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

Chadi Barakat, Alaeddine Al Fawal

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

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

Chadi Barakat — Alaeddine Al Fawal

N° 4752

February 2003

THÈME 1

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--4752--FR+ENG

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

Chadi Barakat* , Alaeddine Al Fawal†

Thème 1 — Réseaux et systèmes
Projet PLANETE

Rapport de recherche n° 4752 — February 2003 — 31 pages

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 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).

Key-words: TCP, wireless links, ARQ-SR, FEC, simulation, modeling.

* Planete research group, INRIA - Sophia Antipolis, 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.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Analyse d'un mécanisme de correction d'erreur FEC/ARQ-SR pour liens sans fil en présence de connexions TCP de longues durées

Résumé : Le protocole TCP est connu par le fait que sa performance dégrade sur un lien sans fil à cause des erreurs de transmission. La raison principale de cette dégradation est que TCP considère les pertes de paquets comme étant des signaux de congestion. Plusieurs solutions ont été proposées, la plus prometteuse est celle qui améliore la qualité du lien sans fil en utilisant de la redondance (FEC), des retransmissions sélectives (ARQ-SR), et une livraison de paquets à la sortie du lien sans fil dans leur ordre d'arrivée. Le problème de la FEC est qu'elle consomme de la bande passante. Le problème des retransmissions sélectives est qu'elles augmentent les délais aller-retour des paquets TCP. Ces mécanismes pourraient détériorer la performance de TCP au lieu de l'améliorer si leurs paramètres ne sont pas bien choisis. Nous étudions dans ce travail les performances de TCP sur un lien sans fil en présence d'un modèle hybride FEC/ARQ-SR. La question que nous nous posons est comment choisir les paramètres du mécanisme de correction d'erreur (quantité de FEC, nombre de retransmissions) afin de maximiser la performance de TCP. L'étude est faite avec des simulations et de la modélisation analytique. Elle est menée pour différentes caractéristiques physiques du lien sans fil (délai, taux de pertes) et pour différents nombres de connexions TCP.

Mots-clés : TCP, liens sans fil, ARQ-SR, FEC, simulation, modélisation.

1 Introduction

For TCP, the loss of a packet is an indication that the network is congested [12, 18]. 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, 6, 8]. 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, 8, 9]. 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 [8], etc.). Other solutions, e.g. I-TCP [3], 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 [11], 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, 6]. The drawback of FEC is that it consumes some extra bandwidth to transmit the redundant information. It has been shown in [6] 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 [15, 17]. 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 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 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. But, this combination also 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 and modeling long-lived TCP transfers over wireless links showing Bernoulli non-congestion losses. The simulations are done with the ns-2 simulator [16]. The modeling is done with an analytical approach close to that used in [6]. 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. Our main findings can be summarized as follows: 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 inde-

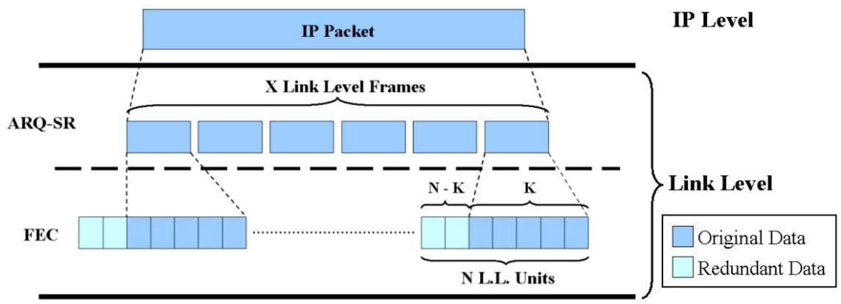

Figure 1: The Hybrid Model FEC / ARQ-SR

pendent of the number of TCP connections. For a certain ARQ-SR persistency level, 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.

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. The appendixes at the end of the paper provide details on the analysis behind some analytical expressions presented in the paper.

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. 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 [10]. 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 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, called the persistency of ARQ-SR and denoted by δ, $\delta = 0, 1, 2, \ldots$. $\delta = 0$ 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.

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 sizes. Usually, errors over wireless links are modeled by a Markov chain of two or more states since they appear in bursts [6, 9, 10, 13]. We choose the Bernoulli model for the simplicity of the analysis it allows. The Bernoulli model is known to hold over fast fading channels (high speed mobile users) [13] 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.

