



Department of Computer Science & Engineering
COMPUTER NETWORKS

Lab #3

Understanding Persistent and Non-persistent HTTP Connections

**UE22CS251B
4th Semester, Academic Year 2023-2024**

Date:14-01-2024

Name: V V Mohith	SRN:PES2UG22CS64 1	Section:-K
------------------	-----------------------	------------

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,

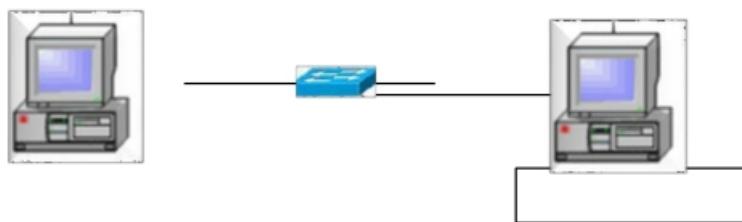
(i) **`sudo apt-get install apache2`**

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

(ii) **`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 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

```
vvMohith@Ubuntu-virtual-mohith:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-02-14 20:55:31 IST; 56s ago
     Docs: https://httpd.apache.org/docs/2.4/
Main PID: 25795 (apache2)
   Tasks: 55 (limit: 4551)
  Memory: 5.1M
    CPU: 26ms
   CGroup: /system.slice/apache2.service
           ├─25795 /usr/sbin/apache2 -k start
           ├─25796 /usr/sbin/apache2 -k start
           └─25797 /usr/sbin/apache2 -k start

Feb 14 20:55:31 Ubuntu-virtual-mohith systemd[1]: Starting The Apache HTTP Server...
Feb 14 20:55:31 Ubuntu-virtual-mohith apachectl[25794]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1.
Feb 14 20:55:31 Ubuntu-virtual-mohith systemd[1]: Started The Apache HTTP Server.
...skipping...
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-02-14 20:55:31 IST; 56s ago
     Docs: https://httpd.apache.org/docs/2.4/
Main PID: 25795 (apache2)
   Tasks: 55 (limit: 4551)
  Memory: 5.1M
    CPU: 26ms
   CGroup: /system.slice/apache2.service
           ├─25795 /usr/sbin/apache2 -k start
           ├─25796 /usr/sbin/apache2 -k start
           └─25797 /usr/sbin/apache2 -k start

Feb 14 20:55:31 Ubuntu-virtual-mohith systemd[1]: Starting The Apache HTTP Server...
Feb 14 20:55:31 Ubuntu-virtual-mohith apachectl[25794]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1.
Feb 14 20:55:31 Ubuntu-virtual-mohith systemd[1]: Started The Apache HTTP Server.
```

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'

```

vvMohith@Ubuntu-virtual-mohith:~$ 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: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 9a:43:23:9a:c4:84 brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.4/24 brd 192.168.64.255 scope global dynamic noprefixroute enp0s1
        valid_lft 84158sec preferred_lft 84158sec
    inet6 fdb9:cf99:1eea:b9ea:1d36:5ca4:540c:fe0d/64 scope global temporary dynamic
        valid_lft 602561sec preferred_lft 83771sec
    inet6 fdb9:cf99:1eea:b9ea:fc81:5e79:b415:6634/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 2591933sec preferred_lft 604733sec
    inet6 fe80::122c:2ec2:cac5:3d68/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

STEP-4(index.html):-Contents of index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Hello</title>
  </head>
  <body>
    <h1>WELCOME TO WEB HOSTING BY USING APACHE</h1>
    
    
    
    
    
    
    
    
    
    
  </body>
</html>

```

Client side:

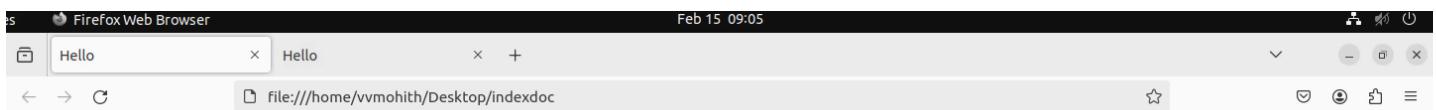
Client IP address can be set by the following command.

```
$sudo ip addr add 172.16.10.2/24 dev enps0 $sudo  
ip addr
```

