

On Large Scale Deployment of Parallelized File Transfer Protocol

Shaleeza Sohail*, Chun Tung Chou*, Salil S. Kanhere*, Sanjay Jha*[†]

*University of New South Wales, 2052, Sydney, Australia .

[†]National ICT Australia, 1430, Sydney, Australia

Email:- (sohails,ctchou,salilk,sjha)@cse.unsw.edu.au

Abstract

The parallelized file transfer protocol (P-FTP) is a novel network resource aware parallel technique for improving file transfer performance on the Internet. Before starting the parallel file transfer sessions, it considers the available resources in the network (available bandwidth) and at the file servers (memory and CPU utilization). The client dynamically changes the file portions being downloaded from different file servers by monitoring the FTP flows and detecting slow servers and congested links. Early experimentation on Planet-Lab [1] for a single P-FTP client suggests that the download time can be reduced by more than 50% for large files. In this paper, our goal is to evaluate whether P-FTP can be widely adopted within the Internet. To this end, we have carried out a simulation-based study to investigate the performance of P-FTP when it is adopted by a large user base. We find that, by virtue of its self-tuning capability, P-FTP continues to exhibit improved performance even with many simultaneous clients. Our results also demonstrate that introducing a large number of P-FTP users has no adverse effect on the performance perceived by users of the traditional single server file transfer. We attribute this improvement to the fact that P-FTP dynamically adapts the parallel sessions in response to changes in network state and server resources. This illustrates that P-FTP is highly scalable and is hence suitable for widespread deployment in the Internet.

1 Introduction

File transfer is one of the most commonly used application over the Internet today. Large files require long duration of time to download. As a result, they are highly susceptible to congestion that results from the high network dynamism that is prevalent in the Internet. This can result in a further increase in the download delay due to bottleneck links along the download path. Numerous studies of Internet traffic characteristics have shown that file size distributions are heavy-tailed [2]. This implies that a significant number of large files are downloaded over the Internet. Hence, it becomes important to reduce the download time

for large files. In recent years, numerous research efforts have addressed the issue of reducing the file transfer duration for large files on the Internet [14, 5, 6]. All these approaches assume the presence of multiple mirror servers with replicated content. The download time can be significantly reduced by concurrently retrieving different parts of the file from different servers. Experimental analysis and simulation studies have shown that these approaches show significant reduction in download times from the perspective of a single client. However, the parallel downloading techniques have raised concerns among a group of researchers who believe that large deployment of such techniques may be counter-productive and the single client test results may not be valid for large number of users. Their main concern in this regard is that if a large number of clients use parallel download methods then, the download delay will not reduce as significantly as it does when tested for a single client. Moreover, they claim that due to the initialization of more file transfer sessions, the load on the file-servers and the network will increase. Hence, the clients not using parallel download techniques will suffer from poor performance.

In [7], we have proposed a parallel downloading technique, called Parallelized File Transfer Protocol (P-FTP) for reducing the download time of large files. Our technique proposes the use of multiple download sessions with different FTP servers to download disjoint portions of a file in parallel. The P-FTP approach is based on the idea that the parallel use of available resources may enhance performance for resource intensive applications such as large file transfers. However, an important distinction between P-FTP and similar parallel download approaches proposed elsewhere, is that P-FTP considers resource availability in the network as well as at the mirror servers, before starting the file sessions. Moreover, our approach can dynamically adapt to the changing conditions of the servers and the network during the file transfers. We claim that due to the consideration of these issues, the large scale deployment of the P-FTP technique is practical and scalable. Moreover, all P-FTP clients enjoy better performance and will not influence other clients who are not using this approach.

In our prior work [7, 8], we have demonstrated that P-FTP shows significant improvement in terms of download time for a single P-FTP client. In this paper, our goal is to evaluate whether P-FTP can be widely adopted within the Internet. To this end, we have carried out a simulation-based study to investigate the performance of P-FTP when it is adopted by a large user base. This paper makes the following important contributions:

- The study shows that even with a large number of simultaneous P-FTP users, the performance of the P-FTP clients is better than the users using the traditional single server download approach.
- Secondly, our experiments indicate that the impact on the servers and the network is minimal due to the self-tuning ability of the P-FTP.
- The paper also presents convincing evidence that if a large number of users are downloading files using P-FTP, there is no detrimental effect of the performance of other clients who are downloading files from a single file server using the traditional file transfer approach.

