

Cross-Layer Loss Differentiation Algorithms to improve TCP Performance in WLANs

Stephane Lohier¹, Yacine Ghamri Doudane², and Guy Pujolle³

¹ IUT- University of Marne la vallée, 77420 Champs sur Marne – France
lohier@univ-mlv.fr

² Computer Engineering Institute - IIE - 18, allée Jean Rostand 91025 Evry Cedex – France
Ghamri@iie.cnam.fr

³ LIP6 - University of Paris VI - 8, rue du Capitaine Scott 75015 Paris – France
Guy.Pujolle@lip6.fr

Abstract. Loss Differentiation Algorithms (LDA) are currently used to determine the cause of packet losses with an aim of improving TCP performance over wireless networks. In this work, we propose a cross-layer solution based on two LDA in order to classify the loss origin on an 802.11 link and then to react consequently. The first LDA scheme, acting at the MAC layer, allows differentiating losses due to signal failure caused by displacement or by noise from other loss types. Moreover, in case of signal failure, it adapts the behavior of the MAC layer to avoid a costly end-to-end TCP resolution. The objective of the second LDA scheme, which acts at the TCP layer, is to distinguish a loss due to interferences from those due to congestions and to adapt consequently the TCP behavior. The efficiency of each LDA scheme and of the whole cross-layer solution are then demonstrated through simulations.

1 Introduction

Due to various and unpredictable reasons (low noise immunity, overhead, throughput related to the distance...), the performance of TCP in the 802.11 [1] networks are not always as sufficient as the current applications require, particularly in SOHO (Small Office Home Office) environments or in public points of distribution (Hot Spot) with a wireless last link. In order to enhance elastic traffic performance in WLANs, several solutions have been proposed during the last few years (see section 2). Most of these proposals deal only with one layer (TCP or MAC) and they are either not adapted to 802.11 networks or require important changes in the current standard. Therefore, in an attempt to propose a solution that is compliant with the 802.11 standard, we suggest a cross-layer approach acting in a coordinated way on the two distinct MAC and TCP resolution levels. To do so, we propose two complementary Loss Differentiation Algorithms (LDA):

- The first one, implemented at the MAC layer, is used to identify losses due to wireless link failures that occur when the distance between the wireless station and its Access Point (AP) increases or when obstacles appear temporarily between them. In this situation, the MAC *Retry Limit* parameter is dynamically adapted in order to maintain MAC retransmissions and thus to avoid a complete TCP resolution.

- The second LDA acts at TCP level and its objective, complementary to the MAC-level one, is to distinguish packet losses due to congestions from those related to short and repetitive signal losses due to interferences caused by other close transmissions in the same frequency band. This differentiation is realized through the monitoring of MAC level parameters. The integration of this second LDA scheme to the TCP *NewReno* process permits to avoid triggering the TCP loss recovery mechanism and reducing the TCP congestion window inadequately.

Hence, a cross-layer solution based on the combined use of both LDA schemes will allow to classify efficiently the three different loss causes (congestions, signal losses and interferences) and to react accordingly either at the MAC or TCP level.

The rest of this paper is organized as follows: section 2 presents a brief analysis of the various approaches proposed in the literature to improve TCP in WLANs. In section 3, the MAC-layer LDA and the associated adaptation are described and evaluated. Section 4 presents and evaluates the cross-layer LDA and the related TCP improvements. A summary of the loss cases and the associated differentiations are given in section 5; this later also shows the effectiveness of the whole cross-layer solution. Finally, section 6 concludes the paper and proposes some future issues.

2 TCP Improvements in WLANs

The TCP performance improvement attempts in wireless networks can be classified into three categories, according to the concerned layer within the protocol stack.

The first category concerns data link layer with two different proposed mechanisms. The first one is the improved *Logical-Link Control* (LLC) algorithm [3] which proposes to introduce queuing capabilities to the LLC sub-layer in order to delay frame transmission during signal losses. This algorithm gives very interesting results, but it requires important updates of AP and station firmware in order to establish and manage the added LLC queues. The second mechanism uses the *Automatic Repeat reQuest* (ARQ) protocol. It is demonstrated in [4] that ARQ improves TCP throughput. However, it is a specific data link layer protocol based on the retransmission requests of the lost frames and thus not easily adaptable to the 802.11 standard.

The second category of proposals concerns the transport layer and is based on the end-to-end resolution for existing TCP versions (Reno, Vegas...) [5-6]. These optimizations are made for the general context of wireless networks and thus do not give always good performance in 802.11 networks where a first level of error recovery is carried out at the MAC layer. Other end-to-end solutions use LDA [7-8] to differentiate loss types with successive measurements of *Round Trip Time* (RTT) or packet inter-arrival times. These algorithms are often efficient and easy to implement but they assume that TCP flows are relatively regular and that signal losses are unusual and not very persistent, which is not always the case when signal failure or interferences occur in WLAN.

The last category also concerns the transport layer but the control is made by a *Snoop Agent* located at the *Access Point* (AP). The *Snoop Agent* can manage two different mechanisms. The first one is the *Explicit Loss Notification* (ELN) mechanism [9] in which the *Snoop Agent* is able to analyze all the transmitted segments and to set the ELN bit of the TCP header consequently. The agent installed at the AP operates at the TCP layer, which supposes a consequent modification of the firmware. In a second proposal [10], the *Snoop Agent* analyzes all the TCP segments in order to detect the duplicated

acknowledgements and not to relay them in the event of signal loss. This permits to avoid alerting TCP and reducing the congestion window. In addition to the firmware modifications, this latter proposal requires a difficult adjustment between the different timeout involved in TCP, *Snoop* and MAC recovery mechanism.

Most of these proposals begin with the distinction between signal loss and TCP congestion. But the proposed solutions are not adapted to the 802.11 standard or require important modification of the firmware and are generally limited to a specific layer. From our point of view, an LDA-based solution, utilizing in a coordinated way the features of both concerned layers would give better performance. Moreover, we show that a distinction between TCP congestions and the two kinds of signal-losses, those due to mobility and those due to interferences, allows realizing a more efficient loss recovery in the targeted environment.

3 MAC-Layer LDA

A MAC-level retransmission occurs when the 802.11 acknowledgment is not received by the transmitter within the specified delay. For each retransmission, a counter is incremented until a threshold, named *Retry Limit*, is reached (default value is fixed to 6). Beyond this threshold, the frame is dropped.

For a connection using TCP, coherence between layers should lead to a fast MAC layer resolution (almost 1ms for a MAC timeout) before TCP is alerted when the segment loss is due to bad channel conditions. Otherwise, TCP will consider this packet loss as a congestion (almost 1s for a TCP timeout), which induces a reduction of the *congestion window* (*cwnd*) and a fall in the global throughput.

The measurements we carried out in [11] show that an increase of the *Retry Limit* value allows, for loss duration lower than the TCP *Retransmission TimeOut* (*RTO*), to recover the flow as soon as the channel is restored and thus provides a recovery mechanism faster than a standard TCP retransmission. We also showed that this mechanism is effective for signal loss of a few hundreds ms or more, typically caused by the displacement of the pedestrian user at the cover limit of its access point or by other pedestrians moving between the AP and the station. For shorter and repetitive signal losses caused by interferences, it is shown in [11] that a simple increase of the *Retry Limit* value does not improve the performance. Thus, we suggest treating these different types of losses differently.

3.1 Principle

As a systematic increase of the *Retry Limit* value can either be inadequate in some cases (congestion) or ineffective in some others (interferences), we suggest here the use of a *Loss Differentiation Algorithm*. The objective of this LDA is to know when this increase is appropriate and to realize a dynamic adaptation of *Retry Limit* accordingly. Let us note that although the *Retry Limit* parameter is configured statically in all the implementations, it can be dynamically modified without contradicting the 802.11 standard.

To identify signal losses caused by the distance or obstacles, it is inappropriate to employ a Transport-layer LDA. Such schemes use successive values of *RTT* or packet inter-arrival times and it is not possible to make these measurements when the channel is

unavailable (signal loss can be of a few hundreds ms or more). A more appropriate parameter for this differentiation is the *SNR (Signal to Noise Ratio)* given at the MAC layer.

In the MAC 802.11 frames, the “signal” field specifies the current throughput used to transmit the following data (*Data Rate*). This throughput indication depends on the measured power received by the station before its transmission and is thus proportional to the SNR. In addition, this throughput is related to the *Auto Rate Fallback (ARF)* procedure implemented by all the 802.11a, b or g card manufacturers. Let us recall that this procedure automatically reduces the throughput when a drop in the SNR is sensed. This can be due either to distance or obstacles.

The proposed LDA is based on the simple fact that if *SNR* (or *Data Rate*) is maximal, the probability that the segment loss is due to a signal failure caused by the distance and not to TCP congestion is very weak. In the mean time, this probability increases with the decrease of *SNR*. Thus, the idea of the proposed LDA is to allow a dynamic *Retry Limit* adaptation according to the *Data Rate* given at the MAC 802.11a, b or g layer. The algorithm depicted in Table 1 applies to the 802.11g standard where the throughput decreases gradually from 54Mbps to 6Mbps while passing by 24Mbps and 12Mbps. The increase of the *Retry Limit* threshold is linear and progressive (the default value of 6 is successively added) to avoid congesting the channel unnecessarily when this latter is used by other transmissions.

Note that the *Retry Limit* increase is bounded by three events:

- the arrival of the MAC acknowledgment for a retransmitted segment;
- the TCP transmission window is emptied;
- the RTO is reached without the channel being restored.

When one of these events occurs, the *Retry Limit* is reset to its initial value (i.e. for future transmissions). This help to avoid occupying the channel unnecessarily. In addition, one should note that, during the RL increase phase, the fairness with other MAC traffics on the same channel is preserved thanks to the 802.11 backoff algorithm which increases exponentially the time between two retransmissions according to the index of this latter. Also note that the others MAC traffics are not directly concerned by the *Retry Limit* increase because the *Data Rate* field transported in the frames is specific to each wireless station and its transmission conditions (distance, obstacles ...)

Table 1. MAC-layer LDA

if (DataRate ≥ 12Mbps) then	// station is closed to AP
RetryLimit = 6	// default value
else if (DataRate > 6Mbps)	// possible signal failure
RetryLimit = 12	// begin to enlarge transmission window
else if (DataRate ≤ 6Mbps)	// probability of failure is max
RetryLimit = 18	// continue to enlarge window
if (new segment) && (last segment dropped)	
RetryLimit = RetryLimit + 6	// new TCP segment and last MAC retry failed
end if	// enlarge again window
end if.	

3.2 Simulation results

In order to analyze the performance improvements brought by the MAC-layer LDA, a set of simulations are carried out using NS-2 [12]. The 802.11 implementation already available within NS-2 have been extended in order to incorporate the more recent 802.11g specificities including the ARF procedure. The simulated network reproduces a usual SOHO environment with a wireless last hop interconnecting an 802.11 station receiving a FTP/TCP flow from a wired network. Signal losses are simulated by moving the wireless station out of the coverage area of its AP. The packet length is 1460Bytes, the default TCP retransmission timeout is 1s and the TCP version is NewReno, The choice of this TCP variant is motivated by the fact that it gives the best results in an access network with a wireless last hop [13].

Fig. 1 shows the evolution of the average throughput for the TCP flow during a signal loss according to its duration, with and without MAC-layer LDA. The average throughput is measured according to the number of TCP segments successfully transmitted during a 1s observation period starting at the beginning of the signal failure (this duration corresponds to the default TCP timeout). The loss duration interval is selected to have a dozen values between a short loss of few ms and *RTO*.

The maximum *Data Rate* of 4.6Mbps (*Signal Rate* of 6Mbps) is obtained, according to the ARF procedure, when the mobile is at the limit of the coverage area. The default values of the TCP window and *Retry Limit* are $Win=8KB$ and $RL=6$. In this case, the MAC loss-recovery is not effective and the average throughput reaches the null value for loss duration of 0.1s or more. Fig. 1 also depicts that the increase of the TCP window size ($Win=32KB$) can improve the loss-recovery as well. However it leads to more end-to-end TCP retransmissions, which increases the delays. The last curve shows that the MAC-layer LDA and the dynamic *Retry Limit* adaptation can improve the performance significantly. In this case, the loss-recovery is effective for loss duration up to 0.8s, and the flow is maintained for all these cases allowing TCP to react immediately when the channel is restored rather than waiting for *RTO*.

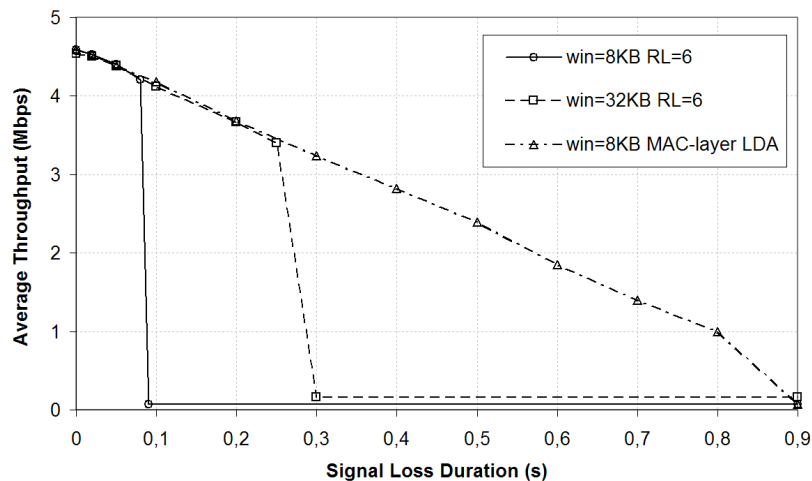


Fig. 1. TCP improvements with MAC-layer LDA

4 Cross-Layer LDA

Since signal failures due to distance or obstacles are resolved by the MAC-layer LDA, the objective here is to differentiate segment losses due to interferences from those due to congestions.

4.1 The proposed LDA scheme

As proposed in the literature, the differentiation decision can be obtained based on TCP parameters, namely *RTT*, packets inter-arrival times and *congestion window (cwnd)*. Our simulations with a wireless last hop show that the TCP variable presenting the most significant variations in the event of signal losses compared to congestions is the *RTT*. We thus selected three TCP-layer LDA schemes based on comparisons of current *RTT* values with different thresholds and on filters giving more or less weight to the recent samples: the *Vegas Predictor* scheme [14], the *Spike* scheme [7] and the *Flip Flop Filter* [15]. Then, rather than using only TCP-layer parameters which do not take into account the 802.11 specificities, we suggest to use a cross-layer approach as an alternative to conventional TCP-Layer LDA schemes. Even if the final objective of the algorithm is to indicate how to adjust the TCP behavior, the use of MAC-layer parameters to identify the cause of segment-losses can lead to a more accurate differentiation.

The idea of our alternative algorithm is to count the number of MAC retransmissions for each of the n segments composing the current TCP window when the TCP layer is alerted by the reception of three duplicated acknowledgements. As described in Table 2, if for one of these segments at least, the number of MAC retransmissions (*RetryCount*) is equal to the threshold (*Retry Limit*), we consider that the loss is due to interferences and not to TCP congestion. Indeed, in the case of congestion, the surplus of segments is eliminated from the queue of the concerned node and MAC retransmissions are theoretically not used; inversely, in case of persistent interferences, the segment is dropped by the MAC layer after reaching the *Retry Limit* threshold. This algorithm assumes that for all the not acknowledged TCP segments, the value of *RetryCount* is stored. The *ACK-FailureCount* counter available in the 802.11 *Management Information Base (MIB)* [1] gives the number of times that an expected ACK is not received and consequently the value of *RetryCount*.

Table 2. Cross-layer LDA

```
if (3 dup ack) then // loss indication in TCP NewReno algo
  LDA_Estimator = 0 // initial value for congestion
  for (i = 0 ; i ≤ n ; i ++ ) // for all the not acknowledged segments
    if (RetryCount = RetryLimit) then // segment is dropped, probably a short loss
      LDA_Estimator = 1 // set value for interferences
    end if
  end for
end if.
```

Note that while the TCP sender is not a wireless host and that the TCP flow is forwarded to the wireless receiver through an AP, an additional stage is necessary. The *LDA_Estimator* is first set at the AP's MAC layer. Then this latter informs the TCP

sender by setting consequently the *ELN* (*Explicit Loss Notification*) bit of the TCP header in the ACK segments (i.e. $ELN=LDA_Estimator=1$ in case of interferences). The loss differentiation is finally made at the TCP sender when receiving three duplicated ACKs. This mechanism is inspired from the one used with the *Snoop Agent* described in section 2. However, our solution does not concern the whole TCP layer but only the *ELN* bit of the TCP header is affected. The modification of the AP's firmware here is minimal compared to the analysis of all the transmitted segments that is performed by the *Snoop Agent*.

In order to realize a comparative study among the 4 selected LDA schemes, a set of simulations targeting a wireless context with a last link undergoing congestions or interferences have been realized. The simulated network is the same one as for section 3. Interferences are caused by the transmission on the same channel of a CBR/UDP flow between two other wireless stations out of the AP coverage and interferences areas. As we deactivated the RTS/CTS mechanism for both transmissions, the AP will not detect CBR transmissions and will thus transmit its TCP segments towards the receiver which is located in the interference area. The duration and the frequency of the interferences will vary according to the size of the packets and the rate of the CBR source. A good compromise is found with packets of 1000Bytes and frequency interval starting from 900packets/s (denoted 0% in the curves) to 1800packets/s (denoted 100% in the curves). Indeed, for lower frequencies, the loss rate is not significant while for higher frequencies, the wireless link is completely saturated by the CBR source. Let us note that the simulated interferences and so the packets losses are carried out in a scenario close to reality (short losses are often caused by transmissions in the same frequency band) and not with a theoretical packet error rate as inaccurately used in most studies.

Another CBR/UDP flow is established between the server and a fourth wireless station in order to saturate the AP and induce congestions. For this flow, the compromise is found with packets of 1000Bytes sent with a frequency varying from 1600packets/s (denoted 0%) to 3500packets/s (denoted 100%). Note that only one CBR source is active at the same time, i.e. interference and congestion cases are analyzed separately for a better understanding of the obtained results.

The simulation results presented in Fig. 2 and Fig. 3 show the accuracy (the percentage of correctly classified losses) of the four LDA schemes according to the interference or congestion rate. For the *Vegas predictor* scheme, we observe that the losses due to low interference rates or high congestion rates are badly classified (less than 60%). This is especially due to the evolution of *cwnd*, which is in these cases inadequately used in conjunction with *RTT* to compute the *Vegas predictor*. The *Spike* scheme, only based on *RTT* variations, gives slightly better results: accuracy higher than 80% in the majority of the cases. The badly classified losses are more random and are mainly due to the calculation mode of the Spike's thresholds. The *Flip Flop* filter is not very efficient, particularly for losses due to interferences. The used algorithm employs many parameters difficult to regulate. Finally, the proposed cross-layer LDA scheme gives the best results. For congestions, there are almost no MAC retransmissions and the *Retry Limit* threshold is never reached, which gives 100% of correctly classified losses. For interferences, some losses are badly classified when the segment is finally received with the last attempt. However accuracy remains in all the cases higher than 90%. For the continuation of the study, we will thus use this scheme as the basis of the TCP behavior enhancement.

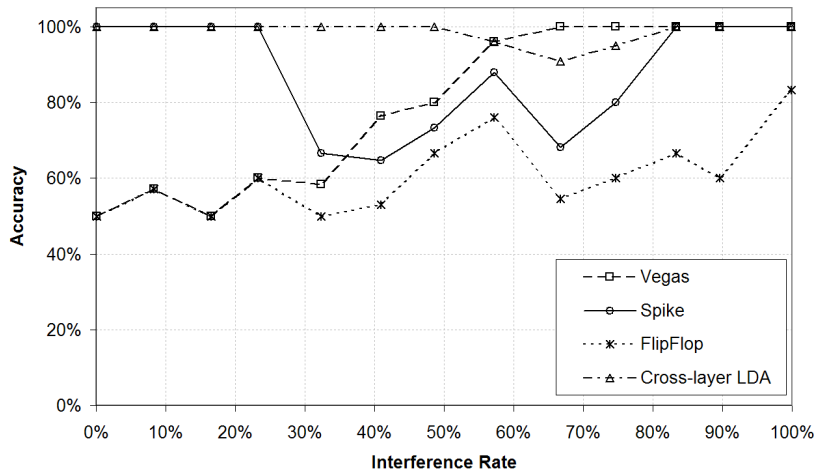


Fig. 2. Accuracy of the 4 LDA schemes with Interferences

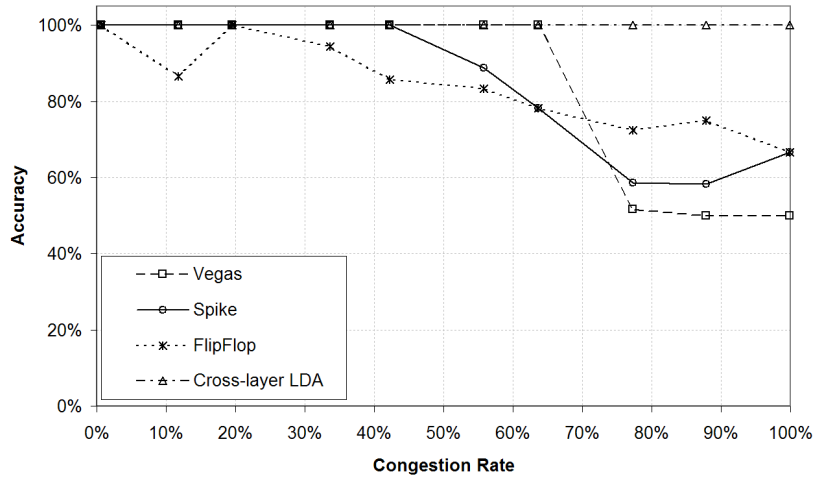


Fig. 3. Accuracy of the 4 LDA schemes with Congestions

4.2 TCP NewReno Enhancement

When the source detects a segment loss, i.e. when 3 duplicate acknowledgements are received, the cross-layer LDA is asked to know the cause of the packet loss:

- If the loss is classified as due to congestion, a normal TCP *NewReno* reaction is triggered and *cwnd* is halved;
- If the loss is classified as due to interferences (short signal loss), *cwnd* is not reduced. This allows the source to achieve higher transmission rates in the event of short suc-

cessive signal losses, if compared to the blind reduction of the throughput performed by the legacy operations of TCP.

This simple extension of the TCP *NewReno* algorithm with an LDA scheme was already proposed in other studies [15]. To this extension, we also added a second adaptation in order to treat the case where a loss due to interference is detected after *RTO*. When a segment-loss, classified as due to interference (i.e. when *LDA_Estimator=1*), is not solved quickly and that a TCP timeout is nevertheless triggered, the idea is to not increment the initial value of *RTO* (in almost all TCP variants, the *RTO* value is doubled after each loss detected through TCP timeout). The aim being to avoid slowing down the loss-recovery process for the following segments.

Simulations are carried out using the same scenarios as for the previous paragraph. Fig. 4 shows the evolution of the average throughput for the TCP flow during congestion or interference periods (the rates are tuned according to the frequencies of the corresponding CBR sources). The maximum *Data Rate* of 25Mbps (*Signal Rate* of 54Mbps) is obtained, according to the ARF procedure, when the station is quite close to its AP. Here, the station is static and the reduction of the throughput is only obtained by varying the rates of the CBR sources. The default values of the TCP window and *Retry Limit* are *Win=32KB* and *RL=6*.

In case of congestion, we verify that there are almost no changes with or without the LDA. Indeed in this situation, the TCP *NewReno* algorithm is not modified allowing this protocol to behave as fairly as the standard TCP protocol in event of congestion. This result confirms moreover the efficiency of the cross-layer scheme to identify congestions. In case of interferences, the throughput is clearly improved for interference rates higher than 40%. Indeed, when the number of duplicate acknowledgements increases (for interference ratio higher than 40%), the non reduction of *cwnd* limits the fall of the TCP throughput. Hence, the slight and linear decrease of the TCP throughput is maintained for interference rates up to 60%. For interference rates above 60%, the maintenance of *RTO* helps to limit the fall in performance and the TCP throughput is maintained.

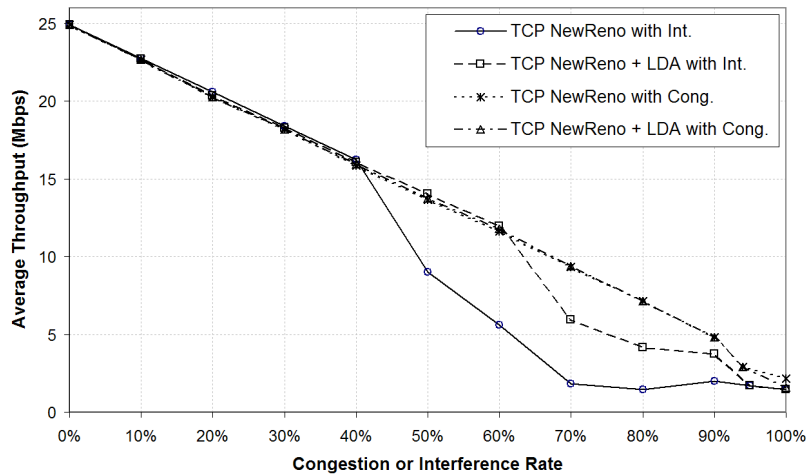


Fig. 4. TCP improvements with Cross-layer LDA

5 Overview of losses and differentiations

Fig. 5 gives an outline of all segment-loss reasons on an 802.11 wireless link:

- The differentiation of cases 4 and 5 is carried out by the cross-layer LDA. In these cases, the TCP *NewReno* adaptation is in charge of improving the performance of elastic flows in the event of interferences (cf. paragraph 4.2).
- The MAC-layer LDA based on *SNR* is used to differentiate the cases 1, 2 and 3 from 4 and 5. For case 3, the *Retry Limit* adaptation is used to improve the performance significantly (cf. paragraph 3.1). Note however that when the station is far away from its AP, the differentiation algorithms do not make possible the distinction between case 3 from the cases 1 and 2. Remember that case 1 is distinguished from case 2 by the cross-layer LDA.

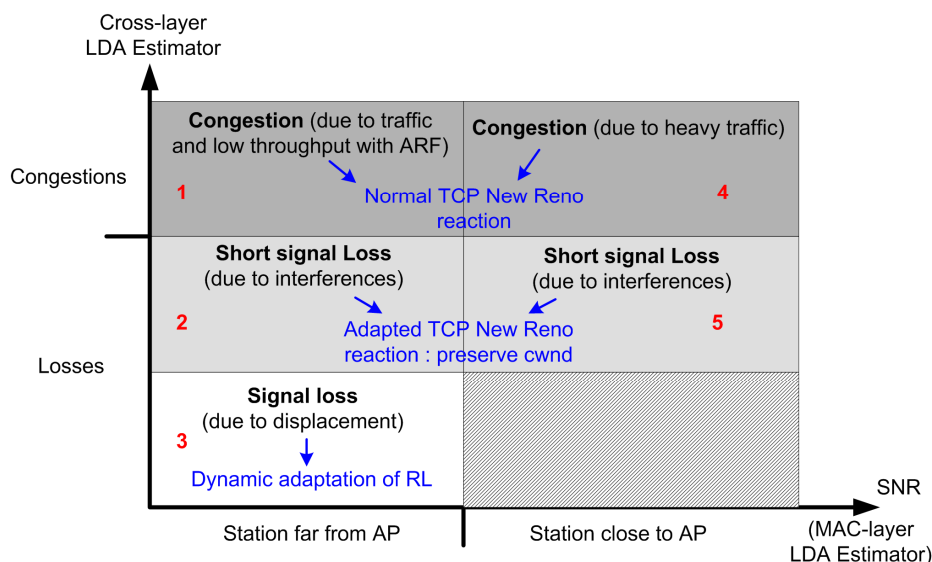


Fig. 5. Loss-recovery for different loss cases

Hence, segment-losses due to distance can not be clearly differentiated from those due to interferences when the *Signal Rate* is reduced to 6Mbps. This is however not constraining. Fig. 6 shows that the evolution of the TCP throughput, when the *Signal Rate* is reduced to 6Mbps, is slightly influenced by the intervention of one or the other LDA. Note however that the use of the MAC-layer LDA gives better improvements compared to the use of the cross-layer LDA. This is more clearly verified for interference rates above 85%. Indeed, for these interference rates, we have an increase of the MAC re-transmissions which limits the number of segment-losses and thus avoids triggering TCP congestion control algorithms. With a reduced *Signal Rate*, the effects of this improvement on the data throughput remain nevertheless limited. Note also that when both proposed LDA schemes are used in conjunction (MAC LDA + cross-layer LDA), the MAC-layer LDA intervention is happening firstly. Finally, Fig. 7 shows that the evolution of the TCP throughput according to the congestion rate is not significantly influ-

enced by the corrections introduced by both LDA schemes. Indeed, in this case a normal TCP reaction is triggered which corresponds to what should be completed. These results thus show the uselessness of a new LDA scheme to distinguish case 3 from cases 1 and 2.

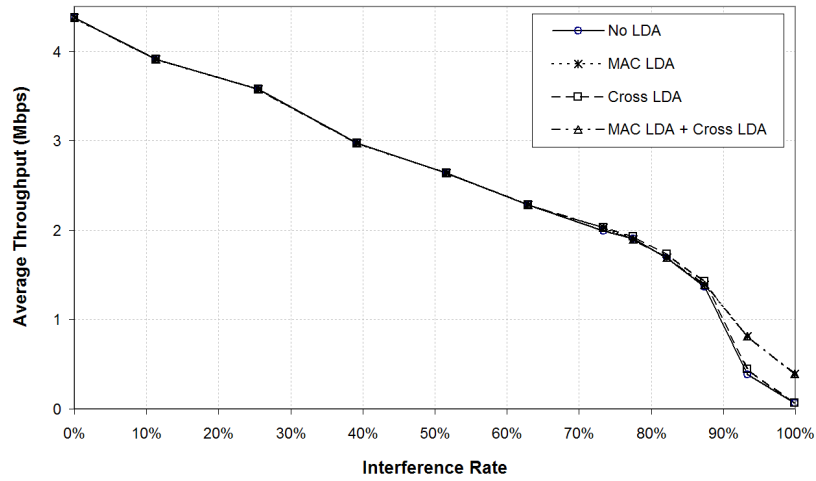


Fig. 6. TCP performance with low throughput (low SNR) in case of interferences

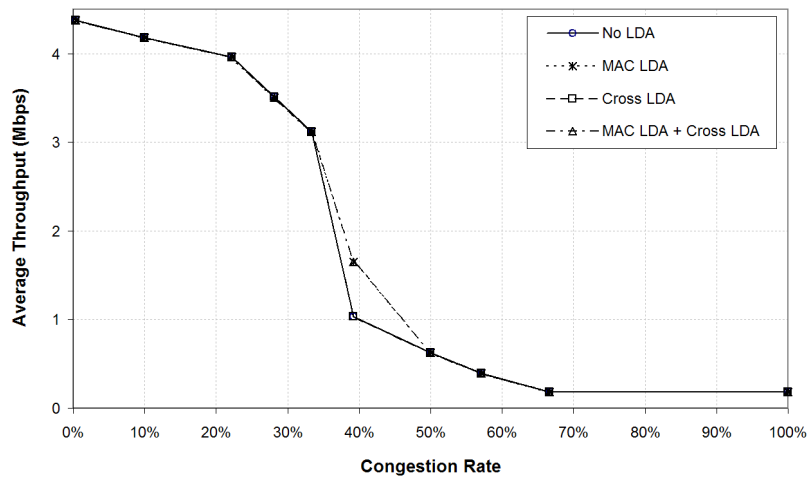


Fig. 7. TCP performance with low throughput (low SNR) in case of congestions

6 Conclusion

According to the characteristics of the various loss causes (mobility and obstacles, interferences caused by other transmissions in the same frequency band and congestion due to increased traffic conditions), we proposed in this paper two Loss Differentiation Algorithms acting respectively at the MAC and TCP layers and both using MAC layer parameters. Depending on the operations of these LDA, adaptations of the MAC and TCP recovery mechanisms are then proposed. The performance evaluation realized with a wireless network close to real situations have highlighted an improvement of the TCP flow performance in all cases. Hence we demonstrated: (i) the gain in performance due to the use of the MAC-layer LDA to react to signal-loss due to mobility and obstacles, and (ii) the efficiency of the cross-layer LDA to distinguish congestions from short losses due to interferences, thus allowing the effective improvement of the TCP behavior.

We are currently working on the optimization of the MAC-layer LDA. Actually, a default value of 6 is used in each RL increase stage. One possible optimization is to use a non-static value for realizing this increase. This value can be derived analytically depending on both TCP and MAC layer parameters.

References

1. IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Standard, IEEE, 1999.
2. IEEE 802.11g WG, Part 11-Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band, 2003.
3. M. Bottigligo, C. Casetti, C.-F. Chiasserini, M. Meo "Short-term Fairness for TCP Flows in 802.11b WLANs", IEEE INFOCOM 2004, Hong Kong, March 7-11, 2004.
4. A. Chockalingam, M. Zorzi, V. Tralli "Wireless TCP performance with link layer FEC/ARQ", IEEE ICC 1999
5. S. Mascolo, M.Y. Sanadidi, C. Casetti, M. Gerla, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks" Wireless Networks J., 2002.
6. K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications", IEEE JSAC, May 2004.
7. S. Cen, P. C. Cosman, and G. M. Voelker "End-to-End differentiation of congestion and wireless losses", IEEE/ACM Transactions on Networking (TON), 2003.
8. S. Bregni, D. Caratti, F. Martignon "Enhanced Loss Differentiation Algorithms for Use in TCP Sources over Heterogeneous Wireless Networks", Globecom 2003.
9. H. Balakrishnan and R. H. Katz "Explicit Loss Notification and Wireless Web Performance", Proc. IEEE Globecom, 1998.
10. C. H. Ng, J. Chow, and L. Trajkovic "Performance Evaluation of TCP over WLAN 802.11 with the Snoop Performance Enhancing Proxy", Opnetwork 2002.
11. S. Lohier, Y.Ghamri Doudane, G. Pujolle "The Benefits of a Cross-Layer Approach for TCP Performance Improvements in WLANs", Proc. IEEE ASWN 2005.
12. The Network Simulator - NS-2, <http://www.isi.edu/nsnam/ns/>.
13. Q. Ni, T. Turletti and W. Fu. "Simulation-based Analysis of TCP Behavior over Hybrid Wireless & Wired Networks", WWIC 2002.
14. S. Biaz and N. H. Vaidya, "Distinguishing Congestion Losses from Wireless Transmission Losses: a Negative Result", Proc. of IEEE 7th Int.Conf. on Computer.
15. D. Barman and I. Matta, "Effectiveness of Loss Labeling in Improving TCP Performance in Wired/Wireless Networks", Boston University Technical Report, 2002.