Implementation of a Loss differentiation Algorithm for differentiation of non-congestion losses from congestion losses for improving TCP Performance over wireless networks

Nikita B. Dalwadi
ME Scholar

Abstract

Presented in this paper is the simulation based study of the TCP performance issues over wireless network. The paper includes description of various LDA algorithms with its strengths and weaknesses. The author has implemented LDA scheme named spike scheme used for discriminating between wireless and congestion losses at TCP The entire discussion open up the grey area where an effort can be put up to further improve TCP performance particularly in presence of wireless losses.

.

1. Introduction

Transport layer is one of the core layers in network protocol stack. Purpose of transport layer is to provide transparent and reliable transfer of data between end to end processes relieving upper layers from any concern of reliability. It ensures the reliable arrival of messages by providing error checking mechanisms and data flow controls. It turns the unreliable and elementary service provided by lower layers into a reliable service. It ensures that whole message arrives intact and in order at the receiver. At a moment there can be more than one application trying to send and receive data. The transport layer should be capable enough for enabling all these applications to send and receive data using the same lower-layer protocol implementation.

TCP is a reliable byte stream based connection oriented transport layer protocol.Traditional Transmission Control Protocol (TCP) is designed and optimized for wired networks under the assumptions that wired media have very low loss and propagation time [6]. It assumes that every packet loss is due to congestion on network and takes congestion control measure by reducing window size [3], [4], [5]. But in case of wireless networks acknowledgement may be delayed due to low wireless bandwidth or wireless link can go down without any warning. TCP can take this scenario as congestion and retransmits lost packets which reduces network bandwidth utilisation and leads to poor performance. Larger window size also poses a problem of competition between too many packets to get the same medium hence causing congestion and increasing end-to-end delay. There is a need to change TCP.

TCP's performance is poorly exaggerated in wireless networks [6].This performance issues are mainly due to inherent characteristics of both; wireless links as well as TCP loss recovery[6].Wireless network links are prone to, Very high BER which is unpredictable, frequent path/link breaks due to high mobility, Bandwidth constraints, Shared Broadcast Radio Channel, Resources constraints.

Packets lost under all above conditions are unfortunately considered as a sign of network congestion. Hence TCP attempts loss recovery using retransmissions at reduced rate. It reduces its window size considering it as congestion [6]. Thus frequent losses due to all above mentioned reasons prevent TCP to transmit more number of packets as much of the time is wasted in loss recovery and throughput (number of packets sent from source to destination in specific time) of the network degrades. As a result more than 60% of the network capacity remains unutilized .One of the approaches preferred by the researchers is to decouple TCP congestion control from its loss recovery for non-congestion losses using proper Loss Differentiation Algorithms. LDA are used to differentiate congestion losses from non -congestion losses.

The paper is organized as follows. Section 2 describes various related work along with description of various LDA schemes section 3 shows analysis of TCP New Reno (Simulations are carried out using Network Simulator ns v.2.35) , section 4 includes

implementation of spike scheme with TCP New Reno and section 5 concludes the paper.

2. Related work

Many schemes are classified into three basic groups, based on their fundamental philosophy: end-to-end proposals, split-connection proposals and link-layer proposals.

2.1. Link Layer Solutions

All these protocols ensure that link layer corrected all errors over wireless interface, hence eliminating need for error handling at TCP layer. Various approaches make use of either link-level automatic retransmission request or forward error correction codes or hybrid of both. Main objective of all approaches is to hide losses other than congestion from the sender TCP The main advantage of employing a link-layer protocol for loss recovery is that it fits naturally into the layered structure of network protocols. The main concern about link-layer protocols is the possibility of adverse effect on certain transport-layer protocols such as TCP. Few of proposed schemes are: Snoop TCP [13], TCP-Unaware Link Layer[13].

2.2. Split Approach Based Solutions

This approach involves splitting of TCP connection based on the wired or wireless domain. The traditional TCP can be used for the wired part and some optimized version for wireless part. The intermediate agent is known as access point (AP) and acts as proxy for MN. Thus the connection is split into two distinct connections, one between MN & AP and another between the AP & Correspondent Node. However it incurs extra protocol overhead and violates end-to-end semantics of TCP. It also complicates handoff due to state information handling at AP where the protocol is split. Few of the algorithms proposed are: Indirect TCP [13], Mobile TCP [13].