The rest of the paper is organized as follows: Section 2 provides a brief description of the work related to the P-FTP approach and the issues concerning its large scale deployment. Section 3 summarizes the P-FTP approach in a nutshell and also briefly describes the implementation results obtained with P-FTP. Section 4 discusses few features of the P-FTP approach that makes it suitable for large scale deployment. Section 5 provides a detailed discussion of the simulations carried out to study the effect of large scale deployment of the P-FTP approach. Finally, Section 6 concludes the paper and provides some future work options.

2 Related Work

There are numerous approaches that use the idea of multiple simultaneous connections to reduce the download time for large files. A few approaches that use the client-server paradigm are ([14], [6] and [5]). However, none of these approaches consider network and server resources as a parameter to influence the selection of the mirror servers, prior to the start of the download process. On the contrary, network parameters such as available bandwidth and end to end delay are the basis for the mirror server selection in our approach. Few other approaches that propose parallel downloading ([9] and [11]), require either a change in the operation of the server or change in the contents encoding method. In contrast, the P-FTP approach does not require any support from the infrastructure and has been tested with traditional FTP server implementation.

Grid-FTP [19] is designed for grid environments by extending the basic FTP to enable terabyte transfers in grid systems. It adds new modes and commands to the basic FTP standards. Grid-FTP is specifically designed for the Grid environment and for its working requires the FTP servers to support the Grid-FTP protocol. In contrast our P-FTP approach uses standard FTP and is designed to work on the existing client-server model of the Internet.

Many peer-to-peer (P2P) applications ([12], [13], [16] and [17]) provide the advantage of simultaneous partial file download. Firstly, the P2P approach is limited in the sense that the P2P application program must be running on all machines sharing the files. Secondly, unknown machines must be trusted to ensure that the received file is not corrupt. Our approach is different as it works with standard FTP implementations. However, note that a special P-FTP client is implemented on the user's machine.

From the point of view of this paper we are mainly concerned with the research efforts that discuss the large scale deployment of parallel downloading approaches. Koo et al [3] presents analytical and simulation results to show that parallel downloading approaches impact servers and the network significantly. Gkantsidis et al [4] compares the performance of three parallel downloading techniques in large scale deployment scenario with simulation. The latter has shown via simulations that by introducing many simultaneous parallel downloading clients in the network other clients experience degradation of performance.

3 Parallelized File Transfer Protocol (P-FTP)

The details of the P-FTP approach are published in [8, 7]. In this section, we present a brief overview of this approach. Interested readers can refer to [8, 7] for further details.

The P-FTP approach involves having a special P-FTP server running within the client's Autonomous System (AS). This server is capable of communicating with the file servers located anywhere in the Internet and of measuring network parameters. As opposed to traditional FTP, P-FTP client first contacts the P-FTP server when a particular file is to be downloaded. The P-FTP server then selects a set of suitable file servers from which the file can be downloaded in parallel. This information is sent back to the client which then initiates parallel file transfer sessions. In addition to this, the P-FTP server also sends a list of *backup* file servers that will be contacted by the client if there is congestion in the network or some particular file server slows down. This makes our technique capable of dynamically adapting to the changing network conditions.

The P-FTP system selects the file servers based on their utilization and the network path quality. A server's utilization depends on the server load, which includes its CPU

and memory utilization. The path quality is determined by such parameters as available bandwidth, round trip time and packet dropping probability. However any other network-related characteristic can be used. A database of these parameters is maintained and regularly updated in order to improve the P-FTP server's response time. All available servers are ranked based on these parameters. The P-FTP server selects a subset of these file servers for the file transfer. The selection limitation is the available bandwidth between the client and its Internet gateway. A number of file servers are selected in such a way that the aggregate P-FTP flow does not produce congestion at the clients access link. Given this information, the P-FTP server determines the respective file portions that can be loaded from the selected file servers. This information is then sent to the client.

