Design and Modeling of Improved Transmission Control Protocol (TCP) Throughput

KOLAWOLE S.F., CHUKWU C.C.
Department of Electrical & Electronic Engineering, Nigerian Defence Academy

kolabimbo@hotmail.com

ABSTRACT

Transmission Control Protocol (TCP) is the dominant transport layer protocol of the internet protocol suite. It is a connection oriented protocol that provides reliable data delivery for internet applications like e-mail, file transfer, and world-wide-web. The task of determining the available bandwidth of TCP packets flow is very tedious and complicated. The complexity arises due to the effects of congestion control of the network dynamics and TCP. Congestion control is an approved mechanism used to detect the optimum bandwidth in which the packets are to be sent by TCP sender. The understanding of TCP behaviour and the approaches used to enhance the performance of TCP remains a major challenge. A considerable amount of researches has been made, in view of developing a good mechanism to raise the efficiency of TCP performance. The paper tries to improve TCP throughput by improving traditional TCP algorithm of Additive Increase with Multiplicative Decrease (AIMD. The work was carried out using OPNET modeler 17.5. The results obtained from a simulated scenario indicated an improved throughput on AIMD algorithm, by suppressing the queuing delay. The reduced queuing delay generated an improvement on the throughput of the traditional AIMD algorithm.

Keywords: TCP, UDP, Throughput, AIMD, Congestion, Slow-start, Fast recovery

1. INTRODUCTION

The Transport Control Protocol (TCP) and User Datagram Protocol (UDP) are the dominant transport layer protocols for the internet. TCP is connection oriented and is reliable for transferring data while UDP is connectionless. TCP was originally made for wired links

(Balver,2013). In an inter-network, congestion occurs at the router when incoming packets arrive at a rate faster than the router can switch or forward them to an outgoing link. TCP controls message/packet size and rate at which packets are exchanged and network traffic congestion. With congestion a switch or router has to start dropping packets because the buffer space is full.

The operation of TCP in wireless/mobile communications has been an important research issue in recent years, owing to the growth experienced in the area of modern telecommunications during the past decades. Significant contributions, such as the one presented in (Rai and Shreevastava, 2012) indicate that the unmodified, standardized operation of TCP is not well aligned with the peculiarities of cellular environments. Terminal movement across cell boundaries, leading to handover, is misinterpreted by common TCP implementations as sign of congestion within the fixed network. To handle such congestion, TCP unnecessarily slows down transmission by reducing window sizes, and performing retransmissions, if the need arises. TCP uses Additive Increase and Multiplicative Decrease (AIMD) algorithm for congestion control. With increase in speed on the internet and as newer technologies emerge there is need for higher throughput hence need for improved AIMD. Research is ongoing on improving the performance of TCP by using different congestion control algorithms

Throughput simply means how much packet is delivered in the given unit of time (Bhargava *et al*, 2012). One of the functions TCP performs is flow control, using the sliding windows method, which permits multiple data packets to be in transit concurrently, making more efficient use of network bandwidth. The destination advertises how much buffer space it has available, and the source restricts its transmissions such that the receiver is not overloaded. A small flow control window however, can adversely affect the throughput of the TCP connection. An overly small flow control window can cause TCP to act like a stop and wait protocol, whereas a very large flow control window would allow the sender to transmit continuously.

In TCP, the receiver acknowledges all received data with an ACK message, and the sender buffers the sent data until an ACK packet is received. If no ACK packet is received, the sender retransmits the data. The TCP sender uses two parameters, also referred as windows, to control the send rate. Both parameters limit the number of the packets that may be sent by the sender

without receiving the acknowledgment. The sender uses the minimal value which is provided by some two parameters. One parameter is the receiver-controlled offered window (also called advertised window), written in the ACK packet, that tells the sender how many bytes can be sent without overflowing the receiver's buffer. This parameter inhibits the sender from flooding the buffer of a slow receiver. The other parameter is the sender-controlled congestion window, which limits the number of packets that may be sent by the sender without receiving the acknowledgment. This parameter prevents network congestion. Packet losses are detected in TCP by using a timer that triggers after the time which is twice network round-trip time. TCP protocol assumes that losses are mainly due to congestion rather than to transmission errors. Therefore after detection of the packet losses, the sender voluntarily reduces the congestion window, which in turn decreases the transmission rate avoiding in such a way packet congestion. The senders decrease their send rate in two ways. First, when a packet loss occurs, the congestion window size is reduced to some threshold, and afterwards it increases slowly. Second, for every consecutive loss, the retransmission interval is doubled, in order to prevent the network from being flooded by retransmissions. Different actual TCP implementations have various ways to increase the window size again. Some of the implementations of TCP are as follows:

a. TCP Tahoe: The TCP implementation added a number of new algorithms and refinements to earlier TCP implementations. The algorithm includes Slow-Start, Congestion Avoidance, and Fast Retransmit (Reena and Maneesh, 2012). It introduced significant improvements for working over a shared network.

b. TCP Reno: The algorithm prevents the communication channel from going empty after Fast Retransmit, thereby avoiding the need to Slow-Start to re-fill it after a single packet loss. It adds Fast Recovery to TCP Tahoe. Fast Recovery is entered by a TCP client after receiving an initial threshold of duplicated ACKs. This threshold, is generally set to three. Once the threshold of duplicated ACKs is received, the server retransmits one packet and reduces its congestion window by one half. Instead of slow-starting after detection of a packet loss, as is performed by a TCP Tahoe server, the TCP Reno server uses additional incoming duplicated ACKs to clock subsequent outgoing packets. TCP Reno's Fast Recovery algorithm is optimized

for the case when a single packet is dropped during a window of data. The TCP Reno server retransmits at most one dropped packet per round-trip time. TCP Reno significantly improves upon the behavior of TCP Tahoe when a single packet is dropped from a window of data, but can suffer from performance problems when multiple packets are dropped.

