Exploration of Latencies and Performance of Network Protocols over Multiple Links with Distinctive Latencies

Delay measurement through request-response interactions implemented using TCP, UDP, HTTP and HTTPS protocols

Ajesh Vijayaragavan
M.S Computer Science
University of New Hampshire
avv1004@wildcats.unh.edu

***Abstract*—Network performance and its quality is measured through various parameters like Bandwidth, Latency, Throughput, Error Rate which determine the quality of network. This paper as the aforementioned title points, focuses primarily on the Exploration of Latency. The experiment is about measuring the performance of different protocols based on the latency. The protocols compared are (TCP) Transmission Control Protocol, (UDP) User Datagram Protocol, (HTTP) Hypertext Transfer Protocol and (HTTPS) HTTP Secure. These protocols are tested on two isolated servers which does not consist of any traffic other than the one generated by the programs which implement the aforementioned protocols. The main ideology is that amongst all the protocols being tested User Datagram Protocol will be the one with the lowest delay as it uses the most reliable transmission model with minimum protocol mechanism.**

I. Introduction

While high quality of Network service is preferred mostly, Low Bandwidth and High Latency are avoided the most. But the question arises that which one is more crippling? Bandwidth is just one such entity that a person perceives as the speed of a network. Latency is another element that contributes to network speed. The term 'Latency' refers to several kinds of delays which typically occur in the processing of network data.

The experiment is performed to measure the performance (i.e. latency in the following experiment) of different network protocols tested include TCP, UDP, HTTP and HTTPS. For the purpose of this experiment two Linux servers were used both have 4 interfaces each named enp4s0f0, enp4s0f1, enp4s0f2, enp4sof3. The following interfaces are configured to induce an inherent delay.

The following table reports on how much of an inherent delay is present in each interface.

Table 1.1: Interface and Inherent delay

Interface	Delay (ms)
enp4s0f0	0
enp4s0f1	3
enp4s0f2	10
enp4sof3	30

Apart from the delays that naturally occur in the network the aforementioned delays add up to the naturally occurring delay further increasing the latency of a request – response operation when the two servers communicate via an interface other than the enp4s0f0. The amount and the type of data that is sent and received among the client and the server is same for all the protocols.

The two Linux servers used un the experiment are for the sake of convenience are named 'RB1' and 'RB2' respectively further in the paper. For all the experiments performed the 'RB2' served as the server and 'RB1' served as the client.

II. TCP Performance over IPv6

TCP (Transmission Control Protocol) is one of the principal protocols in the TCP/IP network. The IP protocol deals only with packets whereas TCP enables two hosts to establish a connection and exchange streams of data. TCP promises delivery of data and also promises that packets will be transported in the same order in which they are sent. To perform the experiment with TCP, the client program is run on 'RB1' and server part of the code is run on the ‘RB2’. The link that the client program connects to is changed after every experiment i.e. the latencies are calculated over the links.

The following table and diagram illustrate the patterns that are observed.

Figure 2.1 : Exploration of mean latency over links

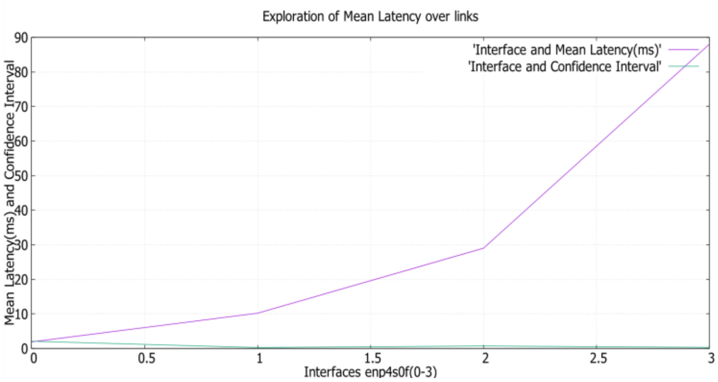


Table 2.1 : Mean Latency and Confidence Interval (TCP)

Interface	Mean Latency(ms)	Standard Deviation	Confidence Interval
enp4s0f0	1.9	3.51	±2.09
enp4s0f1	10.2	0.46	±0.27
enp4s0f2	29.0	1.21	±0.72
enp4sof3	88.0	0.43	±0.31

From the figure, it can be observer that even though there is an inherent delay present in the link, the difference in the mean latencies are significantly greater even after adding the inherent delay.

