Computer Networks Lab Report - Week 2

PES1201800366

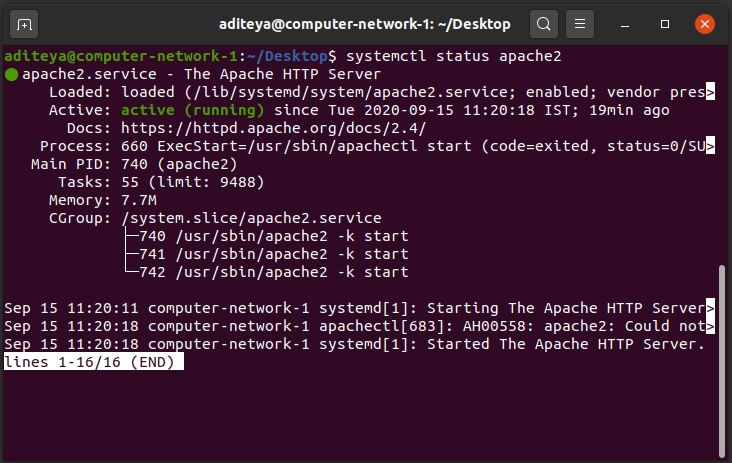
Aditeya Baral

1. Configuration of Apache Server and Client Environment

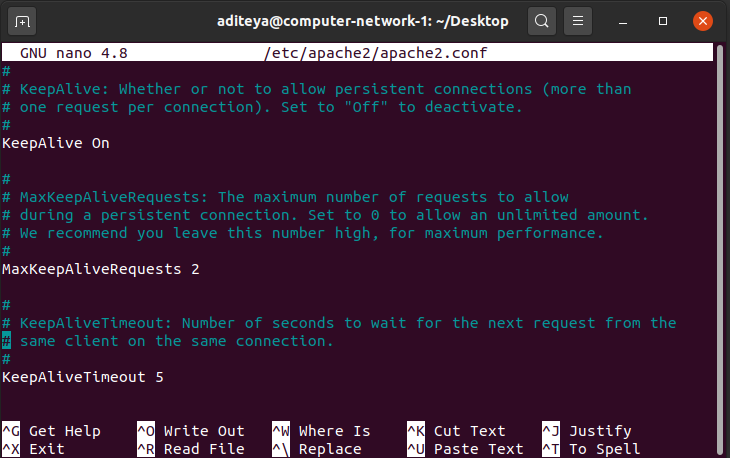
* To create a server – client architecture, two Virtual Machines were setup. The former is referred to as the server machine and the latter is the client machine.
* Apache Server was installed and configured on the server machine, and a static webpage consisting of 10 objects (images) was created and hosted on the local network between these machines.
* We need to observe and determine the effect of the number of persistent connections on the load time of this static webpage.

1.1 Setting up Apache Server

* The Apache Server can be installed with sudo apt install apache2
* The status of the newly installed server can be viewed using systemctl status apache2

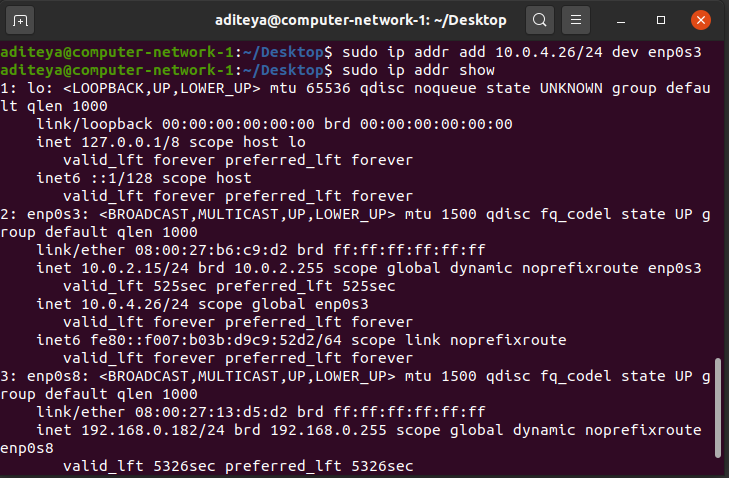


* The Apache Server also needs to be configured to allow persistent connections. This is done by editing the apache2.conf configuration file and setting the options
  + KeepAlive to On
  + MaxKeepAliveRequests to 2



