Internet Protocols (CSC/ECE 573)

Project I – Fall 2017

Niklesh Phabiani
nphabia@ncsu.edu
Gokul Yelchuru
gyelchu@ncsu.edu

NC State University

October 2017

Contents

1 Introduction

The project strives towards developing our understanding of client-server and peer-to-peer architecture. This is done by implementing a peer-to-peer system with distributed index. The system consists of a registration server and six peers- P0, P1, P2, P3, P4 and P5.

2 Flow

Each peer on joining the system has to first register itself with the registration server (RS). The RS stores the list of all registered peers in a database and makes each peer active by setting an activation flag to 'true' in the database. The peers receive a cookie number in response which they will use for all further communication with the RS. In order for a peer to communicate with any other peer, it first contacts the RS to get the list of active peers. The response from RS includes the hostname and the port at which the server of the active peer is listening. Since our tasks involve transfer of RFC files, each peer maintains an RFC index with itself that comprises of the list of all RFCs present with that peer. If the task involves centralized file distribution, all the RFCs and its corresponding entries would be present with P0 (60 in our case). On the other hand, if the task involves P2P distribution, each peer would contain a certain number of RFCs and its corresponding entries in the RFC index (10 with each peer in our case). Now, each peer updates its RFC index by requesting other peers for their RFC index by opening a connection on the server port of the peer. Only P0 is contacted in case of centralized file distribution whereas each peer contacts all other peers in case of P2P distribution. This is followed by downloading of RFC files for each peer. RFC files are downloaded by each peer by contacting P0 in case of centralized file distribution whereas in case of P2P distribution, a connection is opened with a particular peer who contains that RFC, and the peer who needs to be contacted is known by looking out for the hostname corresponding to the RFC number present in the RFC index. Once the hostname is known, server port of that peer is known through the active peers database that leads to the establishment of a TCP socket connection thereby leading to successful download of RFC files.

