Measuring Round Trip Time and File Download Time of FTP Servers

Syed Mehr Ali Shah
Email: Corresponding author mehar81@gmail.com
Department of Information Technology, Shinas College of Technology Sultanate of Oman

Abstract

Today, Internet is a public, cooperative, and self-sustaining facility accessible to hundreds of millions of people worldwide. It has made the world a global village. It uses a layered model such as TCP/IP. Data can transfer easily using ordinary file transfer method such as HTTP and emails protocols. When to transfer or download the large set of data like big collection of file via inexpensive IP networks then throughput collapses. New TCP stacks and new network acceleration devices are market to help .but fail to fully utilize many wide area network paths. Consequently, conventional FTP, and even new "acceleration" solutions, cannot provide the speed and predictability needed for global file transfers. There are some problems which may affect the download time. It may include the hardware/software configurations, location of machines, bandwidth, delay, congestion, size of the files and other type of configurations problems. The response time of the FTP server and Round Trip Time (RTT) plays an important role in download time of a file. In this paper, we investigate the possible relationship between the Round Trip Time and the software download time for the FTP servers. It also shows the factors that could have an impact on the FTP server response time. In our experiment, we assume the existing infrastructure of the Internet with its current technologies and protocols. So we concluded that if the RTT is less, the download time will be less. We presented the Karn's algorithm to improve the accuracy of RTT measurement.

Keywords: File transport protocol, Round trip time, Karn's algorithm, DNS response time, TCP/IP.

1. Introduction

FTP stands for File Transfer Protocol. It is the standard Internet protocol for transferring files from one computer to another. FTP is part of the TCP/IP protocol suite. TCP/IP is the basic protocol that runs the whole Internet. Whether you are checking your email, visiting a web site or downloading files, you are using TCP/IP. There are a number of smaller protocols that run on top of TCP/IP, such as email, HTTP, and Telnet. FTP is one of these. Its sole function is to move a file from a server to a client (download) or from a client to a server (upload). Due to the comprehensive use of File Transfer Protocol, FTP servers are demanded for better performance.

There are different factors that affect the download time of a file. These factors may include file size, link bandwidth, location of the server and client, congestion on link, system configuration and server response time. So for minimum download time, it is important that we have powerful FTP servers, a client with good configuration (hardware and software), server near to client, a link with high bandwidth and minimum Round Trip Time (RTT) of the server. When a client sends a request for a file to an FTP server, the server response time depends on number of requests to the server at that time, the server cache, memory, processor, and the file size. It is very complex to analyze the factors which are involved in a file download .But when we are talking about the round-trip time, the interval between the sending of a packet and the receipt of its acknowledgement, is a key function in many reliable transport protocols [1,2,3] Such estimates are used to ensure that data is reliably delivered. If a packet is not acknowledged for long time, it is supposed to have been lost and transmit it again. Estimated round-trip times are used to find out when these retransmissions will occur.

Three developments in IP networking [4,5,6] have led to increased interest in the problems estimating round-trip times.

First, there has been an explosive growth in the size and complexity of IP internetworks, built by interconnecting existing subnetworks. The best known example is the ARPA Internet. (The ARPANET is just one component subnetwork in the ARPA Internet.) The ARPA Internet has highly variable round-trip times. Because its paths are very complex, it also tends to lose more packets.

Second, there has been a large increase in traffic on some of the major IP networks. Higher traffic loads have led to serious network congestion on some parts of the ARPA Internet [7,8]. Like network size, congestion is known to cause highly variable round-trip times and higher packet loss rates.

Finally, recent research has shown that the standard approaches to estimating round-trip times for the Transmission Control Protocol (TCP) are inaccurate if packets are lost or round-trip times are highly variable [9,10]. This discovery is distressing because it suggests that the mechanism reliable protocols depend upon to handle loss and variable round-trip times, namely the estimation of round-trip times, may not work well. We are discussing about the FTP here ,So FTP is one part of TCP that why Nowadays mostly FTP service is used to download files because of minimum download time.But it has some constrains and factors. Most recently, FastSoft [11] characterized some factors of a file download from FTP servers. They downloaded 700 Mega bytes file from Los Angeles to San Francisco and it took 45 minutes using 3 Mbps DSL/Cable Modem. Then they downloaded the same file from Los Angeles to New York and it took 1 hour and 24 minutes. They have used various speed links. However, they did not consider the other factors like servers' configuration, clients' configuration, link congestion, load on servers and round-trip time of servers.

