

CS3031 - Project 1

Paolo Moloney - 16325409

February 10, 2019

Contents

1	Specification	3
2	Implementation	4
3	Code	5

1 Specification

The objective of this project is to implement a proxy server with the following features:

1. Respond to HTTP and HTTPS requests, displaying them and the responses on a management console.
2. Handle websocket connections.
3. Block selected URLs via the console.
4. Cache requests locally to save bandwidth.
5. Handle multiple requests simultaneously.

2 Implementation

I chose to implement the proxy server in Python 3, using the following modules:

1. **socket**: provides low-level access to the BSD socket interface
2. **threading**: provides higher-level threading interfaces based on the low-level `_thread` module

The following diagram outlines the design decisions I made.

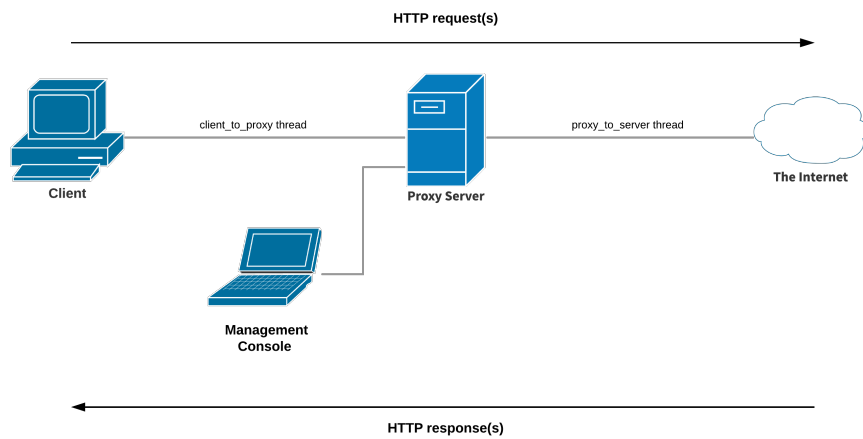


Figure 1: Proxy server diagram

3 Code

```
1 import socket
2 import threading
3 import sys
4
5 MAX_CHUNK_SIZE = 1024
6 PROXY_ADDR = '127.0.0.1'
7 PORT = 8080
8
9 num_threads = int(sys.argv[1])
10
11 class c2p_thread(threading.Thread):
12     # method for thread setup
13     def __init__(self, threadID, addr, port):
14         threading.Thread.__init__(self)
15         self.threadID = threadID
16         self.addr = addr
17         self.port = port
18     # method for thread execution
19     def run(self):
20         print('starting thread', self.threadID)
21         client_to_proxy(self.threadID, self.addr, self.port)
22         print('exiting thread', self.threadID)
23
24 def client_to_proxy(threadID, addr, port):
25     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
26     s.bind((addr, port))
27     while True:
28         s.listen()
29         conn, conn_addr = s.accept()
30         with conn:
31             print('connected by', conn_addr)
32             data = conn.recv(MAX_CHUNK_SIZE)
33             if not data:
34                 break
35             else:
36                 data_str = data.decode()
37                 print('received:', data)
38
39 try:
40     for t in range(num_threads):
41         thread = c2p_thread(t, PROXY_ADDR, (PORT+t))
42         thread.start()
43         thread.join()
44         break
45 except:
46     print('error: unable to start thread(s)')
```