iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** VIII **Month of publication:** August 2017

DOI: http://doi.org/10.22214/ijraset.2017.8254

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue VIII, August 2017- Available at www.ijraset.com

Comparative Analysis of TCP Variants using NS-3.25 and Netanim on Ubuntu Version 16.04 Platform

Archita Saxena[1], Manish Choudhary[2]
[1]PG Student (Software Engg.), [2]Assistant Professor, Department of Computer Science,
Yagyavalkya Institute of Technology, Jaipur, Rajasthan

Abstract*: This paper comparatively analyzes maximum throughput for the TCP variants: New Reno, Westwood, HighSpeed & Hybla. All simulations have been performed in NS-3 (version 3.25) simulation tool and packet flow is observed in NetAnim tool on Linux platform. Throughput for each variant is calculated with reference to various varying network parameters which are: router link simulation time, router link bandwidth, number of traffic sources. Analysis was performed using dumbbell topology to find out the comparative maximum throughput of TCP variants. The analysis gives result as TCP Variant "NewReno" is good when low bandwidth is used, while TCP Variant "Hybla" is good in terms of using large bandwidths in comparison to Westwood and HighSpeed. Average throughput is also calculated to find out the best TCP variant.
Keywords: Congestion, Acknowledge, Router Link, RTT, Bandwidth

I. INTRODUCTION

Nowadays 4G networks are becoming more and more famous for which higher bandwidth and lower latency are being used. Therefore video streaming can be used in 4G environment. This type of application is increasing traffic in network very fast. Maximum amount of the traffic in these types of networks runs over TCP. Therefore, features of the transport protocol should be optimized for these critical situations. It will be very much beneficial for network operators and customers both. Optimization will increase the throughput, efficiency & downloading speed which is required by operator and customers. Queuing delay also affects the network so much so it should also be reduced. Queuing delay is very much essential for online gaming etc.

Transmission Control Protocol (TCP) is very important, as it is the dominant protocol in the Internet today. It is a connection - oriented, end-to end reliable protocol which lies at transport layer as shown below. TCP provides process-to-process communication by using port addresses. TCP also incorporates flow control and error control. Flow control is achieved by sliding window mechanism. For error control, acknowledgement, retransmission timers and retransmissions are used. The Transmission Control Protocol (TCP) has been designed to provide reliable, ordered and error checked end to end transmission of data between two hosts. Thus the protocol in itself can only contain mechanisms to limit the sending rate that are based on the sender and receiver windows. The first implementations did not include any limitations in the sending rate. It turned out not to be enough as it eventually led to degradation in the network conditions, and even to several collapses, as the number of computer connected to the Internet increased.

Many of the TCP variant has been introduced out of which we will analyse the comparative performance among NewReno, Westwood, HighSpeed and Hybla. We will provide analytical research of maximum throughput among above variants with reference to Network Bandwidth, Simulation Time and also with varying number of source count and will figure out the best scenarios for best throughput.

TCP is a reliable, connection-oriented delivery service. Connection-oriented means that a connection must be established before hosts can exchange data. Reliability is achieved by assigning a sequence number to each segment transmitted. TCP peers, the two nodes using TCP to communicate, acknowledge when they receive data. A TCP segment is the protocol data unit (PDU) consisting of the TCP header and the TCP payload, also known as a segment. For each TCP segment sent containing data, the receiving host must return an acknowledgment (ACK). If an ACK is not received within a calculated time, the TCP segment is retransmitted. TCP is able to transfer a continuous stream between two users. Sender can accept a stream of data bytes from the sending application and converts it into appropriate sized "segments" and send them to the receiver. TCP offers full duplex service means data can flow in both directions at same time.

II. TCP CONGESTION CONTROL ALGORITHMS

If the network is congested, then some packets will be dropped at an intermediate node e.g. a router. Consequently the network is further badly affected by the retransmission triggered for the dropped packets cumulatively resulting into more drops and more retransmission, flow control is used to provide relief to an overburdened network. TCP congestion can be controlled by four processes: Slow-start, Congestion avoidance, Fast retransmit and Fast recovery.