2.3. End to End Solutions

This category of techniques assumes that the entire setup is to be changed. Idea is to make sender aware that some losses are not due to congestion and, thus, avoid congestion control when not needed. Protocols under this approach make use of selective acknowledgement for error recovery. To distinguish between congestion and other losses Explicit Loss Notification is used. Advantage of this scheme is that it maintains end to-end semantics and incurs no extra overhead at AP for protocol processing or handoff. But at the same time it requires TCP to be modified. Few of suggested protocols are: Explicit Loss Notification, Wireless TCP, TCP New Reno [4] [13].
Selective Acknowledgement and TCP New Reno [4] [13].

Although a wide variety of TCP versions are used on the Internet, the current de facto standard for TCP implementations is TCP-Reno. We call this the E2E protocol, and use it as the standard basis for performance comparison The E2E-NEWRENO protocol improves the performance of TCP-Reno after multiple packet losses in a window by remaining in fast recovery mode if the first new acknowledgment received after a fast retransmission is “partial”, i.e, is less than the value of the last byte transmitted when the fast retransmission was done. This method enables the connection to make progress at the rate of one segment per round trip time, rather than stall until a coarse timeout. Along with some researchers have a proposed end to end loss differentiation algorithm which differentiates error losses and congestion losses. So using this we can differentiate the type of losses and then make changes at the sender transmission rate. Following are various base loss differentiation algorithms.The Loss differentiation decision can be obtained on TCP variable states namely congestion window (cwnd is the maximum number of packets which can be sent in one transmission), slow start threshold. The Loss differentiation decision can be obtained on TCP variable states namely congestion window (cwnd is the maximum number of packets which can be sent in one transmission), slow start threshold (ssthresh) and Round Trip Time (RTT).These class of LDAs uses delay measures to estimate the congestion status. Higher RTT values are supposed to be the effect of increased queuing delay over the network. The following first three sections describes the base algorithms and next three sections describes the enhanced LDAs[14] [15].

2.4. Biaz scheme

This scheme uses packet inter-arrival time to differentiate between the loss types. It uses the time (T_i) between the arrivals of last in sequence packet received by the receiver before a loss happened (P_i) and the first out of order packet received after the loss (P_{i+n+1}) , where n is the no. of packet loss[1] , [12].

After the arrival of P_i if P_{i+n+1} arrive at the time expected then the losses are assumed to be wireless loss and if it arrive much earlier or later then it is assumed to be congestion loss [1] [12].

2.5. Zig Zag scheme

Zig Zag [1] classifies losses as wireless based on the no. of losses n, and the difference between rtt_i (RTT of the i^{th} packet) and its mean (rtt_{imean}) (mean value of RTT for i^{th} packet). rtt_{dev} is the deviation in RTT value from previous.

A loss is classified as a wireless loss if

$n=1$ and $rtt_i < rtt_{imean} - rtt_{dev}$ (1)

OR $n=2$ and $rtt_i < rtti_{mean} - rtt_{dev}/2$ (2)

OR $n=3$ and $rtt_i < rtt_{mean}$ (3)

OR $n > 3$ and $rtt_i < rtti_{mean} - rtt_{dev}/2$ (4)

OR else the loss is classified as congestion loss.

2.6. Spike scheme

The spike scheme differentiates among degree of congestion and not explicitly differentiates congestion loss from wireless loss [1] [10]. The Relative One way Trip Time (ROTT) is a measure of the time a packet takes to travel from sender to receiver. ROTT is used to find the state of current connection. If the connection is in the spike state losses are assumed to be due to congestion otherwise losses are assumed to be wireless.

Spike state:On receipt of the packet with sequence number i, if the connection is currently not in spike state, and the ROTT for packet i exceeds the threshold $B_{spikestart}$ then the algorithm enters the spike state [1] [10]. Otherwise if the connection is currently in the spike state and the ROTT for packet i is less than the second threshold $B_{spikeend}$ the algorithm leaves the spike state [1] [10].