The client is able to dynamically adapt to changing network conditions. We focus on two such conditions: congested common links and the file server's throughput. Our client can detect servers that are sharing a common congested link on their path to the client. The connections to such servers are terminated upon successful detection and new connections with the *backup* servers nominated by the P-FTP server are initialized. Moreover, the throughput of the file servers, relative to themselves, is regularly monitored to detect low-performing file servers. The file portions to be downloaded from slow servers is reduced. The client starts new sessions with the *backup* servers to download the remaining file portions.

The P-FTP approach has been tested on the Internet using the Planet-Lab infrastructure [1] and the results are presented in [8]. The empirical results show at least 50% improvement in download time when compared to a traditional file-transfer approach. Moreover, download time measured with the P-FTP approach shows significantly less standard deviation than download time measured with a traditional single server file transfer approach. The small standard deviation of download time for the P-FTP approach is another advantage of the approach which enables files to be downloaded in a predictable range of time.

4 Large Scale Deployment of P-FTP

As discussed in the earlier section, the implementation results of the P-FTP approach show significant improvement in terms of the download time for the P-FTP clients. However, it is important to note that these results were collected from the perspective of a single client. The natural question that follows is whether one would expect similar performance improvements if a large number of clients were downloading files using P-FTP. Another issue that needs to be addressed before P-FTP can be widely deployed on the Internet is the effect of P-FTP on other clients not using this approach. It is of paramount impor-

tance that the large user base of P-FTP clients does not affect the performance of other clients who are downloading files using the traditional single client approach. Our main goal in this paper is to answer these questions and to highlight the fact that P-FTP can indeed be suitably deployed on a large scale in the Internet. To this end, we have carried out a rigorous set of simulations and analyzed the results. However, before we present the simulation analysis, we first discuss some of the important features of the P-FTP approach, which we believe are the key factors that make it suitable for large scale Internet deployment:

1. The P-FTP server selects the most suitable file-servers for the clients at the time of the request. The size of file portion downloaded from each server depends upon the network characteristics of its path to the requesting client. Hence, the most responsive servers are selected for file transfer and larger file portions are downloaded from the server with more resources.
2. In case of heavy demand for any file in a small time interval, the P-FTP server tries to distribute load among different set of file servers. In that case the servers with less available resources are also nominated for file transfers.
3. The clients access link is usually the bottleneck for the complete path between the client and the servers. Hence, minimum number of parallel sessions are started by the P-FTP client depending upon the available bandwidth at its access link. The P-FTP clients with very small access bandwidth download all files from a single server.
4. In case of congestion the number of parallel file transfer sessions are reduced to ease congestion.
5. The P-FTP approach is suitable and recommended for large files only.

5 Analysis and Simulation Results

In this section, we study with the help of simulations, the effect on the network, file-servers and the clients when a large number of P-FTP clients are present in the network. Ideally, we would have liked to perform Internet experiments similar to the ones conducted in [8] using PlanetLab. We are interested, however, in considering scenarios with large number of clients and we also desire to provide more control and repeatability than would be possible using Internet experiments. We have, therefore, conducted a set of ns-2 simulations as described below.

5.1 Simulation Topology

A network topology consisting of 100 routers and 10 file-servers was generated with Boston University Representative Internet Topology Generator (BRITE) [18]. The

file servers, the clients and the P-FTP server were connected to a randomly chosen router. The clients connect to the file servers using the Full-TCP model of ns-2. Background load was introduced into the network using on/off UDP flows at random paths of the network. The simulations were conducted with three different number of clients depicting three level of load in the network; 1) 100 clients as low load, 2) 200 clients as medium load, and 3) 300 clients as high load. All the results were collected at steady state and average values are shown in the results.

