Parallelized FTP:- Effective approach for Solving Huge Download Delay Problem over Internet

Shaleeza Sohail and Sanjay Jha
School of Computer Science and Engineering
University of New South Wales, Sydney, Australia
Email: {sohails,sjha}@cse.unsw.edu.au
UNSW-CSE-TR-0411

2004

THE UNIVERSITY OF NEW SOUTH WALES

SCIENTIA

MANU ET MENTE

SYDNEY · AUSTRALIA

Abstract

The file download process over Internet is usually slow and unpredictable. We have designed and implemented a distributed and co-ordinated file transfer protocol for the Internet applications. We have designed and implemented a centralized server that distributes the download process across multiple file servers based on such QoS parameters as available bandwidth and delay. In addition to this, we monitor the FTP flows to detect slow servers and congested links and adjust the file distributions accordingly. Early experimentation suggests that our method can reduce the download time by more than 50% for large files. In addition to reducing the delay our technique has an added advantage that it does not need any modifications to the existing FTP implementations.

1 Introduction

Internet exhibits an inherent dynamic and unpredictable nature for file downloads. Large files require substantial amount of time to download. During that time the resource availability in the Internet can change drastically. This can make the download process unpredictable and usually very slow. Files are replicated on multiple mirror servers for load distribution. Clients usually selects the mirror server that is geographically closest to it for file download. This approach assumes that file download from such server will take minimum time and will produce least congestion on Internet. This assumption can be wrong as the geographically closest server can be highly congested. In that case file download may take large time and may increase congestion at the server. Hence, this selection criteria cannot be acceptable specially if other mirror servers have high a resource availability at that time.

A number of research efforts have addressed the problem of selecting the best server for a particular client on the basis of different metrics. Guyton and Schwartz has proposed a server selection technique [1] on the basis of hop counts and round trip delay. Carter and Crovella proposed the idea of dynamically selecting the best mirror server on the basis of available bandwidth and congestion along the path between server and client [2]. Fu and Venkatasubramanian took the complexity of the server selection technique one step further by introducing the server's availability in terms of available CPU cycles, I/O bandwidth and memory in addition to the characteristics of the path i.e; delay and available bandwidth [3].

Download Accelerator Plus [10] claims to select the most responsive mirror servers and to download the file simultaneously from those mirror servers. However, no technical information is available that can explain the mechanism used for the selection of mirror servers and the simultaneous download of file. Rodriguez and Biersack [5] introduced a dynamic technique to download data from multiple mirror servers in parallel using HTTP.

Byers et al [4] have presented the idea of downloading from multiple servers and peers simultaneously in peer to peer scenario but they failed to consider the server's availability and network QoS issues. Few peer to peer applications like Kazaa [8] and Furthurnet [9] support simultaneous download of a single file from multiple peers. We were unable to find any related technical information to estimate the similarities and the differences between those tools and our approach. All peer to peer applications allow direct download from the peers that have the copy of the desired file. The peer to peer approach tries to facilitate file sharing among users without a centralized server. Due to this common property of these approaches the file transfer process is random and unoptimized. Moreover, the approaches used in [4, 8, 9] do not support a central entity to monitor and dynamically optimize the download process with respect to a desired QoS parameter due to the distributed nature of peer to peer network.

P-FTP is an approach that optimizes the process of downloading a file using FTP by selecting multiple servers on the basis of server availability and path quality. Simultaneous download from multiple servers on the basis of available resources in the network and at the mirror server, optimizes the download process for least delay and better resource utilization. P-FTP proposes a central server, P-FTP server, that calculates the file portions to be downloaded from each mirror server. File portion size is based on the server resource availability. P-FTP server keeps a comprehensive database of mirror server resources for this purpose. P-FTP client monitors the flows to detect slow and congested servers. Amount of file portion downloaded from such servers is reduced to avoid long transfer delays. Additional servers are contacted to download the remaining file portions.

2 P-FTP

The aim of P-FTP approach is to reduce download time for very large files by using available resources without over utilizing. The approach proposes a central entity, P-FTP server for every Autonomous System (AS). P-FTP server maintains a database about numerous mirror servers and the files replicated on those servers. P-FTP client at user machine gets activated when a user wants to download a very large file. P-FTP client requests the P-FTP server of its AS. The request message contains the name of the file that client wants to download and the bandwidth available between the client and its Internet gateway. P-FTP client calculates the available bandwidth to its Internet gateway by sending multiple PING messages and calculating the bandwidth on the basis of average RTT. P-FTP server finds the information about the mirror servers that have copy of the requested file from the database. P-FTP server ranks these mirror servers on the basis of available information. A number of high ranked mirror servers are selected based on the available bandwidth between client and its Internet gateway. P-FTP server calculates the file portions that are to be downloaded from each selected mirror servers on the basis of their ranks. After that P-FTP server replies the client with the information about the mirror servers and the file portion to be downloaded from each of those mirror servers. P-FTP server also sends information about some additional mirror servers that have the copy of the requested file to the client. P-FTP client receives the reply from the P-FTP server and starts simultaneous P-FTP sessions with all mirror server. P-FTP client downloads disjoint portions of a file from each mirror server. After starting simultaneous P-FTP sessions with mirror servers, P-FTP client monitors the flows to detect if any set of mirror servers is sharing a congested link on their respective paths to the client. If that is the case then P-FTP client drops the connection to all the mirror servers that share a congested link except one. The client starts same number of new P-FTP sessions with the additional mirror servers. mentioned in the reply received from the P-FTP server.

The client monitors the flows to find any slow mirror server. In case any mirror server is relatively very slow then others then the file portion to be downloaded from the slow server is reduced. A new connection to an additional mirror server is established to download the rest of the file portion. The client assembles the file after the download of all file portions is completed.

2.1 P-FTP Features

P-FTP is an approach that proposes the use of available network and mirror server resources to download large files in an optimized way, so that the transfer is completed in minimum time without abusing the resources. The salient features of our approach are:

- There is no change required to the existing FTP servers softwares for P-FTP approach to work. Our approach is designed to perform on existing Internet infra structure.
- The use of available resources in the network and at mirror servers reduces the download time for large files and limits the transfer delays in a predictable fashion.
- The placement of a central P-FTP server in user's AS makes it possible for the server to predict the network characteristics along the path between the mirror servers and all the client in the same AS.
- P-FTP approach checks available bandwidth at the last hop i.e, between the client and its Internet gateway before starting the file transfer. This consideration is introduced to avoid the unnecessary download initiations in the situation when the client is attached to slow access link, such as dial-up connection to Internet. In which case starting multiple download sessions cannot reduce the download time as the limiting factor in the download process is the last hop link.
- Downloading parts of a file from different FTP servers simultaneously distributes the load on the mirror servers. This distribution of the load among servers improves performance for all users downloading files from those servers as no one server is being highly utilized.
- After starting the transfer of file portions from mirror servers, P-FTP client monitors the TCP flows. If multiple flows are sharing a common congested link on their path to the client, client terminates such flows to reduce congestion over Internet.

- P-FTP server measures network characteristics on the path between its AS and many mirror servers. The excessive traffic produced by the network measurement tool can produce congestion on the network. This traffic can be reduced by efficient sampling and prediction techniques designed for P-FTP database.

2.2 P-FTP Server

A central P-FTP server is placed in client's autonomous system that is accessible by all users of that AS. On receiving a request, P-FTP server gathers relevant information about mirror servers that contain a copy of the requested file. Mirror server information database is maintained by the P-FTP server. It uses that information to rank the mirror servers. The mirror server that has highest resource availability is ranked highest and vice versa. Depending upon the available bandwidth between client and its Internet gateway, a number of high ranked mirror servers are selected. P-FTP server selects the number of mirror servers by assuring that the aggregate P-FTP traffic does not produce congestion in client's AS. Ranking and selection processes are explained later. P-FTP server calculates the file portions to be downloaded from each selected mirror server. Highest ranked mirror server is allocated the largest file portion and vice versa. The information about mirror servers and the file portions is sent to the requesting client. Information about some additional mirror servers is also sent to client, use of this information is discussed later.

