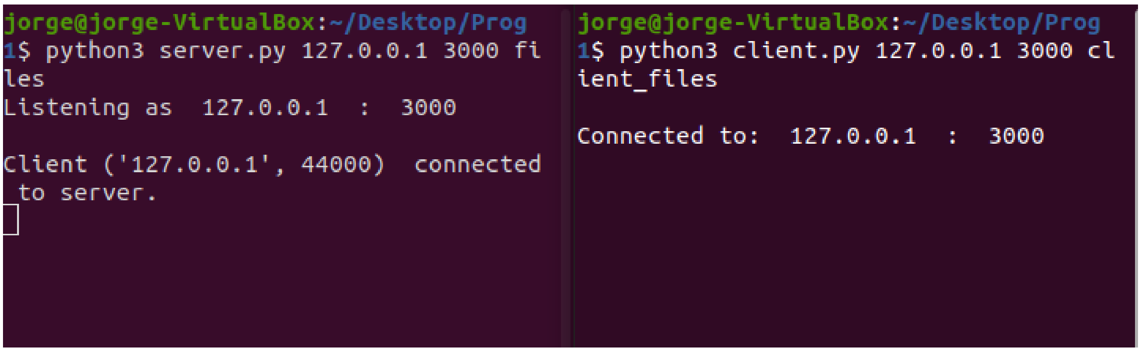
Output files and Performance evaluation results:

1. Connecting with client to the server and ensure requirements and 1 file transfer:

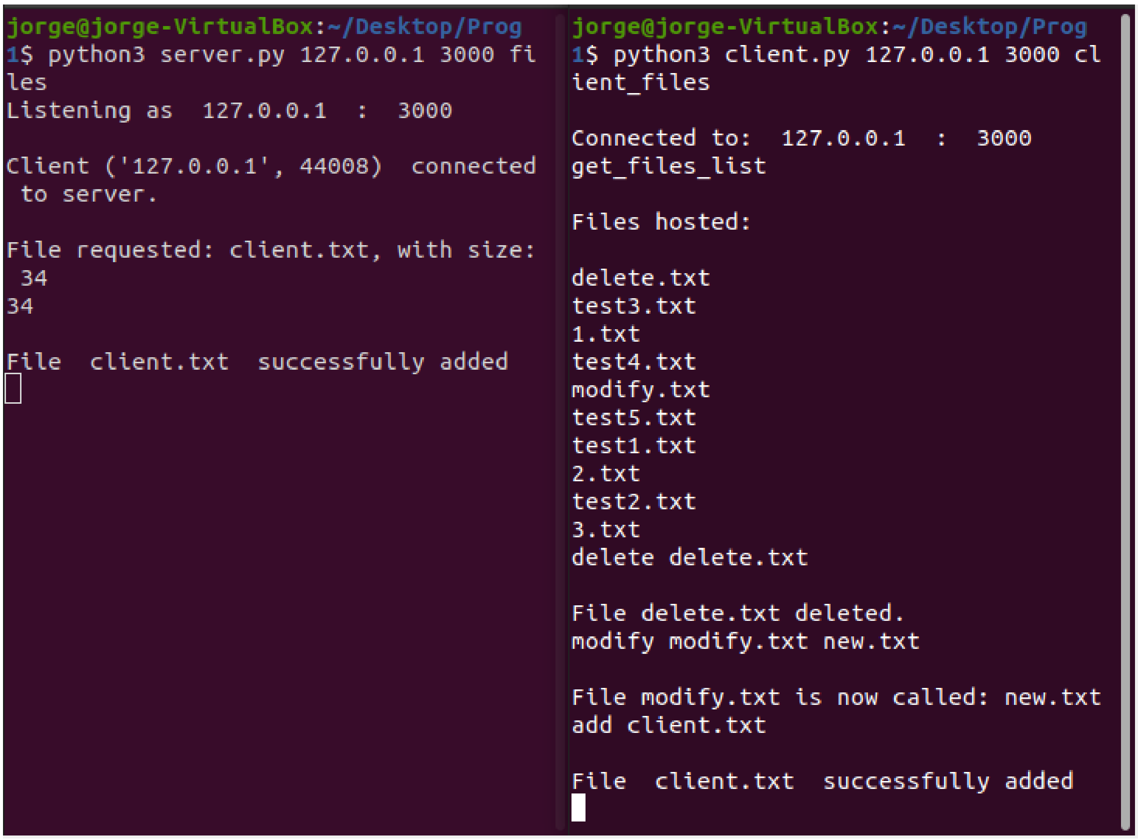
   - Connection: Client – Server Connection:



```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 server.py 127.0.0.1 3000 fi
les
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44000)  connected
 to server.
```

