Parallelized File Transfer Protocol (P-FTP)

Shaleeza Sohail, Sanjay Jha and Hossam ElGindy
School of Computer Science and Engineering University of New South Wales, Sydney, Australia
E-mail: (sohails,sjha,elgindyh)@cse.unsw.edu.au

Abstract

Parallelized FTP (P-FTP) approach, attempts to solve the problem of slow downloads of large multimedia files while optimizing the utilization of mirror servers. The approach presented in this paper downloads a single file from multiple mirror servers simultaneously, where each mirror server transfers a portion of the file. P-FTP server calculates the optimum division of the file for efficient transfer. The dynamic monitoring ability of P-FTP maintains the file transfer process at the optimized level no matter how abruptly network and mirror server characteristics change.

1. Introduction

Downloading large multimedia files on the Internet is a very time-consuming, slow process. Provision of multiple mirror servers and efficient compression mechanisms are two of the approaches that try to address this problem. The most reliable and widely used protocol for file transfer is File Transfer Protocol (FTP). The files are usually replicated on multiple mirror servers and the client tries to download file with FTP, from the server which is geographically closest to it. Selection of geographically closest server is based on the assumptions that, firstly, the transfer will create least congestion on the Internet and secondly, the file is transferred in minimum possible time. However, it is possible that the closest server is the one which is most highly utilized and the links between that server and the client are highly congested so the optimized download process needs other criteria for selection of best mirror server. A number of research efforts have addressed this 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 for dynamically selecting the best mirror server on the basis of the available bandwidth and congestion along the path between server and client [2], however they have not considered server's utilization. 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]. All these approaches address the problem of selecting the best single server to download data. Download Accelerator Plus (DAP) [11] claims to select the most responsive mirror servers and download file simultaneously from those mirror servers. However no technical information is available that can explain the mechanism used for selection of mirror servers and simultaneous download of file. Buyers et al 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 [4]. Few peer to peer applications like Kazaa [9] and Furthurnet [10] support simultaneous download of a single file from multiple peers, however we are unable to find any related technical information by which we can 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] [9] [10] 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.

This paper proposes an approach to optimize 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 decreases the delay for the client and reduces the burden on any single server and path. The server availability is defined in terms of CPU utilization, available memory and I/O bandwidth of the server however, more attributes can be added. The path quality depends upon the availability and QoS provided along that path. Using these values the servers are

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)
0742-1303/03 $ 17.00 © 2003 IEEE

checked for suitability to download data. The amount of data downloaded from any server depends upon its suitability. This paper has two contributions: a) It propose suitability algorithm, which optimizes the file transfer process on the basis of different optimization policies, b) It validates our hypothesis through discrete event simulation using NS-2. P-FTP approach introduces a central entity, P-FTP server, that dynamically monitors the active P-FTP sessions and re-calculates the mirror server's suitability in case there is a change in path quality, mirror server utilization and availability.

The rest of the paper is organized as follows. Section 2 briefly explains P-FTP algorithm and functionality of its components. Section 3 describes the simulations that were performed to study the P-FTP algorithm. Section 4 identifies few future work options and concludes the paper.