2.2.1 P-FTP Database

P-FTP database plays a vital role in ranking and selection process of the P-FTP server. The database has three main parts: network, utilization and file. The network information consists of QoS characteristics along the path between client's AS and different mirror servers. The utilization information consists of mirror server's memory and CPU information. This information can be controlled by the mirror server's administrator. A special client is placed at mirror servers that sends information about mirror server utilization to P-FTP server at a configured rate. File information consists of file replica map of mirror servers, that indicates which file is replicated at which server.

Numerous researchers are trying to design a tool that can accurately measure network parameters between arbitrary Internet end hosts while producing least burden on network. The basic traditional approach to measure latency in the Internet is with tools like Ping and Traceroute. The use of these tools is easy but the measurements cannot be highly accurate. Sting is a tool that uses TCP protocol to measure the network attributes [15]. IDMaps is designed as an underlying service to provides the distance information[16]. King is a tool that estimates the latency between arbitrary end hosts with the help of existing DNS infrastructure [17]. Eugene Ng et al proposes the measurement of transmission delay between the peers in peer-to-peer architecture with the help of coordinates-based mechanism [18].

Few issues related to P-FTP database, such as, its size, the method to collect information about mirror servers and the overhead of maintaining P-FTP database require more investigation. We are in the process of formulating a scalable and efficient mechanism for initialization and maintenance of P-FTP database.

2.2.2 Ranking process

Ranking process ranks the mirror servers on the basis of available information. The available information consists of multiple parameters, such as:

- Network Characteristics:- The QoS parameters along the path between user's AS and different mirror servers provide the base for the ranking process. The mirror servers that have more network resources along their path to the client's AS are ranked higher.

- Utilization: P-FTP server considers the present utilization level of the mirror servers. Highly utilized mirror servers will have lower ranks.

- Demand: P-FTP server tries to utilize all available resources (mirror servers) in the best possible manner. In order to do this the P-FTP server considers the demand of the mirror servers. By demand, we mean that in the situation when there are very few mirror servers to fulfill any particular file request, then those mirror servers have high demand. P-FTP server will use those mirror servers only in unavoidable circumstances.

- Optimization Policies: In addition to above mentioned parameters, any optimization policy can be added to ranking process.

The parameters on the basis of which mirror servers need to be ranked are called optimization variables. For every optimization variable the suitability of each mirror server is calculated and then combined rank is allocated. Suppose there are m mirror servers and the suitability on the basis of an optimization variable, OV is to be calculated. The method to calculate the suitability for mirror server k, S_k, depends upon the type of OV. The OV can be of two types, direct and inverse OV. Direct OV directly influences the suitability of mirror servers. The examples of direct OV are bandwidth along the path and mirror server's available resources. The suitability of mirror servers on the basis of the direct OV is calculated by the following equation:

$$S_k = \frac{OV_k}{\sum_{k=1}^{m} OV_k} \qquad (1)$$

Inverse OV inversely effects the suitability for mirror server. End-to-end delay along a path and mirror server utilization are the examples of inverse OV. In the case of inverse OV, the minimum OV value, OV_z is searched for every OV. S_k values are calculated by the following equations:

$$s_k = \frac{OV_z}{OV_k} \tag{2}$$

$$S_k = \frac{s_k}{\sum_{k=1}^{m} s_k} \tag{3}$$

P-FTP server calculates the suitability for all mirror servers on the basis of each optimization variable. The average of all suitability values is calculated. The mirror server with highest average suitability is ranked first and so on. Complete algorithm for ranking and selection process is given in [6].

2.2.3 Selection Process

P-FTP server selects the mirror servers according to their ranks, high ranked mirror servers are selected first. The selection limitation is the available bandwidth between the client and its Internet gateway. Mirror servers must be selected in a way that the aggregate P-FTP flow does not produce congestion at the link between the client and its gateway.

2.3 P-FTP Client

After receiving a reply from the P-FTP server, the client starts FTP connections with the mirror servers simultaneously. P-FTP client is a type of FTP client with additional capabilities of partial file transfer and flow monitoring. P-FTP client starts downloading disjoint file portions from the mirror servers. Client monitors the inter-packet arrival rate to detect the shared congestion. The client tries to find the set of mirror servers that are sharing a common congested link on their respective path to the client. In case of successful detection, the client terminates connections to all mirror servers of that set except one. The client starts the same number of new connections to additional mirror servers. After that client monitors the throughput of the flows in a small interval of time. Client calculates respective fraction of file portion for each mirror server, downloaded in monitored interval. Based on that if any of the mirror server appears very slow then its allocated file portion is reduced. Client starts a new connection to download remaining file potion. Shared congestion detection algorithm runs first and then slow server detection. The complete P-FTP application at P-FTP client can be step-wise explained by algorithm *P-FTP Client*. Suppose the client sends a request to P-FTP server to download file X. In case the client cant get reply from P-FTP server in three attempts, the client downloads the file with traditional FTP approach. The reply from P-FTP server contains $\{(a_i, f_i)\}$ such that $a_i \in Active$, the name-set of mirror servers and $f_i \in F$, the respective file portions to be downloaded from a_i. Along with this, the P-FTP server also sends a name-set of additional mirror servers, $Passive$.

Algorithm *P-FTP Client*

Initialize $Attempts = 0$
P-FTP-Client-Algo($Attempts$)[
Send request for X, increment $Attempts$
if Receive reply
  then Connect with $Active$
    Shared-Con-Algo($Active$)[
    Slow-Server-Algo($Active$)[
  else Timeout
    if $Attempts \leq 3$
      then P-FTP-Client-Algo($Attempts$)
      else Download X with FTP
    ]
Shared-Con-Algo($Active$)[
Monitor Inter packet arrival rate for all in $Active$
if Determine the set of congested servers, $Congested$, $Congested \subseteq Active$
  then $f' : Congested_k \rightarrow Selected_k$, $Selected_k \subset Congested$ and $|Selected_k| = k$
    $Active - Selected_k$
    if $|Passive| \leq k$
      then $Active \cup Passive$
      else $f'' : -Passive \rightarrow Selected'_k$, $|Selected'_k| = k$ and $Selected'_k \subset 2^{Passive}$
        $Active \cup Selected'_k$
    Readjust file fraction values for all in $Active$
    Shared-Con-Algo($Active$)
    ]
Slow-Server-Algo($Active$)[
do$\forall m \in Active$
Monitor the size of received data, D_m
Calculate $T_m = \frac{D_m}{F_m}$
enddo
Normally distribute T_m and find mean, T_{mean}
Compare all T_m to T_{mean}
if Any $T_m \ll T_{mean}$, $m \in Active$
  then if $Passive \neq \emptyset$
      $F_m = F_{Temp}$

$F_m = \frac{T_m}{T_{mean}} * F_{temp}$
Start one new connection to p, where $p \in Passive$ and $F_p = F_{Temp} - F_m$
else Readjust file fraction values
]

Shared congestion detection algorithm has two main functions, f' and f''. f' takes $Congested$, which is a set of servers that are detected for sharing a congested link. The output of f' is a set of mirror servers, $Selected_k$. $Selected_k$ contains all members of $Congested$ except the one mirror server that responded quickest to the client. After detecting congestion client terminates connection with all members of $Selected_k$. The second function f'' selects a set of mirror servers, $Selected'_k$ from $Passive$. The number of servers in this set is equal to the number of connections that client terminated. Client starts new P-FTP connections with all members of $Selected'_k$.

2.3.1 Shared Congestion Detection Ability

The main aim of P-FTP approach is to reduce download delays of very large files, by utilizing unused, available resources of the network and the mirror servers. P-FTP downloads disjoint portions of a file from multiple mirror servers simultaneously. The amount of data downloaded from each server depends upon each mirror server utilization and the QoS characteristics of the path between that mirror server and the requesting client. P-FTP server maintains information about network characteristics and the mirror server utilization and calculates the amount of data to be downloaded from each mirror server on the basis of that information. Network information is gathered by measurement tools like pathchar etc. Most network measurement tools predicts each path's characteristics independently. P-FTP assumes that the link between the client and its Internet gateway is the only common link along the paths between mirror servers and the client. It is possible that two or more mirror servers share some common links on their path to the requesting client, other than the last hop link. In P-FTP approach requesting client starts simultaneous FTP connections to multiple mirror servers. Any of these mirror servers may share common link on their paths to the client, if enough resources are available on that common link to support multiple FTP sessions then the approach works fine. However, it is possible that common link can only support one FTP session at that point of time, by starting multiple FTP sessions, P-FTP approach produces congestion on that link and the download time will increase due to the competition among P-FTP flows. there is one more possible scenario that P-FTP server calculates the file portions to be downloaded from each mirror server on the basis of the available resources in network and at mirror server. It means that P-FTP server contains measured network information that shows that the path between each mirror server and client can support individual FTP session to those mirror servers. In case some mirror servers are sharing a common link on their respective paths to the client then starting multiple FTP sessions with all such mirror servers by P-FTP client can produce congestion over that common link. P-FTP approach recognizes this problem and has integrated the ability of detecting shared congestion among multiple P-FTP sessions with P-FTP client.

