International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

"Improving TCP's Qos using end-to-end delay based congestion control techniques"

1.Deepika Bhenia
SDBCT,Indore
deepika07bhenia@gmail.com
2. Abha Jain
Professor at SDBCT, Indore
Abh_sethi@rediffmail.com

Abstract –
Transmission control protocol is oriented for reliable data transfer which carry more than 90% of load in wireless network. One of the key features which make TCP, implements a window based flow control mechanism. After each successful and unsuccessful delivery, it allows window to shrink and expand. A number of research's and investigation has been done in this area to control the expansion of TCP window for better congestion control and flow control. In early days, TCP research shows that it works unsatisfactorily in rapid network and low round trip time networks when large data is sent. TCP is not appropriate for the time of low capacity and less delay networks. In this paper, we describe a delay-based end-to-end congestion control algorithm, called TCP-EXT, which is an extension of delay and acknowledge based TCP protocols like Linux,Reno, Newreno etc. The idea behind this extension is to frequently estimate the round-trip-time, which eliminates throughput and queue oscillation when round-trip time oscillates. In this context, we introduce a simulation and an evaluation of the suggested scheme, by comparing our TCP-EXT prototype with various TCP variants in terms of throughput, fairness, stability, RTT and adaptively to changes in network.
Keywords-TCP; RTT; end to end delay; congestion control; TCP-EXT.

1.INTRODUCTION

The TCP is extensively tuned to provide high-quality performance in the conventional wired network. TCP was describe to provide a reliable end-to-end which adapt to network conditions to guarantee transmission of data . TCP is also called as a connection-oriented protocol , which means that a connection is stable and control until such time as the data to be changed by the application at every end have been exchanged. TCP is important for ensuring that a data is divided into the packets that maintained IP and for reassemble the packets aback into the complete data at the receiver end. The performance received by end users of these Internet applications depends on the performance of TCP. Transmission control protocol provide congestion control , byte stream transport, flow control, and congestion control mechanism. However, TCP in its present form is not well suited for wireless networks.because when too many packet try to access same route buffer at same time resulting large amount of packet being drop.and TCP Route failures occure packets drop at the inter-mediate nodes, which will counted as congestion loss. In the case of congestion , TCP limit sender transmission rate to reduce load in path between sender and receiver.it use window –based scheme to control transmission rate because size of window directly impact on transmission. TCP understand this which means decreasing both the congestion window (cwnd) and slow start threshold (ssthresh). This paper ,we organize delay based Congestion Control Algorithm known as EXT-TCP followed by TCP linux ,TCP Reno, NewReno.

2.TCP Congestion Control Algorithms: Four Congestion Control Algorithms: Slow Start, Fast Retransmit, Fast Recovery, Congestion Avoidance.

2.1 Slow Start: Slow-start algorithm is used to control congestion inside the network. It is also known as the exponential growth phase. when new connection is established the congestion window is initialized to one MSS(max. segment size) .Slow-

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

start works by increasing the TCP congestion window each time the acknowledgment is received. It increases the CWND by the number of acknowledged. This happens ,size of congestion window exponentially increase until a threshold value(ssthresh) is reached. When CWND is equal to the ssthresh TCP entered in congestion avoidance phase. In this every connection is associated with a threshold value ,the congestion avoidance algorithm takes over .and CWND size increase linearly growth (additive increase) until time out event or duplicate aknowlegde event occurs.

Congestion can be detected in two ways:

1) If congestion is detection by time out then ,

ssthresh=0.5* window size

Cwnd=1 MSS (max. size window)

2) If congestion is detection by two or three duplicate acknowledgment ,

Then ssthresh=0.5* window size

Cwnd=ssthresh

This phase is called Additive Increase and Multiplicative decrease .

2.2 Fast Recovery: There is a variation in the slow-start phase called as Fast Recovery, which uses fast retransmit followed by Congestion Avoidance. In the Fast Recovery algorithm, during Congestion Avoidance mode, when packets (detected through 3 duplicate ACKS) are not received, the congestion window size is reduced to the slow-start threshold, rather than the smaller initial value. It is assumed that if there is just a reordering of messeges there will be only one or two duplicate acks before the reorder messege is process.If more than three ack is received for the same messege sender send particular messege even before it time expires.

2.3 Fast Retransmit: After fast retransmit sends what appear to be missing segment, congestion avoidance but not slow start phase is performed. This phase is known as fast recovery algorithm.so it get high throughput under moderate congestion.

