

Lab # 2

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

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

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: Create groups of 2 such that one student's desktop will be the server, the other will be client.

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

Ensure that the VM is in bridged adapter network

Server Client



Note : For demonstration purposes, both the desktops can implement the steps for the server side but only the client can ping the server's IP address for the actual experiment.

Server Side:

Step 3: 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
```

If the server is not running , use the following command:

```
Sudo systemctl start apache2
```

```
osboxes@osboxes: ~  
osboxes@osboxes:~$ sudo systemctl status apache2  
[sudo] password for osboxes:  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enab  
   Active: active (running) since Sun 2025-01-26 00:16:08 EST; 34s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Process: 907 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC  
 Main PID: 980 (apache2)  
    Tasks: 55 (limit: 9445)  
   Memory: 7.6M  
      CPU: 69ms  
   CGroup: /system.slice/apache2.service  
           └─980 /usr/sbin/apache2 -k start  
             └─984 /usr/sbin/apache2 -k start  
               └─985 /usr/sbin/apache2 -k start  
  
Jan 26 00:16:08 osboxes systemd[1]: Starting apache2.service - The Apache HTTP >  
Jan 26 00:16:08 osboxes apachectl[973]: AH00557: apache2: apr_sockaddr_info_get>  
Jan 26 00:16:08 osboxes apachectl[973]: AH00558: apache2: Could not reliably de>  
Jan 26 00:16:08 osboxes systemd[1]: Started apache2.service - The Apache HTTP S>  
lines 1-18/18 (END)
```

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.

Find out the others systems(your partner's) ip by using the command ifconfig (named enp0s3)

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**

\$sudo nano /etc/apache2/apache2.conf

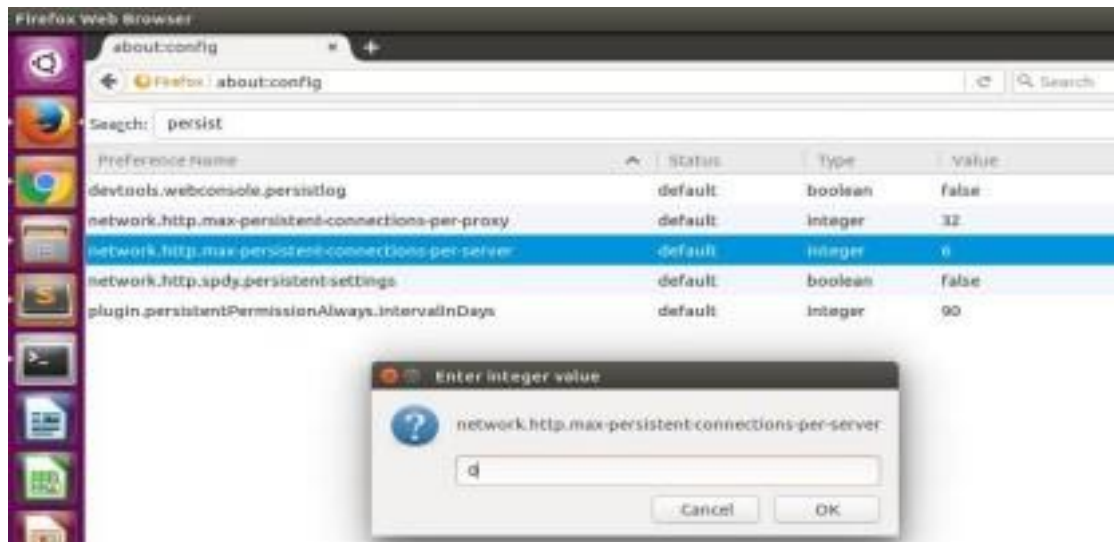
```
[osboxes@osboxes ~]$ sudo nano /etc/apache2/apache2.conf  
#  
# Timeout: The number of seconds before receives and sends time out.  
#  
Timeout 300  
#  
# KeepAlive: Whether or not to allow persistent connections (more than  
# one request per connection). Set to "Off" to deactivate.  
#  
KeepAlive 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

persistent connection experiment, the **max-persistent-connections-per server** 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: On the server's desktop,

The **apache2.conf** file present in the **etc/apache2** directory is modified as: a) The **keep-alive** option was set to False (i.e. value was made **OFF**)