2. Parallelized FTP (P-FTP)

In this section P-FTP approach is discussed briefly. The P-FTP service is to be implemented at application level. The three main entities in the P-FTP application are the requesting client, P-FTP server and the mirror servers. P-FTP server controls the overall functionality of the application and its placement is discussed later in the paper. Whenever a client wants to download a file, it sends a request message to P-FTP server. On receiving the request, P-FTP server collects information from its database about the mirror servers (that contain the copy of the requested file) and runs the suitability algorithm to calculate and evaluate the suitability of those mirror servers. The suitability algorithm is discussed in section 2.1. Respective suitability values indicate the portion of file that is to be downloaded from each mirror server. P-FTP server sends an inform message to all mirror servers, the message contains the suitability of that mirror server, the requested file and requesting client information. On receiving the inform message mirror server stores the information and sends confirm message to P-FTP server. P-FTP server sends the reply message to the requesting client, which contains mirror server's information and their respective suitability. Client initiates the FTP session with all mirror servers and mirror servers transfer the file portion to the client. From the reply message sent by P-FTP server, client knows the portions of file that each mirror server is transferring. The client uses that information to reassemble the file after the file transfer from each mirror server is completed. After the completion of transfer, the mirror servers send complete message to P-FTP server, which triggers the removal of that request from the active sessions. When P-FTP server receives database update, the suitability values for all active P-FTP sessions are recalculated and mirror servers are notified accordingly with reinform messages. The mirror servers update the file portion values for the related active and inactive P-FTP sessions according to the reinform message. A new reply is sent to the client to inform new suitability values. Fig 1 shows the interaction among the P-FTP entities. We propose use of a soft-state based approach for all state information in our protocol. Loss of control messages/state, concurrency and deadlock issues need to be considered for robust operation of this protocol. We intend to use Esterel language to model the protocol in future.

There are few issues related to the implementation of P-FTP approach on large scale. We are not assuming that introduction of P-FTP approach will reduce the file transfer time exponentially. There's not much point in requesting multiple simultaneous reads if all the data gets to an adjacent node quickly, then dribbles to the requesting node through a small pipe. For this reason, a client should only start P-FTP session if the bandwidth between the client and its gateway is more than some minimum threshold. In other words clients that have congestion in their own access network (LAN) will not benefit from using P-FTP service. Requesting fragments of files from multiple server may introduce some Operating Systems overhead. The standard system calls would be an open, a seek and then sequential reads. A standard open would transfer at least the first block of the file to memory, so that transfer would commonly be wasted, but this is not a significant overhead. It is likely that most transfers will be via NFS. As NFS is stateless, the system calls would be an open, a seek and then a set of reads. These are not likely to affect a server much.

2.1. Suitability algorithm

The core of the P-FTP application is the suitability algorithm. The main function of the suitability algorithm is to calculate the suitability of mirror servers on the basis of the optimization policy. After receiving the request from the client, P-FTP server runs the suitability algorithm and optimum suitability of the mirror servers is calculated. The suitability of each mirror server indicates the portion of requested file that should be transferred from that server. However it is possible that due to the utilization of any mirror server at the time of request, the transfer of calculated file portion from that mirror server is not possible. In that case the algorithm runs recursively till the optimized suitability of all mirror servers is calculated while keeping the utilization of mirror servers under maximum threshold value.

The complete P-FTP application at P-FTP server can be step-wise explained by algorithm *P-FTP*. Suppose the client requests for file X, P-FTP server finds the resources required to download that file, FR and the set of mirror servers, M that have replicated copy of that file.