3.TCP Variants:

3.1 TCP Reno: When triple duplicate ACKs received, it will halve the congestion window, perform a fast retransmit, and enters fast recovery. If a timeout event occurs, it will enter slow-start, same as TCP Tahoe. TCP Reno is effective to recover from a single packet loss, but it still suffers from performance problems when multiple packets are dropped from a window of data.

3.2 TCP NewReno

The experimental version of TCP Reno is known as TCP NewReno. It is slightly different than TCP Reno in fast recovery algorithm. NewReno is more competent than Reno when multiple packets losses occur. NewReno and Reno both correspond to go through fast retransmit when multiple duplicate packets received, but it does not come out from fast recovery phase until all outstanding data was not acknowledged . It implies that in NewReno, partial ACK do not take TCP out of fast recovery but they are treated as an indicator that the packet in the sequence space has been lost, and should be retransmitted. Therefore, when multiple packets are lost from a single window of data, at this time NewReno can improve without retransmission time out. The retransmitting rate is one packet loss per round trip time until all of the lost packets from that window have been transmitted. It exist in fast recovery till all the data is injected into network, and still waiting for an acknowledgement that fast recovery was initiated.The critical issue in TCP NewReno is that it is capable of handling multiple packet losses in a single window. It is limited to detecting and resending only one packet loss per round -trip-time. This insufficiency becomes more distinct as the delay-bandwidth becomes greater. However, still there are situations when stalls can occur if packets are lost in successive windows, like all of the previous versions of TCP NewReno which infer that all lost packets are due to congestion and it may therefore unnecessarily cut the congestion window size when errors occur. There are some steps of congestion control for NewReno transmission control protocol.

Step 1: Initially
0<CWND<= min (1*MSS, max ())
ssthresh = max (CWND/2, 2*MSS)
Step 2: Slow Start Algorithm (Exponential Increases)
If (receive acks && cwnd< ssthresh)
CWND = CWND+1;
Step 3: Congestion Avoidance Algorithm (Additive Increase)
If (receive ACKs) {
If (CWND > ssthresh)
CWND = CWND + segsize * segsize / CWND;
Else

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

CWND = CWND+1;
}

Step 4: Congestion Detection Algorithm (Multiplicative Decrease): Fast Retransmission and Fast Recovery

If (congestion) {
If (Receive duplicate Acks 3 time or retransmission time out)
ssthresh = CWND/2;
If (Retransmission time out)
CWND = 1 MSS;
Exit and call Slow Start step;
Else /* Receive same Acks 3 time*/
CWND = ssthresh;
Exit

4.Litrature review:

L.S. Brakmo et. At [1] had proposed a new technique to improve the performance of congestion control, they calculate NewReno RTT,ABRA(adaptive backoff response approach) in TCP NewReno. An improvement (ABRA NewReno) is proposed for mobile ad-hoc networks using calculated New Retransmission Time out, to improve performance in terms of congestion control performance and apply in a MANET and compare its performance with standards of TCP Reno and TCP NewReno over wireless Adhoc network. After applying ABRA on NewReno it is analyzed under varying conditions of speed, Time and number of node. Simultaneously we calculate various performance parameters like data packet received, packet drop, packets retransmitted, throughput. From results we conclude that, ABRA NewReno performs well under the varying conditions of high density node, high node speed and pause time, because of proper utilization of time, optimal paths between nodes, optimal bandwidth exploitation and less packet delay.

Abdul Rajaque et. At [2] measure the performance of TCP Vegas and TCP Reno in heterogeneous wired network. And they studied that the fairness of TCP Reno cannot be accomplished. They discovered the new TCP congestion algorithm known as TCP NewVegas. It conclude that, TCP NewVegas give excellent features in homogeneous wired network and improve the performance in bottleneck link when it compare with TCP Reno.

Chunlei Liu et. At [3] had Approach New Proposal Based on Congestion Coherence. TCP give poor performance over unreliable wireless links where packet losses due to transmission errors. TCP enhancements proposed in the literature differ in their signaling and data recovery mechanisms, applicable network parameters, traffic scenario and locations where appropriate changes are built. In this paper, we use several approaches. Comparison with existing algorithms display this new enhancement accomplishes good performance. In this paper we conclude that, local link layer re-transmissions and signaling give the packet loss, this method eliminates the number of end-to-end retransmissions, unnecessary congestion window reductions and timeouts occurred by transmission errors, and give a high throughput performance.

