

Steps to access EC2 instance and start the tracker

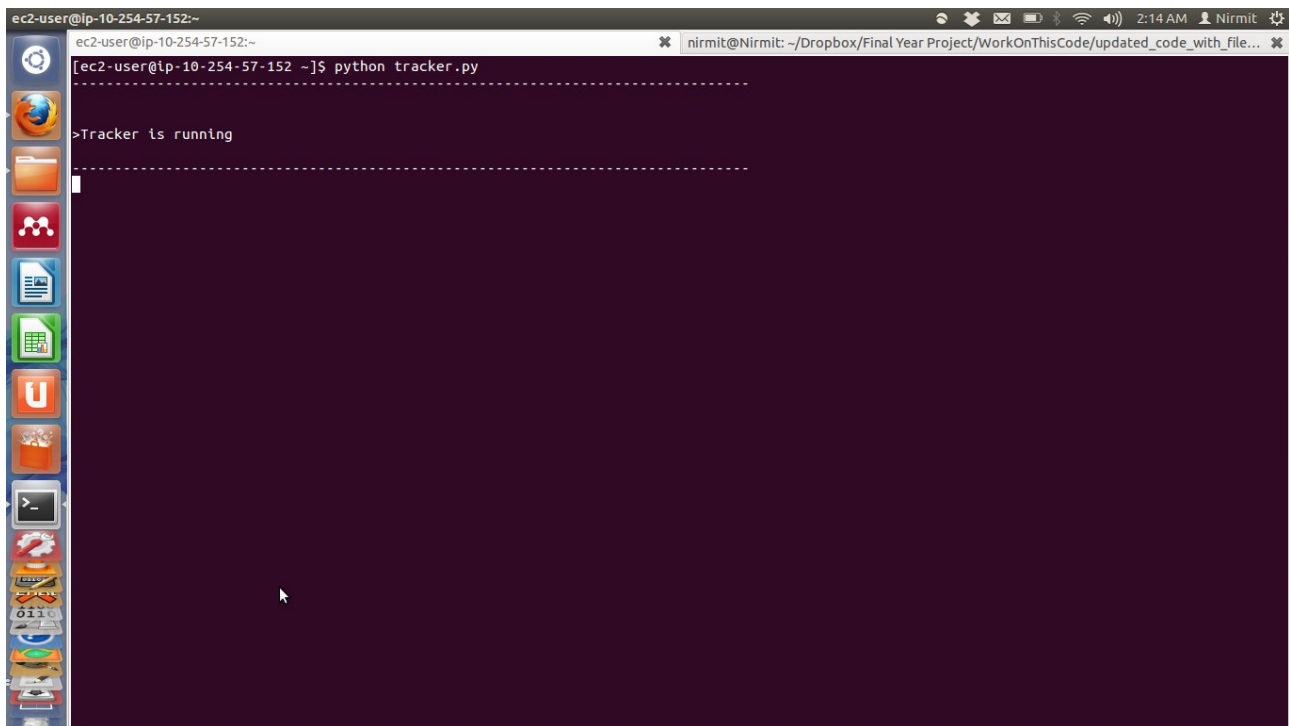
- 1) On linux, the syntax to login to EC2 instance is :
`ssh -i /path/to/private/key ec2_instance_name`

```
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$  
nirmit@Nirmit:~$ ssh -i ~/.ssh/fypkey.pem ec2-user@ec2-54-244-65-114.us-west-2.compute.amazonaws.com  
Last login: Sun May 26 19:38:07 2013 from 103.19.138.206  
  
  _ | _ | _ )  
 _ | ( _ | /  Amazon Linux AMI  
 _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/  
[ec2-user@ip-10-254-57-152 ~]$
```

