Effect of Encryption Delay on TCP and UDP Transport Layer Protocols in Software Defined Networks (SDN)

AHMED MATEEN[1] (*Member, IAENG),* Qingsheng Zhu[1], Salman Afsar[2], and Saad Aleem Sahil[2]

***Abstract*— Software-Defined Networking (SDN) is an emerging technology widely adopted by data centers and enterprise networks to decouple control plane from the underlying data or forwarding plane to enable the network control to become more efficient and directly programmable and have a global view of the underlying network. In computer network, all the application traffic must be transported between end hosts by first encrypting at the transmission side and then decrypting at the receiving side. Numbers of encryption algorithms are proposed by the researcher and most of them are used to encrypt and decrypt at network level. Encryption algorithms require some sort of time to convert plain text message to cypher text and create the overhead to the speed of communication protocol stack. This research work aims to simulate and evaluate the effect of encryption delay caused by the encryption algorithm over TCP and UDP transport layer protocols under SDN network scenario. The performance matrices such as latency, throughput and end-to-end delay were considered and analyzed under SDN environment. Based on the analysis of the obtained results, we concluded that end to end delay of TCP with encryption is increases in both cases such as HTTP and FTP traffic w.r.t simulation time whereas end to end delay of UDP with encryption is decreases in both the cases such as HTTP and FTP traffic. Throughput of TCP with encryption is higher than UDP in case of HTTP traffic whereas throughput of both TCP and UDP remains same in case of FTP traffic.**

***Index Terms*— Encryption Delay, Cryptography, Software Defined Networking, Transmission Control Protocol, User Datagram Protocol.**

I. Introduction

With the advancement of computer networks, nowadays computer network and data centers becomes more featured, complex and data excessive, so with the passage of time network software needs to be upgraded according to the requirements. Conventional network architectures are unable to fulfill the requirements from carriers and end users. for a moment, capability of decision making of the legacy networks in distributed environment different network components are unable to add any new network device therefor network configuration causes errors in the management of network [1]. Software defined network (SDN) proposes network with flexibility by separating control planes from data planes. In SDN architecture, network is logically centralized at control plane whereas data plane is just responsible for packet forwarding in response to control plane. With the characteristics of flexibility, programming and manageability of the SDN, it acts as backbone for the whole network architecture [2].

Due to complex significant attraction from both in academics and in industry, SDN rapidly emerged as an new technology in the field of networks. Due to separation of control plane from the data plane, it enables the features of adding new and powerful networks [34]. There are numerous new concepts and ideas based on SDN/Openflow proposed in the field of SDN [25]. Security vitality proficiency and system virtualization for upgrading the general system execution.

Nonetheless, security is one the real zone for the SDN which requires promote improvement [22]. As expressed above, SDN is thought to be a critical future system innovation which changes the regular system models and administrations. Subsequently, security specialists (or professionals) utilized the SDN innovation to enhance the present security capacities. SDN innovation is progressively sent and acknowledged by number of associations in light of the fact that various system outlines for the SDN stage are being utilized, for example, stack adjusting, WAN administration and system checking applications [3-4].

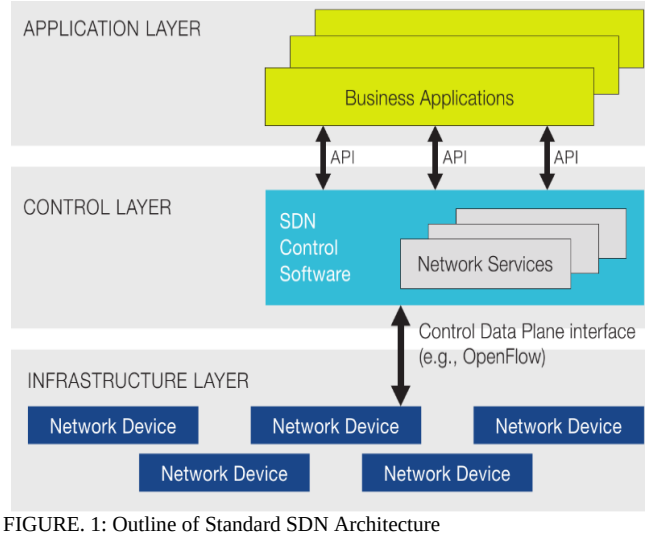


FIGURE. 1: Outline of Standard SDN Architecture

Manuscript received December 15, 2019; revised January 21, 2019.Effect of Encryption Delay on TCP and UDP Transport Layer Protocols in Software Defined Networks (SDN)
Ahmed Mateen is with Computer Science Department, Chongqing University China. e-mail: ahmedmatin@hotmail.com
Qingsheng Zhu is with Computer Science Department, Chongqing University China.
Salman Afsar is with Computer Science Department, University of Agriculture Faisalabad, Pakistan.
Saad Aleem Sahil is with Computer Science Department, University of Agriculture Faisalabad, Pakistan.

In the same manner, engineers and the creators have enthusiasm for sending security capacities, for instance, a system that distinguishes refusal of administration (DoS) assaults with SDN capacities has been proposed and a novel structure for creating security applications has been produced [5,23].