III. UDP Performance over IPV6

UDP (User Datagram Protocol) is another communications protocol used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. Both protocols send short packets of data, called datagrams.

The same experiment performed for the TCP is performed by running the client and server program on the two servers. The following table and diagram illustrate the patterns that are observed.

It can be observed the delay present is just one value greater than the inherent delay that was induced into the interfaces. This suggests that there is no observable delay other than the one induced.

IV. HTTP and HTTPS Performance over IPV6

(HTTP) Hypertext Transfer Protocol is an application protocol for distributed, hypermedia info systems. HTTP is the foundation of data communication for the WWW.

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 the communications between your browser and the website are encrypted.

Using HTTPS, the computers on a ‘code’ between them and then they scamble the messages using the same ‘code’ so that no other outside person can read them. This keeps the information safe from hackers. They use the ‘code’ on a Secure Sockets Layer (SSL), sometimes called Transport Layer Security (TLS) to send the info back and forth.

Similar to the TCP and UDP implementation ‘RB2’ served as a connection for HTTP and HTTPS server. ‘RB1’ was used to send a request with the help of a command which was included in a shell script along with the code that calculated the latencies and the mean latencies over all the interfaces.

Provided the readings below:

Figure 4.1 : HTTP – Mean Latency(ms) and Confidence Interval

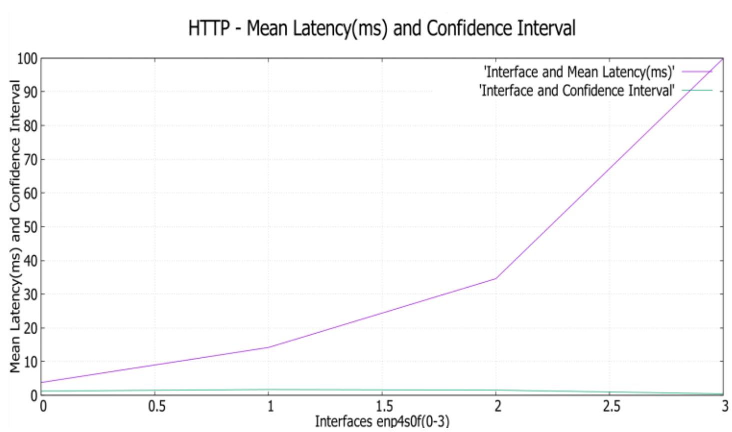


Figure 4.2 : HTTPS – Mean and Standard deviation

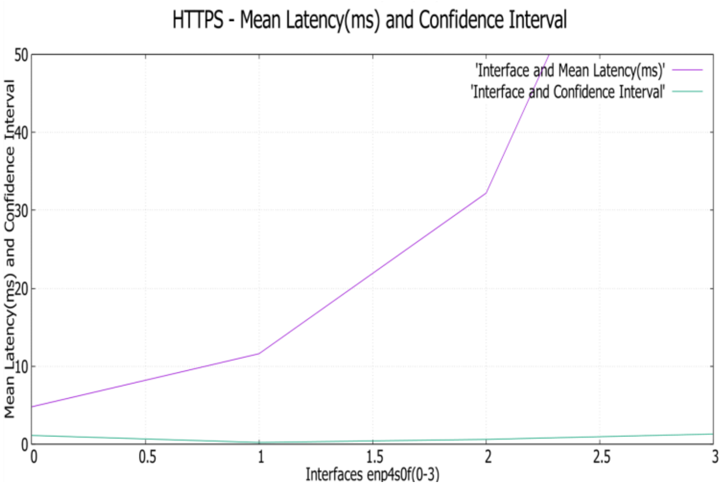


Table 4.1 : Mean Latency and Confidence Interval (HTTP)