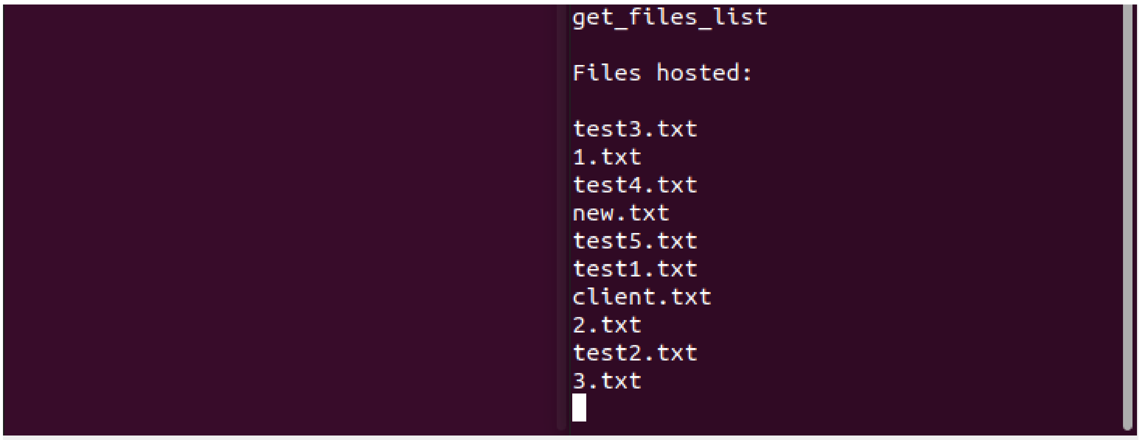
```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 client.py 127.0.0.1 3000 cl
ient_files

Connected to:  127.0.0.1  :  3000
```

   - Commands: get_files_list (Client gets the list of files that the server contains), delete (client removes one file from the server), add (client sends a file into the server) and modify (modify the name of a file from the server).



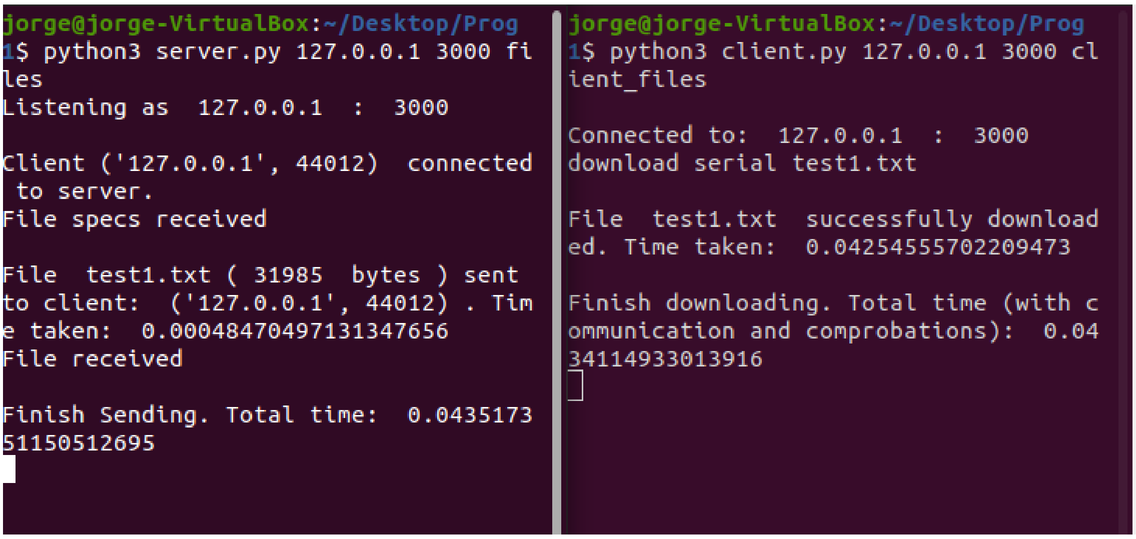
```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 server.py 127.0.0.1 3000 fi
les
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44008)  connected
 to server.

File requested: client.txt, with size:
 34
34

File  client.txt  successfully added
```

```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 client.py 127.0.0.1 3000 cl
ient_files

Connected to:  127.0.0.1  :  3000
get_files_list

Files hosted:

delete.txt
test3.txt
1.txt
test4.txt
modify.txt
test5.txt
test1.txt
2.txt
test2.txt
3.txt
delete delete.txt

File delete.txt deleted.
modify modify.txt new.txt

File modify.txt is now called: new.txt
add client.txt

File  client.txt  successfully added
```



```
get_files_list

Files hosted:

test3.txt
1.txt
test4.txt
new.txt
test5.txt
test1.txt
client.txt
2.txt
test2.txt
3.txt
```