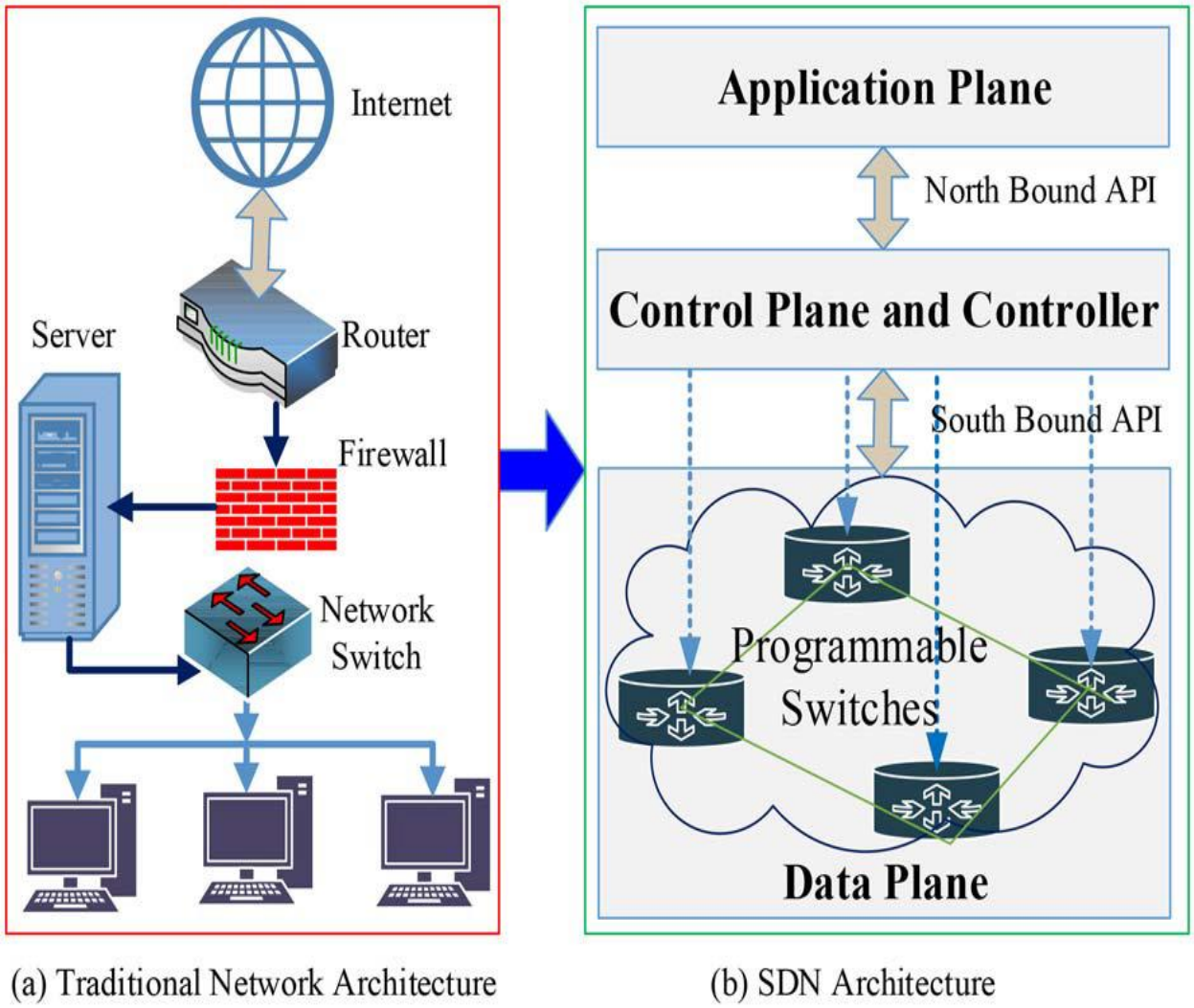


FIGURE. 2: Comparison between Traditional Network Architecture and SDN Architecture

Table 1: SDN Security Architecture

Security Threats	Description
Spoofing	An assailant may camouflage as a head or a SDN controller to expel or change delicate information from the SDN change or to acquire touchy data, for example, stream sections in the stream table.
Eavesdropping	An aggressor may listen in on streams between SDN changes to get the data about related stream, movement and gadget.
Flow table overflow	The flow table capacity bottleneck leads to potential flow table overflow
Repudiation	An administrator or a SDN controller may deny the mistaken setup which he had made

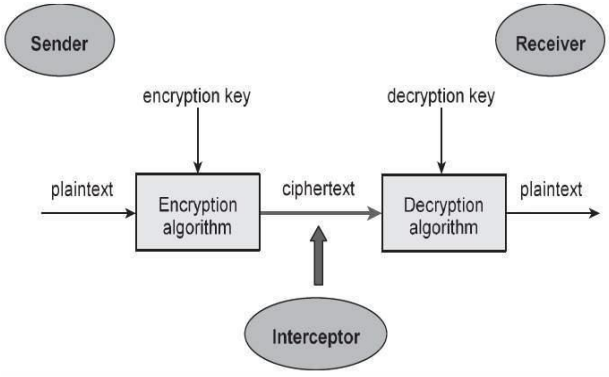


FIGURE.3: Model of Cryptography

A. Cryptography

Cryptography is an art of writing the message in secret forms. In other words, cryptography changes the original form of the message into other forms that cannot be understood [6-7].

There are certain security requirements [8]. For example:

- Authentication: It is the security feature that governs the access to authorized person / device.
- Privacy: It is the confidentiality and insurance between the receiver and the sender and no one else can read the messages.
- Integrity: It refers to the original state of the message that the message has not been changed or modified.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

- Non-repudiation: It refers to the message being sent by the sender and verifies that the message is really sent.

There are three types of the cryptography techniques that are used for the network security [9]. These techniques are categorized on the basis of the keys that are used in the working for the encryption and decryption. They are as follows:

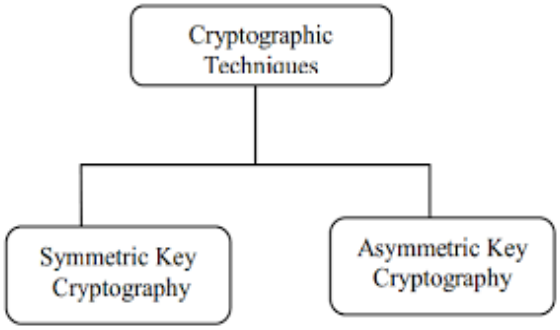


FIGURE.4: Cryptography Techniques

- Symmetric or Secret Key Cryptography: For converting the message it uses only one key.
- Asymmetric or Public Key Cryptography: For converting the message different keys are uses.
- Hash Functions: To extract the actual information it uses mathematical transformation [10,23].

II. PREVIOUS WORK

Authors proposed improved SDN constructional plan, which is suitable for the providing data. Data plan and control plan are separated between two different plans, so that the data can be extracted and encrypted. Services are denied during the data encryption and extraction [11,25].

A key management scheme for encrypting the data plane in SDN – DPKMS – has been proposed. Automating key management in SDN is seen as a necessity due to the highly dynamic nature of the architecture. As no previous work was found on including key management abilities in the OpenFlow protocol, an extension of the protocol has been formulated [12,24]. An effective concealment scheme was imposed for data center environment, which was totally differentiated from traditional network. MIC provides the effective and in design network by changing or modifying sender / receiver nodes [13,26]. Software Defined Networking facilitates the administration and control of system by exchanging the treatment of the control plane from organize gadgets to a controller. This brought together approach has a few focal points that can be profitable for ICS. Specialists have proposed a multipath steering technique where OpenFlow changes are utilized to intermittently modify the way between two has with a specific end goal to makes it harder for meddler to catch a whole correspondence [14].

They have proposed an authentication handover process for multi-domain SDN condition which is displayed and examined. The general issue of the effectiveness and security is outlined. In view of the programmable SDN remote system engineering, a proficient secure SDN multi-domain authentication handover instrument AHMMD is proposed [15,27].

They explained the issues of performance and encryption in the SDN architecture, where packet encapsulation and stateful packets are required. A new flow rate increasing with threshold value causes performance vulnerability [16]. In order to secure the F[information, they used encryption to secure the information transmitted from sender to receiver in a network. But due to separation of the control plan and data plan, inside outbreaks can be executed using encryption [17].

They proposed and demonstrated the plausibility of an approach for misusing stateful SDN information planes to completely appoint the separating rationale for activity arrangement down to the switches. We have actualized it as an OPP application and we demonstrated an intriguing use in situation where all the data required for the application rationale can be kept locally without depending on external controller [18,30].

SDN programmability and its combination with application layer and security layer will improve data center systems administration as it will give another versatile system layer. In such way, the system is turning into a self-guarding system that can modify assets to serve application needs and stretch out security to arrange edge [19,29].

In SDN network design a mechanism of detection of encryption is proposed using the deep packet filtering which allows traffic filtering using SDN/NFV mechanism. With respect to privacy, network is responsible to encrypt and filter the data in terms of user identification [20,31] .

Proposed the SDN characteristics: control plan, data plan and global network view over conventional network. Whereas SDN provides security solutions using encryption. Which is also raises the security threats for the control plan, data plan and the nodes connected to it [21,28].

