CSC/ECE 573 INTERNET PROTOCOL – Fall 2019

PROJECT 1: PEER – TO -PEER SYSTEM WITH DISTRIBUTED INDEX

Team Members:

1. Ayush Bisht (200321513) - abisht@ncsu.edu
2. Dheeraj R Makam (200317241) - drmakam@ncsu.edu
3. Sharath Bangalore Ramesh Kumar (200322751) – sbangal2@ncsu.edu

INTRODUCTION

The recent years have seen a major shift in design of internet distributed system. The P2P (Peer-to-peer) model is used for diverse applications and services like content storage and sharing (file sharing, content distribution, back up – storage. P2P systems have received a great deal of attention in the recent years. Most current applications, however, use client-server technologies. A P2P network distributes information among the member nodes instead of concentrating it at a single server. This paradigm offers exciting advantages in information sharing. Some of the well know P2P Topologies are Centralized, Hierarchical, Decentralized and Hybrid.

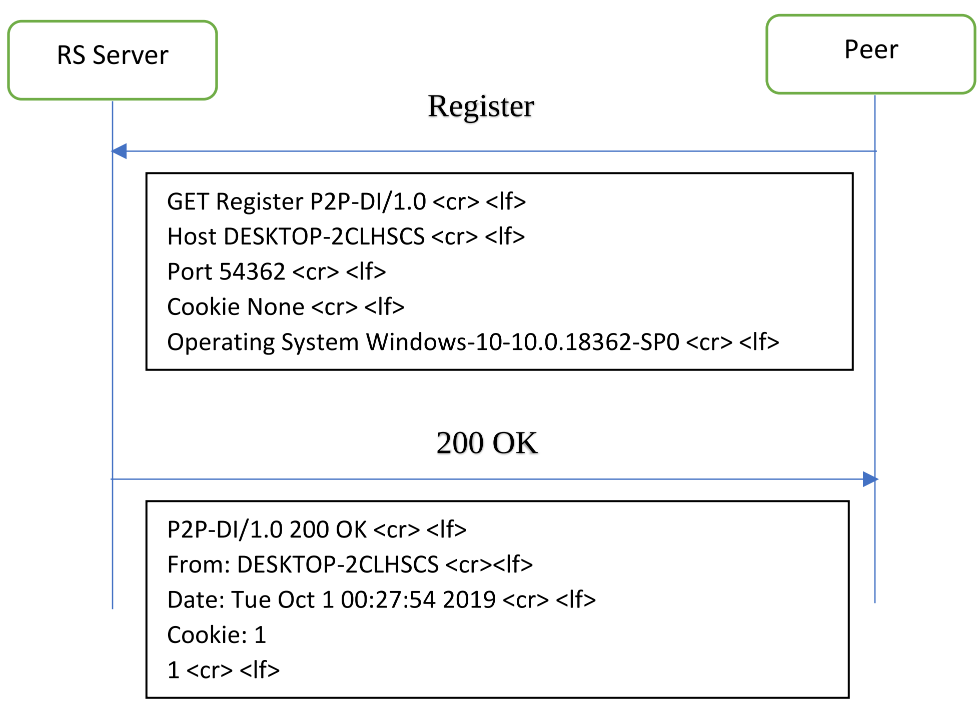
A client-server scenario like the web depends on a single server storing information and distributing it to clients in response to their requests. The information repository remains essentially static, centralized at the server, and subject only to updates by the provider. A P2P network, on the other hand, considers all nodes equal in their capacity for sharing information with other net-work members. Each user makes data repository available for distribution, which, combined with anyone's ability to join the network, leads to the fast growth of a network composed of distributed information repositories. In a P2P model, each member node can make data available for distribution and can establish direct connections with any other member node to download data. Instead of looking at what is available in a centralized repository, a client seeking information from a P2P network searches across scattered collections stored at numerous member nodes.