As for the traffic, we consider that 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. We are aware of the fact that this model for Internet traffic is not very realistic, but we believe at the same time that the study of long-lived TCP traffic is an important step to be crossed before studying other models for Internet traffic that match better the reality e.g. models for HTTP traffic. Note that long-lived TCP connections are known to carry an important of Internet traffic and that this part is expected to increase with the advent of new data-greedy applications as P2P and Grids.

3 Simulation study

We implement our model for link-level FEC/ARQ-SR with in-order delivery into ns-2, the network simulator developed at LBNL [16]. 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 utilization of the wireless link when one TCP connection uses 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 and to obtain convergence of utilization measurements.

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. 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 [6] 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 [6, 15, 17, 18].

3.1 First part: link utilization for 10 connections

3.1.1 Simulation scenarios

In this section, we present how the utilization of the wireless link varies in simulations according to the various parameters of FEC and ARQ-SR. We considered several scenarios that are derived from the network topology shown in Figure 2.

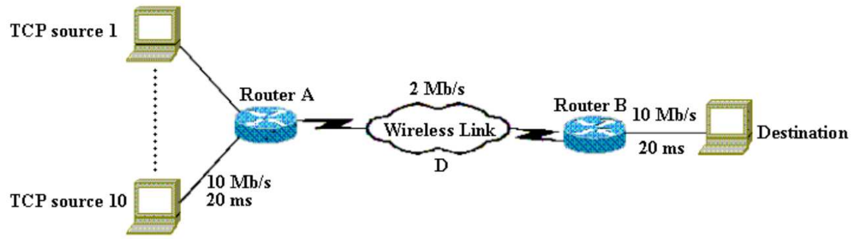

Figure 2: Simulated network topology

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 and enable the Delay ACK functionality [18]. 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}.

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.

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 is high, since the use of ARQ-SR in this extreme case results in a considerable increase in the round-trip time, which

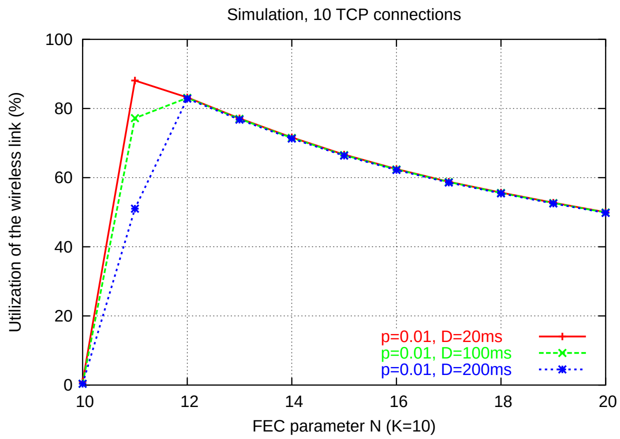

Figure 3: Simulation: Utilization of the Wireless Link for 10 TCP Connections, FEC Alone, $p = 0.01$

is in favor of FEC. In the other scenarios (smaller 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 [6]. For all values of N on the right-hand side of the optimal point, the amount of FEC is more than necessary.

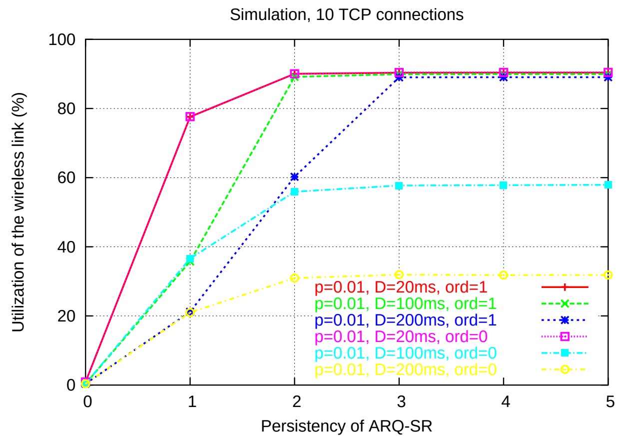

Figure 4: Simulation: Utilization of the Wireless Link for 10 TCP Connections, ARQ-SR Alone, $p = 0.01$

One should expect that the decay of the utilization on the right-hand side of the optimal point is given by $K \times B/N$.