III. MATHODOLOGY

A. Installation of the Network Simulator

Network simulators are devices used to mimic discrete occasions in a network which predicts the practices of a PC network. By and large the reproduced networks have substances like connections, switches, centers, applications, and so on. Once the reproduction demonstrating is finished, it is executed to investigate the execution. Directors would then be able to redo the test system to suit their requirements. Network simulators commonly accompany bolster for the most prominent conventions and networks being used today, for example, WLAN, UDP, TCP, IP, WAN, and so on.

Most test systems that are open today rely upon a GUI application like the NCTUNS while some others including NS2 are CLI based. Replicating the framework incorporates organizing the state segments like associations, switches, focuses, terminals, et cetera and besides the events like package drop rate, movement status and whatnot. The most essential yield of the generations are the take after records. Take after reports log each package and each event that occurred in the generation and are used for examination. Framework test systems can similarly give diverse instruments to energize visual examination of examples and potential burden spots. Most of the diversion is performed in

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

```
# LD_LIBRARY_PATH
OTCL_LIB=/HOME/UBUNTU/NS-ALLINONE-2.35/OTCL-1.14
NS2_LIB=/HOME/ UBUNTU /NS-ALLINONE-2.35/LIB
X11_LIB=/USR/X11R6/LIB
USR_LOCAL_LIB=/USR/LOCAL/LIB
EXPORT
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_L
IB
# TCL_LIBRARY
TCL_LIB=/HOME/UBUNTU/NS-ALLINONE-2.35/TCL8.5.10/LIBRARY
USR_LIB=/USR/LIB
EXPORT TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/HOME/UBUNTU/NS-ALLINONE-2.35/BIN:/HOME/UBUNTU/NS-ALLINONE-
2.35/TCL8.5.10/UNIX:/HOME/UBUNTU/NS-ALLINONE-2.35/TK8.5.10/UNIX
#THE ABOVE TWO LINES BEGINNING FROM XGRAPH AND ENDING WITH UNIX SHOULD
COME ON THE SAME LINE
NS=/HOME/UBUNTU/NS-ALLINONE-2.35/NS-2.35/
NAM=/HOME/UBUNTU/NS-ALLINONE-2.35/NAM-1.15/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

FIGURE 5: Installation path setting

discrete times where events that are in the line are set up in an enduring movement in a demand. Since simulation is a complex task, we can't ensure that every one of the simulators can give correct or exact outcomes to all the distinctive sort of data. Cases of network simulators are: ns, NCTUNS, NetSim, and so forth.

B. Setting the Environment Path

The last advice is to tell the framework, where the records for ns2 are introduced or exhibited. To do that, we added these lines.

Once the changes have been made, save the file and restart the system.

Running NS 2

Once the system has restarted, open a terminal and start ns2 by using the following command:

- ns

C. Visualization Tools

NAM (Network Animator)

Nam is a Tcl/TK based activity instrument for survey arrange reproduction follows and certifiable parcel follows. It underpins topology format, parcel level movement, and different information review apparatuses. Nam started at LBL. It has advanced generously finished the previous couple of years. The Nam advancement exertion was a continuous joint effort with the VINT venture. As of now, it is being created as an open source.

D.Xgraph

XGRAPH is a broadly useful x-y information plotter with intuitive catches for panning, zooming, printing, and choosing show alternatives. It will plot information from any number of documents on a similar diagram and can deal with boundless informational index sizes and any number of information records. XGRAPH produces wysiwyg PostScript, PDF, PPTX, and ODP yield for printing printed copies, putting away, or potentially sharing plotted outcomes, and for bringing in, charts specifically into word-processors for making documentation, reports, and view-diagrams. XGRAPH incorporates the capacity to determine plotting hues for multi-shading plots, and in addition line-thickness. It can utilize any section of a multi-segment document as ordinate and abscissa pivot. It additionally bolsters programmed resizing of its window. You can zoom intuitively into any locale of a diagram by just dragging a crate around the area with your mouse.