$B_{spikestart} = rott_{min} + \alpha * (rott_{max} - rott_{min})$ (5)

$B_{spikeend} = rott_{min} + \beta * (rott_{max} - rott_{min})$ (6)

Where, $\alpha = 0.5$ and $\beta = 0.33$

This research work uses TCP New RENO and LDA spike scheme for the improvement of TCP New Reno

3. Analysis of TCP New Reno

The simulation setup contains 3 nodes N0, N1 & N2 with (N1) as a router. N0 is the TCP and UDP sender and N2 is the receiver for both. An error model is placed in between N1 and N2. The setup is shown below in fig.3.1.

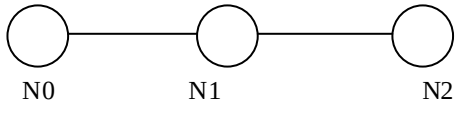


Fig.3.1simulation setup

Case 1: Without congestion and error

In this case simply a TCP runs an ftp application between N0 and N2 for 10 secs. In the graph cwnd is the congestion window which indicates the maximum number of packets which can be sent in one transmission. Throughput is the total number of packets sent in specific time.

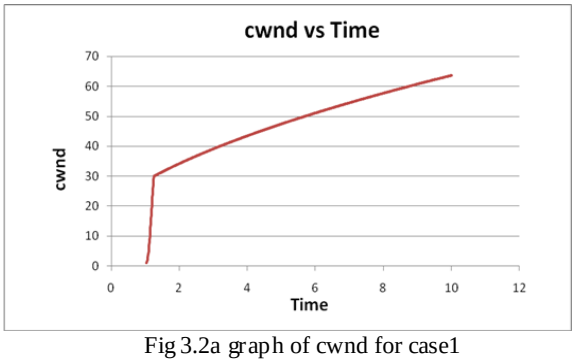


Fig 3.2a graph of cwnd for case1

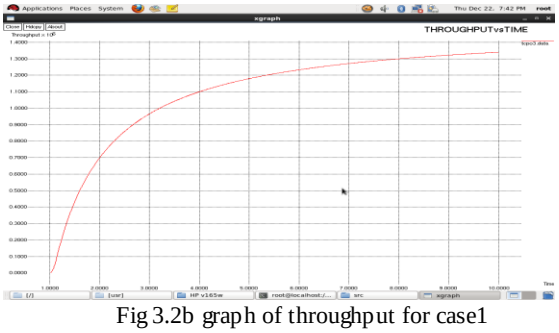


Fig 3.2b graph of throughput for case1

Case 2: With congestion

For the same above network there is a TCP connection between N0 and N2which runs an application of ftp for 10 sec a UDP between N0 and N2 runs for the same time. UDP is a real time application which introduces congestion by flooding the link with additional packets The results for throughput and congestion window (cwnd) are as shown below:

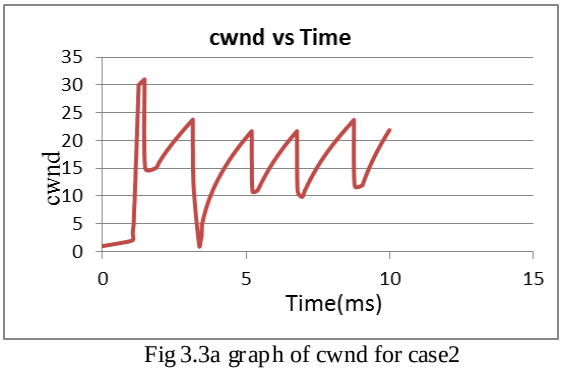


Fig 3.3a graph of cwnd for case2

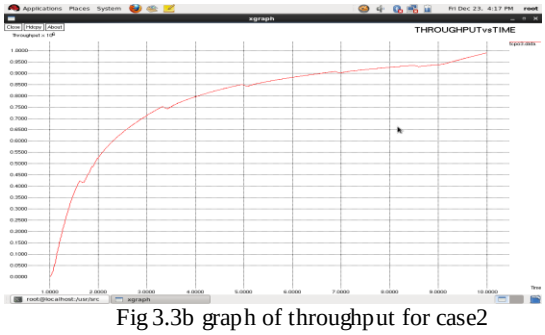


Fig 3.3b graph of throughput for case2

Case 3: With error

For the same above network TCP runs for 10 sec and an error model is placed between N1 and N2. The results for this are shown below. Because of the error model packets are dropped deliberately and so the window decreases.

