Latency Analysis Of Simple Request/Response Interaction Using Various Protocols

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

***Abstract*—The goal of this paper is to explore latency of a simple request/response interaction implemented using various protocols such as Transport layer protocol (UDP,TCP) and Application layer protocol(HTTP,HTTPS) running over links with different latencies. The system consists of a client requesting a simple piece of information from a server, the server responds with the information, the client receives it, and measures, as precisely as possible, the time the entire transaction took. This paper seeks to reflect a comparative analysis among four different protocols. To facilitate my experiment I have used two servers.They are connected via four different interfaces that carry no other traffic than the one produced by the experiments.**

I. Question

What is the latency of simple request response interaction using various protocols running over links with different latency. Explain the reason behind it.

II. Introduction

A. UDP

UDP is a minimal message-oriented transport layer protocol.UDP provides no guarantees to the upper layer protocol for message delivery and the UDP layer retains no state of UDP messages once sent. For this reason, UDP sometimes is referred to as Unreliable Datagram Protocol.It is transaction-oriented, suitable for simple query-response protocols such as the Domain Name System or the Network Time Protocol.

B. TCP

The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite.TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating by an IP network. Major Internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP.

C. HTTP

The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems.HTTP functions as a request response protocol in the clientserver computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

D. HTTPS

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between your browser and the website that you are connected to. The 'S' at the end of HTTPS stands for 'Secure'. It means all communications between your browser and the website are encrypted.

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 and comparision among various protocols

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

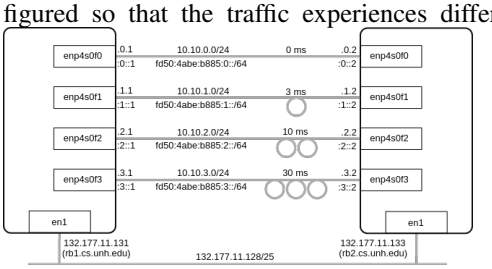
$$StandardDeviation(population) = \sigma = \sqrt{\sum_{i=1}^{k} \frac{f_i (x_i - \mu)^2}{n}}$$

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

IV. Detailed Experiment Description

For this experiment I have used two server.They are connected via four links that carry no other traffic than the one produced by the experiments.Each of the link is con-

figured so that the traffic experiences different link delays.



A. UDP Protocol Environment

UDP Client and Server Experiment: For UDP I created both client and server program .I created a server socket with port number 9990.I Created two byte array to send and receive data.Socket will receive data and find out its ip address to reply.In UDP client in order to calculate average time performance I have used 100 iteration.In UDP client side program I created client socket.Obtained IP address of the host then recorded time and sent the packet to host. client received the data from server then again recorded time in order to find out the server response. then Calculated difference of the time and closed the connection. This process repeats almost 100 times and then calculated average response time of the server.There are four interfaces with latency 0ms,3ms,10ms,and 30ms.I have run my client side and server side program with all mentioned latencies manually.Every time I used different IPV6 address to connect to the server.

B. TCP Protocol Environment

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 can be characterized by the following facilities it provides for the applications using it. A simple TCP might uses the following principle: client send a packet and then wait for an acknowledgment from the receiver before sending the next packet. If the ACK is not received within a certain amount of time, retransmit the packet. While this mechanism ensures reliability, it only uses a part of the available network bandwidth.

For TCP server side ,created a server socket with port no 9451.server accepted the connection.Obtained client address and sent required data to the client.

TCP Client Side:Created a socket with Host IP address and the port number.sent the message.received the data from socket input stream and calculated response time,then closed input stream and socket .This process repeats almost 100 times on each interface.After reasonable no of iteration calculated average response time of the server and variance and standard deviation.Every time I used different IPV6 address to connect to the server.

C. HTTP Protocol Environment

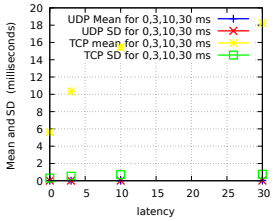
Created URL object which is IPV6 address.Recorded time then open connection HTTP default method is Get.It gets the response data from input stream,then Closed the stream and closed the connection.Performed this operation almost 100 times and the calculated mean,standard deviation and variance for each interface with different latency.

D. HTTPS Protocol Environment

I created RSA private key .Trust manager verify client certificate and server certificate,then created a SSL object and kept all the host verifier name,then recorded time and created URL object in order to get response from the server.Sent get request over the server .Received the output and recorded end time and the closed the connection.In order to get correct performance measure , performed experiment on different interfaces.

V. Result

After iterating over many times UDP response time was less as compared to TCP .UDP is fastest among all the protocols.The reason behind this is UDP does not acknowlede Packet (ACK) that permits a continuous packet stream,instead of TCP that acknowledges a set of packets ,calculated by using TCP window size and round trip time. TCP stops the flow of data until previous packets are successfully transferred.UDP is relatively immune to latency.Latency is completely independent.But in TCP Latency has a profound effect. Below graph shows the comparative analysis of UDP and TCP mean and standard deviation with 0ms,3ms,10ms and 30 ms delays.



The above result shows that UDP mean and TCP mean has significant difference.Now another diagram shows the performance of HTTP and HTTPS over different delays and from the diagram it can be seen that HTTPS takes longer time

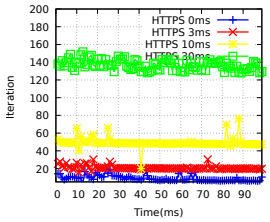
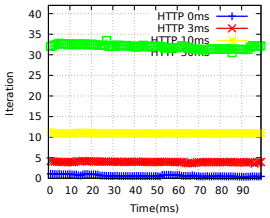


Fig. 1.

than HTTP because it takes time to authenticate the user by verifying its certificate.



HTTPS is taking extra time due to SSL.SSL hanshaking is the major cost of HTTPS.Every Protocol shows linear trendline.

VI. Evaluation

In short sessions that handshaking time will overwhelm the performance.Servers that are heavy on dynamic content will tend to be impacted less by HTTPS because the time spent encrypting (SSL overhead) is insignificant as compared to content generation time.

VII. Conclusion

UDP is faster that TCP.TCP is slower because of three way handshake.With certain delay Similar protocol gives the result in increasing order(TCP (0ms)¡TCP(3ms)¡TCp(10ms)¡TCp(30ms)and same applies to all. It is very difficult to find exact response time as response-time always changes.With Static content Http provides better performance than HTTPS. HTTPS response-time is bigger as compared to HTTP due to SSL hanshaking. For HTTPS The latency can be mitigated to some extent by:Ensuring that your server is using HTTP keepalives - this allows the client to reuse SSL sessions, which avoids the need for another handshake.Reducing the number of requests to as few as possible - by combining resources where possible and encouraging client-side caching.

References

[1] http://www.skullbox.net/tcpudp.phpcontent...
[2] https://systembash.com/a-simple-java-tcp-server-and-tcp-client
[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] http://stackoverflow.com/questions/188266/how-are-ssl-certificates-verified
[7] https://en.wikibooks.org/wiki/Communication-Networks/TCPandUDP-Protocols
[8] http://cnp3book.info.ucl.ac.be/2nd/html/protocols/udp.html
[9] https://oeis.org/wiki/ListofLaTeXmathematicalsymbols
[10] http://knowpapa.com/sd-freq/