E. Installation of MiniNet

Here are some installation commands for the Linux operating system to install the MiniNet on the system locally.

- git clone git://github.com/mininet/mininet
- cd mininet
- git tag # list available versions
- git checkout -b 2.2.1 2.2.1 # or whatever version you wish to install
- cd ..
- mininet/util/install.sh [options]
- To install everything (using your home directory): install.sh -a
- To install everything (using another directory for build): install.sh -s mydir -a
- To install Mininet + user switch + OVS (using your home dir): install.sh -nfv
- To install Mininet + user switch + OVS (using another dir:) install.sh -s mydir -nfv
- install.sh -h
- sudo mn --test pingall

To install from the packages directly, use the following commands:

- sudo rm -rf /usr/local/bin/mn /usr/local/bin/mnexec \

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

/usr/local/lib/python*/*/*mininet* \
/usr/local/bin/ovs-* /usr/local/sbin/ovs-*

- lsb_release -a
- Mininet 2.1.0 on Ubuntu 14.10: sudo apt-get install mininet
- Mininet 2.1.0 on Ubuntu 14.04: sudo apt-get install mininet
- Mininet 2.0.0 on Ubuntu 12.04: sudo apt-get install mininet/precise-backports
- sudo service openvswitch-controller stop
- sudo update-rc.d openvswitch-controller disable
- sudo mn --test pingall

F. OPNET Network Simulation Tool

OPNET [OPNET] is the selected business trademark and the name of thing presented by OPNET Advances joining. It is a champion among the most eminent and standard business sort out test frameworks. Because it has been used for a long time in the business, it has included a noteworthy bit of the pie.

OPNET relies upon a segment called discrete event structure which suggests that the system direct can mirror by showing the events in the structure in the demand of the circumstances the customer has set up. Different leveled structure is used to form the frameworks. OPNET moreover gives programming instruments to customers to describe the bundle course of action of the tradition. The programming mechanical assemblies are in a manner required to accomplish the assignments of portraying the state change machine, describing framework appear and the technique module. OPNET is a popular test framework used as a piece of industry for arranging inventive work. The GUI interface and the programming devices are furthermore important to help the customer with building the system they require.

According to the OPNET whitepaper, OPNET ' s detailed features include:

a. Fast discrete occasion recreation engine
b. Lot of part library with source code
c. Object-oriented modeling
d. Hierarchical modeling environment
e. Scalable remote recreations support
f. 32-bit and 64-bit graphical user interface
g. Customizable remote demonstrating
h. Discrete Occasion, Half and half, and Systematic reproduction
i. 32-bit and 64-bit parallel simulation kernel
j. Grid computing support
k. Integrated, GUI-based investigating and examination
l. Open interface for coordinating outside part libraries

G.Simulation Modeling:

To evaluate the encryption delay of TCP and UDP protocols in SDN, first create a simulation environment of SDN where SDN controller reside on the control plane and SDN OpenFlow switches resides on data plane. The client stations where connected with SDN OpenFlow data plane switch to send and receive the data packets to/from each other.

H.Research Flow

TCP is connection-oriented protocol which provides reliable data delivery with the concept of packet acknowledgement, so it would be better to consider TCP for less delay sensitive traffic like FTP. Where UDP is connection less and is favorable for more delay sensitive traffic like HTTP

Since we are dealing with the FTP and HTTP encrypted traffic under SDN, so we are mainly dealing here with both type (HTTP and FTP) of traffic, along with TCP and UDP as underlying protocols with different input and output parameters. Our main considerations for both applications are:

- Effect of encryption delay on latency, throughput and end-to-end delay.
- Does encryption delay effect on packet end-to-end delay, latency and throughput?

As discussed above, quantitative research approach is used for this research in which we have compared the impact of transport layer protocols (TCP and UDP) for both FTP and HTTP encrypted traffic separately. And then results are drawn in the form of graphs and tables on the basis of output values.

Following steps have been taken to complete this research:

1. Same input data size (in bytes), with varying encryption delay (0.5ms, 1ms, 1.5ms, 2ms, 2.5ms) are sent from sender to destination through simulated network (default encryption of SDN).
2. Two main scenarios are created one for encrypted traffic with variant encryption delay and second is without encryption delay.
3. To start with, Scenario is additionally isolated into two situations. One is agreed to FTP encoded movement against TCP and UDP independently, first FTP information is sent for TCP and after that for UDP and Output parameter i.e. FTP reaction time for both are noted down for all encryption delays. Other is made with HTTP scrambled activity against TCP and UDP independently, first HTTP information is sent over system for TCP and after that UDP and Output parameter i.e. Jitter and end to end parcel delay are noted down for all encryption delays.

1) Network Simulation Elements

- Modelling Flexibility: ability to define the new protocols.
- Fast Modelling: it is a very important part where models become complex and number of events changes.
- Animation: it shows the fundamental elements in the simulation, where faults and errors can be removed.

Parameters and Variables:

Table 2: Simulation Parameters and Variables

List of parameters and Variables	Details
Minimum no. of nodes in each scenario	25 for each.
Encryption Delay (0.5ms, 1ms, 1.5 ms, 2ms and 2.5ms)	Encryption delay is taken for all HTTP, FTP traffic w.r.t TCP and UDP.
Download / Upload Parameters	Only FTP traffic is measuring in ms.
Traffic Received / Sent	HTTP and FTP traffic is measured
End-to-End Delay Variations	Only HTTP traffic is measured.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

2) Performance Matrices

A) Latency: A measure which is used by the networks to reverse the system by processing a message. It is an expression to calculate the time to get a packet of data from one node to another. Sometimes often known as the time required for a packet to be returned to its sender. It depends upon the speed of medium.

B) Throughput: Throughput is the number of actual bits that transmit through the network in a specific period of time, which is always less than or equal to the bandwidth allocated for any network.

C) End-to-End Delay: E2E delay or one-way delay is known as the time taken for a packet to transmit through network from sender node to receiver node, also referred as RoundTrip Time (RTT) which means only one direction from sender to receiver is measured.

Where

Dend-end= N[Dtrans+Dprop+Dproc+Dqueue]

IV. Results

A) Comparison FTP and HTTP w.r.t TCP and UDP

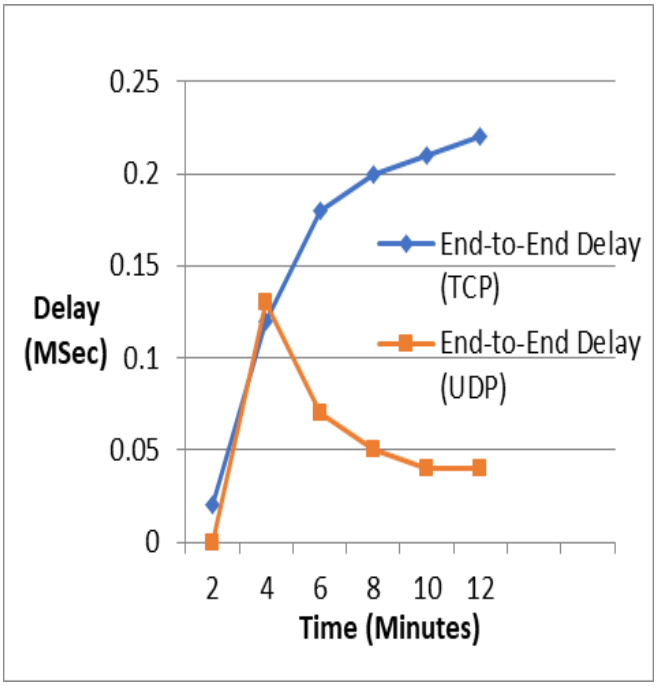
The graphs given below are the representation of FTP and HTTP traffic behavior, for all output parameters w.r.t TCP and UDP. Combine study of graph has been taken to analyze the comparative behavior of traffic for both Transport layer protocols. Traffic sent is lager for UDP which is approximatly 22 Mbps and for TCP it is observed approx. 