CSE 5344 FALL 2016 – COMPUTER NETWORKS PROJECT 1

WEB PROXY SERVER- CACHING

NAME: AISHWARYA RAJAGOPALAN

UTA ID: 1001192345

Objectives:

- Learnt the basics of Socket Programming
- Understood the functionality of a Proxy Server
- Explored the basic structures of HTTP messages

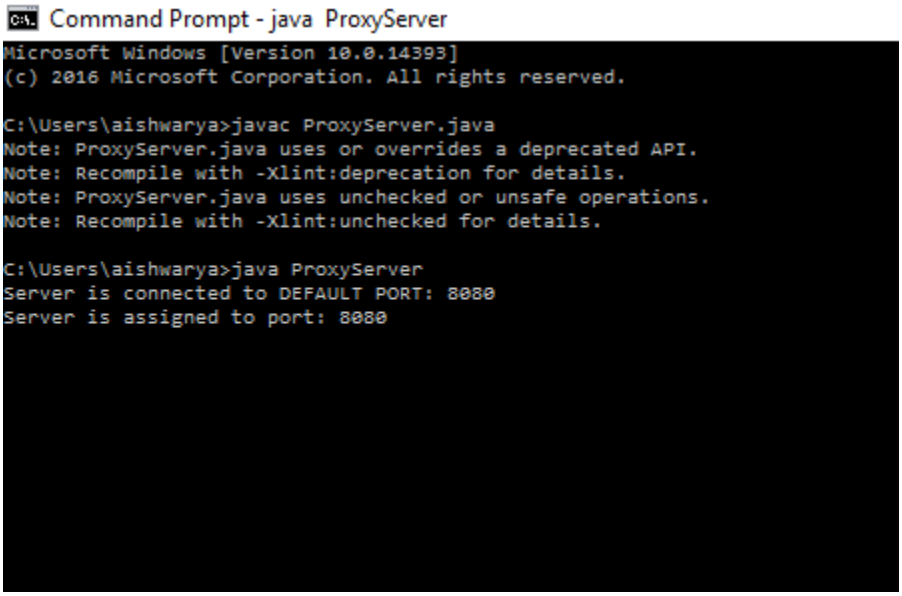
Project Development Environment:

- Developed code in **Java** programming language using **jdk 1.8.0_77** and Eclipse IDE.
- Used the command prompt/Terminal to compile and execute.
- Used Mozilla Firefox browser as the client browser to configure proxy settings and to see the working of the Proxy Server.

Execution Procedure:

- Open **Command Prompt/Terminal**. Compile the ProxyServer.java file by typing **javac ProxyServer.java**
- Run the program in the terminal along with a port number.
  Eg: **java ProxyServer 7635**
  Client can assign any port number other than well-known ports to the Proxy Server

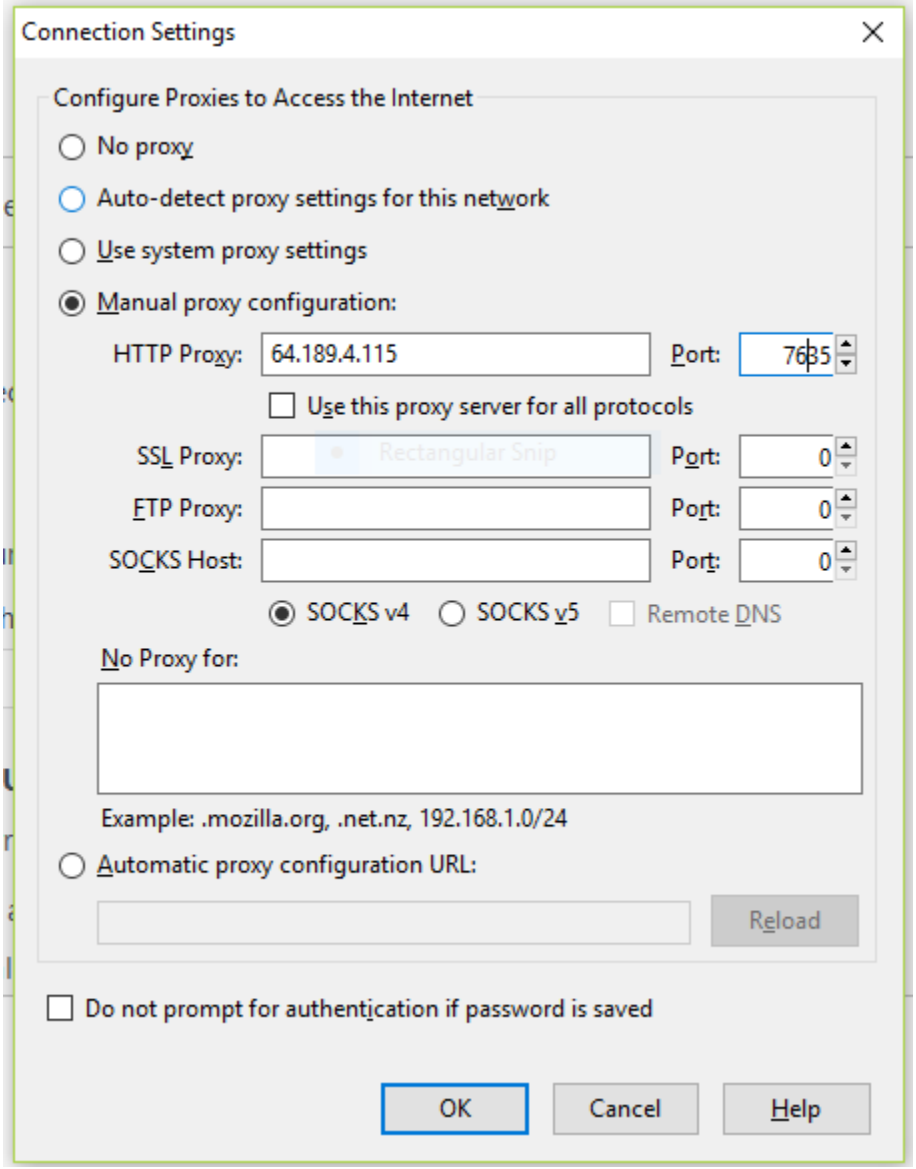
- Also, if the program is compiled and run without the client providing a port number, then the server is assigned to the **DEFAULT** port number which is **8080**