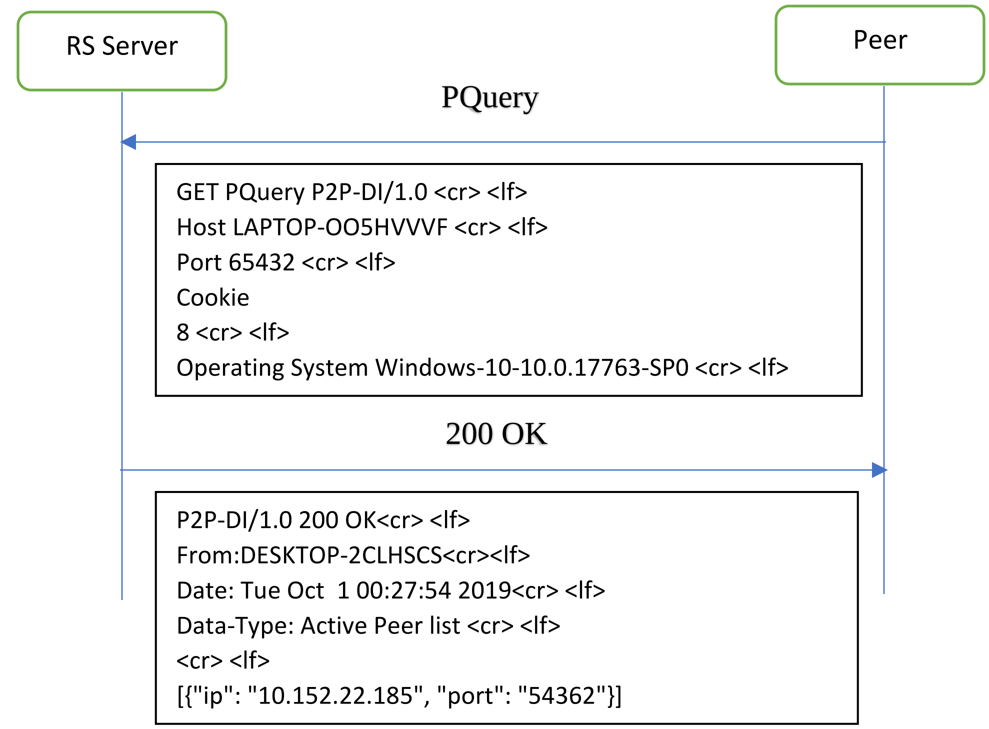
Peer-to-peer file sharing technology has evolved through several design stages from the early networks like Napster, which popularized the technology, to the later models like the BitTorrent protocol. The P2P model allows distributed system to scale up without any need of expensive infrastructure. Some of the area in which we can implement P2P are sharing of content (file sharing, content delivery), sharing of storage, sharing of CPU time.

