CSci846 Project 1

In this programming assignment, you will practice socket and multi-thread programming, and get a deeper understanding of the HTTP protocol. Penalties will be given for late submissions (20% deduction per day).

Language:
You are required to use Java as the implementation language. Please stick to basic Socket and multi-threading libraries.

Deliverables

You should hand in a (1) report which presents your implementation, (2) a README file explaining how to run your program, and of course (3) your source code

Total Marks: 100

A HTTP Proxy Server.

 AIM: In this assignment, you will implement a HTTP proxy server that passes requests and data between web clients and web servers. This will give you a chance to get to know one of the most popular application protocols on the Internet- the Hypertext Transfer Protocol (HTTP), and give you an introduction to the Berkeley sockets API. When you're done with the assignment, you should be able to configure your web browser to use your personal proxy server to browser websites.

Introduction:
HTTP proxy acts as a mediator between the client (for example, web browser in a user's computer) and server (for example, web server site). In the simplest case, instead of sending requests directly to the server the client sends all its requests to the proxy. The proxy then opens a connection to the server, and passes on the client's request. The proxy receives the reply from the server, and then sends that reply back to the client. Notice that the proxy is essentially acting like both a HTTP client (to the remote server) and a HTTP server (to the initial client).
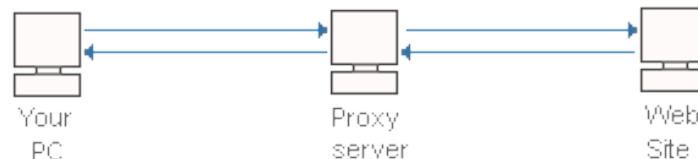
Proxies are used for many purposes. Sometimes proxies are used in firewalls, such that the proxy is the only way for a browser inside the firewall to contact an end server outside. The proxy may do translation on the page, for instance, to make it viewable on a Web-enabled cell phone.

Proxies are also used as anonymizers. By stripping a request of all identifying information, a proxy can make the browser anonymous to the end server. Proxies can even be used to cache Web objects, by storing a copy of, say, an image when a request for it is first made, and then serving that image in response to future requests rather than going to the end server.

In this assignment, you can deal with version HTTP 1.1 as defined in RFC 2616. You should read through the RFC and refer back to it when deciding on the behavior of your proxy.

HTTP communications happen in the form of transactions, a transaction consists of a client sending a request to a server and then reading the response. Request and response messages share a common basic format:

- An initial line (a request or response line, as defined below)
- Zero or more header lines
- A blank line (CRLF)
- An optional message body.

Assignment Details:

Design and implement a multi-threaded simplified HTTP proxy server that passes requests and data between web clients and web servers. The proxy server also logs requests. Below are some basic requirements, but you are welcome to use your own judgment to implement an advanced version that demonstrates various aspects of a HTTP proxy server.

1. Logging
   Your proxy should keep track of all requests in a log file named proxy.log. Each log file entry should be a file of the form:
   Date: browserIP URL
   where *browserIP* is the IP address of the browser, *URL* is the URL asked for. For instance:
   Jan 23 2010 02:51:02 134.129.125.204 http://www.cs.ndsu.edu/

2. GET method
   A full HTTP-1.1 compliant Web server supports HEAD, POST, and GET methods. Your proxy server only needs to support GET method. To serve each request, we first need to parse the request line and headers sent by the client. Since we will only support GET method, we only care about Web page name in request line. The request line for the proxy server typically looks like this:

   GET http://www.cs.ndsu.nodak.edu/aboutus.htm HTTP/1.1
   ... ...

   The requested Web page name contains the Web server name www.cs.ndsu.nodak.edu and the requested file on that server /aboutus.htm. In this case, your proxy server should make a TCP connection to Web server www.cs.ndsu.nodak.edu at the default port 80 and ask for file /aboutus.htm.

After sending a request to the "real" Web server, an HTTP response including the requested file will be received at the proxy server. The proxy server should then forward the content to the client.