- One File download: client download a file from the server.

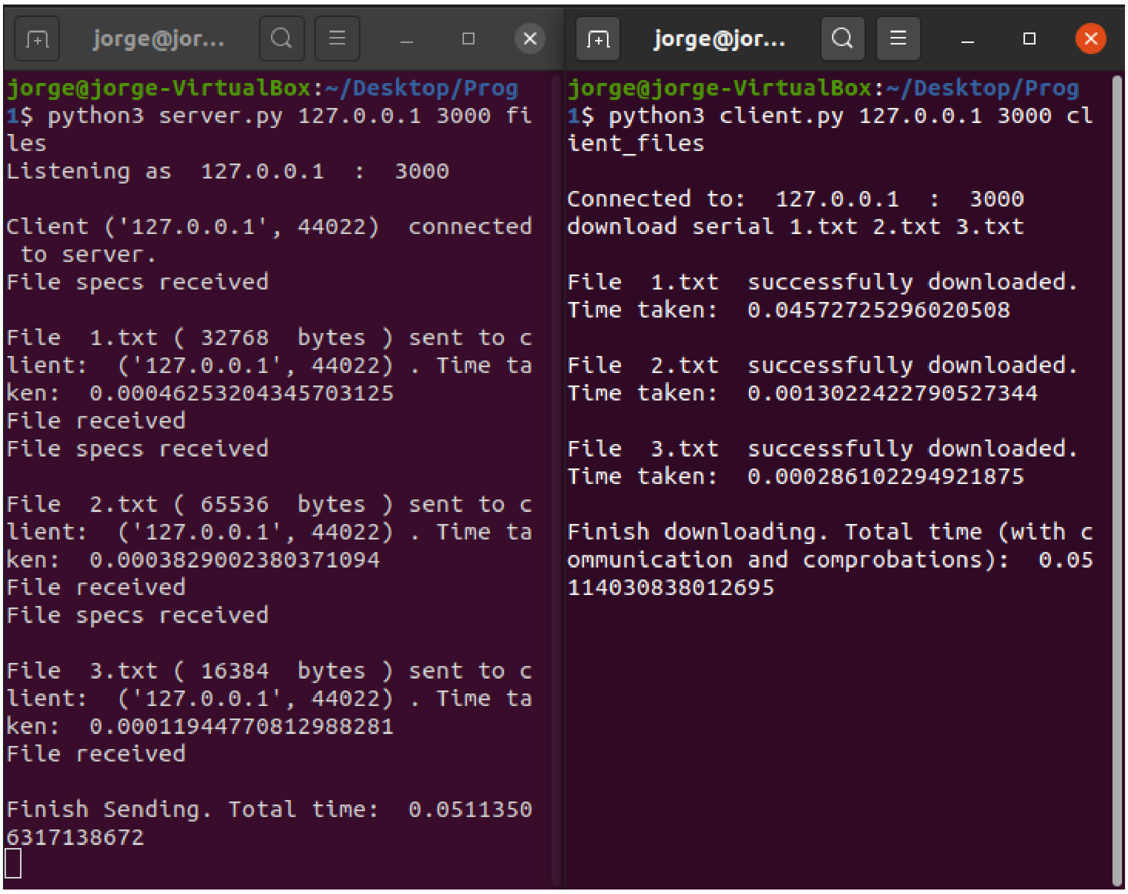


```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 server.py 127.0.0.1 3000 fi
les
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44012)  connected
 to server.
File specs received

File  test1.txt ( 31985  bytes ) sent
to client:  ('127.0.0.1', 44012) . Tim
e taken:  0.00048470497131347656
File received

Finish Sending. Total time:  0.0435173
51150512695
```

```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 client.py 127.0.0.1 3000 cl
ient_files

Connected to:  127.0.0.1  :  3000
download serial test1.txt

File  test1.txt  successfully download
ed. Time taken:  0.04254555702209473

Finish downloading. Total time (with c
ommunication and comprobations):  0.04
34114933013916
```