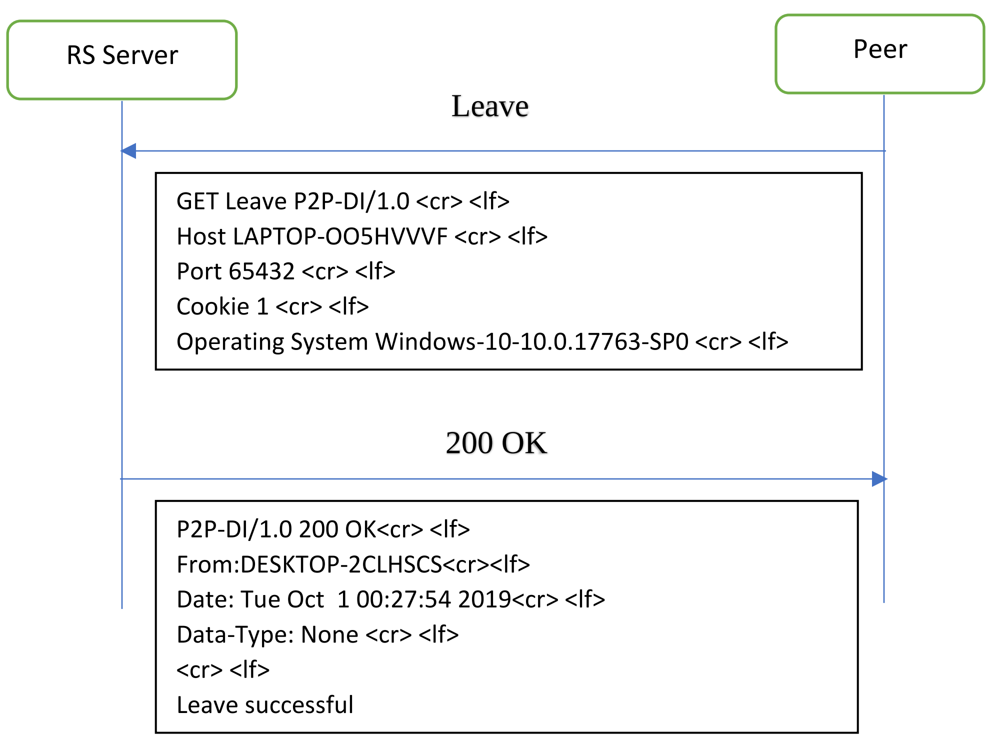
MESSAGE FLOW BETWEEN THE SERVERS

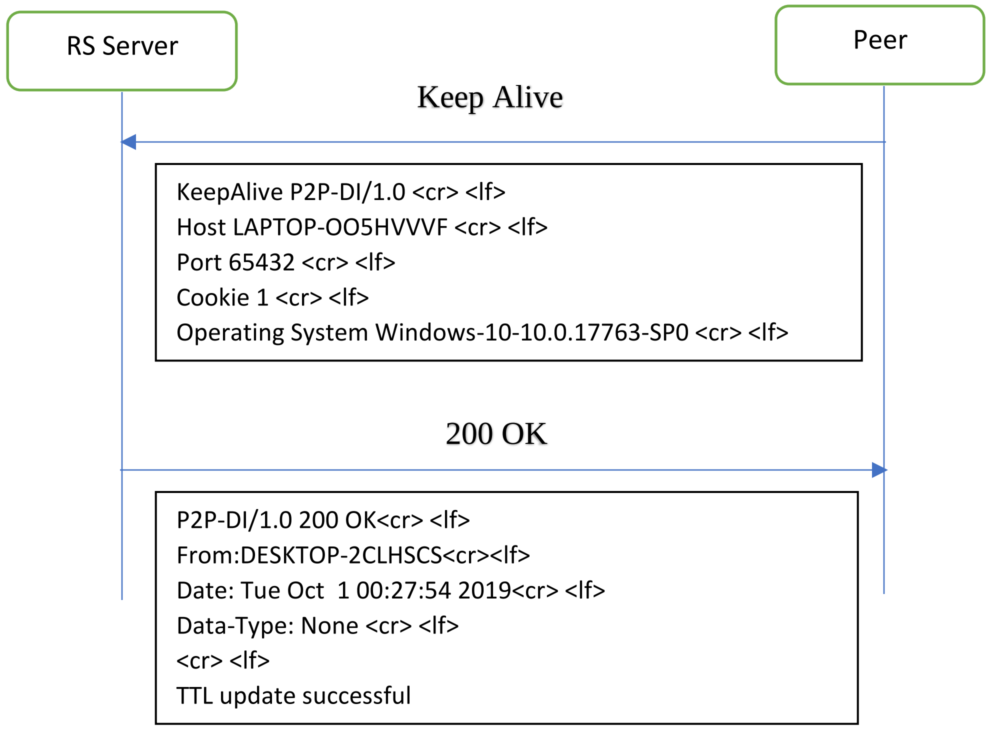
Message Flow between the RS Server (Registration Server) and the RFC Client

a. REGISTER: The register message is used to register the RFC Client / Server to the Registration Server.
b. PQuery: The PQuery message is used to request the active peer details from the RS servers to which the RFC client can contact to get the RFC Files.
c. KeepAlive : The Keepalive message is used to keep the peer registered to the RS. Once the RS receives this message from a particular peer, the RS server will reset the TTL time associated with Peer.
d. Leave: The Leave message is used by the RFC peer to indicate the RS that it wants to leave the system.