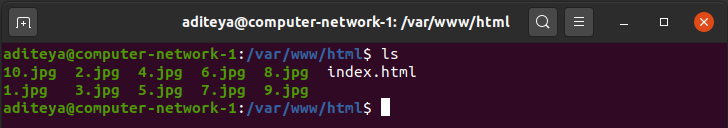
1.2 Adding Custom IP Addresses for Server and Client

* A custom IP Address was set for both the Server and Client machines
* The Server IP Address was set to 10.0.4.26 and the Client IP Address was set to 10.0.4.27
* The IP address were assigned using the sudo ip addr add command



1.3 Hosting the Webpage

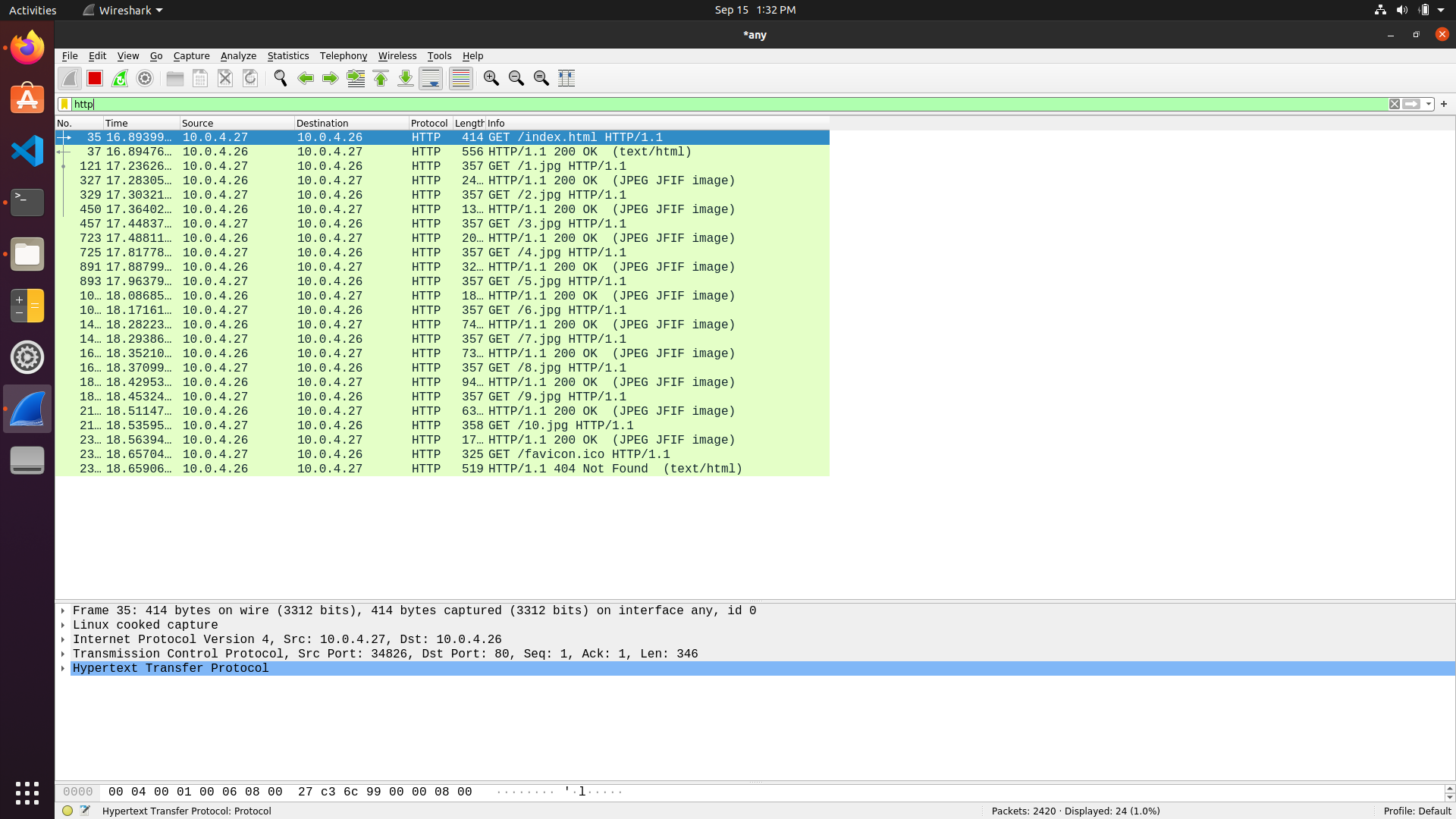
* The webpage can be hosted by moving the html script and the images to the server path
* The server path is /var/www/html/