5.2 Simulation Methodology

In our experiments we compare the download delays perceived by clients that use the following two download techniques:

1. The Single Download technique (SDT) clients download files using TCP/IP protocol suit from a single file-server. Two approaches were used to select the file-server for SDT file transfers, the random selection (SDT-R) and the selection of the geographically closest server (SDT-L) from the client.
2. The P-FTP clients send a request to the P-FTP server before starting the file download process. On receiving the request, the P-FTP server calculate the file portion values for the selected file-servers on the basis of the information maintained in the database. During simulations, the database only consisted of information about end to end delay between the clients and the file servers. The file server utilization information was not available during the empirical evaluation of the P-FTP approach [8]. Hence, to keep the simulation scenario close to the real world experience, we did not consider file server utilization during the selection process of the file servers. However, by including this information the file server selection process would be improved and the download time for the P-FTP clients would reduce further. New monitoring agents were designed for NS-2 to measure the end-to-end delay between the file servers and the clients at predefined intervals. The file download time for the P-FTP clients was measured from the instance the request was sent to the P-FTP server to the instance the complete file was received at the client. This is to incorporate the delay due to the message passing among P-FTP server and the clients in file download.

The inter-arrival time between the file transfer requests were chosen randomly for every request of each client. For accurate comparison of the download techniques, it is very important that the behavior of the clients is independent of the approach being used for file transfer. As a result, the request inter-arrival time is kept independent of the download technique and the download delay.

5.3 The Effect of Increasing Load

The first set of experiments were conducted to measure the file download time for the P-FTP and SDT clients for low, medium and high load. During simulations both clients randomly chose a file size between 1-10 MBytes to download. Figures 1, 2 and 3 show the file download time for different files for the P-FTP and SDT approaches under different load conditions. Under all type of loads the download time for the P-FTP clients is much lower than the download time for both type of the SDT clients. One can readily see that the difference in download time increases with the file size. The results show that the P-FTP approach is more effective than any of the SDT approaches. Moreover, the P-FTP approach reduced the download delay of the large files more significantly than the small files. The download time for SDT-L and SDT-R was almost same for all load types with SDT-L always performing a bit better.

The load increase influences the performance of both the approaches. To study this effect in greater detail, we calculate the percentage increase in download time for P-FTP and SDT-L approaches when number of users were increased from 100 to 200 and 300 respectively. The results are shown in figure 4. The base value for the calculations was the download time of the respective approach with 100 users. Figure 4 shows that the impact of the load on both approaches was almost similar. Figure 5 shows the mean percentage increase in the download time for the P-FTP, SDT-L and SDT-R approaches. The mean increase in download delay for SDT-L and P-FTP for medium and high loads (200 and 300) was almost the same. This indicates that an increase in load degrades the performance of both the approaches in a similar manner. The mean increase in delay of the SDT-R approach was higher for 300 users. This is possibly due to the fact that the randomly selected server in this approach could be geographically distant from the client and under high load this could increase the download time considerably. It is important to point out that the performance degradation of the P-FTP approach with increase in load is similar to what one would expect with any other application in a similar scenario. The effect of load increase on the P-FTP approach is not a disadvantage of its large scale deployment but is the inherent behavior of any application. Any application would exhibit a similar trend, i.e., the waiting time would increase with an increase in load. It is especially important to highlight the fact that the increase in download time with P-FTP is not an exponential function of the load, but is rather some form of linear function. This implies that the saturation effect is not as severe and hence indicates that P-FTP is actually quite scalable. This set of experiments also convincingly shows that the performance of the P-FTP clients is significantly better than the traditional file transfer clients,

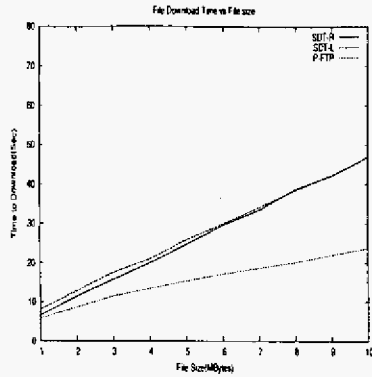


Figure 1: File Size vs Download Time (Low Load)

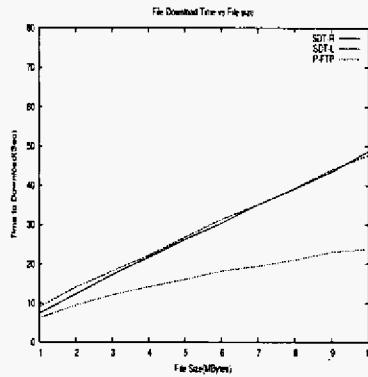


Figure 2: File Size vs Download Time (Medium load)