20 Mbps but there is a minimal difference between the traffic sent through UDP and traffic sent through TCP. Graph shows the average Ethernet delay for TCP and UDP Protocol in seconds. Traffic Delay graph shows that using TCP protocol in IP cloud (Public Network) for FTP services reached up to the value of 0.22 milliseconds that is much higher as compared to UDP protocol which reaches up to 13 milliseconds. This behavior for the TCP is due to the acknowledgement behavior of the protocol. While cross checking traffic received for TCP and UDP protocols with FTP. The receiving traffic for UDP goes up to 24 Mbps and for the TCP protocol it is observed 19 Mbps. So, it is clearly verified that the traffic received at the receiving end is better for UDP. Graph shows the download Response Time behavior for both TCP and UDP protocols. The UDP protocol is giving the good response time almost 0.1 second which mean there is nothing to wait for getting response using FTP for downloading but in contrast with respect to TCP protocol the handshaking behavior and connection orientation behavior increased the response time for TCP using FTP.

A. Encryption Delay Results for FTP and HTTP traffic with TCP

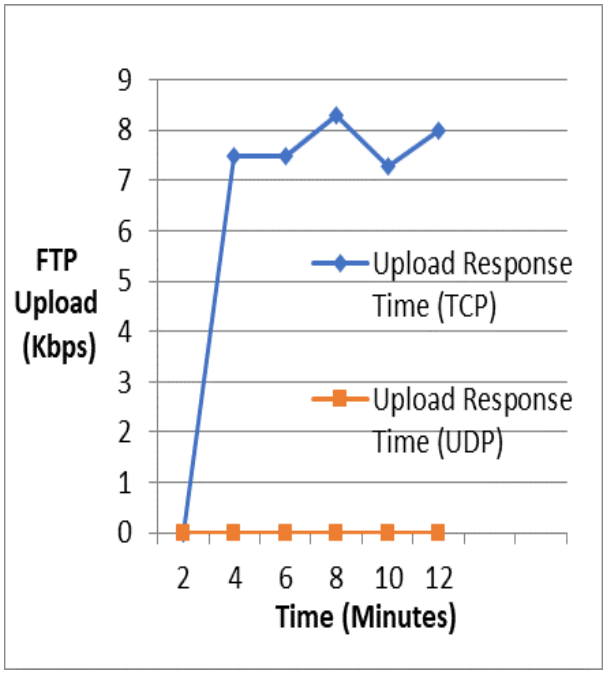
We have discussed above the behavior of UDP after applying all encryption delays on input data traffic with HTTP and FTP against our defined output parameters. Now we are going to consider the same parameters with same encryption delay but with different underlying transport layer protocol i.e. TCP.

Graph in figure 6 is the illustration of combine comparison of encryption with and without delays the output parameter the Ethernet delay. These results show almost the same nature of TCP for Ethernet delay against all encryption delays i.e. the delay in case of encryption delays is lesser than the delay in case of without encryption delay.

Comparison of TCP against FTP Download Response Time with all encryption delays is been taken in graph represented in Figures 6-10. It concluded that as encryption delay adds FTP Download Response Time decreases as in comparison taken with without encryption delay. Maximum download response time without encryption is 6.6 seconds while average FTP Download Response Time with all encryption delays is less than 0.5 milliseconds.

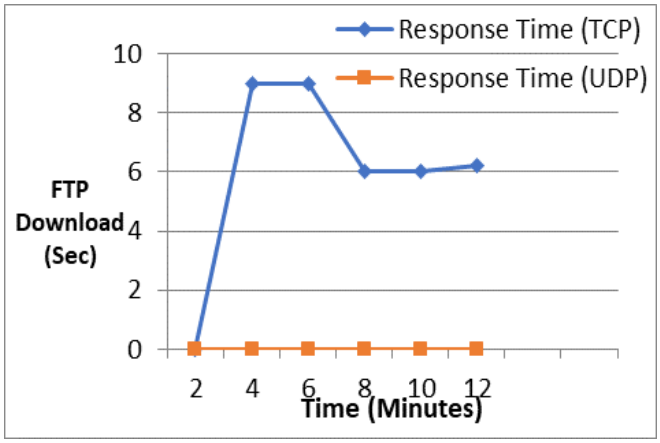


(a) End-to-End Delay (FTP)

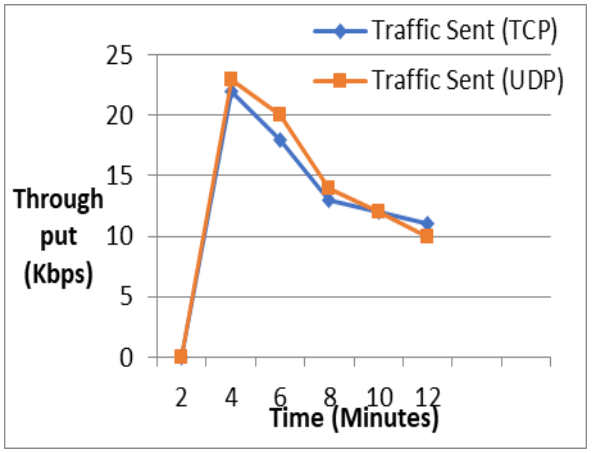


(b) FTP Upload Response Time

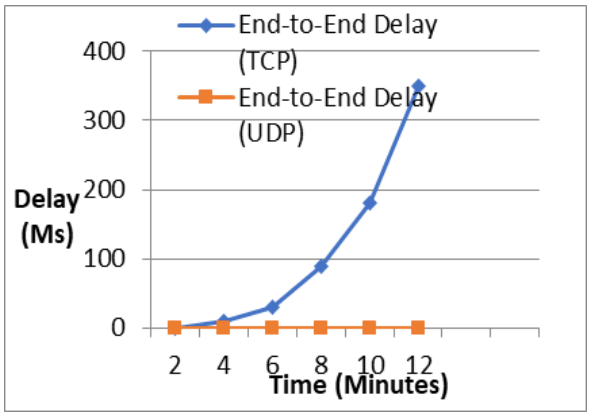
ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



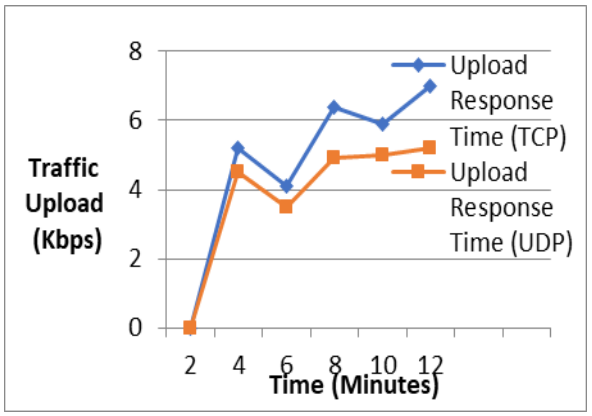
(c) FTP Download Response Time



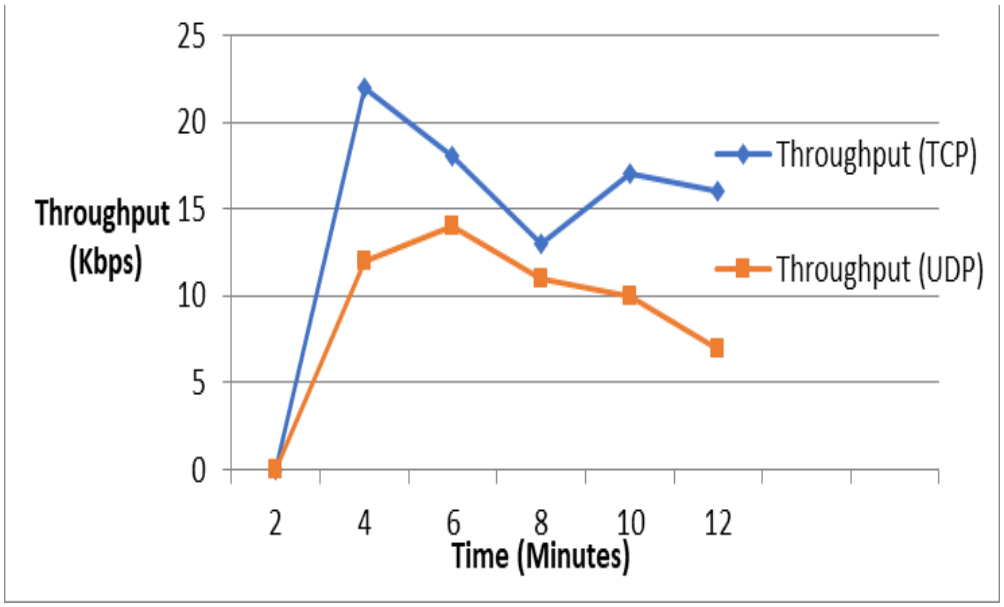
(d) FTP Throughput



(e) End-to-End Delay (HTTP)



(f) HTTP Upload Response Time



(g) HTTP Throughput

FIGURE 6: Combine Comparison of all Encryption delays

Comparison of TCP against FTP Upload Response Time with all encryption delays has been taken in graph represented in Figure 4.31. It concluded that as encryption delay adds FTP Upload Response Time decreases as in comparison taken with without encryption delay. Maximum FTP Upload Response Time without encryption is 7 seconds while average FTP Upload Response Time with all encryption delays is less than 0.5 seconds.

It concludes that as encryption delay adds up FTP/HTTP Throughput in bytes also increase with simulation time in all cases of encryption delays, as with encryption delay-1 i.e. 0.5 milliseconds it achieved maximum of 11.1KB/Sec where with encryption delay-5 i.e. 2.5 milliseconds it achieved minimum of all i.e. 11.8KB/Sec. While FTP/HTTP Throughput without any encryption is maximum of all i.e. almost 11.9 KB/Sec with TCP. This shows TCP doesn't add any delay to FTP/HTTP Throughput.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

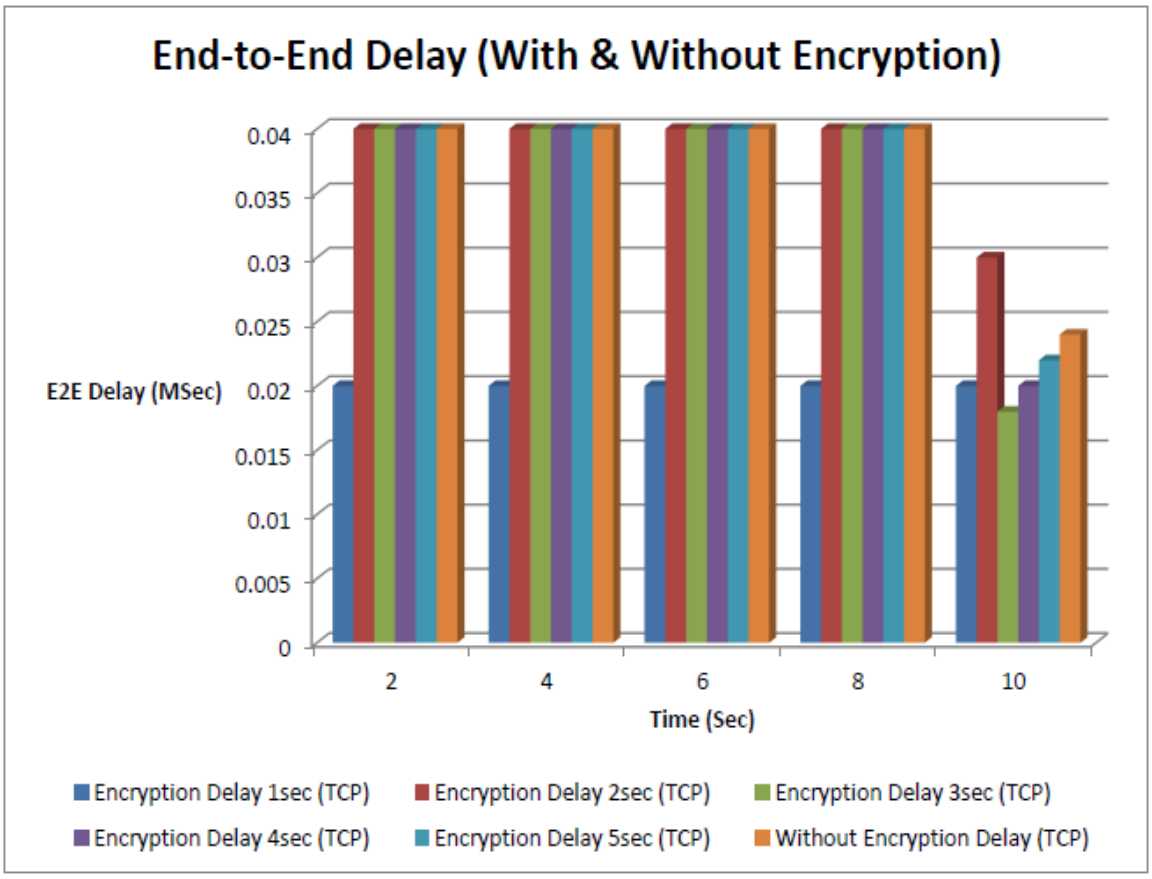


FIGURE. 7: Comparison of End-to-End delay with all encryption delays and without encryption delay for TCP.

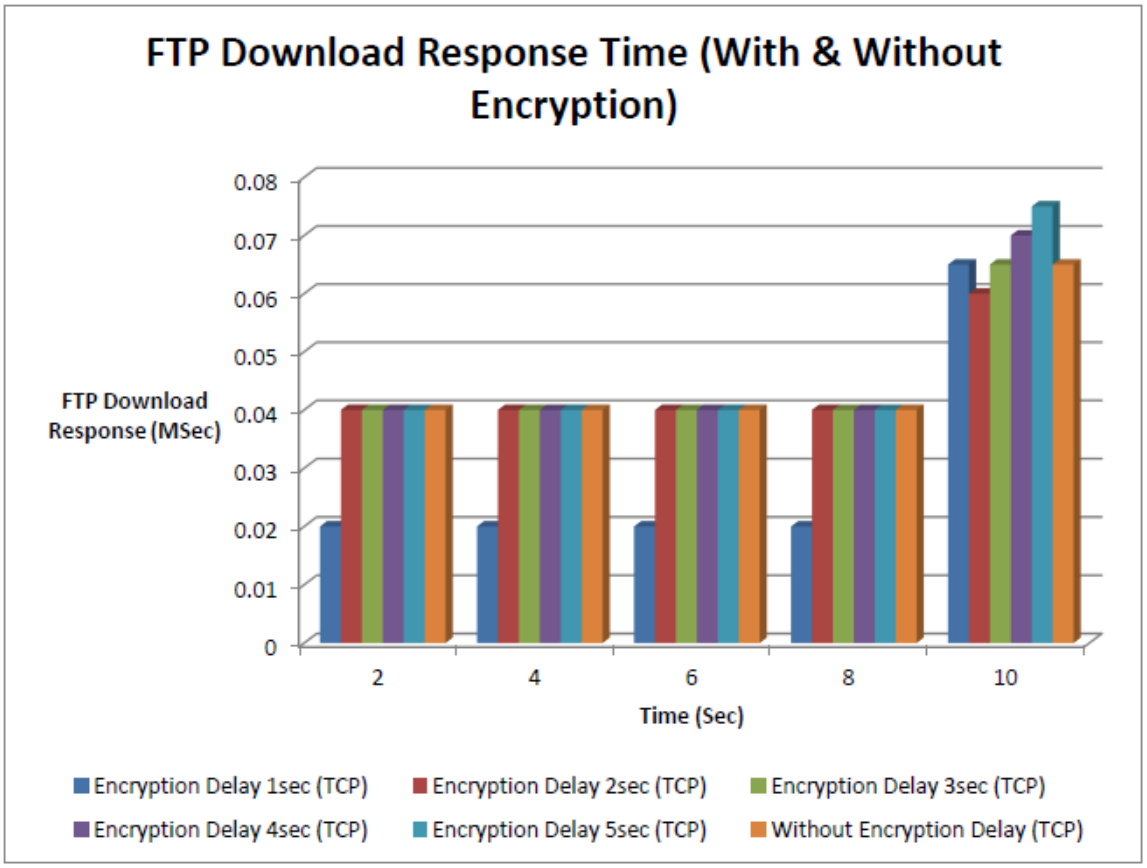


FIGURE.8: Comparison of FTP Download Response Time with all encryption delays and without encryption delay for TCP.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

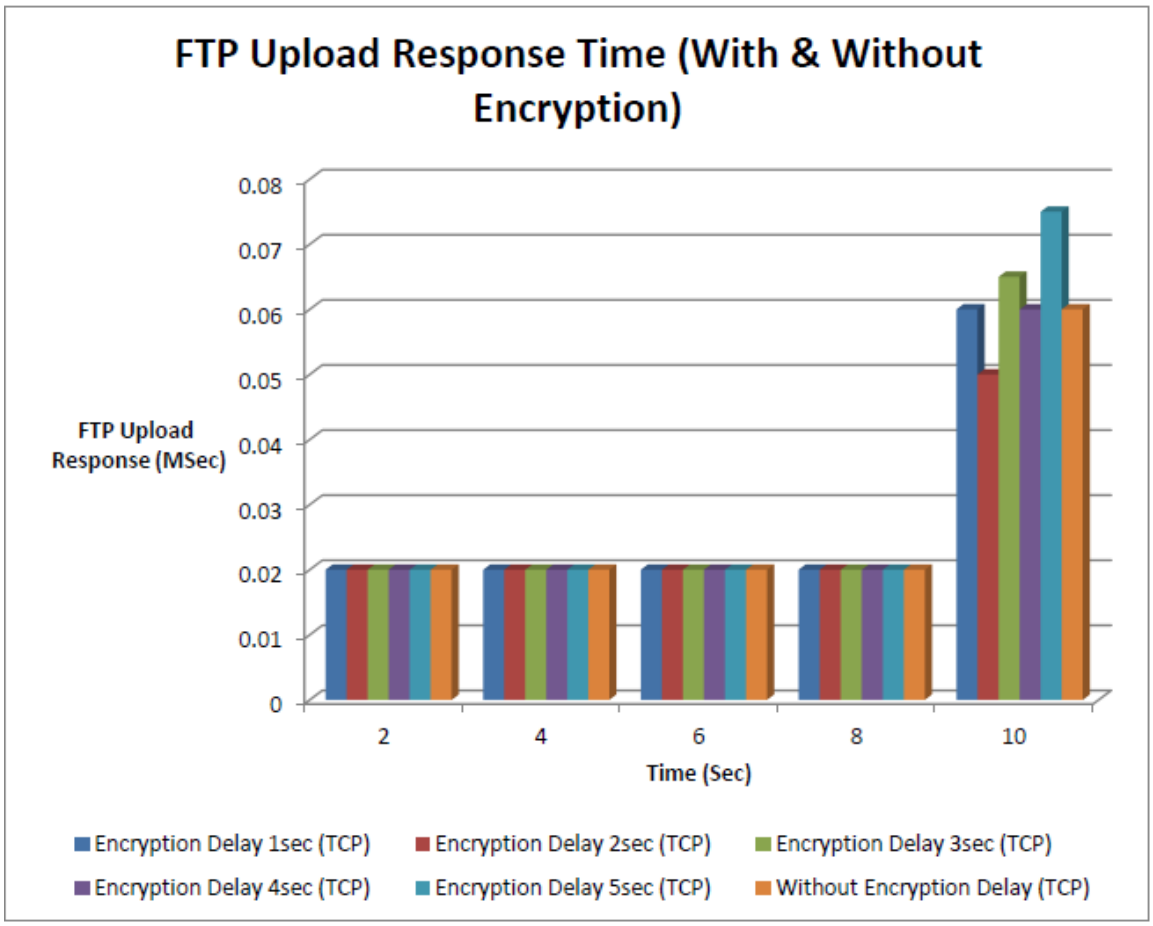


FIGURE.9: Comparison of FTP Upload Time with all encryption delays and without encryption delay for TCP.

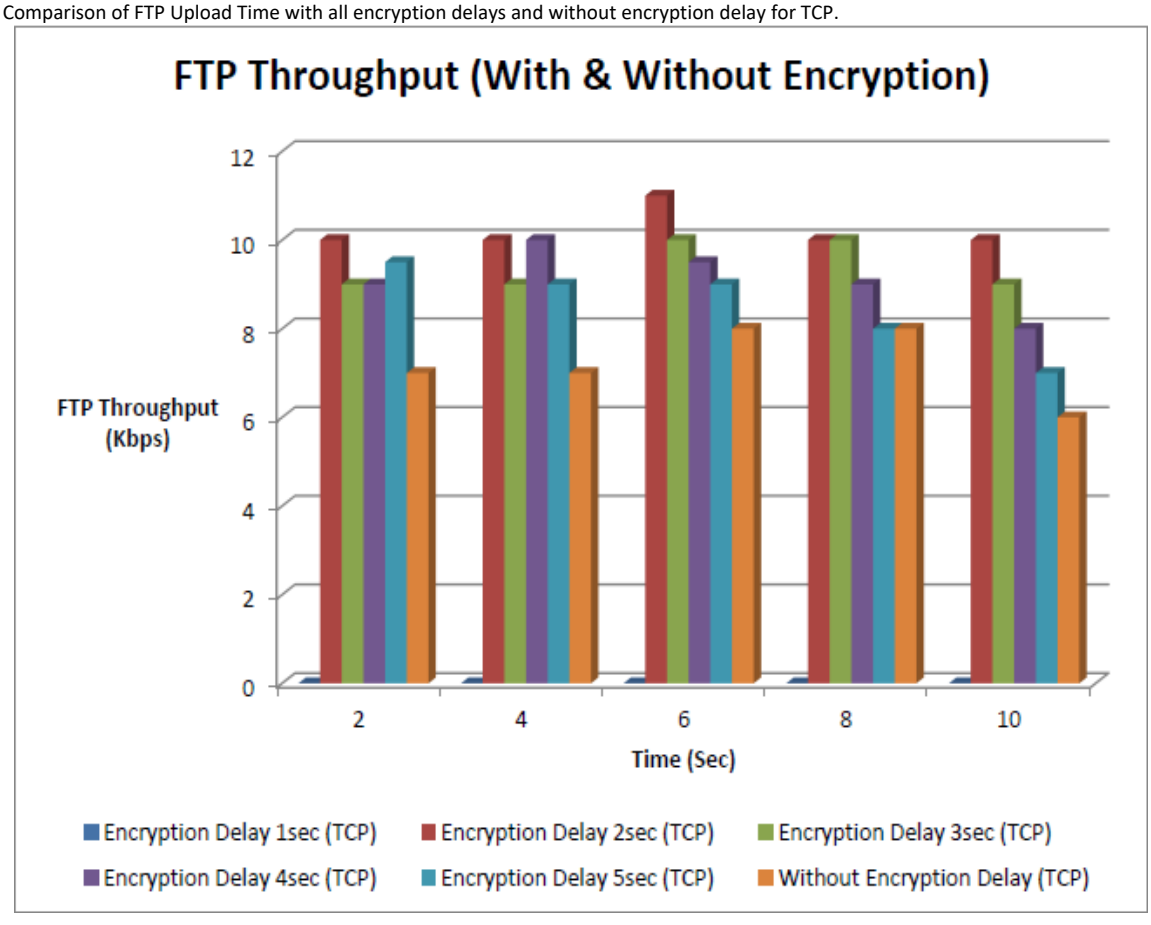


