Analytical Research of TCP Variants in Terms of Maximum Throughput

Archita Saxena, Manish Choudhary, Rakesh Khandelwal

Computer Science Department, Yagyavalkya Institute of Technology, Jaipur, Rajasthan, India

***Abstract*— This paper is comparative, throughput analysis, for the TCP variants as for New Reno, Westwood & High Speed, and it analyzes the outcomes in simulated environment for NS-3 (version 3.25) simulator with reference to multiple varying network parameters that includes network simulation time, router bandwidth, varying traffic source counts to observe which is one of the best TCP variant in different scenarios. Analysis was done using dumbbell topology to figure 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 "HighSpeed" is good in terms of using large bandwidths in comparison to Westwood. Network traffic flow was observed in NetAnim tool.**

***Keywords*— Network Topology, TCP Variant, Router, Bandwidth, Simuation Time**

I. INTRODUCTION

TCP (Transmission Control protocol), is a transport protocol, which is one of the best reliable protocol used in www service, mail and ftp over internet. This is reliable as it provides packet transfer from one host to other with acknowledgment over network .congestion control is the best and important feature for TCP in terms of performance measurement.

The main question of Controlling Congestion over the network (Less packet loss) has shifted to how possibly use the network capacity more efficiently, and here TCP variant's role come in picture as using different topology in different variants how throughput differ slightly.

Many of the TCP variant has been introduced out of which we will analyse the comparative performance among NewReno, Westwood and HighSpeed.

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.

While TCP is important protocol, a part of protocol standards known as TCP/IP, TCP reside at the upper side of IP layer, and to process further it pass the segments to IP layer, which subsequently processed through the lower layers and passed to network. TCP is designed for data/packet flow control with error correction while ensuring reliable message /data delivery from one node or source to other node or destination, and it is adopted in 1981 as standard RFC793, While IP was adopted in 1981 as standard RFC791.

Where mainly IP deal for logical address, which is specific to source address and destination address, and at the same time those addresses play main role in routing the message /information /packet to its specific destination and also give return address to the respective response.

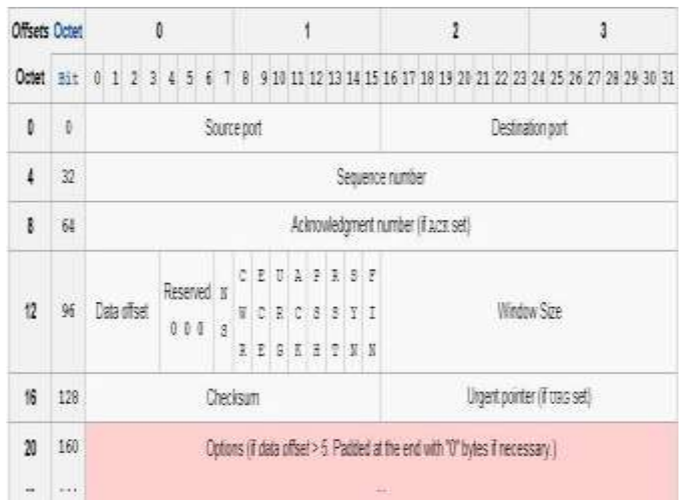

Fig. 1 TCP Header Diagram

For Data transmission, TCP protocol is mainly used as for two way (bidirectional) communication, P2P end to end reliable data transfer. TCP protocol break the data (message) from upper protocol layers to datagram, which is encapsulated to packet to further transmission over network, in further process TCP receiver re arranged /re assembled those packets to original data/message, then forward to next higher level layers.

Once a packet is sent over the network by a source, Acknowledgement (ACK) is always expected from destination. Due to this ACK, source comes to know if sent packet was successfully received at the destination end or not.