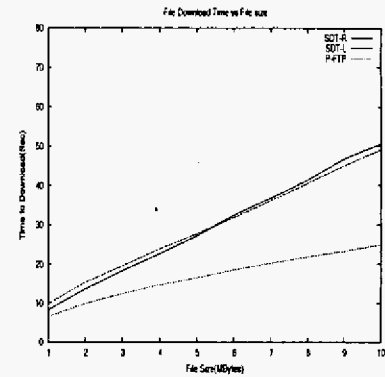


Figure 3: File Size vs Download Time (High load)

Download time vs User population

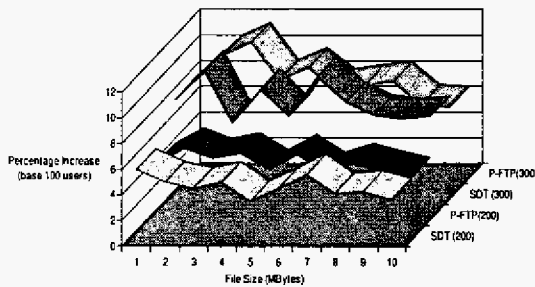


Figure 4: Percentage Increase in Download Time with Increase in Load

Mean Percentage Increase with Increase in Load

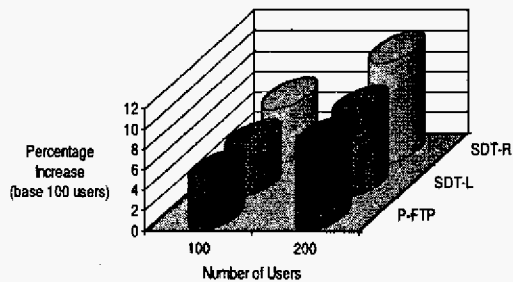


Figure 5: Mean Percentage Increase in Download Time

even when large number of users adopt the respective approach. Thus, P-FTP is highly scalable and suitable for large scale Internet deployment.

5.4 The Number of File Transfer Sessions in the Network

The total number of file transfer sessions in the network were counted at different instances of time during the simulations described in previous section. The network connections were counted by summing the TCP sessions at all file servers. The average number of connections were calculated for every ten second interval. Figures 6, 7 and 8 show the average number of connections in the network for low, medium and high load for the P-FTP and SDT-L approaches. The figures show that the average number of file transfer sessions were almost the same for both the approaches at different loads. The P-FTP clients initiated more file transfer sessions but the duration of these sessions were shorter than the sessions initiated by SDT clients. Hence, the average number of connections in the network remained same. However, for P-FTP the number of connections fluctuated more than that for SDT-L.

It is important to emphasize that the P-FTP approach calculates the optimal number of parallel sessions for a file download on the basis of the bandwidth of the access link which connects the client network to the global Internet. This is due to the fact that the access link is almost always the bottleneck link along the end-to-end paths between the client and the file servers. Increasing the number of parallel sessions beyond the optimal value does not decrease the download delay. In fact, this increase can have a detrimental effect on the performance since it increase the number of connections in the network and at the file servers. The P-FTP approach, hence, minimizes the number of parallel file transfer sessions. As a result, the average number of file transfer sessions started by the P-FTP clients are almost always equal to the number of connections initiated

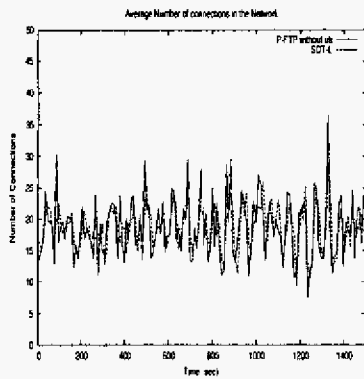


Figure 6: Average Number of Connections (Low load)