.

c. TCP New-Reno: Enhanced TCP Reno using a modified version of Fast Recovery. TCP New-Reno addressed two problems in TCP Reno, these ideas are gradually finding acceptance within the IETF.

d. TCP Vegas: It uses a smart _Additive Increase Multiplicative Decrease(AIMD) technique. It adds rate control to avoid congestion (rather than react after detection of congestion).

e. SACK (Selective Acknowledgment): The probability of multiple packets loss in a window is much greater for a fast network, where many more packets are in transmit. Although TCP is able to recover from multiple packet losses without waiting for expiry of the retransmission timer, frequent packet loss may still not be efficiently recovered. The SACK extension (1996-98) improves TCP performance over such a network, and has been included in some recent TCP implementations. The SACK option is triggered when the receiver buffer holds in-sequence data segments following a packet loss.

In this paper we shall assume that packet losses due to network loss are minimal and most of the packet losses are due to buffer overflows at the router. Thus it becomes increasingly important for TCP to react to a packet loss and take action to reduce congestion

2. Review of Similar Works

The characteristics of wireless packet loss have been studied extensively. Padhye *et al*, 1998 developed a simple analytic characterization of steady state throughput, as a function of loss rate and round trip time for a bulk transfer. The model captures not only the behaviour of TCP's fast retransmit mechanism but also the effect of TCP's timeout mechanism on throughput. The

measurements obtained suggest that this timeout behaviour is important from a modeling perspective, as almost all of the TCP traces contained more time-out events than fast retransmit events. The measurements further demonstrate that the model is able to predict TCP throughput over a wider range of loss rates with a reasonable level of accuracy. However the assumptions used did not take the traffic generated by the fast retransmit mechanism and the possibility of congesting the transmission line especially when the re-transmitted packages are delivered afterwards.

Carrier sense multiple access (CSMA)-like distributed algorithms can achieve the maximal throughput in wireless networks (and task processing networks) under certain assumptions. One important but idealized assumption is that the sensing time is negligible, so that there is no collision. Jean and Walrand, 2011 provide a Markov chain model and give an explicit throughput formula that takes into account the cost of collisions and overhead. The formula has a simple form since the Markov chain is almost time-reversible. The authors also proposed transmission-length control algorithms to approach throughput-optimality in this case. Sufficient conditions are given to ensure the convergence and stability of the proposed algorithms

In a network, most efforts focus on the temporal loss characteristics at individual stations (Tang and McKinley, 2003). For multicast protocols and applications, however, the relationship among losses at multiple nodes can directly affect throughput. In existing models, packet losses are generally assumed to be spatially independent. The authors performed experiments with an IEEE 802.11 wireless LAN, however, they found that the losses at multiple nodes can exhibit a certain degree of correlation. There was an attempt to quantify this correlation, and demonstrate its effect on multicast communication protocols. Analysis and simulation results obtained show that conventional packet loss models do not adequately capture the loss characteristics exhibited in experimental traces..

There is an increasing demand for streaming video applications on the Internet (Feamster and Balakrishnan, 2002). Various network characteristics make the deployment of these applications more challenging than traditional TCP-based applications like email and the Web. Packet loss can be detrimental to compressed video with interdependent frames because errors potentially

propagate across many frames. While latency requirements do not permit retransmission of all lost data, therefore leverage the characteristics of MPEG-4 to selectively retransmit only the most important data in the bit-stream.

Packet loss patterns in Internet Protocol (IP) networks is important for achieving the desired quality of service in multimedia transfers. Markovski, Xue, and Trajkovic, 2001 simulated and analyzed packet loss in video transfers using User Datagram Protocol in a congested packet network. They used trace-driven *ns-2* simulations to collect packet loss traces in networks, and applied wavelet analysis to investigate the behaviour of packet loss on various time-scales. The results obtained showed that time-scales are essential for understanding loss behaviour and that packet loss exhibits long-range dependence over the coarser time-scales.

3. MODEL DESIGN AND IMPLEMENTATION

The round trip time (RTT) may be calculated for each of the windows and the total number of RTT can also be determined by the number of packets to be sent with respect to the channel capacity.

By definition, throughput may be mathematically represented as follows:

$$throughput = \frac{No\ of\ Packets\ (N) \times window\ size(w) \times link\ speed(\frac{Bits}{Sec})}{\frac{time}{RTT} \times No\ of\ RTT's.} \quad (1)$$

Throughput is also dependent on rate congestion by an inversely relationship such that:

$$throughput\ \propto \frac{1}{Congestion\ Rate} \quad (2)$$

Incorporating this relationship, an equation for better throughput is derived by considering the congestion control factor which is a function that affects the round trip time RTT

A general equation of the form

$$throughput = \frac{No\ of\ Packets\ (N) \times window\ size(w) \times link\ speed(\frac{Bits}{Sec})}{\frac{time}{RTT} \times No\ of\ RTT's \times Congestion\ factor\ A_0(t)} \quad (3)$$

Improved AIMD Algorithm for better TCP Throughput is summarized according to the flowchart in Fig. 1