3 Task I: Centralized File Distribution

3.1 Download Time curve for Centralized File Distribution

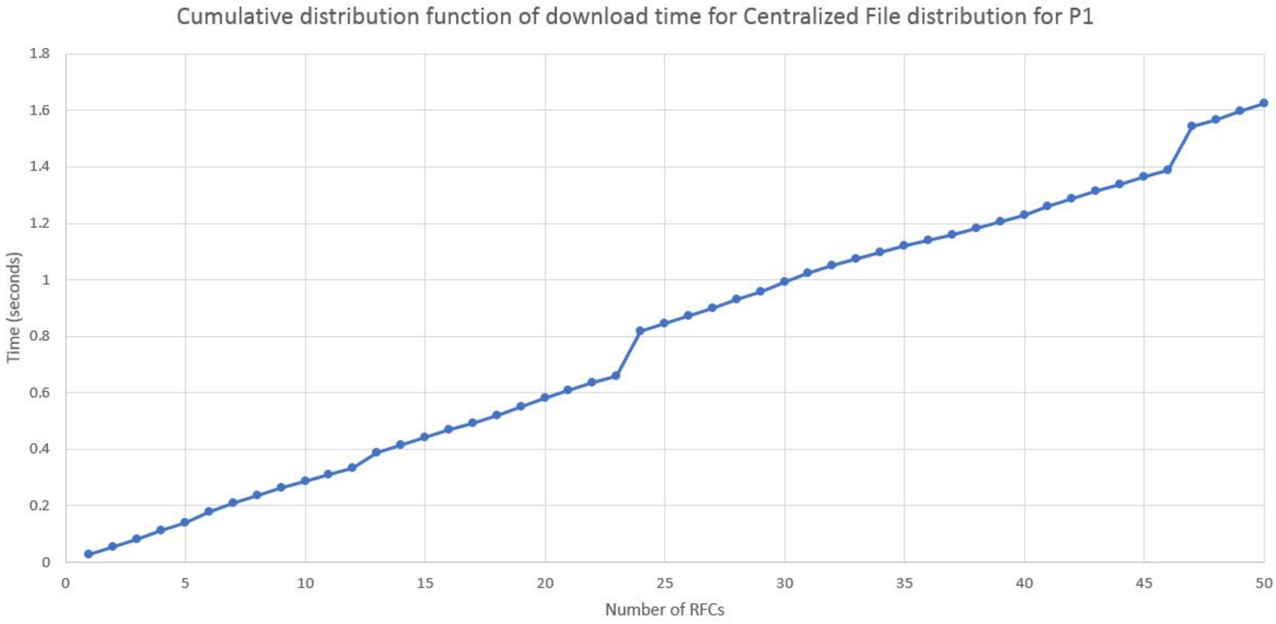

Figure 1: Peer 1

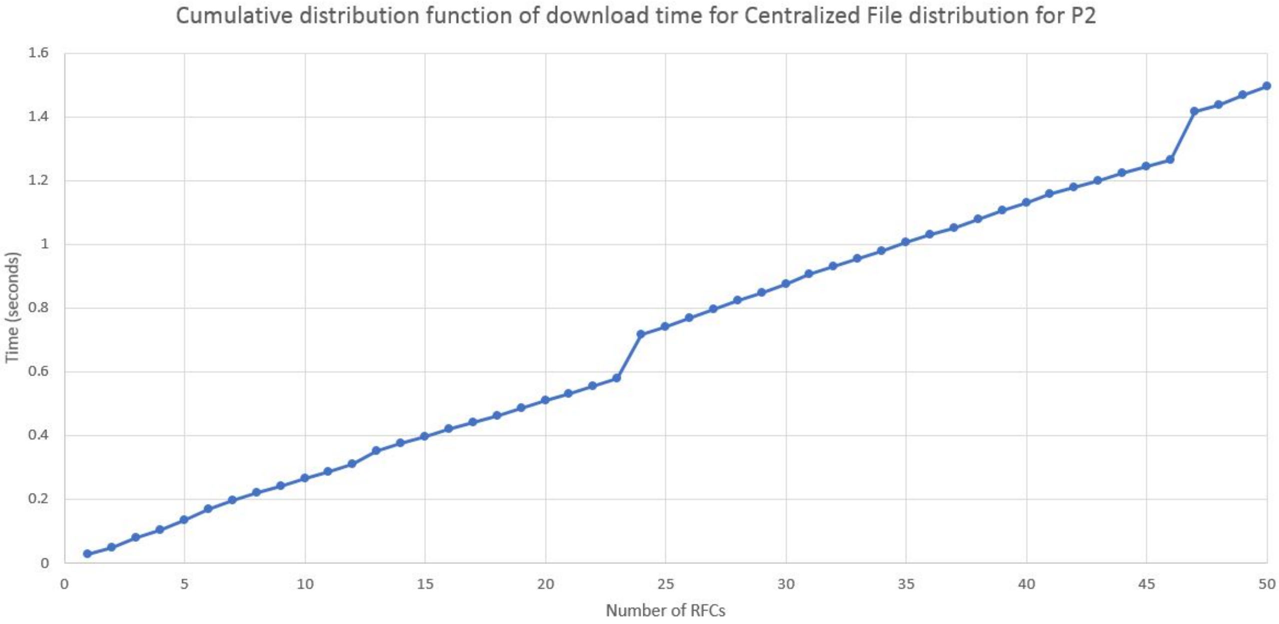

Figure 2: Peer 2

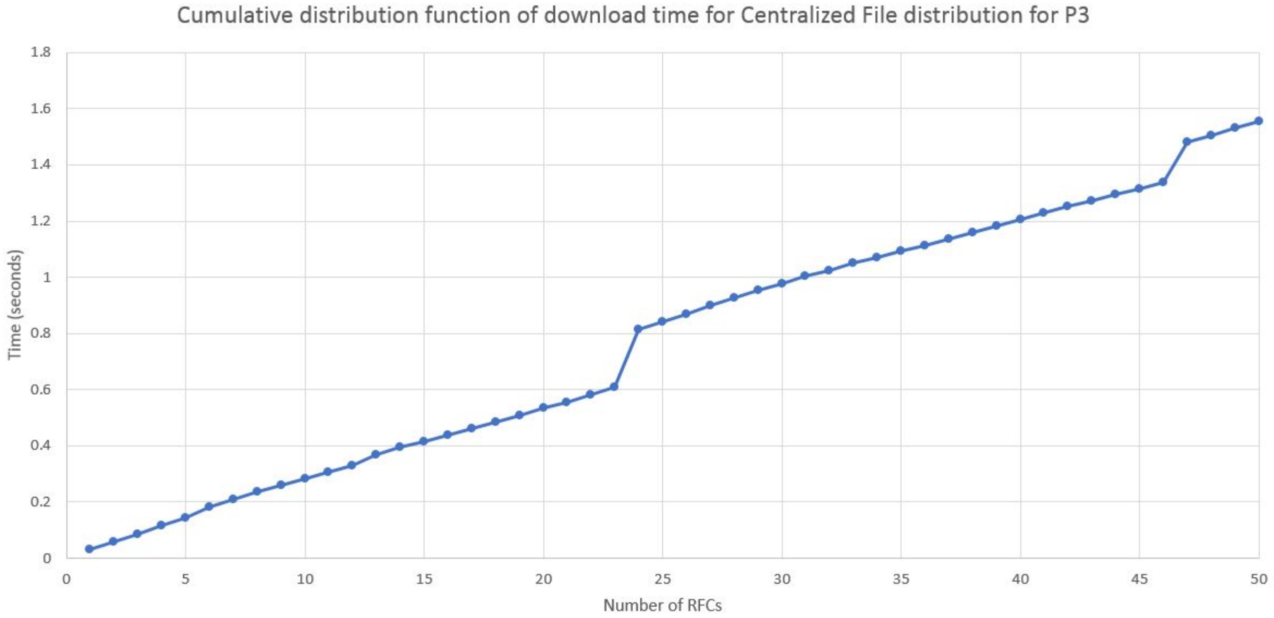

Figure 3: Peer 3

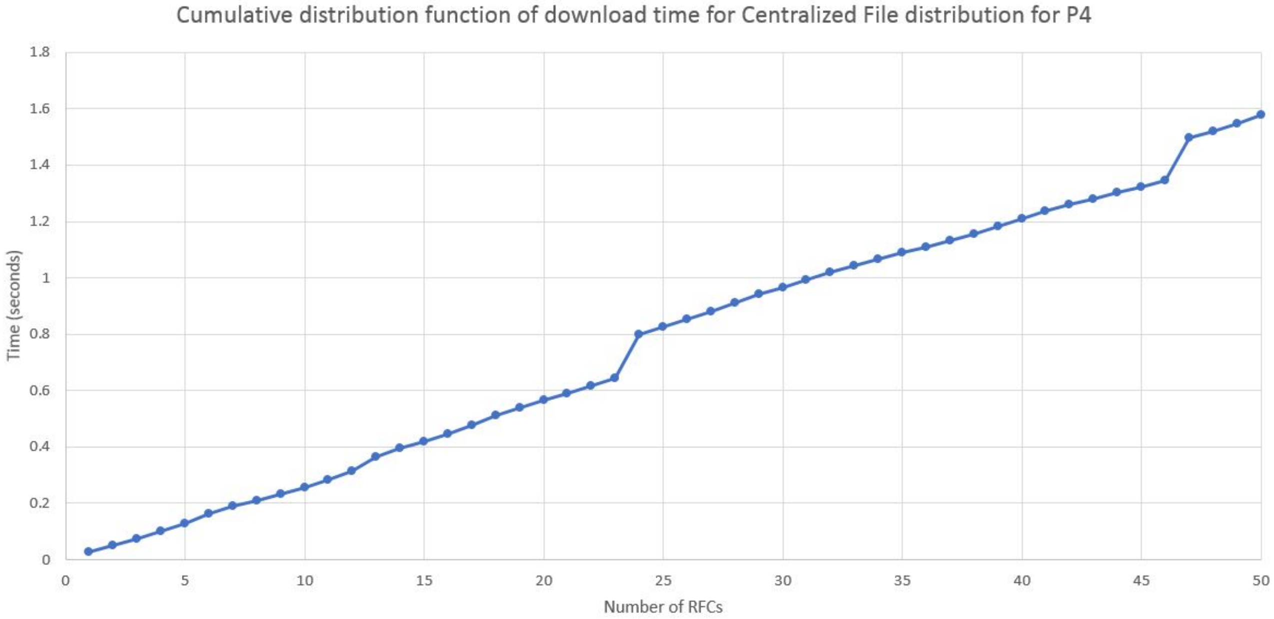

Figure 4: Peer 4

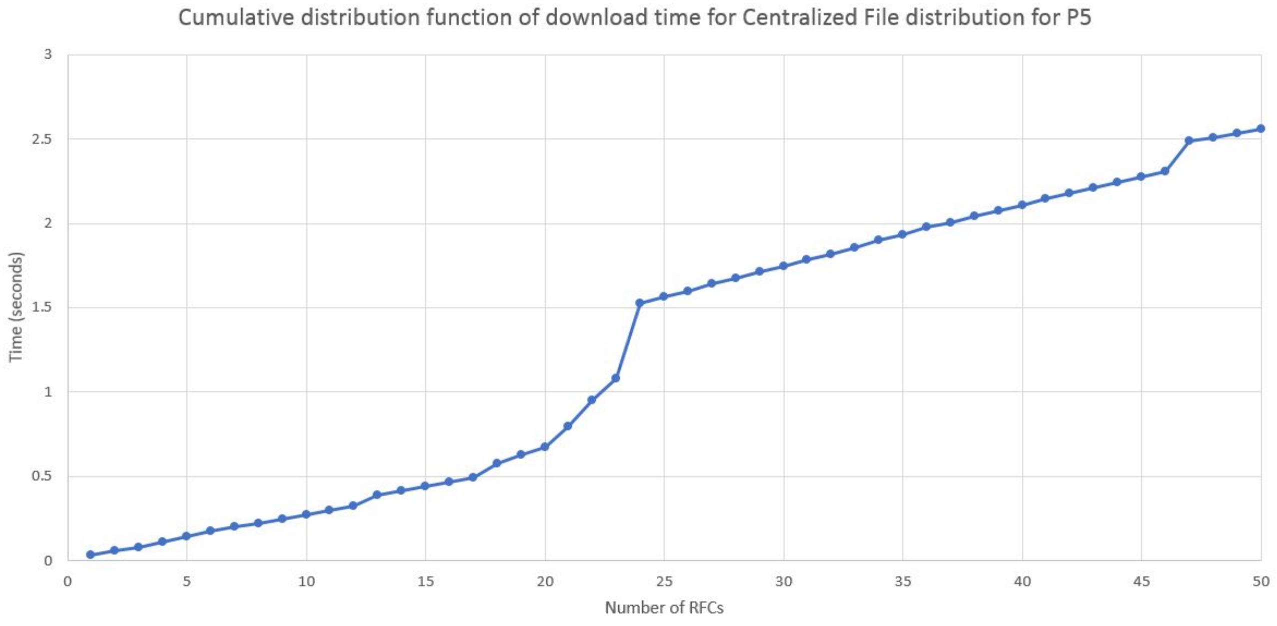

Figure 5: Peer 5

4 Task II: P2P File Distribution

4.1 Download Time curve for P2P File Distribution

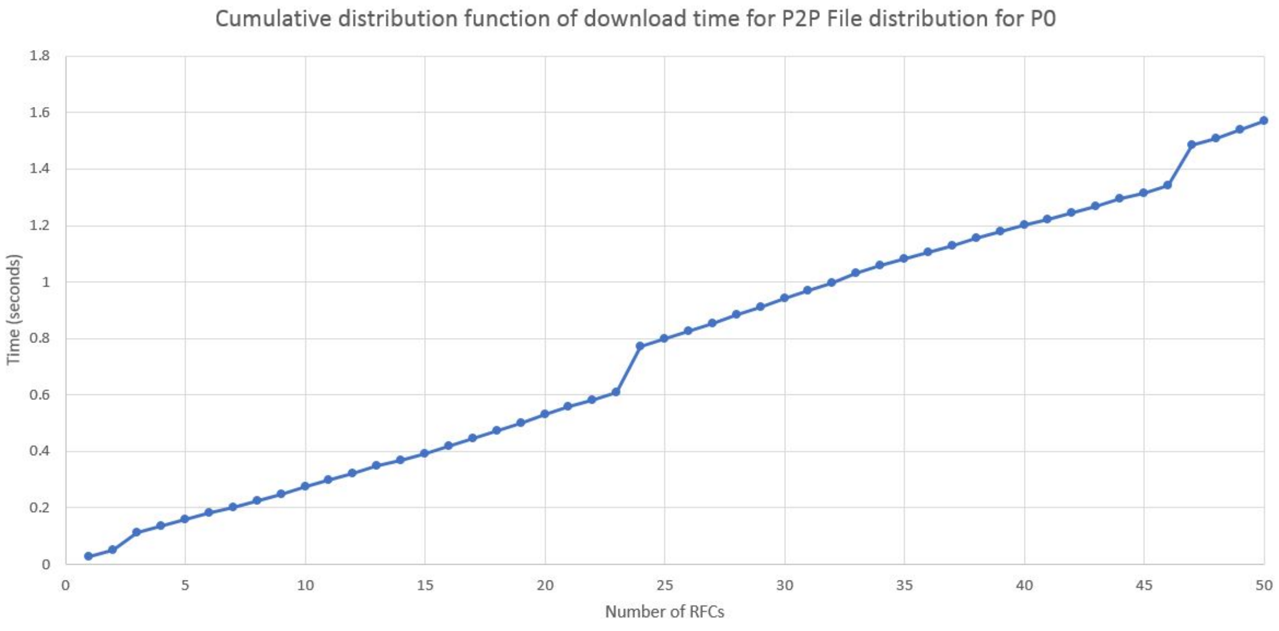

Figure 6: Peer 0

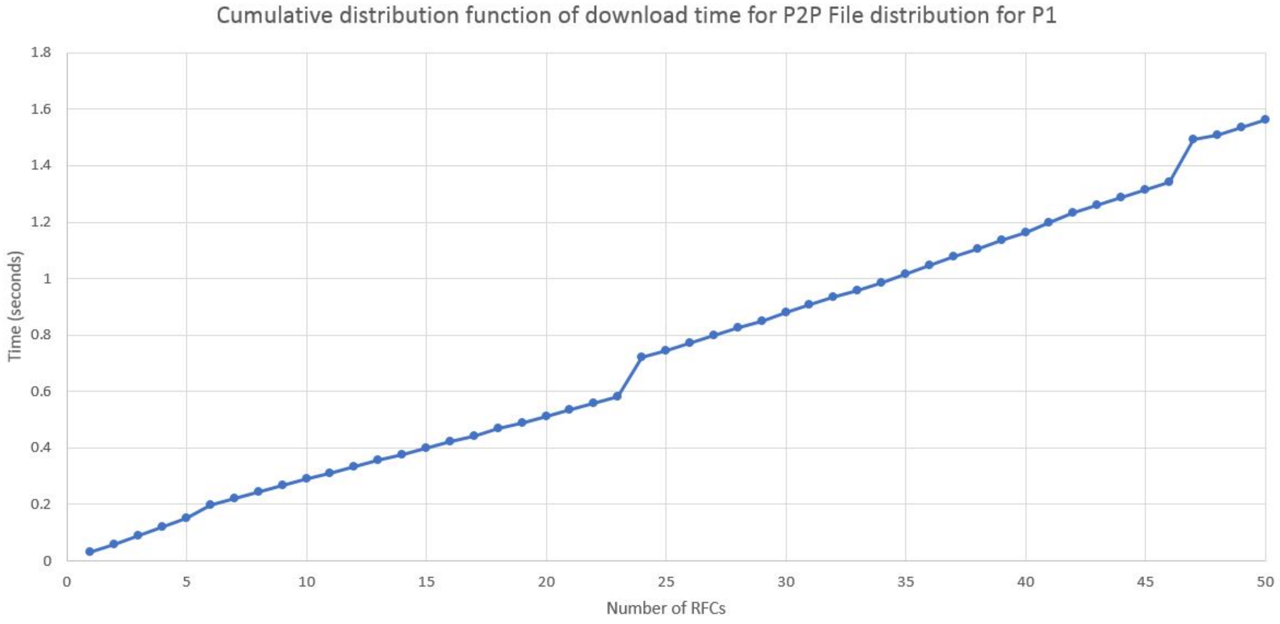

Figure 7: Peer 1

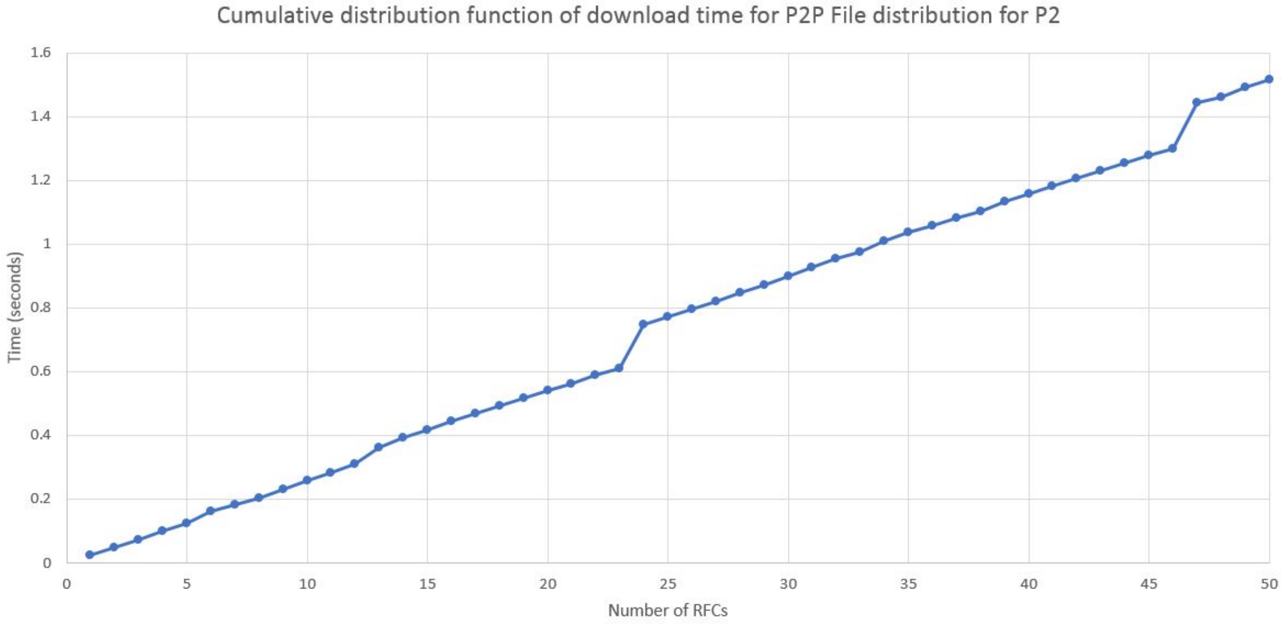

Figure 8: Peer 2

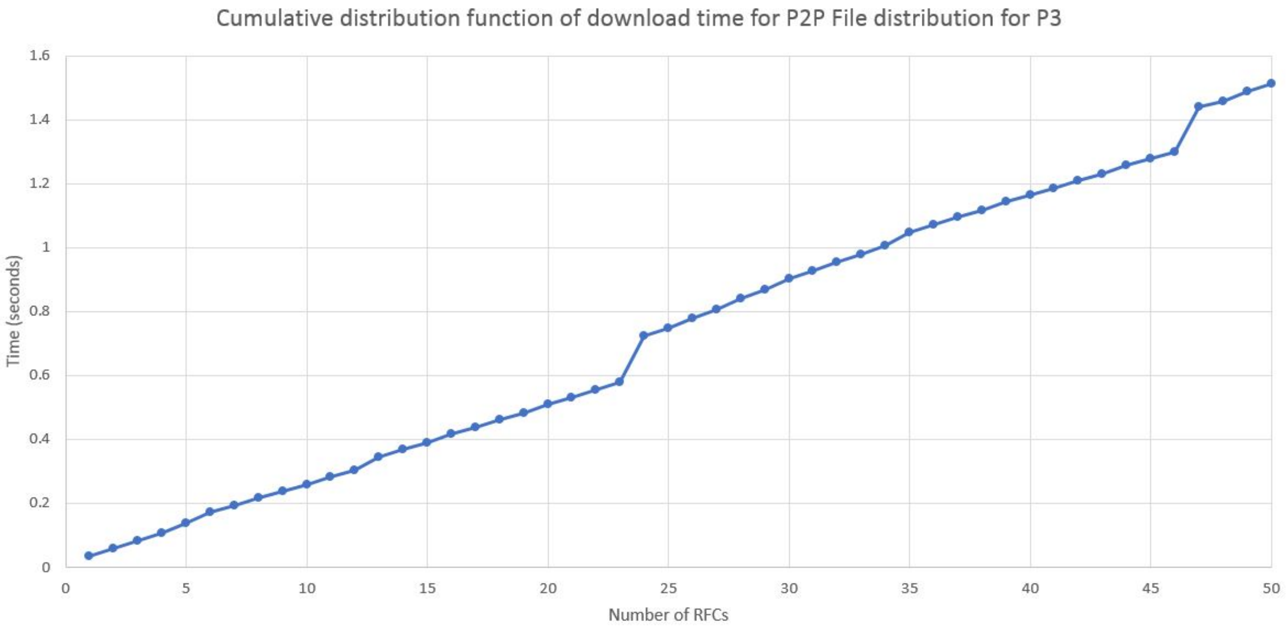

Figure 9: Peer 3