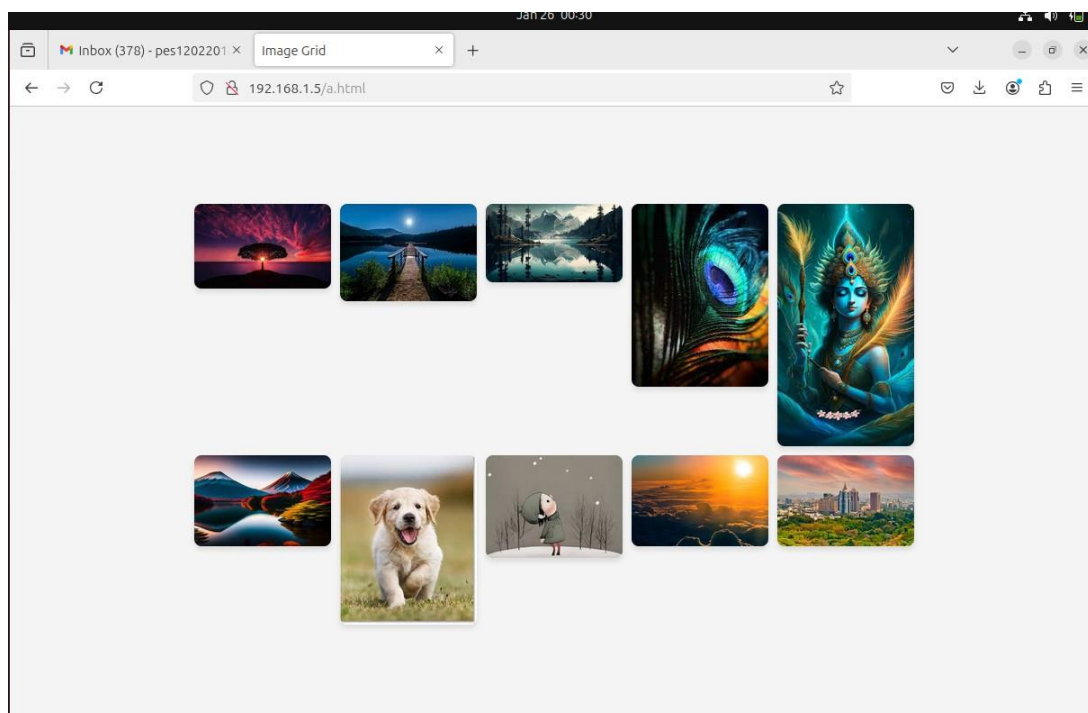
\$sudo nano /etc/apache2/apache2.conf

Step 3: Access web page on client-side browser (Firefox)

The client could access the file as:

serverip /file_name.html where--> **server ip** was found using **ifconfig**

Here the file name is **a.html** present in server. So, by typing **serverip/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.

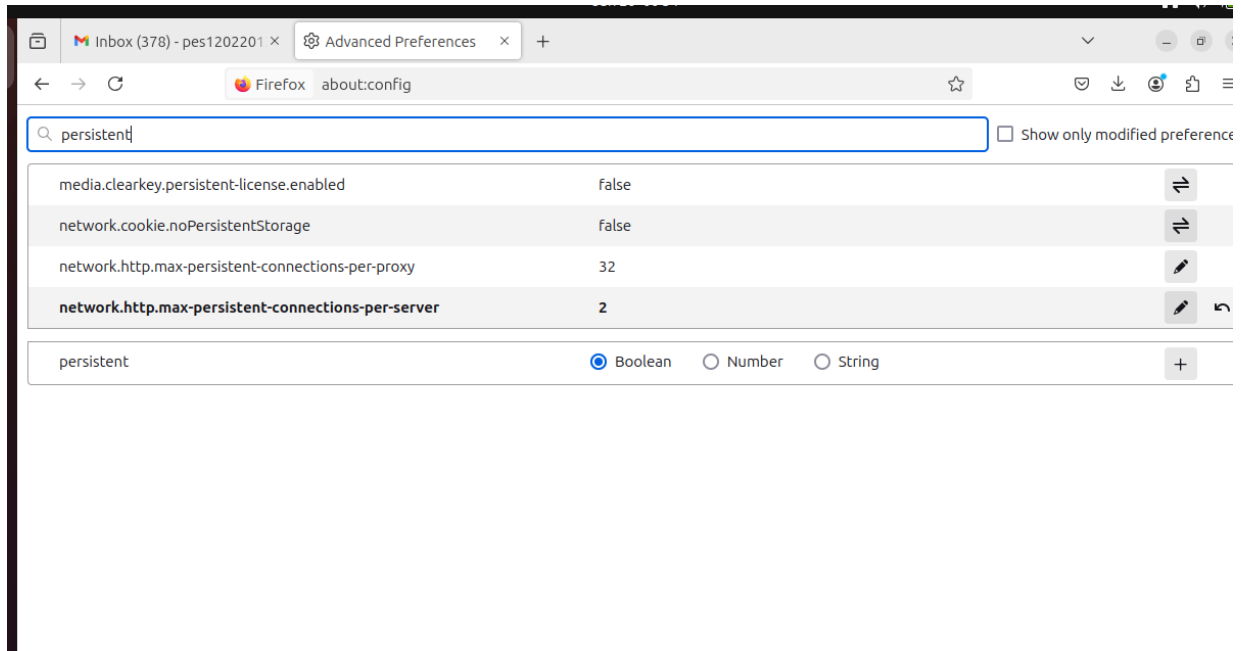
Step 4: Use wireshark. Open wireshark in the server computer while client is trying to access the server's local host webpage. Apply 'http' filter and note the time to capture all the 10 images.

