Simulation-Based Study Of Link-Level Hybrid FEC/ARQ-SR For Wireless Links and Long-Lived TCP traffic

Alaeddine Al Fawal *   Chadi Barakat

PLANETE research group, INRIA - Sophia Antipolis, France
{Alaeddine.Al_Fawal, Chadi.Barakat}@sophia.inria.fr

Abstract: **Since the TCP protocol uses the loss of packets as an indication of network congestion, its performance degrades over wireless links, which are characterized by a high bit error rate. Different solutions have been proposed to improve the performance of TCP over wireless links, the most promising one being the use of a hybrid model at the link-level combining FEC, ARQ-SR (Automatic Repeat Request with Selective Repeat), and an in-order delivery of packets to IP. The drawback of FEC is that it consumes some extra bandwidth to transmit the redundant information. ARQ-SR does not consume much bandwidth, its drawback is that it increases the round-trip time (RTT), which may deteriorate the performance of TCP, if not done appropriately. We study in this paper the performance of TCP over a wireless link implementing hybrid FEC/ARQ-SQ. The study is done by simulating long-lived TCP transfers with ns-2 over wireless links showing Bernoulli errors. We are motivated by how to tune link-level error recovery e.g. amount of FEC, persistency of ARQ, so as to maximize the performance of TCP. We provide simulation results for different physical characteristics of the wireless link (delay, error rate) and for different traffic loads.**

1. Introduction

For TCP, the loss of a packet is an indication that the network is congested. The lost packet is retransmitted by the TCP source and the window is reduced in order to alleviate the congestion of the network. This strategy in the detection of congestion results in a poor performance of the protocol when packets are lost in the network for other reasons than congestion [1,4,5]. Transmission errors on a noisy link, typically a wireless link, form the main source for non-congestion losses. A TCP packet corrupted while crossing a noisy link is discarded before reaching the receiver, which results in an unnecessary window reduction at the TCP source. In the following, we will focus on transmission errors on wireless links and we will call the corrupted TCP packets *non-congestion losses* or *link-level losses* since they appear at a level below IP.

Many solutions have been proposed to improve the performance of TCP when operating on paths with non-congestion losses [1,4,5]. Some of these solutions consist in enhancing TCP with additional mechanisms to help it to recover from non-congestion losses without reducing its window (explicit loss notification [4], loss predictors [6], etc.). Other solutions, e.g. I-TCP [2], propose to shield the sender from these undesirable losses by splitting the TCP connection at the entry of the lossy part of the network, i.e. at the base station in the case of wireless networks. A special well-tuned transport protocol, e.g. STP [10], is then used over the lossy part. Although they improve the overall performance, these solutions break the end-to-end semantics of TCP. A packet is acknowledged before arriving at its final destination.

To preserve the end-to-end semantics of TCP, other promising solutions propose to correct errors at the wireless link level by using a combination of FEC (Forward Error Correction) and ARQ (Automatic Repeat Request) [1,4]. The drawback of FEC is that it consumes some extra bandwidth to transmit the redundant information. It has been shown in [3] that there is an optimal amount of redundancy to be added, above which [1]the performance of TCP degrades instead of improving, although this degradation is slower than the gain in performance we obtain when the first units of redundancy are added. ARQ does not consume much bandwidth, its drawback is that it increases the round-trip time (RTT), which may deteriorate the performance of TCP, if not done appropriately. The throughput of TCP is known to be inversely proportional to the average round-trip time [14,16]. Another problem of ARQ is the interference with TCP timeout. TCP retransmission timer may expire while the lost packet is being retransmitted over the wireless link. FEC does not cause neither an increase in RTT nor an interference with TCP timeout [1]. For these reasons, FEC-alone has been recommended to be used over long delay wireless links as the satellite ones [1].