- 2) Run the httpd service and verify that it's running

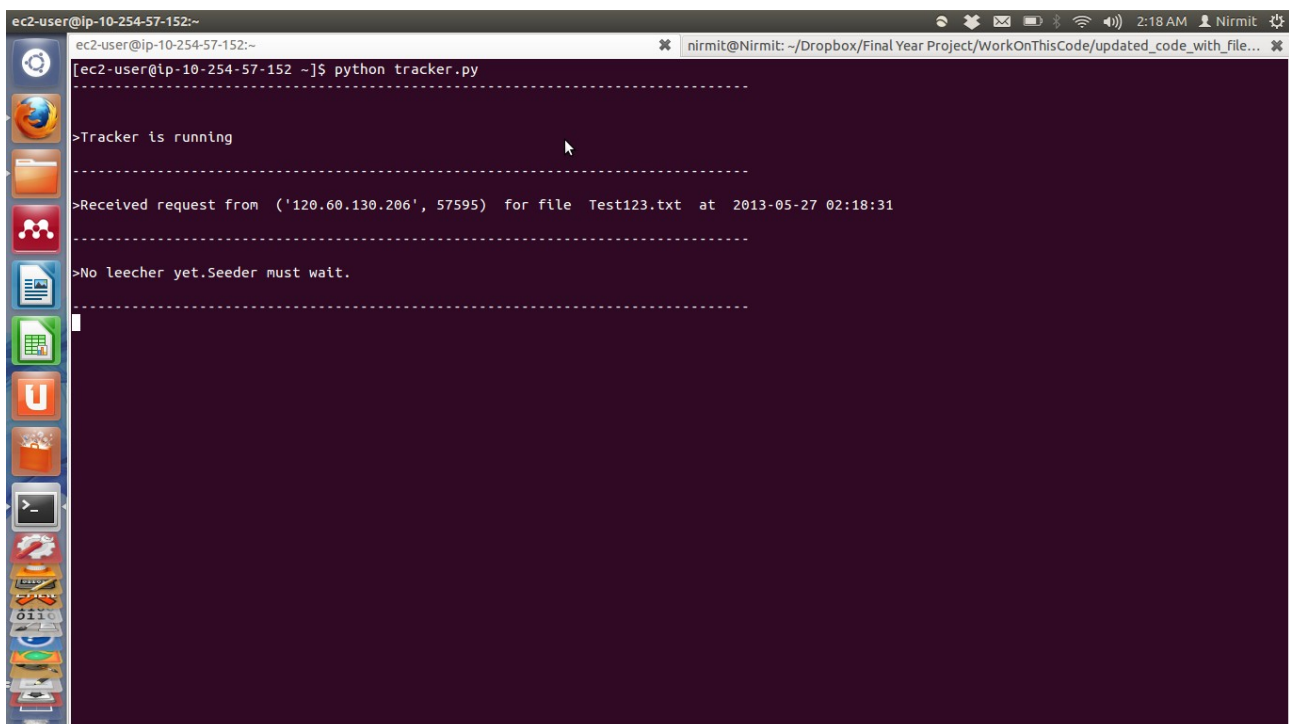
```
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$ sudo service httpd start  
Starting httpd:  
[ec2-user@ip-10-254-57-152 ~]$ sudo service httpd status  
httpd (pid 16612) is running...  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$  
[ec2-user@ip-10-254-57-152 ~]$
```

- 3) Run the tracker by using the command
`python tracker.py`

A terminal window on a Linux desktop. The terminal title bar shows the user 'ec2-user' and IP '10-254-57-152'. The command prompt shows the user has run 'python tracker.py'. The output of the program is 'Tracker is running', which is displayed between two dashed lines. The desktop background is dark purple, and a vertical dock with various application icons is on the left side.

