Estimation Of Download Time Of Media Segment Over TCP Connection

Bindu Kumari
Department of Computer Science
University Of New Hampshire
Durham - NH - 03824
Email: bk1044@wildcats.unh.edu

***Abstract*—The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the Transport Layer of the Internet model.TCP is used extensively by many applications available by internet, including the World Wide Web (WWW), E-mail, File Transfer Protocol, Secure Shell, peer-to-peer file sharing, and streaming media applications. In this paper,I am trying to analyze the transfer time of media file.**

I. Question

What is the download time of media file at client side and sending time at the server side.

II. Introduction

TCP is a connectionoriented transport protocol that sends data as an unstructured stream of bytes. By using sequence numbers and acknowledgment messages, TCP can provide a sending node with delivery information about packets transmitted to a destination node. Where data has been lost in transit from source to destination, TCP can retransmit the data until either a timeout condition is reached or until successful delivery has been achieved. TCP can also recognize duplicate messages and will discard them appropriately. If the sending computer is transmitting too fast for the receiving computer, TCP can employ flow control mechanisms to slow data transfer. TCP can also communicates delivery information to the upperlayer protocols and applications it supports. All these characteristics makes TCP an endtoend reliable transport protocol.

III. Performance Outline Model

The round-trip time (RTT)/ is the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgment of that signal to be received. This time delay therefore consists of the propagation times between the two points of a signal. The RTT is originally estimated in TCP by:

$$RTT = (\alpha.OldRTT) + ((1 - \alpha).NewRTT) \qquad (1)$$

Where α is constant weighting factor ($0 \leq \alpha \leq 1$) Below formula is being used for calculation of response time at client side and server side.

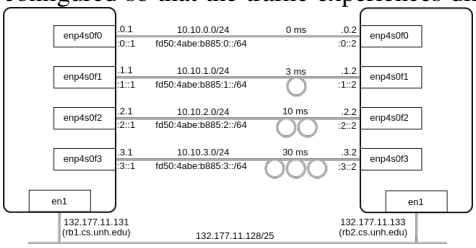
$$Mean(population) = \mu = \frac{\sum_{i=1}^{k} f_i x_i}{n}$$

$$StandardDeviation(population) = \sigma = \sqrt{\sum_{i=1}^{k} \frac{f_i (x_i - \mu)^2}{n}}$$

$$Variance(population) = \sigma^2 = \sum_{i=1}^{k} \frac{f_i (x_i - \mu)^2}{n}$$

IV. Detailed Experiment Description

For this experiment I have used two server.They are connected via four interfaces that carry no other traffic than the one produced by the experiments.Each of the link is configured so that the traffic experiences different link delays.



Before the sending device and the receiving device start the exchange of data, both devices need to be synchronized.During the TCP initialization process, the sending device and the receiving device exchange a few control packets for synchronization purposes.This exchange is known as Three-way handshake. The Three-way handshake begins with the initiator sending a TCP segment with the SYN control bit flag set.TCP allows one side to establish a connection. The other side may either accept the connection or refuse it. If we consider this from application layer point of view, the side that is establishing the connection is the client and the side waiting for a connection is the server.Now connection is established and Client can send message to the server.Each side of a TCP connection has a socket which can be identified by the triple TCP, IP address, port number.This is also called a half-association. If two processes are communicating over TCP, they have a logical connection that is uniquely identifiable by the two sockets involved, that is by the combination TCP, local IP address, local port, remote IP address, remote port. TCP uses the following principle: Client sends a packet and wait for an acknowledgment from server before sending the next packet. If the acknowledgment (ACK) is not received within a certain amount of time then it retransmits the packet.This mechanism ensures reliability.

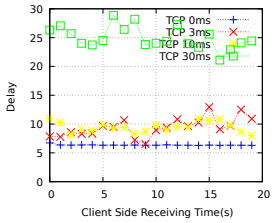
To anyalyze the transfer of media file using the TCP protocol,I created a server side application which accepts client connection at port number 5017.The server application accepts the connection and sends the requested file to the client. For this experiment I have used 2.9 MB(2,941,256 bytes) file at

server side.In file transfer process I have calculated the time it takes to accept a socket connection and file transfer.

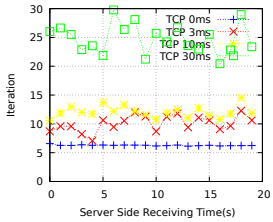
For Client Side application I have created a client application which uses TCP Protocol to connect to server application using IPV6 address and port number.Client application requests file from server application and uses input stream to download the file.Here I have calculated the time it takes to setup the socket connection and file download time.For this experiment I have used 20 iteration and calculated mean and standard deviation.

V. Result

After several iterations,I observed file transfer time at server application and file download time at client application is similar. The overall file transfer time for each iteration is not same because TCP waits for acknowledgement to send next packet which is calculated by TCP window size and round trip time. TCP stops the flow of data until previous packets are successfully transferred . In TCP, latency has a profound effect . Following graph shows the download time with different latency such as 0ms,3ms,10ms and 30 ms sets at four different interfaces.



Following graph shows the sending time at server side with 0ms,3ms,10ms and 30 ms delays.



Following Table shows the time taken at server side and client side with 30 ms delay and we can clearly see that sending time and receiving time is almost similar.

Server Send Time	Client Receiving Time
26.0178 s	26.3204 s
26.6478 s	27.0473 s
25.5189 s	25.6812 s
22.8821 s	23.01563 s
23.5551 s	23.7344 s
21.8674 s	22.4985 s
29.7754 s	29.8674 s
26.3693 s	26.4298 s
28.1057 s	28.1874 s

VI. Evaluation

TCP is sophisticated in that it comprises mechanisms such as reliable transmission, in-order delivery, congestion control, and flow control. In the above graph I analyze the some ups and downs in receiving time this can be the reason of retransmission or in-order delivery can cause unnecessary transmission latencies. The TCP recovers from data that is damaged, lost, duplicated, or delivered out of order by the internet communication system. This is achieved by assigning a sequence number to each octet transmitted, and requiring a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. At the receiver, the sequence numbers are used to correctly order segments that may be received out of order and to eliminate duplicates. Damage is handled by adding a checksum to each segment transmitted, checking it at the receiver, and discarding damaged segments. This can be the reason of taking more time in some iteration.TCP adopts a more conservative approach by starting with a modest amount of data that has a high probability of successful transmission, and then probing the network with increasing amounts of data for as long as the network does not show signs of congestion. When congestion is experienced, the sending rate is dropped and the probing for additional capacity is resumed.The dynamic operation of the window is a critical component of TCP performance for volume transfer.

VII. Conclusion

TCP is not a highly efficient protocol for the transmission of interactive traffic. TCP is not a predictive protocol. It is an adaptive protocol that attempts to operate the network at the point of greatest efficiency.TCP is optimized for accurate delivery rather than timely delivery and can incur relatively long delays (on the order of seconds) while waiting for out-of-order messages or re-transmissions of lost messages .Therefore, it is not particularly suitable for real-time applications such as Voice over IP.

References

[1] http://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-5/ipj-archive/article09186a00800c8417.html
[2] https://en.wikipedia.org/wiki/TransmissionControlProtocol
[3] https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html
[4] http://security.stackexchange.com/questions/56389/ssl-certificate-framework-101-how-does-the-browser-actually-verify-the-validity
[5] http://knowpapa.com/sd-freq/
[6] https://en.wikibooks.org/wiki/Communication-Networks/TCPandUDP-Protocols
[7] http://cnp3book.info.ucl.ac.be/2nd/html/protocols/udp.html