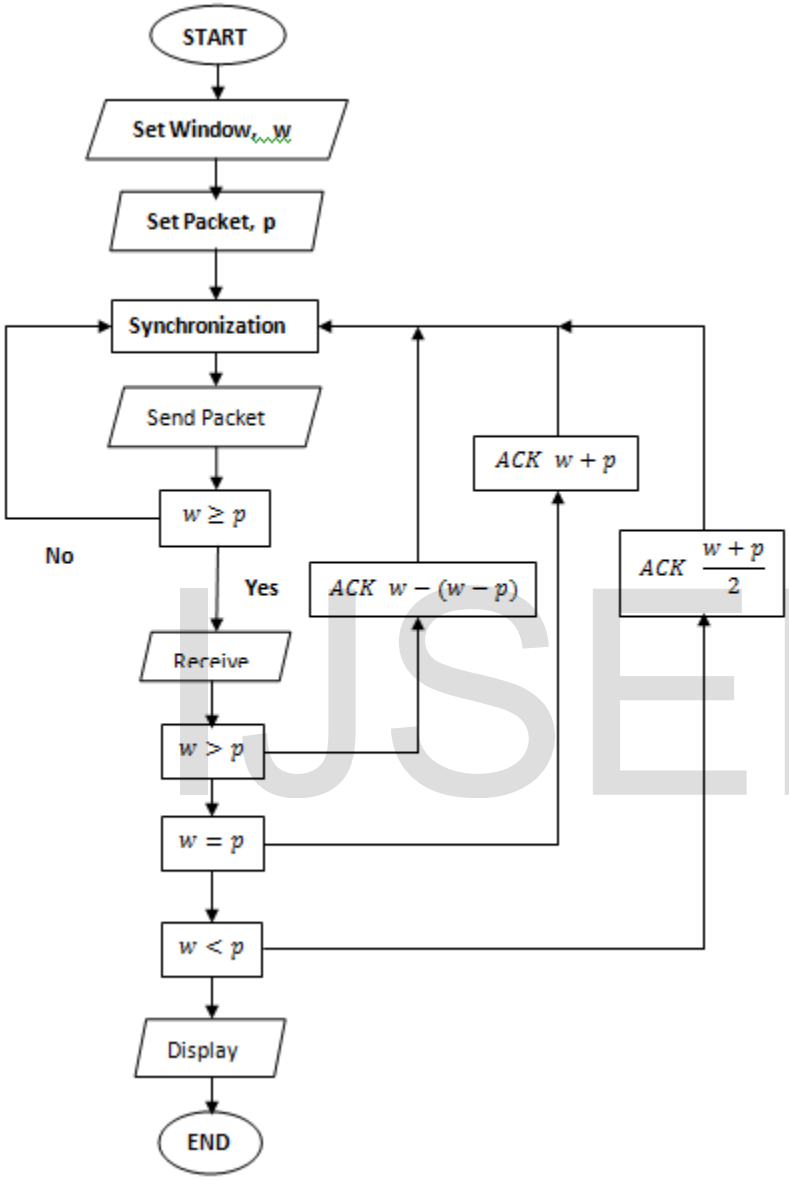

Fig 1 Flow Chart of Congestion Controlled AIMD Algorithm

Creation of Object Palette

The network implementation algorithm starts with Creating and Configuring the Network. This process is initialized by entering the Object Palette dialog box. The internet toolbox item is selected from the pull-down menu on the object palette and the necessary network objects added to the project workspace from the palette. For the purpose of this project the Application Configuration, Profile Configuration, an ip32_Cloud, and three subnets for multimedia interchange are added from a palette as shown in Fig. 2 and the project is saved thereafter.

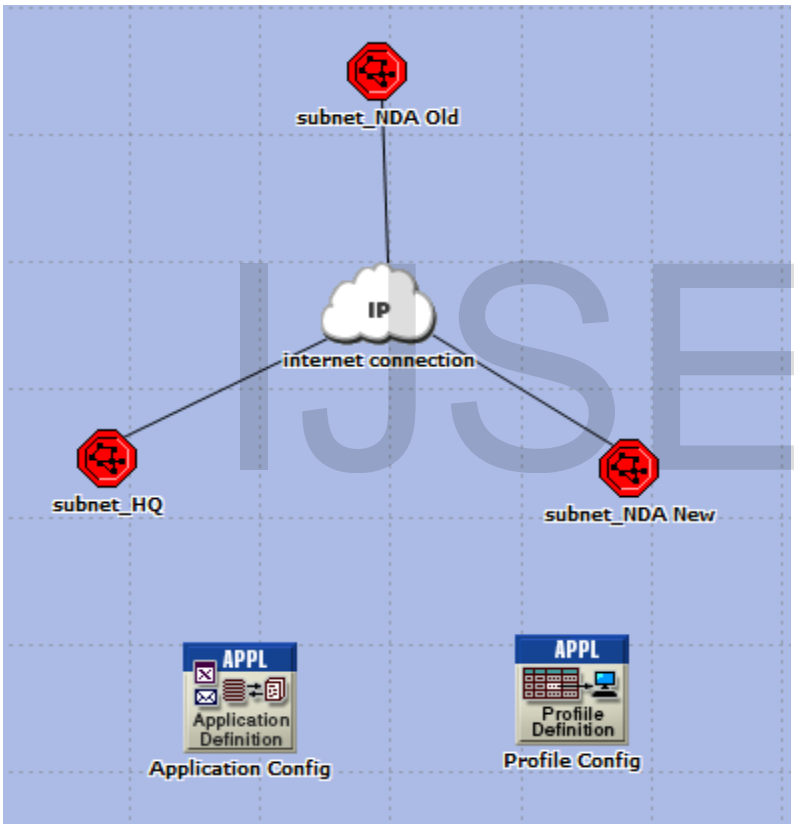

Fig 2 IP Component Selection from OPNET Object Palette

Subnet Configuration

The subnet is expanded by double clicking and an empty subnet space is created immediately. The desired components of the subnet will have workstations, switch, connection links, routers and servers for different applications. For the model simulation an ftp server, http server and database servers are used and connected according to fig 3.