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 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. 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 notice in these results 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 con-

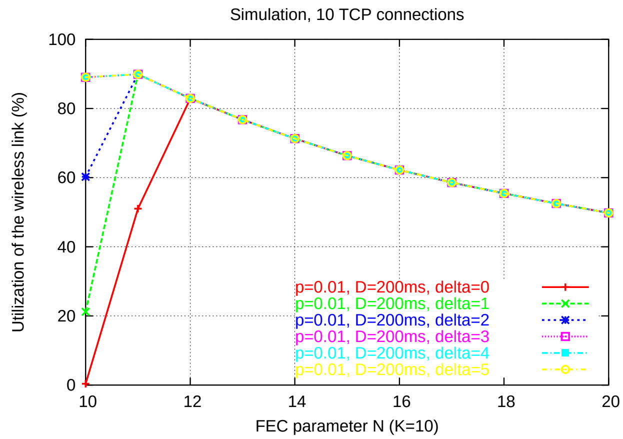

Figure 5: Simulation: Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model, $D = 200$ ms, $p = 0.01$

gestion 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 D, and it is even more pronounced in favor of ARQ-SR.

3.1.4 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 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 $(p, D) = (0.01\ , 200\ \text{ms})$. If we look at the values of the utilization 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 the assumptions of our analysis (long-lived TCP transfers, Bernoulli errors). We remark that the best performance a hybrid FEC/ARQ-SR

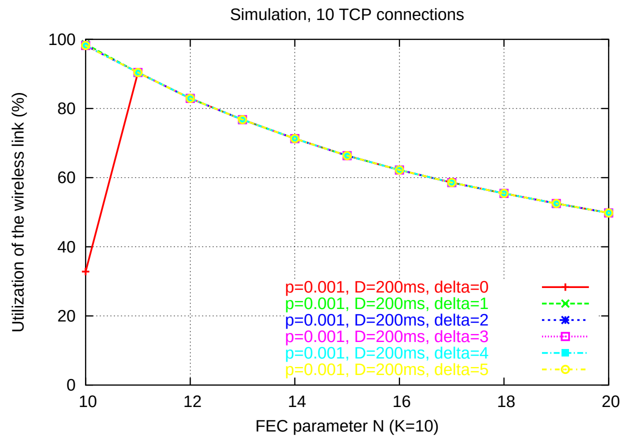

Figure 6: Simulation: Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model, $D = 200$ ms, $p = 0.001$

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.

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 in almost all scenarios 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.

3.2 Second part: link utilization for one 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 the TCP connection is active alone 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 case of one connection than in the case of 10 connections to achieve full link utilization.

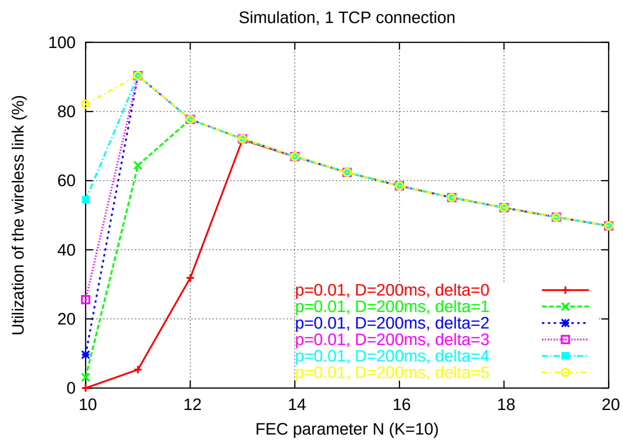

Figure 7: Simulation: Utilization of the Wireless Link for One TCP Connection, Hybrid Model, $D = 200$ ms, $p = 0.01$

We consider the same network topology as that 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 utilization of the wireless link and the x-axis the amount of FEC modeled by "N". 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 close 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 utilization 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 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 in Figure 8 the utilization 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 three lines that

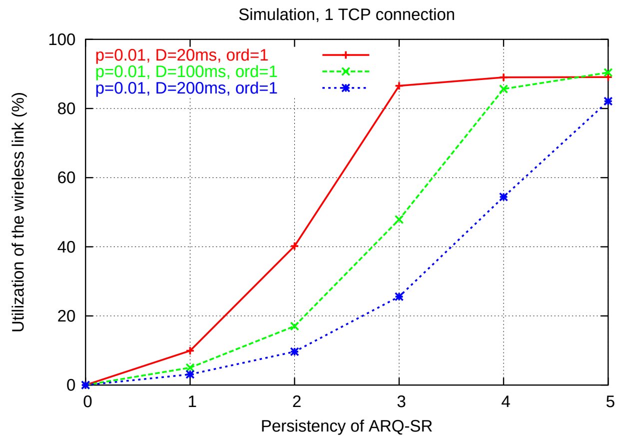

Figure 8: Simulation: Utilization of the Wireless Link for One TCP Connection, Hybrid Model, $p = 0.01$

correspond to three distinct values of D; $D = 20$, 100, 200 ms. For the three lines in the figure, the utilization starts by quickly increasing then saturates. No degradation in the utilization caused by an increase in δ is noticed. Based on our simulations results, we can 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.

4 Analytical study

We complete our study by an analytical modeling of TCP performance over a wireless link disposing of a link-level hybrid FEC/ARQ-SR error recovery mechanism. Our analysis is based on the model described in Section 2. We focus on the computation of the utilization 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.

4.1 Analytical model

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

We denote by T the average two-way end-to-end delay of the TCP connections, excluding the propagation 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 destination), 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 two heuristics 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, say for example a M/M/1/K queuing model, then expressing the queuing delay as a function of the load on the link. Our model does not make any assumption about queuing delay after the full utilization of the wireless link. We account however 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. First, we neglect this additional delay caused by retransmissions. Then, we introduce it into our model in Section 4.3.

We give now our general expression for the utilization of the wireless link. This expression is a function of some quantities that will be computed in the next section. 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, 14, 15, 17]. We will use the one in [17]. According to [17], the throughput of a TCP connection (in transmission units/s) is equal to

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

where A is the average RTT and P the TCP packet loss probability. b is a coefficient usually equal to 2. The term $(1-P)$ at the end accounts for packets lost over the wireless link. Given our above assumptions, the utilization of the wireless link can be written as,

$$U = \frac{\min(\alpha.B, C.R)}{B}. \qquad (2)$$

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 $C.R$. When the wireless link is fully utilized, the throughput of the wireless link will be equal to $\alpha.B$. In this latter case, $C.R$ overestimates the aggregate throughput of TCP connections since in our computation of A, we will ignore the queuing delay of packets at the input of wireless links. When the wireless links is fully utilized, packets are queued, A underestimates the average RTT, and R overestimates the throughput of TCP.

4.2 Analysis

The utilization U is a function of three quantities A, the average RTT, P, the packet loss rate, and α, the coefficient that accounts for the bandwidth consumed by FEC and retransmissions. We compute these quantities in this section. Once these quantities are computed, one can plug them into (2) to obtain the utilization.

4.2.1 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,

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

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 the frame. A trial fails when $N - K$ units or more are lost in the frame. This happens with probability

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

The probability that a frame is lost is then equal to

$$P_F = P_T^{\delta+1}. \tag{5}$$

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

4.2.2 Computation of α

Consider a 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. 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

$$\alpha = (1-P_F)^{X-1}(1-P_T)\frac{K}{N}. \tag{6}$$

4.2.3 Computation of the average round-trip time

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 δ_i, $i = 1, \ldots, X$, $\delta_i = 0, 1, \ldots, \delta$, be the number of times we retransmit frame i of a TCP packet. Then, the round-trip time of a TCP packet can be written as follows

$$RTT = T + 2D + \max(\frac{N}{B} + \delta_1\tau, \ldots, \frac{XN}{B} + \delta_X\tau). \tag{7}$$

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 $k = 0, 1, \ldots, \delta$,

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

We explain now how one can compute the expectation of RTT. The result is complex. Later we will present another technique that allows a simpler expression of A. The problem of the latter technique is that it does not work in all scenarios. Let $Z = \max(\frac{N}{B} + \delta_1\tau, \ldots, \frac{XN}{B} + \delta_X\tau)$. We have $A = T + 2D + \mathbb{E}^0[Z]$. The expectation of Z can be written as follows,

$$\mathbb{E}^0[Z] = \int_0^{\frac{XN}{B}+\tau\delta} \mathbb{P}^0\{Z > z\}\,\mathrm{d}z.$$