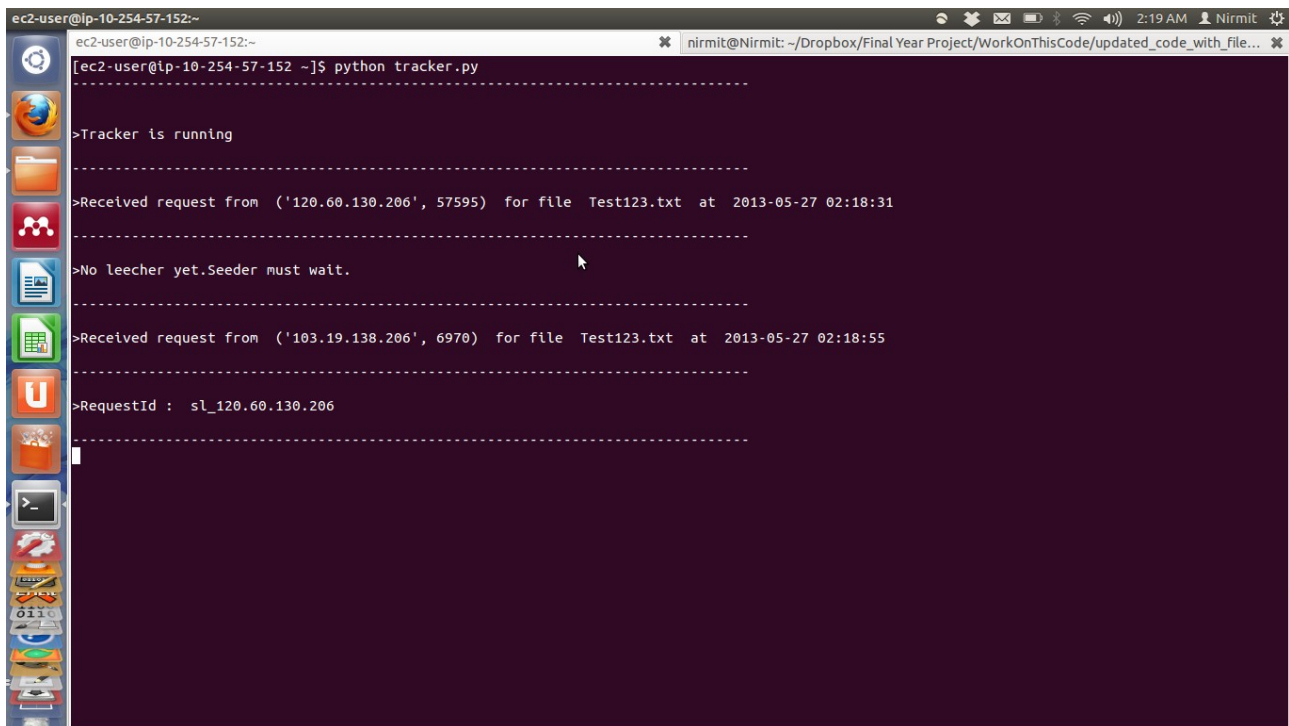
```
ec2-user@ip-10-254-57-152:~  
[ec2-user@ip-10-254-57-152 ~]$ python tracker.py  
-----  
>Tracker is running  
-----
```

4) If seeder , connects first , the tracker shows the following message :

A terminal window on a Linux desktop, similar to the first one. The command prompt shows the user has run 'python tracker.py'. The output shows 'Tracker is running' followed by a message indicating a request from a seeder: '>Received request from ('120.60.130.206', 57595) for file Test123.txt at 2013-05-27 02:18:31'. This is followed by another dashed line and then the message '>No leecher yet.Seeder must wait.'.

```
ec2-user@ip-10-254-57-152:~  
[ec2-user@ip-10-254-57-152 ~]$ python tracker.py  
-----  
>Tracker is running  
-----  
>Received request from ('120.60.130.206', 57595) for file Test123.txt at 2013-05-27 02:18:31  
-----  
>No leecher yet.Seeder must wait.  
-----
```