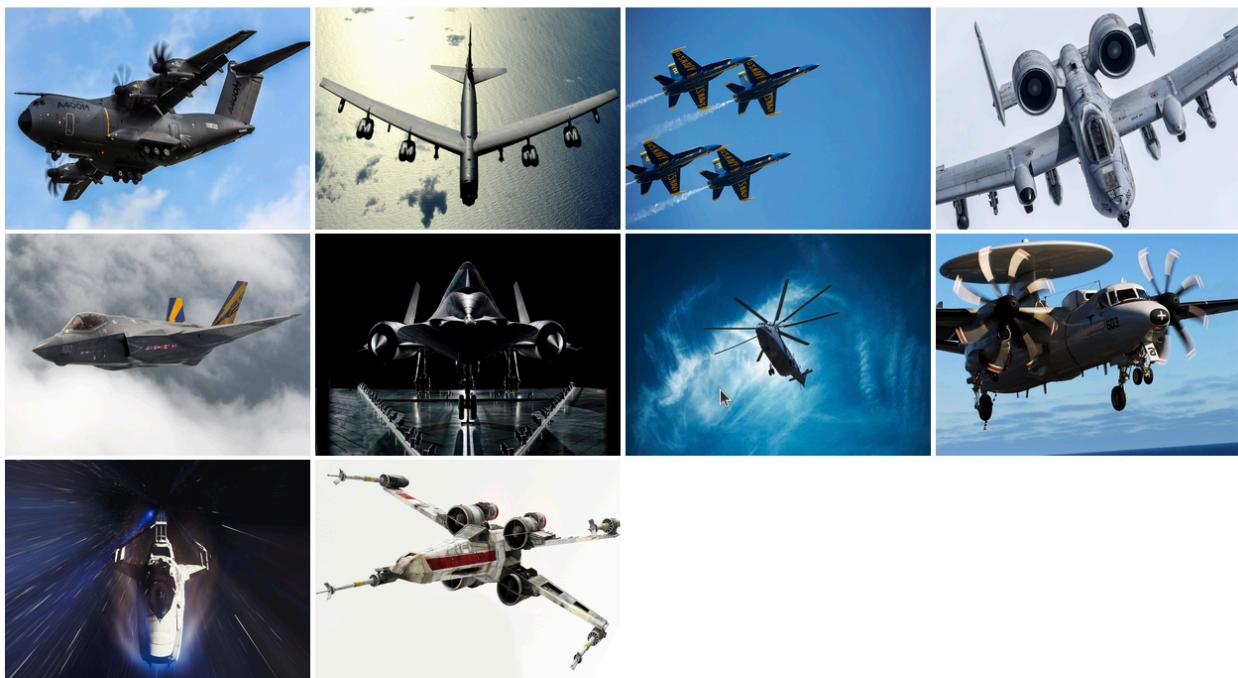
Note: If IP address fluctuates, kindly setup the IP address manually using 'Edit connections'.

```
vvmothit@Ubuntu-virtual-mothit:~$ sudo ip addr add 172.16.10.2/24 dev enps1  
vvmothit@Ubuntu-virtual-mothit:~$ 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: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 9a:43:23:9a:c4:84 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.64.4/24 brd 192.168.64.255 scope global dynamic noprefixroute enp0s1  
        valid_lft 83240sec preferred_lft 83240sec  
    inet 172.16.10.2/24 scope global enp0s1  
        valid_lft forever preferred_lft forever  
    inet6 fdb9:cf99:1eea:b9ea:1d36:5ca4:540c:fe0d/64 scope global temporary dynamic  
        valid_lft 601642sec preferred_lft 82852sec  
    inet6 fdb9:cf99:1eea:b9ea:fc81:5e79:b415:6634/64 scope global dynamic mngtmpaddr noprefixroute  
        valid_lft 2591945sec preferred_lft 604745sec  
    inet6 fe80::122c:2ec2:cac5:3d68/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

THE OUTPUT IN BROWSER



WELCOME TO WEB HOSTING BY USING APACHE



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-connections-per-server** has the value set to **0** and **persistent-settings** value set to false.
- While using 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.

The screenshot shows the Firefox 'about:config' page with the search bar containing 'persist'. The results list includes several preferences related to persistence, with one specific preference highlighted:

Preference	Value	Action
browser.pageActions.persistedActions	{"ids": ["bookmark"], "idsInUrlbar": ["bookmark"], "idsInUrlbarPreProton": [], "version": 1}	edit
browser.sessionstore.persist_closed_tabs_between_sessions	true	edit
devtools.netmonitor.persistlog	false	edit
devtools.webconsole.persistlog	false	edit
dom.serviceWorkers.testing.persistTemporarilyInstalledAddons	false	edit
media.clearkey.persistent-license.enabled	false	edit
network.cookie.noPersistentStorage	false	edit
network.http.max-persistent-connections-per-proxy	32	edit
network.http.max-persistent-connections-per-server	0	edit

At the bottom of the search results, there is a dropdown menu labeled 'persist' and a radio button group for selecting the data type: Boolean (selected), Number, or String.

