

## **NETWORKS LAB ASSIGNMENT - 2**

1. a) i) HTTP 1.1 - with persistent connections:

Port 8111

The response from the server had HTTP/1.1 and Connection type Keep-Alive.

- ii) HTTP 1.1 - without persistent connections:

Port 8110

The response from the server had HTTP/1.1 and Connection type Close.

- iii) HTTP 1.0:

Port 8110

The response from the server had HTTP/1.0 and Connection type Close.

**Note:** In all these cases the request from our machine had HTTP/1.1 and Connection type Keep-Alive, which makes sense for there was no way for our machine to know what configuration the server was running on, so it went with the best thing possible. It also hints that HTTP/1.1 provides backward compatibility which is why the HTTP 1.0 could understand our requests.

- b) 17 GET requests were made in all of the connections (obviously).

- c) Port 8111: Wait time: 1ms - 7ms. Average: 3ms

Port 8110: Wait time: 1ms - 5ms. Average: 2ms

Port 8100: Wait time: 1ms - 5ms. Average: 2ms

- d) Port 8111: Total download time: 694ms

Port 8110: Total download time: 657ms

Port 8100: Total download time: 689ms

**Note:**

1. I believe that these connections times are pretty useless unless you test these connections are made back-to-back (even then it's kinda senseless to compare these) because the connection speed and response time varies with traffic.
2. All conditions being the same, we would expect persistent connection to have a lesser total download time, because all GET requests can be received asynchronously, whereas non-persistent connections wait for the response for the previous response, and have to open connection for each GET request.
3. Persistent connections may have a longer response time though, because they have to handle multiple GET requests from the same user at the same time, whereas non-persistent only have one request per user at a time.

e) User-Agent Field:

Mozilla/5.0 : Denotes that the browser is Mozilla compatible

Linux; Android 6.0; Nexus 5 Build/MRA58N : Denotes native platform the browser is running  
(Basically, the OS)

AppleWebKit/537.36 (KHTML, like Gecko) : The layout engine used to read web content  
(Mozilla developed Gecko, so Firefox uses that, I guess)

Chrome/63.0.3239.132 Mobile Safari/537.36 : Denotes the browser and its version

2. a) This portion is same for both modes:

3 way handshake:

[SYN]: 33989 → 21

[SYN, ACK]: 21 → 33989

[ACK]: 33989 → 21

Then, the server identifies itself with response code 220 as ProFTPD 1.3.5a Server  
(Debian): 21 → 33989

Client requests for features: [FEAT]: 33989 → 21

Server sends back features. It lists the language it uses as English US (en-US)

Client requests to use language English India (en-IN); [LANG] 33989 → 21

Server complies with response code 211 sending Features: 21 → 33989

Client requests to use UTF-8 encoding: [OPTS UTF8 ON]: 33989 → 21

Server complies with response code 200 with message UTF8 set to on: 21 → 33989

Client requests to set the options for MLST command: 33989 → 21

Server complies with response code 200: 21 → 33989

Client logs in as anonymous user: [USER]: 33989 → 21

Server accepts with response code 331. Requests to send email id as password:  
21 → 33989

Client sends empty password [PASS]: 33989 → 21

Server lets us in with response code 230. Displays its welcome message (welcome.msg as seen from ls command): 21 → 33989

Client requests for present working directory: [PWD]: 33989 → 21

Server replies with response code 257. "/" is the pwd: 21 → 33989

When we send the ls command, comes the difference:

i) Active Connection:

Client sends PORT request, informing the server to connect to its data port,  
which it mentions in the 'Active Port' header. (in my case, it was 54820):  
33989 → 21

The server responds with code 200: 21 → 33989

The server then initiates a 3-way handshake between its port 20 and my port 54820

ii) Passive Connection:

Client sends PASV request, requesting the server to inform its data port. which it mentions in the 'Active Port' header. (in my case, it was 54820):

33989 → 21

The server responds with code 227, mentioning its data port in 'Passive Port' header (in my case it was 41332): 21 → 33989

The client then initiates a 3-way handshake between its port 49890 (for my machine) and server's port 41332

Post the connection establishment, the data transfer process is the same. The client sends the LIST command on the command channel. The server sends response, containing information of the packet size it shall be sending. Data is sent on the data channel. After getting EOF, connection is closed.

**Note:**

1. When we run a different instruction, say cat welcome.msg, we send a RETR request.
2. Also, when we change the directory, say cd images, and then send a data request, say ls, then our machine first sends a CWD message, informing the server of our current working directory.
3. In my case, when the cache was on, even on changing the directory, the system was able to identify which list to display on ls, and didn't establish any new connections when I sent ls command from different directories.
4. If left idle, the server itself initiates closing of the idle connection.

b) Command channel - Active mode:

Sends command instructions like [PORT]

Connection establishment is from Client IP to Server IP.

Standard port on server side is 21

Connection is persistent (upto an idle time limit)

Command channel - Passive mode:

Sends command instructions like [PASV]

Connection establishment is from Client IP to Server IP.

Standard port on server side is 21

Connection is persistent (upto an idle time limit)

Data channel - Active mode:

Sends data and data-related metadata (like data packet size).

Connection establishment is from Server IP to Client IP.  
Server side port is the same for all connections made.  
Connection is closed after data transfer is complete.

Data channel - Passive mode:

Sends data and data-related metadata (like data packet size).  
Connection establishment is from Client IP to Server IP.  
Server side port changes for each connection made.  
Connection is closed after data transfer is complete.

- i) Server initiates data connection for active mode.
- ii) Client initiates data connection for passive mode.

c) Ports used for data channel:

Active connection:

Client side: 54820 (depends on your machine configuration)  
Server side: 20 (same for all connections from all users)

Passive connection:

Client side: 49890 (depends on your machine configuration)  
Server side: 41332 (keeps changing for each connection, from each user)

The difference for the passive mode was that for each new request a new port was issued by the server, while for the active connection, server made all connections from its port 20.