The approach used detect shared congestion is proposed by Katabi and Blake [7]. Katabi and Blake uses packet inter-arrival rate to detect if the flows are sharing same bottleneck congested link. Entropy is the measure of randomness for any variable, entropy of the inter-arrival packet time indicates if the flows are traversing same congested link or not. The entropy of independent flows and aggregate flow is calculated, flows are combined together to make aggregate flow. If the aggregate flow packet inter-arrival time exhibits less entropy then that of independent flows then those flows share common congested link and vice versa.

We have devised an algorithm on the same approach and integrated it with P-FTP client. This ability enables P-FTP approach to recognise congestion over Internet and perform appropriate action to dissolve such congestion. P-FTP client monitors flows and after measuring packet inter-arrival time entropy, predicts the flows that share common bottleneck. All the measurements are taken at the application level.

3 Results

P-FTP approach was evaluated with simulations and did produce some promising results [6]. Extending the same work, a C implementation of P-FTP is completed to test the approach in real world on the Internet.

4 Experimental Test-Bed

Planet-Lab is selected to test P-FTP implementation due to its vast deployed infrastructure. Planet-Lab is an open, globally distributed platform for developing , deploying and accessing planetary-scale network services [12]. Numerous Planet-Lab nodes were accessed and the access links of those nodes were studied. After careful observation of the nodes and their access links, eight Planet-lab nodes were selected and configured as FTP mirror servers, to be used in P-FTP implementation tests. The names and locations of these Planet-lab mirror servers are shown in table 1.

4 of the Planet-Lab nodes configured as mirror servers are in USA, three in Europe and one in Asia. Multiple Asian nodes were contacted, most of them were connected with very slow links that is why more American and European nodes were selected for these experiments.

P-FTP client and P-FTP server was placed in UNSW.edu.au domain at valiant.unsw.edu.au. An 8-MB file is replicated on all mirror servers and downloaded with P-FTP and

Node Name	Country
planetlab1.arizona.gigapop.net	USA
planetlab1.bgu.ac.il	Israel
planetlab1.diku.dk	Denmark
planetlab1.mnlab.cti.depaul.edu	USA
planetlab01.ethz.ch	Switzerland
planetlab1.chin.internet2.planet-lab.org	USA
planetlab2.frankfurt.interxion.planet-lab.org	Germany
planetlab1.ls.fi .upm.es	Spain

Table 1: Planet Lab nodes and locations

FTP approaches. Multiple experiments are carried out to test the effectiveness of P-FTP approach in comparison with traditional FTP approach. Each test is repeated 20-40 times and average of those values is calculated after eliminating outages.

5 TEST 0

A set of experiments, TEST 0 was carried out to determine the effect of number of mirror servers involved in P-FTP session on the download time. The servers used for P-FTP download in these experiments were picked randomly and not on the basis of their throughput. The purpose of TEST 0 experiments was to study the effect of number of servers on P-FTP approach even when slow servers are involved in P-FTP sessions. TEST 0 experiments helped in fi nding out the range of number of servers which should be used to carry out more detailed P-FTP download tests.

P-FTP approach was used to download an 8-MB fi le while changing number of servers involved in each session. The average download time and average fraction of fi le downloaded from each server is plotted in fi g 1. Thick black line in fi g 1 shows the download time when fi le is downloaded from 1 to 8 servers, the reference y-axis values are at right hand side. The bars at the background depict the fraction of fi le downloaded from each server and reference y-axis values are at left hand side. Fig 1 shows that download time decreases rapidly when number of servers are increased from 1 to 4. However after 5 servers in P-FTP session the addition of more servers provide minimal and unnoticeable decrease in download time. In the light of these results, it is safe to state that increasing the number of servers in P-FTP session beyond a certain limit is unfeasible.

After taking the measurements of this test, we have studied different overheads involved in P-FTP approach. We noticed that there are multiple overheads involved in P-FTP approach that increase exponentially with increase in number

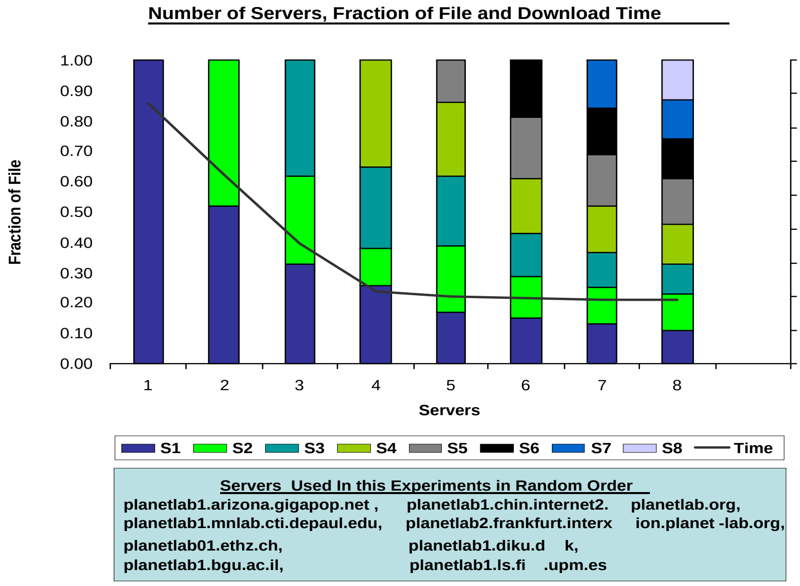

Figure 1: Number of Servers Vs Download Time. TEST0

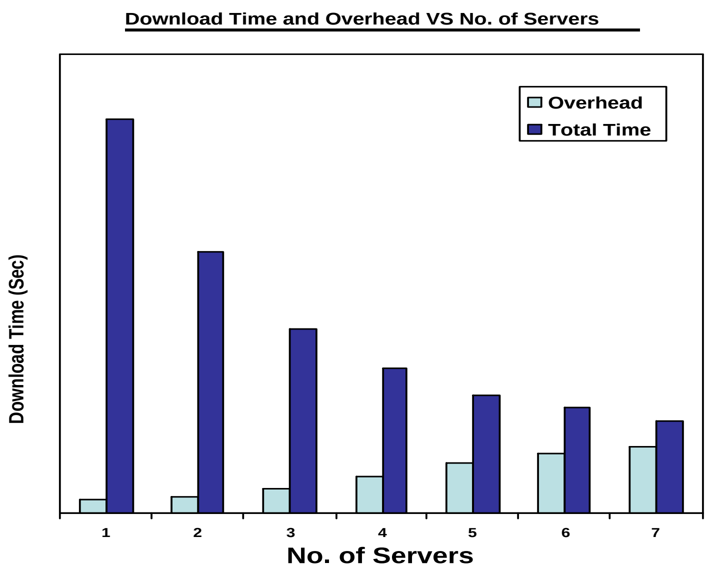

Figure 2: Download Time and Overhead(in terms of time) VS No. of Servers

of servers. This is hypothetically shown in fig 2. We are in the process of devising the method to find out the the factors that contribute to the overhead for P-FTP approach. The important part of that method would be to find out the relation between these factors and the overhead produced by them in terms of time.

6 TEST1 - TEST3

On the basis of the result of these experiments, the test bed of next set of experiments was configured. The number of servers used in next experiments were 3, 4 and 5 respectively. This range of number of servers is chosen to study the effect of P-FTP on file download time.

Tables 2, 4 and 6 show the time taken to download an 8-MB file with P-FTP using 3, 4 and 5 servers respectively. The tables also show the time taken to download the same file from each of those servers using traditional FTP approach. Minimum, maximum, average and standard deviation of the measured download times are shown in the tables. Tables 3, 5 and 7 show the minimum, maximum and average bottleneck bandwidth for every server and the file fractions downloaded from each server. The file fraction value for each server indicates the size of file portion downloaded from that server during P-FTP test. Path quality in terms of bottleneck bandwidth, RTT and dropping probability was considered at P-FTP server while calculating the file portions. One of the major influencing factor among those three parameters was bottleneck bandwidth, so values for that are shown in tables. The server availability was not considered during the file portion calculations due to the fact that the Planet-Lab infra-structure provides limited root access to the users. It was impossible for us to find CPU and memory usage of the mirror servers during tests. It is clear from the results that the download time is reduced considerably when P-FTP approach is used as compared to the FTP approach in which file is downloaded from each mirror server individually.

An interesting aspects of the results shown in tables 2, 4 and 6 is the difference in the standard deviation of the download transfer times. Download time measured with P-FTP approach has significantly less standard deviation as compared to that of download time for FTP from individual mirror servers. Small standard deviation of download time for P-FTP approach indicates that P-FTP transfers follow anticipated and predictable fashion. Hence, these measurement figures facilitate us to claim that the large files can be transferred with P-FTP in predictable time as compared to FTP.

6.1 TEST1

For this experiment 3 mirror servers are selected, two are in Europe and one in Asia. Fig 3 shows download time when 8 MB file is downloaded with P-FTP from those 3 servers and then with FTP from each server individually. The download

Test1:- P-FTP with Three Servers					
Downloaded From	Approach Used	Min (Sec)	Max (Sec)	Avg (Sec)	Std Dev
Asia-1	FTP	146	229	185	16
Europe-1	FTP	94	217	150	31
Europe-2	FTP	74	207	135	32
Asia-1,Europe-1,Europe-2	P-FTP	48	91	72	10

Table 2: Test 0 :- Download Time of 8MB fi le with FTP and P-FTP Approach

Test 1:- P-FTP with Three Servers						
Server	Bottleneck Bandwidth			File Fraction Downloaded		
	Min (Mbps)	Max (Mbps)	Avg (Mbps)	Min	Max	Avg
Asia-1	2	17	11.5	0.1	0.41	0.28
Europe-1	3	18	13.5	0.16	0.39	0.30
Europe-2	11	29	20.5	0.18	0.5	0.39

Table 3: Test 0:- Bottleneck Bandwidth and Downloaded File Fraction Values for P-FTP

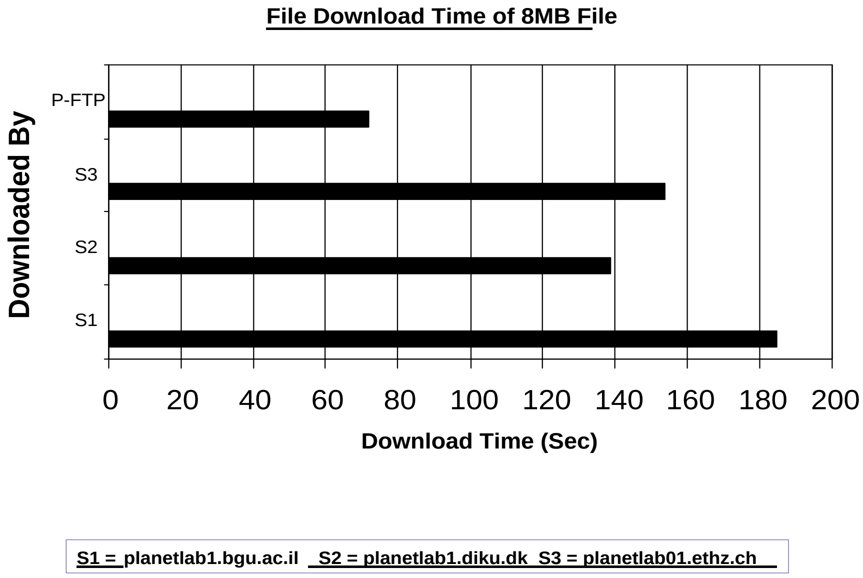

Figure 3: P-FTP download with 3 Servers, TEST1

time with P-FTP approach is less than half the time taken to download same file from the fastest server.

Table 2 shows the minimum, maximum, average and standard deviation of the file download time for this test for P-FTP and FTP approach. Fig 4 shows average available bandwidth and average RTT values of the three mirror servers used in TEST 1. Instantaneous available bandwidth values are plotted in Fig 5 and the fraction of file downloaded from each server is also plotted. The reference y-axis values for available bandwidth are at left side and reference file fraction values are at right side of Fig 5. The average, minimum and maximum of available bandwidth and the file fraction values are also shown in table 3. The server that has high available bandwidth values transfer large fraction of file in P-FTP session. The available bandwidth values for each server are different and in response to that P-FTP server calculates different file fraction values for each mirror server. Fig 5 and table 3 shows that the available bandwidth value for server 3 is mostly higher then the other two servers so most of the time larger portion of file is downloaded from this server as compared to the other two servers during P-FTP sessions.

6.2 TEST 2

A set of experiments labeled as TEST 2 are conducted while keeping 4 mirror servers in P-FTP session and comparing the download time of P-FTP file transfer with that of FTP transfer from each mirror server. Fig 6 shows the download time for 8 MB file with P-FTP and with FTP from each server, download time with

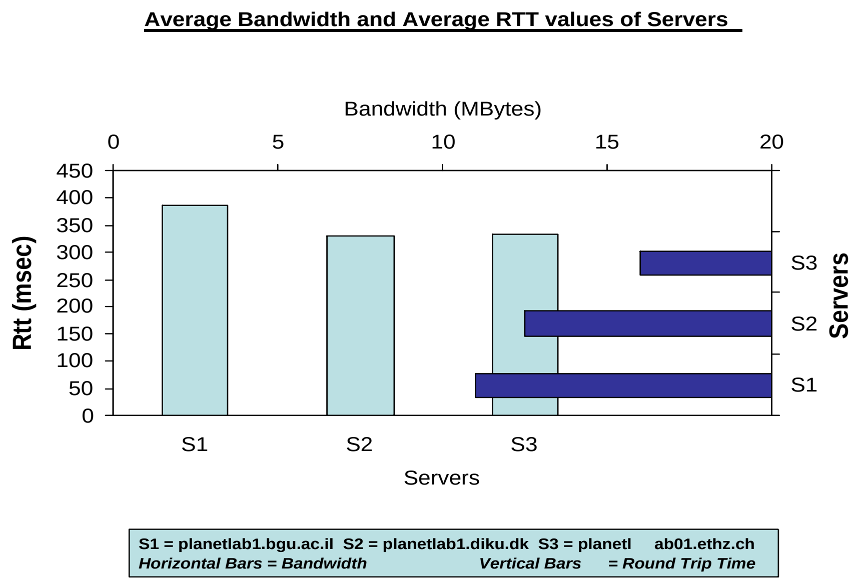

Figure 4: Average Bandwidth and RTT values of Servers involved in TEST1

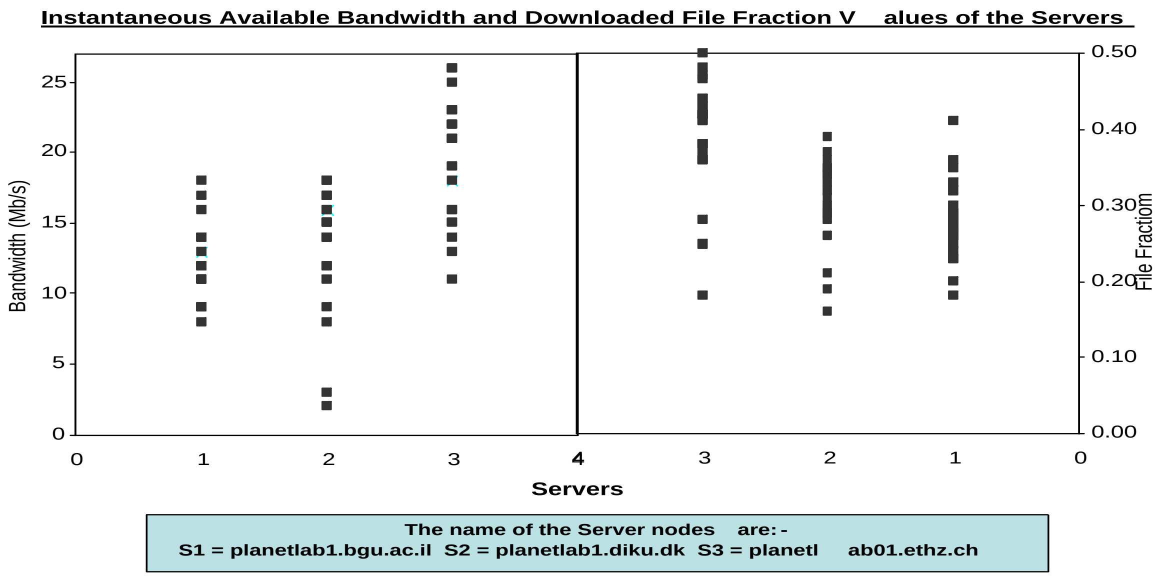

Figure 5: Instantaneous values of available bandwidth for Servers in TEST1

Test 2:- P-FTP with Four Servers					
Downloaded From	Approach Used	Min (Sec)	Max (Sec)	Avg (Sec)	Std Dev
USA-1	FTP	92	118	101	9
USA-3	FTP	90	117	109	8
Europe-3	FTP	109	180	145	24
USA-2	FTP	93	122	109	10
USA-1,USA-3 Europe-3,USA-2	P-FTP	37	55	43	4

Table 4: Test 2:- Download Time of 8MB file with FTP and P-FTP Approach

Test 2:- P-FTP with Four Servers						
Server	Bottleneck Bandwidth			File Fraction Downloaded		
	Min (Mbps)	Max (Mbps)	Avg (Mbps)	Min	Max	Avg
USA-1	3	22	13.4	0.2	0.35	0.25
USA-3	9	26	17	0.17	0.39	0.31
Europe-3	3	27	12.45	0.1	0.25	0.19
USA-2	9	20	13.8	0.2	0.3	0.255

Table 5: Test 2:-Bottleneck Bandwidth and Downloaded File Fraction Values for P-FTP

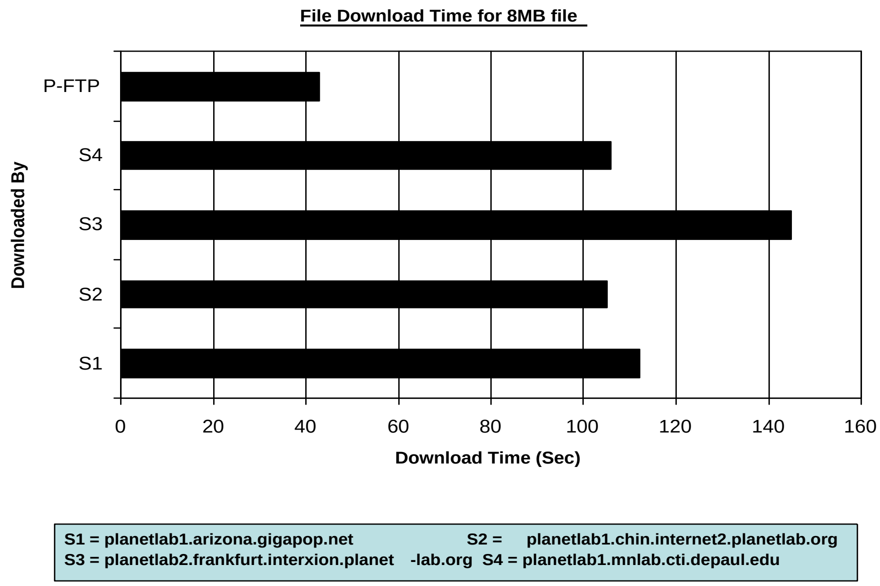

Figure 6: P-FTP download with 4 Servers, TEST2

P-FTP in this case is around one third of the download time of the fastest server. Fig 7 shows average RTT and average bandwidth value for each mirror server involve in this experiment. Fig 8 shows the instantaneous available bandwidth and file fraction values downloaded from each server in this experiment. The average, minimum and maximum of available bandwidth and the file fraction values are also shown in table 5. Fig 8 and table 5 shows that server 2 whose average available bandwidth value is higher then the server 3 transfers higher fraction of file to the client.

6.3 TEST 3

Fourth set of experiment, TEST 3 are conducted while keeping 5 mirror servers in each P-FTP session. The download time with P-FTP in this case is shown in Fig 9 in comparison with download time with FTP from each server. The download time with P-FTP in this case is not much different then the download time in TEST 2 results. The difference in file download time is so small that it does not justify adding another mirror server in P-FTP session. Fig 11 shows the available bandwidth along path to each server at different instances and as there are more servers and most of the time the available bandwidth value of these servers do not differ in large ratios so the file fraction values downloaded from each server does not fluctuate too much. The average, minimum and maximum of available bandwidth and the file fraction values are also shown in table 7. However it is noticeable that as the available bandwidth value of server 5 does not fluctuate too much so the file

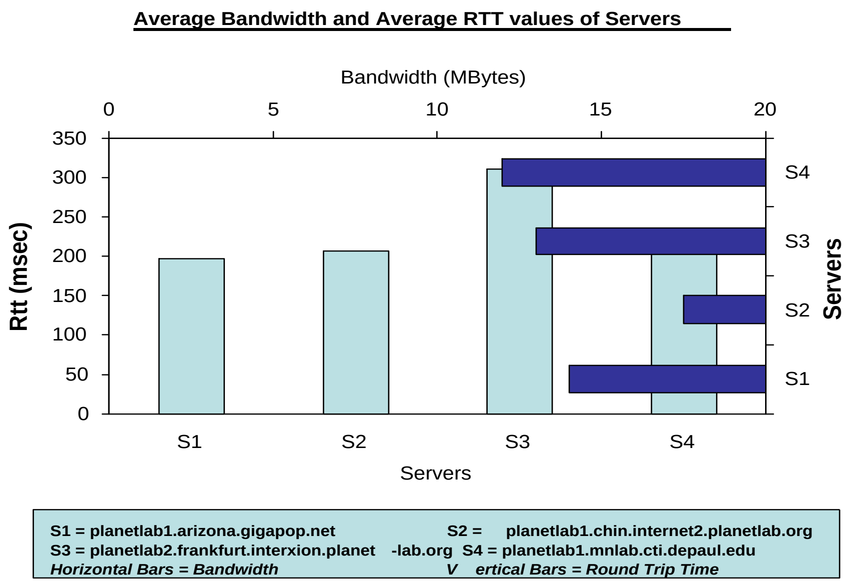

Figure 7: Average Bandwidth and RTT values of Servers involved in TEST2

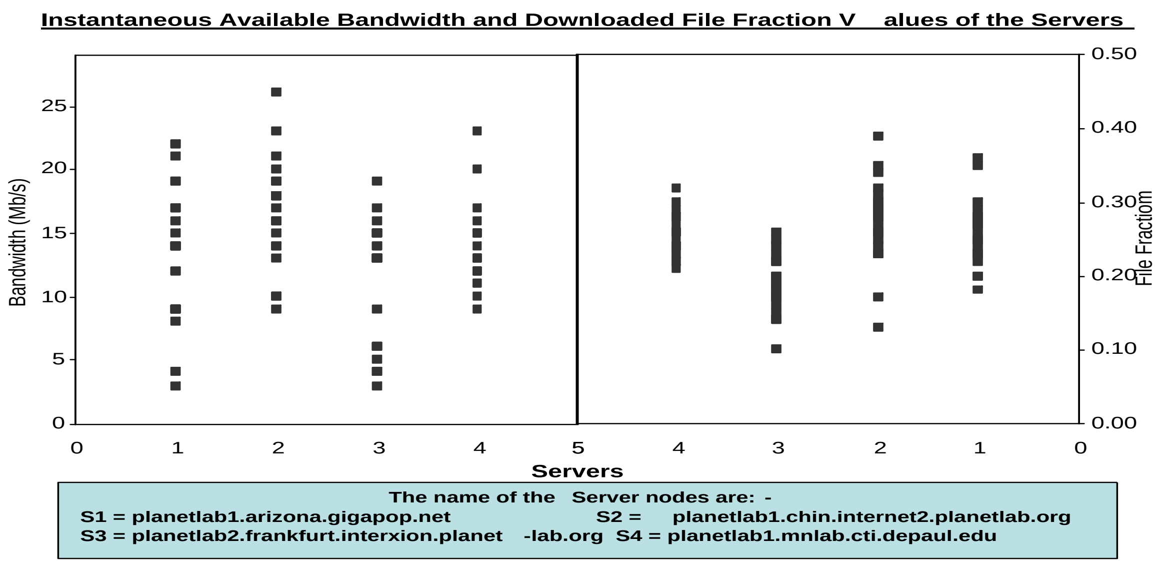

Figure 8: Instantaneous values of available bandwidth for Servers in TEST2

Test 3:- P-FTP with Five Servers					
Downloaded From	Approach Used	Min (Sec)	Max (Sec)	Avg (Sec)	Std Dev
USA-1	FTP	92	118	101	9
USA-3	FTP	90	117	109	8
USA-2	FTP	93	122	109	10
Europe-2	FTP	74	207	135	32
Europe-1	FTP	94	217	150	31
USA-1,USA-3,USA-2 Europe-2,Europe-1	P-FTP	34	44	40	3

Table 6: Test 3:- Download Time of 8MB file with FTP and P-FTP Approach

Test 3:- P-FTP with Five Servers						
Server	Bottleneck Bandwidth			File Fraction Downloaded		
	Min (Mbps)	Max (Mbps)	Avg (Mbps)	Min	Max	Avg
USA-1	3	22	13.4	0.1	0.31	0.133
USA-3	9	26	17	0.14	0.32	0.23
USA-2	9	20	13.8	0.1	0.28	0.145
Europe-2	11	29	20.5	0.2	0.4	0.25
Europe-1	3	18	13.5	0.1	0.25	0.144

Table 7: Test 3:- Bottleneck Bandwidth and Downloaded File Fraction Values for P-FTP

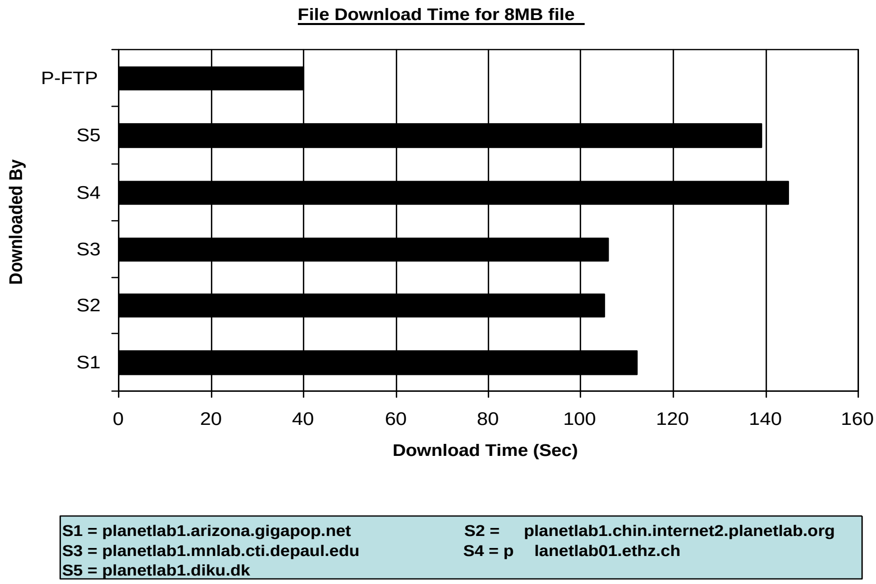

Figure 9: P-FTP download with 5 Servers, TEST3

fraction downloaded from this server also fluctuated in small range. In figure11 and table 7 another noticeable thing is that as server 2 has usually high available bandwidth value then server 3 so the file fraction downloaded from server 2 is higher then that of server 3 in most of the cases.

7 TEST 4

The essence of P-FTP approach is to use available resources between client AS and the mirror servers to improve download performance by starting simultaneous transfers of file portions. P-FTP approach anticipate that the link between the client and its Internet gateway is common for all transfers, so it tries to consider the available bandwidth on that link before starting P-FTP session. However, it is possible that different mirror servers share some common link with each other on their path to the client. If that is the case then when that common link is congested, by starting multiple P-FTP sessions will accelerate the congestion and degrade P-FTP performance. To avoid that , P-FTP approach integrates the client with the ability to detect the P-FTP sessions that share a common congested link. When client detects the P-FTP transfers sharing a congested link, it finishes all those P-FTP session except one, in order to relieve congestion on common link.

The approach used to detect shared congestion among P-FTP sessions is proposed by Katabi and Blake [7]. We implemented an algorithm by using the same approach and integrated it with P-FTP client.

P-FTP server calculates the file portion that should be downloaded from each

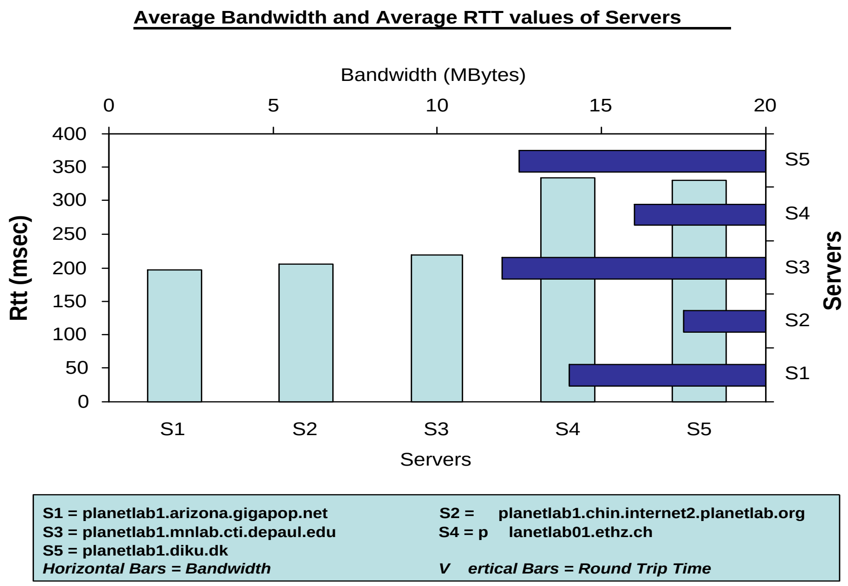

Figure 10: Average Bandwidth and RTT values of Servers involved in TEST3

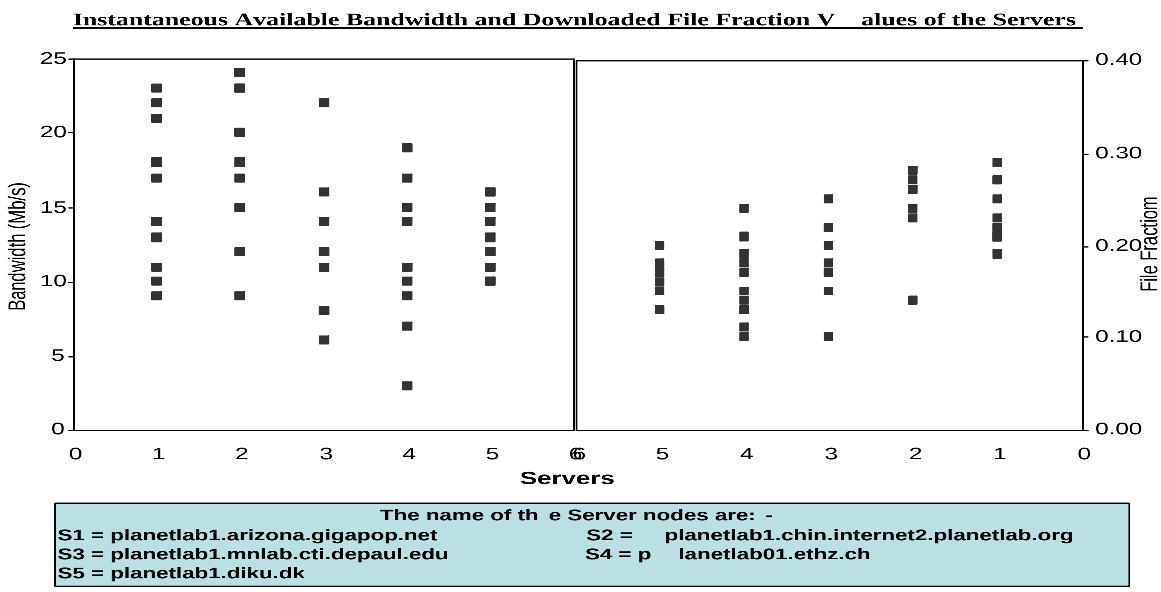

Figure 11: Instantaneous values of available bandwidth for Servers in TEST3

server and sends this information in the first reply to the client. P-FTP server also sends information about some additional mirror servers that have the replicated copy of the requested file. Client starts P-FTP session with the mirror servers indicated in first reply. However, if client detects shared congestion among any of these mirror servers then it terminates the connection to all mirror servers except one that shares congested link and starts new P-FTP session with the additional mirror servers. Among the P-FTP sessions that share a common congested link, client keeps the connection to the mirror server which started first as that servers appears to be the fastest.

