

COMPUTER NETWORKS LABORATORY

By:
Nitish S
PES2201800368
5 'A'

WEEK – 2- Understanding Persistent and Non-Persistent HTTP Connections

Date: 7/09/2020

Objective: To understand persistent and non-persistent HTTP connections and corresponding

performance impact.

Create a web page with N (e.g. 10) embedded images. Each image should be of minimum 2 MB size. Configure your browser (Firefox) with following settings (each setting requires repeat of experiment)

- Non persistent connection
- 2 persistent connections
- 4 persistent connections
- 6 persistent connections
- 10 persistent connections.

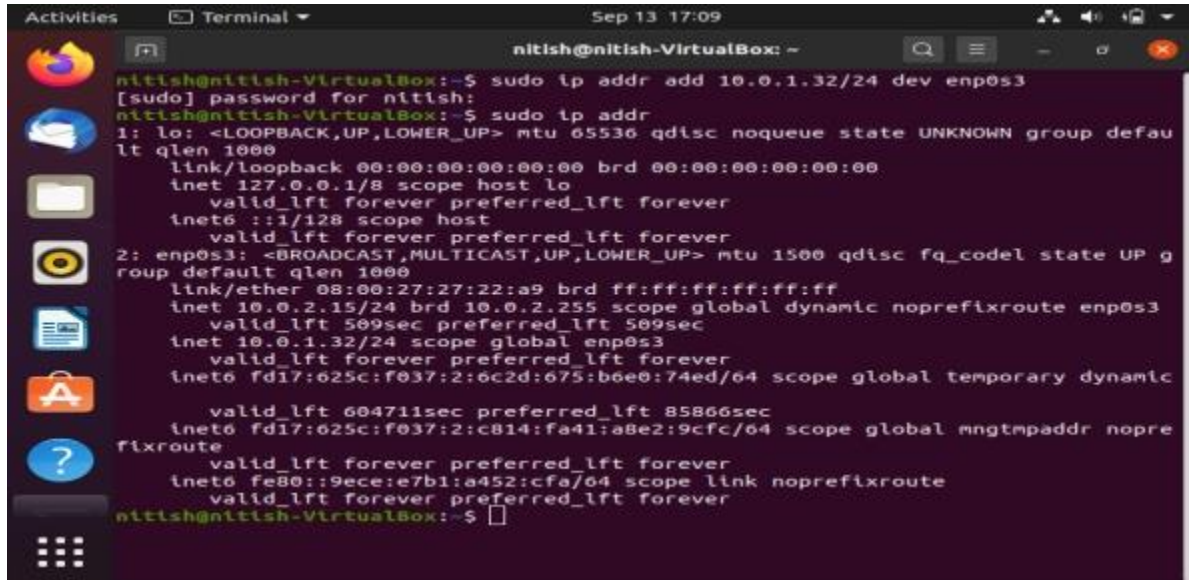
Observation: Note down the time taken to display the entire page in each of the settings.

Ensure that (cache is cleared before starting the web request). Explain the response time differences. What is the optimal number of persistent connections for best performance?

Explain your answer.

Initial Commands executed:-

- 1) `sudo apt-get install apache2` - To install apache2 server.
- 2) `sudo ip addr add 10.0.1.32/24 dev enp0s3` - To set the server IP address



```
nitish@nitish-VirtualBox:~$ sudo ip addr add 10.0.1.32/24 dev enp0s3
[sudo] password for nitish:
nitish@nitish-VirtualBox:~$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:22:a9:bd brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 509sec preferred_lft 509sec
    inet 10.0.1.32/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fd17:625c:f037:2:6c2d:675:b6e0:74ed/64 scope global temporary dynamic
        valid_lft 604711sec preferred_lft 85866sec
    inet6 fd17:625c:f037:2:c814:fa41:a8e2:9cfc/64 scope global mngtmpaddr noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::9ece:e7b1:a452:cfa/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
nitish@nitish-VirtualBox:~$
```

- 3) The **apache2.conf** file present in the **etc/apache2** directory is modified as:

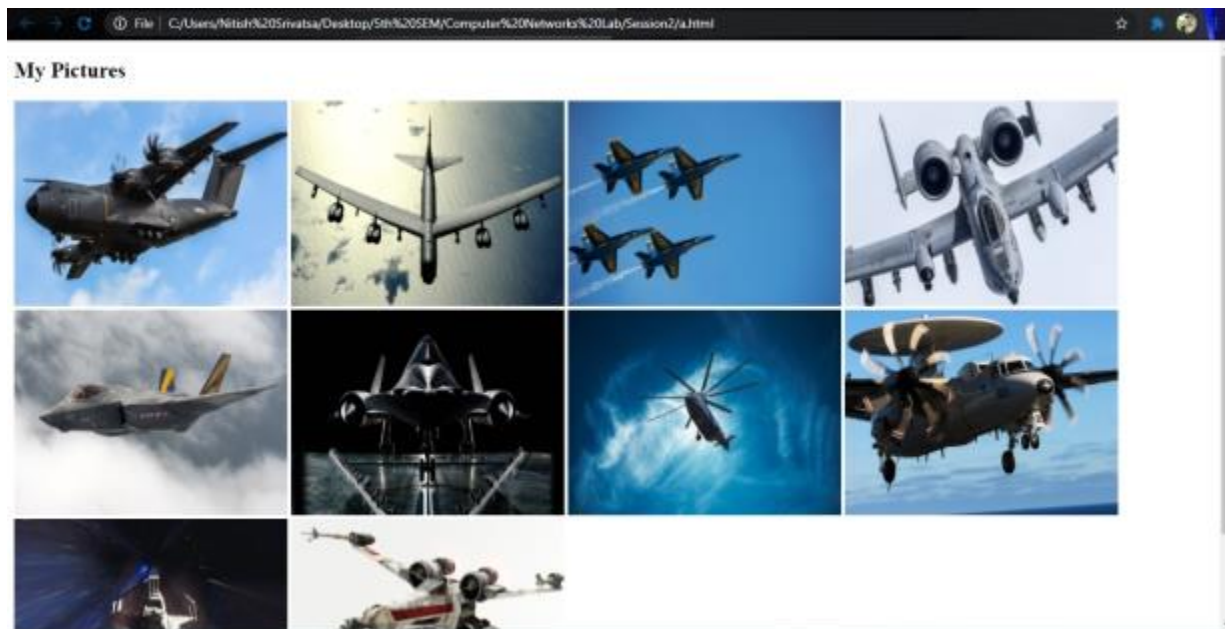
a) The **keep-alive** option was set (i.e. value was made **ON**)

b) The **MaximumKeepAliveRequests** were set to **2**

`sudo nano /etc/apache2/apache2.conf`

- 4) HTML file: Named as: **a.html** , stored in **/var/www/html/a.html**

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>My Pictures</h2>
6 
7 
8 
9 
10 
11 
12 
13 
14 
15 
16 </body>
17 </html>
```



CLIENT SIDE:-

sudo ip addr add 10.0.1.33/24 - To set the IP address in the client side

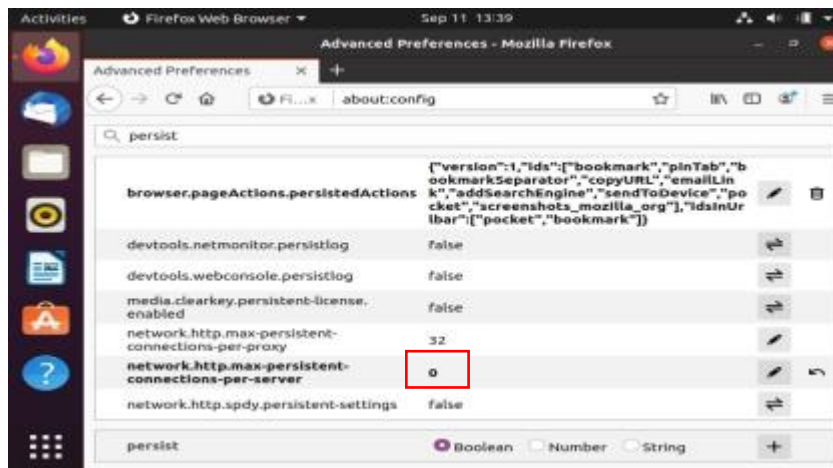
```

Activities  Terminal  Sep 13 17:04
nitish@nitish-VirtualBox: ~
nitish@nitish-VirtualBox:~$ sudo ip addr add 10.0.1.33/24 dev enp0s3
[sudo] password for nitish:
nitish@nitish-VirtualBox:~$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:57:91:24 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 399sec preferred_lft 399sec
    inet 10.0.1.33/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fd17:625c:f037:2:a138:46d7:afd8:2faa/64 scope global temporary dynami
c
        valid_lft 603702sec preferred_lft 85092sec
    inet6 fd17:625c:f037:2:1101:9106:f0a8:3659/64 scope global mngtmpaddr nopre
fixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::bb27:22fa:8150:af92/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
nitish@nitish-VirtualBox:~$

```

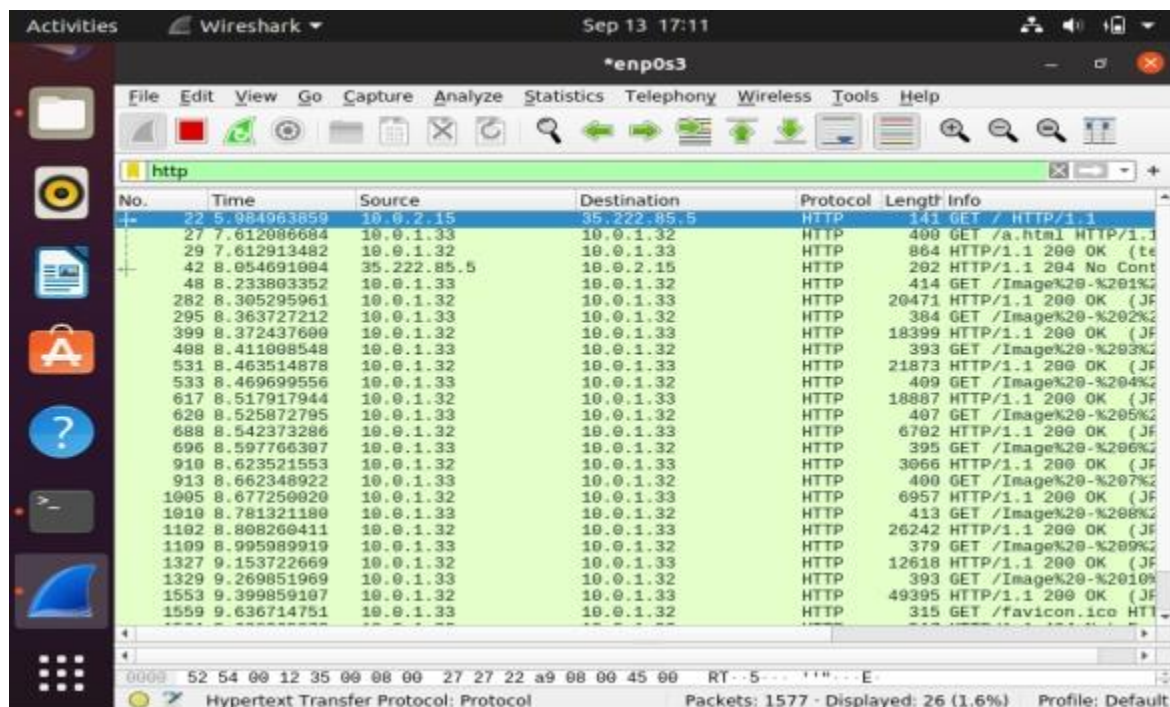
1) NON-PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 0 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.

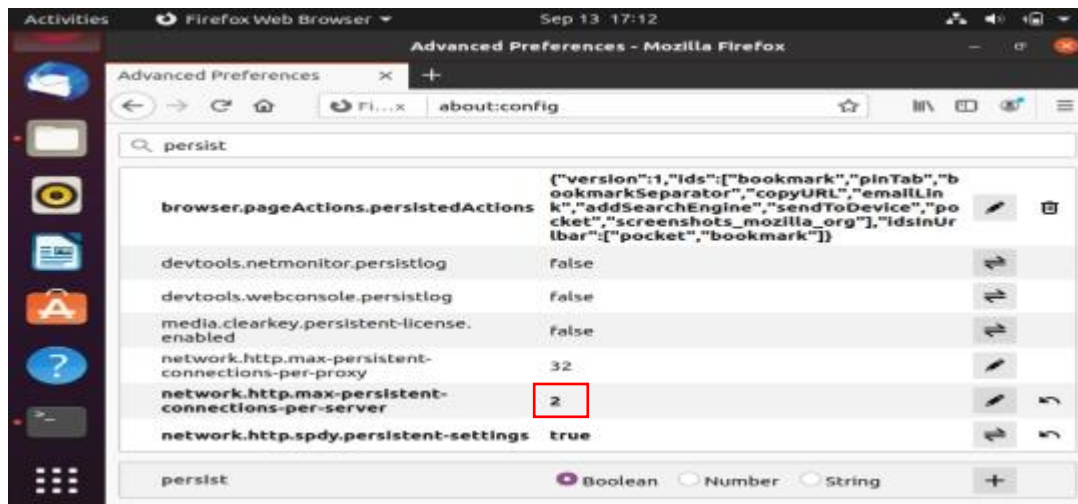


Time taken to capture all the 10 images as analyzed by wireshark for non-persistent connection:

$9.399859107 - 7.612086684 = 1.787772423$ seconds

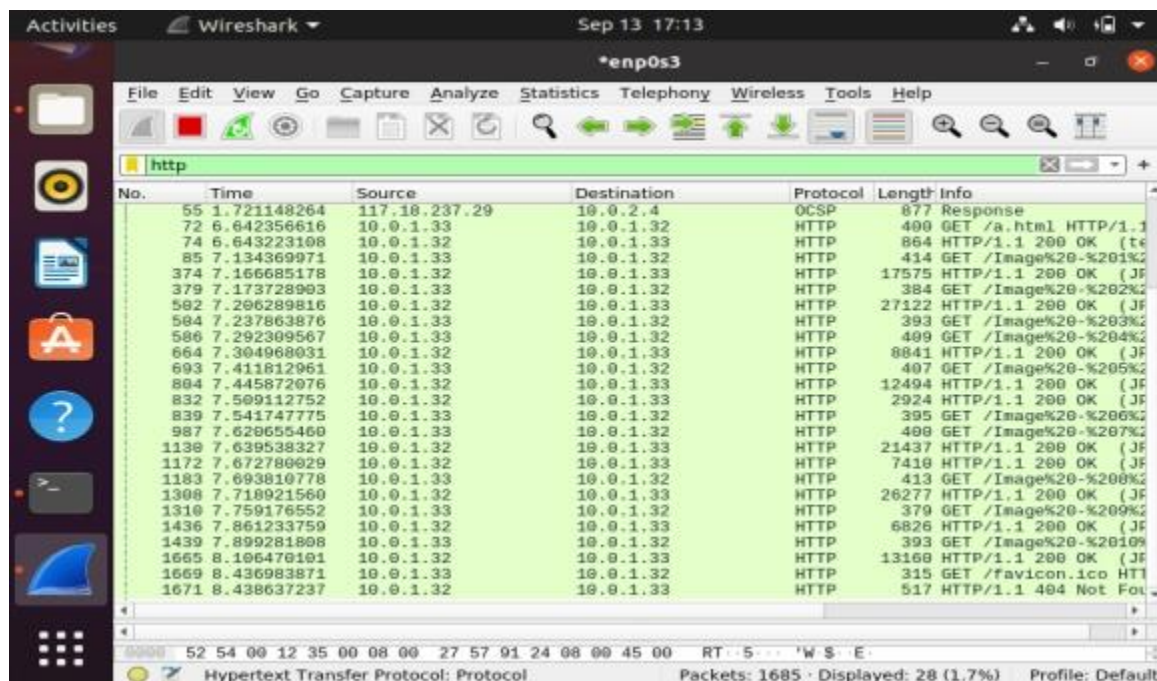
2) 2 - PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 2 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.

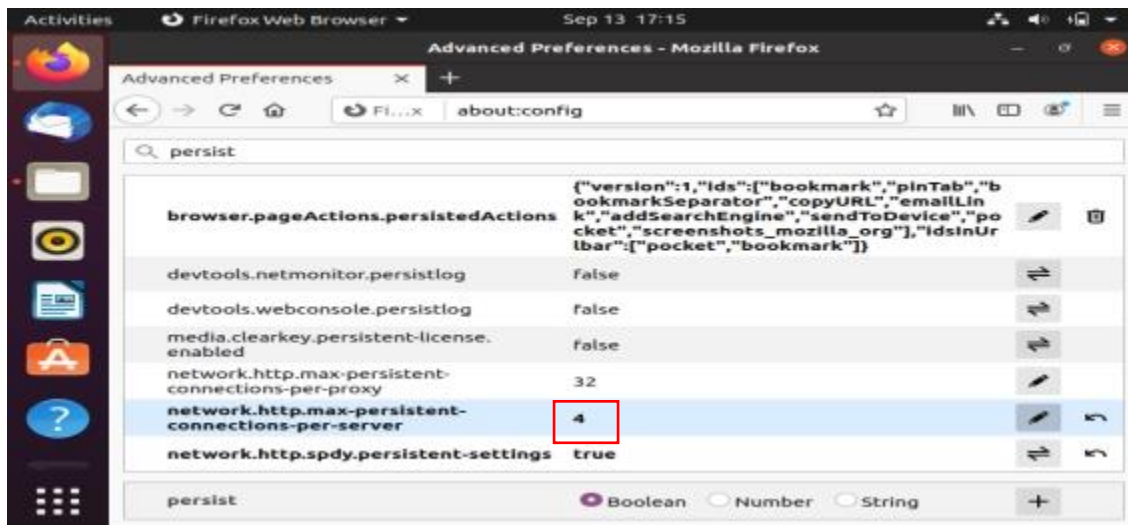


Time taken to capture all the 10 images as analyzed by wireshark for 2-persistent connections:

8.106470101 - 6.642356616 = **1.464113485 seconds**

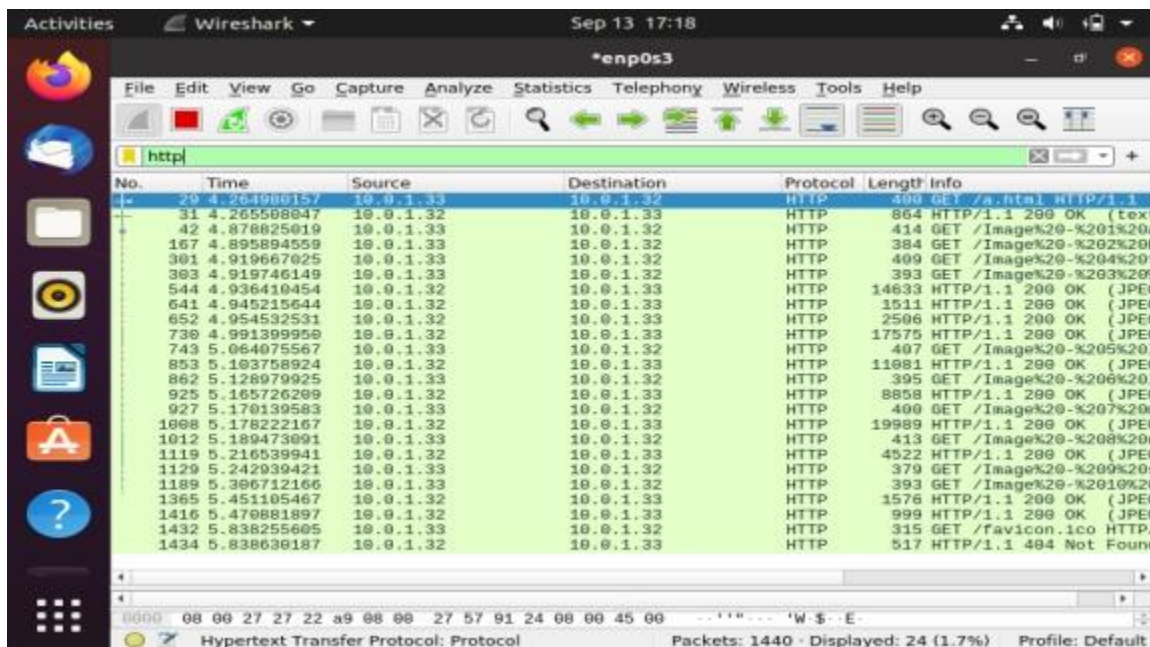
3) 4 - PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 4 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.

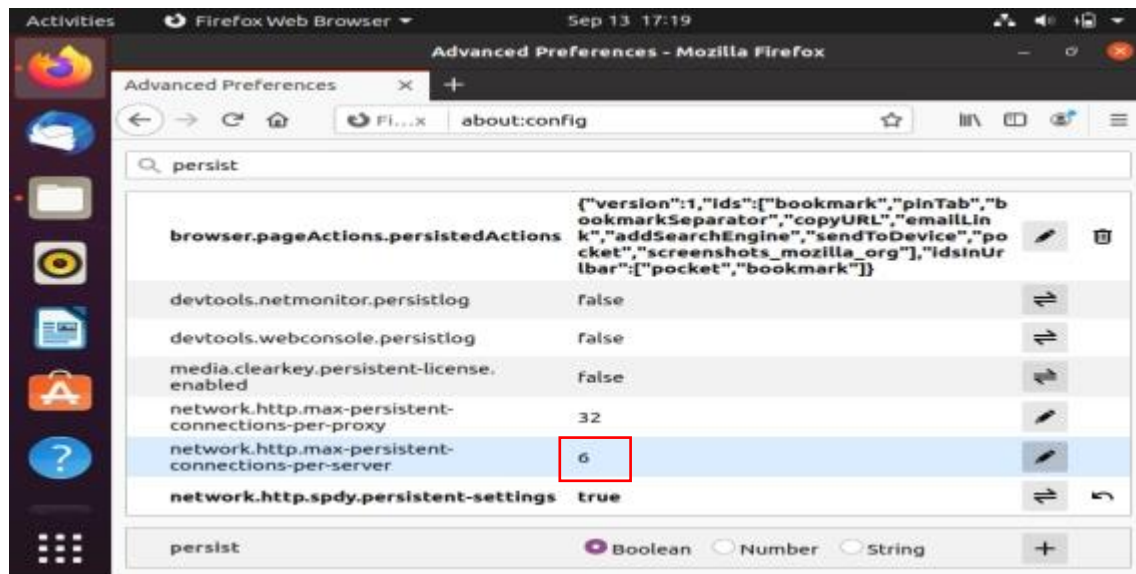


Time taken to capture all the 10 images as analyzed by wireshark for 4-persistent connections:

$$5.470881897 - 4.264980157 = \mathbf{1.213838813 \text{ seconds}}$$

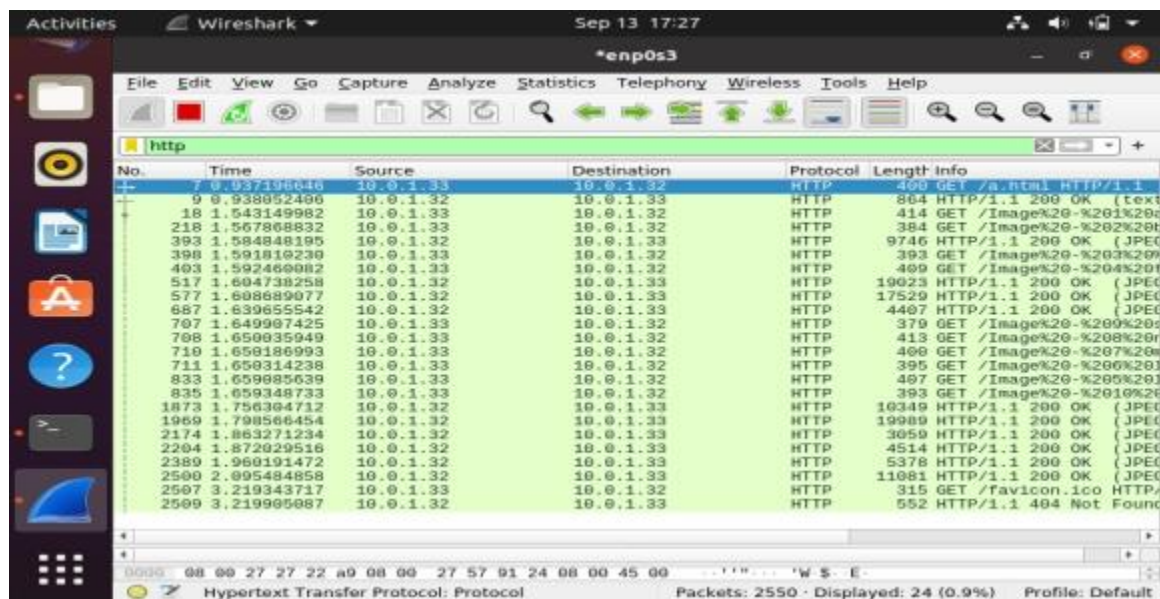
4) 6 - PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 6 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.

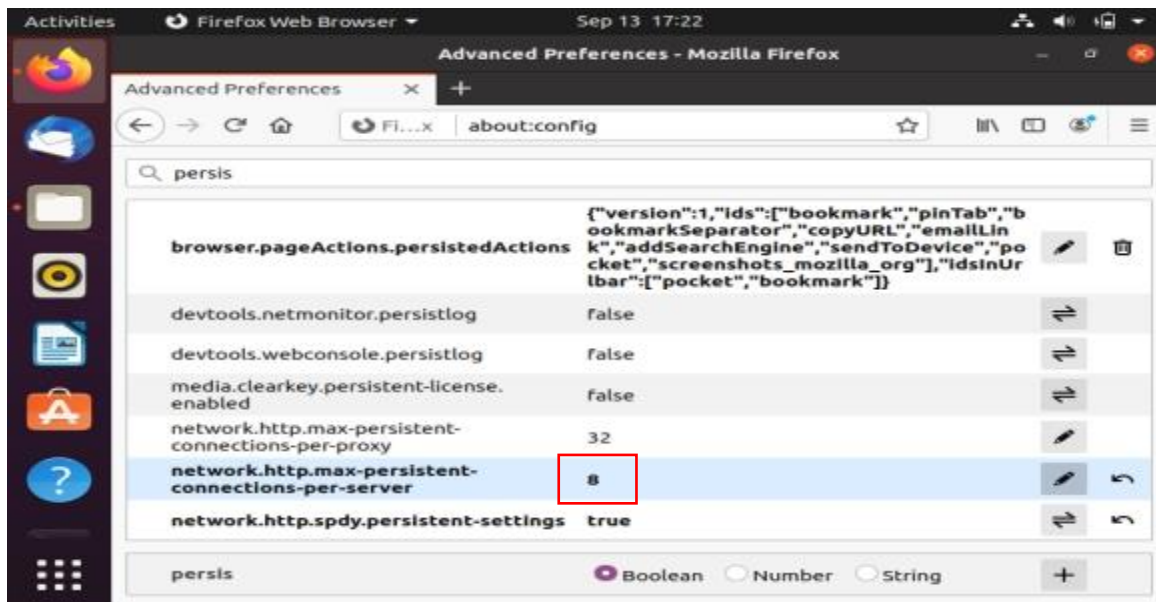