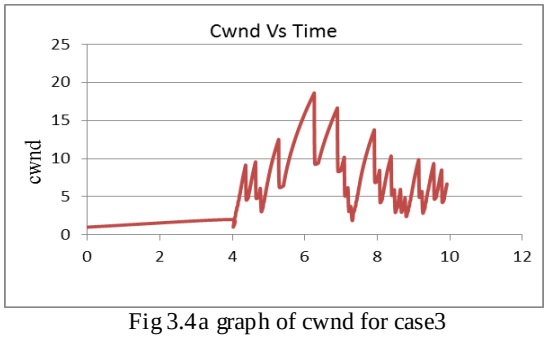


Fig 3.4a graph of cwnd for case3

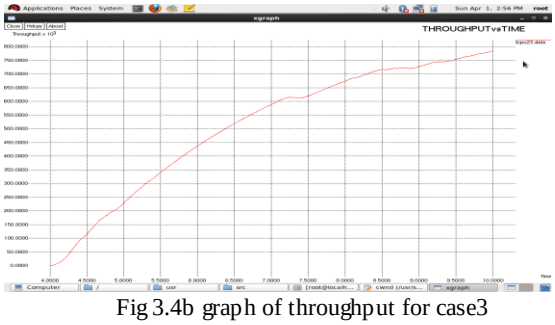


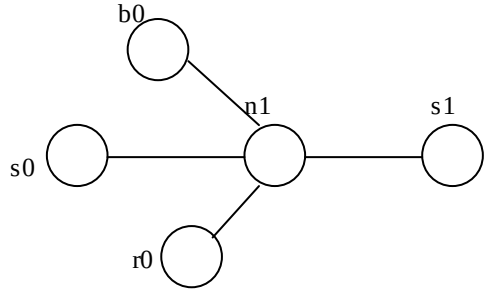
Fig 3.4b graph of throughput for case3

Summary of results

With congestion throughput of the network decreases. For the case of network in which there is only error TCP acts the same for congestion which proves that even though there is no congestion TCP decreases the congestion window and throughput decreases. Thus there is a false triggering of congestion control measures of TCP which should be eliminated for increased throughput. This is shown in the next section in implementation of LDA with TCP New Reno.

4. Implementation of LDA spike scheme with TCP New Reno

The network diagram for the simulation is shown in the figure4.1. The network consists of five nodes of which b0, so, r0 are the senders of UDP, TCP and UDP1 respectively, n 1is the router and s1 is the receiver for all senders. The data rate and delay between b0, s0 and r0 to n1 is 11mb and 20ms. And between n1 and s1 is 2mb and 100ms. Simulation run time is 50ms. An error model is placed between n1 and s1. Simulations are carried for three cases that is with error only, with congestion only and with both error and congestion together. For congestion UDP and UDP1 are run for 5 to 25ms along with TCP. For the implementation of spike scheme RTTmin (Round Trip Time) and RTTmax are calculated on arrival of each ACK from which Bspikestart and Bspikeend is calculated according to the equation 5 and 6 mentioned above.



Fig, 4.1 Simulation setup

After that on detection of packet drop if the loss is due to congestion than TCP reacts as normal TCP reducing congestion window but if it is due to error loss than the cwnd is kept unchanged if 3 dupacks are received till the recovery of the lost packets. But if the loss is error loss and there is time out than cwnd is reduced to one. The graphs of congestion window (cwnd), readings for average cwnd and average throughput are shown in the figures below. All comparison is between TCP New Reno and TCP New Reno with LDA spike scheme.