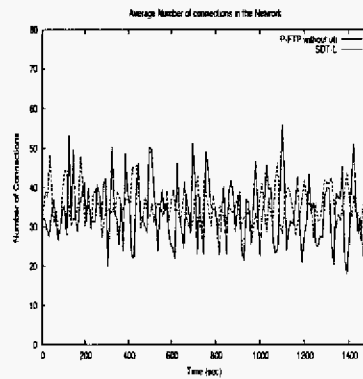


Figure 7: Average Number of Connections (Medium load)

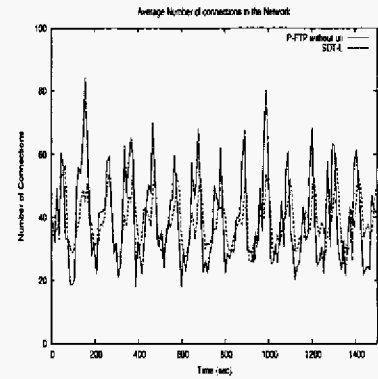


Figure 8: Average Number of Connections (High load)

by the same number of traditional clients.

5.5 The Effect on File-Servers and Other Users

The P-FTP approach is based on the initiation of multiple file transfer sessions by the clients to accelerate the download process. In other words, the P-FTP clients initiate more short file transfer sessions as compared to a single long file transfer session by the traditional clients. Hence, there are more connection initiation requests handled by the file servers when users use the P-FTP approach. In this set of simulations we intend to study the effect of P-FTP on the file-servers. We also investigate if there is any performance degradation observed by users of traditional file transfer when a large number of P-FTP clients are simultaneously downloading files. For the latter set of experiments we use real data obtained from the logs of FTP transfers.

The processing time of the file transfer initiation request at the FTP server was calculated on a Pentium 3 machine with 256 MBytes RAM. The processing time was the time from the instance the file server received the request to the instance the server started sending the data to the client. This included the time required for TCP session establishment, FTP session establishment, change of directory requested by client (if applicable) and the disk seek time for the position in file from where data needs to be sent to the client. A Pure-FTP server implementation [10] was installed on the test machine. Multiple P-FTP clients requested for random portions of different files from the FTP server. The processing time of 50,000 requests was calculated. The average processing time for a single request was 50 msec. The effect of already established FTP connections on the overall response time of the request was also studied by sending different number of simultaneous requests to the server. For every connected client the processing cost increased by approximately 2-3 msec. The difference in processing time of a P-FTP request from that of a SDT request is due to different number of already es-

tablished connections at the file servers in each case. In case of all P-FTP clients the number of connections in the network and at the file server are more than the case of all SDT clients. Figure 8 shows that the maximum number of simultaneous connections for P-FTP was 85 for high load and that for SDT was 55. On average each of the ten file servers have 3 more connection in case of P-FTP, which adds a delay of 6-9 msec to the processing time of the P-FTP request as compared to the SDT request. However, one can readily observe that this small difference in processing time does not produce any noticeable impact on the download time of both approaches discussed in the section 5.3. The above mentioned tests show that the P-FTP approach can produce extra load on the file servers when large number of clients are using this approach. The excess load is in the shape of extra file transfer initiation requests by the clients. If there are large number of P-FTP clients requesting for new connections to the file servers, the excessive load can degrade the performance of the file servers and all the users using those file-servers may get affected. We have considered this factor and hence, restricted the P-FTP approach for transfer of large files only as: firstly, the large files constitute a small ratio of total downloads on the Internet, and secondly, the download time of large files is reduced significantly when downloaded with P-FTP. We have performed some simulations to study the effect of these factors.

For the study of the file size distribution pattern, we have studied the logs of all FTP transfers in the school of Computer Science and Engineering of UNSW for one week from 1st June 2004 to 7th June 2004. The total users of the school network are almost 3,500 and the log files contained information about more than 2,000 FTP transfers. From those logs we were able to come up with the file size and frequency distribution for FTP transfers for the users of CSE. Figure 9 shows the ratio of file sizes

File Size and Frequency Distribution

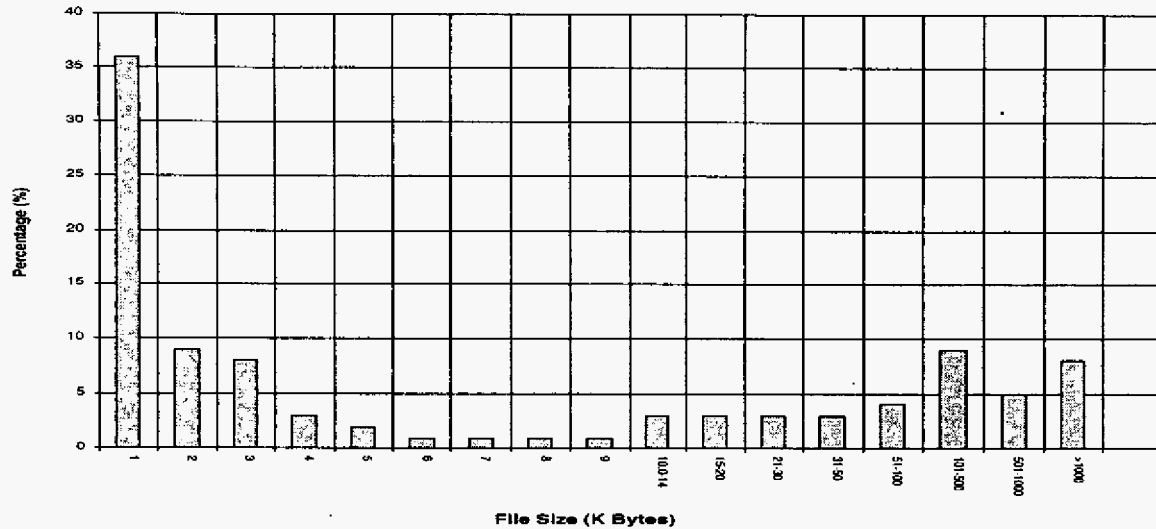


Figure 9: File Size Distribution

transferred with FTP during the first week of June 2004. The figure shows that less than 10% of the files were larger than 1 MByte. We consider these files which are greater than 1 MByte as *large files* and all files smaller than 1 MByte as *small files* for further discussion.

Simulations were performed for this real world file size distribution (taken from figure 10) with 300 clients (heavy load) where only 10% of the files were larger than 1 MByte. The following three cases were used for simulations:

1. Case 1:- All files were downloaded with SDT-L approach.
2. Case 2:- All files were downloaded with P-FTP approach.
3. Case 3:- Large files downloaded with P-FTP and small files downloaded with SDT-L approach.

The download time measured during simulations for the above mentioned three cases is shown in figure 10. By comparing the results of figure 10, it is clear that:

1. The download time of small files was slightly improved with P-FTP approach as compared to the traditional approach. When the small files were downloaded with P-FTP (*ALL P-FTP* in figure), the change in download time was very small as compared to

when they were downloaded with SDT-L (*ALL SDT-L* in figure).

2. Large files were downloaded in significantly less time with P-FTP as compared to the traditional approach. The difference in download time of large files was significant when downloaded with P-FTP (*ALL P-FTP, P-FTP + SDT-L* in figure) as compared to the SDT-L approach.
3. If only the large files were downloaded with P-FTP, the download process of the small files did not get influenced. The download time of small files was the same for *P-FTP + SDT-L* and *SDT-L* cases. This shows that the download delay of the large files can be reduced considerably with P-FTP without affecting other downloads.

From the results of figure 10 it is evident, that by restricting the P-FTP approach for large files we accomplish two goals: 1) The burden of extra file transfer session initiations on the file servers is minimized, and 2) the large files are downloaded in reduced amount of time without affecting the download process of other files.