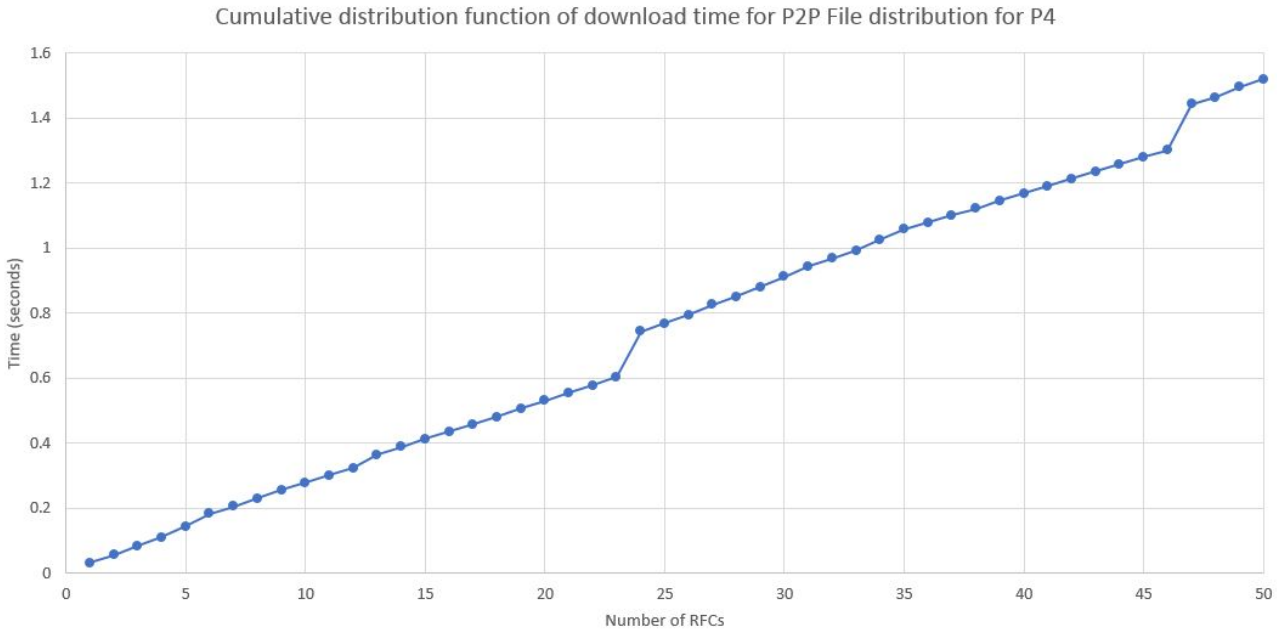

Figure 10: Peer 4

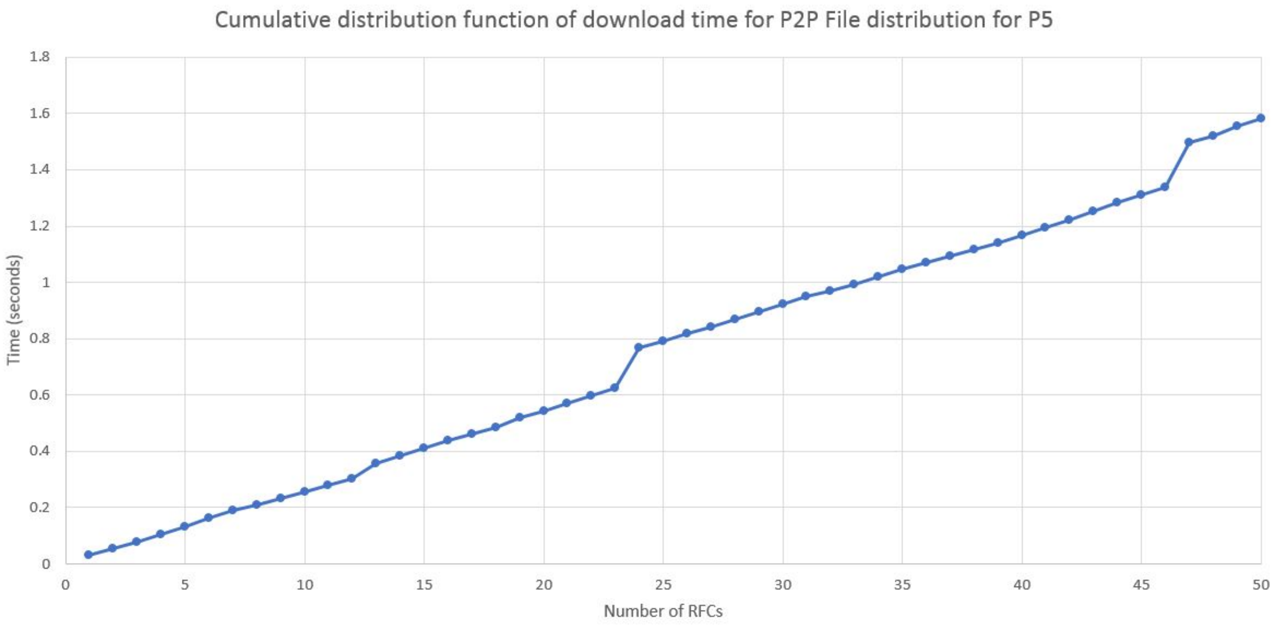

Figure 11: Peer 5

5 Format of Messages

- **Register message from peer**

  Register P0 P2P-DI/1.0
  Host: Registration Server
  RFC server port of ⟨Hostname of peer⟩ is ⟨RFC server port of peer⟩

- **Response to registration message from registration server**

  P2P-DI/1.0 200 Registration successful
  Connection: close
  Cookie number: ⟨Cookie number assigned to peer⟩

- **Request for getting active peer list from RS**

  GET Active peers P2P-DI/1.0
  Host: Registration server
  OS: Windows 10

- **Response to active peer list request**

  P2P-DI/1.0 200 Active peers list received successfully
  Hostname of active peer: ⟨Hostname⟩