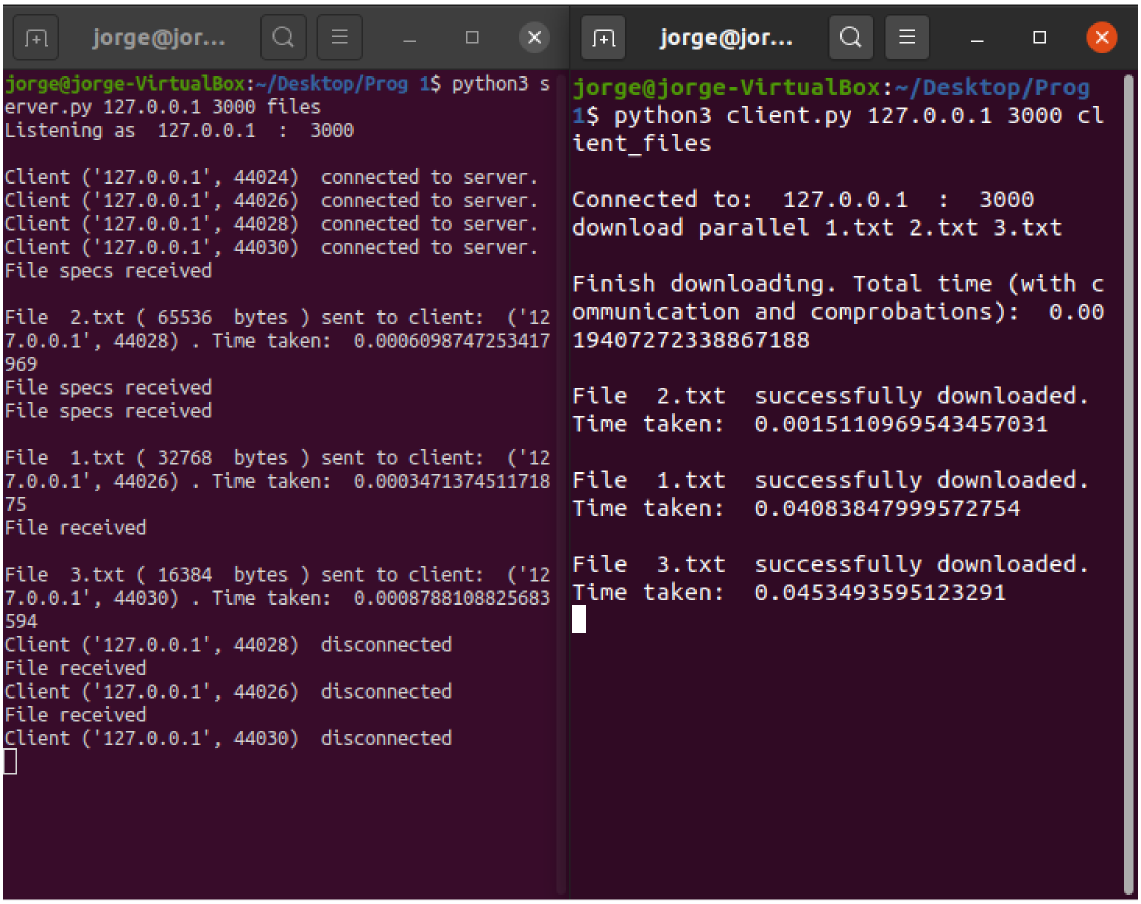
- Multiple File download: client download several files either in serial or parallel. For the parallel download, each file creates a new connection to the server which gets disconnected after the file transfer.

-



```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 server.py 127.0.0.1 3000 fi
les
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44022)  connected
 to server.
File specs received

File  1.txt ( 32768  bytes ) sent to c
lient:  ('127.0.0.1', 44022) . Time ta
ken:  0.00046253204345703125
File received
File specs received

File  2.txt ( 65536  bytes ) sent to c
lient:  ('127.0.0.1', 44022) . Time ta
ken:  0.0003829002380371094
File received
File specs received

File  3.txt ( 16384  bytes ) sent to c
lient:  ('127.0.0.1', 44022) . Time ta
ken:  0.00011944770812988281
File received

Finish Sending. Total time:  0.0511350
6317138672
```

