



Name: Darshan Raju Loni

Srn: PES2UG22CS158

Sec: C

Week #3

Understanding Persistent and Non-persistent HTTP Connections

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.

Introduction

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features including dynamically loadable modules, robust media support, and extensive integration with other popular software.

Objective: Understand persistent and non-persistent HTTP connections and corresponding performance impact.

Experiment: 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)

- a) Non-persistent connection
- b) 2 persistent connections
- c) 4 persistent connections
- d) 6 persistent connections
- e) 10 persistent connections

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.

Note: To install Apache server, use the following command,

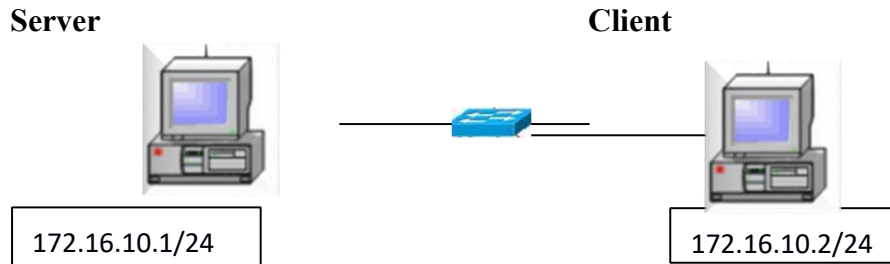
```
sudo apt-get install apache2
```

If there is any error during installation, update the package manager by issuing the command,

```
sudo apt-get update
```

EXECUTION STEPS

Step 1: Connect 2 desktops using switch and cables as shown below. (Use 2 VMs on Virtualbox or VMware instead of physical connections.)



Server: OSBOXES.ORG VM

Client: SEED UBUNTU VM

Server Side:

Step 2: Check your Web Server

At the end of the installation process, Ubuntu 16.04 starts Apache. The web server should already be up and running. We can check with the `systemctl` command to make sure the service is running by typing:

`sudo systemctl status apache2`

or

`sudo service apache2 status`

```
osboxes@osboxes:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enab>
   Active: active (running) since Thu 2024-02-15 10:43:31 EST; 22s ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 3189 (apache2)
      Tasks: 55 (limit: 4613)
     Memory: 5.3M
        CPU: 114ms
      CGroup: /system.slice/apache2.service
              └─3189 /usr/sbin/apache2 -k start
                 3190 /usr/sbin/apache2 -k start
                 3191 /usr/sbin/apache2 -k start

Feb 15 10:43:30 osboxes systemd[1]: Starting apache2.service - The Apache HTTP >
Feb 15 10:43:31 osboxes apachectl[3188]: AH00558: apache2: Could not reliably d>
Feb 15 10:43:31 osboxes systemd[1]: Started apache2.service - The Apache HTTP S>
```

As you can see above, the service appears to have started successfully. However, the best way to test this is to actually request a page from Apache. You can access the default Apache

landing page to confirm that the software is running properly. You can access this through your server's domain name or IP address.

Step 3: Server IP address can be set by the following command

\$sudo ip addr add 172.16.10.1/24 dev enps0

\$sudo ip addr

Note: If IP address fluctuates, kindly setup the IP address manually using 'Edit connections'.

```
osboxes@osboxes:~$ sudo ip addr add 172.16.10.1/24 dev enp0s3
osboxes@osboxes:~$ 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 pfifo_fast state UP group default qlen
    link/ether 08:00:27:ab:de:e1 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 86071sec preferred_lft 86071sec
    inet 172.16.10.1/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feab:dee1/64 scope link
        valid_lft forever preferred_lft forever
```

Step 4: 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**

```

PidFile ${APACHE_PID_FILE}

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 2

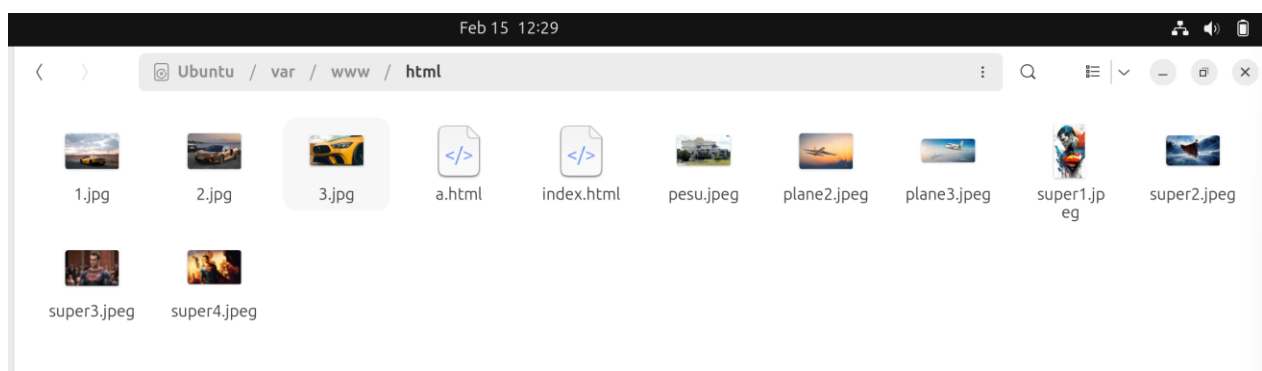
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

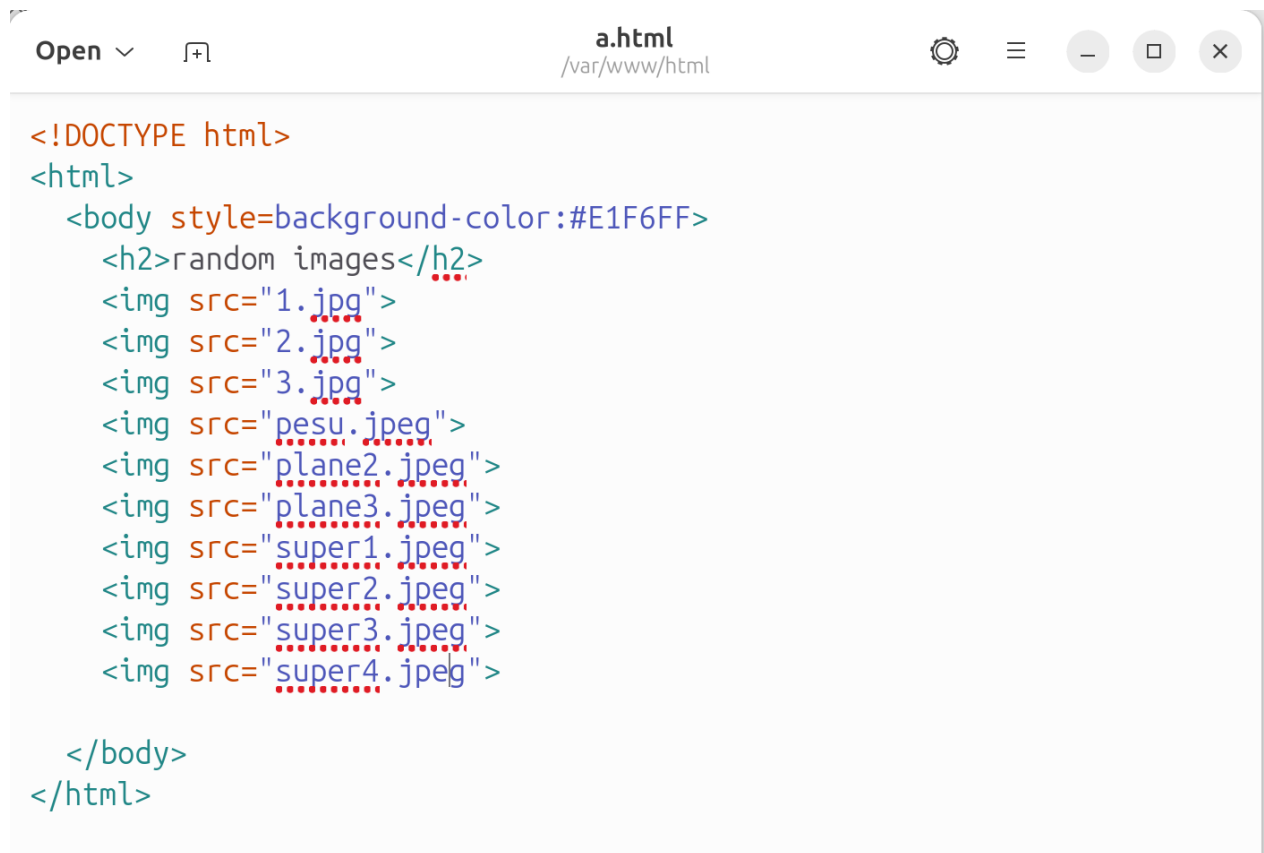
```

Step 5: Store images in the server path. A html page consisting of 10 images having size > 2MB were placed and accessed by the client. This html page is stored in the location - **/var/www/html/file_name.html**.

Note: Use the images provided by faculty incharges.



Step 6: Prepare a web page as shown below. The html file needs to add 10 images. (Kindly skip the style attribute in the below image)



```
<!DOCTYPE html>
<html>
  <body style=background-color:#E1F6FF>
    <h2>random images</h2>
    
    
    
    
    
    
    
    
    
    

  </body>
</html>
```

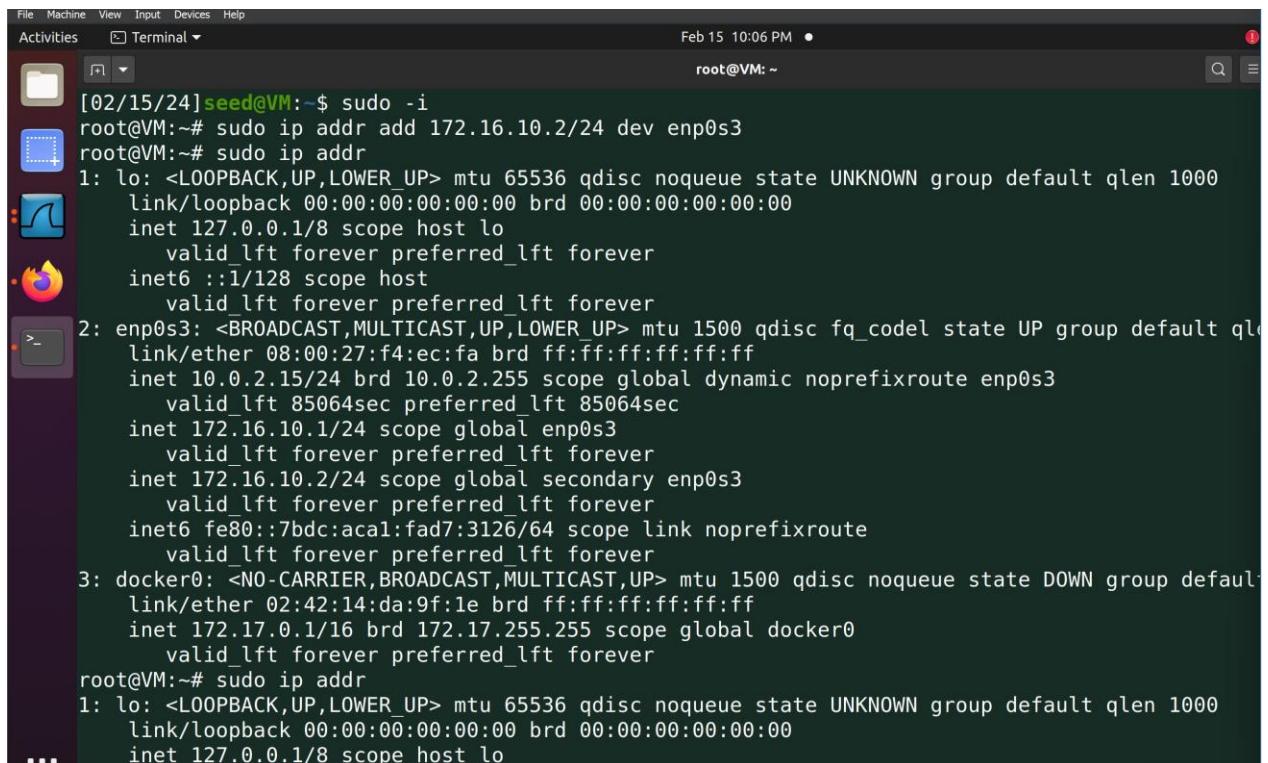
Client side:

Client IP address can be set by the following command.

\$sudo ip addr add 172.16.10.2/24 dev enps0

\$sudo ip addr

Note: If IP address fluctuates, kindly setup the IP address manually using 'Edit connections'.

A terminal window titled 'Terminal' with a dark background. The window shows a series of commands and their outputs. The user is root@VM: ~. The commands are: 'sudo -i', 'sudo ip addr add 172.16.10.2/24 dev enp0s3', and 'sudo ip addr'. The output shows the configuration of the loopback interface 'lo' and the ethernet interface 'enp0s3'. The 'lo' interface is configured with IP 127.0.0.1/8. The 'enp0s3' interface is configured with IP 10.0.2.15/24 and 172.16.10.2/24. The 'docker0' interface is also shown with IP 172.17.0.1/16. The terminal window has a menu bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help'. The status bar shows 'Feb 15 10:06 PM' and 'root@VM: ~'.

```
[02/15/24]seed@VM:~$ sudo -i
root@VM:~# sudo ip addr add 172.16.10.2/24 dev enp0s3
root@VM:~# 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:f4:ec:fa 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 85064sec preferred_lft 85064sec
    inet 172.16.10.1/24 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 172.16.10.2/24 scope global secondary enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::7bdc:aca1:fad7:3126/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 02:42:14:da:9f:1e brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
root@VM:~# 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
```

There are broadly two parts of execution:

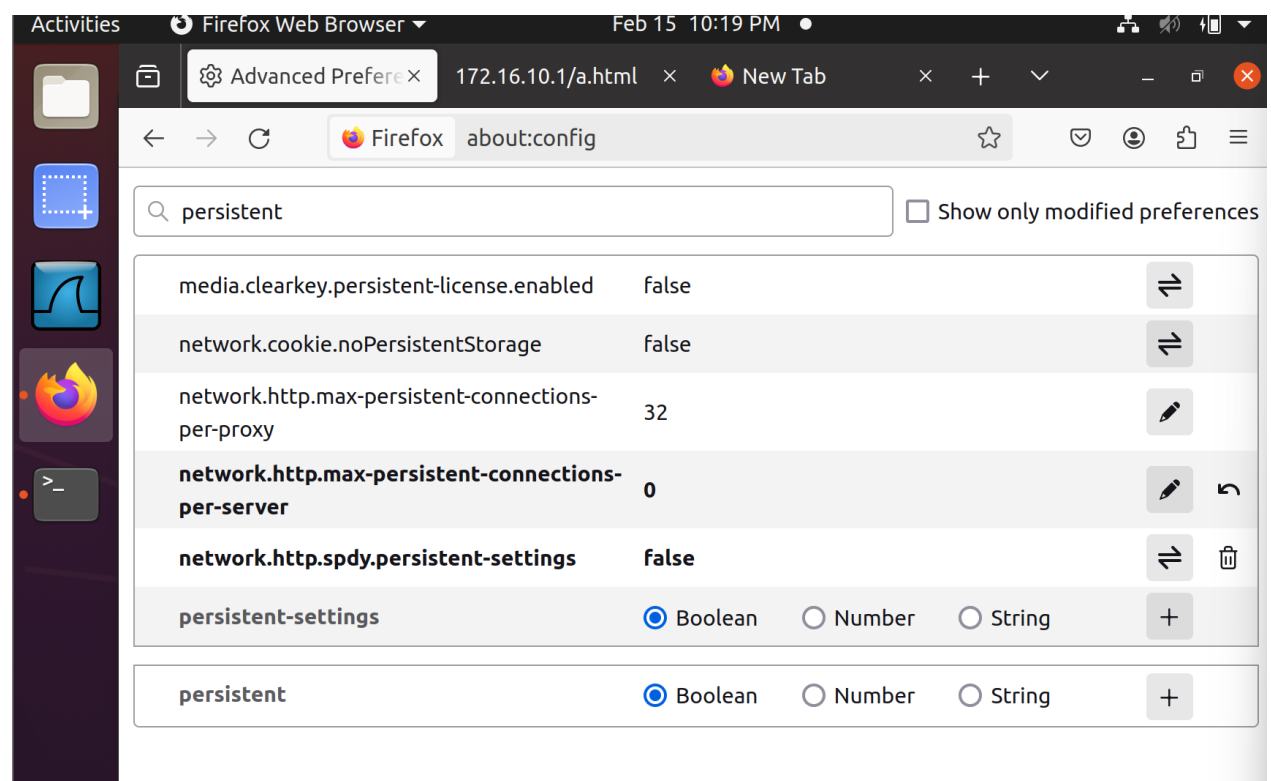
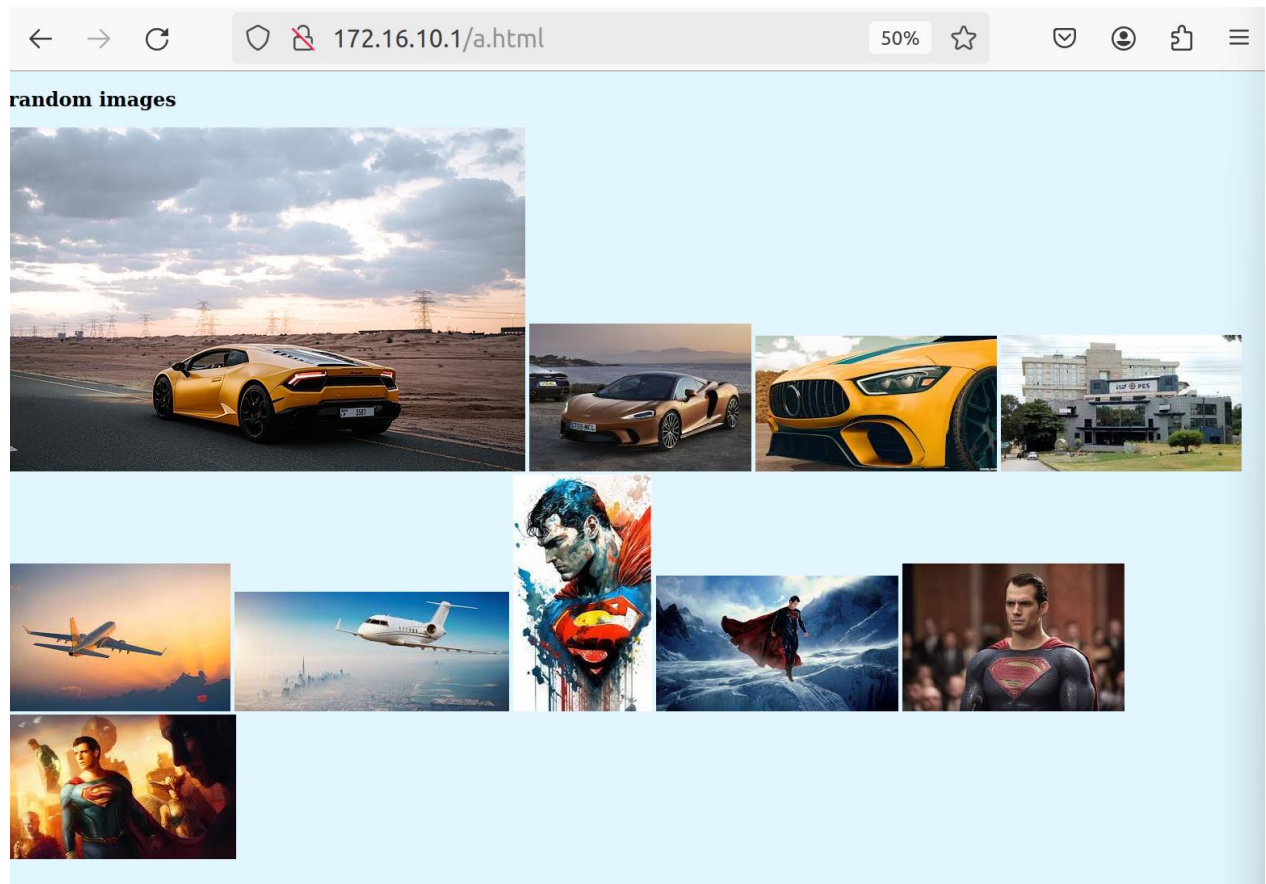
1. Dealing with non-persistent connections
2. Dealing with persistent connections

Open Firefox browser to configure for persistent option. Go to browser and type **about:config** and search for the term '**persistent**'

- While using non-persistent connection experiment, the **max-persistent-connectionsper-server** has the value set to **0** and **persistent-settings** value set to false.
- While using persistent connection experiment, the **max-persistent-connections-perserver** should have value greater than 0 (depending on the number of persistent connections needed) and **persistent-settings** value set to true.

PART 1: NON-PERSISTENT CONNECTION

Step 1: This is done by setting the value of max-persistent-connection-per-server to 0 in the client computer.



Step 2: Access web page on client-side browser (Firefox)

The client could access the file as:

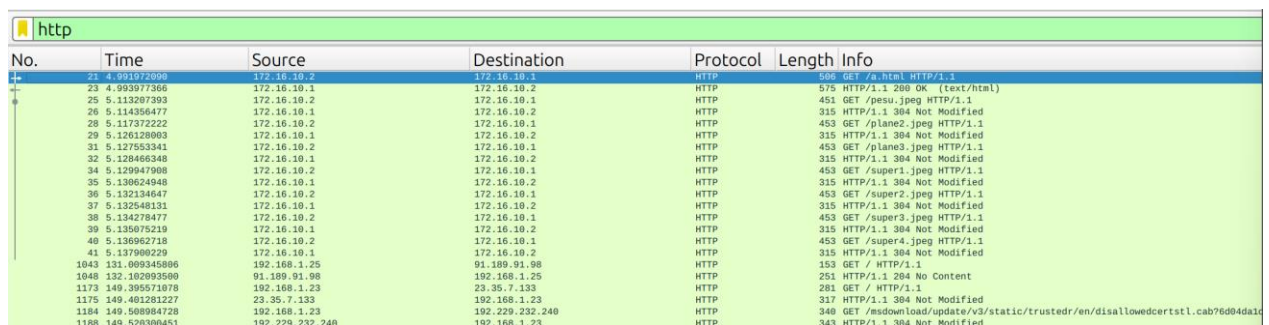
172.16.10.1/file_name.html where--> **172.16.10.1** is Server's IP

Here the file name is **a.html** present in server. So, by typing **172.16.10.1/a.html** in client browser, we will be able to open the requested web page.

Note 1: The Wireshark should capture the packets between the client and the server while the file is accessed.

Note 2: The images in the HTML page should have all the permissions specified through the server for the proper access.

Step 3: Use Wireshark. Open Wireshark in the server computer while client is trying to access the server's local host webpage. Apply 'http' filter and note the time to capture all the 10 images.

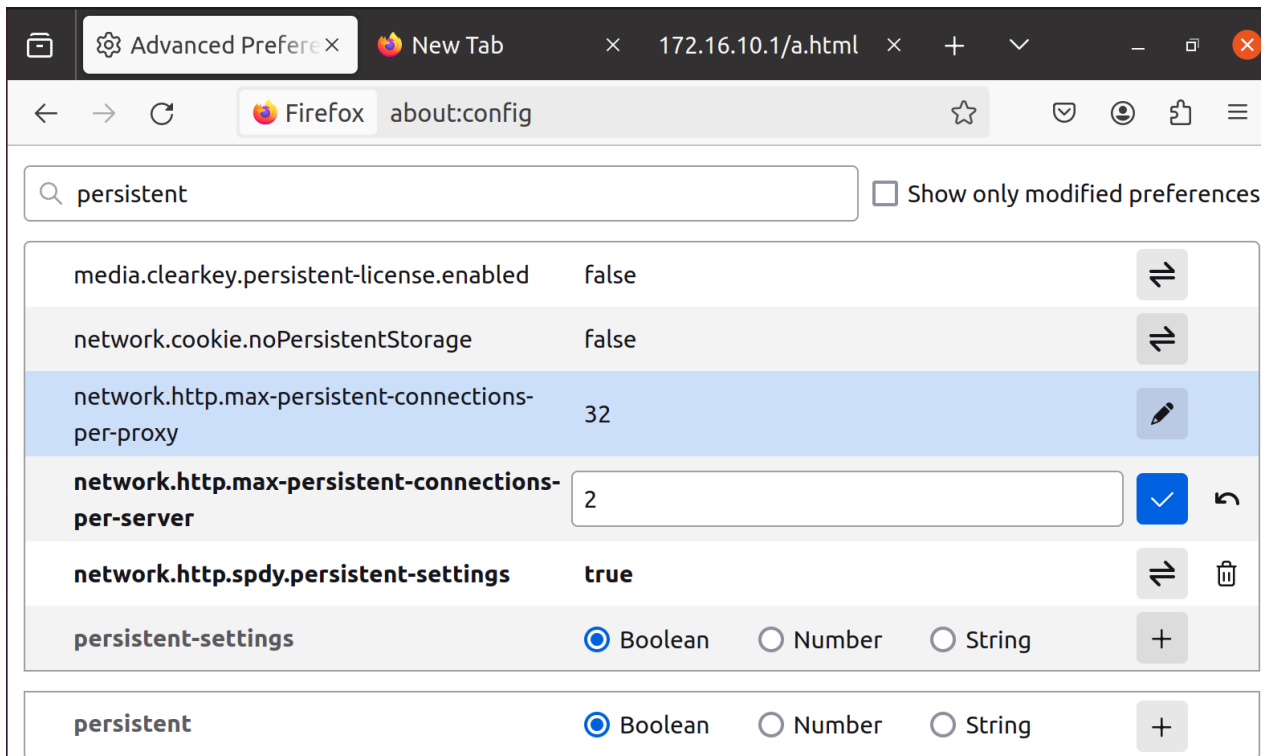


No.	Time	Source	Destination	Protocol	Length	Info
22	4.993172899	172.16.10.2	172.16.10.1	HTTP	505	GET /a.html HTTP/1.1
23	4.993977366	172.16.10.1	172.16.10.2	HTTP	575	HTTP/1.1 200 OK (text/html)
25	5.113207393	172.16.10.2	172.16.10.1	HTTP	451	GET /pesu.jpeg HTTP/1.1
26	5.114356477	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
28	5.117372222	172.16.10.2	172.16.10.1	HTTP	453	GET /plane2.jpeg HTTP/1.1
29	5.126128003	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
31	5.127553341	172.16.10.2	172.16.10.1	HTTP	453	GET /plane3.jpeg HTTP/1.1
32	5.128466348	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
34	5.129947908	172.16.10.2	172.16.10.1	HTTP	453	GET /super1.jpeg HTTP/1.1
35	5.130624948	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
36	5.132134647	172.16.10.2	172.16.10.1	HTTP	453	GET /super2.jpeg HTTP/1.1
37	5.132548131	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
38	5.134278477	172.16.10.2	172.16.10.1	HTTP	453	GET /super3.jpeg HTTP/1.1
39	5.135075219	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
40	5.136962718	172.16.10.2	172.16.10.1	HTTP	453	GET /super4.jpeg HTTP/1.1
41	5.137906229	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
1043	131.009345806	192.168.1.25	91.189.91.98	HTTP	153	GET / HTTP/1.1
1048	132.102093580	91.189.91.98	192.168.1.25	HTTP	251	HTTP/1.1 204 No Content
1173	149.395571078	192.168.1.23	23.35.7.133	HTTP	281	GET / HTTP/1.1
1175	149.401203227	23.35.7.133	192.168.1.23	HTTP	317	HTTP/1.1 304 Not Modified
1184	149.508984728	192.168.1.23	192.229.232.240	HTTP	348	GET /msdownload/update/v3/static/trusted/en/disallowedcertstl.cab76d04daic
1188	149.520390451	192.229.232.240	192.168.1.23	HTTP	343	HTTP/1.1 304 Not Modified

Here it is $149.50089 - 4.9939 = 144.50699$

PART 2: PERSISTENT CONNECTIONS

Step 1: For 2 persistent connections, set the value of **max-persistent-connection-per-server** to **2** in the client computer.

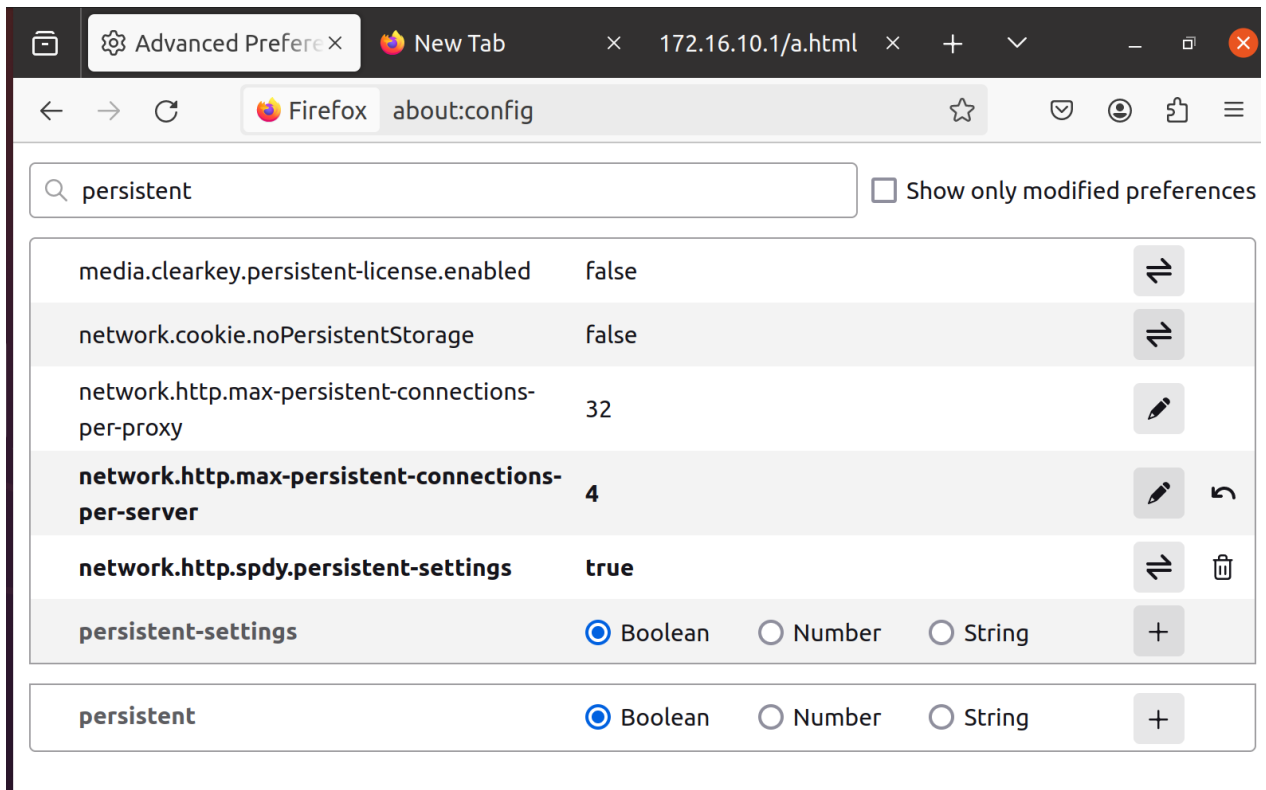


Step 2: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
49	5.457158564	172.16.10.2	172.16.10.1	HTTP	598	GET /a.html HTTP/1.1
51	5.459911460	172.16.10.1	172.16.10.2	HTTP	575	HTTP/1.1 200 OK (text/html)
53	5.535229969	172.16.10.2	172.16.10.1	HTTP	447	GET /3.jpg HTTP/1.1
54	5.536178895	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
56	5.540722767	172.16.10.2	172.16.10.1	HTTP	453	GET /pesu.jpeg HTTP/1.1
57	5.549489872	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
62	5.550898089	172.16.10.2	172.16.10.1	HTTP	453	GET /plane2.jpeg HTTP/1.1
63	5.550898170	172.16.10.2	172.16.10.1	HTTP	453	GET /plane3.jpeg HTTP/1.1
65	5.552182113	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
66	5.552366246	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
69	5.553842606	172.16.10.2	172.16.10.1	HTTP	453	GET /super1.jpeg HTTP/1.1
70	5.554226560	172.16.10.2	172.16.10.1	HTTP	453	GET /super2.jpeg HTTP/1.1
71	5.555748083	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
73	5.557572985	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
74	5.559834339	172.16.10.2	172.16.10.1	HTTP	453	GET /super3.jpeg HTTP/1.1
76	5.560458895	172.16.10.2	172.16.10.1	HTTP	453	GET /super4.jpeg HTTP/1.1
77	5.560635745	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
79	5.562677854	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
1006	157.285159562	192.168.1.23	199.232.210.33	HTTP	409	GET /Builds/UnrealEngineLauncher/CloudDir/C_c8j6iul1HSQjkQUimk
1017	157.346712672	199.232.210.33	192.168.1.23	HTTP	21226	HTTP/1.1 200 OK (text/plain)
1055	160.678896187	192.168.1.23	199.232.210.33	HTTP	377	GET /Builds/UnrealEngineLauncher/CloudDir/ChunksV4/97/63CC31E67
1056	160.671249853	192.168.1.23	199.232.210.33	HTTP	377	GET /Builds/UnrealEngineLauncher/CloudDir/ChunksV4/68/8EFE3EBC7
1873	160.974884201	199.232.210.33	192.168.1.23	HTTP	318	HTTP/1.1 200 OK
1876	160.975528931	199.232.210.33	192.168.1.23	HTTP	22144	HTTP/1.1 200 OK
1943	167.971643932	192.168.1.25	185.125.190.18	HTTP	153	GET / HTTP/1.1
1974	171.728701567	185.125.190.18	192.168.1.25	HTTP	255	HTTP/1.1 204 No Content

Here it is $167.97164 - 5.459911 = 162.511729$

Step 3: For 4 persistent connections, Set the value of **max-persistent-connection-per-server** to **4** in the client computer.

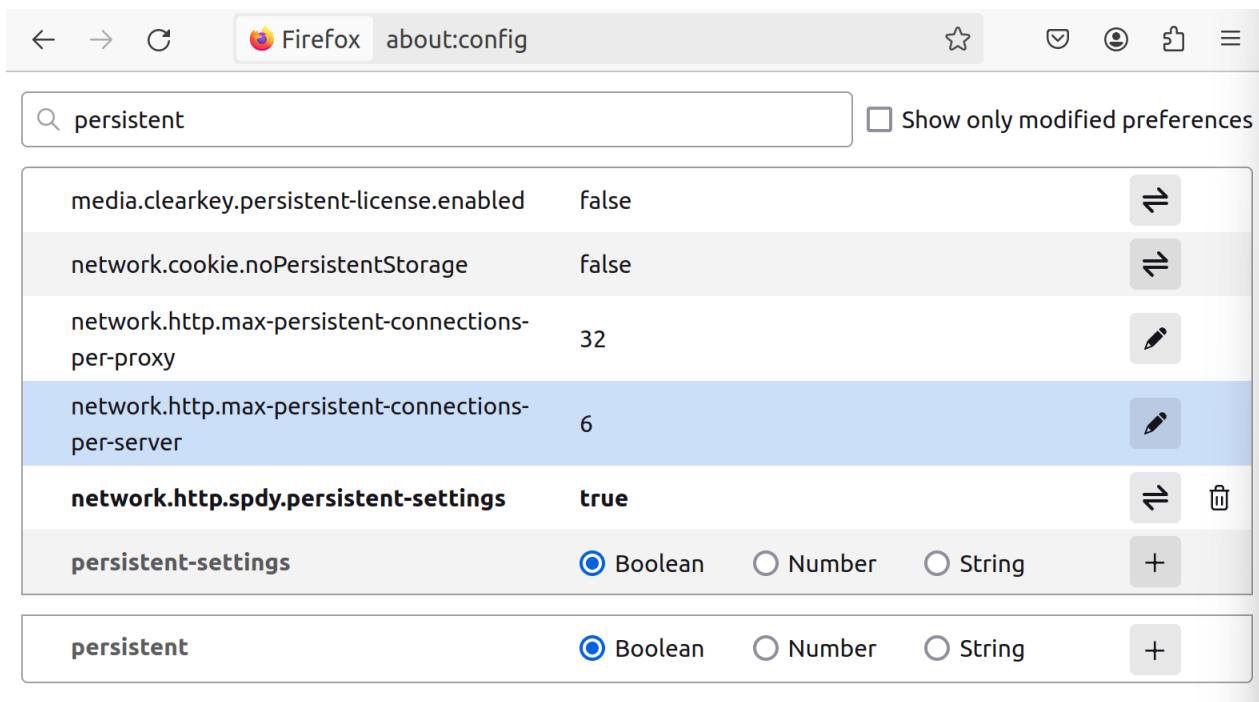


Step 4: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
38	11.681237779	172.16.10.2	172.16.10.1	HTTP	506	GET /a.html HTTP/1.1
40	11.683348544	172.16.10.1	172.16.10.2	HTTP	575	HTTP/1.1 200 OK (text/html)
42	11.800179308	172.16.10.2	172.16.10.1	HTTP	448	GET /1.jpg HTTP/1.1
43	11.800913403	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
48	11.804226772	172.16.10.2	172.16.10.1	HTTP	447	GET /2.jpg HTTP/1.1
49	11.805568640	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
50	11.806088791	172.16.10.2	172.16.10.1	HTTP	451	GET /pesu.jpeg HTTP/1.1
57	11.806929967	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
61	11.808660506	172.16.10.2	172.16.10.1	HTTP	453	GET /plane2.jpeg HTTP/1.1
62	11.809274344	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
63	11.809549926	172.16.10.2	172.16.10.1	HTTP	453	GET /plane3.jpeg HTTP/1.1
64	11.809550229	172.16.10.2	172.16.10.1	HTTP	453	GET /super1.jpeg HTTP/1.1
66	11.811328422	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
67	11.811420945	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
71	11.814786711	172.16.10.2	172.16.10.1	HTTP	453	GET /super2.jpeg HTTP/1.1
73	11.817144468	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
75	11.818851940	172.16.10.2	172.16.10.1	HTTP	453	GET /super3.jpeg HTTP/1.1
76	11.818852159	172.16.10.2	172.16.10.1	HTTP	453	GET /super4.jpeg HTTP/1.1
77	11.819795706	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
78	11.820059118	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
961	191.134677631	192.168.1.25	185.125.190.48	HTTP	153	GET / HTTP/1.1
981	195.520768636	185.125.190.48	192.168.1.25	HTTP	255	HTTP/1.1 204 No Content

Here it is $195.520768 - 11.681237 = 183.839531$

Step 5: For 6 persistent connections, set the value of **max-persistent-connection-per-server** to **6** in the server computer.