R. Dunaytsev et. At [4] calculate the performance of TCP on wired and wireless networks. They proposed improved TCP variants (TCP Tahoe, TCP Reno, TCP NewReno and TCP SACK) that permit to calculate the effect of parameters(bit error rate) on TCP over both correlated and uncorrelated networks .it result give optimized startup performance ,minimum correlated losses, better performance evaluation over range of operating conditions.

5. Proposed method:

In this paper, we have explained a delay-based end-to-end congestion control algorithm, called EXT-TCP, which is an extended version of delay and acknowledge based TCP protocols like reno, NewReno etc. The plan behind this extension is to constantly estimate the round-trip-time, which ends throughput and queue oscillation when round-trip time oscillates. In this context, we instigate a simulation and an evaluation of the suggested scheme, by comparing our TCP-EXT prototype with various TCP variants in terms of throughput, fairness, stability, RTT and adaptively to changes in network.

The TCP transmission scheme that we observed previously displays us that it takes one RTT to detect packet loss. Hence one RTT cannot detect multiple losses. So we found the effective way of boosting up the reaction of TCP to detect multiple packet losses. In order to boost up TCP reaction we have modified a TCP variant, with a effective method called as EXT-TCP. This EXT-TCP variant is based on RTT. RTT is defined as the time duration it takes for a packet's to be sent plus the period of time it takes for an acknowledgment of that packet to be received. The base RTT is estimated as the minimum observed RTT for the connection Advantages of this methodology is that successive transmission that occurs is based on current RTT.As soon as our RTT

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

time will end, our congestion window will shrink and move to the slow start phase, hence this will not allow congestion to occur in the network. Whenever the congestion will occur, we will shrink our congestion window based on RTT,

Now the formula of new Target Window is as follows:

target_cwnd = ((1 - gama) x cwnd + (odd_cwnd x (base RTT / average RTT) + alpha) x gamma);
RTT = average with (1/8, 3/cwnd)
Base RTT = minimum RTT
Alpha = fairness and convergence
cwnd= congestion window
ssthresh= show start threshold
ACK=acknowledgment.
Our congestion window will shrink according to above formula.

6.Simulation Environment and Results Analysis: In this simulation there is scenario in which we use 30,50 nodes in wireless networks.In this paper we use different types of TCP variants Reno,New Reno with new devolped variant TCP-EXT based on ad-hoc wireless network of 30,50 nodes.The paper involves the measurement of different parameters like Throughput, Packet Delivery Ratio,Routingload of the network in each of the TCP variants. Finally the result achieved comparison of TCP variants with no of nodes in the network will be accessed.

Table 1.Simulation parameters

Parameter	Values
Simulation Area	1500m **x** 1500m
Application	FTP
Routing Protocol	DSDV
Node placement	Random
Data Packet	512 byte
Simulation tool	NS2
No. of nodes	50
Simulation time	100 seconds
Antenna model	Omni directional
IEEE Standard	MAC/802.11
Speed	20 mtr/sec
TCP variants	NewReno, Reno, Linux, TCP-EXT

6.1Throughput: Throughput is defined as the total amount of data received by destination node from the source node divided by the total time it takes from the destination to get the last packet and it measures in bits per second (bit/s or bps).