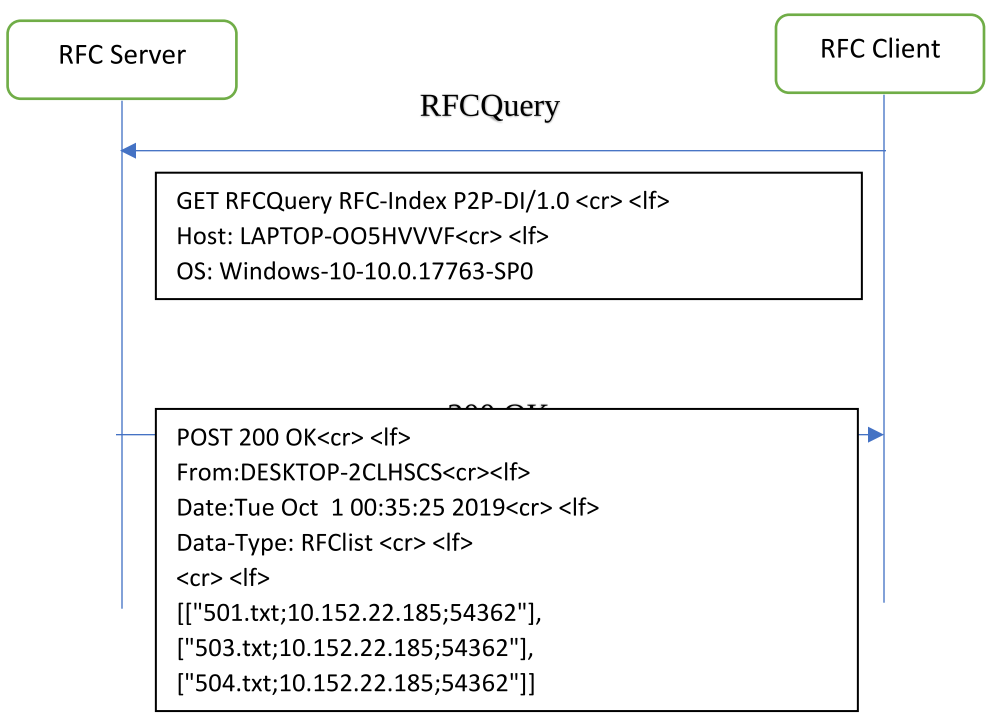


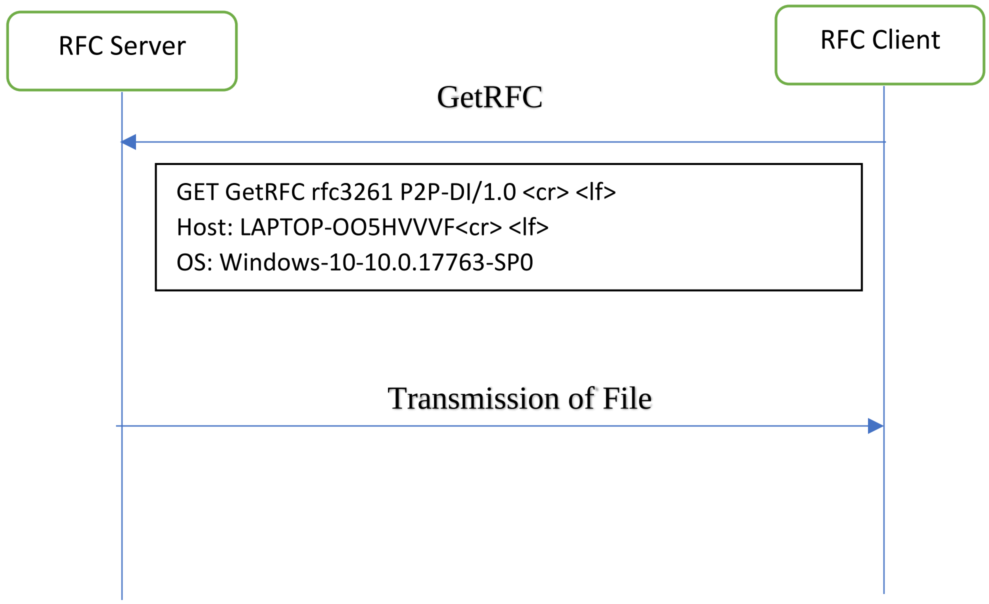

RS Server
Peer
PQuery
GET PQuery P2P-DI/1.0 <cr> <lf>
Host LAPTOP-OO5HVVVF <cr> <lf>
Port 65432 <cr> <lf>
Cookie
8 <cr> <lf>
Operating System Windows-10-10.0.17763-SP0 <cr> <lf>
200 OK
P2P-DI/1.0 200 OK<cr> <lf>
From:DESKTOP-2CLHSCS<cr><lf>
Date: Tue Oct 1 00:27:54 2019<cr> <lf>
Data-Type: Active Peer list <cr> <lf>
<cr> <lf>
[{"ip": "10.152.22.185", "port": "54362"}]


RS Server
Peer
Leave
GET Leave P2P-DI/1.0 <cr> <lf>
Host LAPTOP-OO5HVVVF <cr> <lf>
Port 65432 <cr> <lf>
Cookie 1 <cr> <lf>
Operating System Windows-10-10.0.17763-SP0 <cr> <lf>
200 OK
P2P-DI/1.0 200 OK<cr> <lf>
From:DESKTOP-2CLHSCS<cr><lf>
Date: Tue Oct 1 00:27:54 2019<cr> <lf>
Data-Type: None <cr> <lf>
<cr> <lf>
Leave successful



Message Flow between the RFC Client and the RFC Server





<u>Program Requirement and Message Flow between the nodes (A and B) in case of P2P DI system</u>

A- Peer Client
B- Peer Server

1. The RS server is expected to be listening on a standard port.
2. Peer A registers with the RS, provides the local port number for its RFC server, and receives a cookie.
3. The RS creates a new peer record for A and adds it to its peer index. If already register RS will update the TTL and return the cookie already assigned.
4. Peer A issues a PQuery message to the RS, and in response it receives a list of active peers. List of active peers will have IP and listening port number of the peer server.
5. If Peer B is in active list and B has the RFC that Peer A is looking for, then A issues an RFCQuery message to B, and in response it receives the RFC index that B maintains.
6. A merges B's RFC Index with its own index.
7. Since B is assumed to have the desired RFC, A will send GetRFC request to B and downloads the RFC text document.
8. Peer A will send Leave message once its task is done. The RS will update the records accordingly.

DOWNLOAD TIME GRAPH FOR TASK 1 (CENTRALIZED INDEX)

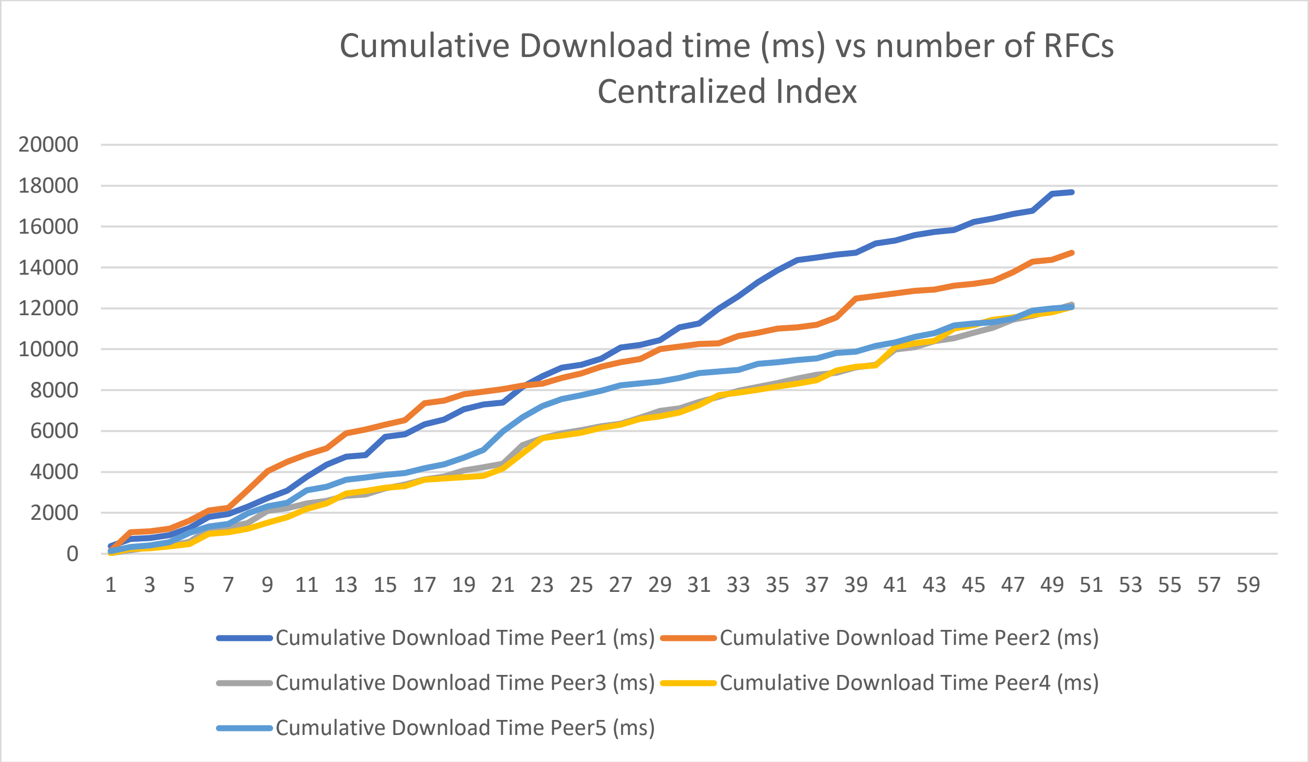
Procedure Followed:

1. Used 6 peers for testing the scenario. (P0 – P6)
2. P0 contains the files of 60 RFCs
3. P1, P2 …. P5 do not contain any RFC files.
4. First step all the six peers are registered to the RS server.
5. P1, P2, … P5 starts in loop to download 50 RFC files.

Plotting the cumulative download time against the number of RFCs for each peer P1, P2 …. P5, we get the graph as below:

x-axis = number of RFCs

y-axis = Cumulative time for download in milliseconds.



DOWNLOAD TIME GRAPH FOR TASK 1 (CENTRALIZED INDEX)

Number of RFCs	Peer 1 Individual RFC Download Time (ms)	Cumulative Download Time Peer1 (ms)	Peer 2 Individual RFC Download Time (ms)	Cumulative Download Time Peer2 (ms)
1	371	371	138	138
2	357	728	911	1049
3	47	775	49	1098
4	140	915	128	1226
5	340	1255	387	1613
6	540	1795	500	2113
7	149	1944	129	2242
8	353	2297	877	3119
9	428	2725	915	4034
10	363	3088	458	4492
11	665	3753	364	4856
12	598	4351	296	5152
13	396	4747	731	5883
14	80	4827	200	6083
15	896	5723	223	6306
16	111	5834	227	6533
17	496	6330	824	7357
18	232	6562	123	7480
19	496	7058	326	7806
20	233	7291	112	7918
21	103	7394	136	8054
22	770	8164	167	8221
23	510	8674	92	8313
24	422	9096	280	8593
25	143	9239	228	8821
26	291	9530	332	9153
27	554	10084	219	9372
28	128	10212	149	9521
29	233	10445	494	10015
30	633	11078	121	10136
31	186	11264	122	10258
32	718	11982	28	10286
33	610	12592	364	10650
34	687	13279	158	10808
35	576	13855	205	11013
36	509	14364	65	11078
37	119	14483	117	11195
38	137	14620	361	11556
39	96	14716	921	12477
40	452	15168	129	12606
41	148	15316	126	12732
42	269	15585	133	12865
43	160	15745	62	12927
44	87	15832	183	13110
45	398	16230	98	13208
46	162	16392	129	13337
47	231	16623	133	13770
48	152	16775	510	14280
49	827	17602	96	14376
50	79	17681	343	14719