FIGURE.10: Comparison of FTP/HTTP Throughput with all encryption delays and without encryption delay for TCP.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

V. Conclusion

This research report illustrated the deep examination of two well-known transport layer protocols for FTP and HTTP traffic flow. All two selected protocols belong to transport layer of OSI model with distinction of attributes. TCP (Transmission Control Protocol) and UDP (User Datagram protocol) perform end to end transportation of data packets with their different restrictions of data transportation. A lot of network simulations have been performed using MININET network simulator to evaluate the performance of TCP and UDP protocols with different scenarios. The main objectives of this research focused over the performance of TCP and UDP for FTP and HTTP traffic in the presence of jitter/delay, Delay, Packet Delay variation, end to end Delay, traffic received, and IP packet dropped. Traffic sent over the public cloud network cloud for two protocols is affected by the jitters/delay and it is concluded that which protocol performed better for FTP and HTTP traffic in IP network. UDP was being used for the transportation of HTTP and FTP traffic over IP network. With UDP, Ethernet delay variation goes up to 54 microseconds max with encryption delay-5 of 2.5 milliseconds and with TCP it goes up to 0.5 milliseconds, so it is cleared from the analysis that in a IP cloud network when data is sent with the TCP it gives good results as compared to UDP.

References

[1]. Braadland, A. S., (2017) “Key Management for Data Plane Encryption in SDN Using Wire Guard,” Springer: 1-112

[2]. Buchanan. W. J. (2015) “RC2 Encryption and Decryption” ResearchGate: 1-8

[3]. Badotra, S., and Singh, J., (2018) “Creating Firewall in Transport Layer and Application Layer Using Software Defined Networking” Springer: 95-103

[4]. Chattarjee. A. and Das. A. K. (2018) “Secret Communication Combining Cryptography and Steganography” Progress in Advanced Computing and Intelligent Engineering, Advances in Intelligent Systems and Computing: 281-291

[5]. C. V. Neu, A. F. Zorzo, A. M. S. Orozco, and R. A. Michelin, (2016) “An approach for detecting Encrypted insider attacks on OpenFlow SDN Networks,”11th International Conference of Internet Technology and Security Transmission ICITST: 210-215

[6]. C. Yoon, T. Park, S. Lee, H. Kang, S. Shin, and Z. Zhang, (2015) “Enabling security Functions with SDN: A feasibility study,” Journal of Computer Networks: 19-35

[7]. Cui, Y., Li, S., Xing, H., Pan, W., Zhu, J. and Zheng, X., (2016) “SD-Anti-DDoS: Fast And Efficient DDoS defense in software-defined networks,” Journal of Network and Computer Applications: 65-79

[8]. D. Sanvito, D. Moro, and A. Capone, “Towards traffic classification offloading to stateful SDN Data planes,” IEEE Conference of Network Softwarization Sustain. a Hyper- Connected World en Route to 5G, NetSoft: 1-4

[9]. Dabbagh, M., Hamdoui, B., Guizani, M., and Rayes, A., (2015) “Software Defined Networking Security: Pros and Cons,” IEEE: 1-7

[10]. Darghi, T., Caponi, A., Ambrosin, M., Bianchi, G. and Conti, M., (2017) “A Survey on the Security of Stateful SDN Data Planes” IEEE COMMUNICATION SURVEYS & TUTORIALS: 1-26

[11]. G. K. Ndonda, R. Sadre, and A. Ics, (2017) “A Low-Delay SDN-based Countermeasure to Eavesdropping Attacks in Industrial Control Systems,” IEEE: 1-7

[12]. H. Hong and Z. Sun, (2017) “Applying SDN for Data Extraction and Mining : An Enhanced Architecture”, Springer: 1-3

[13]. I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, (2014) “A roadmap for traffic Engineering in software defined networks,” Journal of Computer Networks: 1-30

[14]. Khan. F., Rehman. F., Khan. S. and Kamal. S. A, (2018) “Performance Analysis of Transport Protocols for Multimedia Traffic Over Mobile Wi-Max Network Under

[15]. K. Ding, X. Wang, G. Zhang, Z. Wang, and M. Chen, (2017) “A Flow-Based Authentication Handover Mechanism for Multi-Domain SDN Mobility Environment,” Springer: 127-143

