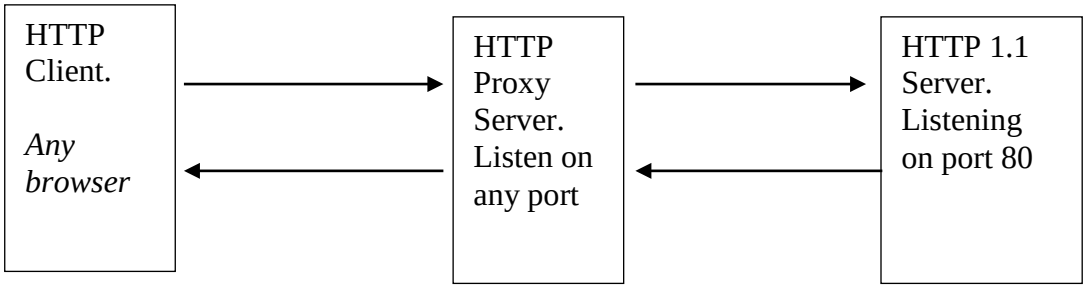
CSci 4131: Internet Programming
Fall 2015
Assignment 7: Building an HTTP Proxy Server in Java
Due Date: December 4, 2015
This assignment is to be done either individually OR in a group of two.

Objective:
The objective of this assignment is to learn HTTP protocol (HTTP1.1) and build a simple HTTP proxy server. In this assignment you would learn how to program using Java and TCP sockets the functionalities of an HTTP client and server.
You will need go through RFC 2616 for HTTP 1.1 protocol. It would helpful for you to first read the paper Key Differences between HTTP/1.0 and HTTP/1.1

Introduction:
When a web client (such as Firefox/Chrome/Safari/Internet Explorer) connects to a web server (such as www.gnu.org) the interaction between them happens through the HTTP protocol. The messages between them can be routed through a proxy server, which acts as an intermediary between the client and the server. In this assignment we will build an http proxy server. The functioning of an HTTP proxy server is as shown in the following figure and explained below:



Working of the HTTP Proxy server in brief:
1. The proxy server acts like an HTTP server to the client and as a client for an HTTP server.
2. An HTTP client connects to the proxy server instead of directly connecting to the origin server. Most of the web browsers can be set to connect to the Internet through a proxy server.
3. In order that the client can connect to the proxy server, the proxy server has to keep on listening on a particular port that is known to the client.
4. Once the proxy server gets a request from the client, it opens a connection on port 80 of the HOST mentioned in the request. The proxy server forwards the client's request to the HTTP server on this connection.
5. The proxy server waits for the origin server to respond. When the proxy server receives the response, it forwards that response to the client.
6. Advantages of using a proxy server can be listed as follows:
    a. Access to certain websites can be blocked
    b. Access to certain type of content can be blocked (for example jpeg files).
    c. Files can be cached locally and the client requests can be satisfied from the local cache. (In this assignment you are NOT required to cache data in the proxy server.)

Problem Statement:
You are asked to write a proxy server in Java. The required functionality of the proxy server is as follows:
1. This proxy server will only handle GET and HEAD requests. For other requests it will simply return 406 (Not Acceptable) status code as response to the client.

2. It should block access to certain websites. The names of the websites will be specified in a configuration file. This file will also contain comments describing the filter. These comments and blank lines will be ignored by the proxy server.
   2.1 If the client request accesses one of the blocked websites then the proxy shall inform the client that the access is denied by returning status code 403 in the response message. The response from the proxy shall be in the form of a HTTP response.
3. It should block certain type of content. The type will be specified in the config file.
   3.1 If the response from the server contains one of the disallowed types then the proxy shall not forward that content to the client.
   3.2 It will create and send a new response message with status code 403 to the client.
4. The proxy server will be continuously writing a log file, recording all interactions with the clients and servers.
   4.1 The headers of all requests received from the client will be written to logfile.
   4.2 The headers of all response messages will be written to the logfile.
   4.3 It will also write if a request was allowed or disallowed, with information about the host name and the content type in the request.
   4.4 It will log if a response contained a disallowed content type and was dropped.
5. The proxy server will be multi-threaded, so it can handle multiple requests simultaneously.

Example:
Suppose that a file named config.txt contains the names of the domains and the names of the content-type that is blocked by the proxy in the following format:

```
# Filename: config.txt
# Description: This is the config file for proxy server
# Author: <your name>
# Date: November 18, 2015

# block complete access to the site
www.badsite.com  *
# block complete access to the site (missing * defaults to 'block complete access' as above)
www.badsite.com
# block all gif files from www.microsoft.com
www.microsoft.com       image/gif
# block all images i.e. gif, jpeg, png, etc.
www.sco.com             image/*
```

Suppose you are working on one of the CSElab machine: kh2170-01.cselabs.umn.edu.
You will run your proxy-server from the command line as follows:

```
% java ProxyServer config.txt PortNumber
```

where, config.txt is the config file, and PortNumber can be some port, such as 50000, on which this server will listen for incoming messages.