A. Slow-Start

TCP Slow Start is part of the congestion control algorithms put in place by TCP to help control the amount of data flowing through to a network. This helps regulate the case where too much data is sent to a network and the network is incapable of processing that amount of data, thus resulting in network congestion. In this process Sender TCP sends a segment in the starting. It continues to increase the congestion window by number of segments acknowledges until reaches ssthresh (a predefined threshold).

B. Congestion Avoidance

Once the ssthresh is reached, and cwnd > ssthresh, the Congestion Avoidance Algorithm is started and the cwnd size is increased linearly after every RTT. When segment acknowledgements are not received, the ssthresh is set to half of the current cwnd size (cwnd), and the algorithm restarts. The Congestion Avoidance Algorithm is used to control the cwnd, from the time when the available maximum rate was reached.

C. Fast retransmit

Implementing the TCP Slow-Start and Congestion Avoidance algorithms creates new problems. The first one is related to the packet loss detection. Normally, a packet loss is recognized when the timeout of the retransmission timer is detected. This, however, may lead to significant delays in the data transmissions, so another way to determine packet loss has been added to TCP which is called Fast retransmit process. In this process if three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost. Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen. At this moment TCP makes cwnd = 1 and ssthresh = cwnd/2. This results in restarting of slow-start process.

D. Fast recovery

In this process if three or more duplicate ACKs are received in a row, the TCP sender believes that a segment has been lost. Then TCP performs a retransmission of what seems to be the missing segment, without waiting for a timeout to happen. At this moment TCP makes cwnd = cwnd/2 and ssthresh = cwnd/2. This results in avoidance of restarting of slow-start process.
All the processes are shown in Fig 1.

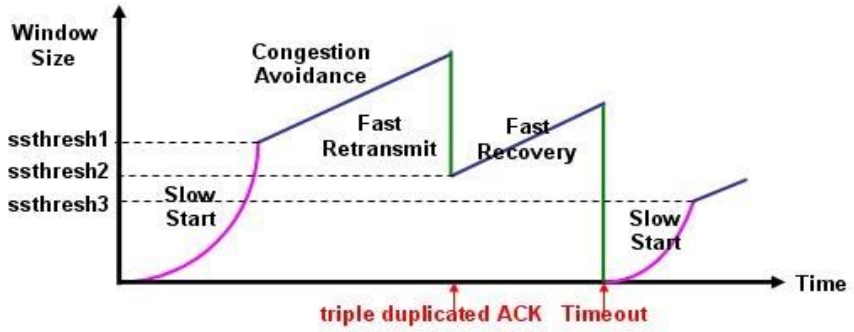


Fig 1 TCP congestion control algorithms

III. TCP VARIANTS

TCP has many variants namely Tahoe, Reno, New Reno, BIC, CUBIC, Westwood, HighSpeed, Hybla and many more. New transport protocols are always evolving with an objective to increase throughput and decrease the chance of getting into congestion. All these variants basically differ in the way they deal with congestion, control the data rate, and react with the lack of arrival of acknowledgments. The distinction of packet loss due to congestion or corruption is also an issue. In this paper we have analysed four variants: NewReno, Westwood, HighSpeed and Hybla.

A. TCP NewReno

New RENO is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like Reno, New-Reno also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from RENO in that it doesn't exit fast-recovery until all the data which was out standing at the time it entered fast-recovery is acknowledged. Thus it overcomes the problem faced by Reno of reducing the CWD multiples times. The fast-transmit phase is the same as in Reno. The difference is in the fast-recovery phase which allows for multiple re-transmissions in new-Reno. Whenever new-Reno enters fast-recovery it notes the maximums segment which is outstanding. The fast-recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases:

If it ACK's all the segments which were outstanding when we entered fast-recovery then it exits fast recovery and sets CWD to ssthresh and continues congestion avoidance like Tahoe.

If the ACK is a partial ACK then it deduces that the next segment in line was lost and it re-transmits that segment and sets the number of duplicate ACKS received to zero. It exits Fast recovery when all the data in the window is acknowledged.

