

Analysis of link-level hybrid FEC/ARQ-SR for wireless links and long-lived TCP traffic*

Chadi BARAKAT [†], Alaeddine AL FAWAL [‡]

INRIA, Planète research group
2004, route des Lucioles
06902 Sophia Antipolis - France

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 (Forward Error Correction), 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 consumes extra bandwidth only when packets are lost, its drawback is that it increases the round-trip time (RTT), which may deteriorate the performance of TCP. Another drawback of ARQ-SR is that a complete packet can be retransmitted to correct a small piece of errored data. We study in this paper the performance of TCP over a wireless link implementing hybrid FEC/ARQ-SR. The study is done by simulating and modeling long-lived TCP transfers 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 results for different physical characteristics of the wireless link (delay, error rate) and for different traffic loads (number of TCP connections).

1 Introduction

For TCP the loss of a packet is an indication that the network is congested [14, 20]. 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, e.g. [1, 4, 7, 9]. 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 only focus on transmission errors over 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, e.g. [1, 3, 4, 9, 11]. 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 [9], etc.). Other solutions, e.g. I-TCP [3], propose to shield the sender from these undesirable

*The simulation results in this paper have appeared in proceedings of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Sophia Antipolis, France, March 2003. The analytical model is an original work.

[†]Corresponding author, email: Chadi.Barakat@sophia.inria.fr

[‡]This work was done while the author was doing his DEA internship in the PLANETE research group at INRIA - Sophia Antipolis.

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 [13], 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, 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, 7]. The drawback of FEC is that it consumes some extra bandwidth to transmit the redundant information. It has been shown in [7] that there is an optimal amount of redundancy to add, above which 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 [17, 19]. 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. ARQ can also be inefficient when large packets are retransmitted to correct a small piece of errored data inside each packet. On the other hand, FEC does not cause neither an increase in RTT nor an interference with TCP timeout [1]. So each one of the two methods has its own advantages and drawbacks, and the choice of the best method is not always clear.

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 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 the wireless link retransmits packets. This (i) shortens the RTT and the resquencing delay in the buffer at the output of the wireless link, and (ii) reduces the bandwidth wasted on retransmissions. At the same time, this reduces the amount of FEC compared to the case when FEC alone is used for error correction. But, this combination also reduces the bandwidth available to TCP, since FEC consists in sending redundant information in addition to TCP data. 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. We call this maximum number of trials the persistency of ARQ-SR. Our study is done by simulating and modeling long-lived TCP transfers over wireless links showing Bernoulli non-congestion losses. The simulations are done with the ns-2 simulator [18]. The modeling is done with an analytical approach close to that used in [7]. 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 consider different physical characteristics of the wireless link (delay, error rate) and different traffic loads (number of TCP connections). First, we present a summary of our simulation results. Then, we present our analytical model and we show that it approximates very well the simulation results. We explain at the end how our analytical model can serve for the optimization of the link-level error recovery mechanism.

Our work is done under two main assumptions: the wireless link shows Bernoulli non-congestion losses and the TCP connections are long-lived. Usually, losses over wireless links are modeled by a Markov chain of two or more states since they appear in bursts [7, 11, 12, 15]. The first reason to choose the Bernoulli model is the simplicity of the analysis it allows. The second reason is that it is not biased against FEC. It is known that FEC suffers of low performance when losses appear in bursts [7]. ARQ-SR does not have this problem. So by assuming Bernoulli losses, we are able to compare both mechanisms in a fair environment where only the average error rate figures. The study of the bursty case is still important, we leave it for a future research. Note that the Bernoulli model is known to hold over fast fading channels (high speed mobile users) [15] and when interleaving is used [4].

As for the assumption that the traffic is composed of long-lived TCP connections, we are aware of the fact that this model is not very realistic, but we believe at the same time that the study of long-lived TCP traffic is an

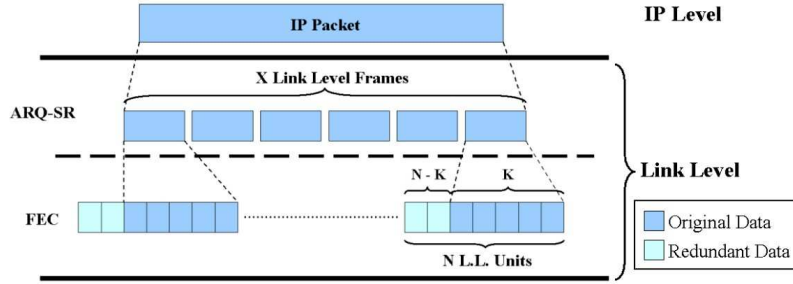


Figure 1: The hybrid model FEC / ARQ-SR

important step before studying other models for Internet traffic that represent better the reality, e.g. models for HTTP traffic. Note that long-lived TCP connections are known to carry an important part of Internet traffic, and this part is expected to increase with the advent of new data-greedy applications as P2P and Grids [10].

Our findings can be summarized as follows. For long-lived TCP transfers and Bernoulli errors, the throughput of the wireless link is an increasing function of the persistency of ARQ-SR, even in extreme cases where the delay is large and the error rate is high. This is because timeouts are rare and TCP adapts its retransmission timer to the variation of delay caused by link-level retransmissions. The throughput increases with the persistency of ARQ-SR until the wireless link is fully utilized. Hence, a large persistency is always needed for a good performance. At moderate error rates, a large persistency is sufficient to achieve the best throughput, and FEC is not needed, or may be needed in very small amounts. However at high error rates, ARQ-SR alone is not enough to achieve the best throughput. Even though a large persistency achieves a full utilization of the wireless link, the throughput is very low in this latter case since most of the packets are lost and the bandwidth is mostly used by link-level retransmissions. The addition of some FEC to this latter case reduces the error rate, which reduces the amount of bandwidth wasted on retransmissions and improves the throughput. In fact, for every error rate there is an optimal amount of FEC to be added above which the performance deteriorates instead of improving. Our analytical model provides an expression for this optimal amount of FEC. As we will see, this optimal amount of FEC is increasing with the error rate, and it is close to zero when the error rate is low to moderate.

Another important remark we make in this paper is that in this context of long-lived TCP traffic and in the case of a large ARQ-SR persistency, the optimal amount of FEC to be added is independent of the number of TCP connections and of the delay of the wireless link. The same thing can be said about the maximum throughput we can reach with a hybrid FEC/ARQ-SR. Now if one wants to use a small persistency level (due for example to the presence of another type of traffic sensitive to retransmissions), there is again an optimal amount of FEC to be used, but this optimal amount of FEC will not lead to the best throughput of the wireless link as that one can reach with a large ARQ-SR persistency and an amount of FEC properly chosen.

We start by a description of the model of our study in Section 2. Section 3 summarizes our scenarios and explains our simulation results. The analytical model and its solution are presented in Section 4. We conclude the paper in Section 5 by some conclusions and perspectives on our future research.

2 The 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. An LL transmission unit can be a bit, a byte, or any other data unit. To the K units of an LL frame, we add $N - K$ redundant units that we obtain using a block code FEC, e.g. Reed-Solomon [12]. 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). An 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 decoded. If FEC does not succeed to decode one LL frame, the link-level error recovery mechanism

resorts to ARQ-SR for the retransmission of the frame. The retransmission will be done a maximum number of times that we call the persistency of ARQ-SR and that we denote by δ , $\delta = 0, 1, 2, \dots$. $\delta = 0$ means that there are no retransmissions and that ARQ-SR is disabled. 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.

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 or more. 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.

We suppose that transmission units are lost independently of each other with probability p (Bernoulli model). 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 sizes. As for the traffic, we consider that it 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 as a consequence the throughput of the wireless link. The throughput of the wireless link is measured by counting the number of bytes that leave the wireless link and by dividing the sum by the simulation time. The throughput is then divided by the link speed (2Mbps) and expressed in the figures in percents.

3 Simulation study

3.1 Simulation scenarios

We implement our model for link-level FEC/ARQ-SR with in-order delivery into ns-2, the network simulator developed at LBNL [18]. The model is only applied to the wireless link, and is transparent to the rest of the network and to TCP peers. We present two sets of simulation results. The first set focuses on the throughput of the wireless link in case of $C = 10$ concurrent TCP connections. The second part focuses on the throughput of the wireless link when one TCP connection uses alone the wireless link for its transfer. With no loss of generality, we set K 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 and to obtain convergence of throughput measurements.

The values given to K and X are arbitrary chosen. We ran simulations for other values of X and K and we noticed that our findings always hold. Note that our purpose is not to optimize X and K , but rather to optimize the amount of FEC to be used and the persistency level δ , given the values of K , X , and of the other parameters of the model. We look for K and X as inputs of the problem rather than outputs. Nevertheless, the optimization of K and X is an interesting problem given the tradeoff it involves. We leave this optimization problem for a future research. For instance, for constant packet sizes (constant S), the size of LL frames decreases when X increases, which decreases the probability that an LL frame is corrupted and hence the bandwidth wasted on retransmissions. This should improve the performance. However when we increase X , K decreases as well as the number of redundant units per LL frame, this is if we do not change the amount of FEC. It has been shown in [7] that this may deteriorate the performance since frames can now resist less errors. The optimization problem becomes more interesting when we allow the packet size S to change. The packet size decides the rate with which TCP increases its congestion window [7, 17, 19, 20].

Next, we present how the throughput of the wireless link varies in simulations according to the various parameters of FEC and ARQ-SR. We consider several scenarios that are derived from the network topology shown in Figure 2. As the figure shows, there are multiple 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 and enable the Delay ACK functionality [20]. The sources and destinations 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

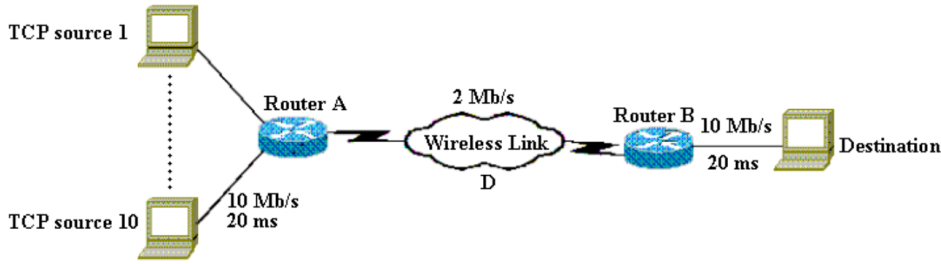


Figure 2: Simulated network topology

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} .

To avoid any limitation of traffic due to other reasons than errors in the wireless link, we take the following measures. (i) The wired links are completely reliable. (ii) The size of TCP advertised window is very large, up to 2000 packets. (iii) The buffers in network routers are very large, they can store up to 500 packets. (iv) 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 best case setting. The optimization problem is not meaningful when the TCP connections are constrained by some other parts of the network.

3.2 First set of results: link throughput for 10 connections

We begin by studying separately the effects of FEC and ARQ-SR on the throughput 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 combined together. For lack of space, we only present results for the most challenging scenarios among the set of scenarios we simulated. By challenging scenarios we mean those where the delay is large and where the error rate is high.

In these extreme scenarios, the need for FEC is the most important since the use of ARQ-SR results in a large increase in the round-trip time and a non-negligible waste of bandwidth on retransmissions, which might impair the performance of TCP and favor the use of FEC. In the other scenarios (smaller delay, less losses), FEC utility is reduced since the increase in RTT caused by ARQ-SR and the bandwidth consumed by retransmissions are smaller. This reasoning is confirmed by our simulations, which indeed show that the need for FEC is more important when the error rate is high. Concerning the link delay, if the buffers at the input of the wireless link are properly dimensioned and if the persistency level δ is set to a large value, the wireless link will be fully utilized and the throughput will be independent of the delay and of the number of connections. The throughput of the wireless link will only depend on the amount of FEC, and clearly on the error rate; we will see that an optimal amount of FEC always exists and this amount increases with the error rate.

3.2.1 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 throughput 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, the transmission unit loss probability p is set to a high value 0.01. Clearly, there is an important improvement in the throughput with the first units of redundancy and this improvement stops when some optimal point is reached (e.g., $N = 11$ for $D = 20$ ms). Increasing N beyond the optimal point results in a decrease in the throughput, but this decrease is slower than the increase in the throughput on the left-hand side of the optimal point. The same behavior has been observed in [7]. 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 throughput on the right-hand side of the optimal point is given by K/N .

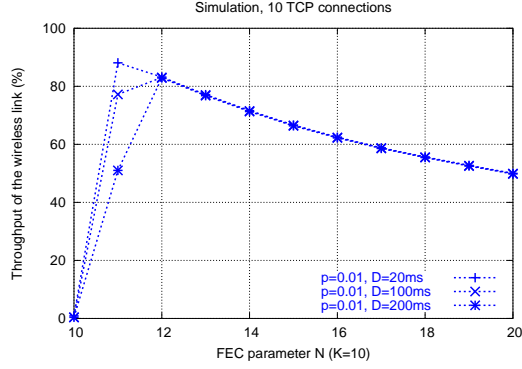


Figure 3: Simulation: Throughput of the wireless link for 10 TCP connections, FEC alone, $p = 0.01$

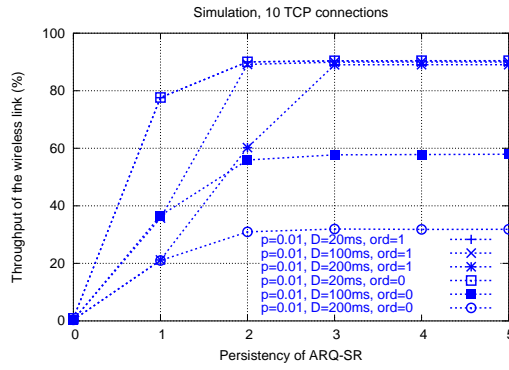


Figure 4: Simulation: Throughput of the wireless link for 10 TCP connections, ARQ-SR alone, $p = 0.01$

As for the utilization of the wireless link, it is less than 100% on the left-hand side of the optimal point and equal to 100% on its right-hand side. The utilization measures the amount of bytes that are transmitted over the wireless link. Some of these bytes are not useful (not counted in the throughput) because of FEC and corrupted frames.

3.2.2 ARQ-SR alone

Now, we examine the effect of ARQ-SR alone by looking at the values of the throughput for $N = K (= 10)$. The results are plotted in Figure 4. This figure shows the throughput 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 throughput of the wireless link is always increasing with δ , even though we are dealing with an extreme case where the delay D is large and the loss rate p is high. The reason is that 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 in performance. We also notice 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 reduction of the congestion window.

Another surprising result is that with ARQ-SR alone, when the reordering of packets is activated, we can reach higher throughput 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 D , and it is even more pronounced in favor of ARQ-SR. Now for larger values of p , one should expect worse performance of ARQ-SR alone since most of the frames will

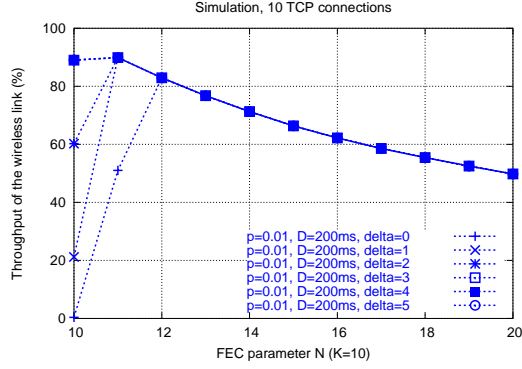


Figure 5: Simulation: Throughput of the wireless link for 10 TCP connections, hybrid model, $D = 200$ ms, $p = 0.01$

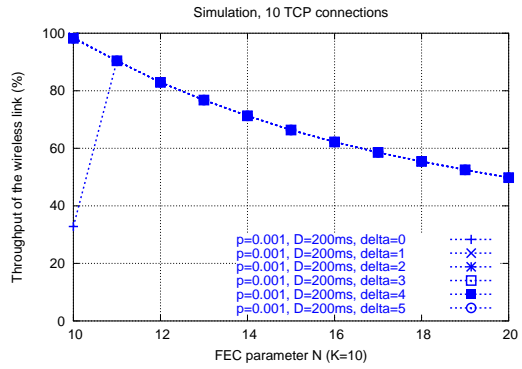


Figure 6: Simulation: Throughput of the wireless link for 10 TCP connections, hybrid model, $D = 200$ ms, $p = 0.001$

be lost and retransmitted.

Two main things can be retained from Figure 4. First, the throughput of TCP, and hence that of the wireless link, increases with δ for all values of p and D . Second, when the buffer at the input of the wireless link is large as in our case, the maximum throughput obtained with ARQ-SR alone is independent of the delay (when reordering is activated). Clearly, the maximum throughput depends on the error rate.

3.2.3 Hybrid FEC/ARQ-SR

Now, we present the results we obtain when we use hybrid FEC/ARQ-SR with in-order delivery. We consider the challenging scenario where $D = 200$ ms and $p = 0.01$. Figure 5 shows the throughput 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 throughput, it is the same line that appears in Figure 3 for the tuple $(p, D) = (0.01, 200)$ ms. If we look at the values of the throughput on the y-axis, i.e. for $N = K = 10$, we 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 our settings (long-lived TCP transfers, Bernoulli errors, $p = 0.01$). We remark that the best performance a hybrid FEC/ARQ-SR scheme can provide 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 maximum throughput.

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 good performance of ARQ-SR is a surprising

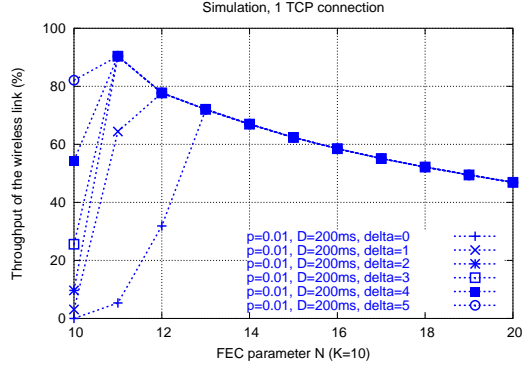


Figure 7: Simulation: Throughput of the wireless link for one TCP connection, hybrid model, $D = 200$ ms, $p = 0.01$

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.

Now, when the error rate p increases beyond 0.01, one should expect that FEC becomes more and more necessary, since ARQ-SR alone will result in lot of retransmissions. But FEC must not be added in large amounts, otherwise the throughput of the wireless link will not be optimal. In general, there is an optimal amount of FEC to add and this optimal amount is an increasing function of the error rate. As for δ , it is always good to set it to a large value. Our analytical model presented later gives more explanations on how to tune the error recovery mechanism so as to achieve the best performance.

3.3 Second set of results: link throughput for one connection

When 10 TCP connections are used, transmission errors are spread over all the connections, so their impact on one TCP connection is smaller than when the connection is active alone over the wireless link. We want to study here the impact of the hybrid error recovery mechanism in the extreme case when one 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 case of one connection than in the case of 10 connections to achieve full link utilization. But, since the throughput of TCP is increasing with δ , and if one uses a large δ and there is enough buffer at the input of the wireless link to allow the TCP connection to fully utilize the available bandwidth, one should expect that the performance of the wireless link will be independent of the fact that we have one TCP connection. Normally for large δ , the performance should only depend on the error rate and on the amount of FEC used.

We consider the same network topology as that used in the previous section, with the difference that now we have only one TCP connection. For lack of space, we only present two sets of results. The first set (Figure 7) corresponds to $D = 200$ ms and $p = 0.01$. The y-axis in the figure shows the throughput of the wireless link and the x-axis shows the amount of FEC modeled by "N". We notice the same trend as that in the previous section (case of 10 TCP connections). The optimal operating point of the hybrid scheme is close to that of ARQ-SR alone (this is certainly not the case for higher error rates). We also notice how we need a larger δ in this case to achieve full utilization of the wireless link. The optimal throughput is close to that obtained for 10 connections, which confirms our claim that the maximum throughput we can reach with a hybrid FEC/ARQ-SR is roughly independent of the number of TCP connections. Clearly, this requires that the buffer size at the input of the wireless link is properly dimensioned so that the TCP traffic is able to fully utilize the available bandwidth (and that the TCP connection is not constrained by other parts of the network).

The second set of results corresponds to the variation of the throughput of the wireless link as a function of the persistency level δ when ARQ-SR alone is used. We want to be sure that the throughput of the wireless link is always increasing with δ , a result that we have seen in the case of 10 TCP connections. We plot in Figure 8 the throughput of the wireless link for different values of δ ranging from 0 to 5. 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

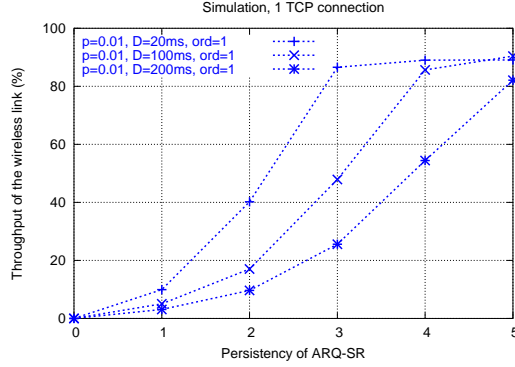


Figure 8: Simulation: Throughput of the wireless link for one TCP connection, ARQ-SR model, $p = 0.01$

three lines that correspond to three distinct values of D , $D = 20, 100, 200$ ms. For the three lines in the figure, the throughput starts by quickly increasing then saturates. As expected, the increase in the throughput is slower than that in Figure 4. The saturation throughput is however the same. No degradation in the throughput caused by an increase in δ is noticed.

Based on our simulation results, we can therefore conclude that under the assumptions of our study, the throughput of the wireless link is always increasing with δ , whatever are the number of TCP connections, the delay and the error rate. This result is very important since it tells us that a large δ is always needed to achieve the best performance for this type of traffic. With large δ , the wireless link is fully utilized, and the amount of FEC must be optimized so as to maximize the throughput.

4 Analytical study

We complete our study by an analytical modeling of TCP performance over a wireless link implementing hybrid FEC/ARQ-SR at the link level for error recovery. Our analysis is based on the model described in Section 2. We focus on the computation of the throughput of the wireless link as a function of our model parameters: C , B , D , p , X , K , N , δ , etc. Then, we solve this model and we compare its performance to our simulation results. The model reports very close results to simulations. The advantage of an analytical model is that it allows to find directly the optimal tuning of FEC/ARQ-SR to be done for a certain network setting, rather than carrying out long simulations. We discuss at the end how our model can assist in wireless link tuning.

4.1 Analytical model

We start by making some assumptions that we will relax as long as we progress in the analysis.

We denote by T the average two-way end-to-end delay of the TCP connections, excluding the delay over the wireless link. Our first assumption is that all connections have the same value of T . This assumption is only made to simplify the exposition of the problem. Connections with different RTTs can be easily introduced.

Second, we consider that the wireless link delivers packets in order to IP (and then to the TCP destinations), and we ignore as a beginning the time introduced by the resequencing of packets at the output of the wireless link. Figure 4 indicates that delivering packets in-order is essential for a good performance of TCP in presence of a link-level ARQ-SR mechanism. Later, we will propose a heuristic to account for the resequencing time in our model.

Our last assumption is about the queuing of packets at the input of the wireless link. We ignore any queuing delay before the full utilization of the wireless link. As a consequence, we ignore any congestion losses that can appear before the full utilization of the wireless link. Since we are working in a best case scenario where the wireless link is the bottleneck, we assume that the only losses seen by the TCP connections before the full utilization of the wireless link are those caused by transmission errors. One can account for the queuing delay (and hence for congestion losses) by coming up with some model for the buffer at the input of the wireless link,

then expressing the queuing delay as a function of the load on the link. Note that our model does not make any assumption about queuing delay after the full utilization of the wireless link.

Our model accounts for transmission times. We said in Section 2 that our model for ARQ-SR assumes entire priority of retransmitted frames over original ones. A frame to be transmitted for the same time may then be delayed by one or more retransmissions. This introduces some delay that adds to the fixed transmission time of the frame N/B .

We give now our general expression for the throughput of the wireless link. This expression is a function of some quantities that we compute in subsequent sections. Let R denote the throughput (i.e. the receiving rate) of a long-lived TCP connection. Different expressions exist in the literature for such expression [5, 16, 17, 19]. We will use the one in [19]. According to [19], the throughput of a TCP connection (in transmission units/s) is equal to

$$R(P, A) = \frac{S}{A \cdot \left(\sqrt{\frac{2bP}{3}} + 4 \min \left(1, 3\sqrt{\frac{3bP}{8}} \right) \cdot P \cdot (1 + 32P^2) \right)} \cdot (1 - P), \quad (1)$$

where A is the average RTT and P the TCP packet loss probability. b is a coefficient usually equal to 2 which models delayed ACKs. The term $(1 - P)$ at the end accounts for packets lost over the wireless link. Without this term we get the sending rate of the TCP connection.

Given our above assumptions, the throughput of the wireless link can be written as (in units/s),

$$U = \min(\alpha \cdot B, C \cdot R). \quad (2)$$

In percents, the throughput is equal to $100 \cdot U/B$. The factor α accounts for the amount of bandwidth wasted on FEC and retransmissions. If the wireless link is not fully utilized, the throughput of the wireless link will be equal to the aggregate throughput of the C TCP connections $C \cdot R$. When the wireless link is fully utilized, the throughput of the wireless link is equal to $\alpha \cdot B$. In this latter case, $C \cdot R$ overestimates the aggregate throughput of TCP connections since in our computation of the average round-trip time A , we will ignore the queuing delay of packets in the router at the input of the wireless link. When the wireless link is fully utilized, packets are queued, A underestimates the average RTT, and R overestimates the throughput of TCP.

4.2 Computation of the packet loss rate

A TCP packet is lost if one of its LL frames or more are lost. Let P_F be the frame loss rate. Under our assumptions, frames are lost independently of each other. Hence,

$$P = 1 - (1 - P_F)^X. \quad (3)$$

We compute now the probability that a frame is lost. This requires that the δ trials fail, in addition to the loss of the original transmission. This gives in total $\delta + 1$ opportunities to correctly transmit a frame. A trial fails when $N - K$ units or more are lost in the frame. This happens with probability

$$P_T = \sum_{i=N-K+1}^N \binom{N}{i} p^i (1-p)^{N-i}. \quad (4)$$

The probability that a frame is lost is then equal to

$$P_F = P_T^{\delta+1}. \quad (5)$$

By substituting (4) in (5), then in (3), we get the probability that a TCP packet is lost P .

4.3 Computation of the maximum throughput α

Consider an LL frame that leaves the wireless link. This frame can correspond to an original transmission, or to a retransmission. If the frame is corrupted, which happens with probability P_T , the amount of data we lose is equal to N units. A frame correctly received can also be discarded. This happens when the packet to which it belongs cannot be reassembled due to the loss of some of its frames. The probability of this last event is equal

to $1 - (1 - P_F)^{X-1}$, and the number of units we lose in this case is also equal to N . On the other hand, when a frame is correctly received and the packet to which it belongs is correctly reassembled, the volume of data we lose is only equal to $N - K$ units. We conclude, that

$$\begin{aligned}\alpha &= P_T \cdot \frac{0}{N} + (1 - P_T)(1 - (1 - P_F)^{X-1}) \cdot \frac{0}{N} + (1 - P_F)^{X-1}(1 - P_T) \frac{K}{N} \\ &= (1 - P_F)^{X-1}(1 - P_T) \frac{K}{N}\end{aligned}\quad (6)$$

4.4 Computation of the average round-trip time

4.4.1 General expression

Suppose that LL frames are quickly acknowledged by the ARQ-SR module at the output of the wireless link and that acknowledgments (TCP, link level) are of negligible size compared to data packets. Thus, the time between the start of the transmission of a frame and the receipt of the acknowledgement telling us about the result of the transmission is

$$\tau = 2D + \frac{N}{B}.$$

Let $\delta_i, i = 1, \dots, X, \delta_i = 0, 1, \dots, \delta$, be the number of times we retransmit frame i of a TCP packet. Denote by σ_i the transmission time of frame i of a TCP packet. The X frames of a TCP packet may not be transmitted back-to-back, since their transmission times may be delayed by retransmitted frames, which in our model are given priority over original frames. Due to this delay, an original frame will see its transmission time increasing by a certain random amount, that depends on how many link-level NACKs are received before the start of its transmission. In the absence of NACKs, the transmission time of a frame is constant and equal to $\sigma_i = N/B$. If n_i NACKs were received, then $\sigma_i = (1 + n_i)N/B$. Using the above notation, the expression of the round-trip time can be written as

$$RTT = T + 2D + \max(\sigma_1 + \delta_1\tau, \dots, \sum_{i=1}^X \sigma_i + \delta_X\tau). \quad (7)$$

The problem is how to compute A , the expectation of RTT . Note that the TCP packet for which we compute RTT is well received by the TCP receiver, otherwise this packet will not be used to measure the round-trip time. This means that the expectation of the round-trip time $A = \mathbb{E}^0 [RTT]$ must be computed under the condition that all the frames of the packet have been well decoded at the output of the wireless link (the superscript 0 denotes this condition). The probability that all frames of a packet are well decoded is equal to $(1 - p_T^{\delta+1})^X$.

RTT is a random variable. The randomness comes from random variables δ_i , which are iid. We have the following, for $k = 0, 1, \dots, \delta$,

$$\mathbb{P}^0 \{\delta_i = k\} = \frac{P_T^k(1 - P_T)}{1 - P_T^{\delta+1}}, \quad \mathbb{P}^0 \{\delta_i \leq k\} = \frac{1 - P_T^{k+1}}{1 - P_T^{\delta+1}}. \quad (8)$$

The randomness also comes from the σ_i . Given this multiplicity of random variables inside the maximum operator in (7), the computation of $A = \mathbb{E}^0 [RTT]$ is quite complex in a general setting. The obtention of a closed-form expression requires some further assumptions. In the following two sections, we give the expression of A in two particular scenarios. For a general scenario, we propose a numerical computation of A as explained in [6].

4.4.2 Case of large bandwidth-delay product wireless networks

In this case, one can neglect the transmission time of frames σ_i compared to the propagation delay D . We write $\mathbb{E}^0 [RTT] = T + 2D + \mathbb{E}^0 [Z]$, with $Z = \max(\delta_1\tau, \dots, \delta_X\tau)$. We also write $\tau = 2D$. Using the fact that $\mathbb{E}^0 [Z] = \int_0^\infty \mathbb{P}^0 \{Z > z\} dz = \int_0^{2D\delta} \mathbb{P}^0 \{Z > z\} dz$, and that $\mathbb{P}^0 \{Z \leq z\} = \prod_{i=1}^X \mathbb{P}^0 \{\delta_i\tau \leq z\}$, we can write

$$\mathbb{E}^0 [Z] = 2D\delta - \int_0^{2D\delta} \prod_{i=1}^X \mathbb{P}^0 \{\delta_i \leq \lfloor z/2D \rfloor\} dz = 2D(\delta + 1) - 2D \sum_{k=0}^{\delta} (\mathbb{P}^0 \{\delta_i \leq k\})^X, \quad (9)$$

where $\mathbb{P}^0 \{\delta_i \leq k\}$ is given in (8) for $0 \leq k \leq \delta$.

4.4.3 Case of medium bandwidth-delay product wireless networks

We define this case as follows: all original frames of a TCP packet are transmitted before we receive any link-level NACK for that packet at the input of the wireless link. In other words, the bandwidth-delay product of the wireless link is larger than one packet size. The bandwidth-delay product does not need to be infinite as in the previous case; the previous case is a particular case of the present one. The gain from this assumption on the bandwidth-delay product is that we are able to write differently the expression of the round-trip time RTT , which allows to find a simpler expression of $A = \mathbb{E}^0 [RTT]$.

Let M be the maximum number of times we retransmit the frames of a TCP packet, $M = \max(\delta_1, \delta_2, \dots, \delta_X)$. We define $Y \in \{1, 2, \dots, X\}$ to be the largest index of the frame that requires the maximum number of retransmissions. Note that more than one frame in a packet can be retransmitted M times. Formally speaking, Y is defined as follows $Y = \max_{i=1 \dots X} (i \mid \delta_i = M)$. It is very easy to see that the expression of RTT given in (7) can be written in this case as

$$RTT = T + 2D + \tau M + \sum_{i=1}^Y \sigma_i.$$

For medium bandwidth-delay product wireless networks, the variables σ_i can be safely assumed to be independent of each other. This does not hold in small bandwidth-delay product networks, since a frame of a packet may be delayed by retransmissions belonging to the same packet. Small bandwidth-delay product wireless networks do not support in general ARQ-SR, but rather simple protocols as Stop-Wait (e.g. IEEE 802.11 WLAN). Given the independence of σ_i , we write

$$A = \mathbb{E}^0 [RTT] = T + 2D + \tau \mathbb{E}^0 [M] + \mathbb{E}^0 [Y] \mathbb{E}^0 [\sigma_i].$$

$\tau \mathbb{E}^0 [M]$ is equal to the $\mathbb{E}^0 [Z]$ given in (9). The expectation of Y is computed in the Appendix. Our analysis gives,

$$\mathbb{E}^0 [Y] = X(\delta + 1 - \mathbb{E}^0 [M]) - \sum_{k=0}^{\delta} \frac{\mathbb{P}^0 \{\delta_i \leq k-1\}}{\mathbb{P}^0 \{\delta_i = k\}} \mathbb{P}^0 \{M = k\}.$$

Thus, to compute $\mathbb{E}^0 [Y]$ we need (8). We also need the expression of $\mathbb{P}^0 \{M = k\}$, $k = 0, \dots, \delta$. We know that

$$\mathbb{P}^0 \{M = k\} = \mathbb{P}^0 \{M \leq k\} - \mathbb{P}^0 \{M \leq k-1\} = (\mathbb{P}^0 \{\delta_i \leq k\})^X - (\mathbb{P}^0 \{\delta_i \leq k-1\})^X.$$

We still need to compute the expectation of σ_i . The transmission of an LL frame for the first time over the wireless link can be delayed by one or more retransmissions. Let π be the probability to find the wireless link busy. π is equal to $\min(1, C.R/(\alpha.B))$. The probability to have a link-level ACK or a link-level NACK arriving just before the transmission of an original frame is given by π . The probability that a NACK arrives is equal to $\pi_N = \pi P_T(1 - \frac{P_T^\delta}{\delta+1})$. We multiply by $P_T(1 - \frac{P_T^\delta}{\delta+1})$ to account for the fact that a frame is lost and that it is not the last trial (the ARQ-SR mechanism is supposed not to retransmit a frame that has been lost $\delta + 1$ times). For a frame to be the last trial, this requires that the first δ trials fail (probability P_T^δ), and that the frame be one of $\delta + 1$ total trials (the ones that fail plus the last one) (probability $1/(\delta + 1)$). This makes the probability that a frame is the last trial equal to $\frac{P_T^\delta}{\delta+1}$.

Suppose that a NACK is received. The transmission of the original frame is then delayed and the lost frame is retransmitted instead. During this retransmission, a second NACK may arrive, and so on. We suppose that this happens with the same probability π_N . Each retransmission delays the transmission of our original frame by N/B . The number of received NACKs is geometrically distributed and its average is equal to $\pi_N/(1 - \pi_N)$. The average delay caused by NACK is then equal to $(N/B)\pi_N/(1 - \pi_N)$. To this average delay, we must add the constant transmission time of the frame N/B , which gives

$$\mathbb{E}^0 [\sigma_i] = \frac{N}{B} \frac{1}{1 - \pi P_T(1 - \frac{P_T^\delta}{\delta+1})}.$$

The difficulty with this expression is that it depends on R via π , R is the TCP throughput that we aim to compute. We have then a system of the type $R = f(R, \dots)$, where f is a function of R and the other parameters of the model as p, δ, X , etc. Given the complexity of the function f , we cannot solve this system for an explicit expression of R (and then U). The solution can be obtained by iteration of the type $R_{n+1} = f(R_n, \dots)$, where R_n is the value of the throughput at the n -th iteration. Another possible solution is to use an approximation of π (e.g. obtained from measurements), in order to get rid of R in the expression of $\mathbb{E}^0 [\sigma_i]$, and then to avoid the iteration.

4.4.4 Correction of our model for the delay introduced by the resequencing module

The ARQ-SR mechanism is known to desequene packets especially when the bandwidth-delay product is large and the loss rate is high. These packets need to be resequenced before leaving the wireless link, otherwise they arrive at the TCP receivers out-of-order and trigger the generation of duplicate ACKs, something harmful to TCP as illustrated by our simulations in Section 3. The resequencing of packets causes however an increase in RTT, which is not considered in our analytical model until now. A TCP packet may be obliged to wait in a buffer at the output of the wireless link until previous packets are well received, or until ARQ-SR decides that these previous packets are definitely lost and cannot be recovered. To account for this resquencing, some term has to be added to the expression of the round-trip time in (7). Note that packets are resequenced at the output of the wireless link on an aggregate basis, not on a flow basis. The technique presented in this section can also account for resequencing on a TCP flow basis.

Let $A_i = \mathbb{E}^0 [RTT_i]$ be the average round-trip time of a TCP packet when the delay introduced by the resequencing module is considered. We are looking for this average, which needs to be substituted in (1). $A = \mathbb{E}^0 [RTT]$ is the average round-trip time already computed and which does not account for the resequencing delay. Let s_n be the necessary time for the n th TCP packet to cross the wireless link, excluding the waiting time in the resequencing buffer, hence $s_n = RTT - T - D$, where RTT is given in (7). Denote by $\lambda = C.R/(S(1-P))$ the arrival rate of TCP packets (of all connections) at the input of the wireless link. We divide by $(1-P)$ to account for packets lost over the wireless link. Define $1/\mu = \mathbb{E}^0 [s_n] = \mathbb{E}^0 [RTT] - T - D$.

We have then a classical resequencing problem, which is well known in queuing theory (see [2] and the references therein). Clients (equivalent of packets) arrive to an infinite server system with rate λ , stay for average time $1/\mu$, then leave the system out-of-order. Another system called the "resequencer" is placed at the output of the first system to resequence the clients. The problem was to compute the average time a client stays in the resequencer before being delivered.

Different works have studied this problem and tried to compute the average resequencing time. The average resequencing time in our case is $A_i - A$. The solution for such resequencing problem is in general difficult. A closed-form expression for the average resquencing time has been obtained in a very strict case [2], which we summarize in the following proposition. This forms our heuristic for the modeling of the impact of the in-order delivery service on the average round-trip time.

Proposition 1 *Assume that TCP packets arrive at the wireless link as a Poisson process of rate λ . Assume that the variables s_n are exponentially distributed. Further assume that the load is not very high. It is stated in [2] that the average round-trip time A_i including the resequencing delay is approximately equal to*

$$A_i = A + \frac{1}{2\mu} \cdot \frac{\lambda/\mu}{1 + \lambda/\mu},$$

In our case, $1/\mu = A - T - D$, and A (computed in the previous sections) denotes the average round-trip time when the resequencing delay is ignored.

We have the problem of identifying λ . The packet arrival rate is a function of the throughput of the TCP connections, which we want to compute. Again, we have a problem of the type $R = f(R, \dots)$, which can be solved by iteration. We can avoid the iteration by making some approximation of λ , for example by using the value obtained from measurements.

4.5 Validation of the model

We compute the throughput of the wireless link given by our model for the different scenarios presented in the section on simulations. For lack of space, we only show the results that correspond to Figures 3 to 8. For each figure plotted using simulation results, we plot its equivalent using our model; this gives Figures 9 to 14. The modeling results on the throughput of the wireless link are obtained by substituting (1) and (6) in (2). As for the average round-trip time A , it is computed under the medium bandwidth-delay product assumption, which holds for all our scenarios. The approximation in Proposition 1 is used to account for the impact of resequencing time at the output of the wireless link. As for π and λ^1 , we set them using the iteration method (by solving the system

¹The first variable models the probability to find the wireless link busy, the second one models the average packet arrival rate at the input of the wireless link.

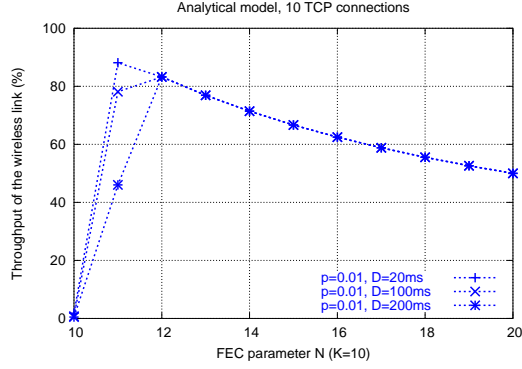


Figure 9: Model: Throughput of the wireless link for 10 TCP connections, FEC alone, $p = 0.01$

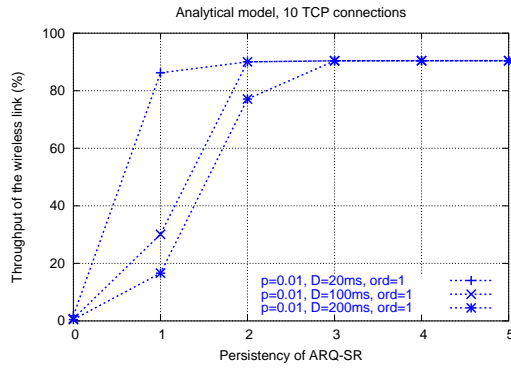


Figure 10: Model: Throughput of the wireless link for 10 TCP connections, ARQ-SR alone, $p = 0.01$

$R = f(R, \dots)$ for R then using it to compute the two parameters).

Figure 10 only shows the lines that correspond to an in-order delivery of packets. The lines that correspond to an out-of-order delivery of packets and that appear in Figure 4 do not appear here since our model only accounts for an in-order delivery of packets. An out-of-order delivery of packets does not increase the round-trip time but instead triggers the generation of duplicate ACKs at the TCP receiver, which in turn can trigger a division of TCP congestion window by two if the TCP sources receive three or more duplicate ACKs.

A comparison between both set of figures clearly show how well the modeling and simulation results match. Note that we are presenting results for challenging scenarios (long delay, high loss rate), where modeling the throughput of the wireless link is difficult because of ARQ-SR and the reordering of packets it introduces. Our model provides better results in other less challenging scenarios. We see however some mismatch when ARQ-SR is enabled ($\delta \geq 1$) and the utilization of the wireless is high, e.g. the line " $p = 0.01, D = 200\text{ms}$ " in Figure 8 and its equivalent in Figure 14. This is mainly caused by an underestimation of the resequencing time at the output of the wireless link. Indeed, the closed-form expression for the average resequencing time stated in Proposition 1 has been established in [2] under a low load assumption. Moreover, when the utilization of the wireless link is high, the packet arrival process at the input of the wireless link deviates from Poisson which constitutes a second source of error that adds to the previous one.

4.6 Optimization of the model

Since the throughput of TCP is an increasing function of δ , it is very easy to conclude that the best performance is achieved when δ is set to a very large value. We take $\delta = \infty$ even though we know that this may not be feasible in practice (at some time we have to stop retransmitting a frame). When $\delta = \infty$, the wireless link is fully utilized

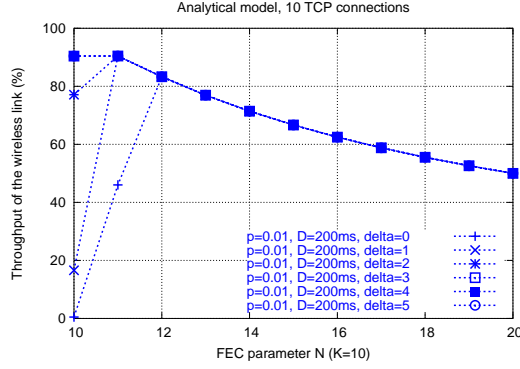


Figure 11: Model: Throughput of the wireless link for 10 TCP connections, hybrid model, $D = 200$ ms, $p = 0.01$

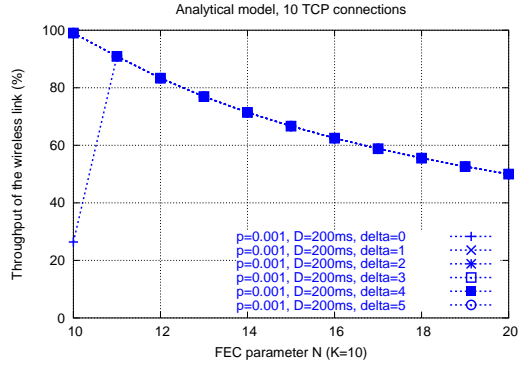


Figure 12: Model: Throughput of the wireless link for 10 TCP connections, hybrid model, $D = 200$ ms, $p = 0.001$

and the throughput of the wireless link in (2) is equal to

$$U = \lim_{\delta \rightarrow \infty} \alpha B = (1 - P_T) \frac{K}{N} B,$$

with P_T given in (4); P_T is the probability that a frame cannot be decoded, which is only a function of the error rate, the frame size, and the amount of FEC. The throughput of a TCP connection R does not figure in this expression of the throughput of the wireless link. We get rid of R since it is found to be an increasing function of δ . The expression of R (and hence our complete model) is however needed if we want to optimize the amount of FEC for small value of δ , since in this case the wireless link will not be always fully utilized and the term $C.R$ in the expression of the throughput in (2) will play a role. The expression of R is also needed if we want to optimize the amount of FEC in the presence of congestion losses in other parts of the network.

The problem is then to maximize the term $(1 - P_T) \frac{K}{N}$ which we denote by α^∞ . When no FEC is used, α^∞ equals $(1 - p)^K$. When we start adding FEC, α^∞ increases until its maximum, then decreases and converges to $\frac{K}{N}$ when N is large (P_T converges to zero). The optimal amount of FEC is the one that maximizes α^∞ . We write

$$N_{optimal} = \arg \max_{N, N \geq K} (\alpha^\infty).$$

For $K = 10$, we plot in Figure 15 the value of α^∞ for different values of p as a function of the amount of FEC. We plot α^∞ in percents. We clearly see that for each value of p , there is an optimal amount of FEC to add. For low to moderate error rates, this optimal amount of FEC is close to zero. For high error rates, it is non-negligible. For example for $p = 0.25$, one needs 6 units of redundancy to achieve the best throughput, which corresponds to a FEC rate equal to 10/16. We also notice in the figure how ARQ-SR alone achieves poor performance at very high error rates since most of the frames are retransmitted. By adding FEC at $p = 0.25$, one can improve the throughput of the wireless link by up to 1100%!

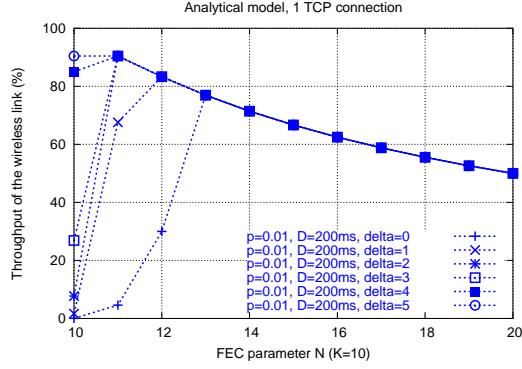


Figure 13: Model: Throughput of the wireless link for one TCP connection, hybrid model, $D = 200$ ms, $p = 0.01$

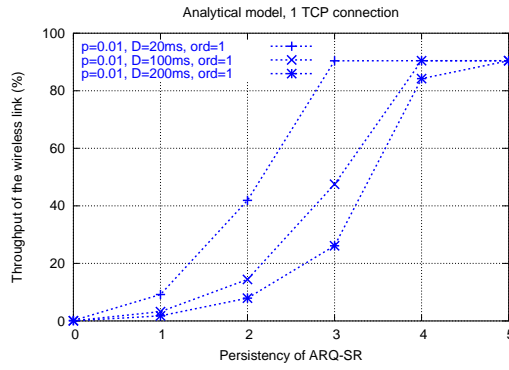


Figure 14: Model: Throughput of the wireless link for one TCP connection, hybrid model, $p = 0.01$

Another conclusion we can make from the expression of α^∞ is that the performance of ARQ-SR alone is a decreasing function of K . Indeed, when K increases, i.e. frames become larger, the value of α^∞ at $N = K$ drops to zero. This is logical since large frames are more sensitive to losses on one hand, and on the other hand retransmitting large frames is not efficient when a small fraction of data is corrupted.

Concerning the optimal amount of FEC, it seems from our analysis that this amount can be assumed to be independent of the frame size. We illustrate this in Figure 16 where we plot α^∞ as a function of the amount of FEC for $K = 50$. If we compare the lines in this figure with their equivalents in Figure 15, we see how the amount of FEC for a certain loss rate is quite independent of the value of K and strongly dependent on the error rate p . We can also notice that increasing K slightly improves the maximum throughput we can achieve with a hybrid FEC/ARQ-SR mechanism. When K is large and for the same amount of FEC, frames resist better to transmission errors, which reduces the number of retransmissions and improves the maximum throughput the wireless link can achieve.

5 Conclusions

The key finding of our analysis is that for long-lived TCP transfers and Bernoulli errors, the use of ARQ-SR is always necessary to achieve good performance, in contrast to FEC which is only needed at high error rates. We show that the throughput of the wireless link increases continuously with ARQ-SR persistency, which indicates that a large persistency level must always be used in presence of such kind of traffic. As a consequence, the maximum throughput we can reach with a hybrid FEC/ARQ-SR is independent of the number of TCP connections. As for the optimal amount of FEC to add, it depends on the loss rate. There is no monotonous improvement of performance with FEC as with ARQ-SR; the performance improves with the first units of FEC then deteriorates. At low and moderate error rates, the optimal amount of FEC is very close to zero, and ARQ-SR alone is enough

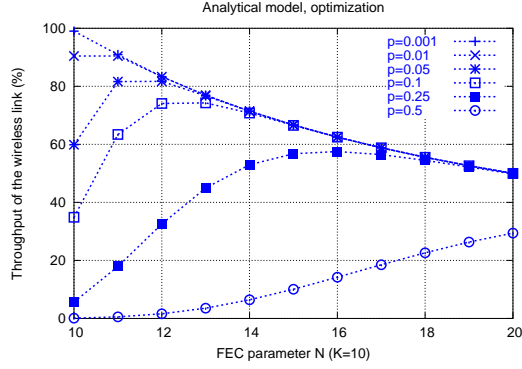


Figure 15: Optimization: Throughput of the wireless link for large δ

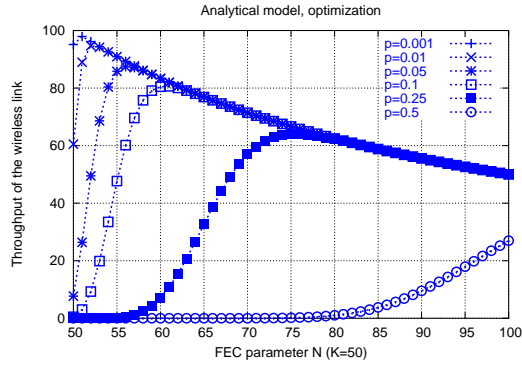


Figure 16: Optimization: Throughput of the wireless link for large δ and $K = 50$

to achieve the best performance. At high error rates, some non-negligible amount of FEC must be added to reduce the number of retransmissions and to improve the throughput.

Another contribution of this work is the proposition of an analytical model for the performance of long-lived TCP traffic over a hybrid FEC/ARQ-SR wireless link. The model accounts for the different parameters of the error recovery mechanism and leads to good results compared to simulations. One main advantage of our model is that it provides an expression for the optimal amount of FEC to use.

Our work can be extended in different directions. We want to check whether our findings hold in the case of finite TCP transfers and bursty transmission errors. Finite TCP transfers are sensitive to retransmissions at the link level since for these transfers, the retransmission timer of TCP does not have enough time to adapt its value to link conditions. This favors FEC over ARQ-SR. On contrary, bursty errors favor ARQ-SR over FEC since lot of FEC is needed to protect frames corrupted by bursts of errors, which is not an efficient solution. Another possible extension of our work is to study what will be the optimal error recovery scheme when delay-sensitive non-TCP applications are used, as voice and video streaming. Extending the analytical model to all these scenarios is also a direction for our future research.

Appendix: Computation of $\mathbb{E}^0 [Y]$

By definition, Y is the largest index of the frame that requires the maximum number of retransmissions. Y is a random variable that takes integer values between 1 to X . We focus here on the computation of its expectation. Let M be the maximum number of retransmissions, $M = \max(\delta_1, \delta_2, \dots, \delta_X)$. Y is defined as

$$Y = \max_{i=1 \dots X} (i \mid \delta_i = M).$$

Note that if there is only one frame retransmitted M times, then Y is uniformly distributed on $[1, X]$, and its average is simply $\mathbb{E}^0 [Y] = (X - 1)/2$. The problem is more difficult when more than one frame are retransmitted M times. We start by computing the distribution of Y . The expectation of Y is known to be equal to

$$\mathbb{E}^0 [Y] = \sum_{y=1}^X \mathbb{P}^0 \{Y \geq y\}. \quad (10)$$

We have,

$$\mathbb{P}^0 \{Y \geq y\} = \mathbb{P}^0 \{\max(\delta_y, \dots, \delta_X) \geq \max(\delta_1, \dots, \delta_{y-1})\}.$$

Define $M_y = \max(\delta_y, \dots, \delta_X)$. By conditioning on the value of M_y (integer between 0 and δ), we can write

$$\begin{aligned} \mathbb{P}^0 \{Y \geq y\} &= \mathbb{P}^0 \{\delta_1 \leq M_y, \delta_2 \leq M_y, \dots, \delta_{y-1} \leq M_y\} \\ &= \sum_{k=0}^{\delta} \mathbb{P}^0 \{M_y = k\} (\mathbb{P}^0 \{\delta_i \leq k\})^{y-1} \\ &= \sum_{k=0}^{\delta} (\mathbb{P}^0 \{M_y \leq k\} - \mathbb{P}^0 \{M_y \leq k-1\}) (\mathbb{P}^0 \{\delta_i \leq k\})^{y-1}. \end{aligned}$$

For $k = 0$, the probability $\mathbb{P}^0 \{M_y \leq k-1\}$ is equal to 0. The expression of $\mathbb{P}^0 \{\delta_i \leq k\}$ is given in (8). We also know that

$$\mathbb{P}^0 \{M_y \leq k\} = \mathbb{P}^0 \{\delta_y \leq k, \dots, \delta_X \leq k\} = (\mathbb{P}^0 \{\delta_i \leq k\})^{X-y+1}.$$

It follows that,

$$\mathbb{P}^0 \{Y \geq y\} = \sum_{k=0}^{\delta} (\mathbb{P}^0 \{\delta_i \leq k\})^X - \sum_{k=0}^{\delta} (\mathbb{P}^0 \{\delta_i \leq k-1\})^{X-y+1} (\mathbb{P}^0 \{\delta_i \leq k\})^{y-1}.$$

To simplify the analysis, we denote $\mathbb{P}^0 \{\delta_i \leq k\}$ by F_k . F_k is equal to 0 for $k < 0$, to 1 for $k \geq \delta$, and is given in (8) for k between 0 and δ . We obtain,

$$\mathbb{P}^0 \{Y \geq y\} = \sum_{k=0}^{\delta} F_k^X - \sum_{k=1}^{\delta} F_{k-1}^{X-y+1} F_k^{y-1}.$$

We substitute this expression in (10), which yields

$$\begin{aligned} \mathbb{E}^0 [Y] &= X \sum_{k=0}^{\delta} F_k^X - \sum_{y=1}^X \sum_{k=1}^{\delta} F_{k-1}^{X-y+1} F_k^{y-1} \\ &= X \sum_{k=0}^{\delta} \mathbb{P} \{M \leq k\} - \sum_{k=1}^{\delta} \sum_{y=1}^X F_{k-1}^{X-y+1} F_k^{y-1} \\ &= X(\delta + 1 - \mathbb{E}^0 [M]) - \sum_{k=1}^{\delta} F_{k-1}^X \sum_{y=1}^X \left(\frac{F_k}{F_{k-1}} \right)^{y-1} \\ &= X(\delta + 1 - \mathbb{E}^0 [M]) - \sum_{k=1}^{\delta} F_{k-1}^X \frac{F_k^X - F_{k-1}^X}{F_k - F_{k-1}} \\ &= X(\delta + 1 - \mathbb{E}^0 [M]) - \sum_{k=0}^{\delta} \frac{\mathbb{P}^0 \{\delta_i \leq k-1\}}{\mathbb{P}^0 \{\delta_i = k\}} \mathbb{P}^0 \{M = k\}. \end{aligned}$$

References

- [1] M. Allman, D. Glover, L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", *RFC 2488*, Jan. 1999.
- [2] F. Baccelli, E. Gelenbe, B. Plateau, "An end-to-end approach to the resequencing problem", *Journal of the ACM*, vol. 31, no. 3, pp. 474-485, July 1984.
- [3] A. Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts", *International Conference on Distributed Computing Systems (ICDCS)*, May 1995.
- [4] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. Katz, "A comparison of Mechanisms for Improving TCP Performance over Wireless Links", *ACM SIGCOMM*, Aug. 1996.
- [5] C. Barakat, "TCP modeling and validation", *IEEE Network*, vol. 15, no. 3, pp. 38-47, May 2001.
- [6] C. Barakat, A. Al Fawal, "Analysis of link-level hybrid FEC/ARQ-SR for wireless links and long-lived TCP traffic", INRIA Research Report RR-4752, February 2003.
- [7] C. Barakat, E. Altman, "Bandwidth tradeoff between TCP and link-level FEC", *Computer Networks*, vol. 39, no. 2, pp. 133-150, Jun. 2002.
- [8] C. Barakat, E. Altman, W. Dabbous, "On TCP Performance in a Heterogeneous Network : A Survey", *IEEE Communications Magazine*, vol. 38, no. 1, pp. 40-46, Jan. 2000.
- [9] S. Biaz, N. H. Vaidya, "Distinguishing Congestion Losses from Wireless Transmission Losses: A Negative Result", *International Conference on Computer Communications and Networks (IC3N)*, Oct. 1998.
- [10] CAIDA, the Cooperative Association for Internet Data Analysis, available at: <http://www.caida.org>
- [11] H. Chaskar, T. V. Lakshman, U. Madhow, "On the design of interfaces for TCP/IP over wireless", *IEEE MILCOM*, Oct. 1996.
- [12] A. Chockalingam, M. Zorzi, R.R. Rao, "Performance of TCP on Wireless Fading Links with Memory", *IEEE ICC*, Jun. 1998.
- [13] T. Henderson, R.H. Katz, "Transport Protocols for Internet-Compatible Satellite Networks", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 2, pp. 326-344, Feb. 1999.
- [14] V. Jacobson, "Congestion avoidance and control", *ACM SIGCOMM*, Aug. 1988.
- [15] 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 Engg. Sciences*, Feb. 1998.
- [16] 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*, vol. 5, no. 3, pp. 336-350, Jun. 1997.
- [17] M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", *Computer Communication Review*, vol. 27, no. 3, pp. 67-82, Jul. 1997.
- [18] The LBNL Network Simulator, *ns-2*, <http://www.isi.edu/nsnam/ns/>
- [19] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", *ACM SIGCOMM*, Sep. 1998.
- [20] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, Jan. 1997.