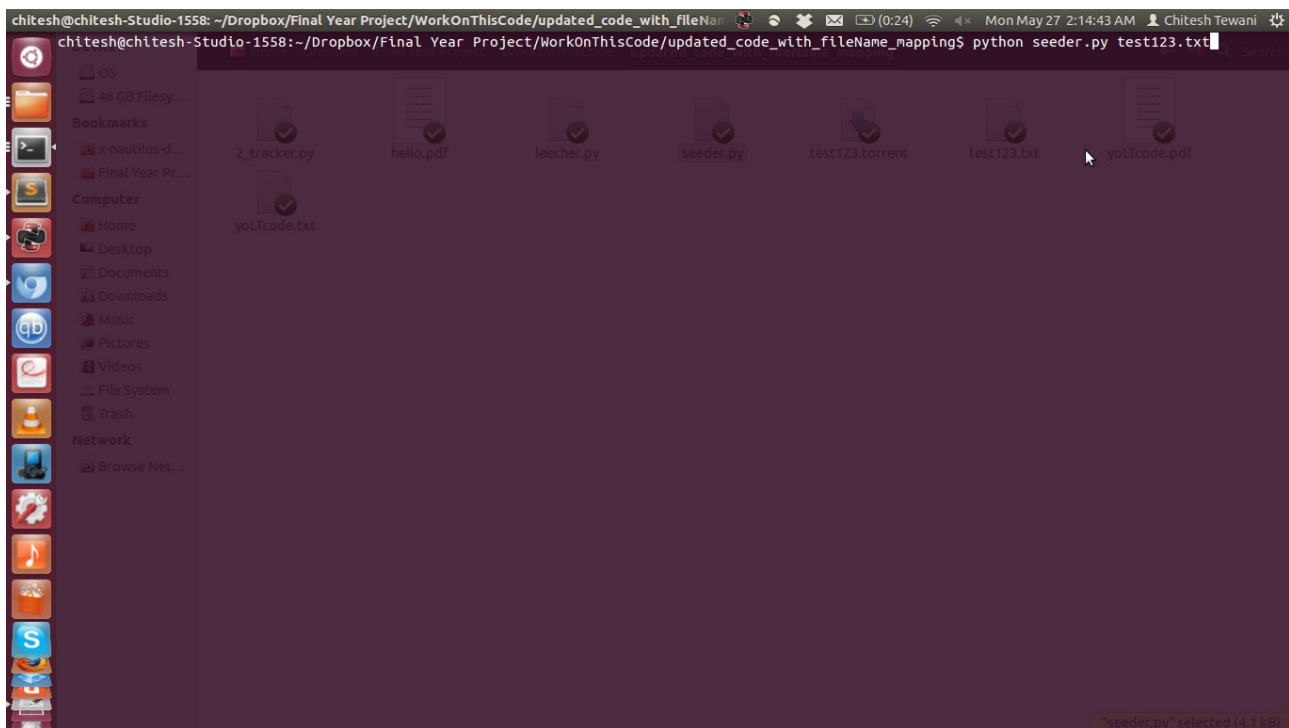
5) Once the leecher also connects, a requestID is generated