B. TCP Westwood

Westwood builds an estimate of the rate of the connection and uses it to compute the slow start threshold and congestion window. Studying the behaviour of such a mechanism in LTE networks makes the CCA worth studying here. According to the introduction paper, the performance is improved both in wired and wireless networks. It is however most effective in wireless networks with lossy links. Thus Westwood solved one of the problems of Reno: its inability to determine the cause of a loss. However it does not address its slowness to reach full link utilization. If the bandwidth estimate falls into the first case, the loss has a high probability of being caused by congestion. Thus, the window is reduced to fit the bandwidth estimation.

C. TCP HighSpeed

High Speed TCP (HSTCP) is a modification proposed by S. Floyd to the TCP response function in order to acquire faster the available bandwidth (and faster reach full utilization of the link) in high bandwidth-delay product networks. The targeted network environments for HSTCP are low packet loss rate networks, therefore HSTCP proposes a faster congestion window increase compared to TCP. In this process, in congestion avoidance phase, cwnd is not increased by 1 packet every RTT, but by a dynamic value that depends on the current value of cwnd.

The AIMD increase and decrease parameters are then varied as functions of cwnd:

$$\text{Ack: } cwnd \leftarrow cwnd + f_\alpha(cwnd) / cwnd$$

$$\text{Loss: } cwnd \leftarrow g_\beta(cwnd) \times cwnd$$

HSTCP has very good convergence time to full utilization but known problems of HSTCP are low fairness with TCP flows (even in low bandwidth environments) and a higher convergence time to fairness among HSTCP flows.

D. TCP Hybla

Hybla is a relatively modest modification to Reno that aims at fixing a particular issue: its bias against connections with longer RTT. It is interesting to study as a variant of Reno taking into account the variability of the RTT into account. As exposed in [4], this CCA can be seen as an extension of the AIMD policy of Reno. Instead of using predefined coefficients for the increase during slow start or congestion avoidance the actual coefficients depends on the link's RTT. The objective is to compensate for the longer RTT by being more aggressive to be able to achieve the same performance as a connection with a reference RTT. This value is called RTT0 by the authors and they suggest of value of 25 ms. To achieve this they first define a constant ρ = RTT/RTT0 and the growth functions are defined as follows:

$$w = w + 2\rho - 1, \text{ In slow start}$$

$$w = w + \rho 2/w, \text{ In Congestion Avoidance}$$

IV. SIMULATION RESULTS

In this research paper we have analyzed four variants (NewReno, Westwood, HighSpeed and Hybla) and maximum throughput is calculated for each variant based on various parameters like bandwidth, simulation time and number of traffic sources. Dumbell topology is used to analyze the variants. Packet flow is observed in NetAnim and simulation is done in NS-3.25 tool. Fig 2 shows the node structure in dumbell topology created in NetAnim and Fig 3 shows node structure with 5 number of nodes.

