CPSC 441: Computer Networks
Assignment 1
due Feb 3, 2017

Important Notes

- This an individual assignment. You should submit your own work.
- The assignment is due 11:59pm on February 3, 2017.
- Start the assignment early and avoid procrastination.
- The programming language for this assignment is Java.

Objective

The objective of this assignment is to practice socket programming and learn HTTP application layer protocol. Specifically, you will be implementing a Web Proxy that accepts requests from HTTP clients for various objects in the Internet and responds back.

Overview

You will develop a simple web proxy server. This proxy server only understands GET requests. The proxy server should support non-persistent HTTP (as in HTTP 1.0). This means that once an HTTP request is served, the proxy server closes the TCP connection. To inform the client that the connection is closed, include the header line "Connection: close" in the server response. You may need to use the same header line while forwarding the client request to the origin server where the object requested resides, in case local cache has no copy of the object. Your web server can be iterative, i.e., only need to handle one connection (or request) at a time.

Proxy Behaviour

Following features need to be implemented in your HTTP proxy server:

- Proxy server accepts connection from HTTP clients like web browsers.
- Reads the incoming GET HTTP request and check if the object requested is available in the local cache. If available, the object is served from the local cache. You may find class *File* useful to check if a file is present. Else, object is requested from the origin server where the object resides on behalf of the client and returns the same to the client once received from the origin server. Your web proxy acts both as a client and a server. It is a server when it communicates with web clients and a client when it communicates with origin servers.

- A local copy of the object is created so that any future request for the object is served from the local cache.
- When objects are locally stored, the file name should match the *hostname/pathname* of the URL in the GET request from the web browser, with the same directory structure being created locally inside the directory where your program is running.

  Note: Whenever web browser sends request to a web proxy, it uses the full URL instead of just pathname in the GET request line. The URL format in the GET request of a browser would be,

  protocolidentifier://hostname/pathname

  Example,

  http://pages.cpsc.ucalgary.ca/∼cyriac.james/sample.txt

  where,
  protocolidentifier = "http",
  hostname = "pages.cpsc.ucalgary.ca" and
  pathname="∼cyriac.james/sample.txt"
- For requests other than GET, web proxy should return response with status code "400 Bad Request"
- For response other than "200 OK" from the origin server, send "400 Bad Request" response to the client.

Assumptions

- Only well formatted GET requests are received by your proxy. But, if request is not GET, then proxy needs to send "400 Bad Request" response.
- The proxy cache always has the latest version of the object once downloaded from the origin server (i.e objects are never updated at the origin server). So, you DON'T need to implement "Conditional GET"
- All requests are for a single object in the Internet (i.e., you can IGNORE the case of base html and embedded objects in it)
- Web clients (like web browsers) and Web proxy run in the same machine.

Program Structure

The skeleton code is provided to you, `WebProxy.java`. Your task is to complete the `WebProxy.java` class. You may introduce additional classes and methods as needed (so, there is no limit on your creativity), but must keep the signature of existing methods unchanged.

Running the Proxy

Your proxy server can be run as `java WebProxy 8888`, where 8888 is the port number on which you want the proxy to listen for incoming connections from HTTP clients.

Testing Your Program

To test your program, you need to configure your web browser to use your proxy. If you are using Internet Explorer, you can set the proxy in *Internet Options* in the Connections tab under LAN Settings. If you are using Netscape (and derived browsers, such as Mozilla), you can set the proxy in *Edit/Preferences* and then select *Advanced and Proxies*. You will then enter the IP address of the proxy as "localhost" and the port number. You can now run the proxy and browser on the same computer without any problems. **Make sure you clean your browser cache after each webpage access**.

Alternatively, you can also test your program by sending HTTP request using a telnet session. TAs will explain how to do this in one of the tutorials.
Few sample URLs which you can use for testing are:

http://pages.cpsc.ucalgary.ca/~cyriac.james/sample.txt and
http://pages.cpsc.ucalgary.ca/~cyriac.james/readlist.pdf

Restrictions

- You are not allowed to change the signature of the methods provided to you
- You are not allowed to use the class URL or URLConnection for this assignment
- Ask the instructor if you are in doubt about any specific Java classes that you want to use in your program