In order to test the ability and performance of P-FTP client in this situation we ran few tests. We isolated the test environment from all outside factors to validate the P-FTP client shared congestion detection ability and tested the algorithm on isolated network. If the algorithm is to be tested on Internet then the dynamic and unpredictable behaviors of Internet may make it difficult to predict the efficiency of the algorithm.

7.1 Test-Bed

Few topologies are created on isolated LAN to test the P-FTP client. The link between client and its gateway (Link A in all topologies) is configured to only support 4 simultaneous P-FTP sessions. Seven FTP mirror servers are placed at the equal hop distance from the client and a 8MB file is replicated on all the mirror servers. All mirror servers are slowed down in order to study the effect of client's shared congestion detection ability on the download time.

7.2 Topologies

Four topologies are constructed, which are shown in figures 12, 13, 14 and 15. The topologies and working of the test are explained below:

1. In fig 12, two mirror servers 1 and 2 share the congested link. When the client starts simultaneous P-FTP session to both those servers then it detects the shared congestion and drops connection to one of the server and starts connection to mirror server 5 as Link A can support up to four connections.

2. In fig 13, two mirror servers 1 and 2 share a congested link and server 3 and 4 also share separate congested link. The client starts simultaneous P-FTP session to all these servers and detects the shared congestion and drops connection to server 2 and 4 and starts connection to mirror server 5 and 6.

3. In fig 14, three mirror servers 1, 2 and 3 share a congested link. When the client starts simultaneous P-FTP session to these three servers it detects the shared congestion and drops connection to server 2 and 3 and starts P-FTP session with mirror server 6 and 7.

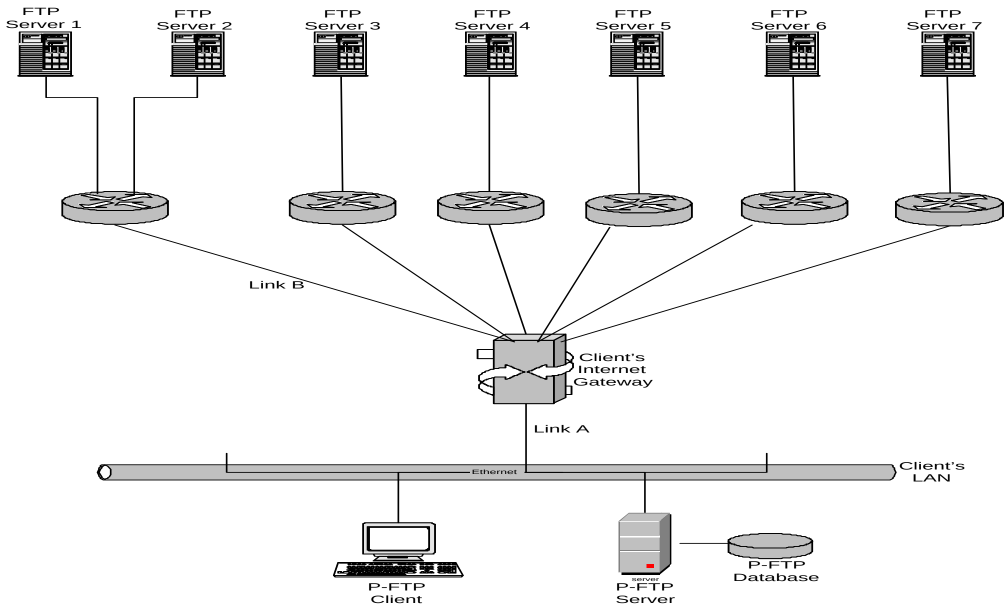

Figure 12: Topology 1

4. In fig 15, four mirror servers 1, 2, 3 and 4 share a congested link. When the client starts simultaneous P-FTP session to all those servers, it detects the shared congestion and drops connection to three of the servers and starts connection to mirror server 5, 6 and 7.

7.3 Results

The results of the tests are shown in fig 8. P-FTP client detects the mirror servers that share congestion. In table 8 the results show considerable improvement in download time when P-FTP client works with congestion detection ability. The second column of the table indicates the servers that client contacts in the beginning and column 5 shows the mirror servers that are contacted by the client after the shared congestion is detected. Column 6 shows the improvement in download time for P-FTP sessions when shared congestion detection algorithm works in comparison with the P-FTP session without shared congestion detection ability. The time taken by the servers to download file when there is no congestion on Link B and C is also indicated, as to compare the performance degradation in P-FTP when servers sharing a common congested path are selected for any session.

When more servers are sharing a common link, it takes more time to detect the shared congestion, the download time in column 6 and minimum number of packets in column 7 in table 8 confirms that. Column 7 of the result table indicates the minimum packets require to detect shared congestion, the packets referred here

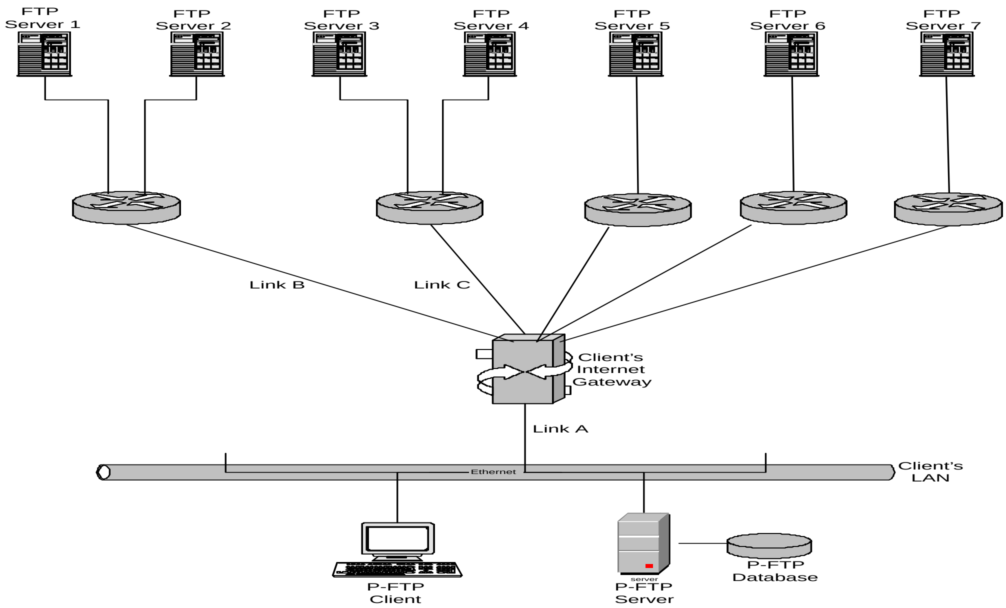

Figure 13: Topology 2

are the ones coming from the common congested link or in other words are the one coming from the servers sharing common congested link. The number of packets are indicated as the shared detection algorithm is based on measuring and analyzing inter packet arrival time of the packets of the flow that are traversing the common congested link. If there is more congestion on the link, it takes more time for the packets to reach the client and hence it takes longer for the P-FTP client to detect. Seventh column indicates a rough minimum number of packets require for the algorithm to detect shared congestion correctly. However, the implementation of shared detection algorithm for P-FTP client has some limitations. The percentage of error for every case is indicated in last column. The error rate is highest when there are more sets of servers sharing common congested link among their set as in topology 2. The reason being that the algorithm takes more inter packet arrival time measurements to detect which server is part of which shared congested set.