3. Multi-threading.
   Real proxies do not process requests sequentially. They deal with multiple requests concurrently. It is possible for multiple peer threads to access the log file concurrently. Thus, you will need to use a method to synchronize access to the file such that only one peer thread can modify it at a time. If you do not synchronize the threads, the log file might be corrupted. For instance, one line in the file might begin in the middle of another.

4. Caching
   Caching is a desirable feature at the proxy server. With caching, the proxy server stores the returned data of past requests in local storage. After downloading a web object successfully, you should cache the object to disk so that subsequent fetches can use the local copy as opposed to fetching it again remotely. If the new request matches a past one, the proxy server will directly return the cached data (in local storage) without actually contacting the remote Web server. This may save wide-area network bandwidth because the proxy server is usually close to the client machines while the "real" Web servers are often far away.

   When you test the caching functionality in the proxy server, keep in mind that your Web browser also has a cache. So if you request the same file twice from the same browser, the browser cache would kick in first and the second request would not even reach the proxy server. A proper way to test the proxy server caching is to have two different browsers (with different browser caches) to access the same Web page.

Testing Your Proxy

Run your client with the following command:
./ProxyServer <port>, where port is the port number that the proxy should listen on.

You can configure your web browser to use your proxy server as its web proxy. Please avoid using Chrome as your client browser, as it keeps sending too much complex information to Google all the time. You can use Firefox.
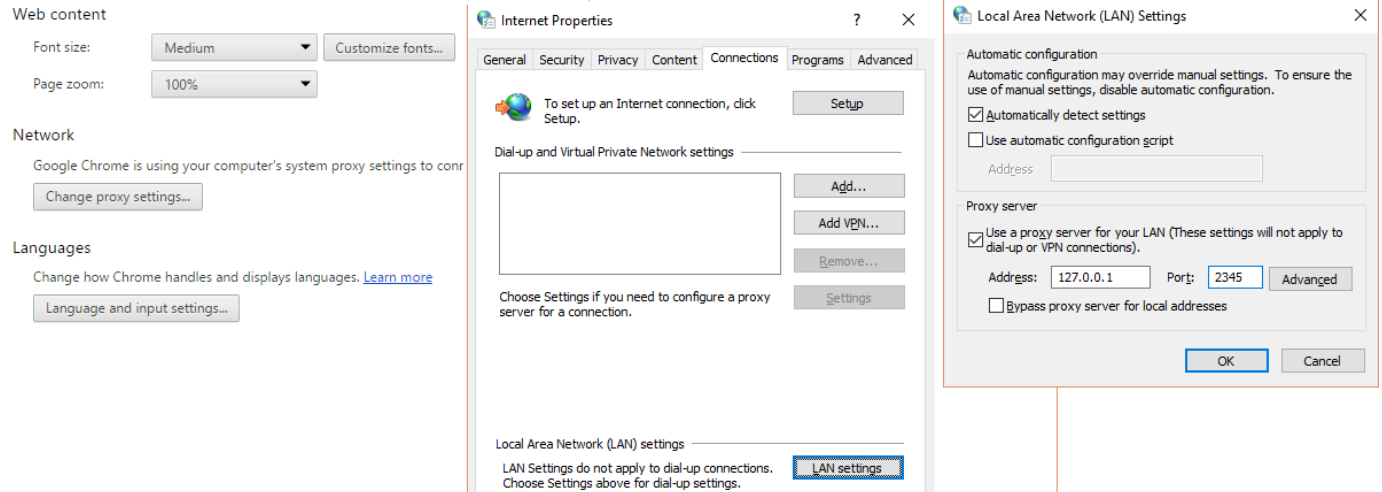
Provided Partial code:

You are given two java files with partial of the code implanted for you. Please fill in the TO DO parts.

Assignment Report:

In your report you should include these parts:

(1) Describe of the design of your program.
(2) Explain the challenges you have faced, solved, and unsolved.
(3) What are the limitations of the program? How it can be solved if you are given extra time. The code given to you is not perfect, it can be improved too. You can also point out the problems of the given code.