[16]. Kumar. P., Rawat. S., Choudhary. T. and Pradhan. S, (2016) “A Performance based Comparison of Various Symmetric Cryptographic Algorithms in Run-time Scenario” 5th International Conference on System Modeling & Advancement in Research Trends: 37-41

[17]. M. Hayes, B. Ng, A. Pekar, and W. K. G. Seah, (2017) “Scalable Architecture for SDN Traffic Classification,” IEEE: 1-12

[18]. M. Ammar, M. Rizk, A. Abdel-Hamid, and A. K. Aboul-Seoud, (2016) “A framework For security enhancement in SDN-based datacenters,” 8th IFIP International Conference of New Technology of Mobile Security NTMS: 3-6

[19].Modi. B. and Gupta. V., (2018) “A Novel Security Mechanism in Symmetric Cryptography Using MRGA” Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Advances in Intelligent Systems and Computing: 195-203

[20].Mathur. M. and Kesarwani. A., (2013) “COMPARISON BETWEEN DES, 3DES, RC2, RC6, BLOWFISH AND AES” Proceedings of National Conference on New Horizons in IT: 143-148

[21]. Mandal. P. C (2012) "Superiority of Blowfish Algorithm," International Journal of Advanced Research in Computer Science and Software Engineering. Nakagami Fading” Information Technology – New Generations, Advances in Intelligent Systems and Computing : 101-110

[22]. Priyadarsini, M., and Bera, P., (2018) “A New Approach for SDN Performance Enhancement” Springer: 115-129.

[23]. Papastergiou. G., Fairhurst. G., Ros. D.and Brunstrom. A, (2017) “De-Ossifying the Internet Transport Layer: A Survey and Future Perspectives” IEEE COMMUNICATIONS SURVEYS & TUTORIALS: 619:639

[24]. Rawat. D. B. and Reddy, S. R. (2017) “Software Defined Networking Architecture, Security and Energy Efficiency: A Survey” IEEE COMMUNICATIONS SURVEYS & TUTORIALS: 325-346

[25]. Sharma. S. and Gupta. Y (2017) “Study on Cryptography and Techniques” International Journal of Scientific Research in Computer Science, Engineering and Information Technology: 249-252

[26]. Shaghaghi, A., Kaafar, M. A. Buyya, R. and Jha, S. (2018) “Software-Defined Network (SDN) Data Plane Security: Issues, Solutions and Future Directions” arxive: 1-24

[27]. Shi, Y., Dai, F. and Ye, Z., (2017) “An Enhanced Security Framework of Software Defined Network Based on Attribute-based Encryption” The 2017 4th International Conference on Systems and Informatics: 1-5

[28]. Thakur. J. and Kumar. N, (2014) "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," International journal of emerging technology and advanced engineering: 6-12.

[29]. W. Li, W. Meng, and L. F. Kwok (2016). “A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures,” Journal of Network and Computer Applications: 126–139

[30] Y. H. Lin, S. H. Shen, M. H. Yang, D. N. Yang, and W. T. Chen, “Privacy-preserving deep packet filtering over encrypted traffic in software-defined networks,” 2016 IEEE International Conference of Communication, ICC: 1-7

[31]. Zhu, T., Feng, D., Wang, F., Hua, Y., Shi, Q., Liu, J., Cheng,Y., and Wan. Y., (2017) “SDN- Based Data Center Networks,” IEEE: 1-13

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Ahmed Mateen received the M.Sc. degree from The University of Lahore and M.S. degrees in computer science from the University of Agriculture Faisalabad, Faisalabad, Pakistan. He is currently pursuing his Ph.D. degree from Department of Computer Science, Chongqing University, China and serve as Lecture Computer Science Department University of Agriculture Faisalabad. He has an Outstanding Academic Carrier. His research interests are Query Optimization, Modeling and Simulation, Machine Learning, Big Data Analysis and Network Security and Management.

Qingsheng Zhu (M'11) received the B.S., M.S., and Ph.D. degrees in computer science from Chongqing University in 1983, 1986, and 1990, respectively. He is currently a Professor with the College of Computer Science, Chongqing University, and also the Director of the Chongqing Key Laboratory of Software Theory and Technology. His main research interests include Ecommerce, data mining, and service-oriented computing.

ISBN: 978-988-14048-5-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)