In this experiment we tried to compile the relationship between round-trip-time and file download time of FTP servers. We performed this experiment with real Internet FTP servers. We selected 4 different FTP servers located in different regions of the world and downloaded approximately 1 Mega Byte (MB) file from all servers.

In section 2 of this paper, we have **described** the background and overview of FTP servers, how the data is transferred, round trip time and response time of FTP servers. Section 3 contains the description of the methodology and measurements used in our experiment, analyzing and finalizing the results. In section 4 we used Karn's algorithm to improves the accuracy of Round Trip Time (RTT) measurement .Finally, we have our conclusions in section 5.

2.1 Overview of an FTP Server

FTP or File Transfer Protocol is used to transfer data from one computer to another over the Internet, or through a network [2]. In FTP transfer two computers are involved, a server and a client. An FTP server is a computer on which FTP server software is installed and listens to requests for connection establishment from another computer on the network. A client is a computer on which FTP client software is installed and initiates a connection request to the server. Once a connection is established between a server and a client, the client can upload files to the server, download files from the server, rename or delete a file on the server etc.

2.2 FTP Data Transfer

Before the popularity of FTP, HTTP will be used to download or upload data. Nowadays, FTP is a very well-known application used for the specified functions. Before data transfer, a connection is established between a client and a server. The client sends a connection request to the server by creating a socket. The client listens on that socket and waits for response from the FTP server as shown in figure 1.

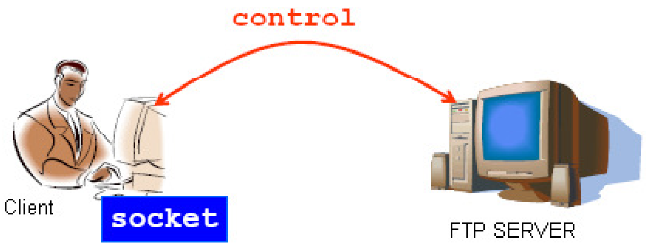


Figure 1 [3]

But the FTP server does not know about the port number. So after getting response from server, the client tells its IP address and port number to the server as illustrated in figure 2.

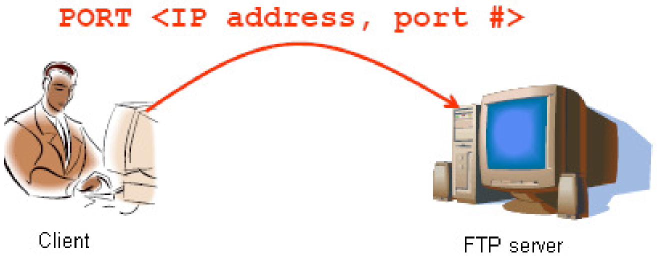


Figure 2 [3]

After getting the information from the client, the server initiates the connection. The two ports are mapped and the connection is established. Now the data flows along the second connection. Figure 3 illustrates this process.

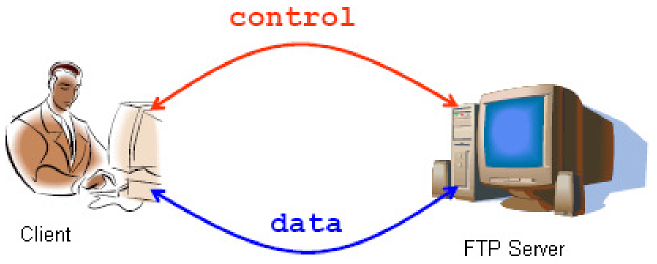


Figure 3 [3]

The following factors could have an impact on the FTP server response time:

- Server software/hardware
- DNS response time
- Link capacity
- Congestion on link
- Location of the server

If we improve any one of the above factors we can decrease the file download time and RTT.A file download time depends on the Round Trip Time (RTT). If we have a shorter RTT then the file download time will be minimal.