DOWNLOAD TIME GRAPH FOR TASK 1 (CENTRALIZED INDEX)

Number of RFCs	Peer 3 Individual RFC Download Time (ms)	Cumulative Download Time Peer3 (ms)	Peer 4 Individual RFC Download Time(ms)	Cumulative Download Time Peer4 (ms)
1	31	31	30	30
2	140	171	199	229
3	135	306	32	261
4	80	386	105	366
5	198	584	101	467
6	612	1196	499	966
7	98	1294	84	1050
8	207	1501	164	1214
9	598	2099	310	1524
10	131	2230	266	1790
11	234	2464	400	2190
12	116	2580	270	2460
13	247	2827	489	2949
14	77	2904	124	3073
15	297	3201	150	3223
16	174	3375	80	3303
17	258	3633	309	3612
18	149	3782	62	3674
19	283	4065	71	3745
20	161	4226	60	3805
21	179	4405	358	4163
22	898	5303	733	4896
23	346	5649	760	5656
24	239	5888	123	5779
25	160	6048	137	5916
26	180	6228	240	6156
27	138	6366	155	6311
28	290	6656	282	6593
29	330	6986	126	6719
30	131	7117	195	6914
31	299	7416	349	7263
32	263	7679	493	7756
33	292	7971	125	7881
34	189	8160	138	8019
35	188	8348	150	8169
36	225	8573	150	8319
37	177	8750	174	8493
38	106	8856	470	8963
39	261	9117	185	9148
40	127	9244	68	9216
41	751	9995	904	10120
42	101	10096	163	10283
43	299	10395	129	10412
44	147	10542	597	11009
45	258	10800	178	11187
46	264	11064	263	11450
47	391	11455	102	11552
48	179	11634	130	11682
49	251	11885	124	11806
50	297	12182	259	12065

DOWNLOAD TIME GRAPH FOR TASK 1 (CENTRALIZED INDEX)

Number of RFCs	Peer 5 Individual RFC Download Time (ms)	Cumulative Download Time Peer5 (ms)
1	125	125
2	203	328
3	77	405
4	155	560
5	452	1012
6	312	1324
7	140	1464
8	515	1979
9	343	2322
10	172	2494
11	609	3103
12	171	3274
13	343	3617
14	109	3726
15	124	3850
16	93	3943
17	234	4177
18	197	4374
19	328	4702
20	374	5076
21	906	5982
22	696	6678
23	546	7224
24	334	7558
25	187	7745
26	231	7976
27	265	8241
28	93	8334
29	93	8427
30	177	8604
31	234	8838
32	77	8915
33	78	8993
34	293	9286
35	78	9364
36	109	9473
37	78	9551
38	262	9813
39	62	9875
40	296	10171
41	162	10333
42	277	10610
43	177	10787
44	373	11160
45	93	11253
46	77	11330
47	171	11501
48	378	11879
49	124	12003
50	62	12065

DOWNLOAD TIME GRAPH FOR TASK 2 (P2P DISTRIBUTED INDEX)

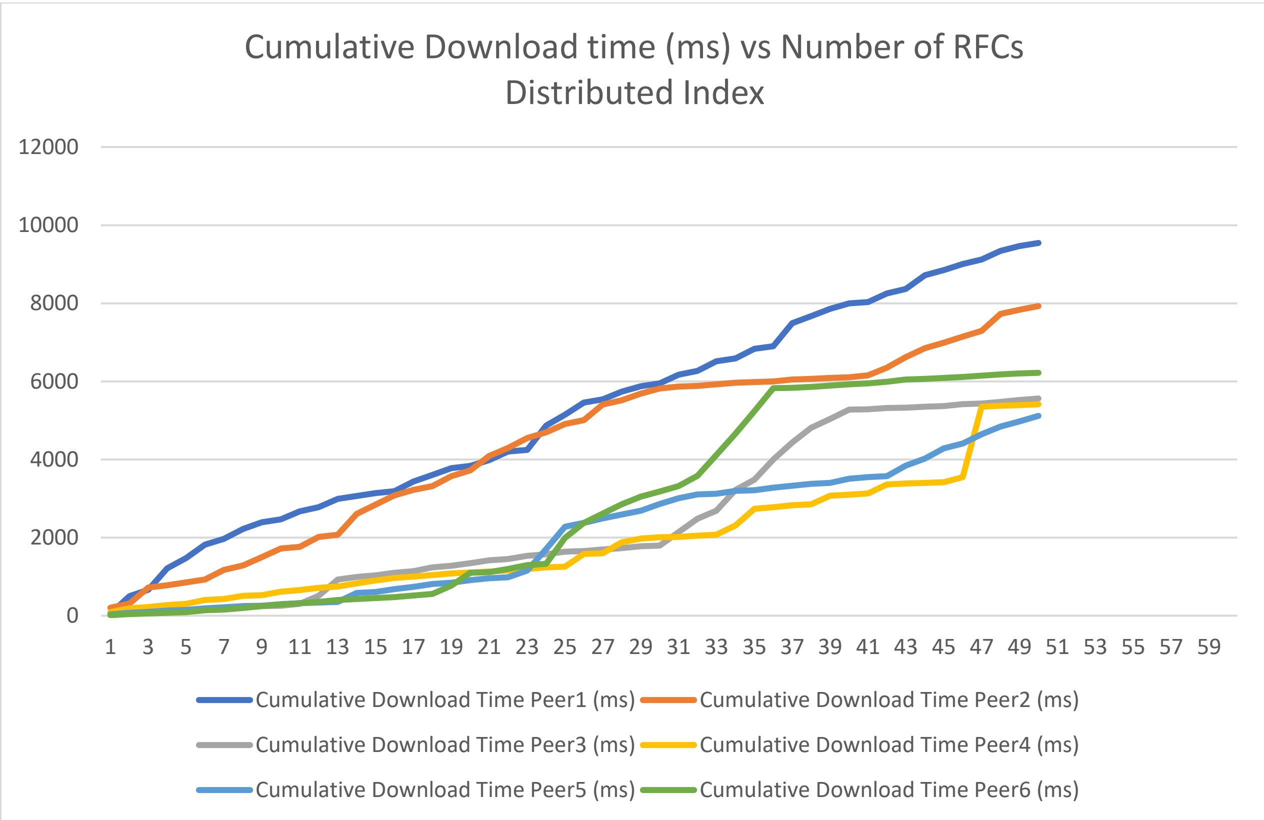
Procedure Followed:

1. Used 6 peers for testing the scenario. (P0 – P6)
2. P0 – P6 each peer will be initialized with 10 RFCSs
3. P0 – P6 peers will register to the RS Server
4. Each of the six peers starts a loop to download the 50 RFC files it does not have from the corresponding remote peer.

Plotting the cumulative download time against the number of RFCs for each peer P0, P1, P2 …. P5, we get the graph as below:

x-axis = number of RFCs

y-axis = Cumulative time for download in milliseconds.



DOWNLOAD TIME GRAPH FOR TASK 2 (P2P DISTRIBUTED INDEX)

Number of RFCs	Peer 1 Individual RFC Download Time (ms)	Cumulative Download Time Peer1 (ms)	Peer 2 Individual RFC Download Time (ms)	Cumulative Download Time Peer2 (ms)
1	50	50	204	204
2	454	504	111	315
3	159	663	410	725
4	547	1210	53	778
5	264	1474	78	856
6	348	1822	72	928
7	142	1964	246	1174
8	260	2224	113	1287
9	167	2391	212	1499
10	74	2465	221	1720
11	206	2671	45	1765
12	106	2777	248	2013
13	213	2990	63	2076
14	71	3061	527	2603
15	75	3136	237	2840
16	50	3186	245	3085
17	250	3436	137	3222
18	170	3606	101	3323
19	175	3781	254	3577
20	58	3839	140	3717
21	140	3979	372	4089
22	222	4201	204	4293
23	47	4248	258	4551
24	618	4866	141	4692
25	283	5149	213	4905
26	306	5455	105	5010
27	84	5539	398	5408
28	197	5736	104	5512
29	141	5877	171	5683
30	75	5952	138	5821
31	215	6167	46	5867
32	105	6272	15	5882
33	245	6517	46	5928
34	74	6591	41	5969
35	246	6837	15	5984
36	65	6902	15	5999
37	589	7491	46	6045
38	180	7671	15	6060
39	188	7859	31	6091
40	137	7996	15	6106
41	32	8028	46	6152
42	221	8249	202	6354
43	120	8369	265	6619
44	351	8720	234	6853
45	130	8850	140	6993
46	154	9004	156	7149
47	113	9117	140	7289
48	224	9341	437	7726
49	123	9464	109	7835
50	80	9544	93	7928

DOWNLOAD TIME GRAPH FOR TASK 2 (P2P DISTRIBUTED INDEX)

Number of RFCs	Peer 3 Individual RFC Download Time (ms)	Cumulative Download Time Peer3 (ms)	Peer 4 Individual RFC Download Time(ms)	Cumulative Download Time Peer4 (ms)
1	46	46	93	93
2	30	76	96	189
3	46	122	34	223
4	15	137	44	267
5	15	152	39	306
6	15	167	99	405
7	46	213	21	426
8	15	228	85	511
9	15	243	14	525
10	15	258	89	614
11	46	304	39	653
12	202	506	63	716
13	421	927	28	744
14	62	989	86	830
15	46	1035	73	903
16	62	1097	63	966
17	46	1143	38	1004
18	93	1236	33	1037
19	46	1282	42	1079
20	62	1344	27	1106
21	78	1422	29	1135
22	31	1453	22	1157
23	78	1531	43	1200
24	46	1577	38	1238
25	62	1639	13	1251
26	15	1654	329	1580
27	46	1700	15	1595
28	31	1731	280	1875
29	46	1777	104	1979
30	15	1792	26	2005
31	343	2135	13	2018
32	352	2487	27	2045
33	199	2686	26	2071
34	531	3217	237	2308
35	265	3482	426	2734
36	515	3997	41	2775
37	437	4434	53	2828
38	374	4808	20	2848
39	234	5042	222	3070
40	234	5276	24	3094
41	13	5289	33	3127
42	27	5316	234	3361
43	14	5330	27	3388
44	17	5347	13	3401
45	18	5365	18	3419
46	48	5413	126	3545
47	18	5431	1804	5349
48	40	5471	27	5376
49	53	5524	17	5393
50	37	5561	20	5413

DOWNLOAD TIME GRAPH FOR TASK 2 (P2P DISTRIBUTED INDEX)

Number of RFCs	Peer 5 Individual RFC Download Time (ms)	Cumulative Download Time Peer5 (ms)	Peer 6 Individual RFC Download Time (ms)	Cumulative Download Time Peer6 (ms)
1	35	35	13	13
2	39	74	27	40
3	23	97	14	54
4	38	135	17	71
5	13	148	18	89
6	38	186	48	137
7	30	216	18	155
8	27	243	40	195
9	15	258	53	248
10	36	294	37	285
11	23	317	38	323
12	21	338	33	356
13	16	354	42	398
14	226	580	27	425
15	30	610	29	454
16	71	681	22	476
17	55	736	43	519
18	73	809	38	557
19	39	848	213	770
20	63	911	329	1099
21	50	961	15	1114
22	24	985	80	1194
23	168	1153	104	1298
24	554	1707	26	1324
25	575	2282	670	1994
26	98	2380	384	2378
27	110	2490	240	2618
28	96	2586	232	2850
29	98	2684	195	3045
30	175	2859	137	3182
31	147	3006	139	3321
32	98	3104	263	3584
33	21	3125	526	4110
34	73	3198	554	4664
35	18	3216	575	5239
36	64	3280	584	5823
37	48	3328	13	5836
38	47	3375	27	5863
39	25	3400	26	5889
40	109	3509	37	5926
41	42	3551	26	5952
42	22	3573	41	5993
43	273	3846	53	6046
44	176	4022	20	6066
45	261	4283	22	6088
46	123	4406	24	6112
47	240	4646	33	6145
48	195	4841	34	6179
49	137	4978	27	6206
50	139	5117	13	6219

OBSERVATIONS

While implementing Task 1 i.e. centralized file distribution, only a single Peer, P0 contains all the 60 RFC files. Thus, entire P2P network there exists only a single centralized server, the RFC server of P0. The remaining peers only act as an RFC client and thus do not contribute to the distribution of files in the network. All the RFC request from the peers will be directed to the single server which leads to delay in processing the request and transfer of files.

On the other hand, in Task 2 i.e. P2P file distribution, all the peers in addition to acting as a client also act as a server, to provide other peers in the network with RFC files it has locally. Peers can request service from other peers and provide service in return to them. This increases the number of servers in the network providing the RFC files and therefore decrease the time required for distribution of the files.

The same observations can be made from the time curves for Task 1 and Task 2. For both the Tasks the cumulative time increases as the number of RFC files being downloaded increases. However, for timing diagram for Centralized file distribution, the total cumulative time required by a particular peer to acquire all the 50 RFCs is much larger when compared with the cumulative time in the case of P2P file distribution.

As the number of peers in the network will increase, the total amount of time required by a peer to download RFC files that it doesn't have will increase at a very high rate in the case of a centralised distribution system. Implementation of a P2P distribution will reduce the load on a single server as the incoming new peers will also bring new service capacity, along with new service demands.

In distributed Index peer to peer communication files are shared between peer directly without any dependency for a server (except for finding the IPs of peers), this setup brings in two scenarios;

1. All the peers are in the same network.
2. Peers are spread across multiple networks.

The best case scenario is when all of the peers are in the same network as the number of hops required to communicate with other peers will be minimal/negligible hence the time that packets spend waiting at routers will be minimal where as if the peers are in different networks it might happen that number of hops are maximum and so will be the delays at routers and links which are shared for heavy traffic.

REFERENCES

1. Data Sharing in P2P Systems - https://hal.archives-ouvertes.fr/hal-00379832/document
2. Wikipedia Peer to Peer - https://en.wikipedia.org/wiki/Peer-to-peer
3. Computer Networking: A Top-Down Approach, Book by Jim Kurose
4. Used Excel for tabulating the results and plotting the graph.