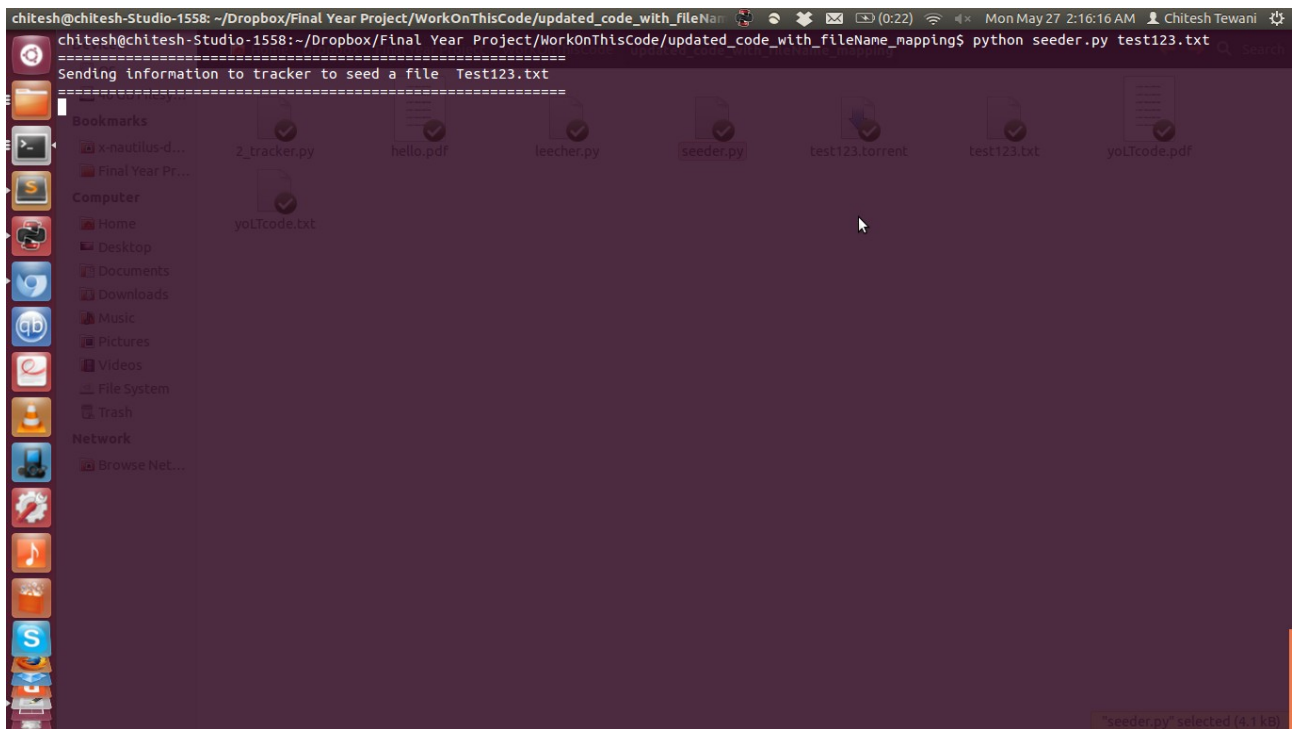
A terminal window titled 'ec2-user@ip-10-254-57-152:~' displays the output of a Python script named 'tracker.py'. The script is being executed in a directory path that includes '~/Dropbox/Final Year Project/WorkOnThisCode/updated_code_with_file...'. The output shows the tracker is running and has received two requests for a file named 'Test123.txt' at the same timestamp (2013-05-27 02:18:31). The first request is from IP '120.60.130.206' with port '57595'. The second request is from IP '103.19.138.206' with port '6970'. A message 'No leecher yet.Seeder must wait.' is displayed between the two requests. The last line of output shows 'RequestId : sl_120.60.130.206'.

```
ec2-user@ip-10-254-57-152:~  
[ec2-user@ip-10-254-57-152 ~]$ python tracker.py  
-----  
>Tracker is running  
-----  
>Received request from ('120.60.130.206', 57595) for file Test123.txt at 2013-05-27 02:18:31  
-----  
>No leecher yet.Seeder must wait.  
-----  
>Received request from ('103.19.138.206', 6970) for file Test123.txt at 2013-05-27 02:18:55  
-----  
>RequestId : sl_120.60.130.206  
-----
```