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 tying **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.

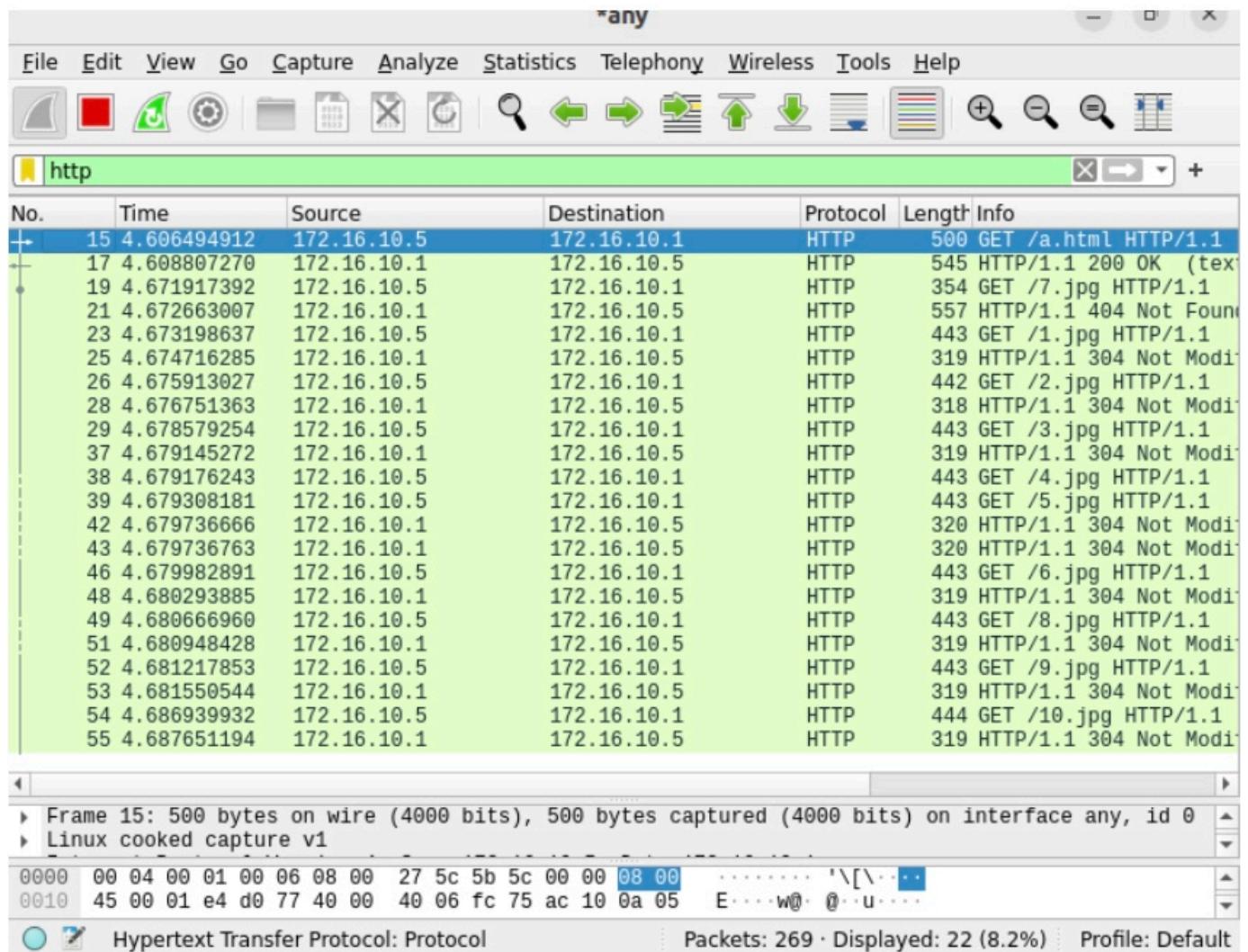
Step 3: Use **wireshark**. Open **wireshark** in the server computer while client is trying to access the server's local host webpage. Apply 'http' filter and note the time to capture all the 10 images.

No.	Time	Source	Destination	Protocol	Length	Info
25	0.211530105	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	2.070866155	172.16.10.1	172.16.10.2	HTTP	641	HTTP/1.1 200 OK (text/html)
30	2.117160769	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 JFIF 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 JFIF 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 JFIF 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 JFIF 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	2.122045168	172.16.10.1	172.16.10.2	HTTP	1226	HTTP/1.1 200 OK (JPEG JFIF 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	2.124700199	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958650_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 JFIF image)
171	2.125125151	172.16.10.1	172.16.10.2	HTTP	711	HTTP/1.1 200 OK (JPEG JFIF image)
184	2.126599573	172.16.10.2	172.16.10.1	HTTP	349	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 JFIF image)
529	2.151487483	172.16.10.1	172.16.10.2	HTTP	73	HTTP/1.1 200 OK (JPEG JFIF image)
3834	2.429637133	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF 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.



$$\text{Time} = 4.686939932 - 4.606494912 = 0.08044502$$

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

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	1.699692009	172.16.10.2	172.16.10.1	HTTP	347	GET /images%20(2).jpg HTTP/1.1
45	1.699908042	172.16.10.1	172.16.10.2	HTTP	463	HTTP/1.1 200 OK (JPEG JFIF image)
46	1.699913003	172.16.10.1	172.16.10.2	HTTP	1200	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	1.702233130	172.16.10.2	172.16.10.1	HTTP	362	GET /soap-bubble-1958658_960_720.jpg HTTP/1.1
122	1.703328136	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	1.705614894	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	73	HTTP/1.1 200 OK (JPEG JFIF image)