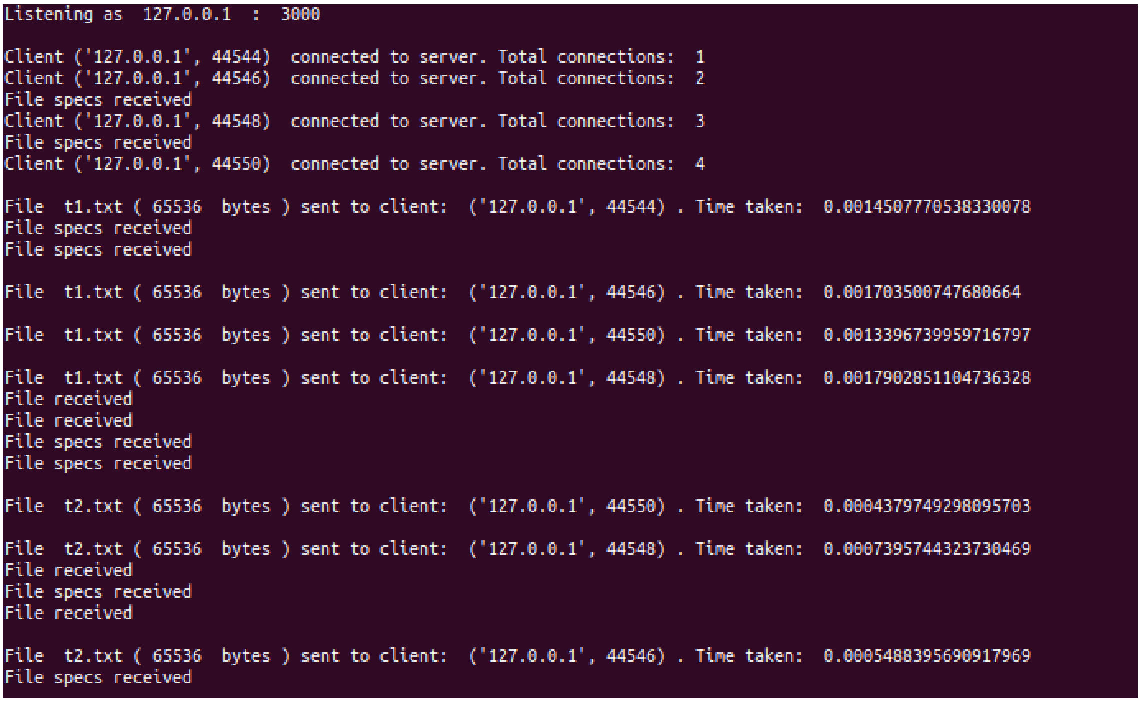
```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 client.py 127.0.0.1 3000 cl
ient_files

Connected to:  127.0.0.1  :  3000
download serial 1.txt 2.txt 3.txt

File  1.txt  successfully downloaded.
Time taken:  0.04572725296020508

File  2.txt  successfully downloaded.
Time taken:  0.0013022422790527344

File  3.txt  successfully downloaded.
Time taken:  0.000286102294921875

Finish downloading. Total time (with c
ommunication and comprobations):  0.05
114030838012695
```