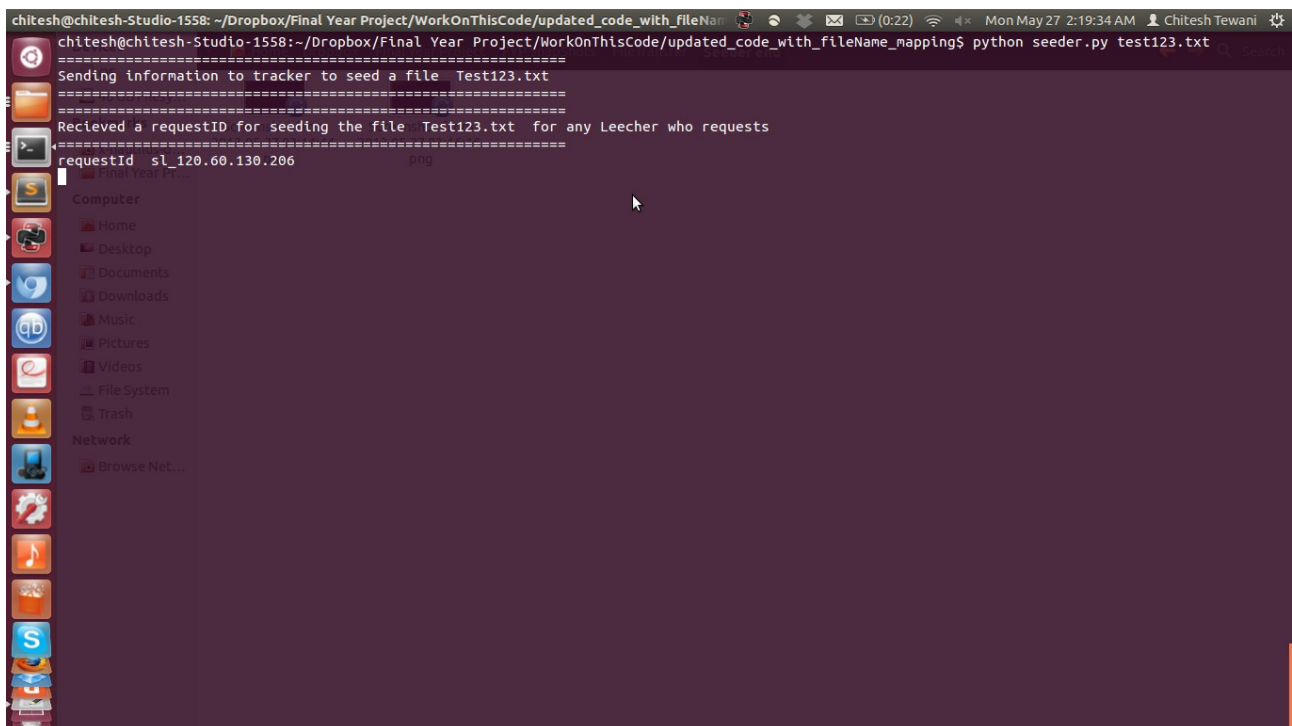
6) The seeder is run using following command :
python seeder.py fileToSeed



7) The seeder sends relevant information to tracker



8) The tracker then sends a requestID to the seeder and leecher



9)The leecher asks seeder for the hash of requestID and then checks if matches it's own requestID. If they match , it sends a confirmation message to seeder and starts receiving the file

```

chitesh@chitesh-Studio-1558: ~/Desktop/Final_Year_Project/File transfer
Datagram received and len of datagram = 16
=====process has started!!
RequestId=sl_127.0.0.1 and SeederTimestamp 2013-05-27 02:29:06
=====
Hash matched!!, now sending Confirmation to Seeder for File Transfer
=====
Datagram received and len of datagram = 18
=====
Receiving file process now begins!
=====
Degree = 2 Seed = 1623261189
=====
Data was formed by xoring blocks
(0, 3)
Datagram received and len of datagram = 18
=====
Receiving file process now begins!lock!!
=====
Degree = 1 Seed = 2072543609
=====
Data was formed by xoring blocks
(3,)
Hashtree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashindex= 1
Datagram received and len of datagram = 18
=====
Receiving file process now begins!
=====
Degree = 1 Seed = 3385713220
=====
Data was formed by xoring blocks
(1,)
Hashtree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashindex= 1fully decoded the file!!
=====
Stop transfers and ending connection with peer
=====
First half of the message received= abcd
=====
chitesh@chitesh-Studio-1558:~/Desktop/Final_Year_Project/File transfer$