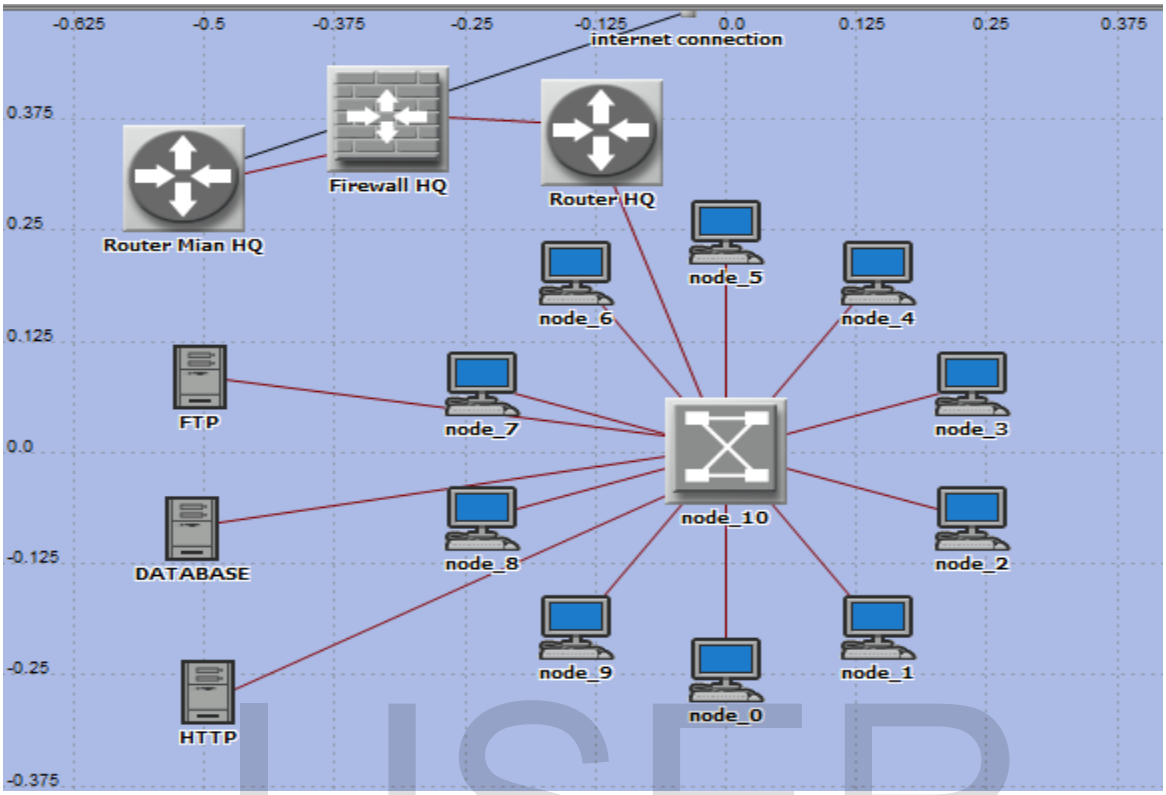

Fig 3 Subnet Network Topology

As can be seen, there are ten users at different nodes labeled node_0 to node_9, whose access to signals are controlled by a common switch using a 100T link. Router_Main and Router_HQ are interconnected with a protective firewall between them and then linked to the Ethernet switch using the 100T link. The various servers are configured appropriately and connected to the switch as well.

Profile Configuration

Profile Configuration Profiles describe the activity patterns of a user or group of users in terms of the applications used over a period of time. Several different profiles run on each LAN or workstation. These profiles represent different user groups for example, Engineering profile, accounts profile and Administration profile to depict typical applications used for each employee group. Profiles can execute repeatedly on the same node. OPNET is used to configure profile repetitions to run concurrently

4. RESULTS AND DISCUSSIONS

For the simulation we chose some static properties for TCP and the following results were obtained:

Response Time and Congestion Rate

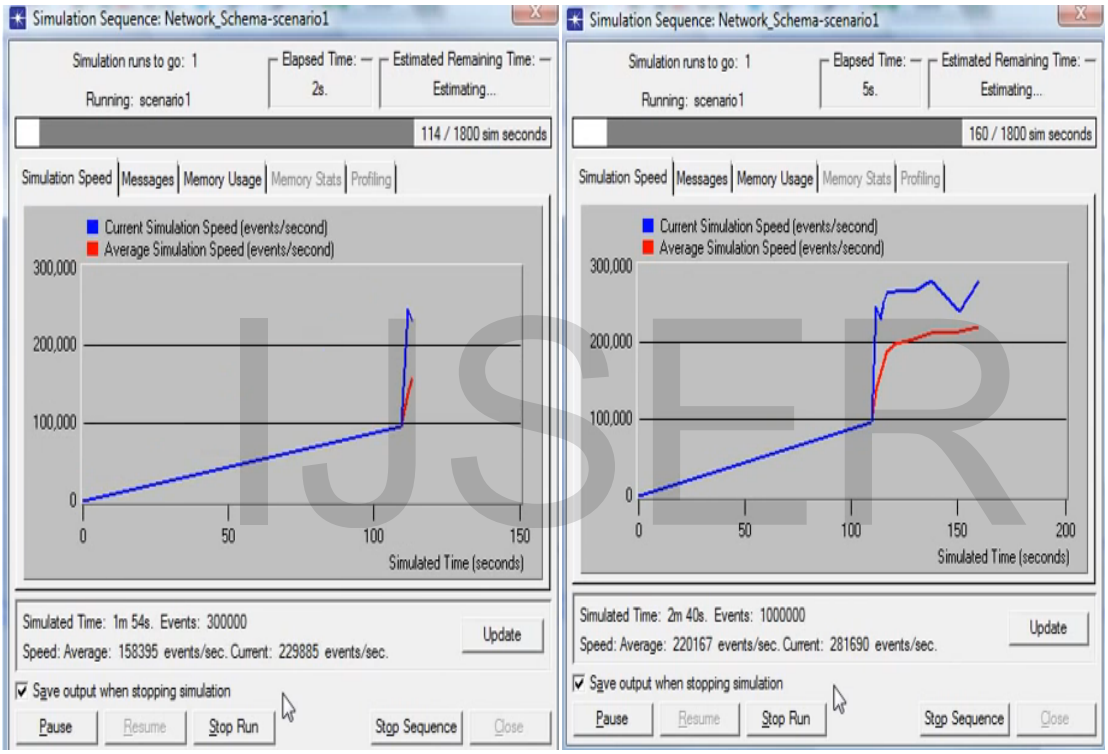

Fig. 4 Simulation Response time for 2secs and 5sec Duration

Fig. 4 shows the queuing process over 2seconds and 5seconds duration. While Fig. 5 shows simulation response time for 9secs and 1m:13sec Duration. The graphs becomes clogged as the simulation time increases.