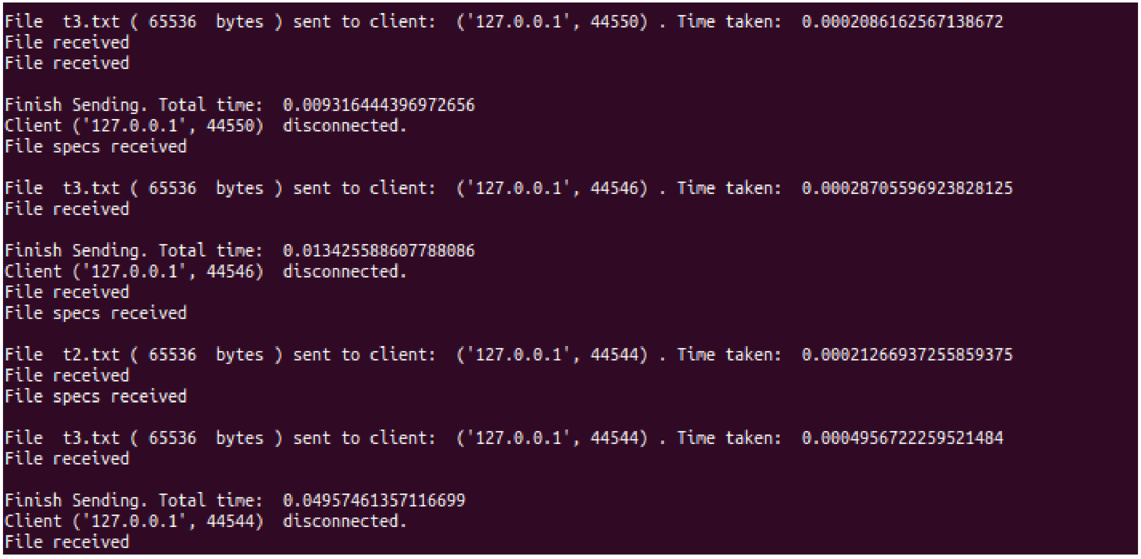
```
jorge@jorge-VirtualBox:~/Desktop/Prog 1$ python3 s
erver.py 127.0.0.1 3000 files
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44024)  connected to server.
Client ('127.0.0.1', 44026)  connected to server.
Client ('127.0.0.1', 44028)  connected to server.
Client ('127.0.0.1', 44030)  connected to server.
File specs received

File  2.txt ( 65536  bytes ) sent to client:  ('12
7.0.0.1', 44028) . Time taken:  0.0006098747253417
969
File specs received
File specs received

File  1.txt ( 32768  bytes ) sent to client:  ('12
7.0.0.1', 44026) . Time taken:  0.0003471374511718
75
File received

File  3.txt ( 16384  bytes ) sent to client:  ('12
7.0.0.1', 44030) . Time taken:  0.0008788108825683
594
Client ('127.0.0.1', 44028)  disconnected
File received
Client ('127.0.0.1', 44026)  disconnected
File received
Client ('127.0.0.1', 44030)  disconnected
```

```
jorge@jorge-VirtualBox:~/Desktop/Prog
1$ python3 client.py 127.0.0.1 3000 cl
ient_files

Connected to:  127.0.0.1  :  3000
download parallel 1.txt 2.txt 3.txt

Finish downloading. Total time (with c
ommunication and comprobations):  0.00
19407272338867188

File  2.txt  successfully downloaded.
Time taken:  0.0015110969543457031

File  1.txt  successfully downloaded.
Time taken:  0.04083847999572754

File  3.txt  successfully downloaded.
Time taken:  0.0453493595123291
```