Simulation results:

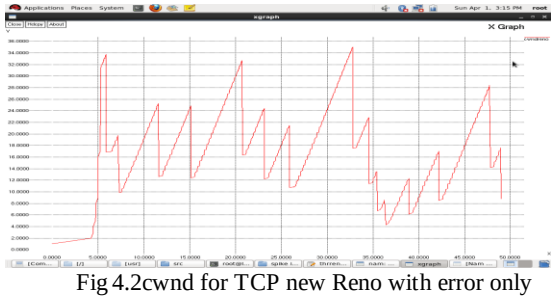


Fig 4.2cwnd for TCP new Reno with error only

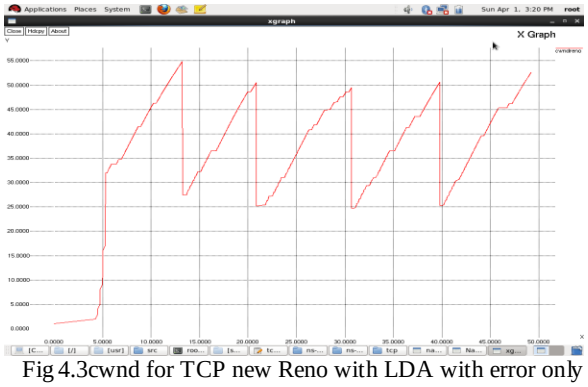


Fig 4.3cwnd for TCP new Reno with LDA with error only

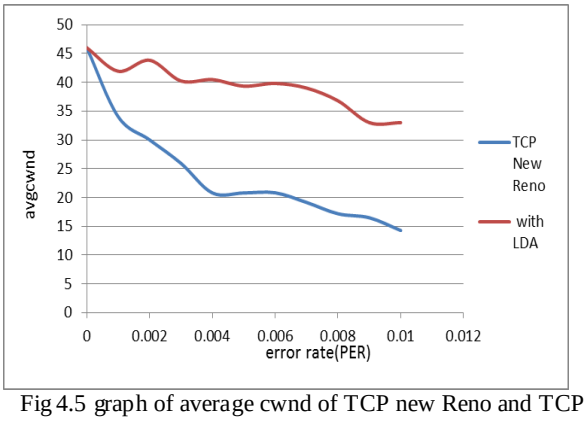


Fig 4.5 graph of average cwnd of TCP new Reno and TCP new Reno with LDA with different error rates with error only

Table 4.1 comparison of average throughput and average cwnd for TCP new Reno And TCP new Reno with LDA with error only.

Error rate	TCP New Reno		TCP New Reno with LDA	
	Average-throughput	Average-cwnd	Average-throughput	Average-cwnd
0	212188	46	212188	46
0.001	161765	34.03	201781	41.92
0.002	141022	30	201860	43.83
0.003	110962	25.93	191119	40.23
0.004	97689.3	20.79	200516	40.47
0.005	97689.3	20.79	187085	39.34
0.006	97387.8	20.82	190905	39.81
0.007	89019.4	19.14	189457	39
0.008	78312.1	17.18	168697	36.79
0.009	75241.9	16.48	153408	33
0.01	62370.4	14.27	144986	33

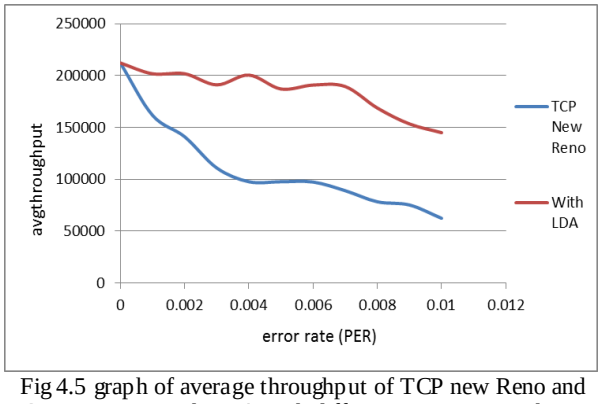


Fig 4.5 graph of average throughput of TCP new Reno and TCP new Reno with LDA with different error rates with error only

Table 4.2 Comparison of average throughput and average cwnd for TCP new Reno and TCP new Reno with LDA with congestion only.