$$M \subset MS \tag{1}$$

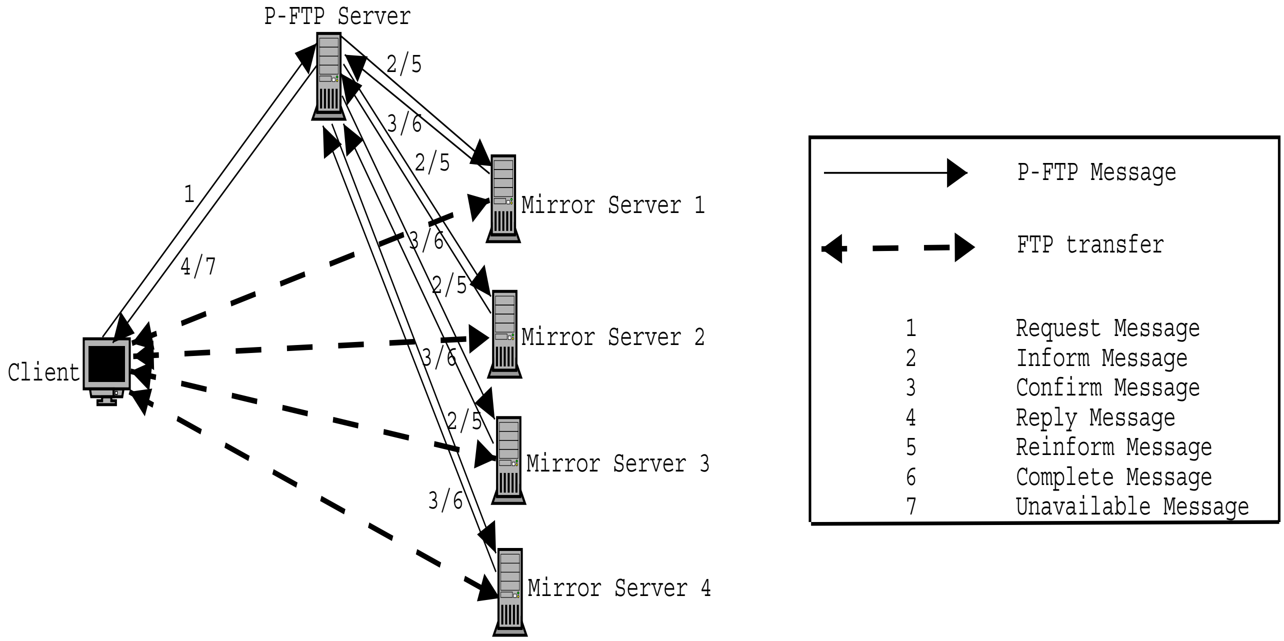

Figure 1. Interaction among P-FTP entities

Where MS is a set of mirror servers, that are registered with P-FTP server. The suitability algorithm finds the suitability $S_m \; \forall \; m$, where $m \in M$, on the basis of the optimization policy. The suitability of all mirror servers is checked against their available resources, AR. If check fails for any mirror server, the suitability of that mirror server is reduced so that its utilization remains less than the maximum threshold value, that mirror server is then replaced from M to M_{Final} set and suitability values for remaining mirror servers in M are recalculated.

$$M_{Final} \subset MS \quad (2)$$

$$M_{Final} \subseteq M \quad (3)$$

Where M_{Final} is the set of mirror servers to whom P-FTP server will send inform message after the suitability algorithm finishes.

Algorithm *P-FTP*

Initialize M_{Final}
Find FR of X
Find M for X
PFTP-algo(M) [
if $M_{Final} = \emptyset$
  then if $M \neq \emptyset$
    then Calculate S_m , $\forall \; m$ using optimization policy
    Calculate $A_m \leftarrow S_m * FR, \forall \; m$
    if $AR_m > A_m$, $\forall \; m$
      then Add all m to M_{Temp}
      Delete all m from M
      Add All m_{Temp} to M_{Final}
      else $S_k \leftarrow AR_k/FR, \forall \; k$ s.t. $k \in [1..m]$
      $FR \leftarrow (FR - (S_k * AR_k))$
      Add k to M_{Temp}
      Delete k from M
    PFTP-algo(M)
  else Send resources unavailable message to client
else Send Inform message to m_{Final}
  if confirm message received from all m_{Final}
    then Send reply message to Client
    Delete all m_{Final} from M_{Final}
    else Copy all m_{Final} from M_{Final} to M
    Delete all m_{Final} from M_{Final}
    Delete not responding m from M
    PFTP-algo(M)
]

2.2. Optimization policy

The optimization policy defines one or more characteristics of mirror server on the basis of which their suitability is calculated, that characteristic of the mirror server is called the Optimization Variable (OV) and is represented as V. The algorithm for the calculation of suitability tries to optimize the process of file download with respect to OV. The optimization policies can be categorized on the basis of number of OV. The two categories of optimization policies for P-FTP are briefly discussed here.