8 Conclusion and Future Work

P-FTP is an approach that uses available resources effectively to reduce transfer time for large fi les. The mechanism and entities involved in this approach were discussed briefly. The impact of P-FTP approach on download delay for large fi les was studied over Internet. Comparison with another similar technique on the basis of simulation study provides promising results.

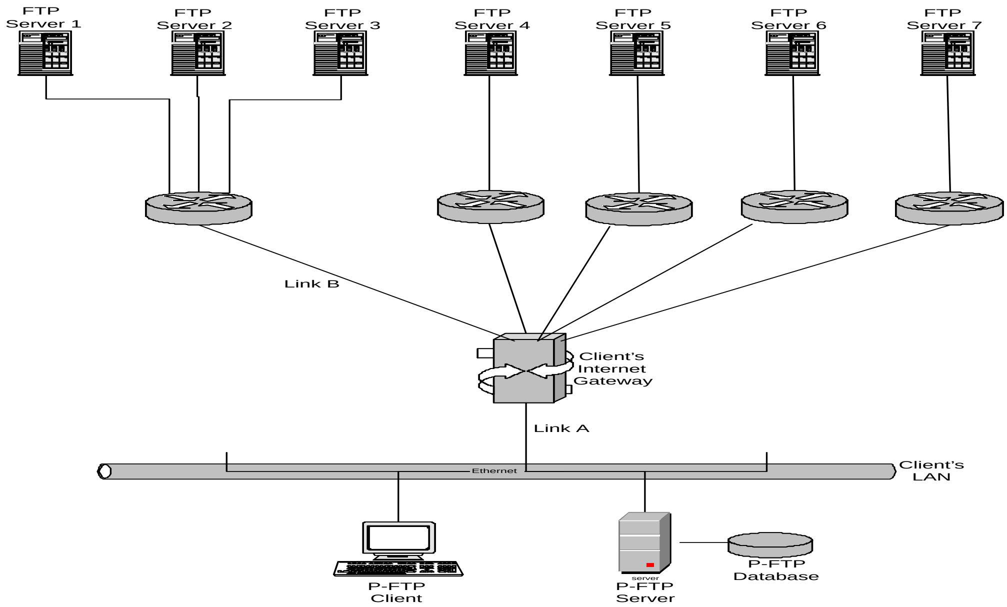

Figure 14: Topology 3

Topo-logy	Servers included in P-FTP at start	Conges-tion in Link B	Shared Conge-stion Detection	Servers added after Detection	Down-load Time (Sec)	Packets Needed	Error %
1	1,2,3,4	No	-	-	165	-	-
1	1,2,3,4	Yes	No	-	320	-	-
1	1,2,3,4	Yes	Yes	5	180	100	5
2	1,2,3,4	No	-	-	165	-	-
2	1,2,3,4	Yes,Link C too	No	-	300	-	-
2	1,2,3,4	Yes, Link C too	Yes	5,6	188	150	12
3	1,2,3,4	No	-	-	165	-	-
3	1,2,3,4	Yes	No	-	495	-	-
3	1,2,3,4	Yes	Yes	5,6	190	160	8
4	1,2,3,4	No	-	-	165	-	-
4	1,2,3,4	Yes	No	-	654	-	-
4	1,2,3,4	Yes	Yes	5,6,7	193	180	10

Table 8: Results For Shared Congestion Detection Algorithm

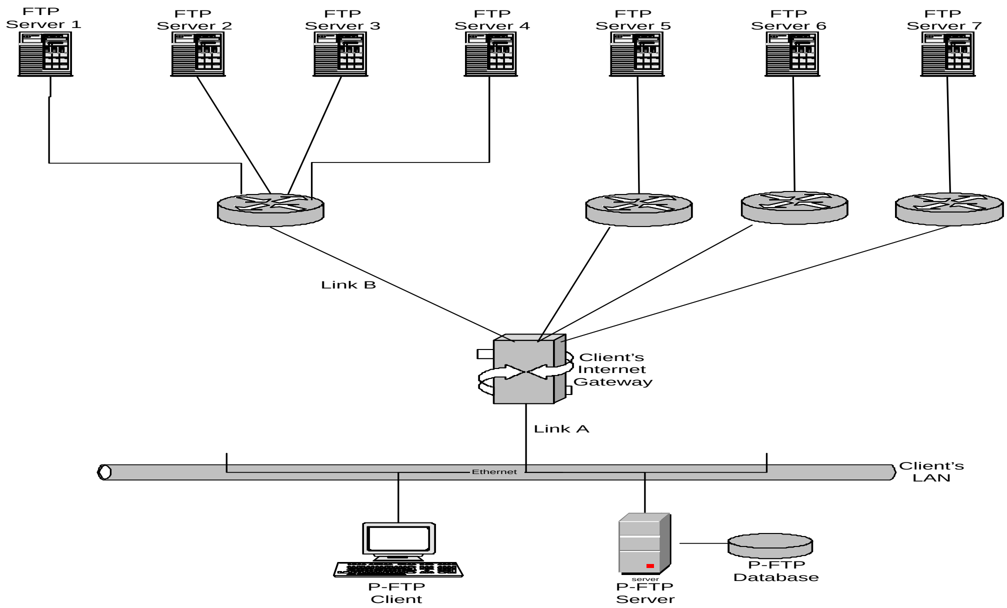

Figure 15: Topology 4

The overhead produced by the network measurement tool to collect information about network characteristics can be large, so we are studying different sampling techniques to reduce this overhead. Integration of P-FTP approach with a peer to peer application is being investigated as this merger can enhance effi ciency of P-FTP approach.

References

[1] J. Guyton and M. Schwartz, *Locating Nearby Copies of Replicated Internet Servers*, In Proceedings of SIG COMM 1995, pages 288-298.

[2] R. Carter and M. Crovella, *Dynamic Server Selection Using Bandwidth Probing in Wide Area Networks*, In Proceedings of INFO COM 1997.

[3] Z. Fu and N. Venkatasubramanian, *Directory Based Composite Routing and Scheduling Policies for Dynamic Multimedia Environment*, In Proceedings of IEEE International Parallel and Distributed Processing Symposium 2001 (IPDPS 2001).

[4] J. Byers et al, *Informed Content Delivery Across Adaptive Overlay Networks*, In Proceedings of ACM SIGCOMM Conference 2002, Pittsburgh, PA.

[5] P. Rodriguez and E. Biersack, *Dynamic Parallel Access to Replicated Content in the Internet*, IEEE/ACM Transaction on Networking, Vol 10, No 4, August 2002, pages 455-465.

[6] S. Sohail, S. Jha and H. Elgindy, *Parallelized FTP*, Technical Report UNSW CSE TR 0301, CSE, University of New South Wales, Sydney, March 2003.

[7] D. Katabi, I. Bazzi, and X. Yang,*A Passive Approach for Detecting Shared Bottlenecks*. IEEE International Conference on Computer Communications and Networks, ICCCN 2001, Arizona 2001.

[8] www.kazaa.com

[9] www.furthurnet.com

[10] www.speedbit.com/DAPDL.asp

[11] http://www.caida.org/tools/utilities/others/pathchar/

[12] http://www.planet-lab.org/

[13] http://www.isi.edu/nsnam/ns/

[14] http://www.cse.unsw.edu.au/s̃ohails/pftp-23jan.tar.gz

[15] S. Savage, *"Sting: a TCP-Based Network Measurement Tool"*, In Proceedings of the 1999 USENIX Symposium on Internet Technologies and Systems (USITS '99), October 1999.

[16] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang, *"IDMAPS: a Global Internet Host Distance Estimation Service"*, In Proceedings of IEEE/ACM Trans. on Networking, October 2001.

[17] K. Gummadi, S. Saroiu, S. Gribble, *"King Estimating Latency Between Arbitrary Internet End Hosts"*, In Proceedings of the SIGCOMM Internet Measurement Workshop (IMW 2002).

[18] E. Ng and H. Zhang, *"Predicting Internet Network Distance with Coordinates-based Approaches"*, In Proceedings of IEEE INFOCOM 2002, New York, NY, June 2002.