ECE573 – Internet Protocols

Project I

Message formats:

* Register request from the peer to RS server: (Register\_Req)

POST Register\_Req PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

* Requesting list of peers in the system (PQuery)

GET PQuery PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

* Request the RSserver for extending the TTL (KeepAlive)

POST Keep\_Alive PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

* Requesting the RSserver to leave the system (Leave)

GET Leave\_Req PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4|  
<DATA>

* Requesting the list of RFCs stored in from another peer(RFC\_Index)

GET Get\_RFC\_Index PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

* Requesting a specific RFC stored in another peer(Download\_single)

GET Download\_single PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

* Requesting all the RFCs stored in another peer(Download\_all)

GET Download\_all PP/DI\_1.0\_LTS  
Hostname: <hostname>  
Port: <port>  
OS: Linux\_16.0.4  
<DATA>

***Distributed peer Index Simulation***

Plotting the RFC(X-Axis) vs. Time(Y-Axis)

**RUN-1**

Peer1-> Total time taken: 0.53127 seconds

Peer2 -> Total time : 0.4876

Peer3 -> Total time : 0.49412

Peer4 -> Total time : 0.84232

Peer5 -> Total time : 0.34885

Mean Total time for Run1: **0.5408seconds**

**RUN-2**

Peer1 -> Total time : 0.5067

Peer2 -> Total time : 0.4555

Peer3 -> Total time : 0.6125

Peer4 -> Total time: 0.6876

Peer5 -> Total time: 0.3041

Mean total time for Run 2: **0.51328seconds**

The average time taken for 5 peers to get 57RFCs each in a centralized system in our simulation is **0.52705seconds.**

***Centralized peer Simulation***

Plotting the RFC(X-Axis) vs. Time(Y-Axis)

**RUN-1**

Peer1 -> Total time : 0.30651seconds

Peer3 ->Total time: 0.32083seconds

Peer4 ->Total time: 0.8704seconds

Peer5 ->Total time: 1.009seconds

Mean time for RUN1 is **0.62668seconds**

RUN-2

Peer1 ->Total time: 0.2467seconds

Peer3 ->Total time: 0.27205seconds

Peer4 ->Total time: 0.76606seconds

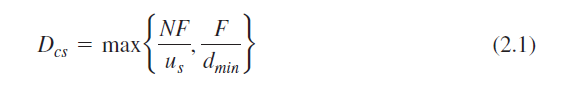
Peer4 ->Total time: 0.81820seconds

Mean time for RUN2 is **0.5257secsonds**

The average time taken for 4 peers to get 57RFCs each in a distributed peer index system in our simulation is **0.57619seconds.**

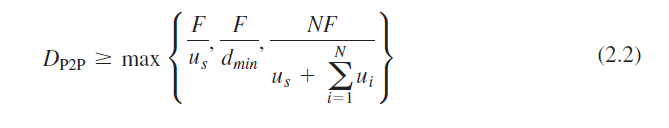
CONCLUSION:

1. We know from the book that centralized server system works well when the download speed and bandwidth of each of the peer system is good. (it also depends on the upload speed of the server, but it is typically good in real life systems).



From this equation given in the book, it is clear what the download time depends on.

1. In a distributed system the files are shared amongst all the connected peers in the system, therefore the server only uploads the file once and then the peers share them with eachother. In this system, the download speed is bottlenecked by this equation below:



Here it is easy to observe that the download rate is going to increase (download time will decrease) and perform better as the number of peers in the system is increased.

In our simulation, there were 2 peculiar things that we observed that we would like to mention:

1. As the simulation is performed amongst just 6 peers, we will not be able to really observe the gains of a distributed system and so we expected a huge gap between the centralized and distributed system’s download time. But the results are quite comparable and goes against what we expected. This is because, the simulated server that we used is another PC connected to the internet like every other peer (but in real life scenario, the central server usually has more bandwidth than the peers). Due to this the performance of the centralized system is observed to be only marginally better than the distributed systems.
2. The conclusion that we learnt from this simulation is the robust and plethora of use cases for the distributed systems in real life as long as the number of peers in the system is large. The conventional centralized system has its own advantages and performs better in interactive/live streaming kind of situations.