Improving TCP Performance over Wireless Networks Using Cross Layer

Sathya Priya S. and Murugan K.

Ramanujan Computing Centre, Anna University Chennai
Chennai, Tamil Nadu, India
sathyapriya80@yahoo.co.in, murugan@annauniv.edu

Abstract. The rapidly increasing importance of wireless communications together with the rapid growth of high speed networks, pose new challenges to transmission control protocol (TCP). To overcome them, a wide variety of TCP enhancements has been presented in the literature with different purposes and capabilities. Cross-layering represents a perspective design principle for adapting natively wired protocols to the wireless scenario and for improving their performance. In this paper, a novel cross-layer approach (Link Layer CLAMP –TCP [L^2CLAMP-TCP]) designed for performance enhancement of TCP over a large variety of wireless networks is proposed. L^2CLAMP-TCP avoids TCP Acknowledgement (ACK) packet transmission over the wireless channel, thereby saving time, which can be utilized by the nodes for data packet delivery. The TCP Acknowledgement is generated at the base station itself. The congestion measure is also calculated at the base station based on which the receiver advertised window is calculated. The protocol performance is compared with existing TCP New Reno and TCP New Reno with CLAMP.

Keywords: Advertised Window, Automatic Repeat Request, Congestion Control, Cross-layering, Performance Enhancement.

1 Introduction

TCP is a connection oriented byte stream transport protocol. The problem of TCP over wireless networks is that wireless links have different characteristics with respect to wired ones, in terms of less reliability, fading / shadowing problems, node mobility, handoffs, limited available bandwidth and large Round Trip Time (RTTs).

In wireless networks, TCP reaction to frequent packet losses severely limits the congestion window and thus underestimates the capacity of the networks. To prevent congestion loss, an active queue management is required to avoid buffer overflow and also a fair scheduling is necessary to allocate bandwidth. The latest trend is having scheduled service at the Base Station (BS) / Access Point (AP) i.e., the traditional first come first served (FCFS) is replaced by a set of queues with a scheduler allocating the capacity between different streams.

This paper presents a novel cross-layer approach, called Link Layer Clamp- TCP (L^2 CLAMP-TCP), where the acknowledgement at the receiver is suppressed and is generated at the base station. The congestion measure is also calculated at the base station and is attached with the acknowledgement.

N. Meghanathan et al. (Eds.): NeCoM, WiMoN, and WeST 2010, CCIS 90, pp. 1–10, 2010.

2 Related Work

A useful survey of TCP enhancements for wireless networks is provided in [3]. Most work on TCP over wireless channels had focused on the issue of packet loss, its detrimental effect on TCP [4], [5], and the mechanisms at layers 1 and 2 to reduce packet loss rates [6], [7], [8], [9], [10], [11]. Retransmissions can also be handled at layer 4 [12].TCP ACK transmission over the wireless link can be avoided through local generation of ACKs at the sender node or at the base station, thus improving throughput. [1] A protocol CLAMP was suggested [2] on the receiver side and provides a separate queue at the AP. The CLAMP protocol removes the window fluctuations and achieves much better fairness than TCP New Reno.

3 TCP New Reno and CLAMP

TCP New Reno is able to detect multiple packet losses and thus is more efficient than Reno in the event of multiple packet losses. New-Reno enters into fast-retransmit when it receives multiple duplicate packets, but 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.

CLAMP algorithm controls the receiver Advertised Window (AWND) based on feedback from the access point. This algorithm runs at the mobile device and ACKs are sent back to the sender containing the new values of AWND as calculated by CLAMP. Since all versions of TCP interpret the AWND as an upper limit on the allowable window size, which is a mechanism to avoid an overflow of the receiver buffer, this provides an effective method of control provided that the AWND value is smaller than the CWND value calculated by the sender.

4 The Proposed Protocol (L^2 CLAMP- TCP)

4.1 L^2CLAMP Architecture

The proposed scheme enhances the protocol stacks of the wireless sender (or a base station /Access Point) and the receiver with L^2CLAMP agents which support ACK suppression. The network scenario taken is infra-structure based with four different senders S1, S2, S3and S4 and three mobile nodes MN1, MN2 and MN3.

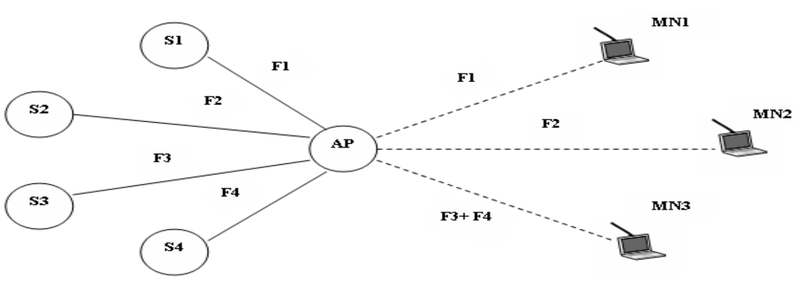


Fig. 1. Infrastructure WLAN Model

Figure 1 shows four different flows F1, F2, F3 and F4. The flow F1is between S1 and MN1 and F2 is between S2 and MN2. F3 and F4 start from sources S3 and S4 respectively and reaches the same mobile node MN3 (shared flow).

The L^2CLAMP agent suppresses the outgoing L^2CLAMP -TCP ACKs at the receiver side and generates them locally at the sender or base station. The basic idea behind approach is to shift TCP ACK generation point from mobile receiver to the base station. L^2CLAMP -TCP requires implementation of a software module, called L^2CLAMP agent, inside BS / AP protocol stack above the link layer. The L^2CLAMP agent sniffs the ingress traffic from the fixed network assuming to have access to the network and transport layer headers. Whenever a TCP data packet is detected, the L^2CLAMP Agent in the BA / AP, performs service rate, congestion measurement, received rate and calculates the AWND value, stores flow-related information such as flow sequence number carried by the packet, port numbers, ACK flag and ACK sequence number, along with the AWND value. L^2CLAMP agent calculates the desired size of the AWND window based on the values of congestion measure, received rate and RTT. To calculate congestion measure, a separate Sync packet is sent from the base station to the receiver and the acknowledgement for the Sync is received back by the sender, based on which the transmission time for each packet is calculated. This is done before actual TCP packet transmission. The parameters used for calculation of congestion measure is b, q,α μ_c

$$p(q) = (bq - \alpha) / \mu_c \qquad (1)$$

where b = 1 for safer purpose. τ is the parameter (fixed size in bytes) used to calculate the value of AWND and α is used to calculate the congestion price signal. Similarly α is free to be chosen by the AP, but is fixed once chosen.

On the receiver side, module referred as L^2CLAMP client in Figure 2 silently drops all standalone non-duplicate TCP ACK packets. The generated TCP ACK packet is then forwarded to the sender. The AP maintains a separate queue for each receiver and acts as a router, routing the packets to the destined MN. The TCP flows destined for the respective MN are put into the same queue.

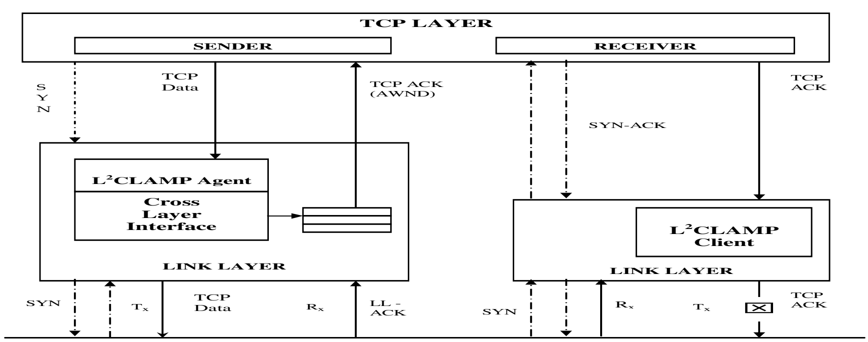


Fig. 2. Architecture of the Proposed System

In the scenario taken, a separate queue is maintained for flows F1 and F2 as they are destined to different mobile nodes MN1 and MN2. The flows F3 and F4 share a same queue as they are destined to the same mobile receiver MN3. A scheduler incorporated in the access point takes care of scheduling the two flows. The scheduler is based on the principle of multi-user diversity. At each slot, a new scheduling decision is made, and the general preference is for scheduling a mobile that is in a good channel state. At each slot, the scheduler picks the queue i with the largest utility that has data to send in this slot. The utility U_i is calculated as

$$U_i = \mu_i / r_i \tag{2}$$

μ_i is the current rate for the mobile 'i' at the beginning of the slot and r_i is the exponential moving average of the rate obtained by mobile- i with an averaging time constant of 100 ms.

4.2 Algorithm

Receive New Data Packet from Network
If (Data Packet == TCP) then
Generate a TCP Acknowledgement with the respective fields.
Store it in the buffer.
Calculate the Congestion measure.

$$p_s(q)= \max((q-a)/\mu_c , 0) \tag{3}$$

Calculate the AWND window size

$$\mu^*(t_k) = \frac{\sum_{i=k=\alpha}^{k} S_i}{(t_k - t_{k-1})} \tag{4}$$

$$\Delta w(t_k) = \frac{r - p(q(t_k))\mu^*(t_k))}{d^*(t_k)}(t_k - t_{k-1}) \tag{5}$$

if ($\Delta w(t_k) < 0$) or (3 Duplicate ACK's)

Stop Slow start

Set $w(t_k+1) = w(t_k)/2$

endif

$$AWND(t_k) \cong \min(w(t_k), AWND) \tag{6}$$

Add the information to the Stored ACK Buffer.
Forward the TCP Data Packet for Transmission
Wait for Link Layer Acknowledgement
if (LL-ACK received==yes) then
Pop the generated ACK from the buffer and transmit to the sender
else
Drop the ACK generated stored in the buffer.
end if
else
Forward the packet to the wireless node.
end if

where q is the current value of moving average, t_k is time instant of k^{th} packet when received by MN, $\Delta w(t_k)$ is the change in window size, α, β are smoothing factors, AWND (t_k) is the actual Window size, d^* is the RTT of the flow in seconds(propagation delay + queuing delay), μ_c is the service rate of a queue, d is the propagation delay and q(t) is calculated as τ / b (amount of flow in bytes).

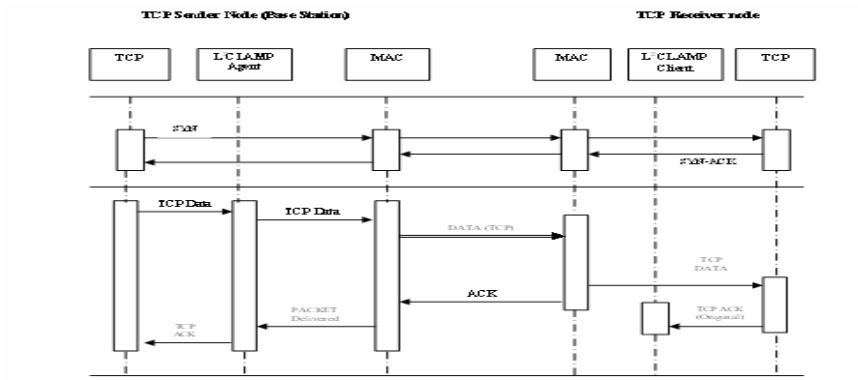


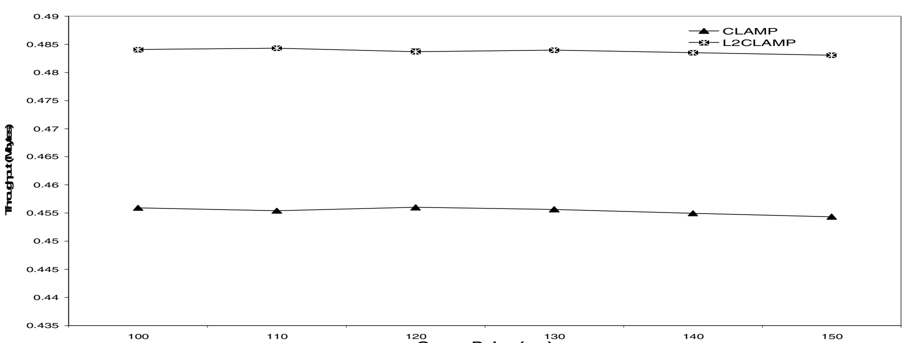
Fig. 3. Sequence of Process for Packet Delivery

5 Performance Evaluation

To observe how TCP performance is influenced by our L^2CLAMP approach, it is compared with existing TCP New Reno and TCP New Reno with CLAMP. The simulation is carried out with Network Simulator (ns-2) with a transmission range of 75 m, packet size of 250 to 1250 bytes, queue delay of 100 to 150ms, simulation time of 50 to 200s, bandwidth of 0.5Mbytes/s and a propagation delay of 20ms.

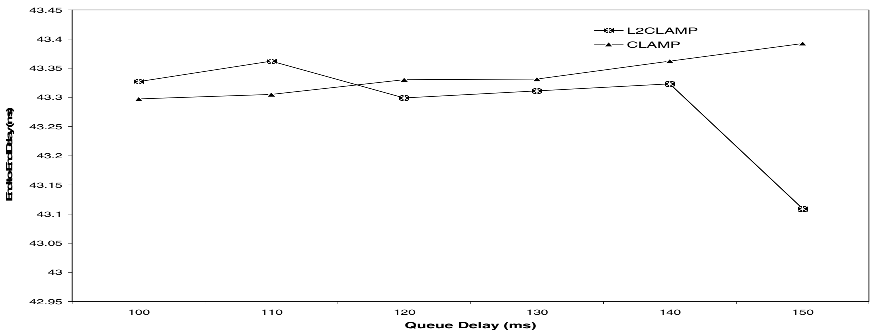
5.1 Analysis by Varying Queue Delay

Figure 4. shows that the queue delay is varied between 100ms and 150ms with a packet size of 1250 bytes and a simulation time of 200 secs.

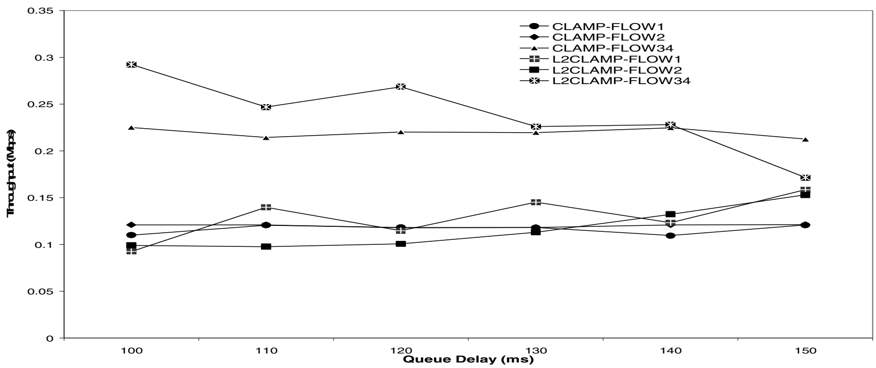


(a)

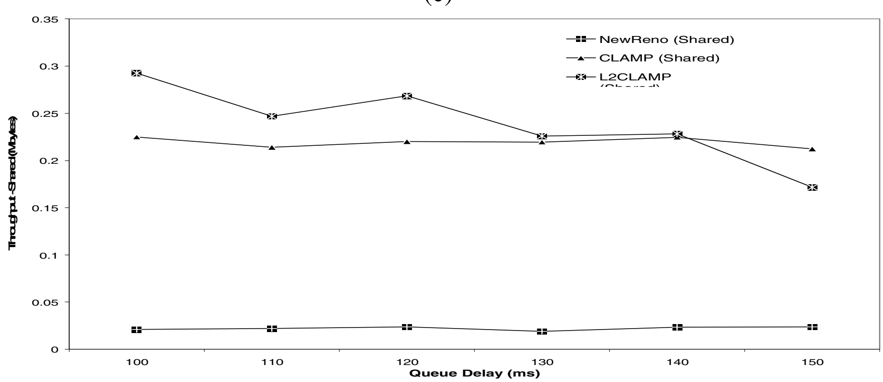
Fig. 4. Queue delay (ms) with respect to (a) Throughput (Mbps), (b) End-to-end Delay (ms), (c) Throughput per flow (Mbps), (d) Throughput (shared)



(b)



(c)



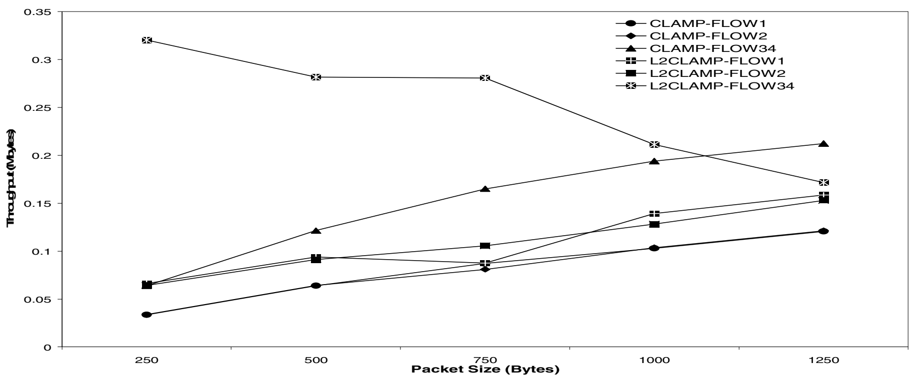
(d)

Fig. 4. (*continued*)

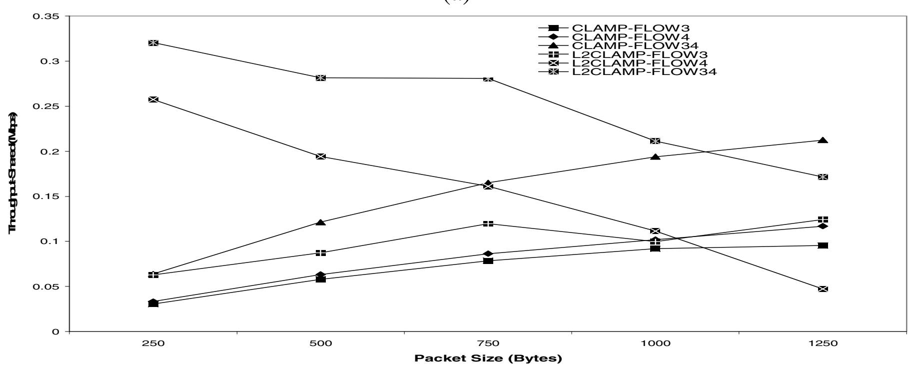
The inference made from Figure 4 is that throughput of L^2CLAMP is approximately 6% higher than that of CLAMP. The end-to-end delay of L^2CLAMP is very less when compared to that of CLAMP. The throughput for individual flows F1 and F2 as well as the combined flow F3+F4 of L^2CLAMP shows considerable improvement in performance when compared to that of CLAMP. The throughput of shared flow for CLAMP is approximately 8 times higher when compared to that of New Reno and performance of L^2CLAMP is nearly twice better than that of CLAMP.

5.2 Throughput Analysis by Varying Packet Size

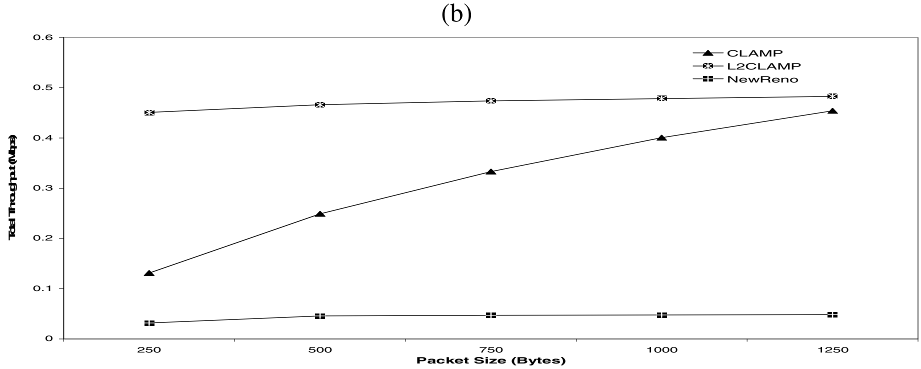
Figure 5 shows that the packet size is varied between 250 bytes and 1250 bytes with a Queue delay of 150 ms and a simulation time of 200 sec.



(a)



(b)



(c)

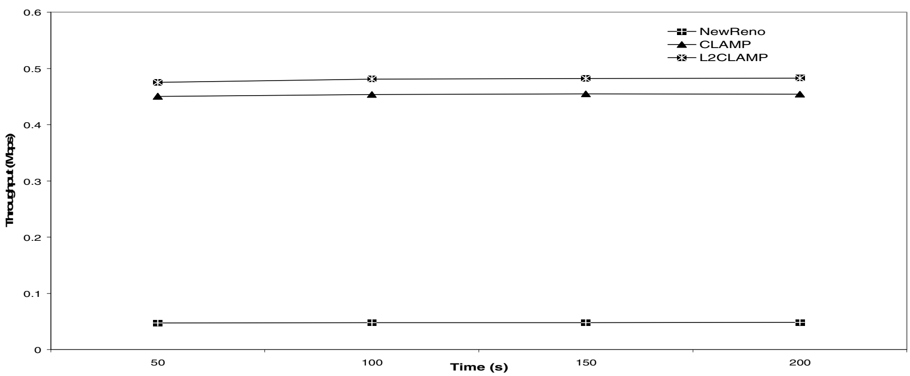
Fig. 5. Packet Size with respect to (a) Throughput per flow (Mbps), (b) Throughput (shared), (c) Throughput (Mbps)

It can be observed from Figure 5 that the throughput of CLAMP is approximately twice higher than New Reno and that of L^2CLAMP is nearly 3.5 times higher than CLAMP as we vary the size of the packet. The throughputs in case of individual flows as well as the shared flows of L^2CLAMP show considerable performance improvement when compared to CLAMP.

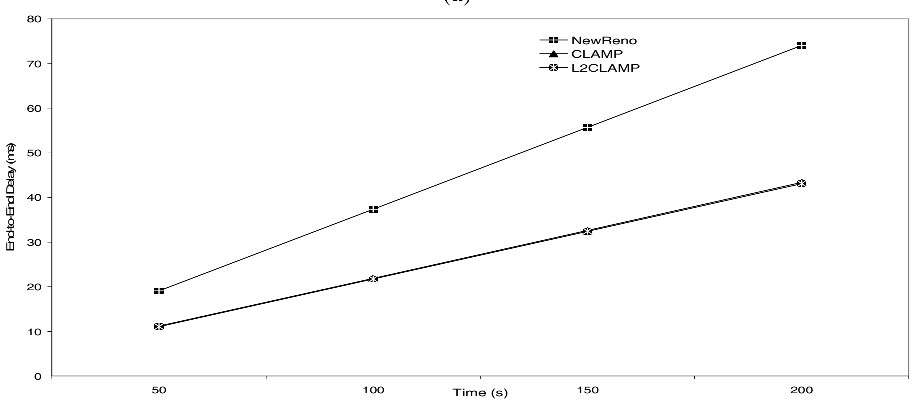
5.3 Analysis by Varying Simulation Time

Figure 6 shows that the duration of simulation is varied between 50 secs and 200 secs with a Queue delay of 150 ms and packet size of 1250 bytes.

From figure 6, it can be noted that the throughput of L^2CLAMP is nearly 8 to 9 times higher when compared to that of New Reno and is approximately twice higher than that of the CLAMP. The end-to-end delay of L^2CLAMP is nearly half that of New Reno. The queue performance is also studied for all the three different protocols with varying simulation time.

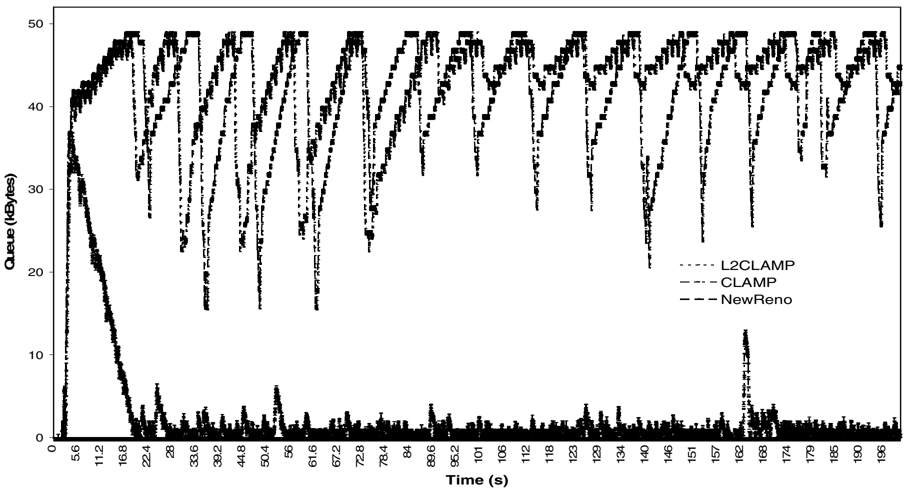


(a)

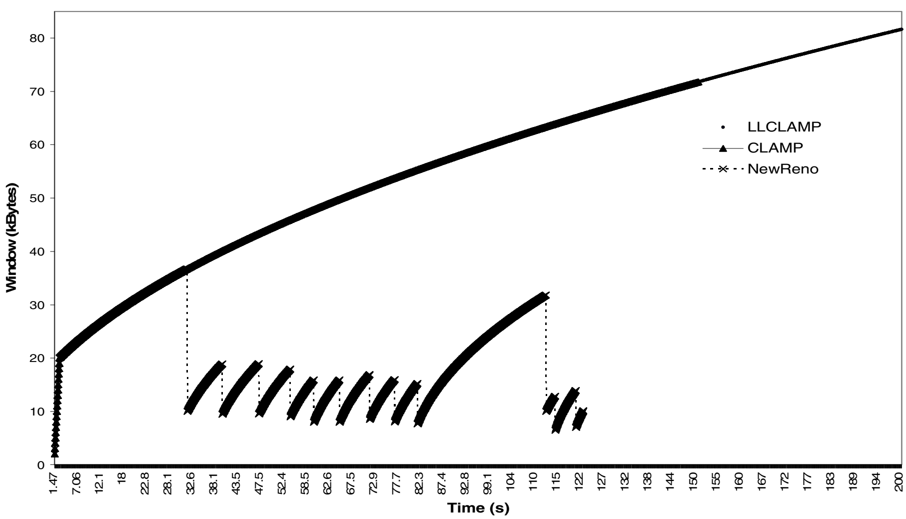


(b)

Fig. 6. Simulation time with respect to (a) Throughput (Mbps), (b) End-to-end delay (ms) (c) Queue (Kbytes), (d) Window size (Kbytes)



(c)



(d)

Fig. 6. (*continued*)

6 Conclusions and Future Work

This paper presented a novel yet generic approach for performance enhancement of TCP over wireless networks. Performance improvement comes from cross-layer optimization. The proposed solution, L^{2}CLAMP, avoids TCP ACK transmission over the wireless link through local generation of ACKs at the sender node or at the base station. The congestion measure is also calculated at the base station based from which the receiver advertised window is calculated. The protocol performance is compared with existing TCP New Reno and TCP New Reno with CLAMP. The performance of L^2CLAMP is better than the CLAMP whose performance is observed to be much better than the TCP New Reno.

A dynamic switch-over of the modules on-demand can be provided depending on the traffic load of the Access Point. The performance of the protocols can also be studied in an error-prone environment by the introduction of error models.

References

1. Kliazovich, D., Granelli, F., Gerla, M.: Performance improvement in wireless networks using cross-layer ARQ. Computer Networks 51, 4396–4411 (2008)
2. Andrew, L.L.H., Hanly, S.V., Mukhtar, R.G.: Active Queue Management for Fair Resource Allocation in Wireless Networks. IEEE transactions on Mobile Computing 7(2), 231–246 (2008)
3. Andrew, L.L.H., Hanly, S.V., Mukhtar, R.G.: CLAMP: Maximizing the Performance of TCP over Low Bandwidth Variable Rate Access Links. Technical report, Univ. of Melbourne (2004)
4. Eryilmaz, A., Srikant, R.: Fair Resource Allocation in Wireless Networks Using Queue-Length-Based Scheduling. In: Proc. IEEE INFOCOM, vol. 3, pp. 1794–1803 (March 2005)
5. Andrew, L.L.H., Hanly, S.V., Mukhtar, R.: CLAMP: A System to Enhance the Performance of Wireless Access Networks. In: Proc. IEEE Global Telecomm. Conf. (GLOBECOM 2003), vol. 7, pp. 4142–4147 (December 2003)
6. Andrew, L.L.H., Hanly, S.V., Mukhtar, R.G.: CLAMP: Differentiated Capacity Allocation in Access Networks. In: Proc. 22nd IEEE Int'l. Performance Computing and Comm. Conf. (IPCCC 2003), April 2003, pp. 451–458 (2003)
7. Xu, S., Saadawi, T.: Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? IEEE Communications Magazine 39, 130–137 (2001)
8. NS-2 simulator tool home page, `http://www.isi.edu/nsnam/ns`
9. Kliazovich, D., Granelli, F.: Cross-layer Congestion Control in ad hoc Wireless Networks. Journal on Ad Hoc Networks, 687–708 (August 2005)
10. Sardar, B., Saha, D.: A Survey of TCP Enhancements for Last- Hop Wireless Networks. IEEE Comm. Surveys 8, 20–34 (2006)
11. Chen, X., et al.: A Survey on Improving TCP Performance over Wireless Networks. Network Theory and Applications 16 (July 2006)
12. Lakshman, T., Madhow, U.: The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. IEEE/ACM Trans. Networking 5, 336–350 (1997)