client has its own dedicated thread supporting it without holding back each other.

3. Streaming

Describe how streaming is implemented in your Proxy and the parameter (i.e. num of bytes) used for streaming. Justify the choice of your parameter.

Streaming is implemented within the proxy by sending and receiving all data in packets of 2048 bytes each, the chosen buffer size for this project. This buffer size was chosen as it was used for the previous project and is large enough to contain a significant amount of data while also being small enough to not cause transmission delay. The only exception to this streaming are the get requests as they are small enough to not have to bother about buffering.

4. Data structures

In the cell below, list any notable data structures you used and justify the use of the data structure. Also specify how you handle synchronization for the data structure if there were any need for synchronization. If none, you can say "None".

None.

5. How shutdown is handled

Describe how you handled the shutdown gracefully.

Shutdown is handled at the end of the proxy method by closing both the client_socket and emerald sockets. This allows for the proxy to keep running to handle future requests while not dedicating resources to previous requests that have been fulfilled. Additionally, this closes unnecessary communication channels between the proxy and the client and server.

6. Error handling

Describe how you handle unexpected errors by specifying 1) what kind of errors you may encounter 2) what you do for each error.

Error handling such as time out errors or bad gateway errors when attempting to connect to the server are handled by sending a bad gateway response back to the client, logging this error if the LOG_FLAG is true, and closing both client and server sockets.

7. Any libraries used

List any libraries that you used for the implementation and justify the usage.

8. Reflection

What was the most challenging part working on this project? Most fun part?

The most challenging part of this project was understanding the instructions as some diagrams and instructions were contradictory, hence causing confusion on how to proceed about this project. The most fun part of this project was when it was working at the end.

If you are to do this all over again, how would you do it differently?

I would clarify more details about this project as the instructions were vague and not very informative.