II. TCP CONGESTION CONTROL

TCP send packets to the network without reservation further react to the events that get occur. TCP works with fair queuing and assumes for FIFO queuing in network's routers, but main issue is the Internet suffering of congestion collapse—hosts try to send packet over network as fast as once advertised window allows, while congestion may occur at some router (causing packets drop), and hosts observe time out and retransmit the packets, which further result to more congestion. Mainly idea of congestion control is for each sender to understand and find how much bandwidth/free capacity is left in the network, so that can determine how many packets can be transmit safely .So a source has count this many packets are in transit, then source uses the ACK as one of the packet has been transmitted and now can send new packet over the network without any congestion. So TCP is called self-clocking. But at the same time other connections also continue to in and out and the available bandwidth keep on changing , so source must be able to readjust number of packets it has in transmit, In this research we will show TCP variants and its congestion control mechanism. TCP congestion control mechanism can be defined by two processes:

A. *Slow Start*

Success of TCP data packet transmissions is identified by the incoming acknowledgements from receiver. Moreover there is a problem when a TCP connection is first established causes to have acknowledgements, so we need to have data packets in the TCP network and to put data packets in the TCP network we need acknowledgements from receiver. To eliminate this problem initially Tahoe suggests that whenever a TCP connection is established or connection re-starts after the loss of packet, network should follow a process called '*slow-start*'. The main cause for this process is that an initial burst might overwhelm the network and the connection might never get started. Slow Start suggests that the sender set the congestion window to 1 and then for each ACK received it increase the CWD by 1. So in the first round trip time (RTT) we send 1 packet, in the second we send 2 and in the third we send 4. Thus we increase exponentially until we lose a packet which is a sign of congestion. When we encounter congestion we decreases our sending rate and we reduce congestion window to one. And start over again. In usual implementations, repeated interrupts are expensive so we have coarse grain time-outs.

B. *Congestion Avoidance*

For congestion avoidance Tahoe uses 'Additive Increase Multiplicative Decrease'. A packet loss is taken as a sign of congestion and Tahoe saves the half of the current window as a threshold value. It then set CWD to one and starts slow start until it reaches the threshold value. After that it increments linearly until it encounters a packet loss. Thus it increase it window slowly as it approaches the bandwidth capacity.

III. TCP VARIANTS

They are various types of variants of TCP protocol: Tahoe, Reno, New Reno, Sack, Vegas, Westwood, HighSpeed and Hybla. Tahoe, Reno, Sack and Vegas have been removed from NS-3.25 simulation tool. We have worked on NewReno, Westwood and HighSpeed TCP.

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. The basic mechanism is presented as under:

Step 1: Initially
0<CWND<= min (4*mss, max (2*mss, 4380 bytes))
SSThreshold = max (CWND/2, 2*MSS)

Step 2: Slow Start Algorithm (Exponential Increases)
if (receive ACKs && CWND SSThreshold)
CWND = CWND+1;

Step 3: Congestion Avoidance Algorithm (Additive increase)
if (receive ACKs) {
if (CWND > SSThreshold)
CWND = CWND + (segsize * segsize / CWND);
else
CWND = CWND + 1}

Step 4: Congestion Detection Algorithm (Multiplicative Decrease): Fast Retransmission and Fast Recovery
if (congestion) {
if (Receive same ACKs 3 time or retransmission time out) {

```
                    SSThreshold = CWND/2;
                        if (Retransmission time out) {
                              CWND = initial;
                               exit and call Slow Start
                                   step;
else
                   CWND = SSThreshold;
            exit and call congestion avoidance step
                        }
            }
}
```

New-Reno suffers from the fact that it takes one RTT to detect each packet loss. When the ACK for the first re-transmitted segment is received only then can we deduce which other segment was lost.

B. TCP Westwood

TCP Westwood proposes an end-to-end bandwidth estimation algorithm based on TCP Reno. TCP Westwood implements slow start and congestion avoidance phases as TCP Reno, but instead of halving the congestion window size as in TCP Reno when congestion happens, TCP Westwood adaptively esti-mates the available bandwidth and sets the congestion window size and slow start threshold accordingly to improve the link utilization. In TCP Westwood, packet loss is indicated by the reception of 3 duplicated acknowledgements (DUPACKs) or timeout expiration. When 3 DUPACKs are received, TCP Westwood sets SSTreshHold and CWND as follows:

```
if (n DUPACKs are received)
if (CWND>SSThreshhold)  /* congestion avoid. */
SSThreshhold = BWE*RTTmin;
CWND = SSThreshhold;
endif
if (CWND<SSThreshhold)  /*slow start */
SSThreshhold= BWE*RTTmin
if (CWND > SSThreshhold)
CWND = SSThreshhold
endif
endif
endif
```

