[](https://www.myperfectresume.com/membership/RegisterGuestUser.aspx?wizard=true&productid=17&utm_source=hloom-com&utm_medium=referral&utm_campaign=word-template)

**Project Report**

[EE353 Computer Networks]

Dr. Arsalan Ahmed

**12-13-2018**

­­

Syed Abbas Hussain Naqvi

Muhammad Roshan Mughees

In our project basically we take a few arguments specifying multithreading level, type of connection, where to download the file from i.e., web address and where to save it like the path specified. Also there is a matric reporting the download speed after certain amount of time as specified by user.

For multithreading, we first retrieve the header of the web address and then check if “Accept-Ranges: bytes” specifying that it supports multithreading where as if it’s not bytes then the data from the website cannot be retrieved by multiple threads. So in that case we are just simply running the single thread.

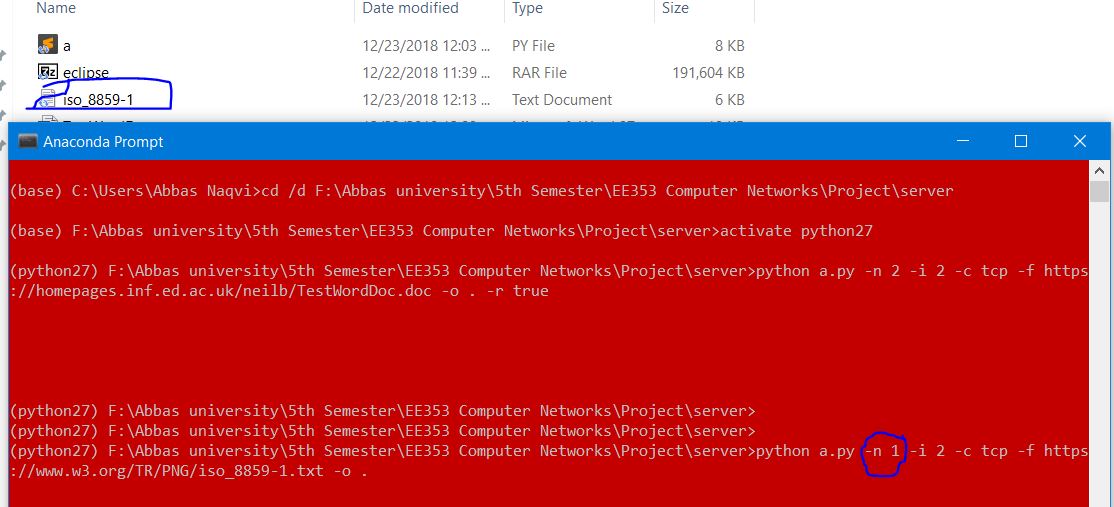
For resuming we first check the -r bit before creating thread. If resume bit is set then we check if some file chunk is present, if so, then starting range gets added with the file size already present. So that it starts downloading the bytes that remain instead of start downloading from start. If not then it creates all new files from scratch.

For downloading at a specific place we appended the path with the file name by using os.path.join() and then created file in append mode so that we can write in it all bytes downloaded by different threads which executed concurrently.