  RFC server port: ⟨Server port of peer⟩

  The hostname and RFC server port of all the active peers would be displayed.

- **Request for RFC index**

  GET RFC-Index P2P-DI/1.0
  Host: ⟨Hostname of peer contacted⟩
  OS: Windows 10

- **Response to RFC index request**

  RFC Number: ⟨RFC Number⟩
  Host: ⟨Hostname of responding peer⟩
  P2P-DI/1.0 200 RFC-Index received successfully

- **Request for RFC file**

  GET RFC ⟨RFC number⟩ P2P-DI/1.0
  Host: ⟨Hostname of peer contacted⟩
  OS: Windows 10

- **Response to RFC file request**

  P2P-DI/1.0 200 RFC ⟨RFC number⟩ downloaded successfully

- **Keepalive message request**

  Keepalive P2P-DI/1.0
  Host: Registration server
  OS: Windows 10

- **Leave message request**

  Leave P2P-DI/1.0
  Host: Registration server
  OS: Windows 10

- **Response to leave message request**

  P2P-DI/1.0 200 Leave successful for host ⟨Hostname of peer⟩ at ⟨Date⟩ ⟨Time⟩

- **Message format on registration server on receiving registration message**

  Hello first time user. You are trying to connect to the registration server at 65423.
  Please wait while you are being registered
  Registering...
  Peer ⟨Hostname of peer⟩ registered successfully.

- **Message format on registration server on receiving request for active peer list**

  List of active peers getting forwarded to ⟨Hostname of peer⟩.

- **Message format on registration server on receiving keepalive request**

  KeepAlive message received. Updating TTL value for ⟨Hostname of peer⟩

- **Message format on registration server on receiving leave request**

  Leave request received from ⟨Hostname of peer⟩
  Hostname: ⟨Hostname of peer⟩
  Activation Flag: false

6 Results and Findings

As per the results in Task 1 and Task 2, we observed that the cumulative download times for both centralized and P2P distribution are very close as all clients are not downloading the files at the very same instant due to lack of multiple systems. The cumulative download times vary between 1.5 and 2.5 seconds for our test cases and these can be seen in the download time curves for both, centralized as well as P2P distribution. However, in an ideal scenario, we would expect the time taken for P2P model to download data to be lesser than centralized distribution model because the peers can act both as distributors as well as consumers of data. Also, the overall upload rate in the medium increases because of the ability of each peer to upload data. Even the load will be distributed across multiple peers thereby making the overall system more efficient as compared to centralized distribution since it involves handling of the entire load by a single system.

6.1 Best case scenario for P2P File Distribution

In this scenario, we observed low cumulative download time i.e approximately 1.5 seconds because the downloads were done from the peer one after the other. So, each peer was the sole owner of the medium while it was downloading the files from other peers.