- Waits for the request from the client i.e Web Browser

Configuring the Browser to set up Proxy:

- Any Browser such as Chrome, Firefox, Internet Explorer can be used to set up proxy connection.
- Open the Firefox browser, click on **Settings -> Advanced -> Network -> Connection Settings**
  For example, In Firefox, in the Connection Settings window enable the **manual proxy settings radio button** and in the **HTTP Proxy**, type the **Public IP address of the system.** Eg: 64.189.4.115
  In the port field, type the port number. Eg: 7635
  If the client dint specify any port number, then set the port number to 8080.
  Click OK to finish configuring. Now the browser acts as the client.



- Enter any URL in the address bar to request for the web page.
  Eg.: http://www.google.com

- HTTP Request is sent to the Proxy Server
  - If the requested web page is available in the Proxy Server's cache, then it returns the page.
  - If the requested page is not available in the cache, it forwards the request to the web Server.
    The requested page is then returned.

Implementation:

- The Proxy Server must be compiled and executed.
- Once it receives a request from client, TCP connection is established and the URL is parsed.
- All the information including HTTP Request method i.e GET method, Host name, Contenttype , (Accept field), Contentlanguage, Elapsed Date and Time is **extracted and displayed in Terminal.**
- The HTTP Request information in also stored in a "log.txt" file which gets generated during execution of Proxy Server.
- From the parsed information, the host address of each request is appended into the Cache.
- If the host address match with any of the existing address in Cache, the page is returned to Client with a HTTP Response message "HTTP/1.1 200 OK".
- If the address doesn't exist, the request is then forwarded to Web Server and the requested file is received at Proxy server with the same HTTP Response message.
- TCP connection is established each time client requests for a web page. And a separate thread handles each request.
- Connection is kept alive until all connections are closed and thread is stopped.

Error Handling:

In Browser:

- If Client enters a URL that is not available, Eg.: w ww.djfgvnaerg.com, The Server responds with a HTTP Response Error message "HTTP/1.0 404 Not Found"
- If Client enters a URL with the port number in browser, Eg.: http://localhost:8080/www.google.com, then Proxy Server responds with an error message "Proxy server cannot process request from WWW browser". This error is because the port is already configured in the proxy settings of the browser and it cannot be explicitly specified again.

In Terminal:

If client requested object is not available, HTTP Response Error message "HTTP/1.0 404 Not Found" is displayed in Terminal as well.

RTT:

RTT for the client request is calculated as 2RTT + Transmission Time. The RTT is lesser when the response is returned from the proxy server than the actual server. Using proxy server improves the performance as it stores the frequently visited web pages.

References:

http://www.jtmelton.com/2007/11/27/a-simple-multi-threaded-java-http-proxy-server/

http://www.tutorialspoint.com/java/java_networking.htm

https://abet.soe.ucsc.edu/sites/default/files/ce155-middle.pdf

http://www.cs.princeton.edu/courses/archive/spr08/cos461/web_proxy.html

http://www.cs.princeton.edu/courses/archive/spr10/cos461/assignments-proxy-thread.html