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

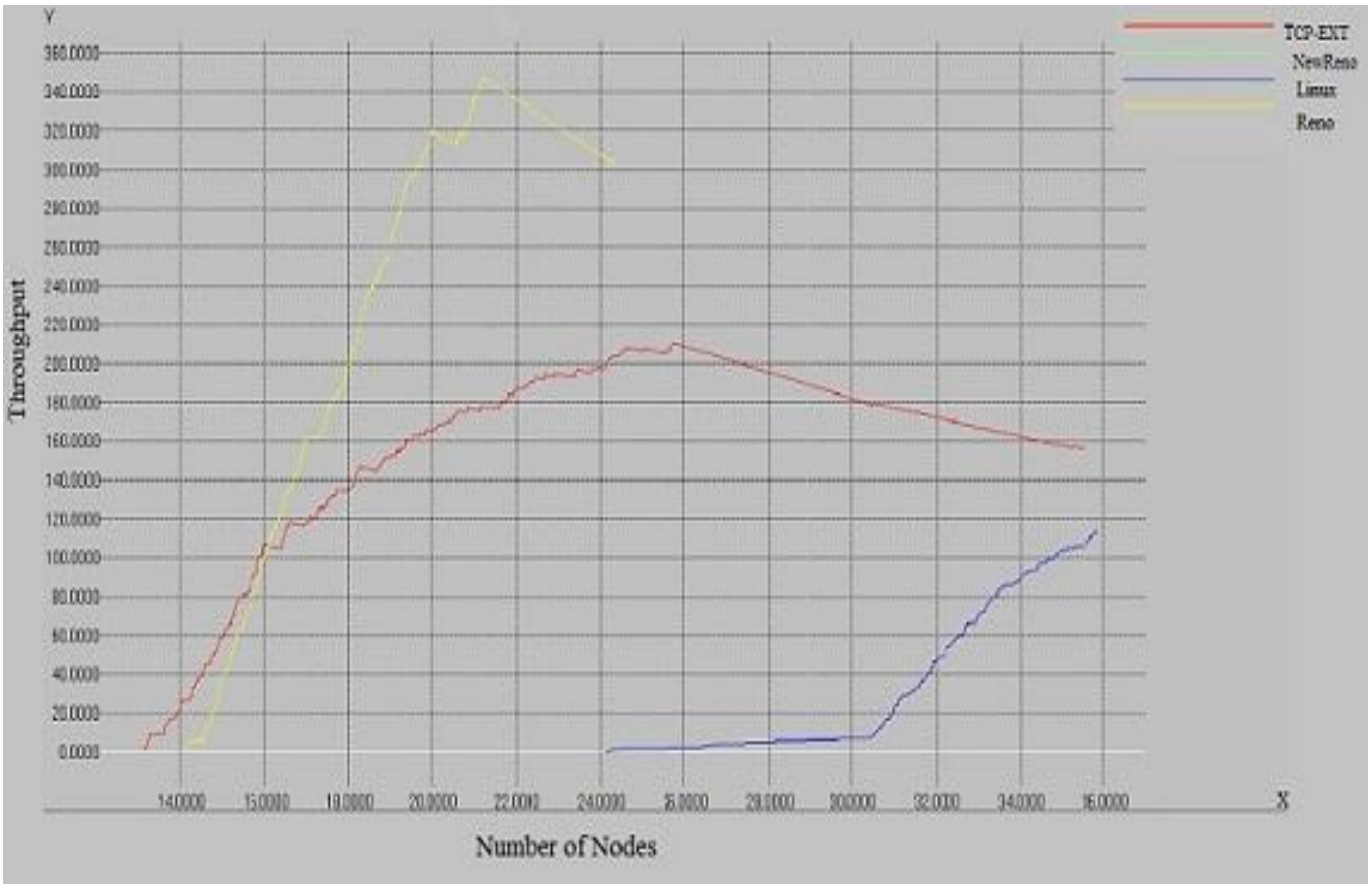

Fig 1.shows the Throughput in terms data in bytes under 30 nodes

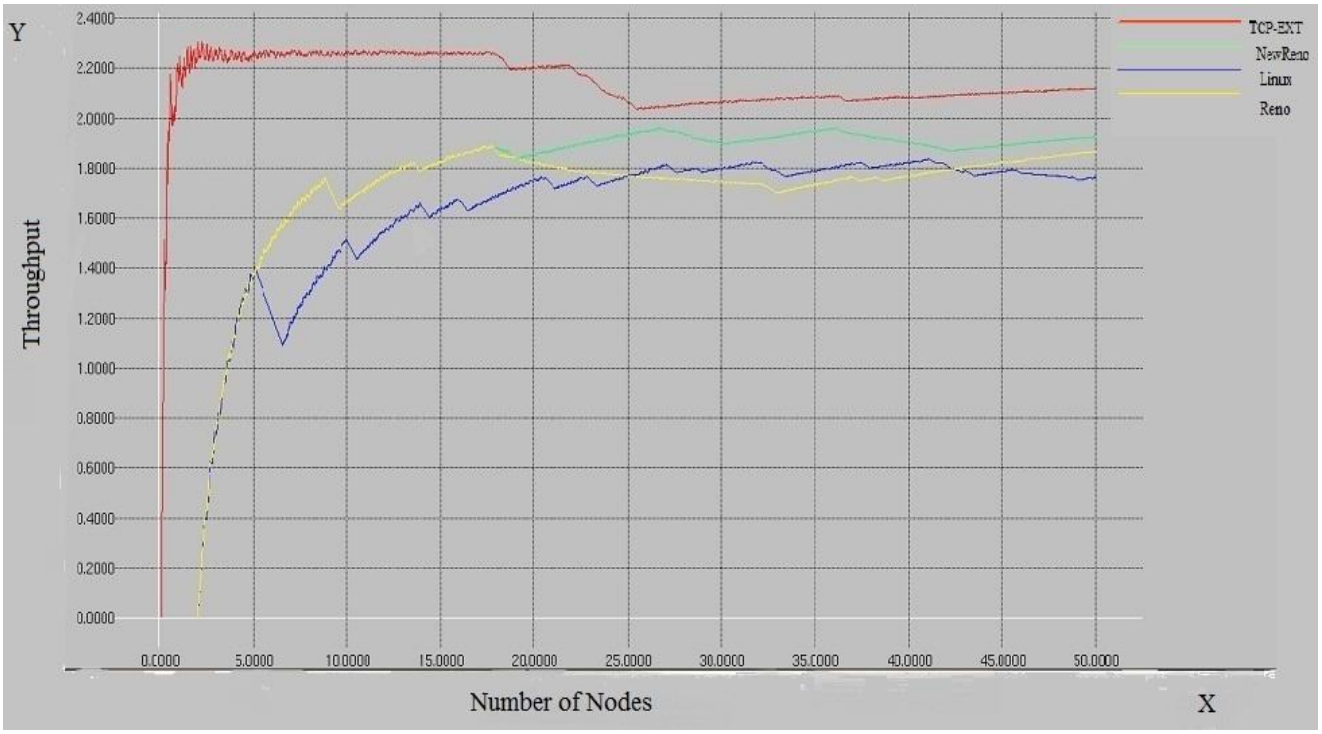

Fig 2.shows the Throughput in terms data in bytes under 50 nodes..

.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

6.2Packet Delivery Ratio: Packet delivery ratio is the ratio of total packets sent by the source node to the
successfully received packets by the destination node.