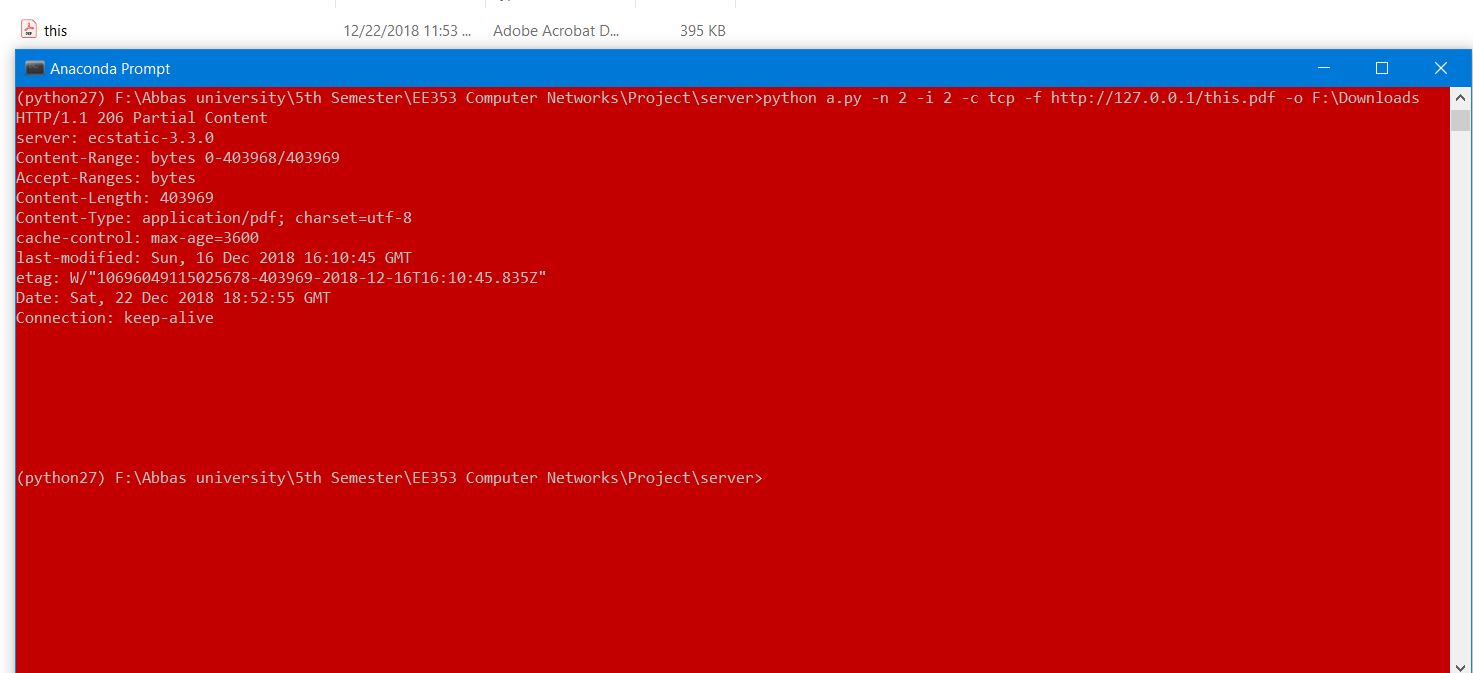
For matric reporting we created a thread which compares the current time in seconds with the time specified by user and when it becomes greater than the specified time then it issues a command os.system(“cls”) for clearing everything on the screen and prints the download speed and current bytes of all the threads present.

Here we implemented TCP and could not implement UDP as it shows the connection error. HTTP uses TCP because the files, images, web pages which we get from the remote host should not be dropped on the way and it should be delivered in order to the HTTP client. Even if UDP works then the packets will get lost in the way and the client will not ask to retransmit them.