3825	2.005934395	172.16.10.1	172.16.10.2	HTTP	1124	HTTP/1.1 200 OK (JPEG JFIF image)

$$\text{Time} = 2.005934395 - 1.667969557 = 0.337964838$$

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

http						
No.	Time	Source	Destination	Protocol	Length	Info
36	8.515465141	192.168.1.8	192.168.1.15	HTTP	479	GET /a.html HTTP/1.1
38	8.516332425	192.168.1.15	192.168.1.8	HTTP	589	HTTP/1.1 200 OK (text/html)
47	8.532432291	192.168.1.8	192.168.1.15	HTTP	453	GET /1.jpg HTTP/1.1
50	8.532432517	192.168.1.8	192.168.1.15	HTTP	452	GET /2.jpg HTTP/1.1
53	8.532886211	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
54	8.532906953	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
55	8.534528143	192.168.1.8	192.168.1.15	HTTP	453	GET /3.jpg HTTP/1.1
57	8.534528473	192.168.1.8	192.168.1.15	HTTP	453	GET /4.jpg HTTP/1.1
60	8.534953432	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
61	8.535015174	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
63	8.538337400	192.168.1.8	192.168.1.15	HTTP	453	GET /5.jpg HTTP/1.1
65	8.538337623	192.168.1.8	192.168.1.15	HTTP	453	GET /6.jpg HTTP/1.1
68	8.538649006	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
69	8.538666261	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
70	8.540992508	192.168.1.8	192.168.1.15	HTTP	453	GET /7.jpg HTTP/1.1
72	8.540992732	192.168.1.8	192.168.1.15	HTTP	453	GET /8.jpg HTTP/1.1
74	8.541323255	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
75	8.541357501	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
76	8.554812843	192.168.1.8	192.168.1.15	HTTP	453	GET /9.jpg HTTP/1.1
78	8.554813130	192.168.1.8	192.168.1.15	HTTP	454	GET /10.jpg HTTP/1.1
80	8.555225468	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
81	8.555273740	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified

$$\text{Time} = 8.554813130 - 8.532432291 = 0.022380839$$

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

No.	Time	Source	Destination	Protocol	Length	Info
49	4.624096321	192.168.1.8	192.168.1.15	HTTP	479	GET /a.html HTTP/1.1
51	4.624915606	192.168.1.15	192.168.1.8	HTTP	589	HTTP/1.1 200 OK (text/html)
63	4.647147765	192.168.1.8	192.168.1.15	HTTP	453	GET /1.jpg HTTP/1.1