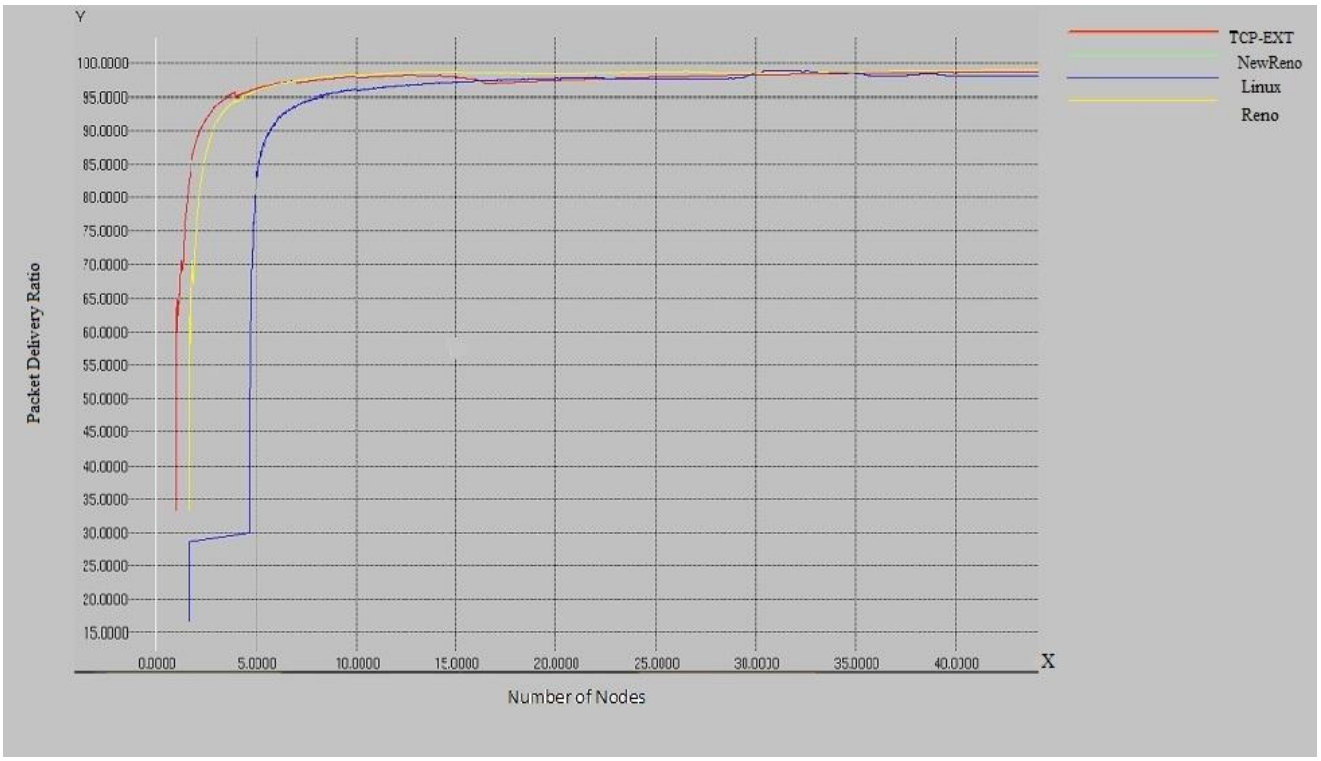

Fig 3.shows the Packet Delivery Ratio in terms of bytes under 30 nodes.

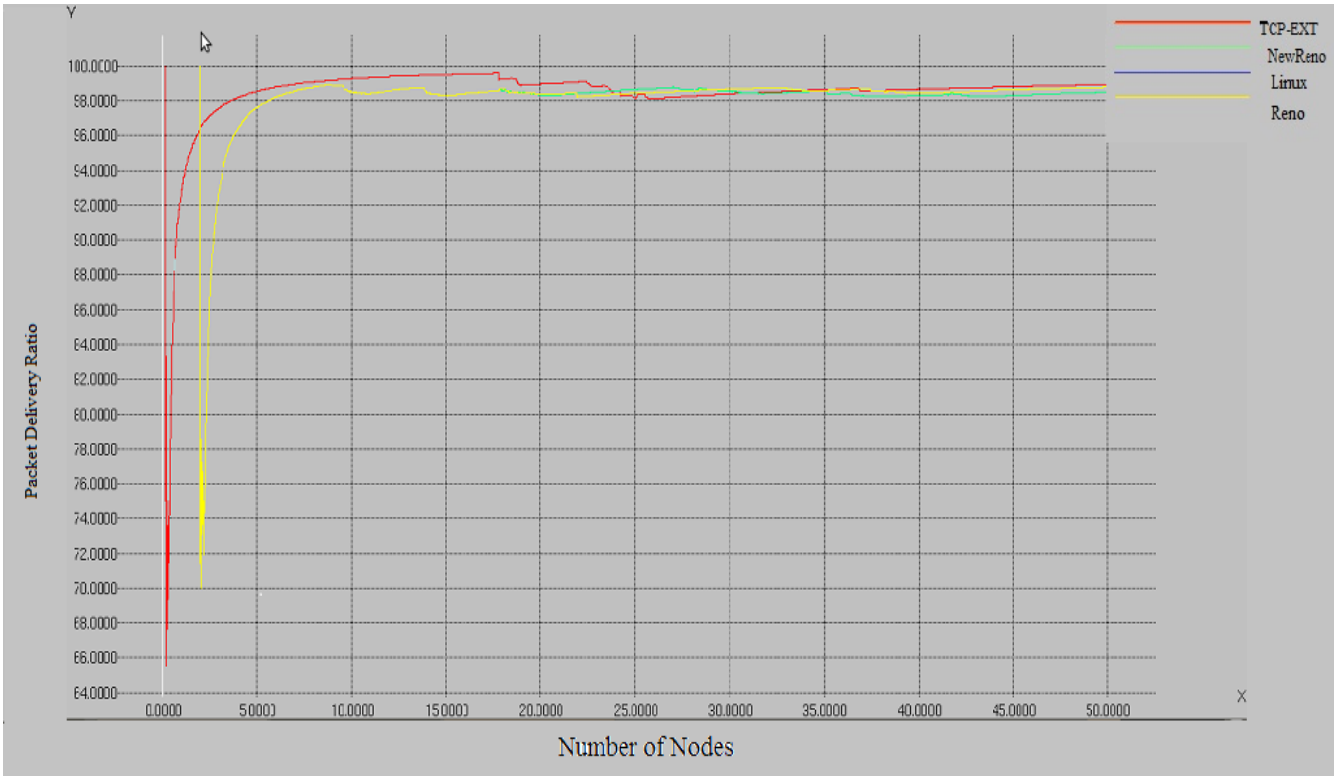

Fig 4.shows the packet delivery ratio in the terms of bytes under 50 nodes.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 3, Issue 01, January 2016
Available at http://internationaljournalofresearch.org

6.3 Normalized Protocol Overhead/ Routing Load: Routing Load is the ratio of total number of the routing packets to the total number of received data packets at destination.