Time taken to capture all the 10 images as analyzed by wireshark for 6 - persistent connections:

2.095484858 - 0.937196646 = **1.158288212 seconds**

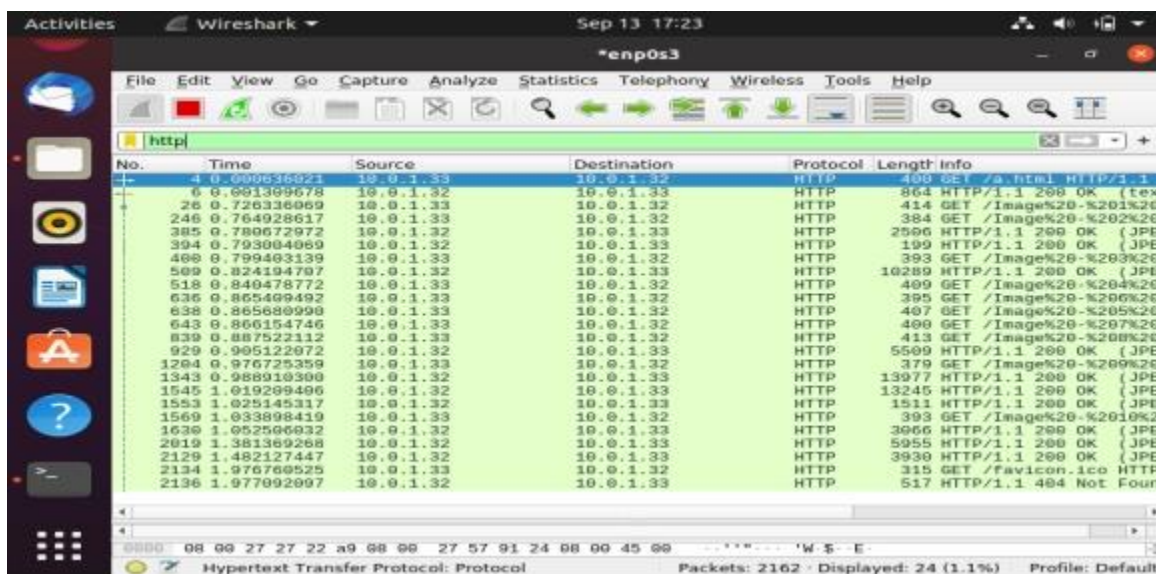
5) 8 - PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 8 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.

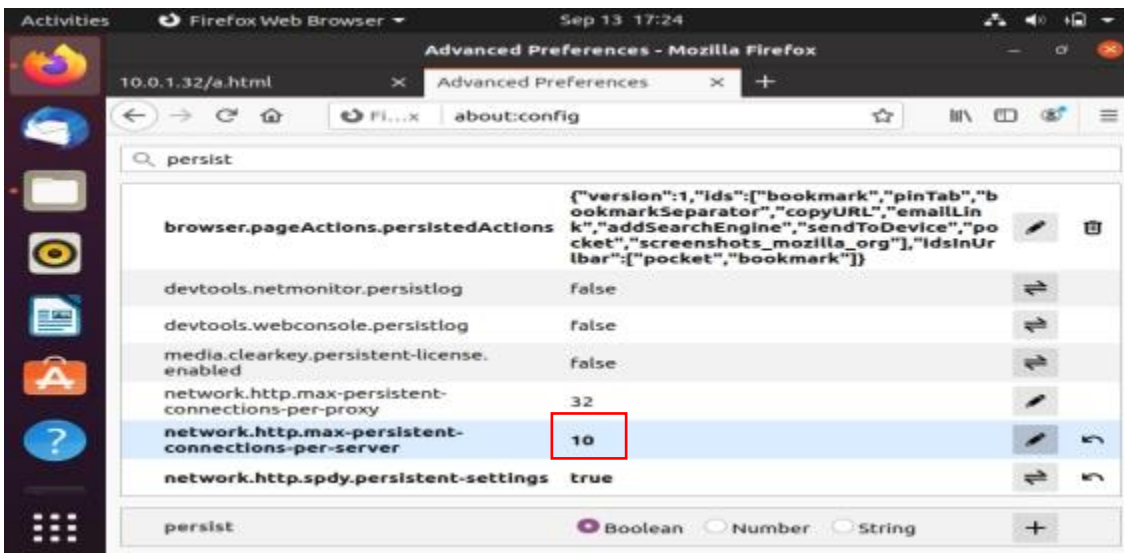


Time taken to capture all the 10 images as analyzed by wireshark for 8 - persistent connections:

$$1.482127447 - 0.000636021 = \mathbf{1.481491237 \text{ seconds}}$$

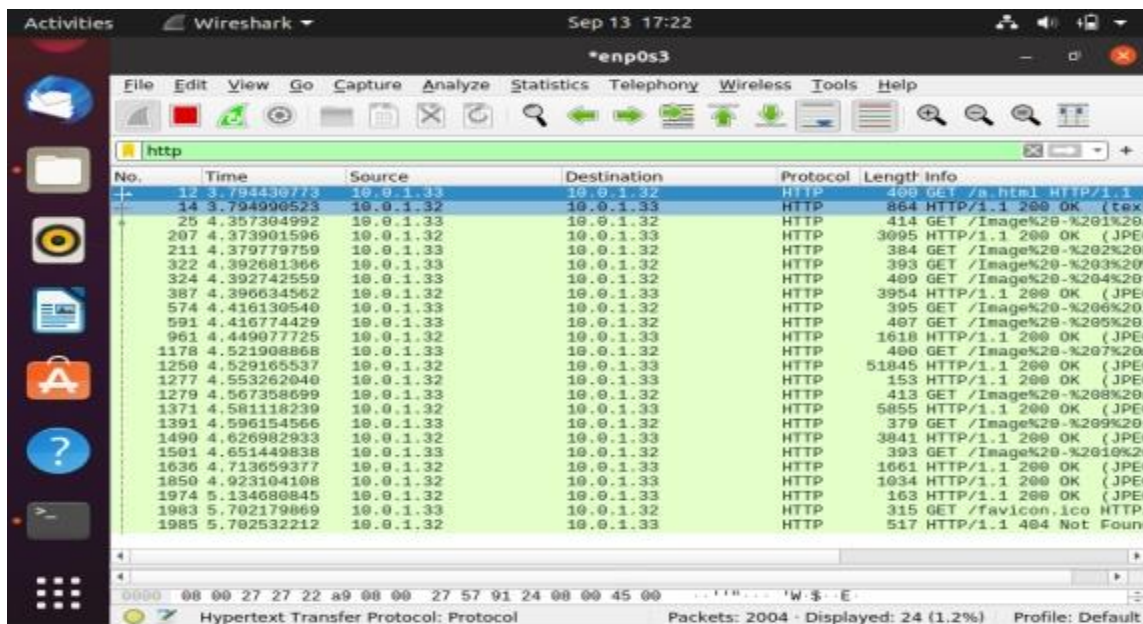
6) 10 - PERSISTENT CONNECTION:-

Setting the max-persistent-connection-per-server to 10 in the client computer.