3.1 Planning for the Experiment

As already mentioned, this experiment is performed on live FTP servers. We have a connection from the internet. We used a laptop with the following specification:

- Processor: 1.6 GHz
- RAM: 512 MB
- Cache: 128 MB
- Ethernet Card: 10/100 Mbps
- Location: The Blekinge Institute of Technology Ronneby, Sweden

We have used the following tools:

- PING
- Gnuplot
- Perl

We have to investigate the relationship between round-trip-time and file download time of FTP servers. We have to download approximately 1 Mega Byte (MB) file from several servers which are located at different regions of the world.

3.2 Performing the Experiment

First of all we selected a list of FTP servers. These servers are located at different regions of the world. The list consists of the following servers:

- Aland; ftp.ax.kernel.org
- Tajikistan; ftp.tj.kernel.org
- United Kingdom; ftp.uk.kernel.org
- Denmark; ftp.dk.kernel.org

Then we used the PING tool to determine the round-trip time (RTT) for each server. We pinged all the servers 10 times each. We calculated the average values for each server as given in table 3.1.

Table 3.1

Server Name	**Average RTT in ms**
Aland	8.758
Tajikistan	29.922
United Kingdom	30.459
Denmark	31.292

We noticed that almost all servers have the same round- trip time except the Aland server which has average round-trip time (RTT) 8.758 milliseconds.

Then we start downloading approximately 1 MB file from all servers 30 times each.

We downloaded the same file 30 times and noticed the download time for each server. We calculated the average download time for each server.

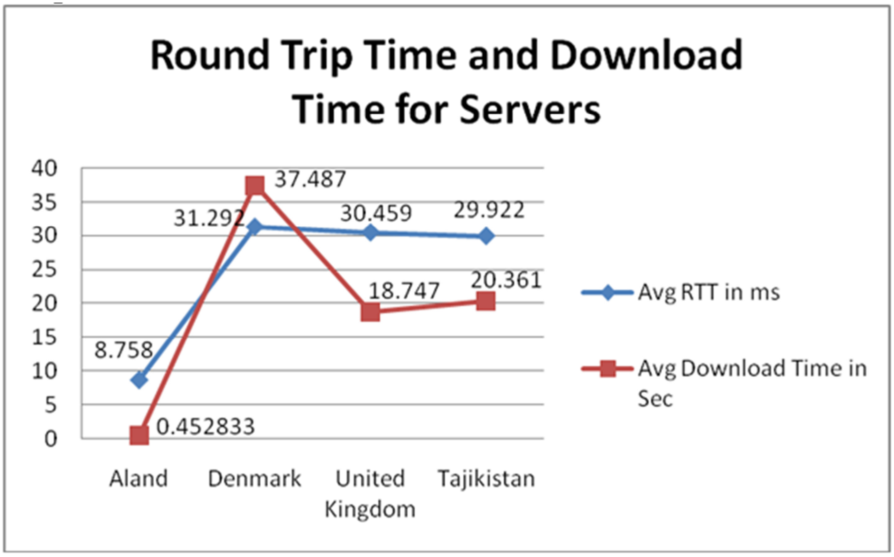
The summarized results for the round trip time and the download time for the file from all servers are given in the following table:

Table 3.2

Server Name	Average RTT in milliseconds	Average Download time in Seconds
Aland	8.758	0.4528
Denmark	31.292	37.487
United Kingdom	30.459	18.747
Tajikistan	29.922	20.361

Table 3.2 shows that the Aland server has less round-trip time than other servers that is why it has minimum download time. For Denmark, United Kingdom and Tajikistan servers, they do not have much difference in round-trip time but they have variation in the file download time. This variation may be possible due to factors like link bandwidth, congestion on link, number of requests to the server, location of the server and system configuration. The following graph can illustrate the above summary clearly:

Graph A

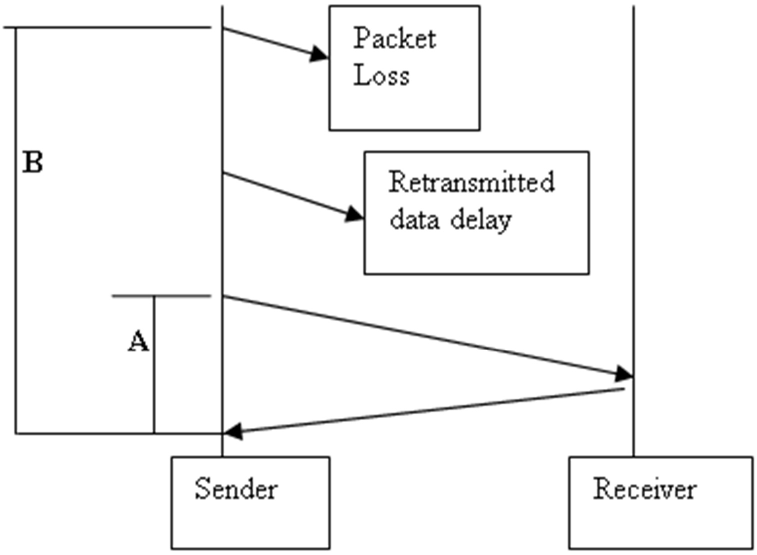


The above graph indicates the possible relationship between round-trip time and download time for all the servers. We observed that the server with less round-trip time has minimum download time and vice versa.

The matter is not finish here; another question is still unclear that how we can reduce the round trip time. To answer this question, we use algorithm know as Karn's Algorithm **.**This algorithm basically used to solve the problem of accurately and perfectly measurement of RTT.Below we are going to discuss Karn's Algorithm by which we will able to analyze that how karn's algorithm influence on RTT, even then when packets loss increases. As you know that when packet loss increases the RTT also increases .Let analyze the karn's algorithm

4. Karn's Algorithm

Karn's algorithm is used to improve the accuracy and efficiency of the RTT measurement .This algorithm works well when packets loss. As you know when packet loss increases the round trip time also increase .Let we take two type of RTT measurement scenario with packet loss



Case A:

Associating the ACK with the original transmission. So it can lead to longer estimates for RTT. In this case the RTT estimate can eventually grow arbitrarily large

Case B:

Associating the ACK with the retransmitted packet for RTT measurement .So it can lead to shorter estimates. For example, if network delay increases sharply, it will cause retransmissions. Since the ACKS were not lost, So associating the ACKS with the retransmitted packet will lead to shorter RTT estimates. In this case the another question arise should we use this case B. I will say yes because karn's algorithm present a simple solution for these above cases .For removing ambiguity we need two rules of karn's algorithm

Rule 1:

Ignore the RTT Sample from a retransmitted packet, by the help of this rule we can remove the ambiguity from RTT measurement .For example what will happen when the network delay increases sharply ? .The first packet that sees the higher delay will have a smaller time out ,so causing retransmission of packet .If ignore the RTT sample ,RTT will never increases

Rule 2

RTO should be doubled after retransmission (This is called Exponential Back Off) .So Continue back off till you get an ACK for original transmission .If the increase is large, the RTO may oscillate between the backed-off value necessary to avoid an unnecessary retransmission and the value calculated from SRTT. However, the SRTT will converge to the correct value, and unnecessary retransmission will stop. How quickly the SRTT converges to the new round-trip time depends on the back-off algorithm and the SRTT smoothing algorithm, but typically this convergence is quite fast. To prevent unnecessary retransmissions, the RTO must be greater than the new round-trip time [12]

A TCP implementation using Karn's algorithm and Mills' nonlinear filter has been in heavy use on perhaps the worst medium ever used to pass IP datagrams: amateur packet radio [13]. Despite packet loss rates often exceeding 50%, SRTT values remain quite stable, changing only in response to true changes in round-trip time. Packets lost due to noise leave the SRTT unaffected.

5. Conclusions and Future Work

This paper examines some of the factors that are involved in the downloading time of a file. In first experiment we have some limitations such as the link bandwidth from a client to the server, the server configuration and server location. But still we have performed this experiment to point out the possible factors which affect the overall performance. From clients' perspective, they want that the FTP server should respond quickly. The file should be downloaded immediately. They want to have knowledge about the best server. Generally, it is difficult to point out the best server prior because each time the server will respond differently to a request from the same client. In our experiment, we have tried to conclude the possible relationship between round-trip-time and download time of an FTP server. There might be some deficiencies such as to perform this experiment with different file size and at different time.

Our empirical study revealed that lesser Round trip time results in a minimum download time of a file from an FTP server. Hence, given best configuration for the server and client, a high bandwidth link without any congestion and minimum distance between the server and client, round trip time is a factor that directly affects the total download time for any ftp transaction. We observed that the server with less round-trip time has minimum download time.

At the end we used the Karn's algorithm to improve the accuracy and efficeny of RTT measurement. And we conclude that if the packet loss exceeded to 50% the SRTT value remain stable and leave it unaffected.

.

References

[1] International Organization for Standards, Information processing systems — Open Systems Interconnection — Connection oriented transport protocol specification. International Standard, no. 8073 ISO, Switzerland. 1986

[2] Morris, Robert J.T. Fixing timeout intervals for lost packet detection in computer communication networks.In AFIPS Conference Proceedings, 1979 National Computer Conference. AFIPS Press,Montvale, Jew Jersey

[3] Velten, David, Hinden, Robert and Sax, Jack. Reliable Data Protocol; RFC908. In ARPANET Working Group Requests for Comments, no. 908. SRI International, Menlo Park, Calif., July 1984

[4] Postel, J., ed. Internet Protocol; RFC791. In ARPANET Working Group Requests for Comments,no.791. SRI International, Menlo Park, Calif., Sep. 1981.

[5] Postel, J., ed. Internet Control Message Protocol; RFC792. In ARPANET Working Group Requests for Comments, no. 792. SRI International, Menlo Park, Calif., Sep. 1981

[6] Postel, Jon, ed. Transmission Control Protocol; RFC793. In ARPANET Working Group Requests for Comments, no. 793. SRI International, Menlo Park, Calif., Sep. 1981

[7] Nagle, John. Congestion Control in IP/TCP Networks; RFC896. In ARPANET Working Group Requests for Comments, no. 896. SRI International, Menlo Park, Calif., Jan. 1984

[8] Perry, Dennis G. Congestion in the ARPANET. Letter to the TCP-IP Mailing List, October 1, 1986

[9] Jain, Raj, Divergence of Timeout Algorithms for Packet Retransmissions. In Proceedings Fifth Annual International Phoenix Conference on Computers and Communications, Scottsdale, AZ, March 26-28, 1986.

[10] Zhang, Lixia. Why TCP Timers Don't Work Well. In Proceedings of SIGCOMM '86, Association for Computing Machinery

[11] FastSoft "Transfer Time Benchmarks for Common File Downloads over Various Links Speed " Published on July 01,2007

[12] K. Phil , P. Craig "Improving Round-Trip Time Estimates in Reliable Transport Protocols", in ACM SIGCOM

[13] Karn, P. R., Price, H., Diersing, R. Packet Radio in the Amateur Service. In IEEE Journal on Selected Areas in Communications, May 1985.

[14] http://en.wikipedia.org/wiki/File_Transfer_Protocol, last Visited April 18th 2008.

[15] http://www.eecs.berkeley.edu/~binetude/course/ee122/FTP-6up.pdf, Last visited April 18th 2008

[16] A. Dell'Aera, et al., "Linux 2.4 implementation of Westwood+ TCP with rate-halving: A performance evaluation over the Internet," Institute of Electrical and Electronics Engineers Inc., Piscataway, United States, 2004, pp. 2092-2096.

[17] M. Kwon and S. Fahmy, "On TCP reaction to Explicit Congestion Notification," *Journal of High Speed Networks*, vol. 13, no. 2, 2004, pp. 123-138.