```

10) The seeder starts seeding file on receiving confirmation message from leecher. It randomly corrupts a block to send

```

chitesh@chitesh-Studio-1558: ~/Desktop/Final_Year_Project/File transfer
=====
Hash verified!!
Seeder and Leecher are now Authenticated, File transferring process has started!!
=====
Host = ('127.0.0.1', 6970)
File size = 8 from ((0,): 'ab', (1,): 'cd', (2,): 'qf', (3,): 'g\n', (1, 3): '\x04n') edges
Number of Blocks in the file = 4
=====
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
Host = ('127.0.0.1', 6970) is correct
stuct packed
Host = ('127.0.0.1', 6970)
=====
Datagram received and len of datagram = 18
=====
Receiving file process now begins!Sending Corrupt Block!!
=====
Degree = 2 Seed = 2983805593
=====
The character 'a' appearing at some places in file
Data was formed is sent as character 'u'
(2, 3)
Hashtree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashindex= 1
=====
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
hash of piece 1 verified!
=====
=====
All blocks successful Reciever has successfully decoded the file!!
Keys generated= (0,)
Keys generated= (1) Stop transfers and ending connection with peer
Keys generated= (2,)
Keys generated= (3,)

```

11) The leecher verifies the blocks on the fly by using homomorphic hashing


```

chitesh@chitesh-Studio-1558: ~/Desktop/Final_Year_Project/File transfer
=====
Seeder and Leecher are now Authenticated, File transferring process has started!!
=====
First half of the message received= abcd
File size = 8
Number of Blocks in the file = 4
=====
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
Host = ('127.0.0.1', 6970)
stuct packed
Host = ('127.0.0.1', 6970)
Homomorphic Hash of piece 0 verified!!
=====
Start receiving Second half of the message
Sending Corrupt Block!!
=====
Datagram received and len of datagram = 18
Receiving file process now begins!
Degree = 2 Seed = 3322332987
Data was formed by xoring blocks
(1, 3)
HashTree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashIndex= 1
Datagram received and len of datagram = 18
Receiving file process now begins!
Degree = 4 Seed = 1592518114
Data was formed by xoring blocks
(0, 1, 2, 3)
HashTree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashIndex= 1

```

12) If a block is corrupted , it is deleted

```

chitesh@chitesh-Studio-1558: ~/Desktop/Final_Year_Project/File transfer
Degree = 4 Seed = 1592518114
Data was formed by xoring blocks
(0, 1, 2, 3)
HashTree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashIndex= 1
secondHalf message = qfg
Host = ('127.0.0.1', 6970)
Updating local HashTree for verification!
HashTree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 2 i = 1 j = 1 hashIndex= 5
Sending Corrupt Block!!
First half of the message received= qf
The character 'a' appearing at some places in file
Updating local HashTree for verification!
deleting key 2 from {(1, 2): '\x12\x02', (0,): 'ab', (1,): 'cd', (2,): 'qf', (3,): 'g\n', (1, 2, 3): 'u\x08'} edges
secondHalf message = g
Receiver has successfully decoded the file!!
Homomorphic hash of block(2) is correct
Stop transfers and ending connection with peer
Datagram received and len of datagram = 18

```

13) If homomorphic hash of each piece is verified , the decoding process ends and leecher gets the entire file

```
chitesh@chitesh-Studio-1558: ~/Desktop/Final_Year_Project/File transfer
=====
Seeder and Leecher are now Authenticated, File transferring process has started!!
=====
Updating local HashTree for verification!
File size = 8
-----
deleting key 2 from {(0,): 'ab', (1,): 'cd', (2,): 'gf', (3,): 'g\n', (1, 3): '\x04n'} edges
-----
secondHalf message = g
Host = ('127.0.0.1', 6970)
-----
Host = ('127.0.0.1', 6970)
stuct packed
Homomorphic hash of block(2) is correct
=====
Datagram received and len of datagram = 18
=====
Receiving file process now begins!
=====
Degree = 2 Seed = 2983805593
Data was formed by xoring blocks
(2, 3)
Hashtree in function = [1358, 1375, 9, 654, 806, 1540, 892]
Start = 0 i = 2 j = 2 hashIndex= 1
secondHalf message = efg
stuct packed
-----
stuct packed
Homomorphic hash of piece 1 verified!
=====
Reciever has successfully decoded the file!!
=====
All blocks successfully decoded rs and ending connection with peer
Keys generated= (0,)
Keys generated= (1,)
Keys generated= (2,)
Keys generated= (3,)
o-1558:~/Desktop/Final_Year_Project/File transfer$
```