2.2.1. Single variable optimization policy. Single variable optimization policy defines one optimization variable. If there are m mirror server that has the replicated copy of requested file then OV values, V_k for all m are collected from the database. The method to calculate the suitability of mirror servers depends upon the type of OV. The OV can be of two types, direct OV and inverse OV. Direct OV values directly influence the suitability of mirror servers. The example of direct OVs are bandwidth along the path and mirror server's available resources. The suitability, S_k of mirror servers depending upon the direct OV values is calculated by the following equation:

$$S_k = \frac{V_k}{\sum_{k=1}^{m} V_k} \quad (4)$$

Inverse OV values inversely effect the suitability of 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, V_z is searched and S_k values are calculated by the following equations:

$$s_k = \frac{V_z}{V_k} \quad (5)$$

$$S_k = \frac{s_k}{\sum_{k=1}^{m} s_k} \quad (6)$$

Two of the most obvious and important single variable optimization policies are lowest delay policy and best utilization policy. In Lowest Delay Policy (LDP) end-to-end delay between mirror servers and the requesting client is the optimization variable. LDP tries to download file in minimum possible time. The optimization variable in Best Utilization Policy (BUP) is the utilization of mirror servers. Due to the highly unreliable and unpredictable nature of Internet, it is possible that QoS is not the primary focus for the parallelized FTP transfers, in that case BUP is considered which tries to keep the utilization of all mirror servers at minimum.

2.2.2. Multiple variable optimization policy. The importance of different characteristics of mirror servers can vary with respect to time of day or network characteristics. Multiple variable optimization policy can cater for these requirements. One example of such policy is delay and utilization policy.

Delay and Utilization Policy (DUP) is a double variable optimization policy which has optimization variables of end-to-end delay and the server utilization. The suitability of mirror servers is firstly calculated using LDP and BUP. The final suitability, S_F is calculated as:

$$S_F = S_{LDP} * a + S_{BUP} * b \quad (7)$$

$$a + b = 1 \quad (8)$$

S_{LDP} and S_{BUP} are calculated suitability values using LDP and BUP policies respectively. a and b are parameters whose values depend upon the relative importance given to each policy.

2.3. P-FTP Server Monitoring

The mirror servers send alive messages to P-FTP server frequently for monitoring purpose. If the P-FTP server does not receive three consecutive alive messages from any mirror server, that mirror server is considered to be down. The down mirror server is not included in any new suitability calculations, however if that mirror server is part of any active P-FTP session then P-FTP server recalculates the suitability for that active session without considering down server and send reinform messages to all other mirror servers. If all mirror servers participating in an active P-FTP session are down, a reply message is sent to the requesting client to inform the failure of the request. The P-FTP server needs two consecutive alive messages from a down mirror server to change its status to up.

The P-FTP server has to monitor the mirror servers as well as the active requests, that is why each request contributes significantly to the P-FTP server processing burden. Before accepting a new request, the P-FTP server considers its own utilization. In case P-FTP server finds its own utilization near maximum threshold value, it responds the requesting client with unavailable message. The unavailable message indicates that the P-FTP server cannot provide service to the client at that time, however, P-FTP server sends optimization variable values for the related mirror servers with the unavailable message to facilitate the client in choosing best server to download the file using traditional FTP approach. The self monitoring ability of the P-FTP server prevents the creation of processing bottleneck at the P-FTP server, which can degrade the overall network performance.

2.4. Database

P-FTP server keeps a comprehensive database about mirror servers and network. The mirror server information consists of load conditions and the replica map of stored files. The network information includes values for different network parameters. The mirror servers register themselves with P-FTP server at the time of initialization. These servers provide information to P-FTP server about their own utilization and replicated files. The soft state information