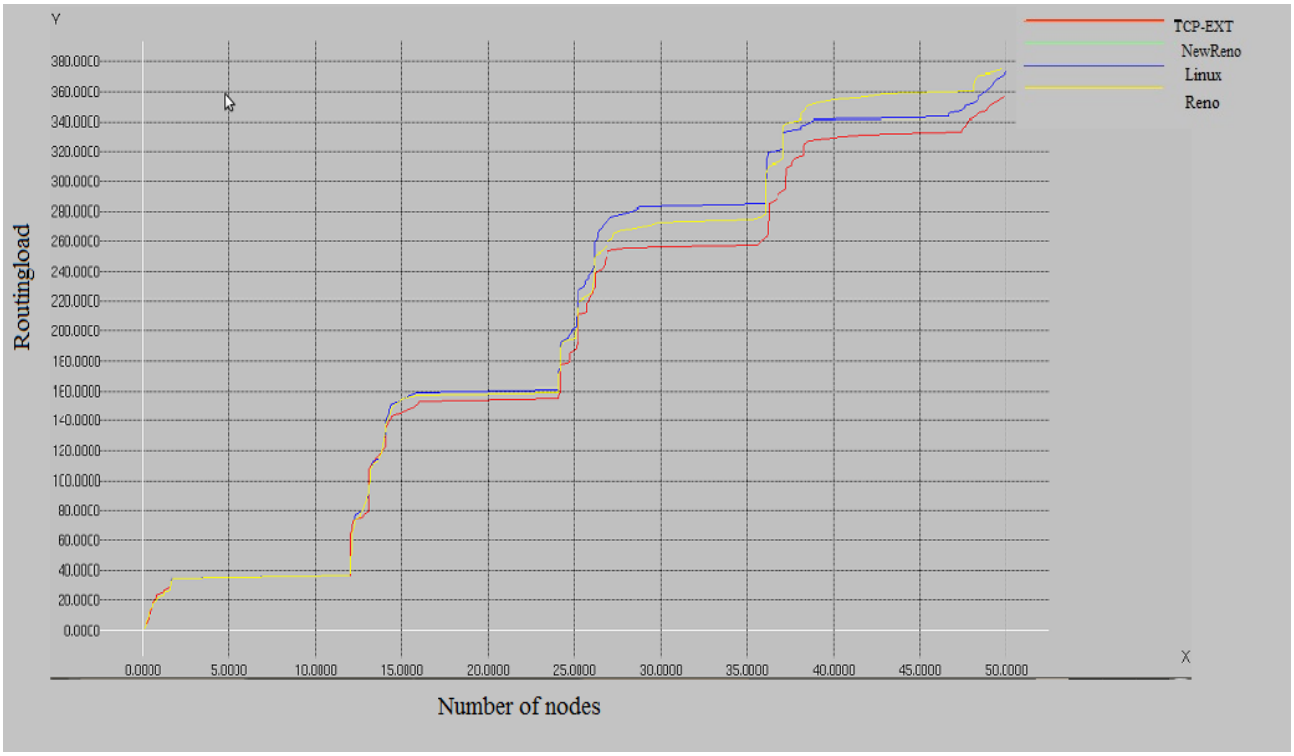

Fig 5.shows the Routingload under 30 nodes.