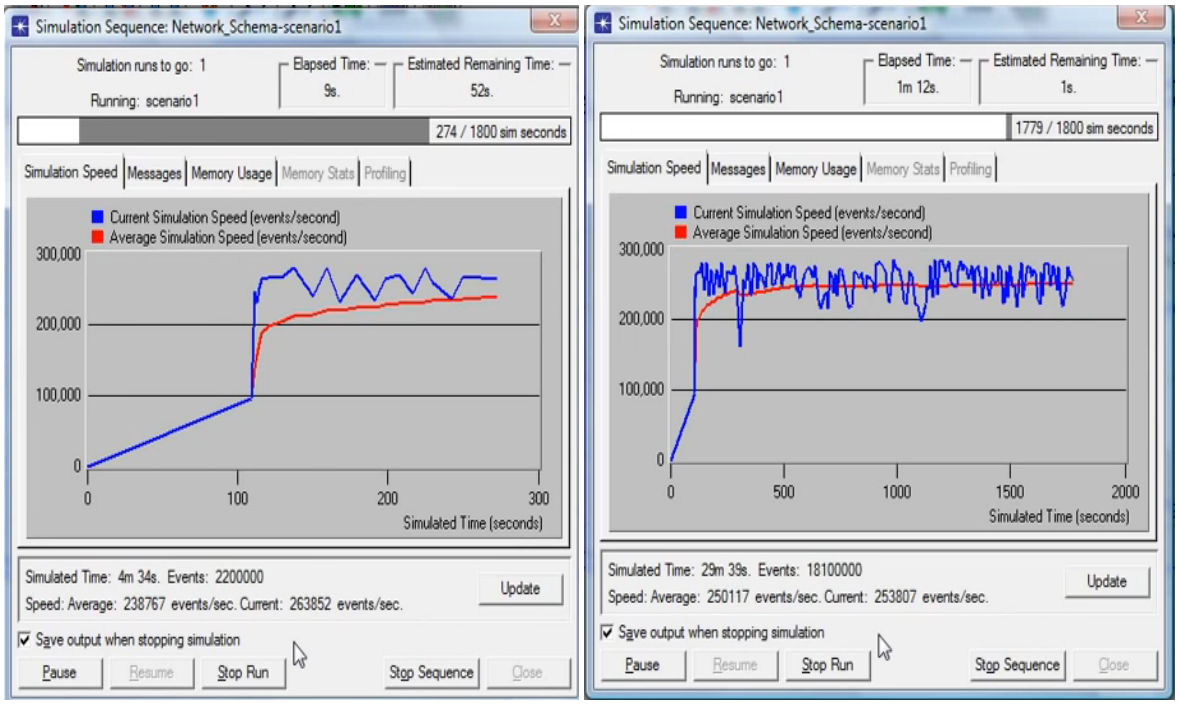

Fig. 5 Simulation Response time for 9secs and 1m:13sec Duration

Database Traffic Analysis

Typically, much of the data collected during simulations is stored in output scalar and output vector files. The Analysis Tool supports a variety of methods for processing simulation output data and computing new traces. Analysis Tool also supports the use of mathematical filters to process vector or trace data. Mathematical filters are defined as hierarchical block diagrams based on a predefined set of calculus, statistical, and arithmetic operators the result obtained is shown in Fig. 6