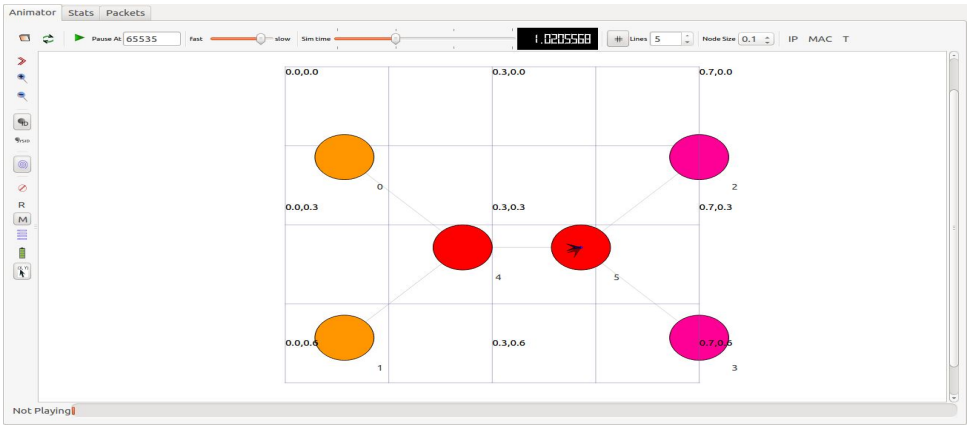


Fig 2 Node structure in NetAnim

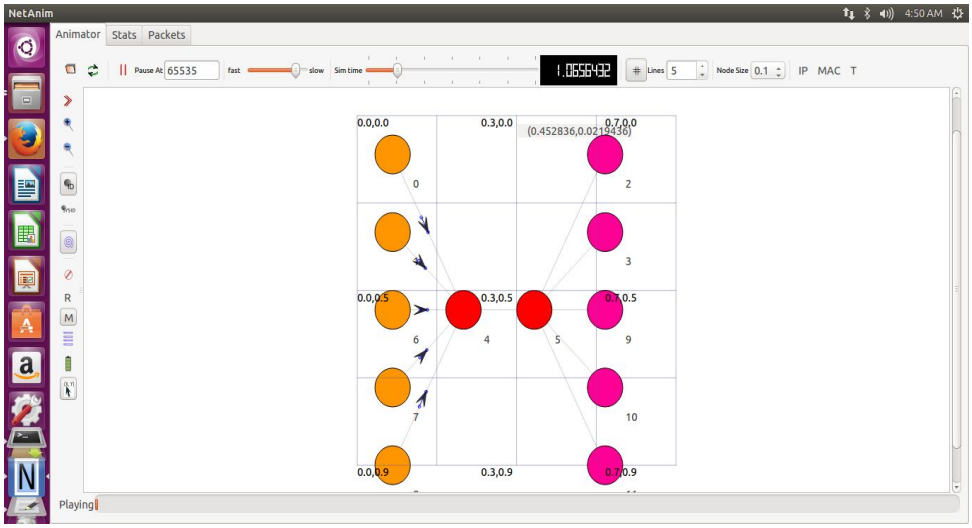


Fig 3 Node structure with 5 number of nodes

In Fig 2 yellow circles are senders, red circles are router and pink circles are receiver. Node 0 is connected to node 2 via nodes 4 & 5 having one TCP variant. Node 1 is connected to node 3 via nodes 4 & 5 having other TCP variant. Similarly process is followed for other two variants. Some values which are kept constant are mentioned in Table 1.

Table 1: Constant Values

Packet Size = 1500 bytes and network simulation time = 100 seconds		
Link	Bandwidth (MB)	Simulation time (ms)
n0-n4	5	10
n1-n4	5	10
n5-n2	5	10
n5-n3	5	10

A. Case I

Table 2 shows value of maximum throughput for different types of TCP variants based on different values of bandwidth of link between routers n4 & n5. Simulation time of router link is 20 ms.

Table 2: Throughput based on Router link Bandwidth

Bandwidth (Kb)	NewReno (Kbps)	Westwood (Kbps)	HighSpeed (Kbps)	Hybla (Kbps)
100	53.0317	45.597	46.1331	54.0348
400	185.312	163.627	171.198	198.138
600	277.358	229.399	241.807	296.632
800	347.781	301.794	325.791	379.944
1000	406.963	364.428	390.071	468.078
1200	499.873	409.651	480.035	541.324
1500	600.66	512.448	594.435	658.68
1700	649.64	575.707	663.272	728.631
2000	767.023	639.295	783.362	842.892
3000	1062.51	916.119	1140.55	1202.64
5000	1514.11	1454.41	1836.92	1901.05
7000	1983.78	1983.53	2569.02	2569.21
10000	2705.78	2705.39	3606.31	3606.62

B. Case II

Table 3 shows value of maximum throughput for different types of TCP variants based on different values of simulation time of link between routers n4 & n5. Now bandwidth of router link is fixed as 1 Mbps.

Table 3: Throughput based on Router link Simulation Time

Simulation Time (ms)	NewReno (Kbps)	Westwood (Kbps)	HighSpeed (Kbps)	Hybla (Kbps)
5	430.101	382.375	407.069	479.361
15	412.26	376.356	393.884	472.037
30	395.611	345.848	382.355	460.037
45	374.504	331.048	370.202	447.28
60	356.808	317.394	358.529	426.274
100	275.778	258.669	332.542	356.885
125	233.728	227.220	297.072	337.415
150	209.770	199.529	286.035	297.925
175	185.653	179.866	255.977	263.066
200	168.613	166.583	230.297	243.836

C. Case III

Table 4 shows value of maximum throughput for different types of TCP variants based on different values of number of senders and receivers. Now bandwidth of router link is fixed as 5 Mbps and simulation time is fixed as 10 ms.