## Case 1: Downloading with 1 connection

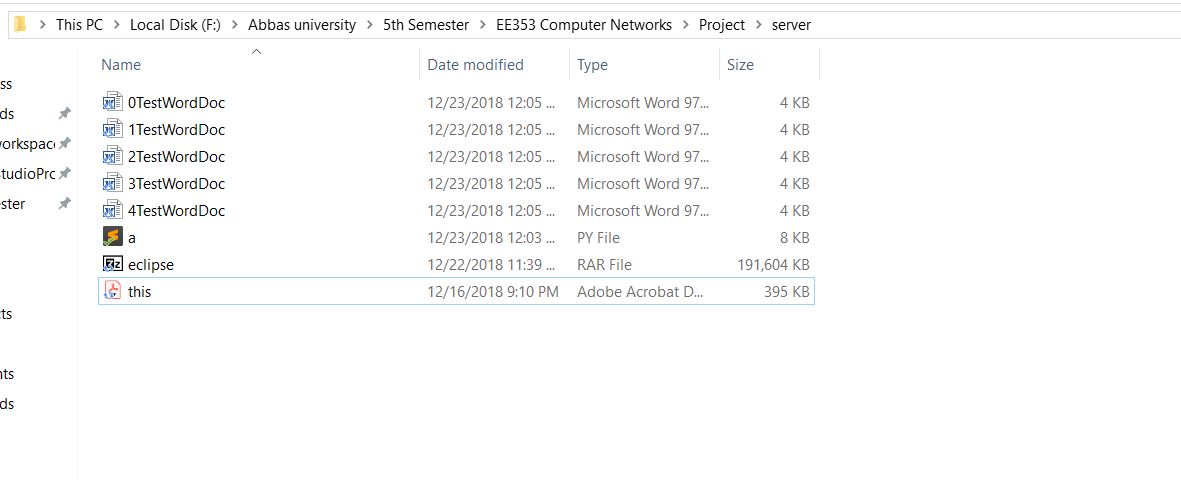


## Case 2: Downloading with Multiple connections

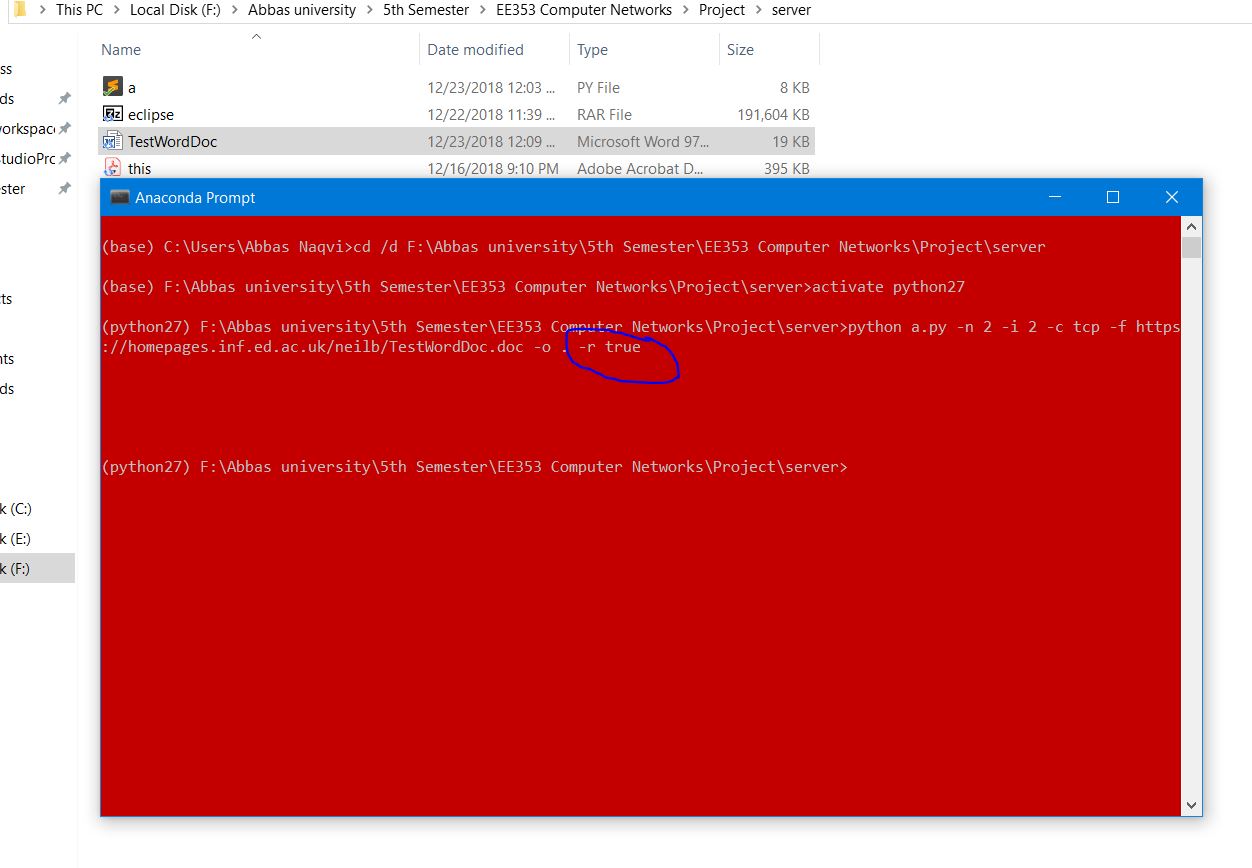