2. Four clients are connected to the server and downloading files at the same time:



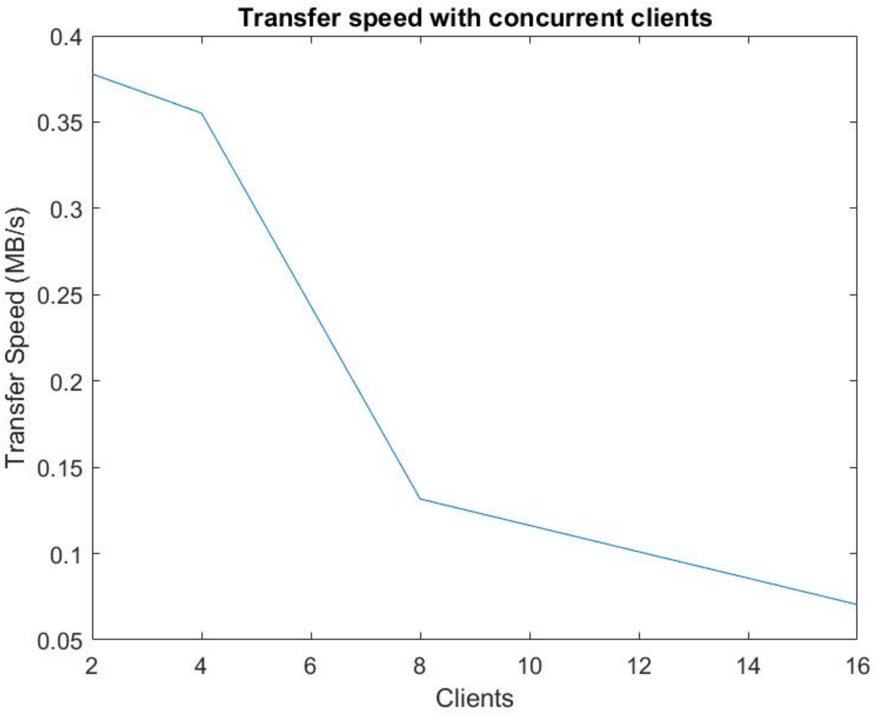
```
Listening as  127.0.0.1  :  3000

Client ('127.0.0.1', 44544)  connected to server. Total connections:  1
Client ('127.0.0.1', 44546)  connected to server. Total connections:  2
File specs received
Client ('127.0.0.1', 44548)  connected to server. Total connections:  3
File specs received
Client ('127.0.0.1', 44550)  connected to server. Total connections:  4

File  t1.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44544) . Time taken:  0.0014507770538330078
File specs received
File specs received

File  t1.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44546) . Time taken:  0.001703500747680664

File  t1.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44550) . Time taken:  0.0013396739959716797

File  t1.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44548) . Time taken:  0.0017902851104736328
File received
File received
File specs received
File specs received

File  t2.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44550) . Time taken:  0.0004379749298095703

File  t2.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44548) . Time taken:  0.0007395744323730469
File received
File specs received
File received

File  t2.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44546) . Time taken:  0.0005488395690917969
File specs received
```



```
File  t3.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44550) . Time taken:  0.0002086162567138672
File received
File received

Finish Sending. Total time:  0.009316444396972656
Client ('127.0.0.1', 44550)  disconnected.
File specs received

File  t3.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44546) . Time taken:  0.00028705596923828125
File received

Finish Sending. Total time:  0.013425588607788086
Client ('127.0.0.1', 44546)  disconnected.
File received
File specs received

File  t2.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44544) . Time taken:  0.00021266937255859375
File received
File specs received

File  t3.txt ( 65536  bytes ) sent to client:  ('127.0.0.1', 44544) . Time taken:  0.0004956722259521484
File received

Finish Sending. Total time:  0.04957461357116699
Client ('127.0.0.1', 44544)  disconnected.
File received
```