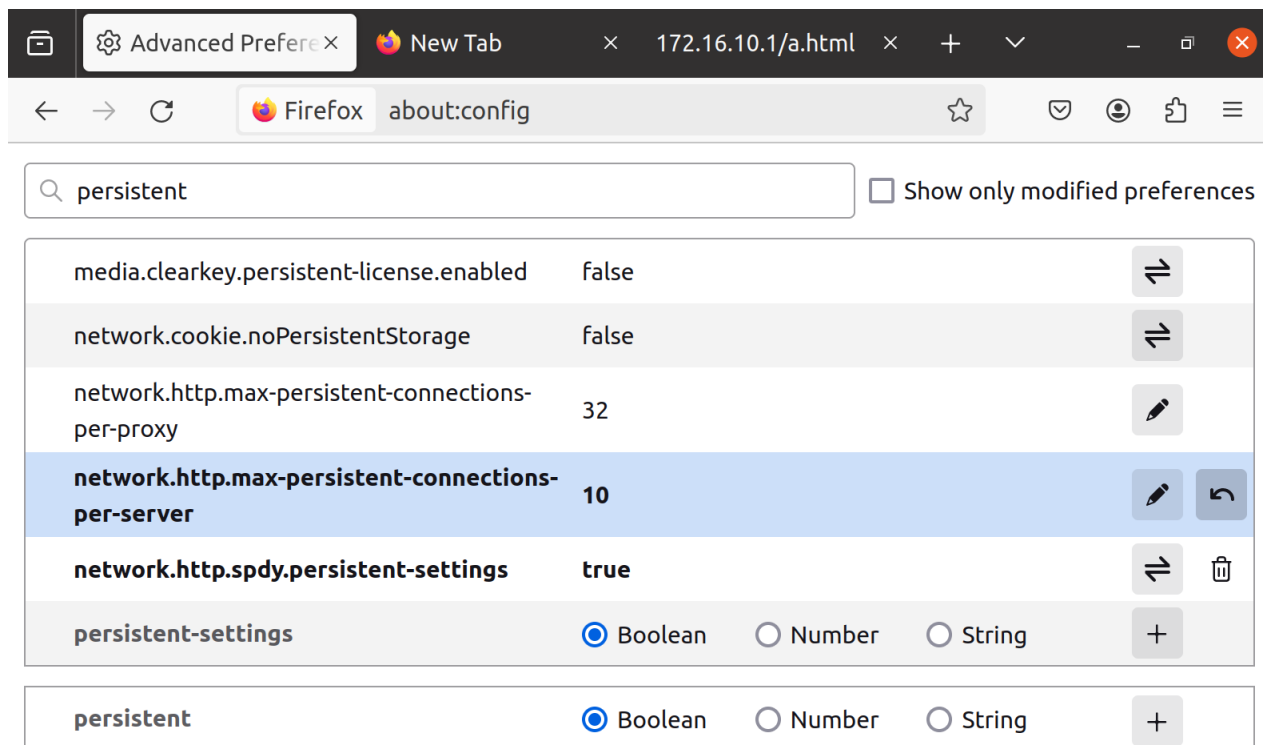


Step 6: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
44	14.149032410	172.16.10.2	172.16.10.1	HTTP	508	GET /a.html HTTP/1.1
45	14.149032248	172.16.10.1	172.16.10.2	HTTP	575	HTTP/1.1 200 OK (text/html)
46	14.216632651	172.16.10.2	172.16.10.1	HTTP	447	GET /3.jpg HTTP/1.1
49	14.217443732	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
51	14.230119377	172.16.10.2	172.16.10.1	HTTP	451	GET /pesu.jpeg HTTP/1.1
54	14.231890855	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
57	14.233288938	172.16.10.2	172.16.10.1	HTTP	453	GET /plane2.jpeg HTTP/1.1
59	14.233803965	172.16.10.2	172.16.10.1	HTTP	453	GET /plane3.jpeg HTTP/1.1
63	14.235081544	172.16.10.2	172.16.10.1	HTTP	453	GET /super1.jpeg HTTP/1.1
71	14.237769842	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
72	14.238822756	172.16.10.1	172.16.10.2	HTTP	316	HTTP/1.1 304 Not Modified
73	14.238898479	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
77	14.240451839	172.16.10.2	172.16.10.1	HTTP	453	GET /super2.jpeg HTTP/1.1
78	14.240786179	172.16.10.2	172.16.10.1	HTTP	453	GET /super3.jpeg HTTP/1.1
79	14.241255606	172.16.10.2	172.16.10.1	HTTP	453	GET /super4.jpeg HTTP/1.1
82	14.245925052	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
84	14.247126469	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
86	14.248355280	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
764	85.947124994	192.168.1.23	152.195.38.76	HTTP	306	GET /sha2-ha-server-g6.crl HTTP/1.1
766	85.958046104	152.195.38.76	192.168.1.23	HTTP	339	HTTP/1.1 304 Not Modified

Here it is $85.958046 - 14.149032 = 71.899014$

Step 7: For 10 persistent connections, set the value of **max-persistent-connection-perserver** to **10** in the client computer.



Step 8: Repeat the **steps 1-3** in the previous section.

http						
No.	Time	Source	Destination	Protocol	Length	Info
80	10.263841707	172.16.10.2	172.16.10.1	HTTP	506	GET /a.html HTTP/1.1
82	10.264989548	172.16.10.1	172.16.10.2	HTTP	576	HTTP/1.1 200 OK (text/html)
84	10.324895371	172.16.10.2	172.16.10.1	HTTP	453	GET /super4.jpeg HTTP/1.1
85	10.325914808	172.16.10.1	172.16.10.2	HTTP	315	HTTP/1.1 304 Not Modified
550	81.266015271	192.168.1.25	91.189.91.48	HTTP	153	GET / HTTP/1.1
554	82.235613937	91.189.91.48	192.168.1.25	HTTP	255	HTTP/1.1 204 No Content

Here it is $81.23561 - 10.263841 = 70.971769$

OBSERVATIONS REQUIRED ON EDMODO:

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.

SCREENSHOTS REQUIRED FOR EDMODO:

```

osboxes@osboxes:~$ sudo wireshark
[sudo] password for osboxes:
** (wireshark:13564) 11:42:20.062147 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to
** (wireshark:13564) 11:42:23.915502 [Capture MESSAGE] -- Capture Start ...
** (wireshark:13564) 11:42:24.109341 [Capture MESSAGE] -- Capture started
** (wireshark:13564) 11:42:24.109454 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s31JM7I2.pcapng"
** (wireshark:13564) 11:52:22.019406 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:13564) 11:52:22.119121 [Capture MESSAGE] -- Capture stopped.
** (wireshark:13564) 11:52:22.119153 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:13564) 11:52:25.948544 [Capture MESSAGE] -- Capture Start ...
** (wireshark:13564) 11:52:26.321436 [Capture MESSAGE] -- Capture started
** (wireshark:13564) 11:52:26.321519 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3HNG8I2.pcapng"
** (wireshark:13564) 11:54:15.179221 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:13564) 11:54:15.267479 [Capture MESSAGE] -- Capture stopped.
** (wireshark:13564) 11:54:15.267590 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:13564) 11:54:21.215790 [Capture MESSAGE] -- Capture Start ...
** (wireshark:13564) 11:54:21.560658 [Capture MESSAGE] -- Capture started
** (wireshark:13564) 11:54:21.560727 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s30M41I2.pcapng"
** (wireshark:13564) 11:57:24.410112 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:13564) 11:57:24.479136 [Capture MESSAGE] -- Capture stopped.
** (wireshark:13564) 11:57:24.479340 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:13564) 11:57:28.263997 [Capture MESSAGE] -- Capture Start ...
** (wireshark:13564) 11:57:28.544281 [Capture MESSAGE] -- Capture started
** (wireshark:13564) 11:57:28.544315 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s34Y0I2.pcapng"
** (wireshark:13564) 11:58:30.386241 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:13564) 11:58:30.461408 [Capture MESSAGE] -- Capture stopped.
** (wireshark:13564) 11:58:30.461495 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
** (wireshark:13564) 11:58:33.664059 [Capture MESSAGE] -- Capture Start ...
** (wireshark:13564) 11:58:33.938908 [Capture MESSAGE] -- Capture started
** (wireshark:13564) 11:58:33.940052 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s306KUI2.pcapng"
** (wireshark:13564) 11:59:11.675160 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:13564) 11:59:11.708067 [Capture MESSAGE] -- Capture stopped.

```

- 1) Non-persistent connection wireshark capture (should include all 10 images)
- 2) Persistent connections wireshark capture – 2, 4, 6, 8 & 10 respectively (should include all 10 images).