Table 4: Throughput based on Number of Senders and Receivers

Number of Sources	NewReno (Kbps)	Westwood (Kbps)	HighSpeed (Kbps)	Hybla (Kbps)
2	1660.1	1595.3	1624.3	1695.8
5	1967.5	1854.6	1916.5	2118.2
7	1653.2	1538.4	1626.2	1728.6
9	1723.4	1685.2	1705.8	1785.8
12	1986.5	1865.7	1938.6	2123.3
15	1993.7	1856.8	1958.4	2085.6
18	2115.3	1968.7	1995.6	2154.5
20	1993.7	1878.2	1942.5	2058.6
23	2105.8	1923.6	1978.4	2134.3
28	2110.3	1942.5	1985.2	2146.4

Graphical representation of Case I, II & III is shown in Fig. 4, 5 & 6 respectively.

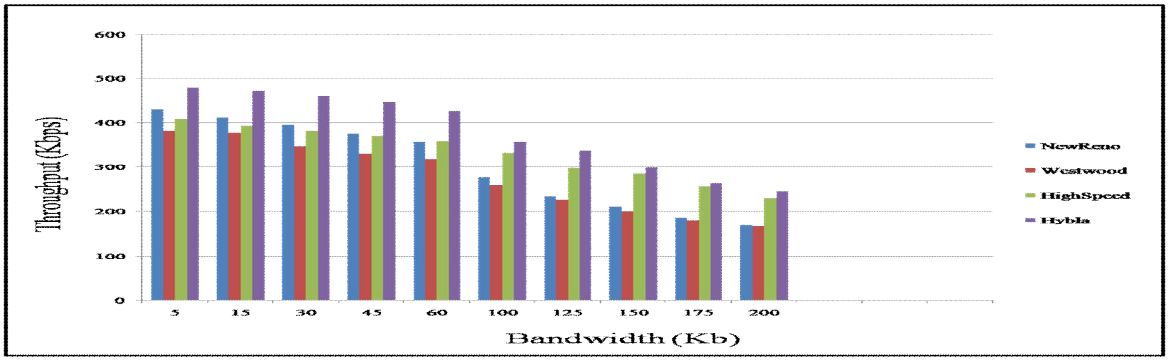


Fig 4 Throughput versus Router Link Bandwidth

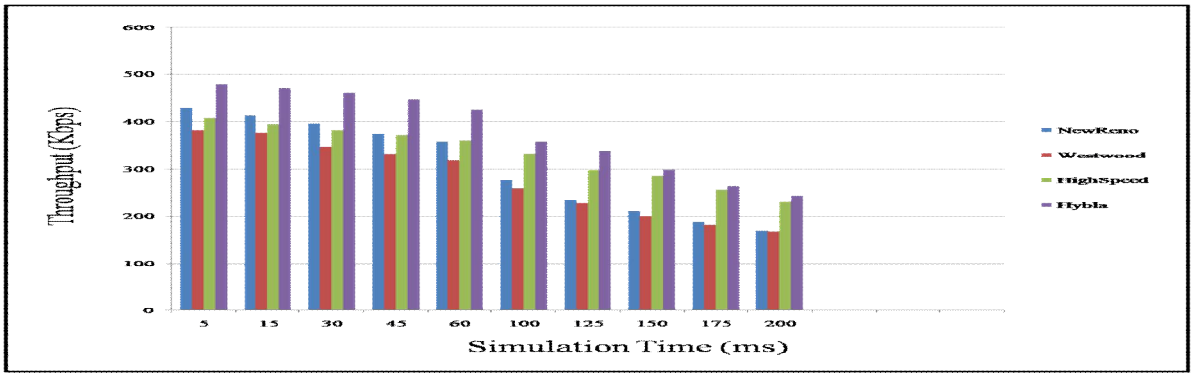


Fig 5 Throughput versus Router Link Simulation Time

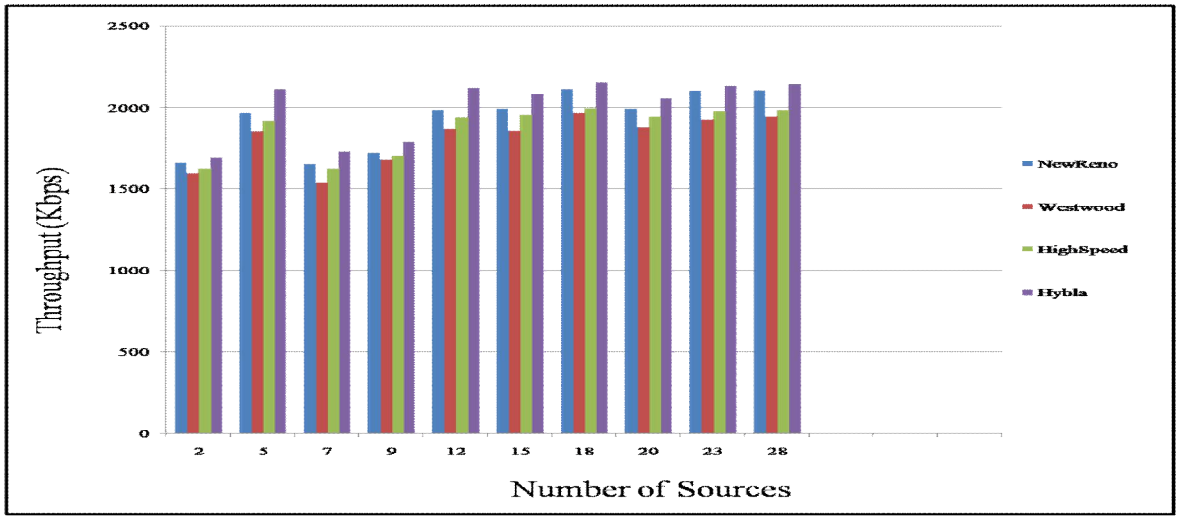


Fig 6 Throughput versus Number of Sources

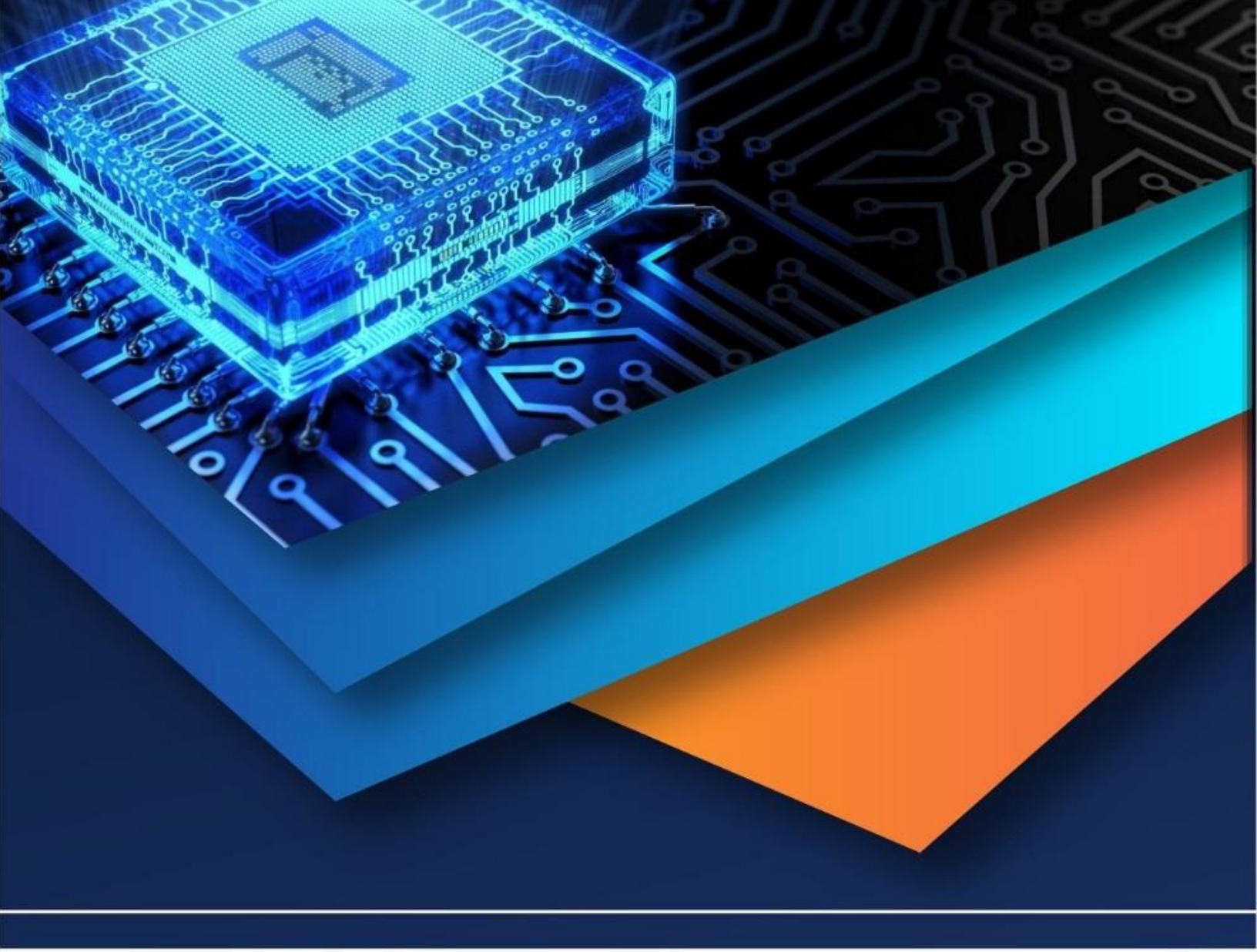
V. CONCLUSION

It was observed that we may use many types of TCP variants to control congestion in network. Increase in traffic in network increases the number of packets lost. Due to this more and more congestion occurs. In our research we have analysed four variants and compared according