In TCP Westwood, the setting of SSThreshold and CWND is based on the bandwidth estimation, which is obtained by measuring the rate of the acknowledgments and collecting the information of the amount of packets delivered to the receiver in the ACK. Samples of bandwidth are computed as the amount of packet delivered divided by the inter-arrival time between two ACKs. Those sample bandwidth estimates are then filtered to achieve an accurate and fair estimation. TCP Westwood modifies the Additive Increase and Multiplicative Decrease (AIMD) in TCP Reno and adaptively sets the trans-mission rates to remove the oscillatory behaviour of TCP Reno and to maximize the link utilizations. But this also causes TCP Westwood to degrade the performance of TCP Reno connections when they coexist in the network.

It performs poorly if it estimates incorrect bandwidth because of unpredictability in the behaviour of the bandwidth estimation scheme used in TCP Westwood.

The sensitivity of TCP Westwood Ackd Interval is variable.

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.

HS-TCP uses the current TCP cwnd value as an indication of the bandwidth-delay product on a path. The AIMD increase and decrease parameters are then varied as functions of cwnd:

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

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

In a literature logarithmic functions are proposed for $f\alpha(cwnd)$ and $g\beta(cwnd)$, whereby $f\alpha(cwnd)$ increases with cwnd and $g\beta(cwnd)$ decreases. HS-TCP uses a mode switch so that the standard TCP update rules are used when cwnd is below a specified threshold.

and TCP are similar. The HSTCP response function could be expressed by

$$Cwnd = 0.12/p^{0.835}$$

Therefore, 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. In case of packet losses, the multiplicative decrease factor is also dynamic (but is lower than 1/2). 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.

IV. SIMULATION AND RESULTS

In this paper we have compared three protocols (NewReno, Westwood and HighSpeed) in terms of maximum throughput based on different parameters like bandwidth, simulation time and number of traffic sources. Dumbell topology is used to analyze the protocols. All simulations are performed on NS-3.25 simulation tool NetAnim. Fig. 2 shows the dumbell topology used for simulation.

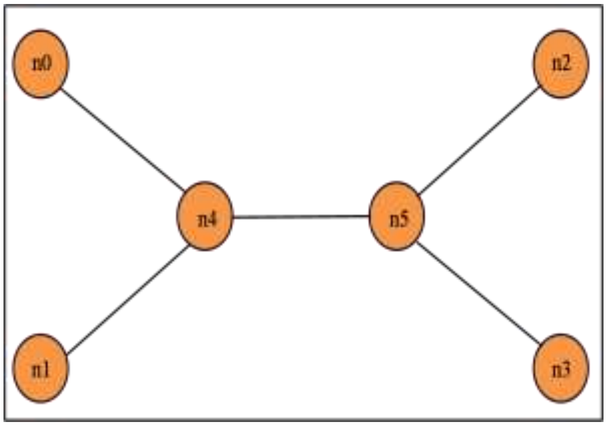

Fig. 2 Dumbell Topology

In Fig. 2, nodes n0 & n1 are senders, nodes n2 & n3 are receivers and nodes n4 & n5 are routers. Packets flow from n0 and received by n2. Similarly packets flow from n1 and received by n3. TCP connection is established between n0 & n2 and n1 & n3 through n4 & n5. All nodes have been given different IP Address.

To analyze all the protocols packet size is kept constant (1500 bytes) and simulation time of network is 100 seconds. Table I shows the values which have been kept constant throughout the simulation.

Table I
Constant Values

Link	Bandwidth (MB)	Simulation time (ms)
n0-n4	5	10
n1-n4	5	10
n5-n2	5	10
n5-n3	5	10

Table II 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 II
Throughput based on Router link Bandwidth

Bandwidth (KB)	NewReno (Mbps)	Westwood (Mbps)	HighSpeed (Mbps)
100	0.053	0.045	0.046
400	0.185	0.163	0.171
800	0.347	0.301	0.325
1200	0.499	0.409	0.480
1700	0.649	0.575	0.663
3000	1.062	0.916	1.140
5000	1.514	1.454	1.836
7000	1.983	1.983	2.569
10000	2.705	2.705	3.606