No.	Time	Source	Destination	Protocol	Length	Info
25	0.211538185	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not found (text/html)
27	2.070581279	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
28	2.078866155	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
30	2.117168769	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
35	2.117571913	172.16.10.1	172.16.10.2	HTTP	1200	HTTP/1.1 200 OK (JPEG 3FIF image)
36	2.117753115	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
45	2.117944288	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG 3FIF image)
51	2.118574057	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
63	2.119058490	172.16.10.1	172.16.10.2	HTTP	242	HTTP/1.1 200 OK (JPEG 3FIF image)
65	2.119487932	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
77	2.119784374	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG 3FIF image)
79	2.120323770	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
94	2.121263792	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
110	2.122045160	172.16.10.1	172.16.10.2	HTTP	1226	HTTP/1.1 200 OK (JPEG 3FIF image)
117	2.122719543	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
138	2.123847115	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
160	2.124708199	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958050_960_720.jpg HTTP/1.1
164	2.124733805	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG 3FIF image)
171	2.125125151	172.16.10.1	172.16.10.2	HTTP	711	HTTP/1.1 200 OK (JPEG 3FIF image)
184	2.126599573	172.16.10.2	172.16.10.1	HTTP	340	GET /download%20(2).jpg HTTP/1.1
252	2.131056667	172.16.10.1	172.16.10.2	HTTP	114	HTTP/1.1 200 OK (JPEG 3FIF image)
529	2.151487483	172.16.10.1	172.16.10.2	HTTP	73	HTTP/1.1 200 OK (JPEG 3FIF image)
3834	2.429637133	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG 3FIF image)

Here it is $2.429637133 - 2.070581279 = 0.359055854$

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: On the server's desktop,

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**) (since we're now checking for persistent connections)

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

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

No.	Time	Source	Destination	Protocol	Length	Info
28	0.158495852	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
30	2.685888334	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
31	2.686488793	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
33	2.734801898	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
38	2.734592637	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
39	2.734656958	172.16.10.1	172.16.10.2	HTTP	1280	HTTP/1.1 200 OK (JPEG JFIF image)
48	2.735025557	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
49	2.735188565	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
66	2.736079156	172.16.10.1	172.16.10.2	HTTP	243	HTTP/1.1 200 OK (JPEG JFIF image)
68	2.738374645	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
82	2.736755733	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
85	2.737381832	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
92	2.737840608	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
101	2.738335488	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
119	2.738809142	172.16.10.1	172.16.10.2	HTTP	1226	HTTP/1.1 200 OK (JPEG JFIF image)
121	2.739875438	172.16.10.1	172.16.10.2	HTTP	1016	HTTP/1.1 200 OK (JPEG JFIF image)
139	2.740908758	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
143	2.741814891	172.16.10.2	172.16.10.1	HTTP	362	GET /scoop-bubble-1958658_968_728.jpg HTTP/1.1
148	2.741285777	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
179	2.7428807475	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
198	2.743723338	172.16.10.1	172.16.10.2	HTTP	712	HTTP/1.1 200 OK (JPEG JFIF image)
402	2.764894977	172.16.10.1	172.16.10.2	HTTP	72	HTTP/1.1 200 OK (JPEG JFIF image)
574	3.042252027	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $3.042252027 - 2.685888334 = 0.356363$

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.