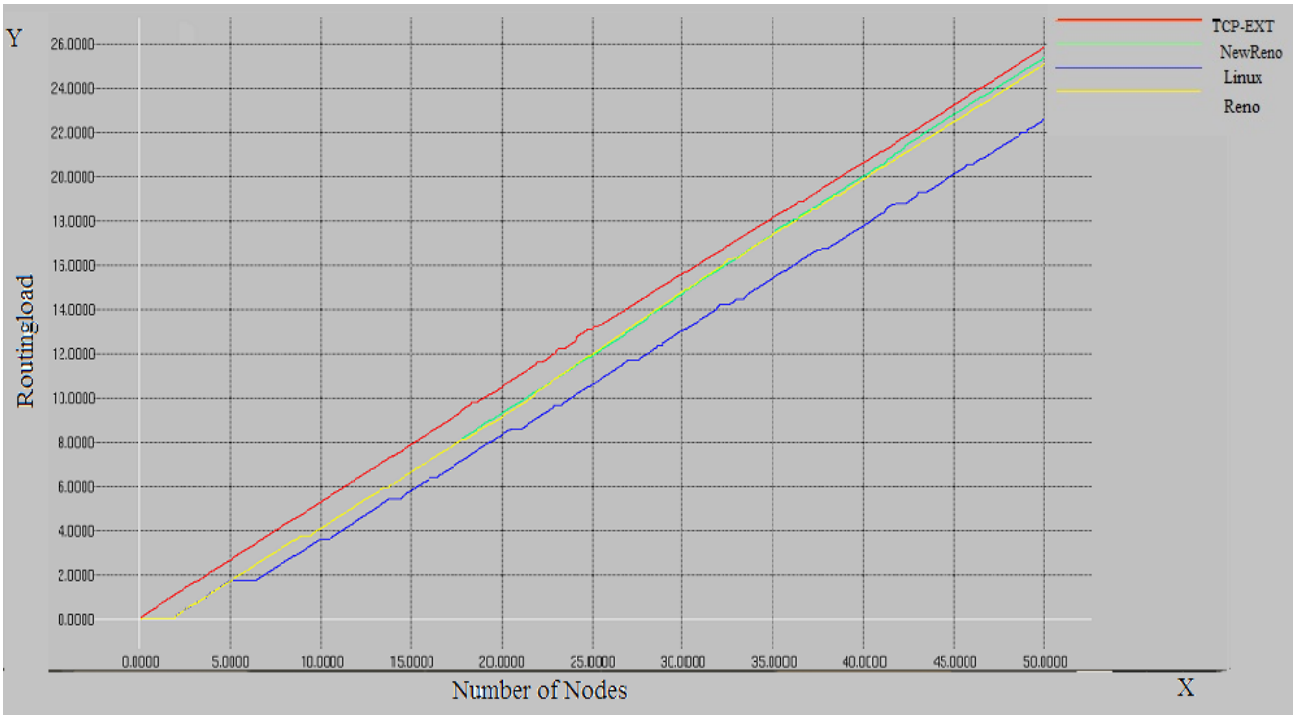

Fig6.shows the Routingload under 30 nodes.

7.Conclusion: This paper implement TCP-EXT and compare it with exiting TCP variants on NS2 simulator with different parameters Throughput,Packet Delivery Ratio,Routingload.Simulation results shown through graphs represent overall performance of TCP-EXT . In this simulation when we increase no. of nodes under such environment, TCP-EXT shows better performance compare to other variants This can also be verified from the data represented through table 1.

Effects of simulation studies on performance of TCP-EXT and other TCP variants under experimental conditions mentioned above were represented graphically.

.

Reference:

[1] T. V. Lakshman, Upamanyu Madhow "ThePerformance of TCP/IP for Networks withHigh Bandwidth-Delay Products and Random Loss" in IEEE/ACM Transaction onNetworking. vol. 5. no. 3. June 1997.

[2] Anurag Kumar "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link" in IEEE/ACM Transaction on Networking. vol. 6. no. 4. June 1998.

[3] Aldar C. F. Chan, Danny H. K. Tsang, Sanjay Gupta "Performance Analysis of TCP in the Presence of Random Losses/Errors"

[4] David B. Johnson, David A. Maltz "Dynamic Source Routing in Ad Hoc Wireless Networks", Computer Science Department, Carnegie Mellon University, 1996.

[5] Charles E. Perkins, "Ad-hoc On-Demand Distance Vector Routing", 2000

[6]. Allman, M., Paxson, V. and Blanton, E. 2009. TCP Congestion Control. RFC 5681. 2 B.S.Manoj, and C. Siva Rama Murthy 11 March 2012,"Adhoc wireless networks:architectures and protocols", Second Edition, Prentice Hall.

[7] C. Perkins, E. B. Royer and S. Das,July 2003. "AdHoc On-Demand Distance Vector (AODV) routing", RFC 3561, IETF Network Working Group,

[8] C. Siva Ram Murthy and T. Venkatesh, TCP Vegas An Analytical Approach to Optical Burst Switched Networks.

[9] Fall, K. and Floyd, S. 1996. Simulation-based Comparisons of Tahoe, Reno and SACK TCP. ACM SIGCOMM Computer Communication Review.

[10] H., et al.,Wang et al.2000, A simple refinement of slow-start of TCP congestion control, 2000, pp. 98-105.