We only integrate until $\frac{XN}{B} + \tau\delta$ since this is the maximum value that Z can take. We also know that

$$\mathbb{P}^0\{Z > z\} = 1 - \mathbb{P}^0\{Z \le z\} = 1 - \prod_{i=1}^{X} \mathbb{P}^0\left\{\frac{iN}{B} + \delta_i\tau \le z\right\}.$$

It follows that,

$$\mathbb{E}^0[Z] = \frac{XN}{B} + \tau\delta - \int_0^{\frac{XN}{B}+\tau\delta} \prod_{i=1}^{X} \mathbb{P}^0\left\{\delta_i \le \lfloor (z - \frac{iN}{B})/\tau \rfloor\right\} \mathrm{d}z.$$

This expression can be solved numerically by using (8) and the expression of P_T in (4). In Section 4.3 we explain how to account for the delay introduced by retransmissions. In Sections 4.4 and 4.5, we give simpler expressions of A in two particular scenarios.

4.3 Correction of our model for the delay introduced by retransmissions

The X frames of a TCP packet may not be transmitted back-to-back, since their transmission times may be delayed by retransmitted 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 N/B.

Denote by σ_i the transmission time of frame i of a TCP packet. σ_i is equal to N/B plus some variable delay caused by retransmissions. Suppose that n_i NACKs were received, then $\sigma_i = (1 + n_i)N/B$. The expression of the round-trip time given in (7) will change to

$$RTT = T + 2D + \max(\sigma_1 + \delta_1\tau, \dots, \sum_{i=1}^{X} \sigma_i + \delta_X\tau). \tag{9}$$

This is the general expression of RTT that we will use throughout the rest of the paper. The expectation of this expression can be computed in a general scenario using the same technique as in Section 4.2.3. In the following two sections, we give simpler expressions of $A = \mathbb{E}^0\left[RTT\right]$ in two particular scenarios.

4.4 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 $\mathbb{E}^0\left[RTT\right] = T+2D+\mathbb{E}^0\left[Z\right]$, with $Z = \max(\delta_1\tau, \dots, \delta_X\tau)$. We also write $\tau = 2D$. Using the same technique as in Section 4.2.3, we get

$$\mathbb{E}^0\left[Z\right] = 2D\delta - \int_0^{2D\delta} \prod_{i=1}^{X} \mathbb{P}^0\left\{\delta_i \leq \lfloor z/2D \rfloor\right\} \mathrm{d}z = 2D(\delta+1) - 2D\sum_{k=0}^{\delta} \left(\mathbb{P}^0\left\{\delta_i \leq k\right\}\right)^X, \tag{10}$$

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

4.5 Case of medium bandwidth-delay product wireless networks

This case is more general than the previous one since the transmission time of frames is no longer negligible. We define this case as follows: all the 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 for $A = \mathbb{E}^0[RTT]$.

Let M be the maximum number of times we retransmit the frames of a TCP packet, $M = \max(\delta_1, \delta_2, \ldots, \delta_X)$. We define $Y \in \{1, 2, \ldots, X\}$ 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 $Y = \max_{i=1 \ldots X}(i \mid \delta_i = M)$. It is very easy to see that the expression of RTT given in (9) can be written in this case as

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

In this case of medium bandwidth-delay product, the variables σ_i can be safely assumed to be independent of each other. This does not hold in small bandwidth-delay product wireless 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. 802.11 WLAN). Given this 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 (10). The expectation of Y is computed in Appendix A. We find,

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

To compute $\mathbb{E}^0[Y]$, we need (8). We also need the expression of $\mathbb{P}^0\{M = k\}$, $k = 0, \ldots, \delta$. We have,

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

We still need to compute the expectation of σ_i. The transmission of a 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). 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

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

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 $R = f(R, \ldots)$, 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 $R_{n+1} = f(R_n, \ldots)$, 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 $\mathbb{E}^0\left[\sigma_i\right]$, and then to avoid the iteration.

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

The ARQ-SR mechanism is known to desequence 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 will arrive at the TCP receivers out-of-order and will trigger the generation of duplicate ACKs, something harmful for 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 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 (9). Note that packets are resequenced at the output of the wireless link on a 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 plugged 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 nth TCP packet to cross the wireless link excluding the waiting time in the resequencing buffer, hence $s_n = RTT - T - D$. Denote by $\lambda = C.R/(S(1-P))$ the arrival rate of TCP packets (of all connections) at the input of 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). 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.