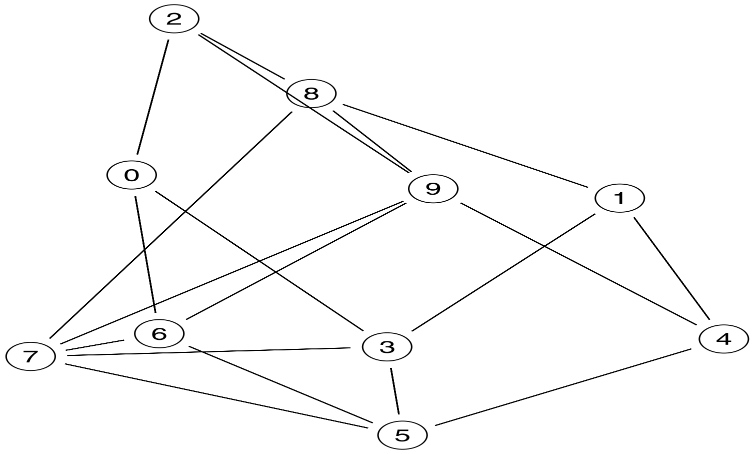

Figure 2. Topology

about mirror servers is updated by the mirror servers. The network information which needs update consists of QoS parameters along multiple paths between networks and the utilization of network links.

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 [5]. IDMaps is designed as an underlying service to provides the distance information[6]. King is a tool that estimates the latency between arbitrary end hosts with the help of existing DNS infrastructure [7]. 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 [8].

The P-FTP approach may use any of the above mentioned approaches to accurately calculate network parameters. Due to large number of clients on the Internet, it is impractical to measure network parameters from every client to every mirror server. To keep the P-FTP implementation scalable, few monitoring agents calculate the QoS parameters between their own networks and the mirror servers (or their networks). The monitoring agents provide the measured QoS values to the P-FTP database. The QoS values calculated this way do not indicate the accurate properties along different paths between two hosts, however they do provide comparative values for different QoS parameters along different paths between client's network and mirror servers. Another very important factor for efficient working of P-FTP approach is the frequency at which these network measurements are taken. By setting the frequency value too low the efficiency of P-FTP algorithm can be reduced considerably and setting the value unnecessarily high can increase network load exponentially. Due to the dynamic nature of Internet there exists a direct relation between the accuracy of these measurements to the frequency at which the measurements are made. However, the direct relation between the overhead produced by such measurements to the measurement accuracy greatly effects measurement frequency. The nature and size of the network topology greatly influences these two relations. To optimize P-FTP benefits more investigation is required in order to fix these two relations. A major part of our future intended work is the study of these relations in different topologies.

3. Simulation

A simulation is performed using discrete event simulator Network Simulator 2 (NS-2). A new application and a new agent is introduced to test the idea of multiple FTP transfers to a single client simultaneously. The simulation results are validated using quantitative stochastic simulator Akaroa-2 in independent observation mode, due to the independent nature of the observed parameters.

3.1. Topology

Following network topologies are considered in our simulation:

1. Topology 1 is generated with Boston University Representative Internet Topology Generator (BRITE) with 10 nodes and 19 links shown in fig 2.

2. Topology 2 is created to study the affect of P-FTP approach on the download time for very large files. A

Table 1. Mirror servers delay and suitability

Mirror Server	Delay(sec)	Suitability
3	0.03024	0.2691
4	0.0452311	0.17994
5	0.02016	0.40372
6	0.0553004	0.1471

Table 2. Download time from single server

Mirror Server	Download Time (sec)
3,4,5,6 (P-FTP)	2.3163
3 (FTP)	6.23067
4 (FTP)	9.33167
5 (FTP)	4.15397
6 (FTP)	11.4014

flat router level topology based on Waxman's probability model is generated using BRITE with 900 network nodes, where each network node is a router. The capacities of the network links is distributed in the interval [5,155] Mbps.

Ten mirror servers and five requesting clients are connected to the routers randomly with 10 Mbps link having 5 msec delay. Multiple traffic sources and sinks are introduced in the network randomly to produce the competing traffic.