Table III 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 III
Throughput based on Router link Simulation Time

Simulation Time (ms)	NewReno (Mbps)	Westwood (Mbps)	HighSpeed (Mbps)
5	0.430	0.382	0.407
15	0.412	0.376	0.393
30	0.395	0.345	0.382
45	0.374	0.331	0.370
60	0.356	0.317	0.358
100	0.275	0.258	0.332
125	0.233	0.227	0.297
175	0.185	0.179	0.255
200	0.168	0.166	0.230

Table IV 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 IV
Throughput based on Number of Senders and Receivers

Number of Sources	NewReno (Mbps)	Westwood (Mbps)	HighSpeed (Mbps)
2	1.660	1.642	1.638
5	1.967	1.856	1.948
7	1.653	1.534	1.655
9	1.723	1.685	1.748
12	1.986	1.894	1.996
15	1.993	1.912	2.084
18	2.115	1.968	2.158
20	1.993	1.892	2.132
23	2.105	1.945	2.194

Graph representation for Table I, II & III is shown in Fig. 3, 4 & 5 respectively.

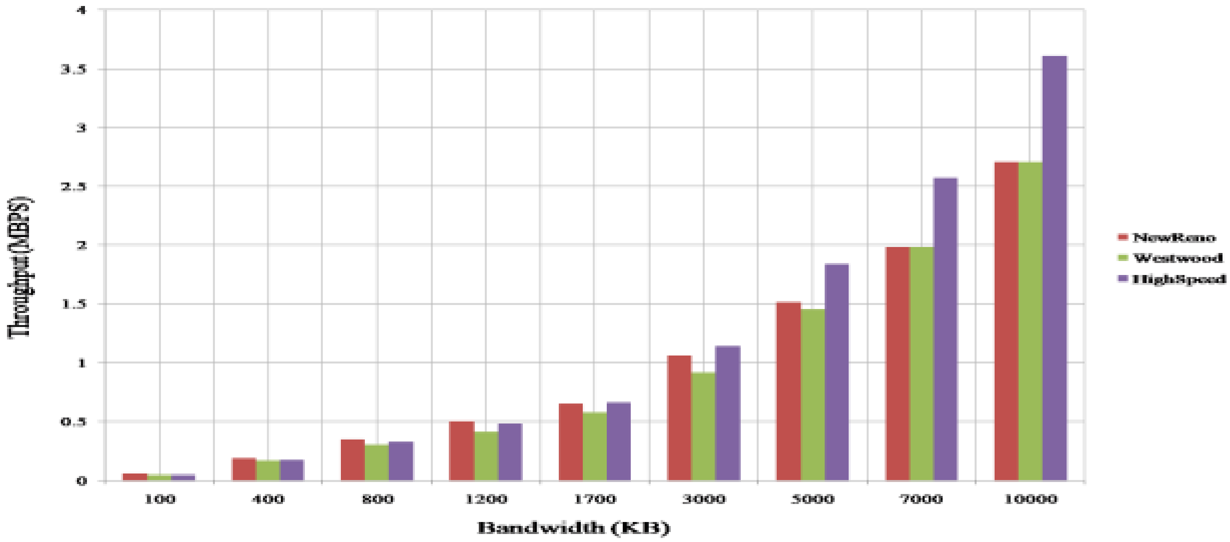

Fig.3 Throughput based on Bandwidth

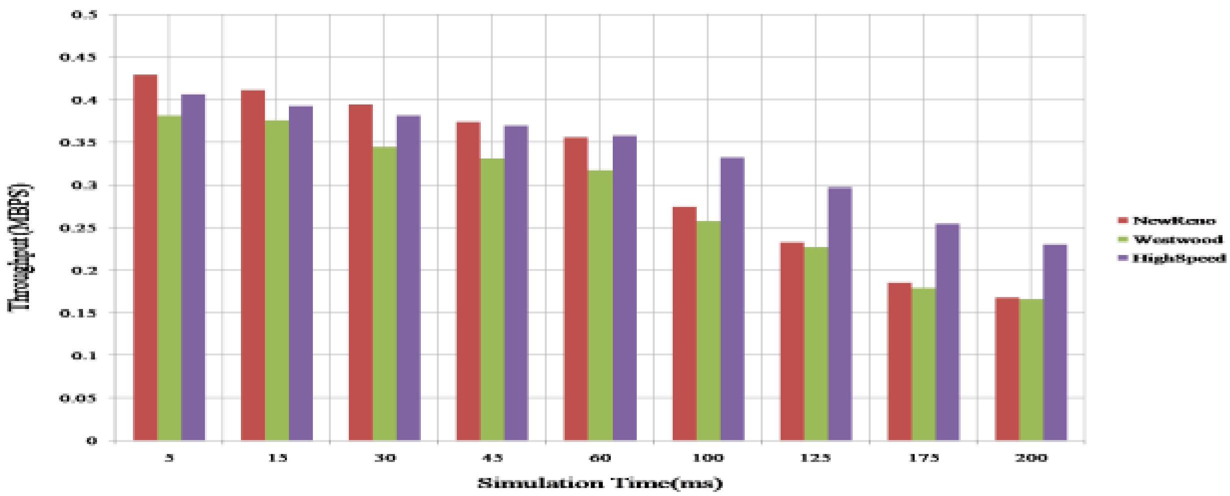

Fig.4 Throughput based on Simulation Time

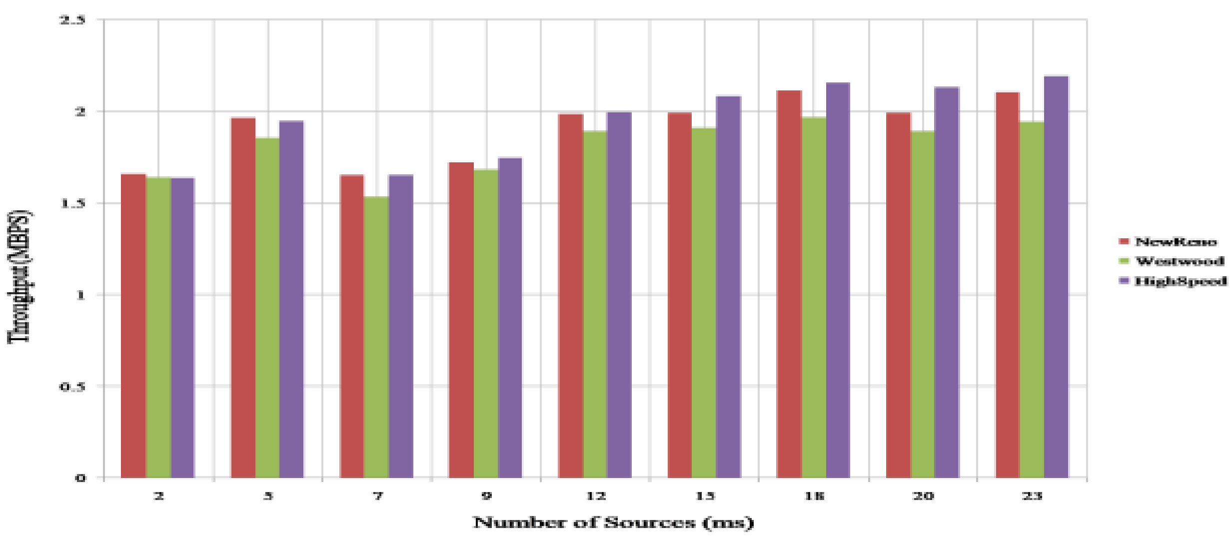

Fig.5 Throughput based on Number of sources

V. CONCLUSION

As a result, In New Reno, detection of multiple losses is available, so it is beneficial when there are several losses of data packets. New Reno is good to use for low bandwidth but HighSpeed is good at large bandwidths. As simulation time increases HighSpeed becomes more useful.

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] Balveer Singh, "A Comparative Study of Different TCP Variants in Networks", International Journal of Computer Trends and Technology (IJCTT), August 2013.

[4] 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.

[5] Abdeljaouad, H. Rachidi, S. Fernandes, A. Karmouch, "Performance Analysis of Modern TCP Variants: A Comparison of Cubic, Compound and ew Reno", IEEE, 2010.

[6] 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.

[7] 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.

[8] Grieco, L. A. & Mascolo, S. "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control" SIGCOMM, 2004