2. Non-Persistent Connection

* To setup a non-persistent connection, we need to configure a few settings on our browser
* On Firefox, we set the max-persistent-connections-per-server to 0 and persistent-settings to false

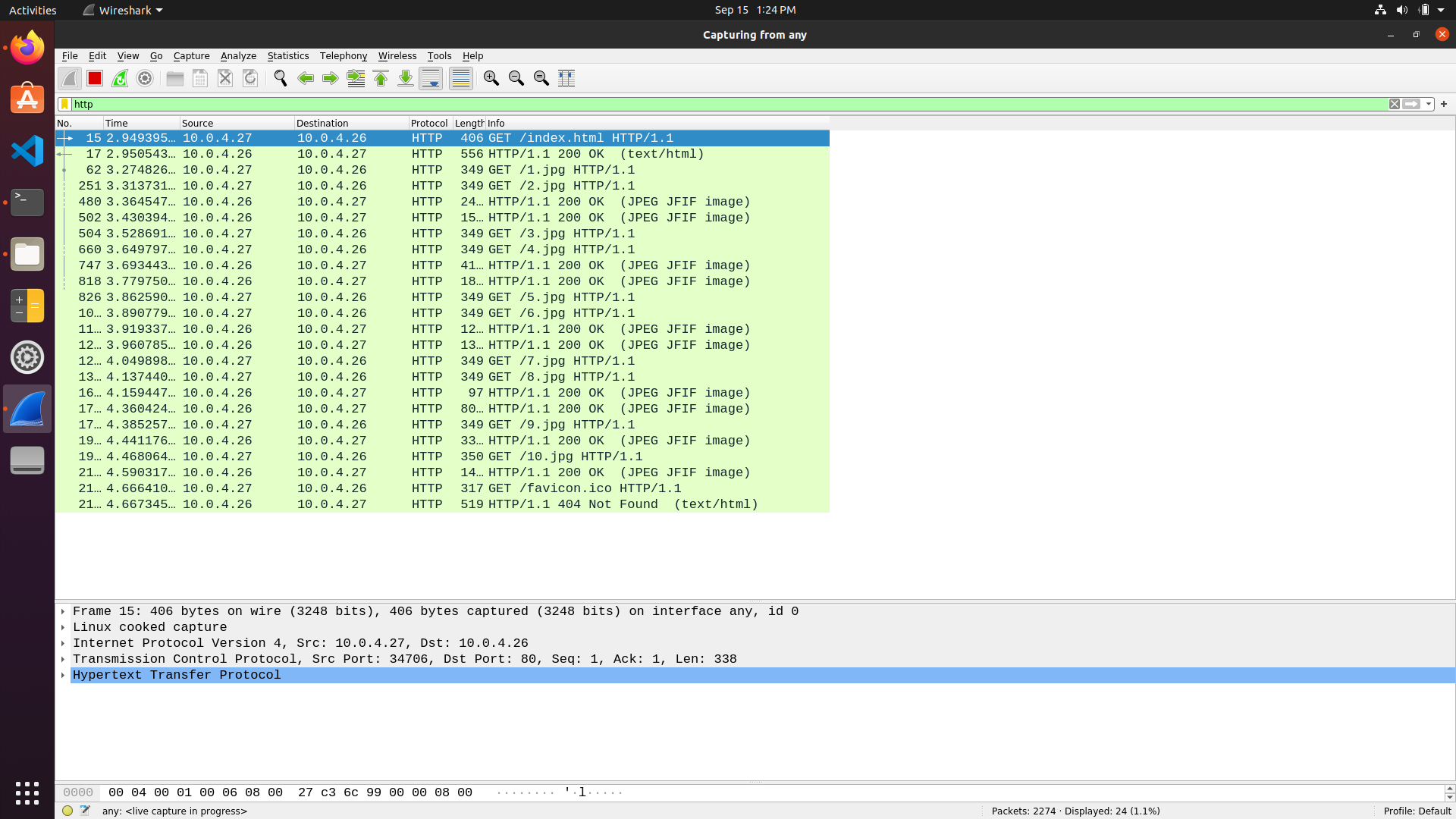
2.1 Packet Capture Screenshot