Interface	Mean Latency(ms)	Standard Deviation	Confidence Interval
enp4s0f0	3.8	2.04	±1.26
enp4s0f1	14.2	2.78	±1.72
enp4s0f2	34.6	2.50	±1.55
enp4sof3	99.9	0.73	±0.45

Table 4.2 : Mean Latency and Confidence Interval (HTTPS)

Interface	Mean Latency(ms)	Standard Deviation	Confidence Interval
enp4s0f0	4.8	1.87	±1.12
enp4s0f1	11.6	0.43	±0.23
enp4s0f2	32.2	1.06	±0.62
enp4sof3	96.3	2.48	±1.3

Standard Deviation (S.D) = $(X-\mu)^2$

To confirm the results obtained they were compared to a real-world test performed by "GRC" – A Certificate Authority. http://www.grc.com/ is the URL where the test was performed. Performing the tests on the aforementioned website revealed that HTTPS in all the scenarios was 15% faster when compared to HTTP.

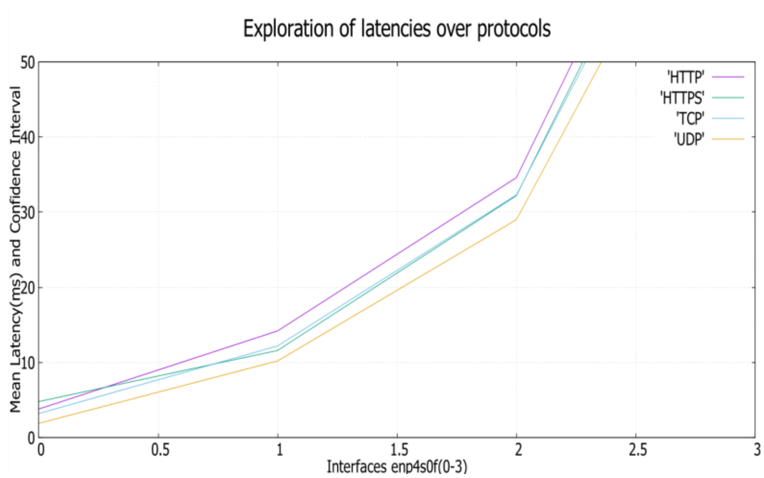
The aforementioned behavior suggesting HTTPS being faster was also observed in the experiment performed on 'RB1' and 'RB2'. It is also evident that the difference is not very much and in some cases HTTP is faster as it is observed in the enp4s0f0 link.

The equivocation of which one amongst can be addressed by considering that HTTPS requires an initial handshake which can be very slow. The actual amount of data transferred as part of handshake isn't huge (under 4Kb), but for very small requests, this can be quite a bit of an overhead. However, once the handshake is done, a fast form of symmetric encryption is used, so the overhead is minimal.

The bottom line is, making masses of short requests over HTTPS will be quite a bit sluggish than HTTP, but we transfer a lot of data in a single request, the variance will be minor.

V. Comparing the Performance of All Protocols

Figure 5.1 : Exploration of latencies over protocols



This graph is obtained after comparing the performance of TCP, UDP, HTTP and HTTPS. IT can be observed from the graph that TCP, HTTP and HTTPS protocols are almost similar when it comes to their performance but the User Datagram Protocol results show a very good performance when compared to the other protocols over different interfaces.

Significant reason why UDP results show significantly less delay is since UDP uses the most reliable transmission model with minimum protocol mechanism. Each packet is sent without delivery notes whether it reaches or not. The packet delivery and the acknowledgements are handled by the upper layers. TCP handles the connection, the successful delivery and the throughput.

Although UDP is significantly faster it limited due to its unreliability (in contrast to TCP). This, it could be useful for sensor networks and packet-oriented applications where data streaming and real-time applications are not used and it can be developed further as well.

References

[1] J. Kurose and K. Ross, *Advanced Computer Networking:* A Top-Down Approach (7th Edition).

[2] Python.org, '*Python Socket Programming Guide*'.
Available: https://docs.python.org/

[3] jfreechart.org, '*Graphs and mathematical design plot*'
Available: http://www.jfreechart.org/