

CN Assignment: Proxy Server

Deadline : 31st March, 2019

Aim

You have to build a multithreaded HTTP proxy server, which can handle and serve many requests, using sockets.

Problem Statement

You are a sysadmin in IIIT and are asked to set up a proxy server for the institute. You have a simulated IIIT network on your local machine, which consists of several python SimpleHttpServers. You have to build a proxy server which is capable of servicing requests from clients through sockets, and allowing clients on the inside to access outside servers.

Assume the server runs on port 20100 of your machine. The ports 20000-20099 are the IP addresses of systems inside IIIT, and 20101-20200 are servers outside IIIT. The clients inside IIIT send HTTP requests to the proxy server. The proxy server must send those requests to the destination IP (the destination can be any server on the intranet or on your machine), and get the response, which must be passed on to the initial requester. On receiving a request, the proxy must create a new thread. The new thread will service the request, while the parent continues to listen.

Features

- Threaded Proxy server (8 marks)
- The proxy must keep count of the requests that are made. If a URL is requested more than 3 times in 5 minutes, the response from the server must be cached. In case of any further requests for the same, the proxy must utilise the “If Modified Since” header to check if any updates have been made, and if not, then serve the response from the cache. The cache has a memory limit of 3 responses. (8 marks)
- The proxy must support blacklisting of certain outside domains. These addresses will be stored in “proxy/blacklist.txt” in CIDR format. If the request wants a page that belongs to one of these, then, return an error page. (3 marks)
- Handle proxy authentication using Basic Access Authentication and appropriate headers to allow access to blacklisted sites as well. The

authentication will be username/password based, and can be assumed to be stored on the proxy server. (6 marks)

Clarifications and Notes

- Using servers like *pyftplib* for Python is not allowed. Sockets must be used.
- You do not need to write code to create the other python servers or clients. This code will be posted on moodle.
- The server must be able to handle both GET and POST requests (No other types).
- **Note:** that all HTTP requests cannot be cached, and is dependant on the Cache-Control header.
- The server must only serve requests from within IIIT. Requests for files from outside must be rejected with appropriate responses.
- The languages permitted for this assignment are C, Python, C++.
- All error scenarios must be gracefully handled (Programs crashing during testing WILL be penalised).
- Plagiarism in any form shall not be tolerated (MOSS will be used) and a straight F grade for the course will be given.

Submission Format

Put ONLY your codes and no other files in a single folder, along with a README and compress this to <roll number1 rollnumber2>.tar.gz and upload it to moodle.