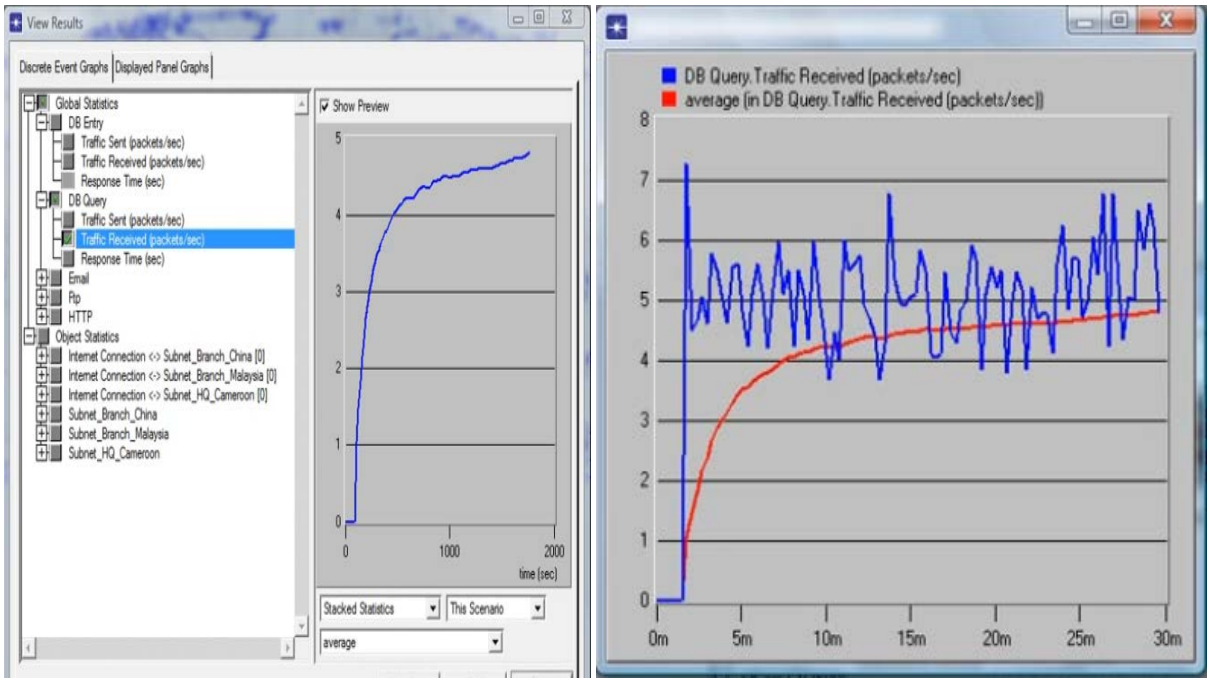

Fig.6 Database Traffic

HTTP Traffic Result

The HTTP traffic generated in both scenarios are slightly similar with just a minimal reduction in the congestion controlled HTTP traffic. The result clearly suggests that if congestion is controlled appropriately traffic is managed better. The HTTP traffic between sent and received files shows a slight drop, which is an indication of a better throughput as shown in Fig. 7

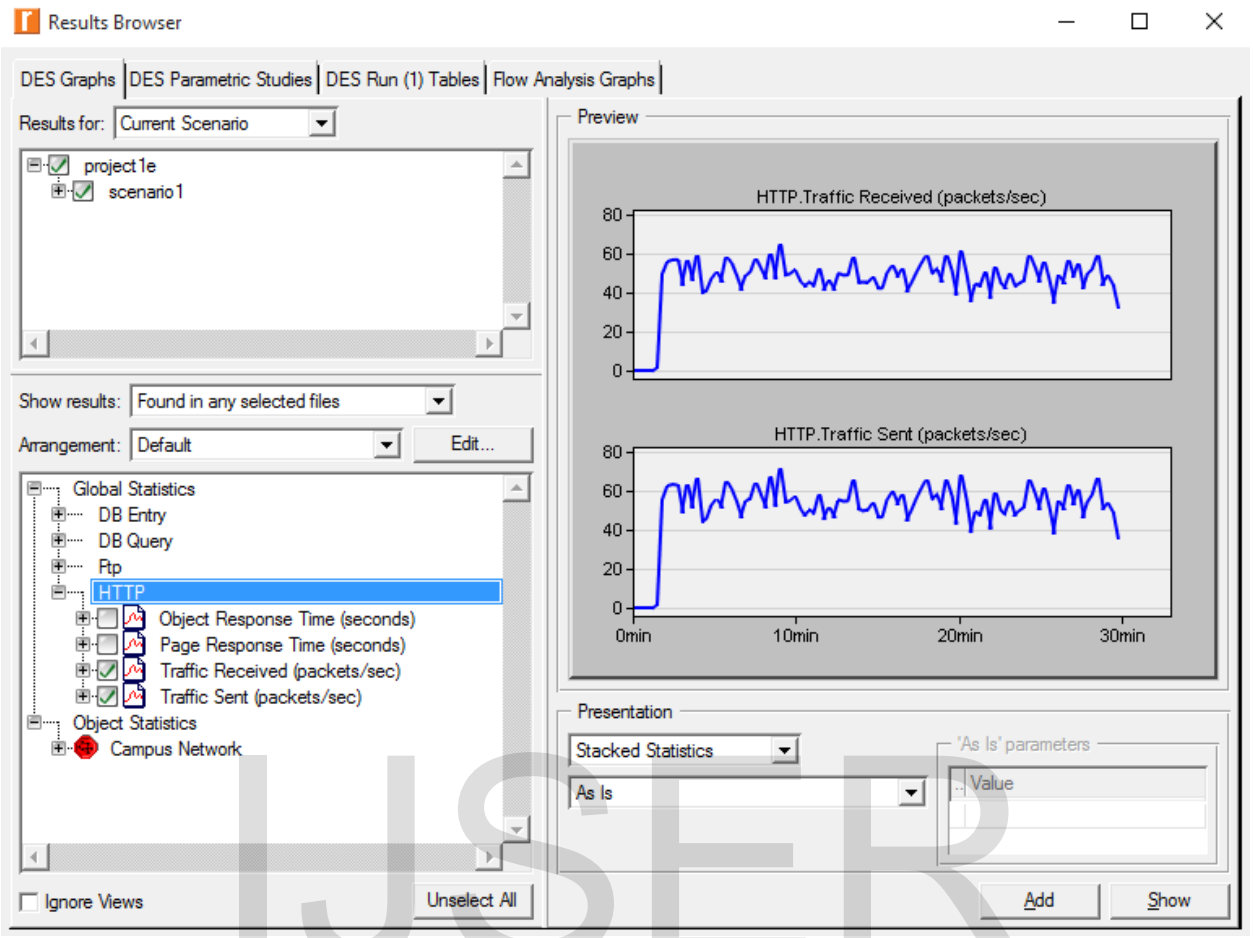

Fig. 7 HTTP Traffic for Packets Sent against Packets Received

Queuing Delay Analysis

Point to point queuing delays are compared for Congestion controlled AIMD and the traditional AIMD algorithms, tested within 20minutes duration and from the results obtained, the congestion controlled mechanism created a queuing suppression. This suppression implied that there is less delay on the link and packets loss is very much reduced due to the control offered by the congestion controlled process. The result obtained shows that the AIMD algorithm average a delay of 0.00011 sec/packet delay and the improved algorithm had an initial 0.0001sec/packet delay which was further suppressed 0.00002secs/packet as shown in Fig. 8

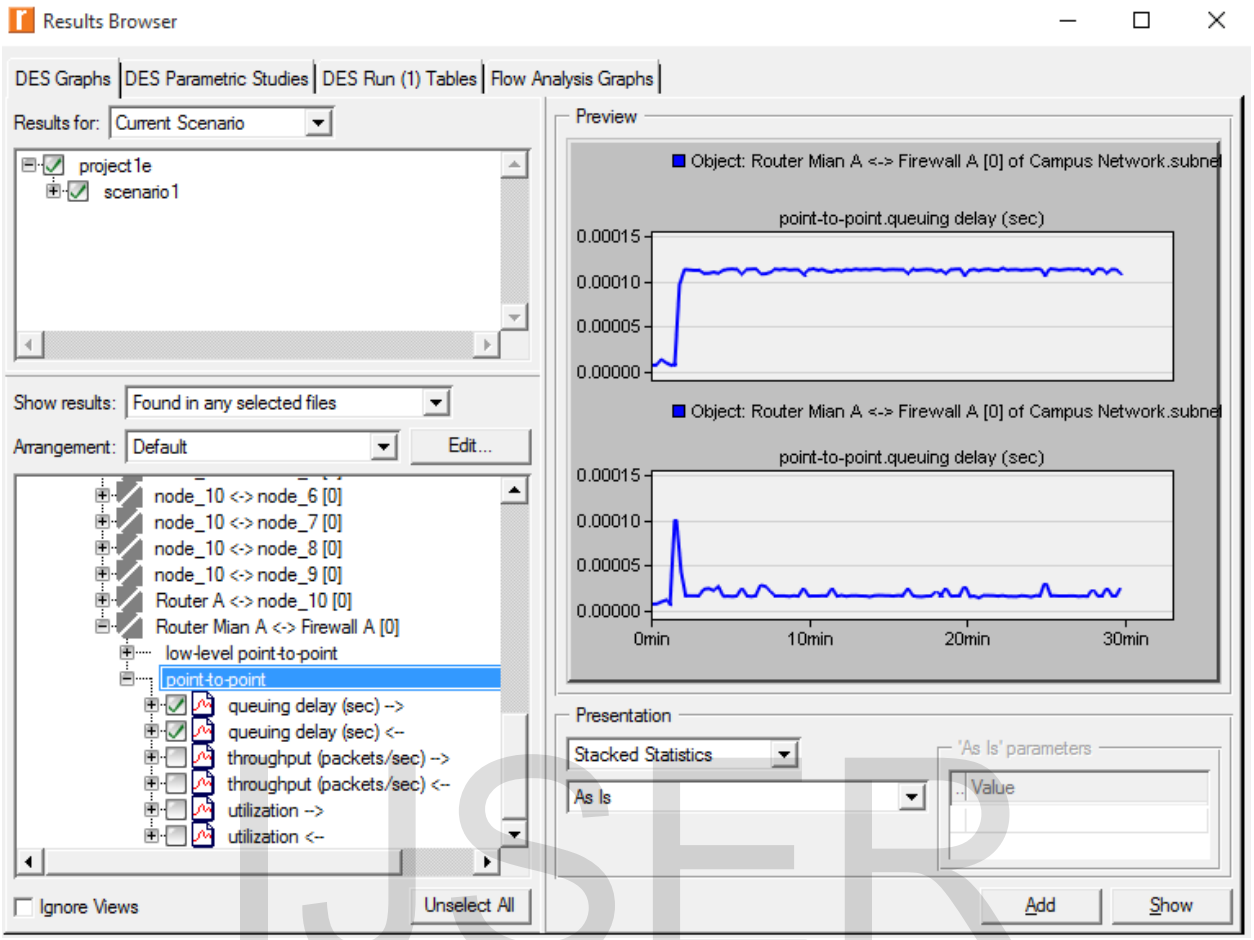

Fig.8 Point –To- Point Queuing Delay Result

Congestion Controlled AIMD Throughput against AIMD

The result obtained from AIMD throughput algorithm as compared with the congestion controlled AIMD throughput algorithms are displayed in Fig, 9 and as seen from the simulation result, the congestion controlled AIMD algorithm clearly gives an improved throughput as the number of packets delivered within same duration for both AIMD and Congestion controlled AIMD within a 20minutes duration shows an estimated 20 packets more delivered, which invariably connotes better quality of service delivery.