3. Measuring the transfer speed when varying the number of concurrent clients:

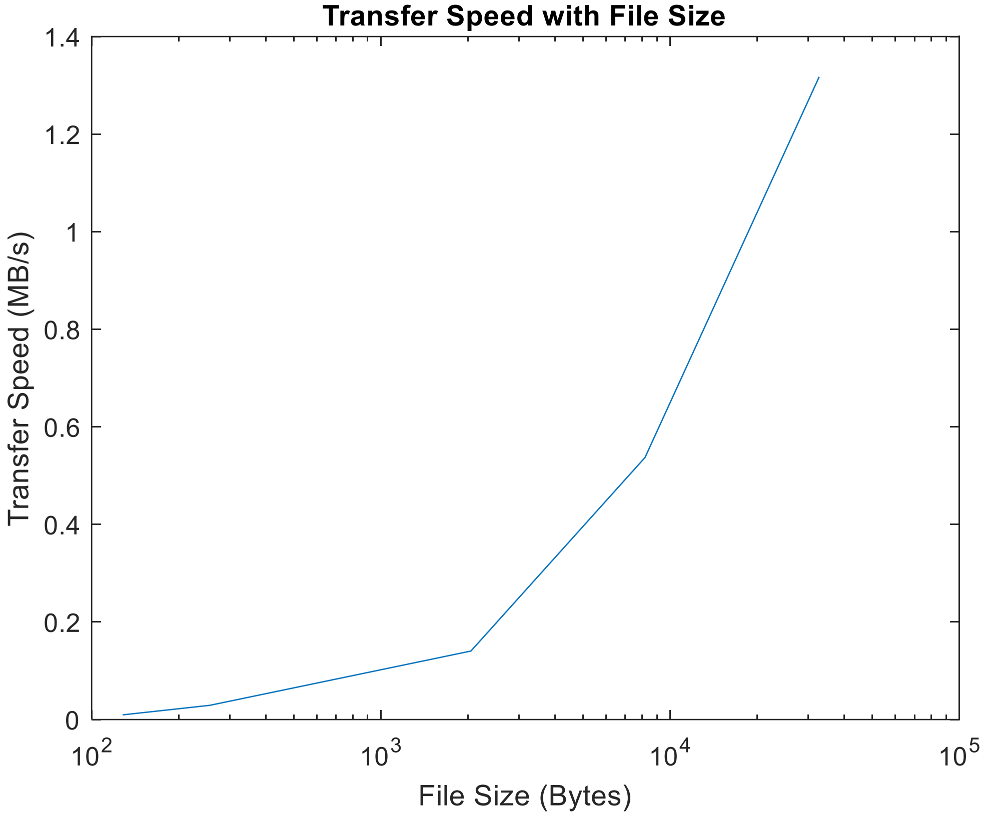
   A bash script has been created to connect N concurrent clients to the server and download in parallel 10 files each (each file creates a different thread). The size of the files is 64 KB each and the experiment has been repeated thrice for each number of clients (2, 4, 8 and 16). The results can be seen in the following graph:



   Therefore, increasing the number of concurrent clients downloading at the same time, decreases the transfer speed of the downloads.

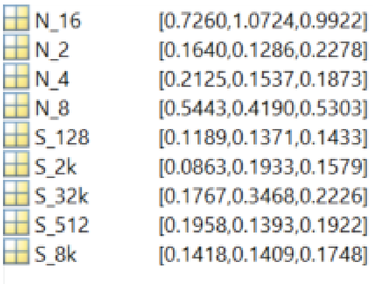
4. Measuring the transfer speed when varying the file size:

   A bash script has been created to connect 4 concurrent clients to the server and download in parallel 10 files each (each file creates a different thread). The size of the files varies (128, 512, 2k, 8k and 32k Bytes) and the experiment has been repeated thrice for each file size. The results can be seen in the following graph:



Although it may seem weird that the transfer speed increases with the file size, it is just a misconception.
Actually, the time measured in the graph is the one needed to receive all the data in the clients. However, this is not the transfer time but the time it took the client to read the data from the "buffer". The same happens at the server, the times printed are not the transfer times, but the times to write the data in the "buffer". Therefore, as all the sizes are relatively small, the time taken to read the data is approximately the same for all the files and, when these times get scaled by their file size, it looks as if the speed in increasing.

** As it was not compulsory to create a log for the clients, the output times (in seconds) for both graphs are the following:



N_16	[0.7260,1.0724,0.9922]
N_2	[0.1640,0.1286,0.2278]
N_4	[0.2125,0.1537,0.1873]
N_8	[0.5443,0.4190,0.5303]
S_128	[0.1189,0.1371,0.1433]
S_2k	[0.0863,0.1933,0.1579]
S_32k	[0.1767,0.3468,0.2226]
S_512	[0.1958,0.1393,0.1922]
S_8k	[0.1418,0.1409,0.1748]

N_xx: Time taken for the three experiments with xx concurrent clients.

S_xxx: Time taken for the three experiments with xxx Bytes files.