Accessing the html file by typing **10.0.1.32/a.html** in the browser of the client computer.

sudo wireshark in the terminal to open wireshark tool and capture the HTTP packets.



Time taken to capture all the 10 images as analyzed by wireshark for 10 - persistent connections:

$$5.134680845 - 3.794430773 = \mathbf{1.340250072 \text{ seconds}}$$

Find out the time taken to load images for 2 4 6 persistent connections is lesser or greater than 10 persistent compared to non-persistent. Why? Find out the optimal persistent connections.

Time taken in seconds for non-persistent and 2,4,6,8,10 persistent connections are as follows:

- ◆ Non-persistent connections: 1.787772423
- ◆ 2- persistent connections: 1.464113485
- ◆ 4- persistent connections: 1.213838813
- ◆ 6- persistent connections: 1.158288212
- ◆ 8- persistent connections: 1.481491237
- ◆ 10- persistent connections: 1.340250072

The time taken for persistent connection is evidently less than the time taken for non-persistent connections. Further more, the time taken when the max-persistent-connections-per-server is set to 6 is the least. This is because for a persistent connection, the TCP connection is not closed after receiving 10 objects. On the other hand, for a non-persistent connection, the TCP connection is closed after sending each object. Hence the time taken is more.

The optimal no of persistent connections from the above analysis is 6.