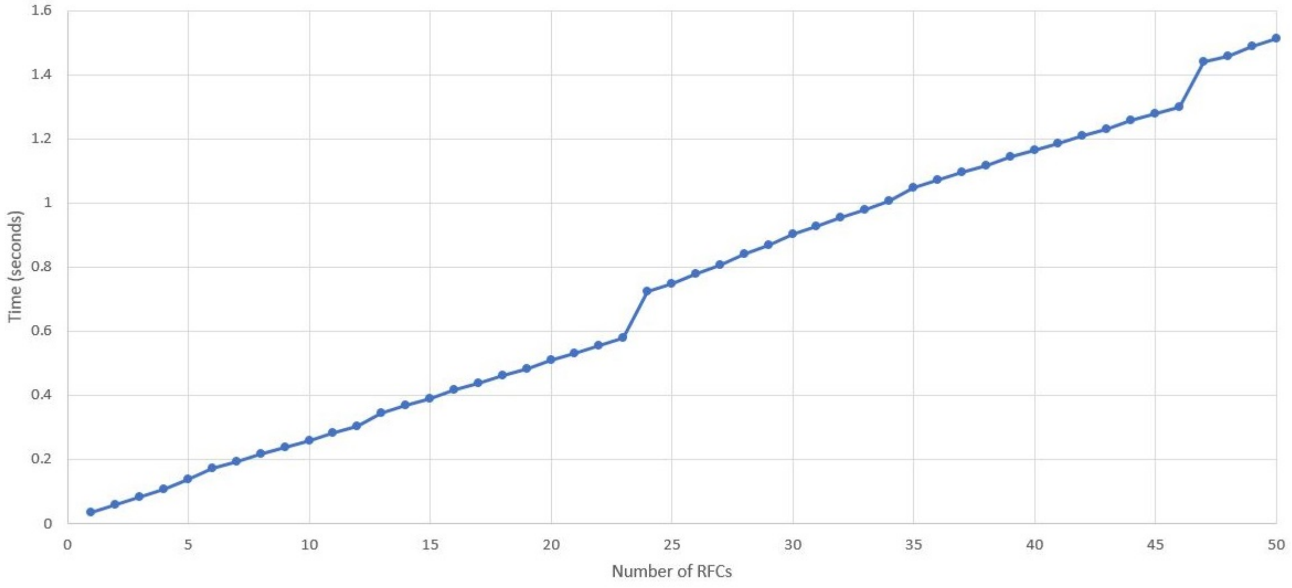

Figure 12: Best Case Scenario

6.2 Worst case scenario for P2P File Distribution

In this scenario, the execution of the peer programs were done almost concurrently, and hence there might be more load on a peer at a given instant as multiple peers might want to download the files from a peer during the same time frame. In our testing scenarios, we were able to observe a maximum cumulative download time of nearly 4.68 seconds. The download times are expected to increase with more number of peers wanting to access the medium during the same time frame.

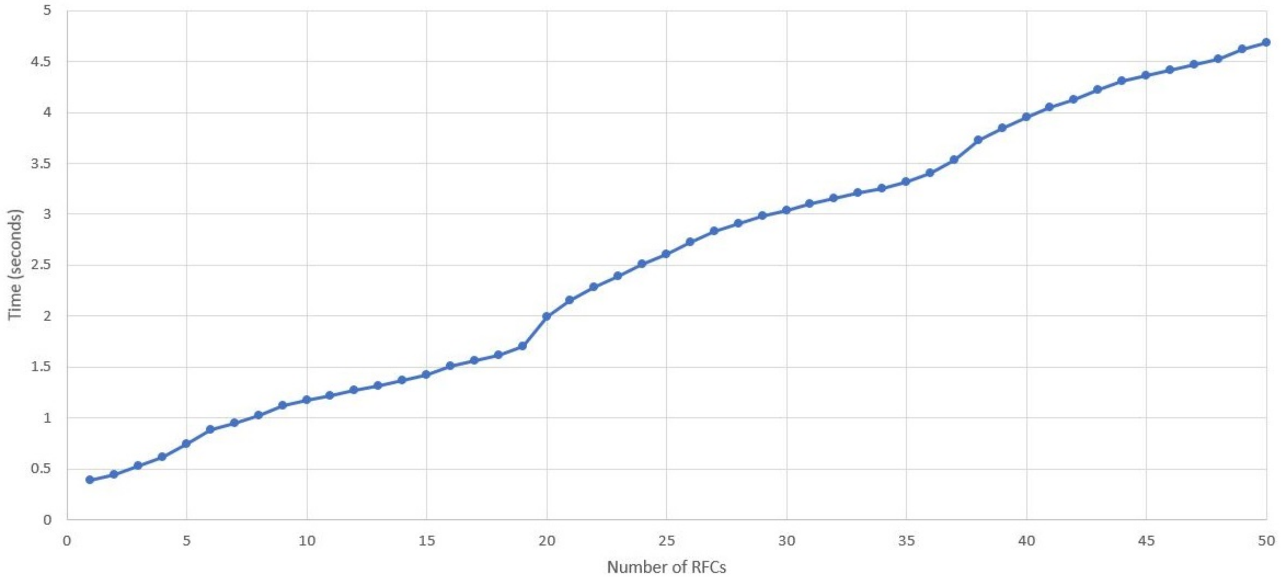

Figure 13: Worst Case Scenario

7 Conclusion

The primary difference between centralized and peer-to-peer file distribution is with respect to data management and control. In a centralized system, there is a dedicated server which provides services to multiple clients whereas in a P2P system, each peer contains data which it can distribute to other peers i.e. each peer can act both as a client and a server. In a large network, a P2P system is more scalable than centralized systems because the overall capacity of the system increases with an increase in the number of peers.

The distribution time in a centralized model depends on various factors including the size of the file, upload rate of server and the download rates of the peers whereas in a P2P model, the download time depends on how each peer distributes portions of the file to the other peers and the various factors which decide these are the upload rates of the server and the individual peers, file size, and lastly the number of peers. A simple implementation of how these factors actually affect the system has been done by us and helps strengthen our understanding of the two distribution models. The project even provides us with an opportunity to develop a well structured system and gain good programming experience.