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. The average round-trip time A_i including the resequencing delay is approximately equal to*

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

with $1/\mu = A - T - D$, and A (computed in the previous sections) 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 $R = f(R, \ldots)$, which can be solved by iteration. We can avoid the iteration by making some approximation of λ, for example by considering that the wireless link is fully utilized, and taking $\lambda(1-P) = \alpha B/S$. We multiply λ by $(1-P)$ to account for packets that are lost over the wireless link and that cannot be recovered. Another possible approximation is to set λ to the value obtained from measurements.

One can suggest to take the packet arrival process as deterministic instead of Poisson, and find an expression for the average resequencing time. The variables s_n

are always assumed to be exponentially distributed. To our knowledge, no explicit expression exits for such a problem. By simple computation (see Appendix B for details), one can show that the average round-trip time can be written as

$$A_i = T + D + \int_0^\infty 1 - \prod_{i=0}^{\infty}(1 - e^{-\mu(x+i/\lambda)})\mathrm{d}x, \tag{11}$$

with $1/\mu = A - T - D$. This expression can be solved numerically.

4.7 Validation of the model

We compute the utilization of the wireless link given by our model for the different scenarios presented above. 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 about the utilization of the wireless link are obtained by plugging (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 to the values obtained from simulations.

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 extreme scenarios (long delay, high loss rate) where modeling the utilization of the wireless link is the most challenging due mainly to ARQ-SR and the reordering of packets it introduces. Our model provides even 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 "$p = 0.01$, $D = 200$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

[1] The 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.

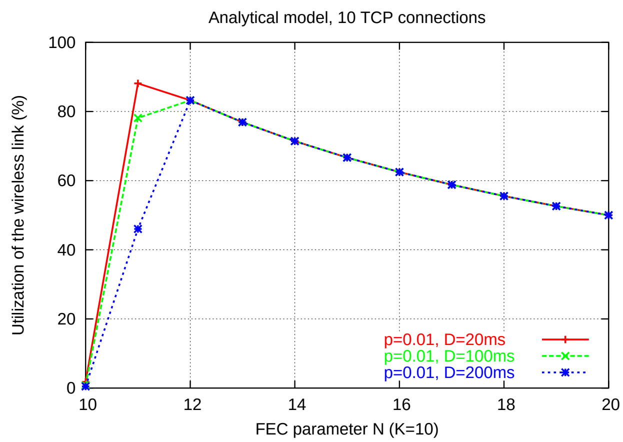

Figure 9: Model: Utilization of the Wireless Link for 10 TCP Connections, FEC Alone, $p = 0.01$

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.

5 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.

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.

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.

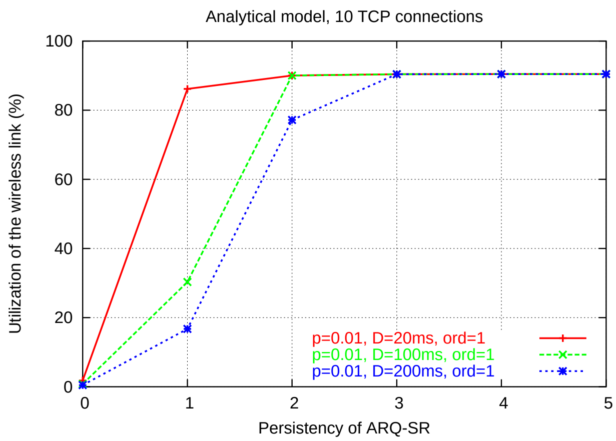

Figure 10: Model: Utilization of the Wireless Link for 10 TCP Connections, ARQ-SR Alone, $p = 0.01$

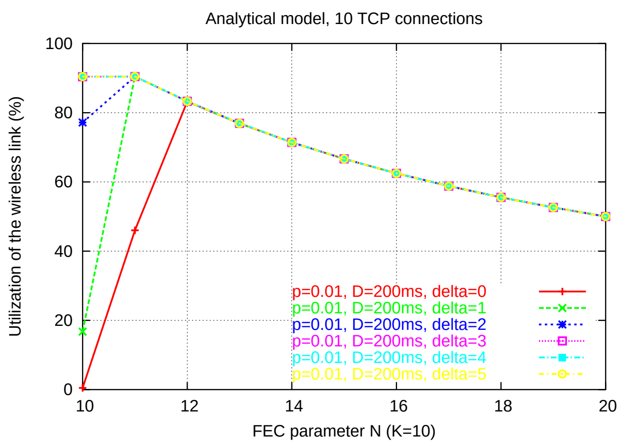

Figure 11: Model: Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model, $D = 200$ ms, $p = 0.01$

