**14740 – Project 2**

**Venkatesh Sriram (vsriram@andrew.cmu.edu)**

**Ramya Balaraman (rbalaram@andrew.cmu.edu)**

**To Test Multi Threaded Server**

1. Run edu.cmu.multithreadedserver Server.java **from Eclipse** and follow the Prompts to enter input.

Note: Do not enter inputs as command line arguments.

1. Enter path to www root as follows

/Users/ramya/Documents/Softwares/Ramya\_Programs/Project2-Telecom

**Note: No slash at the end**

1. Set Max connections and Max connections per client at least greater than 10.

Server listens at the specified port and creates a new Thread (ProxyServer) for servicing the request. The server then continues to listen for new connections.

**Test using browser**

1. localhost:<port>
2. localhost:<port>/
3. localhost:<port>/index.html
4. localhost:<port>/abcd.html

**Errors handled:**

HTTP 400, 404, 500, 501, 503, 505

**Bonus:**

**Handling DOS attacks:**

The maximum number of allowable connections per client has been limited to a number which is taken as input while starting the server. This is done so that no particular malicious client can hog the server’s resources. The server will refuse further connections if the limit has been reached on the maximum allowed connections per client. Similarly, the total active connections has also been limited so the server can efficiently serve outstanding requests instead of being slowed down by more worker threads than t can handle.

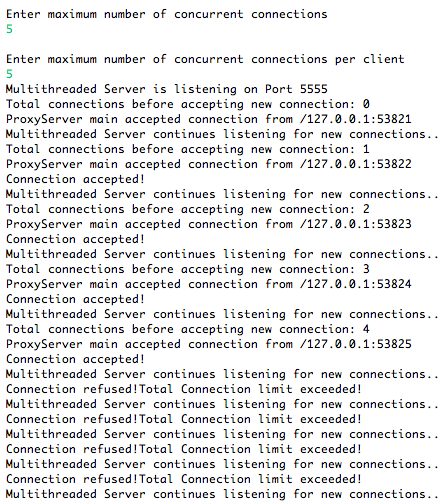
To test:

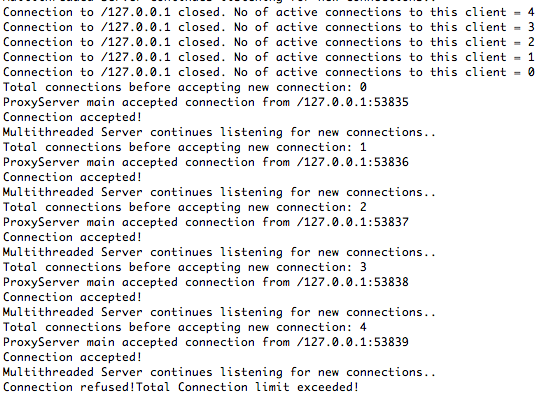
1. Set Maximum no of concurrent connections &

Maximum no of concurrent connections per client: 5 while starting the server

1. Run Client.java (has been programmed to send 12 non-HTTP messages in a loop)
2. Client receives a response for 5 requests after which it receives a ‘503 Service Unavailable’ response as the server cannot accept anymore requests currently.
3. The ProxyServer thread has been made to sleep for a while before closing for testing purposes.
4. After the ProxyServer is done servicing some requests and space is freed up, it can accept more connections.
5. Running the client program again shows that some more connections are accepted.

Server output is attached below:





Client Output is attached below:

