Delay Reduction In File Download Through Parallelization

Miss Priyanka Sahu
Pursuing M.E. from Shri Ram Institute of Technology
Jabalpur, India
priyanka.sahu7@yahoo.com

Mr. Brajesh Patel
Head Of Department of Computer Science Deptt.
Shri Ram Institute of Technology
Jabalpur, India
brajesh.patel@rediffamil.com

Abstract-- **File downloading process on Internet is generally slow. Here we will discuss about design and simulation of a distributed and synchronized file transfer protocol meant for the internet applications. The approach consists of centralized server which distributes the downloading process across multiple file servers and this is based on QoS parameter as available bandwidth. The client continuously monitors the FTP flows to detect slow servers and accordingly adjusts the file distribution. This parallelized FTP approach, stabs for the solution of the problem of slow downloads of bulky multimedia files while at the same time optimizing the utilization of mirror servers. This approach proposes simultaneous downloads of file from multiple file servers, and is combination of DAP and P-FTP. The performance can be increased by usage of slow server detection and link failure algorithms. In this paper, simulation is presented; which shows at least 50% reduction in download time, when compared to the traditional file-transfer approach.**

Keywords- Distributed Applications, Performance Measurements, Internet, Download file, NS2.

I. INTRODUCTION

Inherent dynamic and unpredictable nature for file downloads is exhibited by Internet. More time is required to download large files, during which the resource availability on the Internet changes drastically, making the download process unpredictable and usually very slow. For load distribution files are replicated on multiple mirror servers. File Transfer Protocol (FTP) is the most reliable and widely used protocol for file transfer. Clients are in the habit of selecting a mirror server that is geographically closer for file download; this approach assumes file download from such server will take minimum time and will produce least congestion on Internet, which is wrong as the geographically closest server can be highly congested, which would further increase the file downloading time and on top of it increase congestion at the server. Therefore this selection criterion is not acceptable especially if other mirror servers have high resource availability at that time. Some early researches proposed a server selection technique on the basis of hop counts and round trip delay, while others 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. Others took the complexity of the server selection technique a step further by introduction of 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.

(DAP) Download Accelerator Plus has claimed to select the most responsive mirror servers and to download the file simultaneously from mirror servers. Some researches introduced a dynamic technique to download data from multiple mirror servers in parallel using HTTP. Our approach shall optimize the process of downloading a file using FTP but by selection of multiple servers on the basis of server availability and path quality. Concurrent downloading from multiple servers taking the base as available resources in the network and at the mirror server shall optimize the downloading process for least delay and better resource utilization. Our server calculates the file portions to be downloaded from each mirror server and the file portion size is based on the server resource availability. The client monitors the flows to detect slow server. Amount of file portion downloaded from such servers is reduced to avoid long transfer delays. Back-up servers are contacted to download the remaining file portions.

II. RELATED WORK

Various research groups have projected different criteria for the selection of a single mirror server for any client. Our proposal is for the use of multiple mirror servers when retrieving a single file to reduce the download time. QoS parameter of the network and the file-servers has been used in our approach to select a suitable set of mirror servers for file downloading. In case of client-server scenario, a small number of researchers have worked on transferring a single file from multiple mirror servers simultaneously. Closely related work to our proposal is the Dynamic Parallel Access Technique (DPAT), which proposes downloading large files by linking to multiple HTTP servers simultaneously. The file to be transferred is partitioned into small blocks which are then downloaded from all the selected servers. Largest number of blocks are sent via server having highest throughput. In DPAT the file server algorithms based on bandwidth prediction performed best, and on top of that, incorporating server load in server selection may further improve the performance. For server selection multiple QoS parameters related to network and mirror servers can be used. Most approaches that propose parallel downloading require either change in the operation of the servers or change in the content encoding method. Ours can work even if no support from the file servers is available. Our protocol uses standard FTP and is designed to work on the existing client-server model of the Internet. Some researchers compared the performance of four server selection techniques for file transfer. Proposed protocol system has also based its server selection technique on prediction of resource availability in network and at file servers. Network and server probing for resource usage is required for resource prediction.

Many peer-to-peer applications provide the advantage of simultaneous partial file download, and needs change in file format or special files on the servers in order to operate successfully. The P2P approach is limited since the application program must be running on all machines sharing the files. Trust has to be done on unknown machines for some P2P applications that require users to allow uploading from their machine in order to obtain good download speed. The protocol is not on competing grounds with the P2P application as it addresses the file download in client-server paradigm of the Internet. Second, it can work even if there is no support available from the file servers.

III. HYBRID PARALLELIZED FILE TRANSFER PROTOCOL.

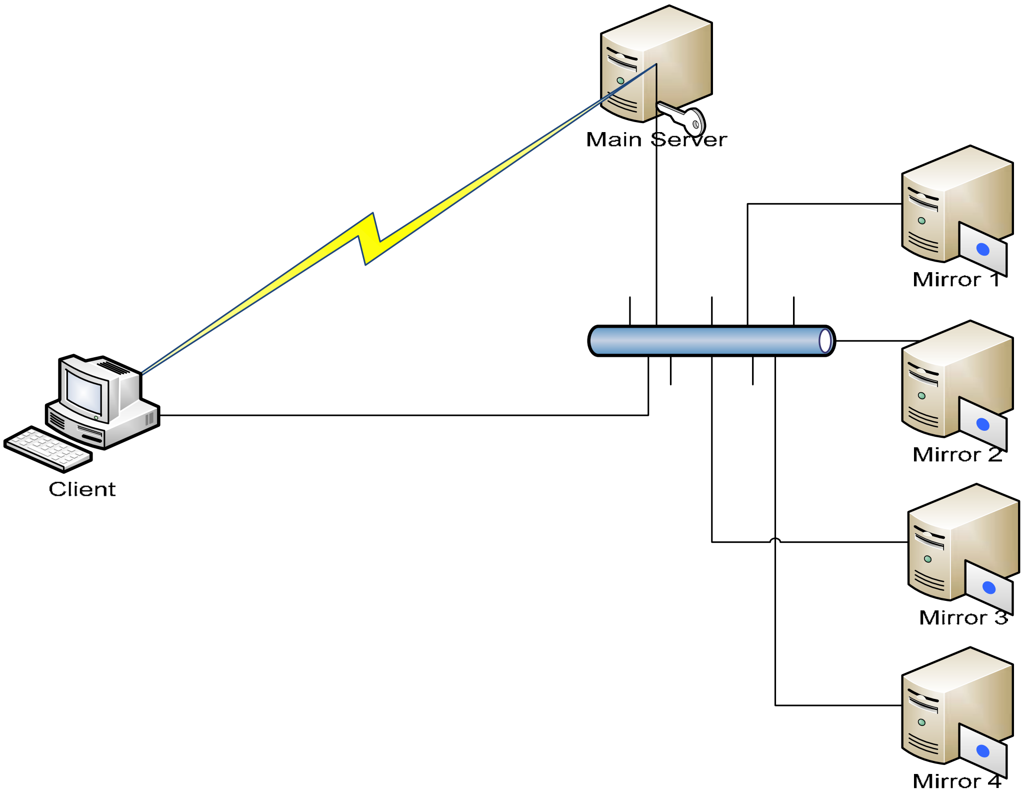

Figure 1. Interaction among entities

In our Protocol, there are two central entities 1) Server 2) Client

A. Server

Central entity in the system is the server, which controls the overall functionality of the application. Whenever a client wants to download a file, it sends a request message to the server. On request being received, the server collects information from its database about the mirror servers that contain the copy of the requested file. Depending upon the available bandwidth server gives rank to mirror servers, which reduces congestion in the network. The server is also responsible for calculating the file portions to be downloaded from each selected mirror server. The highest ranked mirror server is allocated the largest file portion and vice versa. After that server replies the client with the information about the mirror servers and the file portion to be downloaded from each of those mirror servers.

SERVER DATABASE

The database plays an important role in ranking and selection process of the server. The database contains information on the network.

RANKING AND SELECTION PROCESS

The ranking process makes use of the following information:

1-Network Characteristics: The QoS parameters, i.e. the available bandwidth the round-trip time.

2-Utilization: The H-PFTP server considers the CPU and memory utilization level of the mirror servers. Highly utilized mirror servers have lower rank. The highest ranked mirror server is allocated the largest file portion and vice versa.

B. Client

A client sends a request containing the file name and the available bandwidth to the server. After receiving a reply from the server, client starts multiple FTP connection simultaneously with the mirror servers and starts downloading files. If one or more servers fall below their expected rate, then it reduces the size of the respective allocated file portions and downloads the remaining file portions from the backup servers.

IV. ALGORITHMS

A. The Suitability Algorithm

The core of the application is the suitability algorithm, which calculates the suitability of mirror servers on the basis of the optimization policy. After receiving the request from the client, the 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. Suppose the client requests for file X, the 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 ⊂ MS Where MS is a set of mirror servers that are registered with server. The suitability algorithm finds the suitability Sm for all members of m ε 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 then the maximum threshold value and that mirror server is replaced from M to MFinal set.

MFinal ⊂ MS

MFinal ⊆ M

Where MFinal is the set of mirror servers to whom server will send inform message after the suitability algorithm finishes.

B. Algorithm

1. Initialize MFinal
2. Find FR of X
3. Find M for X
4. algo(M) [
5. **if** MFinal = ∅ **then if** M ∨ = ∅
6. **then** Calculate Sm’ ∀m using optimization policy

7. Calculate Am ← Sm * FR' ∀m
8. **if** ∀m'ARm > Am **then** Add all m to MTemp
10. Delete all m from M
11. Add All mTemp to Mfinal **else** ∀k' Sk ← Ark'FR
12. FR ← (FR − (Sk * ARk))
13. Add k to MTemp
14. Delete k from M
15. algo(M) **else** Send resources unavailable
message to client **else** Send Inform message to mFinal
16. **if** confirm message received from all mFinal
  then Send reply message to Client
17. Delete all mFinal from MFinal
18. **else** Copy all mFinal from MFinal to M
19. Delete all mFinal from MFinal
20. Delete not responding m from M
21. algo(M)]

Whenever a client wants to download a file, it sends a request message to the server. On receiving the request, the server collects information from its database about the mirror servers that contain the copy of the requested file. The server runs the suitability algorithm to calculate and evaluate the suitability of the mirror servers. The server sends an inform message to all mirror servers. On receiving the inform message mirror server stores the information and sends confirm message to the server. The server sends the reply message to the requesting client, The client initiates the multiple FTP session with all mirror servers and download of file is started.

C. Algorithm Client

Suppose the client sends a request to server to download file X. In case the client can't get reply from server in three attempts, the client downloads the file with traditional FTP approach. The reply from server contains f(ai; fi)g such that ai ϵ Active, the name-set of mirror servers and fi ϵ F, the respective file portions to be downloaded from ai. Along with this, the server also sends a name-set of additional mirror servers, Passive.

D. Algorithm

1. Initialize Attempts = 0
2. Client-Algo(Attempts)[
3. Send request for X, increment Attempts
4. **if** Receive reply
  then Connect with Active
5. Slow-Server-Algo(Active)[
6. **else** Timeout
7. **if** Attempts < 3
8. **then** H-PFTP-Client-Algo(Attempts)
else Download X with FTP
9. Slow-Server-Algo(Active)[
10. **do**Tm ϵ Active
11. Monitor the size of received data, Dm
12. Calculate Tm = Dm ← Fm
13. Normally distribute Tm and find mean, Tmean
15. **if** Any Tm <<Tmean, m ←Active← **then if** Passive = ∅;
16. Fm = FTemp
17. Start one new connection to p, where p Passive and Fp = FTemp – Fm else Readjust file fraction values

V. IMPLEMENTATION AND RESULTS

This section presents an extensive experimental evaluation. There are five protocol versions analyzed in this report.

- FTP (File Transfer Protocol)
- DAP (Download Accelerator Plus)
- DPAT (Dynamic Parallel Access Technique)

- P-FTP (Parallelized File Transfer Protocol)
- H-PFTP (Hybrid P-FTP)

All the experiments were performed on a Personal Computer running at 3.00 GHz with 512 MB of memory and 80 GB hard disk.

SIMULATION OF (HYBRID PARALLELIZED FILE TRANSFER PROTOCOL) TOPOLOGY

Following network topology is considered for (Parallelized File Transfer Protocol) simulation

- Topology contains one clients, and 4 mirror servers and ten links.
- By taking random variable the link is made to be dynamic. All links are varying between ranges 1MB – 3MB.
- The queue size of link is 20 i.e if buffer capacity of queue exceeded then last packets arrived is dropped.
- The propagation delay of link is 10ms.

METHODOLOGY

In the simulation, the client requests a file from server. The 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.

The end-to-end delay from each mirror server and each client is calculated by monitoring agents introduced on the servers and clients The packet size is 500 bytes and 100 packets are sent from mirror server to client and see the what time taken to download 100 packets. After successful execution 150, 200, 250, 300 packets are sent from mirror server to client. Packet color represents packets coming from particular connection.

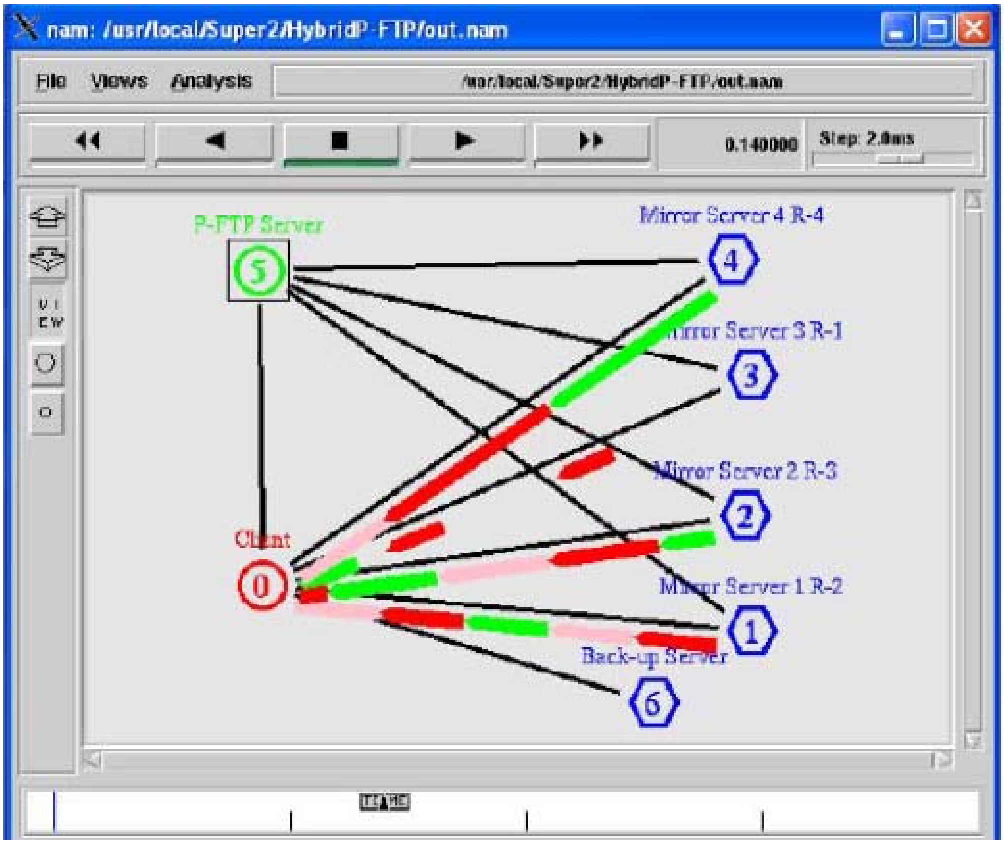

Figure 2. Mirror server sends packets to client

LINK FAILURE

If any link between client and mirror server fails, then the backup servers are invoked to download the rest of the file portion. So there is no data loss. In fig3 link between server and Mirror server 4 fails (Represented by red line). As soon as links fails client create FTP connection with back-up server and remaining data will be downloaded.

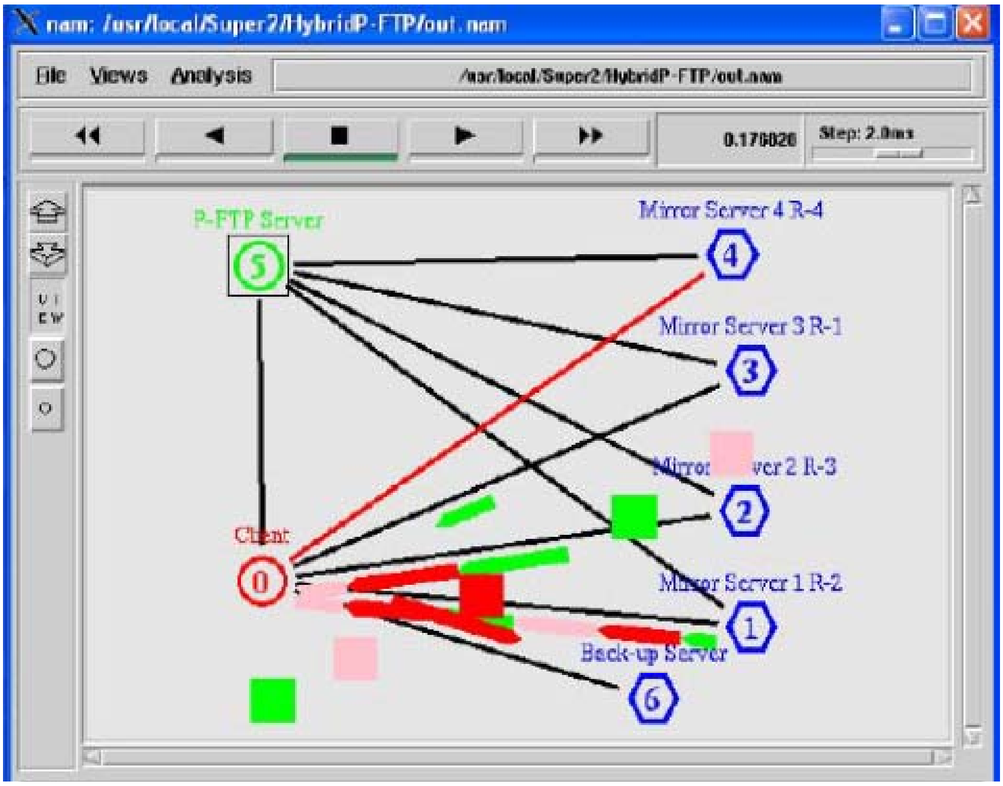

Figure 3. Link between Client and mirror server no.4 is failed.

SLOW SERVER DETECTION ALGORITHM

The server calculates the file portions, such that all file servers finish sending the file-data at approximately the same time. If any of the file servers slows down or fails during a session, then that will increase the total file download time. Client can detect this situation and act accordingly for better performance. If any server fails or slows down then the backup servers are invoked to download the rest of the file portion. In fig 4 Mirror servers 3 slows down and its remaining data will be downloaded from back-up server.

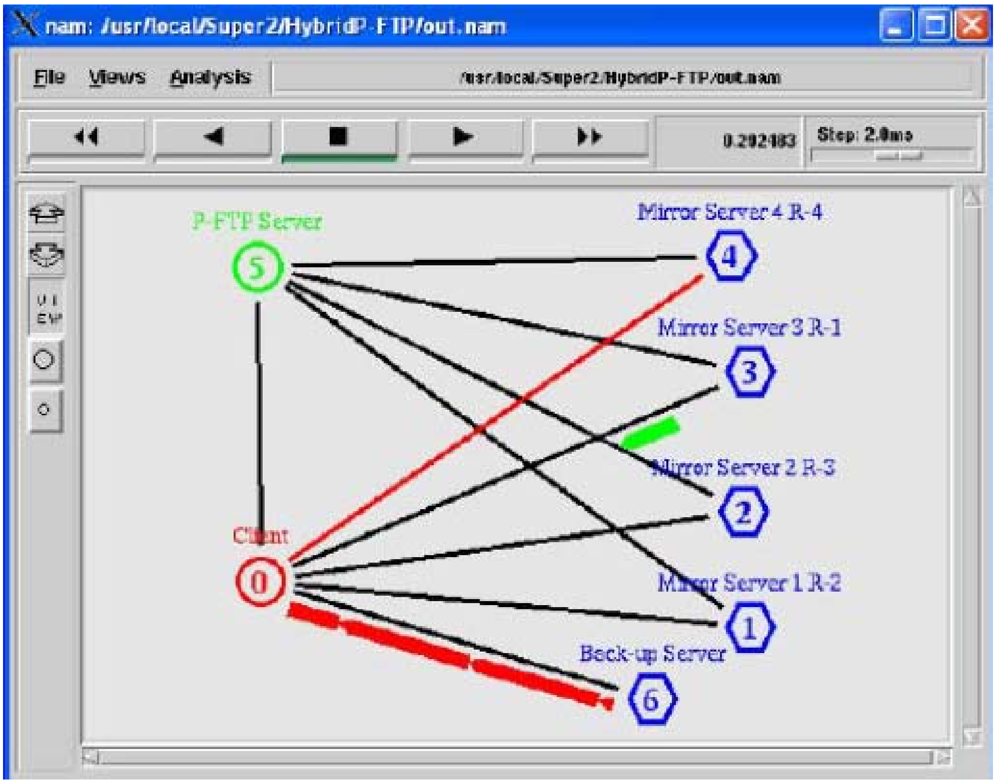

Figure 4. Mirror server no.2 is slows down and reduced data is downloaded from back-up server

PERFORMANCE ANALYSIS

There various analyzing tool for Network Simulator (NS2) .The various Network Simulator (NS2) tools that are used .

1) Tracegraph

2) NANS (Network Analyzer for Network Simulator)

The performance evaluation for the FTP (File Transfer Protocol), DAP (Download Accelerator Plus), DPAT (Dynamic Parallel Access Technique) and Hybrid Parallellized File Transfer Protocol is discussed in this chapter. The performance is evaluated based on download time, total number of packets. For comparison of approaches, files constituting from 100 – 300 packets were downloaded using these approaches.

The DPAT showed a significant impact on download delay compared to normal FTP. The simulation results showed that both approaches were better alternatives for downloading large files than the traditional download method. The impact is more vivid for large files as, for small files, the message passing among FTP entities overshadows the download delay.

VI. CONCLUSION

The file transfer protocol presented here is called Hybrid Parallelized File-Transfer-Protocol, which reduces the download time for large files, it proposes simultaneous downloads of file from multiple file servers. The performance of the protocol can be increased by using slow server detection algorithm and link failure algorithm. Mirror server is ranked by ranking process, which allows the highest ranked mirror server to be allocated the largest file portion and vice versa.

The numbers of packets are tested in current implementation. The time taken by the protocol to download these packets is very less as compared to FTP DAP and DPAT. Hybrid Parallelized file transfer Protocol and DPAT (Dynamic Parallel Access Technique) approaches are tested by taking number of servers. As we increase number of servers file download time reduced. The self tuning ability of the approach reduces the effect on the network and the file servers. Our approach shows that there is a reduction by 50% in the large file transfer delays in most cases.

VII. REFERENCES

[1] S. Sohail, S. Kanhere, S. Jha, A. Baig, and M. Malik, “A Parallelized File-Transfer-Protocol for On-Board IP Networks,” Proc. Vehicular Technology Conf., Spring 2007.

[2] G. Pultar, “Automatically Selecting a Close Mirror Based on Network Topology,” Proc. 12th Systems Administration Conf.,pp. 159-165, 2006.

[3] J. Guyton and M. Schwartz, “Locating Nearby Copies of Replicated Internet Servers,” Proc. SIGCOMM, pp. 288- 298, 2006.