	TCP New Reno		TCP New Reno with LDA	
	Average-throughput	Average-cwnd	Average-throughput	Average-cwnd
UDP	174659	39.9	174659	39.97
UDP 1	173940	40.04	173940	40.4
Both	149829	37.43	149829	37.43

Table 4.3 comparison of average throughput and average cwnd for TCP new Reno and TCP new Reno with LDA with both congestion error.

Error rate	TCP New Reno		TCP New Reno with LDA	
	Average-throughput	Average-cwnd	Average-throughput	Average-cwnd
0.001	156019	34.55	170016	39.03
0.002	113439	25.79	169568	38.12
0.003	114404	26.41	169837	38.16
0.004	110734	25.64	157032	35.93
0.005	81715.6	17.37	162974	34.92
0.006	73371.8	16.1	160686	33.59
0.007	67559.3	14.91	152054	32
0.008	62321.2	13.69	150901	31.32
0.009	53792.7	12.01	149012	33.2
0.01	58814.9	13.01	144798	31.18

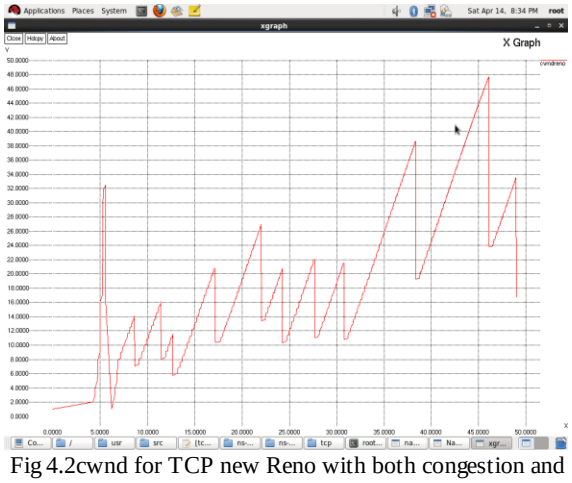


Fig 4.2cwnd for TCP new Reno with both congestion and error

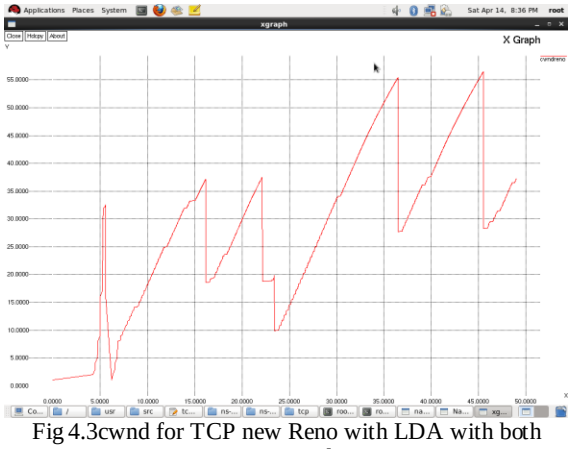


Fig 4.3cwnd for TCP new Reno with LDA with both congestion and error

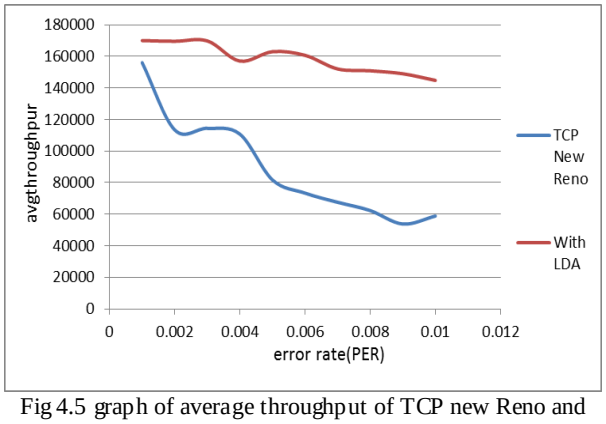


Fig 4.5 graph of average throughput of TCP new Reno and TCP new Reno with LDA with different error rates for both congestion and error

