Lab 2: Using Sockets to build a HTTP Proxy server

50.012 Networks

Hand-out: September 22
Hand-in: September 29

1 Objectives

- Implement a simple HTTP proxy server in Python
- The server will cache web pages, including objects such as images
- The server will only implement a subset of HTTP, the GET requests
- For a tutorial on sockets in Python, have a look here: `https://docs.python.org/2/howto/sockets.html`

2 Introduction

Generally, when an HTTP client makes a request, the request is directly sent to the web server. The web server then processes the request and sends back a response message to the requesting client. Proxies can be used to improve the performance of the web server and network, in particular if the same resource is requested often. When a proxy is used, the request message sent by the client and the response message delivered by the web server pass through the proxy server. The proxy server will forward the client's request to the web server. The web server will then generate a response message and deliver it to the proxy server, which in turn sends it to the client.

In this exercise, you will modify a skeleton proxy server to successfully act as simle, usable, HTTP proxy.

3 Skeleton code

We provide you with proxyServer.py, the skeleton code for the client. You are to complete the skeleton code. The places where you need to fill in code are marked with #Fill in start and #Fill in end. Each place may require one or more lines of code.

4 Using the Proxy

- You need to configure your web browser to use your proxy. The exact configuration depends on your browser.

  - In Internet Explorer, you can set the proxy in Tools > Internet Options > Connections tab > LAN Settings.
  - In Firefox, you can set the proxy in Tools > Options > Advanced tab > Network tab > Connection Settings. In both cases you need to give the address of the proxy (e.g. 127.0.0.1) and the port number that is used by your proxy server.

- Start your proxy on the command line, leave the command line open to observe logging output
- You should be able to run the proxy and the browser on the same computer without any problem.

With this approach, to get a web page using the proxy server, you simply provide the URL of the page you want. Simple websites should work directly

5 What to Hand in

5.1 eDimension submission:

You will submit in the complete proxy server code via eDimension. You can collaborate with a friend on the code, in that case please state both your names in the comments at the start of the file. Both students will then submit the file individually.

5.2 Checkoff:

- Demo your Python code to the TA and explain it. In particular, show how you retrieve webpages via the proxy by using your browser.

6 Extra features

The following extra features might be interesting:

6.1 Content modification

- Add a dictionary of words with replacements – everytime the original word is contained in the original cached HTML file, replace it with the new replacement word.
- It should be easy to replace complete websites with your own webpages, without the user noticing. Define a replacement webpage for a specific URL, and show that it works.