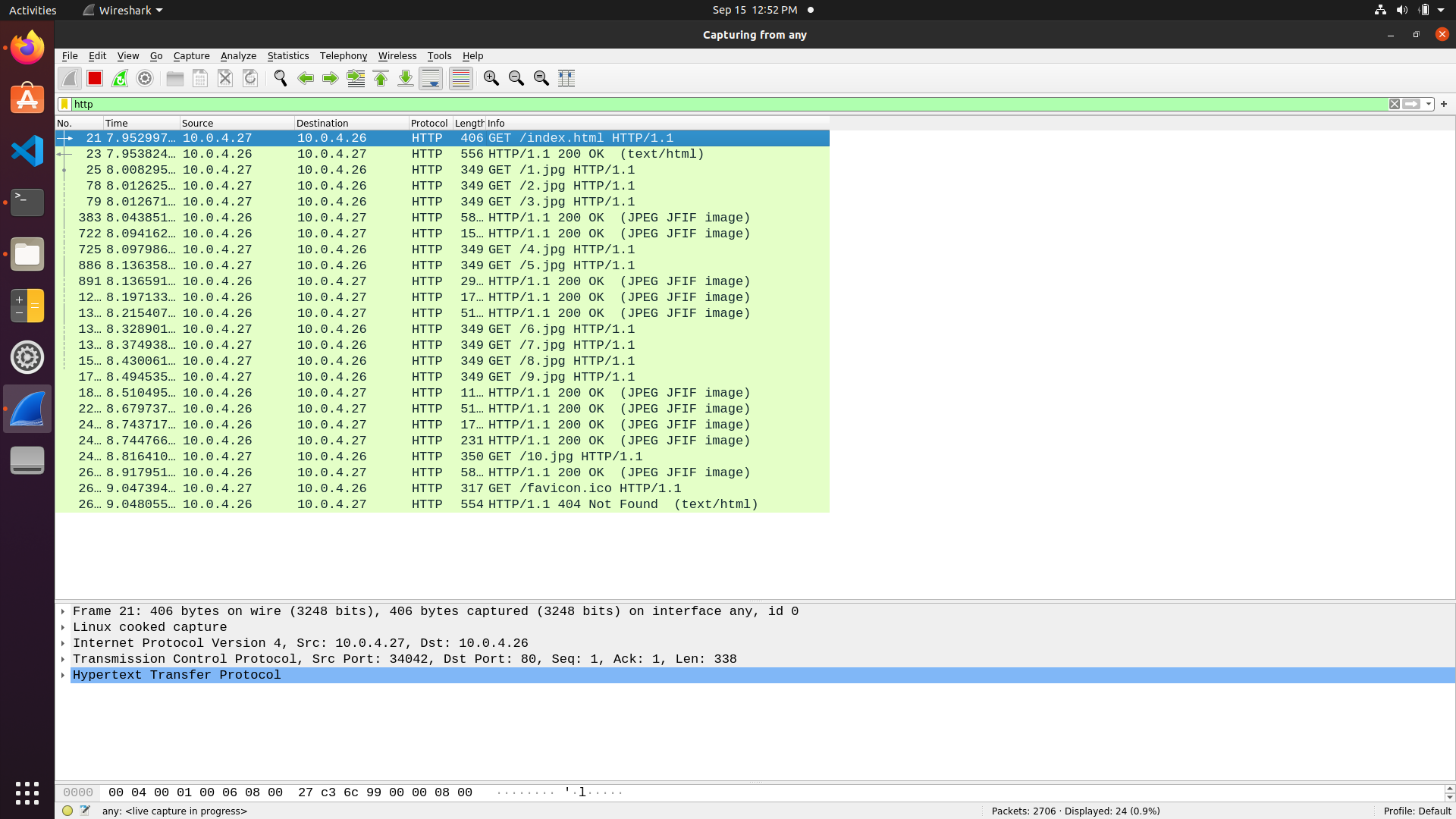
3. Persistent Connection

* To setup a persistent connection, we need to configure a few settings on our browser
* On Firefox, we set the max-persistent-connections-per-server to anything greater than 0 and persistent-settings to true