No.	Time	Source	Destination	Protocol	Length	Info
28	0.152642908	172.16.10.1	172.16.10.2	HTTP	568	HTTP/1.1 404 Not Found (text/html)
30	1.667969551	172.16.10.2	172.16.10.1	HTTP	421	GET /a.html HTTP/1.1
31	1.668311781	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
33	1.699473631	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(1).jpg HTTP/1.1
35	1.699692089	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
45	1.699900042	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
46	1.699913083	172.16.10.1	172.16.10.2	HTTP	1280	HTTP/1.1 200 OK (JPEG JFIF image)
47	1.700012712	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(4).jpg HTTP/1.1
63	1.700901747	172.16.10.1	172.16.10.2	HTTP	242	HTTP/1.1 200 OK (JPEG JFIF image)
69	1.701341018	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(3).jpg HTTP/1.1
70	1.701432635	172.16.10.2	172.16.10.1	HTTP	359	GET /lion-wild-africa-african.jpg HTTP/1.1
86	1.701888988	172.16.10.1	172.16.10.2	HTTP	565	HTTP/1.1 200 OK (JPEG JFIF image)
93	1.702192885	172.16.10.2	172.16.10.1	HTTP	341	GET /images.jpg HTTP/1.1
95	1.702219175	172.16.10.2	172.16.10.1	HTTP	343	GET /download.jpg HTTP/1.1
97	1.702228220	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(1).jpg HTTP/1.1
98	1.702233138	172.16.10.2	172.16.10.1	HTTP	362	GET /scoop-bubble-1958658_968_728.jpg HTTP/1.1
122	1.703328138	172.16.10.1	172.16.10.2	HTTP	711	HTTP/1.1 200 OK (JPEG JFIF image)
126	1.703773424	172.16.10.2	172.16.10.1	HTTP	349	GET /download%20(2).jpg HTTP/1.1
157	1.705498971	172.16.10.1	172.16.10.2	HTTP	1227	HTTP/1.1 200 OK (JPEG JFIF image)
159	1.705614494	172.16.10.1	172.16.10.2	HTTP	113	HTTP/1.1 200 OK (JPEG JFIF image)
167	1.706637782	172.16.10.1	172.16.10.2	HTTP	1017	HTTP/1.1 200 OK (JPEG JFIF image)
414	1.724541388	172.16.10.1	172.16.10.2	HTTP	72	HTTP/1.1 200 OK (JPEG JFIF image)
1025	2.005934395	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

Here it is $2.005934395 - 1.667969557 = 0.337964838$

OBSERVATIONS REQUIRED:

Calculate the time taken to load objects from the server for non-persistent and persistent connections (2, 4,). Find out the optimal number of HTTP persistent connections based on your observations.