We study in this paper the performance of TCP over a wireless link implementing a link-level hybrid error correction model implementing FEC, ARQ-SR (ARQ Selective Repeat) and an in-order delivery of packets to IP. ARQ-SR is an efficient ARQ scheme that avoids the unnecessary retransmissions that we see with ARQ Go-Back-N. In contrast to ARQ Stop-Wait, ARQ-SR allows an efficient utilization of the available bandwidth, since

---

* This work has been done while the author was doing his DEA internship in the PLANETE research group at INRIA - Sophia Antipolis.

many packets can be transmitted over the wireless link before receiving any acknowledgment. The main problem with ARQ-SR is that it desequences packets, hence a buffer is needed at the output of the wireless link for the purpose of resequencing packets and delivering them in-order to IP. Out-of-order packets are harmful for TCP since they result in duplicate ACKs, with the TCP source dividing its congestion window by two if three or more duplicate ACKs are received. The combination of FEC and ARQ-SR reduces the number of times we retransmit packets, which shortens the RTT and the resquencing delay in the buffer at the output of the wireless link. On the other hand, this combination reduces the bandwidth available to TCP, since FEC consists in sending redundant information in addition to TCP data. At the same time, this combination reduces the necessary amount of FEC to be used compared to the case when FEC alone is used for error correction. We are motivated in this study by how to tune the parameters of a link-level hybrid FEC/ARQ-SR model so as to maximize the performance of TCP. A typical example of parameters to tune is the amount of FEC and the maximum number of trials we allow ARQ-SR to do before deciding that the packet cannot be locally recovered, and that it has to be recovered by TCP on end-to-end basis.

Our study is done by simulating long-lived TCP transfers with ns-2 simulator [15] over wireless links showing Bernoulli non-congestion losses. For the purpose of the study, a hybrid FEC/ARQ-SR error correction model with in-order delivery has been implemented in ns-2. We have done simulations for different physical characteristics of the wireless link (delay, bandwidth, error rate) and for different traffic loads. For lack of space, we only present in this paper a summary of our simulation results. All our results have been validated by an analytical model based on stochastic processes.

We start by a description of the model, then we present the simulated scenarios. Sections 3 summarizes our results. We conclude the paper by some conclusions and perspectives on our future research.

2. Model

We consider a wireless link where data are transmitted in link-level (LL) frames ***(Figure 1)***. We denote by B the bandwidth of the wireless link, and by D its one-way propagation delay. As we see in ***Figure 1***, each LL frame is divided into K link-level transmission units. A LL transmission unit can be a bit, a byte, or any other data unit. To the K units of a LL frame, we add $N - K$ redundant units that we obtain using a Reed-Solomon code. N is called the length of the code, K its dimension, and K/N its rate. We suppose that we have an erasure channel (the position of the error is known). A LL frame is decoded if we correctly receive K or more units of the frame at the output of the wireless link. A TCP packet that arrives at the input of the wireless link is divided into X frames. All TCP packets are of constant size S (MSS + TCP/IP header). Hence, $S = X \times K$ transmission units. To be delivered to the destination, the X frames of a TCP packet have to be well received. If FEC does not succeed to decode one frame, the error recovery mechanism resorts to ARQ-SR for the retransmission of the frame. The retransmission will be done a maximum number of times, called the persistency of ARQ-SR. We denote this persistency by δ, $\delta = 0, 1, 2, \dots$ When δ trials are done and the frame did not get through the wireless link, ARQ-SR assumes that the frame cannot be locally recovered, and leaves for TCP the correction of the frame on end-to-end basis.

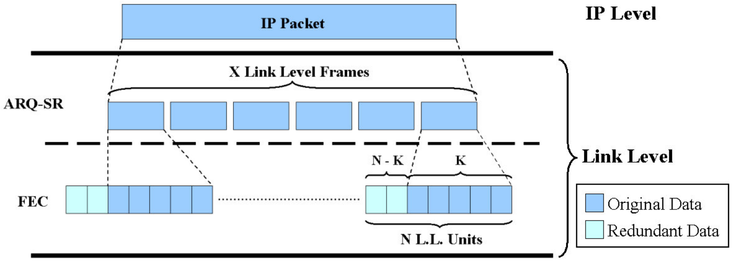

Figure 1 : The Hybrid Model FEC / ARQ-SR.

The ARQ-SR receiver at the output of the wireless link acknowledges each LL frame either with a positive ACK or a NACK. When a NACK is received at the input of the wireless link, the corresponding frame is directly retransmitted, and given priority over all other frames that have not yet been transmitted. One can imagine the use of a priority queue for retransmissions. The packets correctly received at the output of the wireless link are resequenced before being delivered to IP. An out-of-order delivery takes place only if a packet cannot be locally corrected, due to the failure of the retransmission of one of its frames. Note that packets are resequenced at the output of the wireless link on an aggregate basis, not on a flow basis. This simplifies the implementation of the error recovery mechanism and respects the principle of layering in the Internet.

Concerning errors over the wireless link, we model them using a Bernoulli process, where transmission units are lost independently of each other with probability p. The error process is only applied to data packets. ACKs of ARQ-SR and those of TCP are supposed to cross the wireless link without being dropped, given their small size. Usually, errors over wireless links are modeled by a Markov chain of two states [3,7,8,11,12]. We choose the Bernoulli process for its simplicity. The Bernoulli process is known to hold over fast fading channels (high speed users) [12] and when interleaving is used [4]. We are only interested in this paper by the impact of the average error rate. The impact of burstiness of errors will be the subject of a future research.

Our traffic is composed of C long-lived TCP connections crossing the wireless link in the same direction. The focus is on how to maximize the aggregate throughput of the C connections, and hence the utilization of the wireless link.

3. Simulations

We implement our model for link-level FEC/ARQ-SR with in-order delivery into ns-2, the network simulator developed at LBNL [15]. The model is only applied to the wireless link, and is transparent to the rest of the network and to TCP peers. Our simulations are divided into two parts. The first part focuses on the utilization of the wireless link in case of C = 10 concurrent TCP connections. The second part focuses on the throughput of one TCP connection using alone the wireless link for its transfer. In all studied scenarios, K is fixed to 10, X to 6 and the size of link-level units to 25 bytes. TCP packets are then of constant size equal to 1500 bytes (MTU of Ethernet). Simulations are run for 2000 seconds. This relatively long duration is necessary to absorb the initial slow start phase of TCP.

The values given to K and X are randomly chosen. Other values are possible. Our purpose is not to optimize these quantities, but rather to optimize the amount of FEC to be used and the persistency level δ. We look for K and X as inputs of the problem rather than outputs. This reduces the number of parameters to optimize. Note that the optimization of K and X is also an interesting problem given the tradeoff it involves. We are studying this optimization problem with our analytical model. For instance, for constant packet sizes (constant S), the size of LL frames decreases when X increases, which decreases the probability that a LL frame is corrupted. This should improve the performance. However, when we increase X, K decreases, hence the number of redundant units per LL frame decreases, if we do not change the amount of FEC. It has been shown in [3] that this may deteriorate the performance since frames can now resist to less errors. The optimization problem becomes more interesting when we allow the packet size S to change. The packet size decides on the rate with which TCP increases its congestion window [3,16].

3.1 First part: Link utilization

3.1.1 Simulation scenarios

In this section, we present how the utilization of the wireless link varies according to the various parameters of FEC and ARQ-SR. Ten long-lived TCP connections cross the wireless link in the same direction. We considered several scenarios that are derived from the network topology shown in ***Figure 2***.

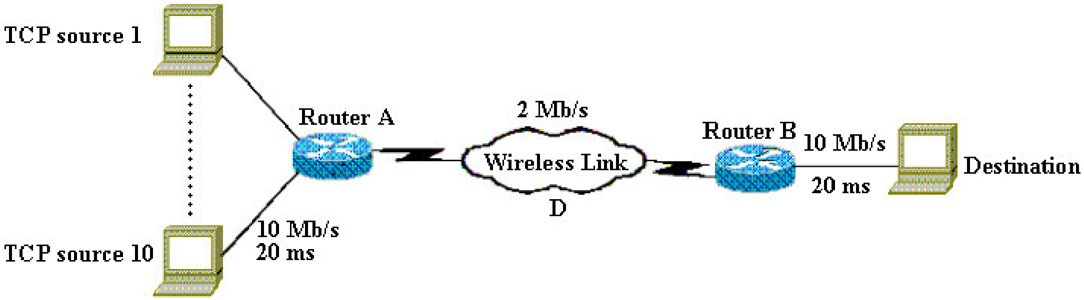

Figure 2 : Network Topology.

As the figure shows, there are 10 TCP sources that transmit FTP data simultaneously and continuously to the same destination. Each source corresponds to one TCP connection. The sources use the NewReno version of TCP. The sources and destination are connected to the wireless link via 10 Mbps links having a one-way propagation delay equal to 20 ms. The bandwidth of the wireless link B is set to 2 Mbps, its delay D varies between 20 ms and 200 ms depending on the scenario. The values we give to D model different types of wireless links ranging from terrestrial links to satellite ones. Transmission units are lost (corrupted) over the wireless link with probability p, with the value of p ranging from 10^{-5} to 10^{-2}.

The destination enables the Delay ACK functionality of TCP [16]. To avoid any limitation of traffic due to other reasons than errors in the wireless link, we took the following measures:

- The wired links are completely reliable.
- The size of TCP advertised window is very large, up to 2000 packets.
- The buffers in network routers are very large, they can store up to 500 packets.
- The buffer used for resequencing packets at the output of the wireless link is very large.

Under these conditions, it is clear that the wireless link is the only bottleneck on the path of the TCP connections. Congestion losses do not appear in network routers unless the wireless link is fully utilized. We want to optimize the parameters of FEC and ARQ-SR in this setting. The optimization problem is not meaningful when the TCP connections are constrained by some other parts of the network.

We begin by studying separately the effects of FEC and ARQ-SR on the utilization of the wireless link, i.e. the effect of FEC alone and that of ARQ-SR alone. Second, we study the performance of the hybrid model, i.e. FEC and ARQ-SR are combined together. Intuitively, the need for FEC is important when the delay of the wireless link is large and when the error rate high, since the use of ARQ-SR in this case results in a considerable increase in the round-trip time. In the other scenarios (lower delay, less losses), FEC utility is reduced since the increase in RTT caused by ARQ-SR retransmissions has less impact on the utilization than the amount of bandwidth consumed by FEC. This reasoning is confirmed by our simulations. For lack of space, we only present the challenging scenarios that yield the most significant results.

3.1.2 FEC alone

The case FEC alone can be obtained by setting $\delta = 0$. In this case, LL frames are not retransmitted but only protected by FEC. ***Figure 3*** shows three lines that illustrate the variation of the utilization of the wireless link as a function of the parameter of FEC "N", which we recall models the amount of redundancy. These three lines correspond to three distinct values of the delay, D = 20, 100, 200 ms. In all the cases, p is set to a high value 0.01. Clearly, there is an important improvement of the utilization with the first units of redundancy. When N=12, the maximum utilization is attained. At this point, the amount of FEC is optimal; about 80% of link bandwidth is used by TCP data, and the remaining 20% is used by the redundant information. Increasing N beyond 12 results in a decrease in the utilization, but this decrease is slower than the increase in the utilization on the left-hand side of the optimal point. The same behavior has been observed in [3]. For all values of N on the right-hand side of the optimal point, the amount of FEC is more than necessary. One should expect that the decay of the utilization on the right-hand side of the optimal point is given by K.B/N.

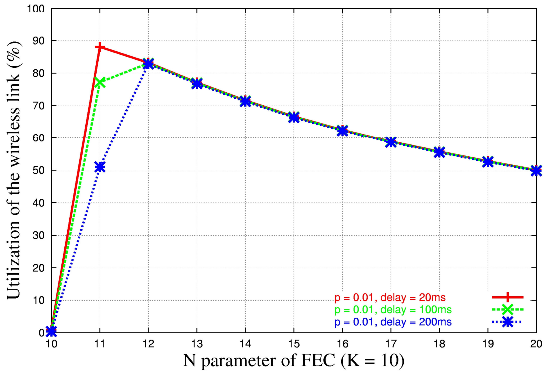

Figure 3 : Utilization of the Wireless Link for 10 TCP Connections, FEC Alone.

3.1.3 ARQ-SR alone

Now, we examine the effect of ARQ-SR alone by looking at the values of the utilization for N = K (=10). The results are plotted in ***Figure 4***. This figure shows the utilization of the wireless link plotted as a function of δ (ARQ-SR persistency). We examine two cases: (i) the in-order delivery of packets to IP at the output of the wireless link is activated, which is indicated in the figure by ord = 1, and (ii) the in-order delivery of packets to IP is not activated, which is indicated in the figure by ord = 0. For D and p, we consider the same values as above (case of FEC alone); we have three values of delay, D = 20, 100, 200 ms, and p is set to 0.01. Surprisingly, the utilization is always increasing with δ, even though we are dealing with an extreme case where the delay D is large and the loss rate is high. ARQ-SR reduces the loss rate of TCP packets much more than it increases the end-to-end delay, which results in this monotonous improvement. We notice in these simulations that the resequencing of packets at the output of the wireless link is essential to obtain good performance with ARQ-SR, otherwise packets arrive out-of-order at the TCP receiver and trigger the transmission of duplicate ACKs, something very harmful for TCP since it results in unnecessary division of the congestion window. Another surprising result is that with ARQ-SR alone, when the reordering of packets is activated, we can reach higher utilization than what we can reach with the optimal amount of FEC, when FEC is used alone. The same finding applies to other scenarios with smaller p, and it is even more pronounced in favor of ARQ-SR.

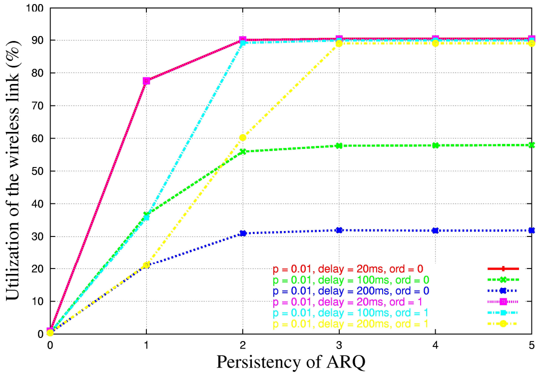

Figure 4 : Utilization of the Wireless Link for 10 TCP Connections, ARQ-SR alone

3.1.4 Hybrid FEC/ARQ-SR

Now, we present the results we obtain when we use the hybrid FEC/ARQ-SR with in-order delivery. We consider the most challenging scenario, where the delay of the wireless link is the largest D = 200 ms, and where the error rate is the highest p = 0.01. For this scenario, ***Figure 5*** shows the utilization of the wireless link as a function of the parameter of FEC “N”. We see six lines in the figure that correspond to six values of persistency, $\delta = 0, 1, 2, 3, 4, 5$. The line $\delta = 0$ gives the impact of FEC alone on the utilization, it is the same line that appears in ***Figure 3*** for the tuple (p , D) = (0.01 , 200 ms). By looking at the values of the utilization on the y-axis, i.e. for N = K = 10, one can examine the effect of ARQ-SR alone, which is detailed in ***Figure 4***. By combining FEC and ARQ-SR, we hope to realize better performance than when using both schemes separately. The results in ***Figure 5*** seems contradicting this idea, at least under the assumptions of our analysis (long-lived TCP transfers, Bernoulli errors). We remark that the best performance a hybrid FEC/ARQ-SR scheme can give is close to what is given by ARQ-SR alone (for $\delta = 5$). No more than one unit of redundancy (N=K+1) is needed to attain the highest utilization.

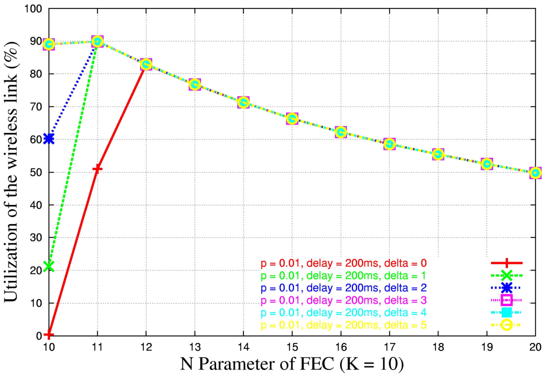

Figure 5 : Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model
D = 200 ms, p = 0.01

In the other scenarios where the delay and the error rate are smaller (like in ***Figure 6*** which corresponds to smaller p = 0.001 and same D = 200 ms), our simulation results show that there is no need at all for FEC, and that ARQ-SR alone is able to realize the best performance. This is a surprising result, but it seems logical, since

FEC consumes extra bandwidth all time, whereas ARQ-SR consumes extra bandwidth only when packets are lost. One idea could be to use FEC only to protect retransmissions, not original frames. We are currently investigating this idea by simulations and analytical modeling.

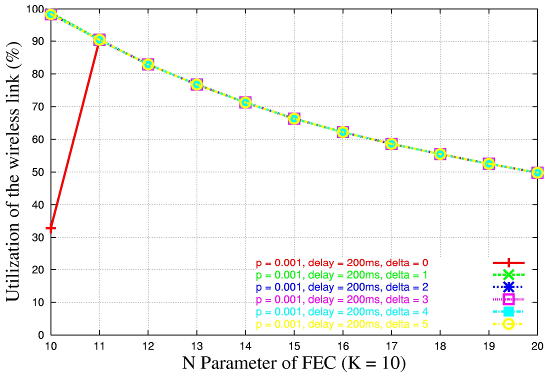

Figure 6 : Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model
D = 200 ms, p = 0.001

3.2 Second part: Throughput of a single TCP connection

When 10 TCP connections are used, transmission errors are spread over all the connections, so their impact on one connection is smaller than when a TCP connection alone is active over the wireless link. We want to study the impact of the hybrid error recovery mechanism in the extreme case when the TCP connection suffers alone from transmission errors, and try to optimize the parameters of FEC and ARQ-SR for this case. Clearly, more effort (more FEC, larger δ) is needed in the case of one connection than in the case of 10 connections to achieve full link utilization.

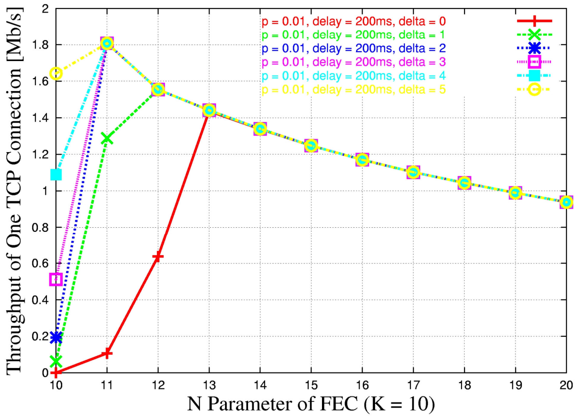

Figure 7 : Throughput of One TCP Connection over the Wireless Link, Hybrid Model.

We consider the same network topology as the one used in the previous section, with the difference that now we have one TCP source that corresponds to one TCP connection. For lack of space, we only present two sets of results. The first set (***Figure 7***) corresponds to the extreme case D = 200 ms and p = 0.01. The y-axis in the figure shows the throughput of the TCP connection expressed in Mbps. We notice the same trend as that in the previous section (case of 10 TCP connections). With ARQ-SR alone, TCP is able to achieve very high utilization, which can not be realized by FEC alone. The optimal operating point of the hybrid scheme is close to that of ARQ-SR alone. We also notice how we need a larger δ in this case to achieve full utilization of the wireless link. The optimal utilization is equal to that obtained for 10 connections, which is a good indication that the maximum utilization we can reach with a hybrid FEC/ARQ-SR is roughly independent of the number of TCP connections.

The second set of results corresponds to the variation of the throughput of TCP as a function of the persistency level when ARQ-SR alone is used. We want to be sure that the utilization of the wireless link is always increasing with δ, a result that we have shown in the case of 10 TCP connections. We plot the throughput of one TCP connection for different values of δ ranging from 0 to 10. This second set of results is given in

Figure 8. The x-axis in the figure corresponds to δ and the y-axis to the throughput of TCP. For these results, the value of p is set to 0.01 and the in-order delivery of packets at the output of the wireless link is enabled (ord=1). The figure shows three lines that correspond to three distinct values of D; D = 20, 100, 200 ms. For the three lines in the figure, the throughput of TCP starts by quickly increasing then saturates. No degradation in the throughput caused by an increase in δ is noticed. Based on our simulations results, we conclude that under the assumptions of our study, the utilization of the wireless link is always monotonously increasing with δ, whatever are the number of TCP connections, the delay and the loss rate.

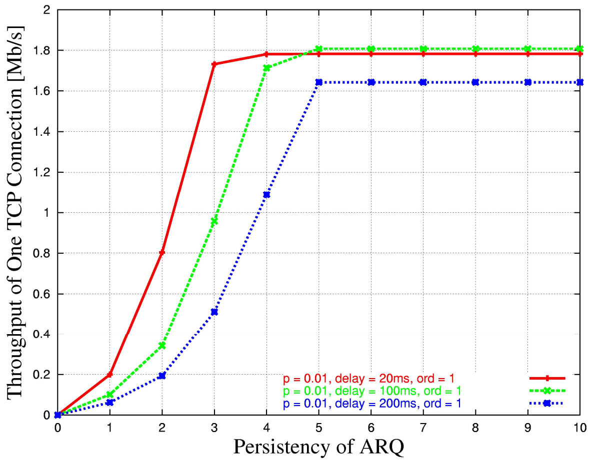

Figure 8 : Throughput of One TCP Connection over the Wireless Link, ARQ-SR alone.

4 Conclusions

The key finding of our analysis is that for long-lived TCP transfers and Bernoulli errors, the use of ARQ-SR is more beneficial than the use of FEC, even in extreme cases where the delay is large and the error rate is high. As a consequence, the maximum utilization we can reach with a hybrid FEC/ARQ-SR is roughly independent of the number of TCP connections. For a certain persistency δ, there is an optimal amount of FEC to be added in order to achieve full utilization of the wireless link. Any extra amount of FEC deteriorates the performance instead of improving it.

Our work can be extended in different directions. We want to check whether our finding holds in the case of finite TCP transfers and bursty transmission errors. Another possible extension is to study what will be the optimal error recovery scheme when delay-sensitive non-TCP traffic is used, as voice and video streams.

References

[1] M. Allman, D. Glover, L. Sanchez, Enhancing TCP over satellite channels using standard mechanisms, RFC 2488, January 1999.
[2] A. Bakre, B.R. Badrinath, I-TCP: indirect TCP for mobile hosts, International Conference on Distributed Computing Systems, May 1995.
[3] C. Barakat and E. Altman, Bandwidth tradeoff between TCP and link-level FEC, Computer Networks, vol. 39, no. 2, pp. 133-150, June 2002.
[4] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R. Katz, A comparison of mechanisms for improving TCP performance over wireless links, ACM SIGCOMM, August 1996.
[5] C. Barakat, E. Altman, W. Dabbous, On TCP performance in a heterogenous network: a survey, IEEE Communications Magazine 38 (2000) 40 46.
[6] S. Biaz, N.H. Vaidya, Distinguishing congestion losses from wireless transmission losses: a negative result, Seventh International Conference on Computer Communications and Networks (IC3N), October 1998.
[7] H. Chaskar, T.V. Lakshman, U. Madhow, On the design of interfaces for TCP/IP over wireless, IEEE MILCOM, 1996.
[8] A. Chockalingam, M. Zorzi, R.R. Rao, Performance of TCP on wireless fading links with memory, IEEE ICC, June 1998.
[9] E.N. Gilbert, Capacity of a burst-noise channel, Bell Systems Technical Journal 39 (1960) 1253 1266.
[10] T. Henderson, R.H. Katz, Transport protocols for Internet- compatible satellite networks, IEEE Journal on Selected Areas in Communications 17 (1999) 326 344.
[11] V. Jacobson, Congestion avoidance and control, ACM SIGCOMM, August 1988.

[12] A. Kumar, J. Holtzman, Performance analysis of versions of TCP in a local network with a mobile radio link, Sadhana: Indian Academy of Sciences Proceedings in Engineering Sciences, February 1998.

[13] T.V. Lakshman, U. Madhow, The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Transactions on Networking 5 (1997) 336 350.

[14] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, Computer Communication Review 27 (1997) 67 82.

[15] The LBNL Network Simulator, ns-2, available at http://www.isi.edu/nsnam/ns/

[16] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, ACM SIGCOMM, September 1998.

[17] C. Stevens, TCP Slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, RFC 2001, january 1997.