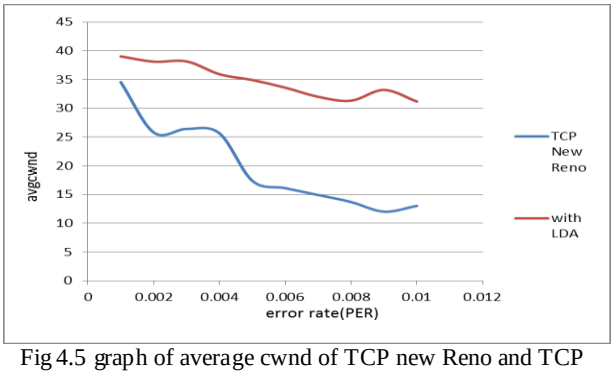


Fig 4.5 graph of average cwnd of TCP new Reno and TCP new Reno with LDA with different error rates for both congestion and error

The result shown above proves that the throughput of the network increases in presence of error with the use of loss differentiation Algorithm. Also it does not affect the normal operation of TCP for congestion. Only during the presence of error it alters the window. But doing so, that is in increasing the cwnd the number of dropped packets also increases. This is due to the congestion in the network and network as well as routers capacity. This is shown in the table below.

Table 4.4 comparison of dropped packets

TCP New Reno		TCP New Reno with LDA	
No of Dropped Packets	Average Throughput	No of Dropped Packets	Average Throughput
8	191954	10	211816
10	166443	12	209727
10	143814	13	209848
11	126190	16	209315
14	112117	18	208337
14	97000.4	20	208819
16	78049.8	22	207685
18	65829.3	24	207369
20	60328.2	25	196258

5. Conclusion

The paper presents the simulation of a network for TCP new Reno and the same with including LDA spike scheme. It has been shown that the performance of TCP new Reno improves for error losses with the use of LDA. Thus it can be used for networks having various other losses in the wireless networks to eliminate false triggering during non-congestion losses and improve the throughput of the network. But also in keeping the cwnd unchanged for error losses it increases congestion

losses and so the no. of dropped packets. Thus care must be taken while running the simulation for a longer duration.

6. References

[1] Song Cen , Pamela C. Cosman , and Geoffrey M. Voelker , "End to end Differentiation of congestion and wireless losses. Submission to ACM / IEEE transaction on networking, volume 11, issue 5, October 2003.
[2] TCP / IP Protocol suite by Behrouz A. Forouzan. Tata McGraw Hill Publishers
[3] RFC 2581 TCP congestion control April 1999.
[4] RFC 2582 TCP New Reno Modification to TCP'S Fast Recovery April 1999.
[5] Simulation based comparison of Tahoe, Reno and SACK TCP, Kevin Fall and Sally Floyd. S
[6] TCP performance issues over wireless link George Xylomenos and George c. polyzos, Athens University of Econonics and Business, Greece Perti Mahonen and Mika Saaranen, University of Oulu, centre for wireless Communications, Finland. IEEE communications Magazine 2001, Volume 39.
[7] C. Paras, J.J. Garcia-Luna-Acevez, "Differentiating congestion vs. random loss: a method for improving TCP performance over wireless links," Proceedings of IEEE WCNC, pp. 90-93.2003.
[8] "The Network Simulator – ns v.2.35.
[9] S. Biaz N. Vaidya, "Discriminating congestion losses from wireless losses using inter-arrival times at the receiver," in Proc. 1999 IEEE symposium on Application-Specific Systems and Software Engr. and Techn., pp. 10-17, Richardson, TX, March 1999.
[10] Y. Tobe, Y Tamura, A. Molano. S. Ghosh and H. Tokuda, "Achieving moderate fairness for UDP flow by path status classification," in Proc. 25th Annual IEEE conf. on Local Computer Networks (LCN 2000), pp.252-61, Tampa, FL, Nov 2000.
[11] D. Burman and I. Matta, "Effectiveness of Loss Labelling in Improving TCP Performance in Wired/Wireless networks", Boston University Technical Report, 2002.
[12] S. Biaz and N. H. Vaidya, "Distinguishing Congestion Losses from wireless Transmission Losses: a Negative Result", Proc. of IEEE 7th Int. conf. on computer communications and Networks, New Orleans, LA, USA, October 1998.
[13] Balakrishnan, H., Padmanabhan, V., Seshan, S. and Katz, R. "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Transactions on Networking, pp. 756-769, December, 1997.