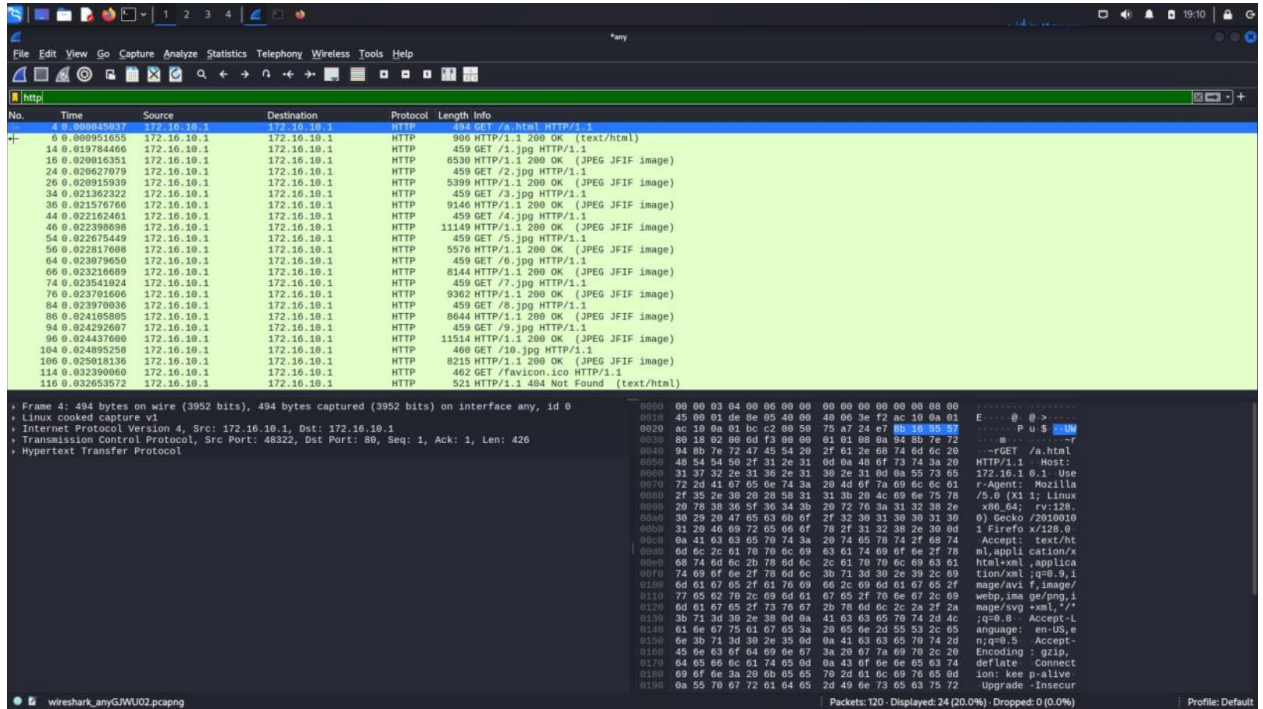
SCREENSHOTS REQUIRED FOR SUBMISSION:

1) Non-persistent connection wireshark capture (should include all 10 images) 2)

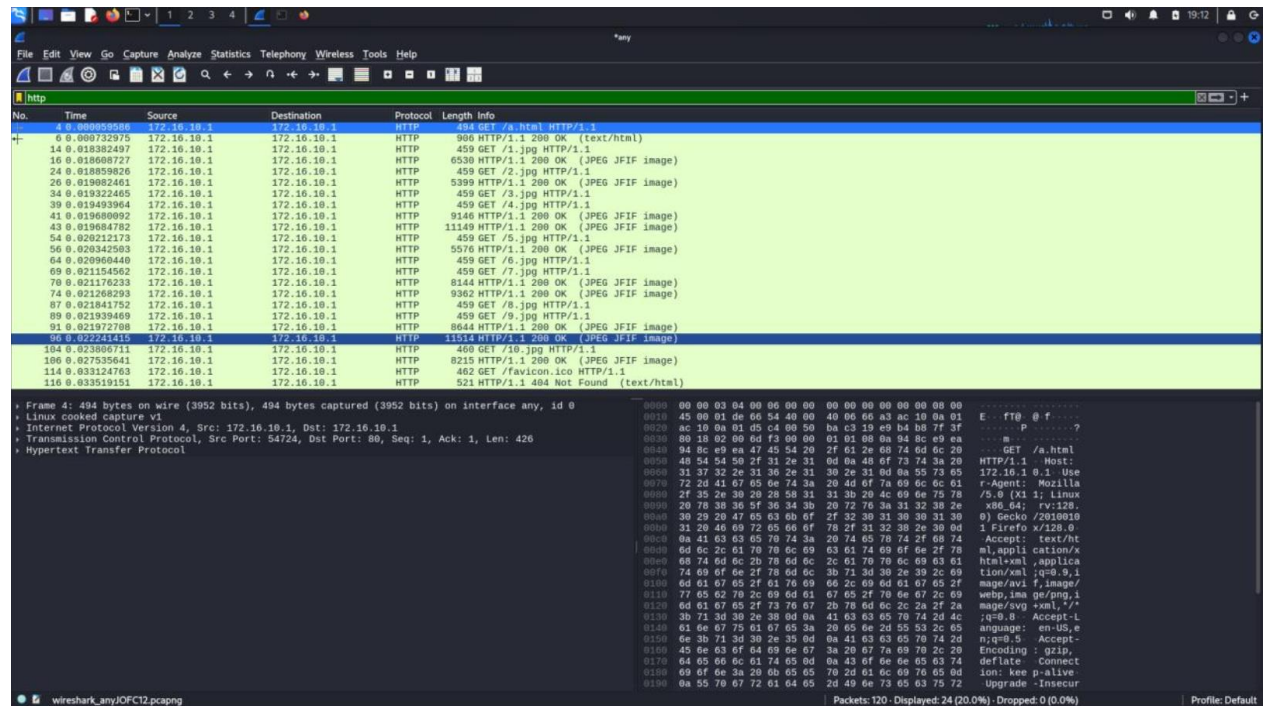
Persistent connections wireshark capture – 2, 4, 6, 8 & 10 respectively (should include all 10 images).