3.2. Simulation methodology

In simulation, the client requests a file randomly from P-FTP server, file size ranges from 500 KByte to 5 MByte for topology 1 and 5 MByte to 50 MByte for topology 2. Each file is replicated on at most five mirror servers. The P-FTP server calculates the mirror server's suitability and indicates the result to the mirror servers and the requesting client with inform and reply messages respectively. The mirror servers start transferring the portion of file to the client after receiving a FTP request from client. On receiving complete file the client calculates the time taken to complete the request. The time is calculated from the instance when the client first requested the P-FTP server to the instance file transfer was completed. The mirror server's utilization is updated by the mirror servers before the start of transfer of file and after the file transfer is finished for simplicity. The end-to-end delay from each mirror server and each client is calculated by monitoring agents introduced on the servers and clients.

3.3. Simulation results

The effect of P-FTP on download time is explained by comparing the file transfer process of P-FTP to simple FTP by considering an example of client 5 and file 4 in topology 1. File 4 is replicated on four mirror servers 3,4,5 and 6. The end-to-end delay values between client 5 and mirror servers taken from the database and mirror servers suitability values calculated by P-FTP server using LDP are shown in table 1. The suitability values are calculated by using eq 5 and 6 as end to end delay is an inverse OV. The minimum OV value, V_Z (0.02016) is searched, then s_k and S_k values for mirror servers 3, 4, 5 and 6 are calculated using eq 5 and 6 in the following manner:

$$\begin{aligned}
s_3 &= \frac{V_z}{V_3} = \frac{0.02016}{0.03024} = 0.6667 \\
s_4 &= \frac{V_z}{V_4} = \frac{0.02016}{0.0452311} = 0.44571 \\
s_5 &= \frac{V_z}{V_5} = \frac{0.02016}{0.02016} = 1.0 \\
s_6 &= \frac{V_z}{V_6} = \frac{0.02016}{0.0553004} = 0.36455 \\
S_3 &= \frac{s_3}{\sum_{k=1}^{m} s_k} = \frac{0.6667}{2.47696} = 0.2691
\end{aligned}$$

Similarly the suitability values for mirror servers 4, 5 and 6 are calculated and are shown in table 1. The suitability value of each mirror server indicates the fraction of file to be downloaded from each mirror server. LDP optimizes the file transfer process on the basis of end to end delay, that is why the mirror server 6 which has the highest end to end delay from the client, is supposed to download the smallest fraction of file, 0.14 only. Mirror servers 3,4,5, and 6 transfer the parts of file 4 to client 5 according to their respective suitabilities. Client 5 calculates the download delay after receiving complete file using P-FTP approach. Client 5 downloads file 4 from each mirror server separately using traditional FTP approach and calculates the respective download delays. Table 2 shows the download time for P-FTP and traditional FTP approach.

Fig 3 and 4 show improvement in terms of time when files of different sizes are downloaded using P-FTP approach as compared to traditional FTP approach in topology 1 and topology 2 respectively. The time difference is more obvious for large files than small files as in later case the overhead of the message exchange between the P-FTP entities overshadows the delay measurement. The time measurements in fig 3 and 4 with P-FTP approach does not follow a straight line, however the download time for normal FTP is directly proportional to the file size. The reason is that for P-FTP each file is downloaded from different set of servers whereas normal FTP downloads all files from geographically closest mirror server.