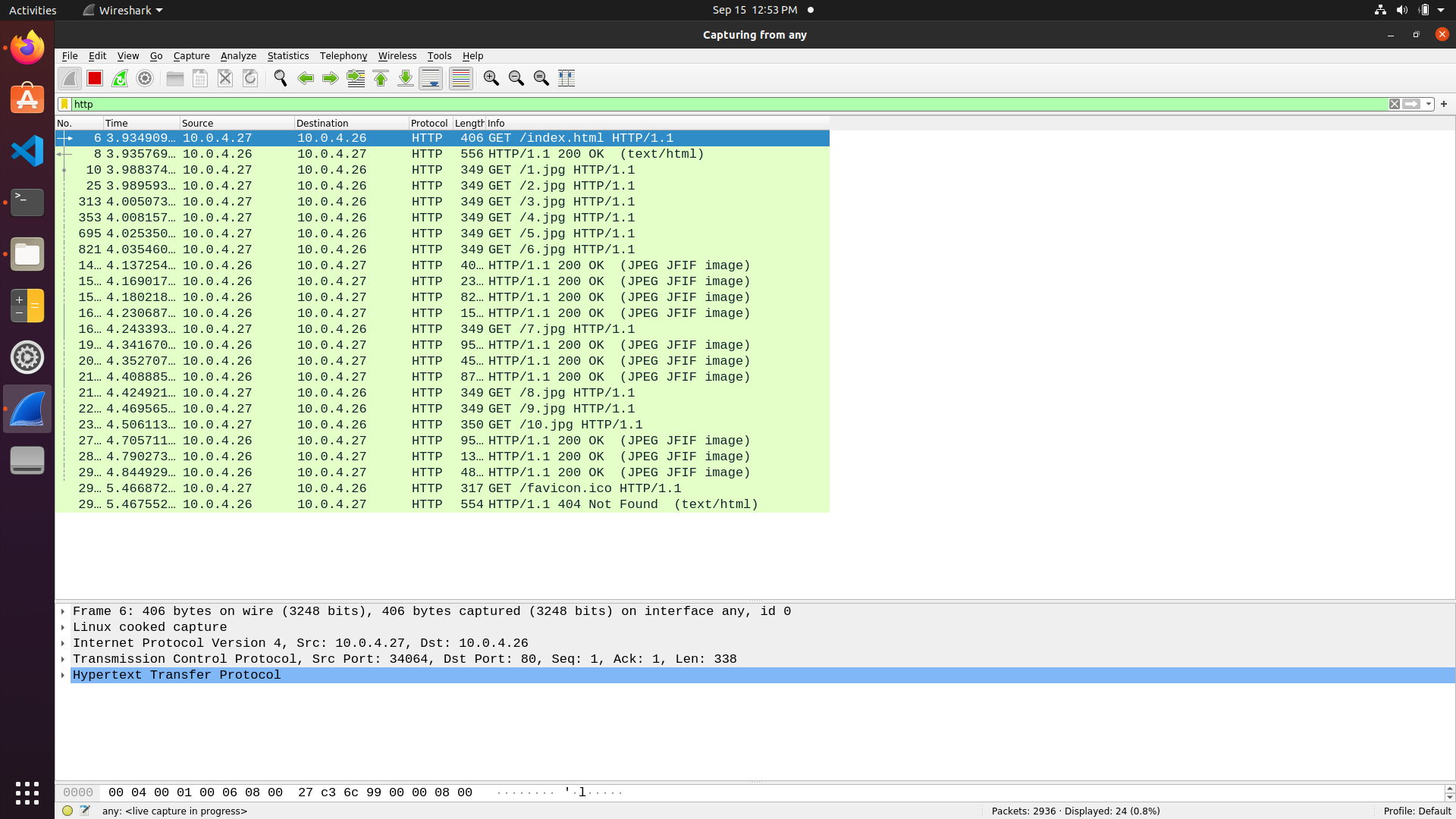
3.1. 2 Persistent Connections



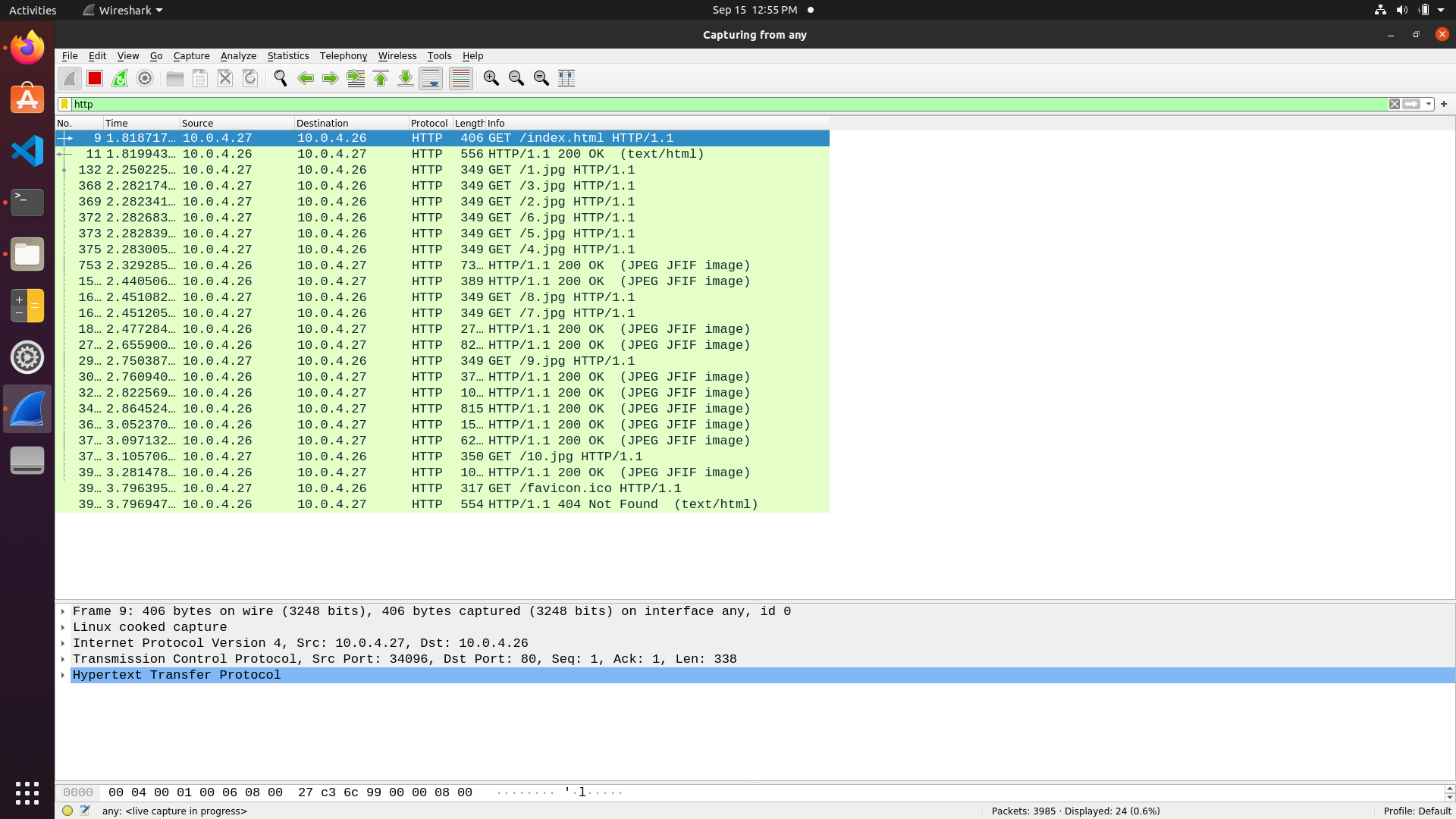
3.2. 4 Persistent Connections



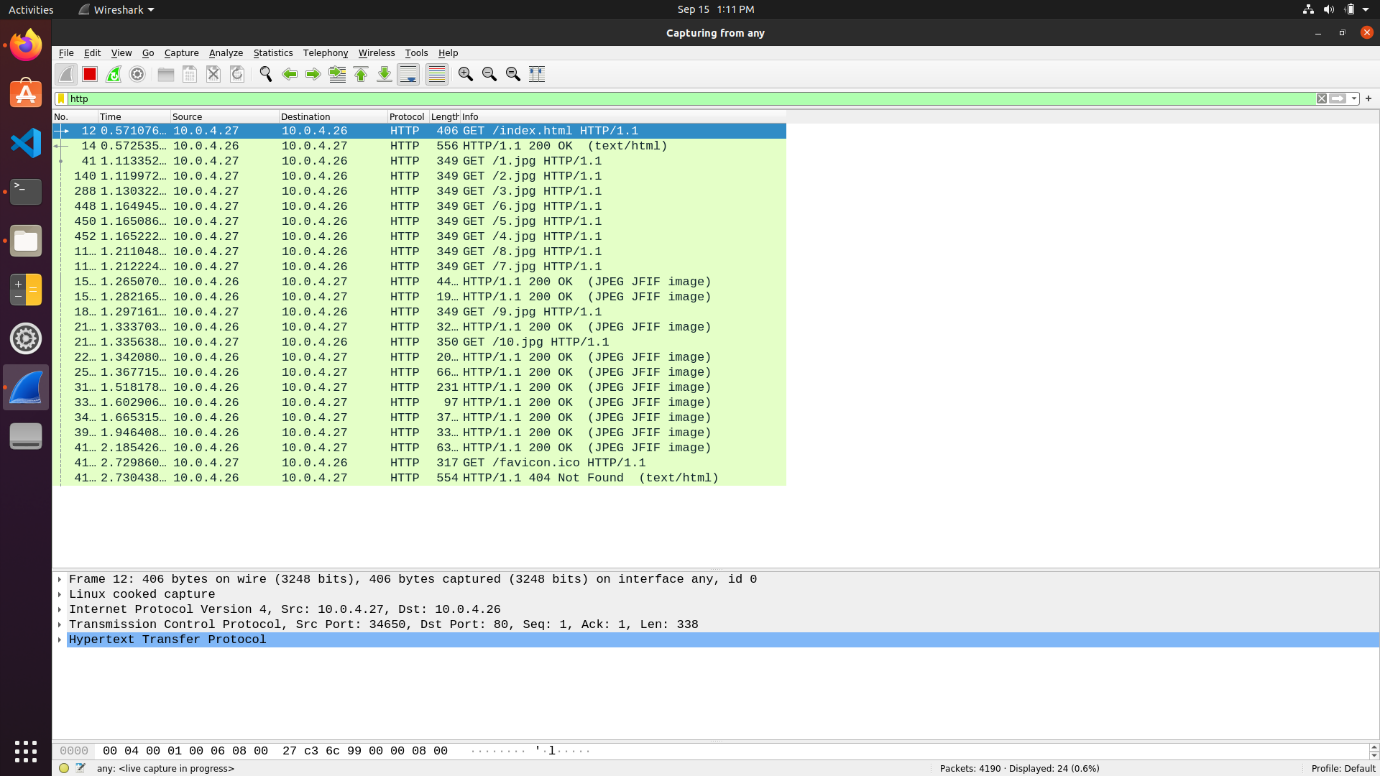
3.3. 6 Persistent Connections



3.4. 8 Persistent Connections



3.5. 10 Persistent Connections

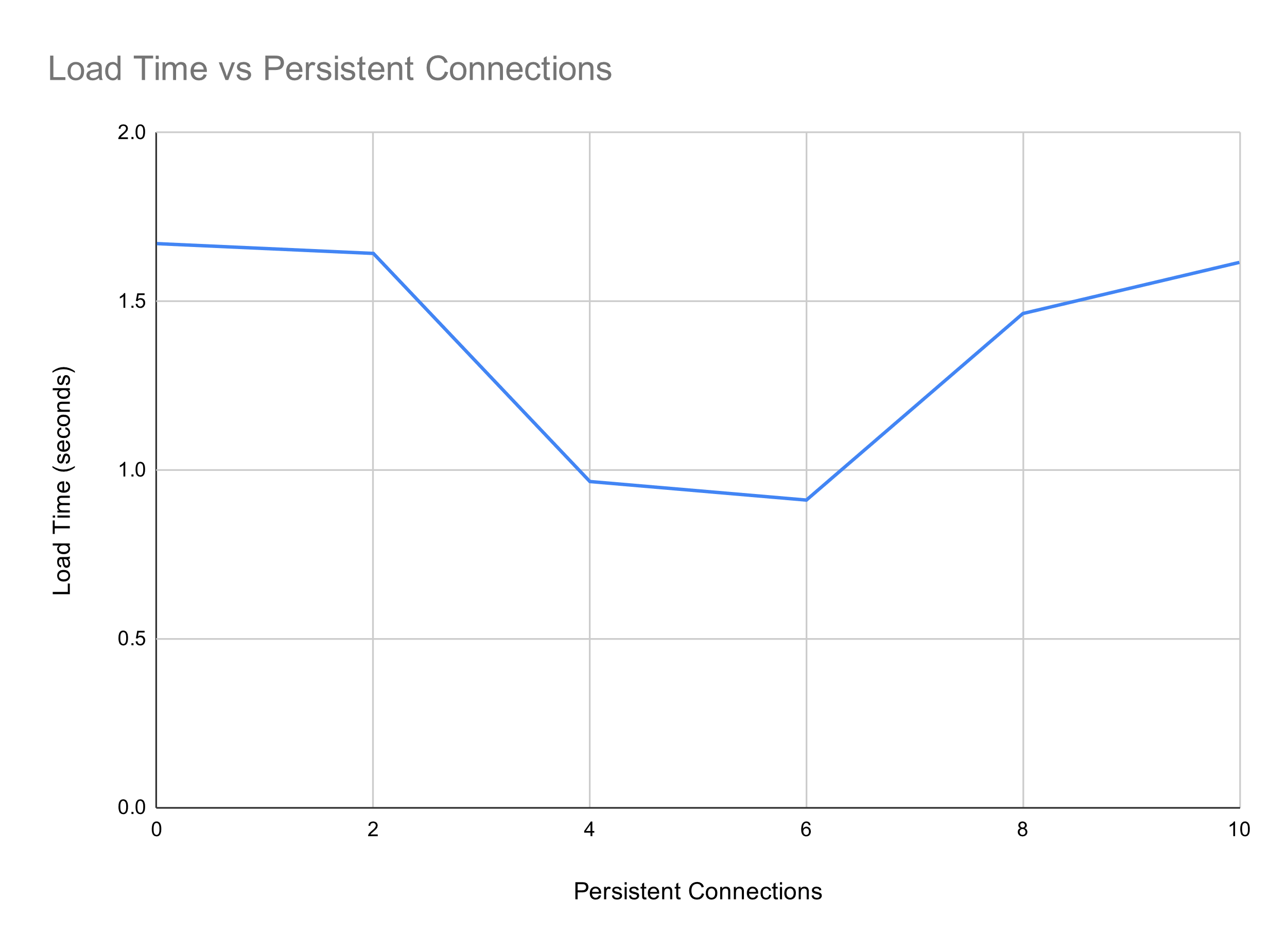


4. Observations

* We can calculate the total **load time** as the difference between the first GET time which corresponds to the time when the html page was requested and the last response time, which corresponds to when the last image was sent back.