## Case 3: Resume Downloading

#### **Before:**

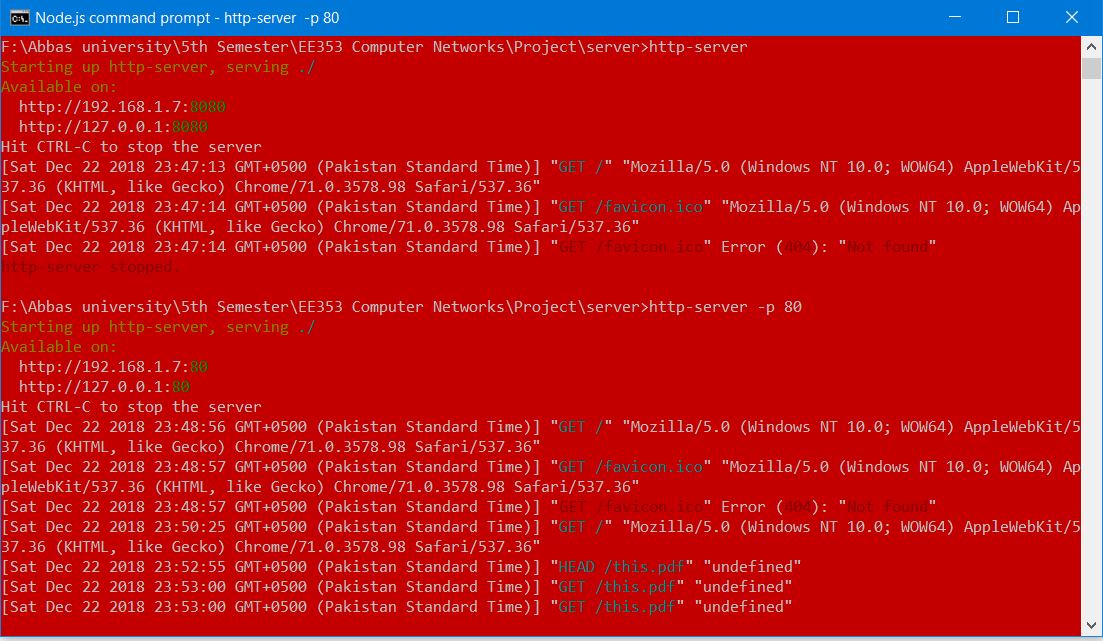


### **After**

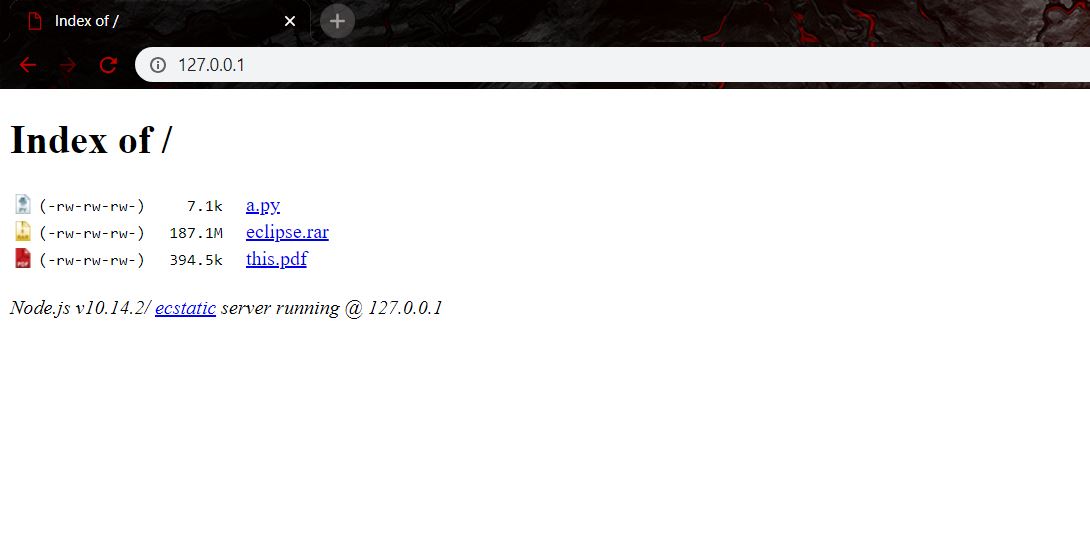