to average throughput. Our results conclude that Hybla variant performs very well in some cases and NewReno performs well in some cases but performance of Hybla is good in case of average throughput.

For case I average throughput for Hybla is 1034.45 Kbps & for NewReno is 850.29 Kbps. For case II average throughput for Hybla is 378.41 Kbps & for NewReno is 304.28 Kbps. For case III average throughput for Hybla is 2003.11 Kbps & for NewReno is 1930.95 Kbps.

REFERENCES

[1] Nosiba Ibrahim Alfadil Altahir & Hamid Abbas Ali, “Performance Evaluation of TCP Congestion Control Mechanisms Using NS-2,” IEEE, 2016.
[2] Prakash B. Khelage and Dr. Uttam Kolekar, “Survey and Simulation based Performance Analysis of TCP-Variants in terms of Throughput,Delay and drop Packets over MANETs”, IJSER, 2014.
[3] Subramanya P, Vinayaka K S, Gururaj H L, Ramesh B, “Performance Evaluation of High Speed TCP Variants in Dumbbell Network”, IOSR Journal of Computer Engineering, 2014.
[4] Balveer Singh, “A Comparative Study of Different TCP Variants in Networks”, International Journal of Computer Trends and Technology (IJCTT), August 2013.
[5] Madiha Kazmi, Muhammad Younas Javed and Muhammad Khalil Afzal, “An Overview of Performance Comparison of Different TCP Variants in IP and MPLS Networks”, Springer, 2011.
[6] Suhas Waghmare, Aditya Parab, Pankaj Nikose and Prof. S. J. Bhosale, “Comparative Analysis of different TCP variants in a wireless environment”, IEEE, 2011.
[7] Abdeljaouad, H. Rachidi, S. Fernandes, A. Karmouch, “Performance Analysis of Modern TCP Variants: A Comparison of Cubic, Compound and New Reno”, IEEE, 2010.
[8] Md. Shohidul Islam, M.A Kashem, W.H Sadid, M. A Rahman, M.N Islam, S. Anam, “TCP Variants and Network Parameters: A Comprehensive Performance Analysis”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009.
[9] R.L. Cottrell, S. Ansari, P. Khandpur, R. Gupta, R. Hughes-Jones, M. Chen, L. MacIntosh, F. Leers, Characterization and Evaluation of TCP and UDP-Based Transport On Real Networks. Proc. 3rd Workshop on Protocols for Fast Long-distance Networks, Lyon, France, 2005.
[10] Grieco, L. A. & Mascolo, S., “Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control” SIGCOMM, 2004.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY