Communications and Network, 2014, 6, 49-60
Published Online May 2014 in SciRes. http://www.scirp.org/journal/cn
http://dx.doi.org/10.4236/cn.2014.62007

Link Layer Correction Techniques and Impact on TCP's Performance in IEEE 802.11 Wireless Networks

Purvang Dalal[1], Mohanchur Sarkar[2], Kankar Dasgupta[3], Nikhil Kothari[1]

[1]Department of Electronics and Communication, Dharmsinh Desai University, Nadiad, India
[2]Indian Space Research Organization, Ahmadabad, India
[3]Indian Institute of Space Science and Technology, Thiruvanthapuram, India
Email: pur_dalal.ec@ddu.ac.in, nil_kothari@ddu.ac.in, msarkar@sac.isro.gov.in, ksd@iist.ac.in

Received 28 November 2013; revised 30 December 2013; accepted 20 January 2014

Abstract

TCP performance degrades when end-to-end connections extend over wireless links which are characterized by high *Bit Error Rate* and intermittent connectivity. Such degradation is mainly accounted for TCP's unnecessary congestion control actions while attempting TCP loss recovery. Several independent link loss recovery approaches are proposed by researchers to reduce number of losses visible at TCP. In this paper we first presented a survey of loss mitigation techniques at wireless link layer. Secondly performance evaluation for TCP through Type 0 *Automatic Retransmission Request* mechanism in erroneous *Wireless LAN* is presented. In particular, simulations are performed taking into account the wireless errors introduced over IEEE 802.11 link using a well-established 2-*State Markov* model. TCP performance is evaluated under different settings for maximum link retransmissions allowed for each frame. Simulation results show that, link retransmission improves TCP performance by reducing losses perceived at TCP sender. However, such improvement is often associated with adverse effect on other TCP parameters that may cost a lot in return under extreme network conditions. In this paper an attempt is made to observe impact of link retransmissions on the performance of multiple TCP flows competing with each other. The analysis presented in this paper signifies the scope for maximizing TCP's throughput at the least possible cost.

Keywords

Wireless LAN, Cross-Layer Scheme, Medium Access Control, Round Trip Time, Bandwidth Estimation

How to cite this paper: Dalal, P., Sarkar, M., Dasgupta, K. and Kothari, N. (2014) Link Layer Correction Techniques and Impact on TCP's Performance in IEEE 802.11 Wireless Networks. *Communications and Network*, **6**, 49-60.
http://dx.doi.org/10.4236/cn.2014.62007

1. Introduction

In recent scenario, the IEEE 802.11 [1] or 3G/4G wireless networks represent a significant milestone in the provisioning of "*anywhere - anytime*" *Internet* connectivity to the wandering users [2] [3]. Together, *Transmission Control Protocol* (TCP) [4] has remained as the dominant transport layer protocol for majority of *Internet* applications [2] [5]. Unlike wired links, wireless links have high, bursty and random errors due to atmospheric conditions, terrestrial obstructions, fast and multi-path fading, active interference, mobility and resource constraint [3] [6]. Consequently, significant amount of efforts [7] [8] have been devoted to the provisioning of reliable TCP delivery for a wide variety of applications over different wireless infrastructures.

Impact of the losses on TCP performance has been characterized in [9] using the following Equation (1). From the view of TCP, the throughput is inversely proportional to the packet losses perceived by TCP sender.

$$B\left(p\right) \approx \left(\frac{cwnd}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 \min\left(1, 3\sqrt{\frac{3bp}{8}}\, p\left(1+32p^2\right)\right)} \right) \tag{1}$$

Nevertheless, when the losses are seen at TCP sender due to reasons other than network congestion, end-to-end throughput is compromised due to its loss recovery attempts with detrimental transmission rate. Therefore it is advocated to endeavor initial loss recovery at link layer (link recovery) and to attempt end-to-end TCP recovery [10] [11], only if inescapable. Attempts to make wireless links resemble wired ones for high level protocols are reflected in several approaches operating at a link layer.

In this paper, some of the prominent approaches made at wireless link layer for improving TCP's performance are discussed first. In the modern *Wireless LAN* (WLAN), due to the technology progress, the physical channel condition is comparatively good enough to provide low *Packet Error Rate* (PER) and consequently higher data-rate [2] [12]. On the other hand, to guarantee the correctness of control information, the control data rate increases very slowly (e.g. in IEEE 802.11 g data-rate is raised up to 54 Mbps and control data-rate is increased up to 6 Mbps only). In this situation, the overhead of control information including the link layer acknowledgement (ACK) increases significantly. The above grounds for very low efficiency of *Automatic Retransmission reQuest* (ARQ) protocol. As a result, the effect of link recovery in such networks should be thoroughly analyzed. In earlier work [13], the performance of a series of ARQ mechanisms are analyzed in the link layer without considering the TCP layer effect. In this work, a set of simulations are performed and results are analyzed to show impact of link recovery on TCP's performance. Analysis presented in this paper not only addresses the merits and demerits of link recovery attempts but also give a new direction towards enhancing TCP's performance further.

Our results show that with light errors on the IEEE 802.11 link, better TCP performance is achieved using link layer attempts. Since, link recovery attempt increases the *Round Trip Time* (RTT) estimate at TCP, it may adversely effects on the TCP performance by deteriorating the transmission rate at TCP sender (It is controlled using TCP's internal parameters *congestion window* (*cwnd*) and RTT). In fact, even with the use of link recovery, sub-optimum TCP's performance is seen exclusively in the extreme network environments; wherein total loss recovery is not attained at link layer. The rest of the sections are as follows. Section 2 discusses the pure link layer proposals designed to mitigate wireless errors. Section 3 provides a brief summary of link recovery mechanism over IEEE 802.11 links and Section 4 discusses the simulation results to show its influence on the end-to-end TCP's performance in wireless scenario. Section 5 concludes this paper.

2. Pure Link Layer Enhancements

Addressing link errors near the site of their occurrence appears intuitively attractive for several reasons.

1) Link layer approaches are likely to respond more quickly to changes in the error environment and therefore local recovery over a point to point link is more efficient than end-to-end TCP recovery [10].

2) Since, local recovery mechanism commonly operates on exactly the link that require its rendering, the deployment of new and existing wireless link protocol is more feasible than applying novel transport layer solution.

3) The link recovery approaches do not violate the modularity of the protocol stack [13] and in this context

they are different from *Cross-layer* [14] approaches.

Various approaches have been proposed at link layer to optimize the performance of TCP in wireless networks. These approaches are broadly classified into two groups on the basis of the awareness of the transport layer protocol; (a) TCP *unaware* approaches and (b) TCP *aware* approaches.

2.1. TCP *Unaware* Approaches

IEEE 802.11 and Cellular networks employ TCP *unaware* link and physical layer mechanisms with an objective to reduce the PER to a level that would not cause significant performance degradation at TCP. In the above networks, physical layer achieves high coding gain with the use of convolution and turbo coding [15]. Additionally interleaving provides time diversity as a further safeguard against burst errors. The most remarkable TCP *unaware* link layer implementations in the above wireless technologies uses ARQ mechanism and packet scheduling to reduce effect of losses on TCP's performance. For example, *Selective Repeat* ARQ with scheduling protocols such as RLP and RLC are used in 3G1X and UMTS, respectively [15]. On the other side, IEEE 802.11 uses *stop and wait* ARQ [13] alone to recover from transmission losses locally. ARQ is a closed-loop mechanism that requires feedback and retransmission and is invoked when packets containing bit errors are discarded. Link ARQ consumes additional network resources only when a packet is retransmitted. The mechanism generally operates more efficiently for low bit rates. On a downside, the above mechanisms may cause delay variability and out-of-order delivery of packets at TCP. This can result in duplicate retransmissions from the two layers and hence inefficient utilization of the network capacity. An undesirable side effect of ARQ is that it may interfere with independent TCP mechanisms. *Forward Error Correction* (FEC) is a popular error mitigation mechanism for detection and recovery of transmission losses [16] without retransmissions, which is critical for lossy links exhibiting long delays. Unlike ARQ, FEC doesn't interfere with TCP mechanisms. FEC suffers from dead-weight overheads in favorable conditions, resulting in a waste of limited bandwidth. Furthermore FEC requires additional resources (processor, memory) and power consumption.

Based on the recommendations made by IETF and 3GPP, most of the link layer techniques for wireless lossy links, involve making the link layer reliable using hybrid FEC/ARQ [17]. Formerly a good summary of hybrid FEC/ARQ techniques for the link layer is provided in [13]. In a Type 0 scheme, recovery for losses is offered only through ARQ. This technique is applicable to links having losses subject to infrequent interference. In Type I technique, initial transmission is protected using FEC. If this fails, ARQ is used to repeat the transmission. The main characteristic of Type II schemes is that ARQ is used first followed by FEC. Typically the initial transmission has error detection built in but no error correction. Though hybrid ARQ/FEC is better than either FEC or ARQ alone, its performance also significantly degrades for higher loss rates despite unreasonably high amounts of ARQ retries, fragmentation of IP packets, FEC overhead and buffering [15]. *Asymmetric Reliable Mobile Access In Link Layer* (AIRMAIL) and *Transport Unaware Link Improvement Protocol* (TULIP) are the known link layer implementation based on TCP *unaware* link layer techniques as discussed in [7].

Barakat *et al.* [11] reported that ARQ mechanisms may vary the characteristics of the network, affecting the functionality of the upper layer protocols. Moreover lack of knowledge of the protocol operating at the transport level may results into end-to-end performance degradation in presence of independent link layer mechanisms. For instance, an approach without awareness of the transport protocol may cause local link layer retransmission of a packet, as well as duplicate acknowledgement (*dupack*), since retransmissions can be performed on both layers. This led to significant efforts in development of TCP *aware* link layer approaches [18].

2.2. TCP *Aware* Approaches

The link layer approaches in which enhancements tailored for wireless environments are made known to TCP are broadly referred as TCP *aware* link approaches. A good overview of these approaches can be found in [7] [17] [18]. The representative mechanisms employed in various approaches include *Snooping*, *Delayed Acknowledgement* and *Performance Enhancement Proxy* (*PEP*).

2.2.1. Snooping Mechanism

Most of the approaches which operate at this level rely on some intermediate point within the end-to-end connection for the introduction of performance improvement. In *Snoop* protocol [19], agent located at the *Base Station* (BS) monitors every packet that passes through the TCP connection in both directions. The *Snoop* agent suppresses the *dupacks* for lost TCP packets and retransmits locally, thereby preventing unnecessary invocation of congestion control mechanism by sender. Other enhancements proposed in similar category are TCP *SACK-aware Snoop* Protocol and SNACK-NS (*New Snoop*) [7]. This protocol may cause additional delay for TCP packets, which results into *futile* TCP retransmissions at TCP sender over the slower wireless link. However, the problem can be solved by minimizing the number of retransmissions at the link level [7].

WTCP [20], although operating in a similar way, introduces more accurate RTT estimation and thus preventing reduction in TCP throughput. WTCP conceals the difference in RTT from the sender, thus avoiding needless timeout caused by reason of local retransmissions. Both *Snoop* and WTCP must, however, have access to the header of TCP packets in order to function, which reduces their usefulness value if traffic is encrypted. The use of *Snoop* agent is also exploited in wireless TCP refinements with split connection schemes [7]. *Snoop* and WTCP increase complexity at BS, especially when transport layer per flow support is required. TCP PEPs [21] are TCP *aware* link layer mechanisms for handling bandwidth asymmetry, TCP *aware* FEC provisioning or adaptive link frame sizing [22]. Split connection approaches [7] and PEPs, though effective in many cases, break TCP end-to-end semantics and its modularity and hence not considered as pure link layer approaches.

2.2.2. Delayed Acknowledgements

Another class of work includes *Delayed Duplicate Acknowledgment* (DDA) [23], which preferred particularly when IP encryption is used. With a view to reduce interference between the TCP and link retransmissions, TCP receiver delays the third and subsequent *dupacks* for some interval "*d*". This provides enough time for link layer to recover the lost packet using local retransmission, and to prevent TCP sender from fast retransmit. Different from the *Snoop* protocol, in DDA *dupacks* are not dropped immediately but rather delayed a certain length of time. *Delayed Acknowledgements over Wireless Link* (DAWL) [24] tries to simplify the system by introducing modifications only in the ARQ scheme at the link layer and does not consider local retransmission at the BS. This design is advantageous in case of a BS crash [7]. However, in presence of congestion, the performance of these schemes is degraded, as the essential fast retransmissions are unnecessarily delayed.

Since different link layer technologies have diverse capabilities for error recovery, there is no *de-facto* standard for link layer protocols. All link layer approaches try to reduce the effect of erroneous wireless links on the performance of TCP, but do not completely shield the TCP sender from all types of wireless errors particularly in the event of lengthy disconnection or high *Bit Error Rate* (BER). The main advantage of link layer approaches is the maintenance of end-to-end semantics, without modification of higher protocol layers. This makes it possible to leave untouched the existing implementations of the protocol stack in the various operating systems.

3. IEEE 802.11 Link Recovery Mechanism

The basic channel access method in 802.11 networks is the *Distributed Coordination Function* (DCF) in which 802.11 *Medium Access Control* (MAC) layer uses the *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) mechanism before transmitting any frame. The IEEE 802.11 MAC layer employs Type 0 ARQ concept for loss recovery with the contention based transmission policy. The ARQ is the only error control method specified in the standard and no FEC coding is used as stated earlier.

Over an IEEE 802.11 link, whenever a wireless node notices unsuccessful transmission for a frame, it attempts local retransmission of a frame in accordance with *RetryLimit* (*RL*) [1]. To explain, as shown in **Figure 1**, a wireless node after sending a frame waits for a positive ACK from wireless receiver. The sender attempts retransmission of the frame whenever ACK is not received before expiration of ACK timeout. The node is allowed to attempt retransmissions for a particular frame maximum up to *RL*. Then after node initiates for new transmission and the previous frame is considered to be lost. Thereby the upper layer is made responsible for recovery of the frames lost after maximum retransmissions.

Such link retransmissions additionally delay the transmission of subsequent TCP packet residing at *Interface Queue* (IFQ). This *additional delay* (T_{ARQ}) is approximated using Equation (2).

$$T_{ARQ} = n \cdot \left(t' + T_{CONT} \right) \tag{2}$$

$\mathrm{T}_{\mathrm{ARQ}}$ increases with (a) increase in the number of unsuccessful retransmissions (*n*) and (b) increase in link delay (*t′*). Here T_{CONT} signifies the contention delay over a shared medium [1]. TCP's RTT estimation includes wired link delay and processing delay at nodes (the combined is referred as RTT_0), delay due to network con-

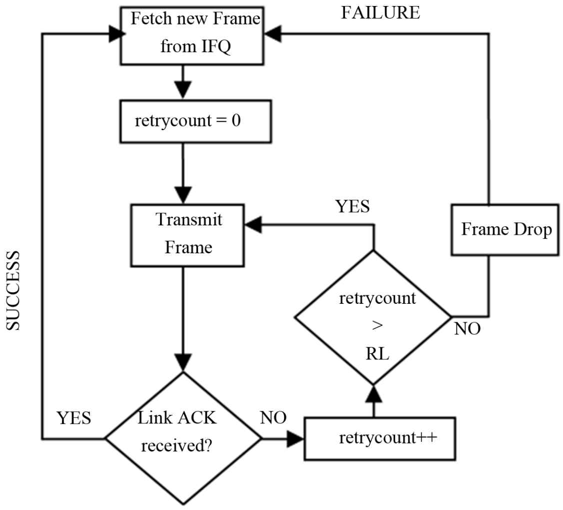


Figure 1. IEEE 802.11 link loss recovery mechanism.

gestion (d_c) and T_{ARQ} as shown in Equation (3).

$$\mathrm{RTT} = \mathrm{RTT}_0 + d_c + T_{ARQ} \tag{3}$$

In short, link recovery mechanisms shield TCP from the wireless losses and hence prevent undue reduction of *cwnd*, but at the same time increase RTT estimation at TCP sender. The earlier effect prevents sacrificed transmission rate and the later accounts for diminished growth in transmission rate. In the next section, an analysis of results obtained through simulations is presented to illustrate impact of link recovery on TCP's performance.

4. Simulation Results and Analysis

4.1. Simulation Scenario

A network topology illustrated in **Figure 2** was used for analyzing *Medium Access Control* (MAC) and TCP behavior using a TCP connection between sender (S1) and receiver (D). Wireless losses over an IEEE 802.11b link between S1 and BS were introduced using 2-*State Markov* model. The queue length at intermediate nodes is set in accordance to *Bandwidth Delay Product* (BDP) in the network.

A TCP/FTP flow was introduced for 100 sec. To investigate the impact of link recovery on the efficiency of link and transport layer protocols, the simulations were performed using different value for *RL* and *Frame Error Rate* (FER) in the range mentioned along with results. TCP's RTT estimation and layer efficiency are the measurable parameters used for performance evaluation. Layer efficiency is defined as the ratio of total number of successful transmissions to the total transmissions made by that layer during simulation interval. Nevertheless the most critical parameter under observation is (*cwnd*/RTT). For better understanding, analysis of results is presented with different traffic conditions; (a) a single TCP flow and (b) multiple TCP flows competing over a bottleneck link. In the mentioned topology, TCP New Jersey [25] (TCP NJ) has demonstrated better throughput compared to that achieved using other well-known TCP variants (*i.e.* NewReno, Vegas and Westwood [26]). Therefore the analysis is presented for TCPNJ as TCP variant. However similar trend in the results is observed with other TCP variants as well.

4.2. Analysis with Single TCP Flow

Figures 3-8 present analysis based on the results obtained from the experiments.

As shown in **Figure 3(a)**, with link recovery (RL = 7), increase in FER resulted into increase in link retransmissions in order to combat transmission losses locally. Pl. note that even with 0% FER (introduced using error model), losses were witnessed due to channel contention (due to transmission of TCP data and acknowledgement packets on the same channel in reverse direction). It led to noticeable MAC retransmissions for recovering from such losses. During simulations with link recovery (RL = 7), % of wireless TCP drops were significantly

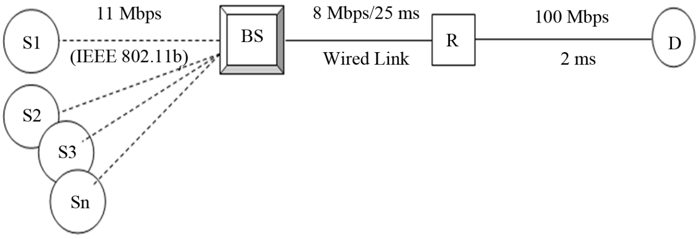
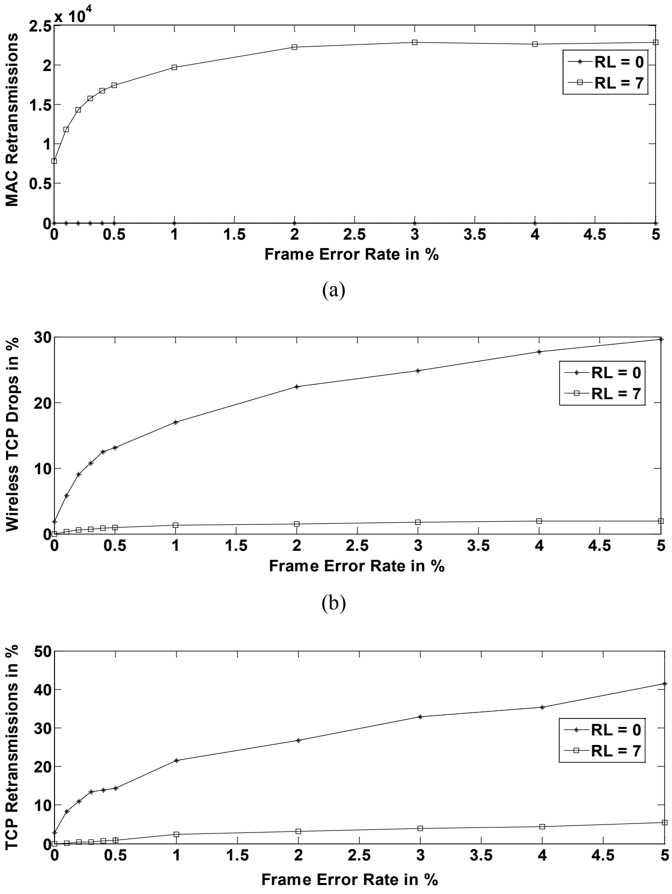


Figure 2. Network Topology.



(a)

(b)

(c)

Figure 3. (a) Impact of link ARQ on MAC retransmissions, (b) Impact of link ARQ on Wireless TCP Drops in %, (c) Impact of link ARQ on TCP Retransmissions (in %).

reduced together with increase in FER compared to those witnessed during simulations without link recovery (RL = 0) (refer **Figure 3(b)**). Since, the error model introduces losses in % of total transmissions; in order to demystify the impact of link loss recovery correctly the analysis is presented in terms of % of TCP drops. Reduction in wireless TCP drops has caused significant reduction in end-to-end TCP retransmissions (analysis is presented in % of total TCP transmission due to similar reasons stated earlier), as shown in **Figure 3(c)**.

The simulations were performed using single TCP flow and therefore the packet losses were attributed only to the transmission errors on wireless link. During simulations TCP flow encountered timeouts due to either loss of retransmissions or due to loss of TCP acknowledgement in presence of errors on the wireless links, those are very costlier to TCP. **Figure 4** represents comparison for number of TCP timeouts with RL = 0 and RL = 7. As

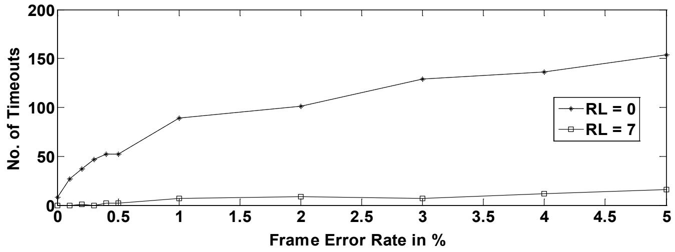
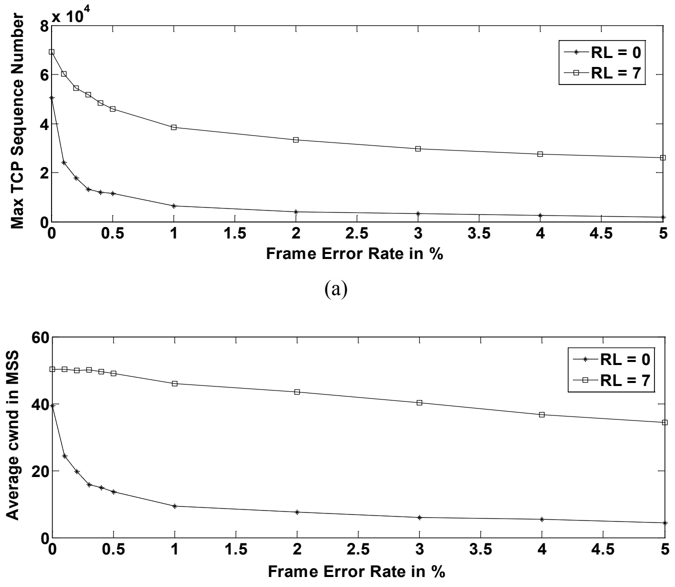


Figure 4. Impact of link ARQ on TCP timeouts.



(a)

(b)

Figure 5. (a) Impact of link ARQ on TCP's maximum sequence number, (b) Impact of link ARQ on *congestion window*.

seen from the Figure, number of timeouts were considerably condensed when link retransmissions were enabled (*i.e.* RL = 7).

In absence of any support for loss discrimination, TCP reacts to timeout with drastic reduction in *cwnd*, which is inappropriate for wireless losses. **Figure 5(a)** shows that without link recovery (RL = 0), TCP's maximum sequence number dropped rapidly with increase in FER; since loss of TCP segments has reduced the *cwnd* and slowed down TCP's transmissions per RTT. However with link recovery, significantly higher value of TCP's maximum sequence number is obtained, relative to that obtained with *RL* = 0.

This improvement is additionally attributed to the massive reduction in costlier TCP timeouts, in presence of link loss recovery attempts (RL = 7). It is apparent that link ARQs can improve the performance of TCP by shielding it from wireless losses as much as possible. Comparison for average *cwnd* with and without link loss recovery is shown in **Figure 5(b)**. When RL = 7, TCP's average *cwnd* and maximum sequence number decreased linearly with increase in FER.

Figure 6(a) and **Figure 6(b)** represent transmission efficiency for MAC and TCP protocols with RL = 0 and RL = 7. As anticipated, link recovery (RL = 7) improves efficiency for both the layers. It must be noted that without link recovery (RL = 0), TCP's efficiency is reduced to a great extent with higher FER. Hence, link recovery improves end-to-end performance in the network by improving TCP's transmission efficiency on large scale. In **Figure 6(b)**, about 30% of improvement for TCP's efficiency at 5% FER is recorded. Improved efficiency of MAC and TCP protocols result into significant improvement in TCP's throughput as shown in **Figure 6(c)**.

In **Figure 7(a)**, comparison for TCP's RTT estimated with RL = 0 and RL = 7 is presented for different FER.

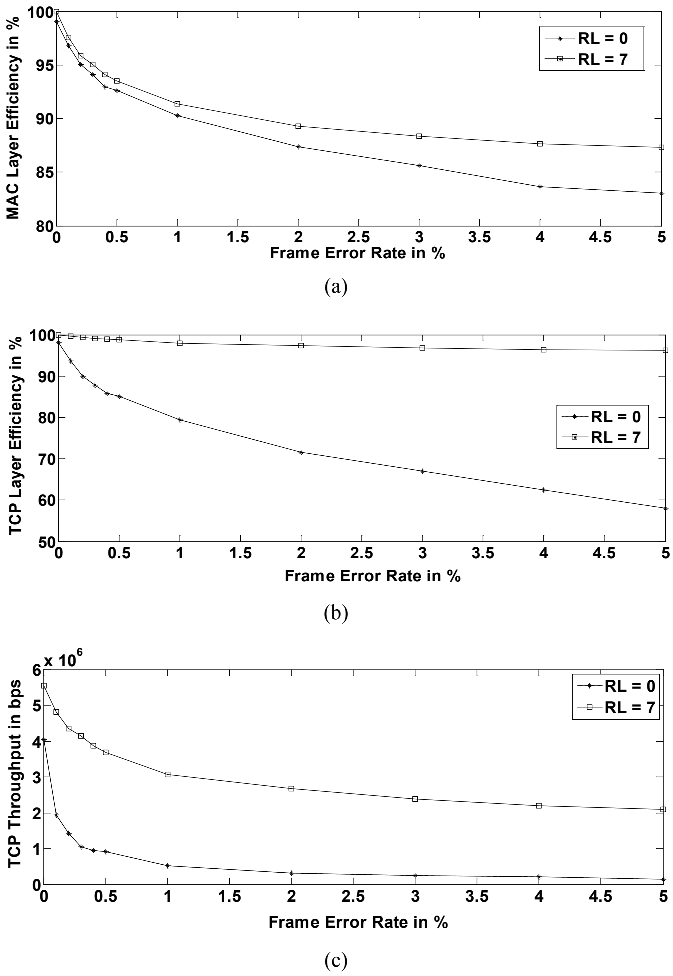
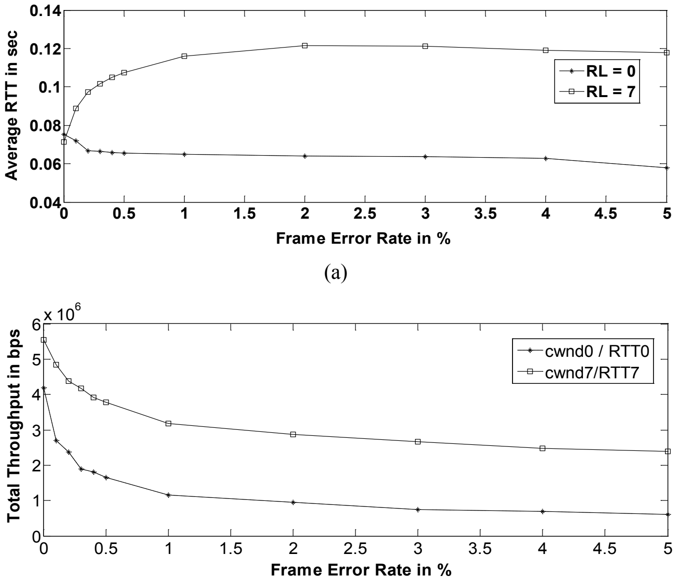


Figure 6. (a) Impact of link ARQ on MAC layer efficiency, (b) Impact of link ARQ on TCP layer efficiency, (c) Impact of link ARQ on TCP's throughput.

Without link recovery (RL = 0), the average RTT (say RTT_0) decreases with increase in FER. As shown in **Figure 5(b)** earlier, $cwnd_0$ has reduced sharply with increase in FER up to 2%. This reduction in $cwnd_0$ grounded for less number of TCP segments at an IFQ and hence as expected RTT_0 estimated at sender was also lowered sharply. It must be noted that the RTT_0 is found very close to the theoretical RTT in the network. A bit higher value is attributed to the contention delay at wireless interface and processing delay at nodes. In contrary, with link recovery (RL = 7), average RTT (say RTT_7) is increased along with increase in FER; as the link layer on average took a longer time for error recovery before delivering the TCP segments or acknowledgements in either directions. **Figure 7(a)** shows that the average RTT is increased with increase in FER until it reaches about 2%. With increase in FER above 2%, there is insignificant change in MAC retransmissions (refer **Figure 3(a)**) and hence similar was the impact on RTT_7 estimation. With FER above 2%, marginal reduction in RTT_7 was observed due to reduction in *cwnd*.

Based on the foremost observation it seems that the TCP with link layer recovery (RL = 7) utilizes network with higher value for average *cwnd* ($cwnd_7$) compared to that accomplished without link recovery ($cwnd_0$). Consequently, it leads to enhanced TCP's performance. In fact TCP with link recovery has achieved higher value of average *cwnd* at the cost of additional rise in average RTT estimation.

Since TCP's throughput is proportionate to its effective sending rate (*i.e.* $cwnd/\text{RTT}$), it is apparent that the net improvement in end-to-end TCP throughput is realized only when $(cwnd_7/\text{RTT}_7) > (cwnd_0/\text{RTT}_0)$. In **Figure 7(b)**, average value of $cwnd_7/\text{RTT}_7$ and $cwnd_0/\text{RTT}_0$ is plotted for different value of FER. It is revealed that the



(a)

(b)

Figure 7. (a) Impact of link ARQ on TCP's RTT Estimate, (b) Impact of link ARQ on TCP's RTT Estimate.

improvement in TCP's throughput is following the net improvement for average *cwnd*/RTT. Pl. note that even with 0% FER (introduced using error model), transmission losses were witnessed due to channel contention (due to transmission of TCP data and ack packets on the same channel in reverse direction). Link ARQs recovered from majority of contention losses and hence higher TCP's throughput is seen compared to that observed without link recovery.

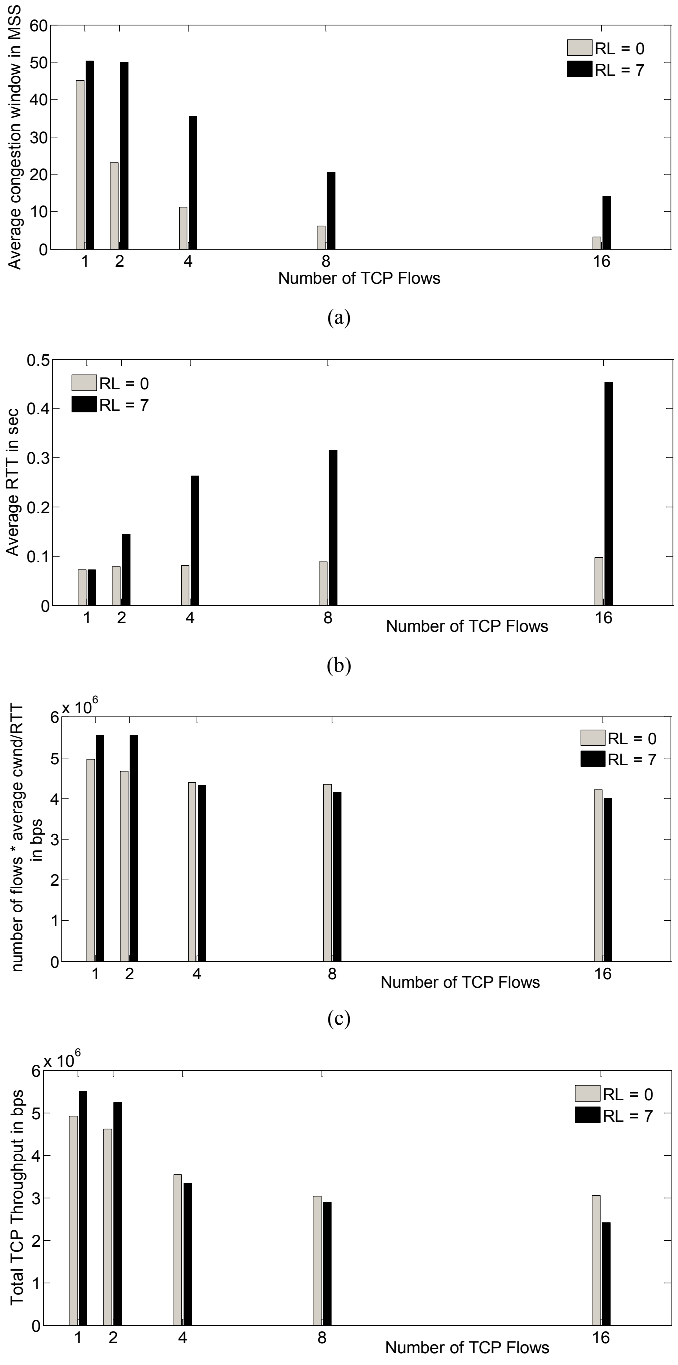
4.3. Analysis with Multiple TCP Flows

After analyzing impact of link ARQ on TCP's performance in non-congested erroneous wireless network, our simulations were extended for further investigations in a congested wireless network. For that number of simultaneous TCP senders in the networks were increased in the range of 2,4,8 and 16. The statistics obtained for FER of 0.001% are as shown in **Figure 8**.

As seen from **Figure 8(a)**, with increase in number of competing flows, effective value of *cwnd* (for all flows over a bottleneck link) reduces with or without loss recovery as anticipated. However, for given number of competing flows, effective value of average *cwnd* is found much higher when link recovery was enabled. With increase in the number of TCP flows, congestion in the network increases.

Moreover link recovery in presence of transmission losses gave additional rise in RTT (as shown in **Figure 8(b)**). This aggravated for congestion in the network and consequently large number of congestion drops are witnessed, particularly with increase in the competing flows (for a given error rate). This has remarkably pulled down growth in *cwnd* due to rise in RTT, which can be seen from the **Figure 8(c)**. As mentioned in the Figure, with link recovery (RL = 7), improvement in *cwnd*/RTT is seen when there was a single TCP flow. However with increase in number of TCP flows improvement for *cwnd*/RTT using link recovery is found insignificant. In fact when number of competing TCP flows increased upto 4 or above, link recovery failed in protecting *cwnd*/RTT and consequently, as presented in **Figure 8(d)**, TCP performance is found deteriorated.

This gives insight to the issue related to the TCP's performance over an IEEE 802.11 network, even in presence of link recovery mechanism. The problem really occurs when link layer mitigation substantially diminishes growth in *cwnd* by increasing RTT to a much higher value with increase in FER. The similar problem may occur when link layer mitigation fails due to substantial underlying error conditions. In fact use of link layer approaches may degrade performance especially in presence of highly variable error rates. In the stated situation, reduction in *cwnd* and rise in RTT is unavoidable at sender on account of failed ARQs. This adversely effects on the throughput conceived by TCP.



(d)

Figure 8. (a) Impact of link ARQ on average *congestion window*, (b) Impact of link ARQ on average RTT, (c) Impact of link ARQ on effective *cwnd*/RTT, (d) Impact of link ARQ on total TCP throughput.

5. Conclusions

We have performed a detailed analysis of using link ARQs to improve the end-to-end performance of TCP in wireless networks. The activities are concluded with the following outcomes.

1) Link recovery attempts using ARQ in IEEE 802.11 networks effects on two important TCP parameters; *i.e.* *cwnd* and RTT. TCP performance is decided jointly by both of the above parameters; *i.e.* always attributed to the

improvement in the *cwnd*/RTT value.

2) Link ARQs in absence of network congestion and light wireless errors yield superior value of *cwnd*/RTT in comparison to that observed without link ARQs. This in turn results in better TCP performance.

3) Recovering from all types of wireless losses using link mitigation techniques is not always possible; particularly with high and bursty error environments. This may lead to performance degradation due to unavoidable cutback in *cwnd* and increase in RTT.

4) Link ARQs may degrade performance in congested networks even with very small amount of wireless errors. The performance degradation is attributed to any of the following reasons:

- When link level error mitigation fails, reduction in *cwnd* at TCP sender is unavoidable. Since loss recovery attempts increase RTT and unable to protect *cwnd*, TCP''s end-to-end performance is compromised.
- If multiple TCP flows are sharing common wireless channel for transmission, loss recovery attempts made for one TCP flow may affect RTT of all competing flows and therefore overall network performance is found compromised.

This shows possibility for further improvement in *cwnd*/RTT (and hence TCP throughput) using a corrective mechanism that makes RTT unaffected from link recovery attempts. Nevertheless, any correction that reduces RTT inappropriately may cause the sender to timeout earlier when the retransmission is being performed on the wireless link. Therefore, a well thought upon approach for RTT correction may be considered for maximizing TCP's throughput in existence of link layer ARQs. This highlights the scope for further performance enhancement of TCP proposals those advocate for retaining of *cwnd* based on loss discrimination.

References

[1] (2003) IEEE Standard for Information Technology-Telecommunications and Information Exchange between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Std* 802.11, 1999 Edition (R2003), I-513. http://dx.doi.org/10.1109/IEEESTD.2003.95617

[2] Inamura, H., Ludwig, R. and Khafizov, F. (2003) TCP over Second (2.5G) and Third (3G) Generation Wireless Networks. *IETF RFC* 3481.

[3] Lohier, S., Doudane, Y.G. and Pujolle, G. (2007) Cross-Layer Loss Differentiation Algorithms to Improve TCP Performance in WLANs. *Springer US Journal of Telecommunication Systems*, **36**, 61-72.

[4] Floyd, S., Henderson, T. and Gurtov, A. (2004) RFC 3782: The NewReno Modification to TCP's Fast Recovery Algorithm. *IETF RFC* 3782.

[5] Kliazovich, D., Gerla, M. and Granelli, F. (2007) Performance Improvement in Wireless Networks using Cross Layer ARQ. *The International Journal of Computer and Telecommunications Networking*, **51**, 4396-4411.

[6] Leung, K.-C. and Li, V.O.K. (2006) Transmission Control Protocol (TCP) in Wireless Networks: Issues, Approaches and Challenges. *IEEE Communications Surveys & Tutorials*, **8**, 64-79. http://dx.doi.org/10.1109/COMST.2006.283822

[7] Sardar, B. and Saha, D. (2006) A Survey of TCP Enhancements for Last-Hop Wireless Networks. *IEEE Communications Surveys & Tutorials*, **8**, 20-34. http://dx.doi.org/10.1109/COMST.2006.253273

[8] Tian, Y., Xu, K. and Ansari, N. (2005) TCP in Wireless Environments: Problems and Solutions. *IEEE Communications Magazine*, **43**, s27-s32.

[9] Padhye, J., Firoiu, V., Towsley, D. and Kurose, J. (1998) Modeling TCP Throughput: A Simple Model and Its Empirical Validation. *Proceedings of SIGCOMM*, 303-314.

[10] Ludwig, R. and Gurtov, A (2005) The Eifel Response Algorithm for TCP. *IETF RFC* 4015.

[11] Barakat, C. and Altman, E. (2002) Bandwidth Tradeoff Between TCP and Link-level FEC. *Computer Networks*, **39**, 133-150. http://dx.doi.org/10.1016/S1389-1286(01)00305-X

[12] Singh, J.P., Li, Y., Bambos, N., Bahai, A., Xu, B. and Zimmermann, G. (2007) TCP Performance Dynamics and Link-Layer Adaptation Based Optimization Methods for Wireless Networks. *IEEE Transactions on Wireless Communications*, **6**, 1864-1879.

[13] Barakat, C. and Fawal, A.A. (2004) Analysis of Link-Level Hybrid FEC/ARQ-SR for Wireless Links and Long-Lived TCP Traffic. *Performance Evaluation*, **57**, 453-476. http://dx.doi.org/10.1016/j.peva.2004.03.002

[14] Cheng, R.S. and Lin, H.T. (2008) A Cross-Layer Design for TCP End-to-End Performance Improvement in Multi-Hop Wireless Networks. *Computer Communications*, **31**, 3145-3152. http://dx.doi.org/10.1016/j.comcom.2008.04.017

[15] Chan, M.C. and Ramjee, R. (2004) TCP/IP Performance over 3G Wireless Links with Rate and Delay Variation.

INFOCOM.

[16] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and Crowcroft, J. (2002) RFC 3452: Forward Error Correction (FEC) Building Block. *IETF RFC* 3452.

[17] Kim, D., Choi, Y., Jin, S., Han, K. and Choi, S. (2008) A MAC/PHY Cross-Layer Design for Efficient ARQ Protocols. *IEEE Communications Letters*, **12**, 909-911. http://dx.doi.org/10.1109/LCOMM.2008.081259

[18] Scott, J. and Mapp, G. (2003) Link Layer-Based TCP Optimization for Disconnecting Networks. *ACM SIGCOMM Computer Communications Review*, **33**, 31-42.

[19] Sun, F., Li, V.O.K. and Liew, S.C. (2004) Design of SNACK Mechanism for Wireless TCP with New Snoop. *IEEE Wireless Communications and Networking Conference*, **5**, 1046-1051.

[20] Sinha, P., Nandagopal, T., Venkitaraman, N., Sivakumar, R. and Bharghavan, V. (2002) WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks. *Wireless Networks*, **8**, 301-316. http://dx.doi.org/10.1023/A:1013702428498

[21] Border, J., Kojo, M., Griner, J., Montenegro, G. and Shelby, Z. (2001) RFC 3135: Performance Enhancing Proxies Intended to mitigate Link Related Degradation. *IETF RFC* 3135.

[22] Tickoo, O., Subramanian, V., Kalyanaraman, S. and Ramakrishnan, K.K. (2005) LT-TCP: End-to-End Framework to Improve TCP Performance over Networks with Lossy Channels. *IEEE International Workshop on Quality of Service*.

[23] Vaidya, N. and Mehta, M. (2002) Delayed Duplicate Acknowledgments: A TCP-Unaware Approach to Improve Performance of TCP over Wireless. *Wireless Communications and Mobile Computing*, **2**, 59-70. http://dx.doi.org/10.1002/wcm.33

[24] Kliazovich, D. and Granelli, F. (2005) DAWL: A Delayed-ACK Scheme for MAC-Level Performance Enhancement of Wireless LANs. *ACM/Kluwer Journal on Mobile Networks and Applications* (*MONET*), **10**, 607-615.

[25] Kim, J., Koo, J. and Choo, H. (2007) TCP NJ+: Packet Loss Differentiated Transmission Mechanism Robust to High BER Environments. *Ad Hoc and Sensor Networks*, *Springer Wireless Networks*, *Next Generation Internet Lecture Notes in Computer Science*, **4479**, 380-390.

[26] Marcondes, C., Sanadidi, M.Y., Gerla, M. and Shimonishi, H. (2008) TCP Adaptive Westwood-Combining TCP Westwood and Adaptive Reno: A Safe Congestion Control Proposal. *ICC '08*, *IEEE International Conference on Communications*, Beijing, 19-23 May 2008, 5569-5575. http://dx.doi.org/10.1109/ICC.2008.1044