## Case 4: Downloading From local Server

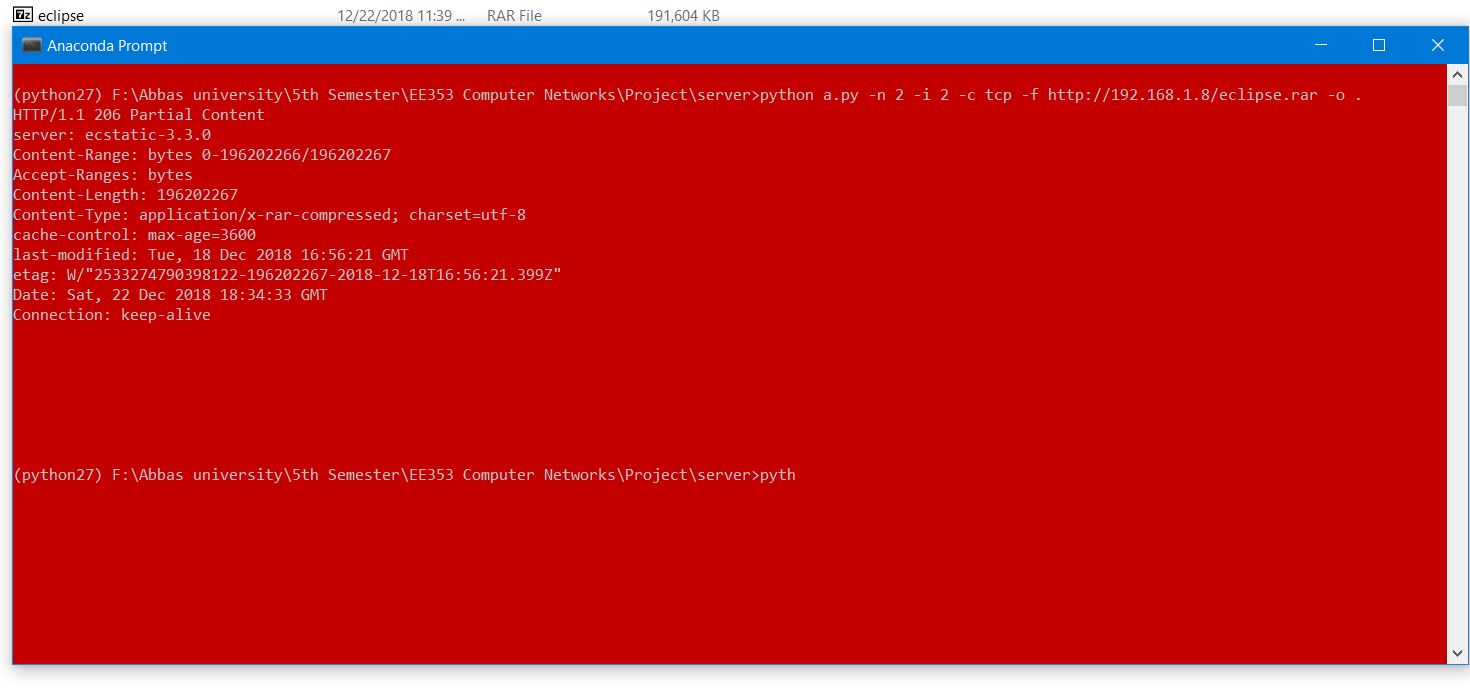
### **Server:**



**Data on Server**



### **Downloading file**



## Case 5: Performance Metrics