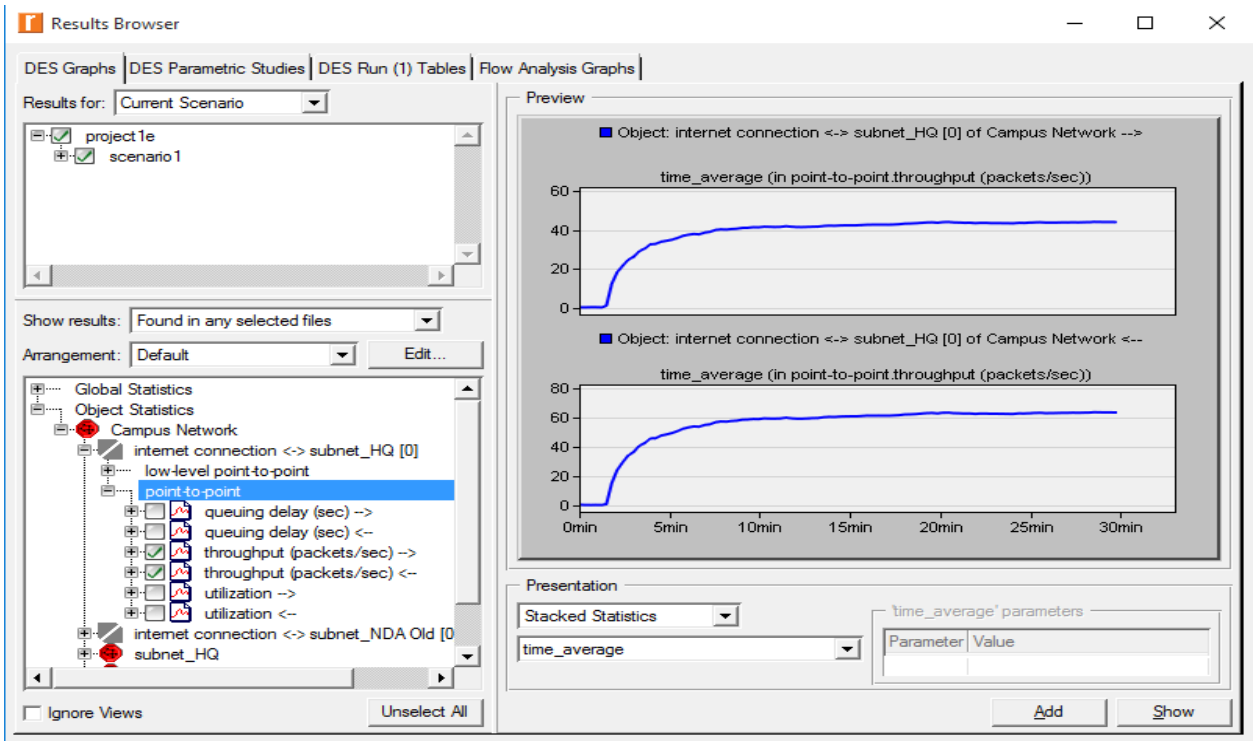

Fig.9 Comparison of AIMD with Congestion Controlled AIMD Throughput Algorithms

A summary of the results obtained for AIMD as compared with the Improved AIMD algorithm over a 100Mbps link simulated in OPNET modeller 17.5 is as represented in table 4.1

Table 1 Summary of Results

Time (Sec)	AIMD Throughput (Packets/Sec)	Improved AIMD Throughput	Total Packets Sent	Total Packets Received for Improved AIMD		Total packets Received for AIMD		Difference in Packets Received	
					%		%		%
150	25.0	30.1	5,000	4,515	**90.30**	3,750	**75.0**	765	**16.94**
300	34.0	50.0	16,800	15,000	**89.30**	10,200	**60.71**	4,800	**32.00**
600	40.0	59.3	40,000	35,580	**88.95**	24,000	**60.00**	11,580	**32.54**
900	41.0	61.1	62,000	54,990	**88.69**	36,900	**59.52**	17,190	**31.26**

1200	41.0	63.8	85,000	76,560	**90.07**	49,200	**57.88**	27,360	**35.74**
1500	40.9	63.4	105,000	95,100	**90.57**	61,350	**58.42**	33,750	**35.49**
1800	41.2	64.0	127,000	115,200	**90.71**	74,160	**58.39**	41,100	**35.68**

Average percentage improvement normal AIMD algorithm is **31.38%.**

5. CONCLUSION

In this paper we present an improved Additive Increase with Multiplicative Decrease for better TCP Throughput. The work was carried out using OPNET modeler 17.5. The results obtained from a simulated scenario indicated an improved throughput on AIMD algorithm, by suppressing the queuing delay. The reduced queuing delay generated an improvement on the throughput of the traditional AIMD algorithm.

REFERENCES

Azeddien M. Sllame and Mohamed Aljafari "Performance Evaluation of Multimedia over IP/MPLS Networks" *IACSIT, International Journal of Computer Theory and Engineering, Vol. 7, No. 4,* (2015)

Balver S. "A Comparative Study of Different TCP Variants in Networks" International Journal of Computer Trends and Technology (IJCTT) Volume 4 Issue 8, August 2013

Bhargava N., Kumawat A., Bhargava R and Kumar B. "Study of TCP Throughput on Network Simulator OPNET++ By Using Different Parameters" Journal of Environmental Research and Development Vol. 7 No 2 October-December 2012

Chiping Tang and Philip K. McKinley "Modeling Multicast Packet Losses in Wireless LANs" Conference Proceedings of the 6th International Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems MSWiM 2003, San Diego, CA, USA, September 19, 2003

Hayder Radha,Mihaela van der Schaar and Shirish Karande (2004) “Scalable video transcaling for the wireless internet” EUEASIP Journal on Applied Signal Processing volume 2004 pages 265-279 Hindawi Publishing Corp. New York, NY, United

Huda Adibah Mohd Ramli, Kumbesan Sandrasegaran, Riyaj Basukala and Leijia “Modeling and simulation of packet scheduling in the downlink long term evolution system” published in proceeding of the 15th Asia Pacific Conference on Communications pages 66-69, IEEE Press Piscataway, NJ, USA, 2009

Libin Jiang and Jean WalrandApproaching Throughput-Optimality in Distributed CSMA Scheduling Algorithms With Collisions IEEE/ACM Transactions on Networking(Volume: 19, Issue: 3, June 2011)

Nick Feamster and Hari Balakrishnan(2014) “Packet Loss Recovery for Streaming Video” 12th International Packet Video Workshop, Pittsburgh, PA, April 2002

Padhye J, Firoiu V, D Towsley J, Kurose “Modeling TCP throughput: A simple model and its empirical validation” ACM SIGCOMM Computer Communication Review, 1998

Rai R., Shreevastava M. “Performance Improvement of TCP by TCP Reno and SACK Acknowledgement”, International Journal of Advanced Computer Research, volume 2 number 1 Issue 3, March 2012

Velibor Markovski, Fei Xue, and Ljiljana Trajkovi´(2001) “Simulation and analysis of packet loss in video transfersusing User Datagram Protocol”The Journal of Supercomputing September 2001, Volume 20, Issue 2, pp 175–196