Non-Persistent:

TIME: 0.0326 – 0.00004 = 0.03256



Persistent 2:



TIME: 0.0335 - 0.00005 = 0.03345

Persistent 4:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.00055936	172.16.10.1	172.16.10.1	HTTP	484	GET /a.html HTTP/1.1
6	0.000829587	172.16.10.1	172.16.10.1	HTTP	906	HTTP/1.1 200 OK (text/html)
14	0.020573479	172.16.10.1	172.16.10.1	HTTP	459	GET /1.jpg HTTP/1.1
19	0.020823444	172.16.10.1	172.16.10.1	HTTP	459	GET /2.jpg HTTP/1.1
21	0.020961918	172.16.10.1	172.16.10.1	HTTP	5399	HTTP/1.1 200 OK (JPEG JFIF image)
22	0.020962059	172.16.10.1	172.16.10.1	HTTP	5399	HTTP/1.1 200 OK (JPEG JFIF image)
34	0.021323114	172.16.10.1	172.16.10.1	HTTP	459	GET /3.jpg HTTP/1.1
36	0.021476109	172.16.10.1	172.16.10.1	HTTP	9146	HTTP/1.1 200 OK (JPEG JFIF image)
44	0.021652768	172.16.10.1	172.16.10.1	HTTP	459	GET /4.jpg HTTP/1.1
46	0.021753598	172.16.10.1	172.16.10.1	HTTP	11149	HTTP/1.1 200 OK (JPEG JFIF image)
54	0.021932996	172.16.10.1	172.16.10.1	HTTP	459	GET /5.jpg HTTP/1.1
59	0.022097810	172.16.10.1	172.16.10.1	HTTP	459	GET /6.jpg HTTP/1.1
61	0.022197996	172.16.10.1	172.16.10.1	HTTP	8144	HTTP/1.1 200 OK (JPEG JFIF image)
64	0.022316982	172.16.10.1	172.16.10.1	HTTP	5576	HTTP/1.1 200 OK (JPEG JFIF image)
80	0.022895084	172.16.10.1	172.16.10.1	HTTP	459	GET /8.jpg HTTP/1.1
82	0.022965224	172.16.10.1	172.16.10.1	HTTP	459	GET /7.jpg HTTP/1.1
84	0.023049996	172.16.10.1	172.16.10.1	HTTP	8644	HTTP/1.1 200 OK (JPEG JFIF image)
87	0.023187159	172.16.10.1	172.16.10.1	HTTP	459	GET /9.jpg HTTP/1.1
94	0.023297130	172.16.10.1	172.16.10.1	HTTP	460	GET /10.jpg HTTP/1.1
95	0.023303510	172.16.10.1	172.16.10.1	HTTP	9363	HTTP/1.1 200 OK (JPEG JFIF image)
99	0.023373739	172.16.10.1	172.16.10.1	HTTP	8215	HTTP/1.1 200 OK (JPEG JFIF image)
102	0.023391878	172.16.10.1	172.16.10.1	HTTP	11514	HTTP/1.1 200 OK (JPEG JFIF image)
114	0.031998062	172.16.10.1	172.16.10.1	HTTP	462	GET /favicon.ico HTTP/1.1
116	0.032111861	172.16.10.1	172.16.10.1	HTTP	521	HTTP/1.1 404 Not Found (text/html)

Frame 4: 484 bytes on wire (3952 bits), 494 bytes captured (3952 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.1
Transmission Control Protocol, Src Port: 45330, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
Hypertext Transfer Protocol

Packets: 120 - Displayed: 24 (20.0%) - Dropped: 0 (0.0%)