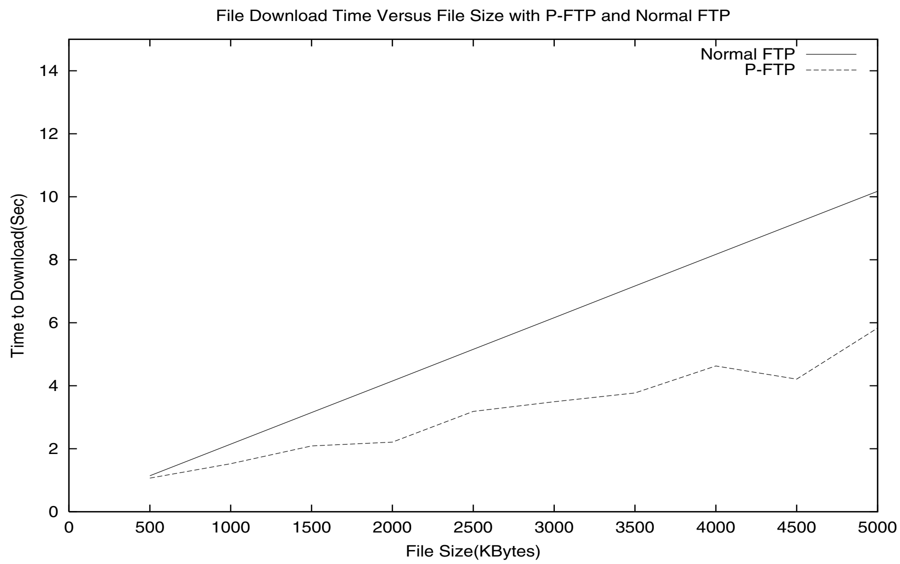

Figure 3. Simulation results for topology 1

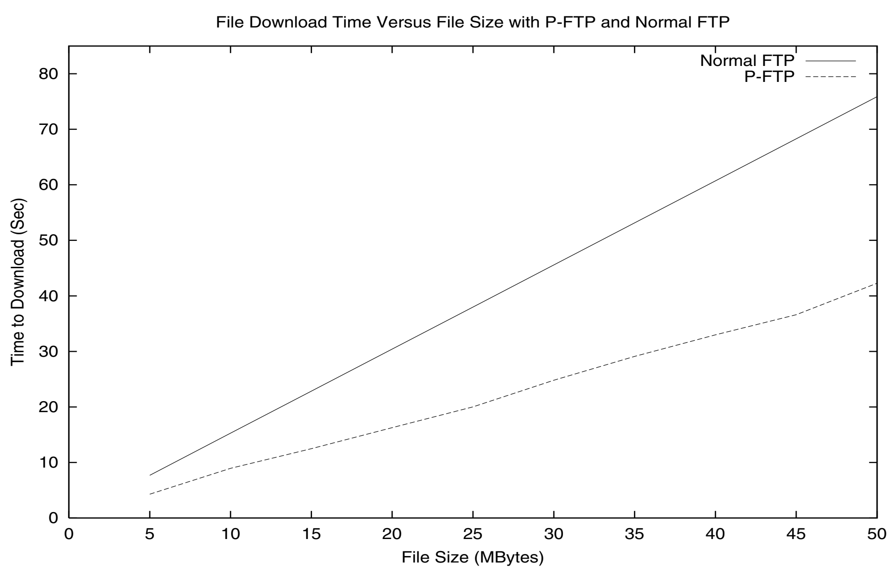

Figure 4. Simulation results for topology 2

4. Future work and conclusion

The simulation results show significant improvement in download time for large files when P-FTP approach is used as compared to traditional FTP, however much work remains to be done. Several issues such as frequency of state update, deadlock resolution and prototype implementation of protocol are some of the intended future work. The impact of different approaches used to calculate the network parameters is to be studied in detail in future. The introduction of P-FTP mechanism in video on demand and video streaming scenarios needs more investigation.

The paper presents a parallelized FTP approach which tries to solve the slow download problem of large multimedia files on the Internet. Simultaneous download of a single file from multiple mirror servers optimizes the download process with respect to defined network and/or server characteristic. Different optimization policies provide mechanism to enforce particular criteria for selection of appropriate mirror servers to download file. The P-FTP monitoring ability dynamically updates the division of file during file transfer to keep the file transferring process at optimum level.

Acknowledgments. We appreciate feedback and comments from Dr. Bruce Howarth and Dr. Tim Moors.

References

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

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

[6] 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.

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

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

[9] www.kazaa.com

[10] www.furthurnet.com

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