Note on using port numbers:
Port numbers are divided into 3 ranges:
1. Well-known ports: 0 through 1023.
2. registered ports: 1024 through 49151
3. dynamic or private ports: 49152 through 65535 ***Use any port from the 3rd range.***

Setting up Browsers to Use Proxy:

Firefox:

Tools->Options->Advanced->Connection->Settings. Once you reach this window you will find four radio buttons. Click on "Manual proxy configuration". Then in the "HTTP Proxy" field give

the complete host DNS name of the machine on which you are running your proxy, in our case it can be "kh2170-01.cselabs.umn.edu". In the "Port" field enter the port number on which you are running your proxy.

Internet Explorer (IE):

For the IE browser the settings are as follows:
Tools->Internet Options->Connections->Lan Settings->Proxy Server

Chrome:

Tools>Settings> Show advanced settings> (Under Network Section) Change Proxy Setting>
LAN Settings > Select Checkbox for Proxy Server and go to > Advanced
In the table fill entry for HTTP :
DNS name for the host where proxy is running , and the port number

To test your program, you will be given one or more sample configuration files. During grading, we will test your program with these as well some additional test files.

PLEASE READ THIS: If you are running your proxy server on some CSELab machine, on some port say 50000, and try to connect to it from a browser running outside of CSELb network, then you will get some error, because connection requests to port other than some subset of well-known ports are blocked (refused) by the CSELab firewall admin policies. The best thing to do will be to run your proxy and browser in the same network environment or machine.

Once the above setup is complete, you can try connecting to any website through the web browser. Suppose we connect to www.gnu.org. Then you should be able to see that site without any problems. On the other hand if you connect to www.badsite.com then you should not be able to visit this site. Instead, you should get a message: "You are trying to visit a blocked site!" along with 403 status code.

Assignments Details:

1. You will have to read the following parts from the HTTP 1.1 (RFC 2616). The link for the RFC can be found on the class web page.
    a. HTTP Request format
    b. HTTP Response format
2. The HTTP client (eg: Fireofx) will send a complete HTTP request to the proxy server. You can pass the same request to the target host's HTTP server after you have extracted any information necessary for your purpose. Hop-to-hop headers such as Keep-Alive should be removed, unless your proxy server is designed to work with persistent connections.
3. Proxy server will have to extract the hostname of the HTTP server to establish a connection with it. It can extract the hostname from the *Host: header* field in client's request. Once the proxy server opens a socket connection to the HTTP server, it can pass the same request that it received from the client. The HTTP server will respond to proxy server's request with a HTTP response.
4. Proxy server will check the response headers if it contains a Content-Length header. If there is no such header, then the response may be a "chunked" response, where each chunk is preceded by its length. Hence you will have to read the response from the HTTP server appropriately in a while loop. If the request to the origin server contains "Connection: close" header, then you can rely on EOF to detection that the entire response message has been received, but this can be misleading if the connection snaps due to some failures.
5. Record the information about the requests in a Logfile.
    a. All request and response headers must be logged.
    b. If the request is denied because of a disallowed domain or disallowed Content-Type then the logfile should contain entries like this:

    www.cnn.com::blocked
    www.gnu.org/tux.gif::File type not allowed

Make sure the writing to the logfile by concurrent threads is properly synchronized. Otherwise the logfile will be garbled and meaningless.

Useful Java Classes:
Here is a list of Java classes that may be useful for this Assignment.

- Socket, ServerSocket, URL
- Check the online Java documentation for additional information.
- Lecture Notes #12 are also useful. Specifically, check the following example programs:
    a. HTTPget program
    b. EchoServer program
    c. HTTPTunnel program (This is a good example for you to start on this assignment,)

Things to submit:
Submit a tar file with the name: assignment7.tar

The tar file (assignment7.tar) should contain:
1. ProxyServer.java
2. Any other Java files that your program may be using
3. Include a sample config.txt which you used for your testing.
Include a README file if there are specific instructions that need to be followed to execute your proxy server.

Please include the following the information in your ProxyServer.java file:
1. Names of all students
2. Student IDs of all students
3. README. This file should contain any specific instructions for running your proxy.

Grading Criteria:

• Successful access to unblocked websites	20%
• Successfully block the website a. block all content (10%) b. Selective certain file types (e.g. img/png) (10%)	20%
• Correctly printing allowed/disallowed status	15%
• Correctly printing request headers	15%
• Correctly printing response headers	15%
• Correctly logging to the Logfile	10%
• Coding style and comments	5%

Note on Comments:
Please comment your code sufficiently. Up to 5% of the points are allocated for coding style and comments