TIME: 0.0321 - 0.00005 = 0.03205

Persistent 6:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000457300	172.16.10.1	172.16.10.1	HTTP	484	GET /a.html HTTP/1.1
6	0.010546596	172.16.10.1	172.16.10.1	HTTP	906	HTTP/1.1 200 OK (text/html)
14	0.010546596	172.16.10.1	172.16.10.1	HTTP	459	GET /1.jpg HTTP/1.1
19	0.010546596	172.16.10.1	172.16.10.1	HTTP	459	GET /2.jpg HTTP/1.1
21	0.017806935	172.16.10.1	172.16.10.1	HTTP	5399	HTTP/1.1 200 OK (JPEG JFIF image)
24	0.017806935	172.16.10.1	172.16.10.1	HTTP	5399	HTTP/1.1 200 OK (JPEG JFIF image)
40	0.017803917	172.16.10.1	172.16.10.1	HTTP	459	GET /3.jpg HTTP/1.1
42	0.018044167	172.16.10.1	172.16.10.1	HTTP	459	GET /5.jpg HTTP/1.1
44	0.018066727	172.16.10.1	172.16.10.1	HTTP	459	GET /4.jpg HTTP/1.1
49	0.018143287	172.16.10.1	172.16.10.1	HTTP	459	GET /6.jpg HTTP/1.1
54	0.018227809	172.16.10.1	172.16.10.1	HTTP	459	GET /7.jpg HTTP/1.1
56	0.018280796	172.16.10.1	172.16.10.1	HTTP	9146	HTTP/1.1 200 OK (JPEG JFIF image)
59	0.018320234	172.16.10.1	172.16.10.1	HTTP	5576	HTTP/1.1 200 OK (JPEG JFIF image)
62	0.018359462	172.16.10.1	172.16.10.1	HTTP	11149	HTTP/1.1 200 OK (JPEG JFIF image)
65	0.018621805	172.16.10.1	172.16.10.1	HTTP	8144	HTTP/1.1 200 OK (JPEG JFIF image)
70	0.018982217	172.16.10.1	172.16.10.1	HTTP	9362	HTTP/1.1 200 OK (JPEG JFIF image)
87	0.019925805	172.16.10.1	172.16.10.1	HTTP	459	GET /8.jpg HTTP/1.1
89	0.020025606	172.16.10.1	172.16.10.1	HTTP	8644	HTTP/1.1 200 OK (JPEG JFIF image)
95	0.020241573	172.16.10.1	172.16.10.1	HTTP	459	GET /9.jpg HTTP/1.1
97	0.020440905	172.16.10.1	172.16.10.1	HTTP	11514	HTTP/1.1 200 OK (JPEG JFIF image)
102	0.020570411	172.16.10.1	172.16.10.1	HTTP	460	GET /10.jpg HTTP/1.1
104	0.020638826	172.16.10.1	172.16.10.1	HTTP	8215	HTTP/1.1 200 OK (JPEG JFIF image)
114	0.020910993	172.16.10.1	172.16.10.1	HTTP	462	GET /favicon.ico HTTP/1.1
116	0.027013705	172.16.10.1	172.16.10.1	HTTP	521	HTTP/1.1 404 Not Found (text/html)

Frame 4: 494 bytes on wire (3952 bits), 494 bytes captured (3952 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.1
Transmission Control Protocol, Src Port: 54064, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
Hypertext Transfer Protocol

Packets: 120 - Displayed: 24 (20.0%) - Dropped: 0 (0.0%)

TIME: 0.0270 - 0.00004 = 0.02696

Persistent 8:

Frame 4: 494 bytes on wire (3952 bits), 494 bytes captured (3952 bits) on interface any, id 0

- Ethernet II, Src: Linux cooked capture v1, Dst: 172.16.10.1
- Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.1
- Transmission Control Protocol, Src Port: 48266, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
- Hypertext Transfer Protocol

GET /a.html HTTP/1.1

Host: 172.16.10.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

TIME: 0.0414 – 0.00007 = 0.04133

Persistent 10:

Frame 4: 494 bytes on wire (3952 bits), 494 bytes captured (3952 bits) on interface any, id 0

- Ethernet II, Src: Linux cooked capture v1, Dst: 172.16.10.1
- Internet Protocol Version 4, Src: 172.16.10.1, Dst: 172.16.10.1
- Transmission Control Protocol, Src Port: 58748, Dst Port: 80, Seq: 1, Ack: 1, Len: 426
- Hypertext Transfer Protocol

GET /a.html HTTP/1.1

Host: 172.16.10.1

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

TIME: 0.0410 – 0.00078 = 0.04022