* On doing so, we can construct the following observations table –

|  |  |  |  |
| --- | --- | --- | --- |
| Persistent Connections | Time at first GET | Time at last Response | Load Time |
| 0 | 16.89399 | 18.56394 | 1.66995 |
| 2 | 2.949395 | 4.590317 | 1.640922 |
| 4 | 7.952997 | 8.917951 | 0.964954 |
| 6 | 3.934909 | 4.844929 | 0.91002 |
| 8 | 1.818717 | 3.281478 | 1.462761 |
| 10 | 0.571076 | 2.185426 | 1.61435 |

* We can also plot the values of Load Time against the number of Persistent Connections to obtain the following visualisation.



* We can hence see that the optimal number of persistent connections is 6, since it corresponds to the lowest load time.
* Initially as the number of persistent connections increase, we can observe that the load time decrease gradually and then steeply. This occurs due to the parallelism and pipelining performed while processing and requesting for image objects.
* This allows for multiple images to be requested at the same time, hence decreasing the load time taken and is much lesser than requesting each individual image serially and individually.
* However, as the number of persistent connections increase, the load time again starts increasing. This is due to the decrease in throughput of each connection with the constant link capacity. Hence the load times increase with an increase in number of persistent connections above a certain threshold.
* It is therefore not suggested to keep an exceedingly high number of persistent connections.