69	4.647153690	192.168.1.8	192.168.1.15	HTTP	453	GET /3.jpg HTTP/1.1
76	4.647575617	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
77	4.647602793	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
81	4.657511903	192.168.1.8	192.168.1.15	HTTP	453	GET /4.jpg HTTP/1.1
82	4.657509684	192.168.1.8	192.168.1.15	HTTP	452	GET /2.jpg HTTP/1.1
86	4.657678356	192.168.1.8	192.168.1.15	HTTP	453	GET /5.jpg HTTP/1.1
89	4.657678417	192.168.1.8	192.168.1.15	HTTP	453	GET /6.jpg HTTP/1.1
91	4.658041806	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
92	4.658058667	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
93	4.658084999	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
94	4.658103995	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
97	4.659880046	192.168.1.8	192.168.1.15	HTTP	453	GET /7.jpg HTTP/1.1
99	4.659880103	192.168.1.8	192.168.1.15	HTTP	453	GET /8.jpg HTTP/1.1
103	4.660414099	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
104	4.660432835	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified
110	4.662460604	192.168.1.8	192.168.1.15	HTTP	453	GET /9.jpg HTTP/1.1
112	4.662512580	192.168.1.8	192.168.1.15	HTTP	454	GET /10.jpg HTTP/1.1
114	4.662788274	192.168.1.15	192.168.1.8	HTTP	249	HTTP/1.1 304 Not Modified
115	4.662910900	192.168.1.15	192.168.1.8	HTTP	250	HTTP/1.1 304 Not Modified

$$\text{Time} = 4.662512580 - 4.647147765 = 0.015364815$$

The optimal number of persistent connections for best performance = 10

POINTS NOTED

- Higher the number of persistent connections, lower is the time taken. The optimal connection is 4 or 6. For best case, use 10. Clients should have **at most 2** persistent connections to any server to prevent the server from being overloaded. A non-persistent connection takes the most time.

- The non-persistent connection takes the connection time of $2RTT + \text{each file transmission time}$. A persistent connection takes the connection time of $2RTT + \text{transfers all objects at a go over the single connection}$. This is the reason why a persistent connection is faster.

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.

- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:



Name:V V Mohith

SRN:PES2UG22CS641

Section: K

Date:15-02-2024