Therefore, we propose that the P-FTP approach is suitable for transfers of such large files. When only the 10% of the files are downloaded with the P-FTP approach then the excessive load on the file servers is very low. This small extra load on the FTP servers, which are usually dedicated

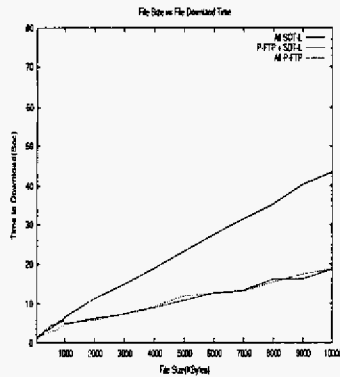


Figure 10: File size vs Download Time (High Load)

machines, can be justified against the gain in the download time for the P-FTP clients. Secondly, the P-FTP server tries to distribute load among different set of file servers during intense load conditions so the impact on every server is minimized.

6 Conclusion and Future Work

Parallelized FTP (P-FTP) is a novel network resource aware parallel downloading technique for improving the performance of file transfer. This paper addresses important issues which are related to the feasibility of large scale deployment of the P-FTP approach. A rigorous simulation based evaluation is presented to highlight the important properties of P-FTP which make it suitable for widespread deployment in the Internet. In particular, the two key features that distinguish P-FTP from other parallelized download schemes are: (i) the network aware selection of file servers and (ii) its ability to dynamically adapt to the changing network conditions. We conclusively show that the download time for P-FTP clients is significantly lower as compared to traditional clients, even when large number of simultaneous clients are present in the network. Secondly, the effect on the network and the file servers is minimized due to the self tuning ability of the approach. In addition, by restricting the use of P-FTP for downloading large files only, we also show that there is no degradation in the perceived performance of other clients who are using the traditional single server file transfer techniques.

As part of our future work, we are currently developing large-scale experiments for carrying out similar tests on PlanetLab. The design of a scalable mechanism for P-FTP database initialization and maintenance is under way. Finally, the issues involved in the deployment of P-FTP on mobile nodes are also being investigated.

References

[1] Planet-Lab <http://www.planet-lab.org/> Jun-2004

- [2] M. Crovella and A. Bestavros, *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes*, IEEE/ACM Transaction on Networking, 5 (6), 835-846, 1997.
- [3] S. Koo, C. Rosenberg and D. Xu, *Analysis of Parallel Downloading for Large File Distribution*, In Proceedings of IEEE International Workshop on Future Trends in Distributed Computing Systems, 128-138, 2003.
- [4] C. Gkantsidis, M. Ammar and E. Zegura, *On the Effect of Large-Scale Deployment of Parallel Downloading*, In Proceedings of The Third IEEE Workshop on Internet Applications, 79-90, 2003.
- [5] J. Byers et al, *Informed Content Delivery Across Adaptive Overlay Networks*, In Proceedings of ACM SIGCOMM, 275-283, 2002.
- [6] P. Rodriguez and E. Biersack, *Dynamic Parallel Access to Replicated Content in the Internet*, IEEE/ACM Transaction on Networking, 10 (4), 455-465, 2002.
- [7] S. Sohail, S. Jha and H. Elgindy, *Parallelized File Transfer Protocol*, High Speed Local Networks, In Proceedings of IEEE LCN, 624-631, 2003.
- [8] S. Sohail and S. Jha, *QoS Driven Parallelized File-Transfer-Protocol*, In Proceedings of ATNAC, 2004.
- [9] J. Buyers et al, *Accessing Multiple Mirror sites in Parallel: Using Tornado Code to Speed up Downloads*, In Proceedings of IEEE INFOCOM, 275-283, 1999.
- [10] Pure-FTP, <http://www.pureftpd.org> July 2004
- [11] Jigsaw Download, <http://atterer.net/jigdo/> Jun-2004.
- [12] Kazaa, www.kazaa.com Dec-2003
- [13] Further Network, www.furthurnet.com Dec-2003
- [14] Download Accelerator Plus, www.speedbit.com/DAPDL.asp Dec-2003
- [15] Network Simulator-2, <http://www.isi.edu/nsnam/ns/> Jun-2004
- [16] Bittorent, <http://bitconjurer.org/BitTorrent/> Dec-2003
- [17] Slurpie, <http://slurpie.sourceforge.net/> Jun-2004
- [18] BRITE, <http://www.cs.bu.edu/brite/> Jan-2004
- [19] Globus, <http://www.globus.org/datagrid/gridftp.html> Jun-2004