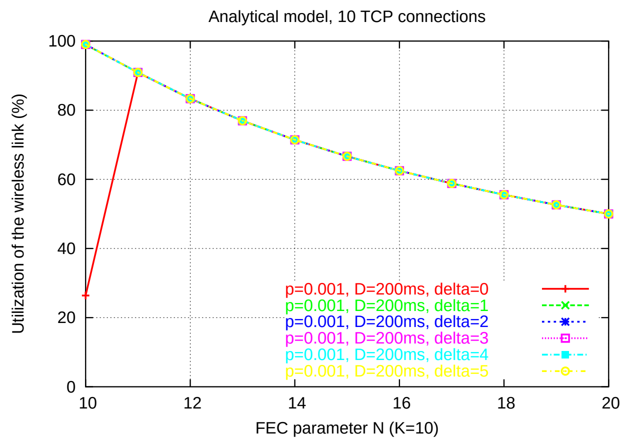

Figure 12: Model: Utilization of the Wireless Link for 10 TCP Connections, Hybrid Model, $D = 200$ ms, $p = 0.001$

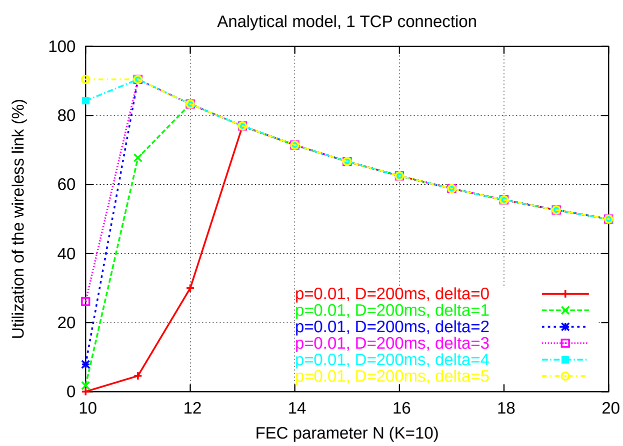

Figure 13: Model: Utilization of the Wireless Link for One TCP Connection, Hybrid Model, $D = 200$ ms, $p = 0.01$

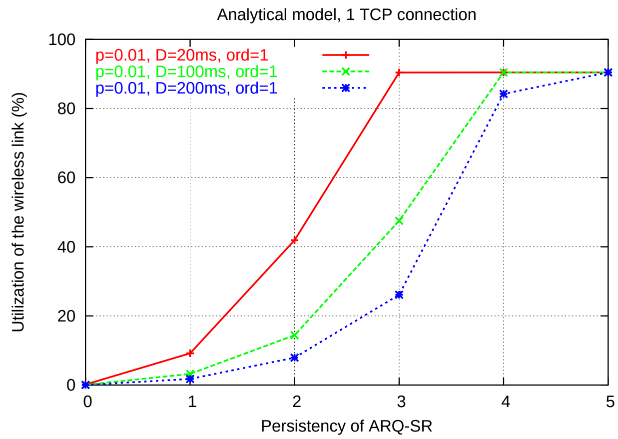

Figure 14: Model: Utilization of the Wireless Link for One TCP Connection, Hybrid Model, $p = 0.01$

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. Extending the analytical model to these scenarios is also a direction for a future research.

Appendix A: 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, $M = \max(\delta_1, \delta_2, \ldots, \delta_X)$. Y is defined as

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

We 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

$$\mathbb{E}^0[Y] = \sum_{y=1}^{X} \mathbb{P}^0\{Y \geq y\}. \tag{12}$$

We have,

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

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

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

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

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

It follows that,

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

To simplify the analysis, we denote $\mathbb{P}^0\{\delta_i \leq k\}$ by F_k. F_k is equal to 0 for $k < 0$, and to 1 for $k \geq \delta$ and is given in (8). 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 plug this expression into (12), which yields

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

Appendix B: Computation of the average resequencing time in case of a deterministic packet arrival process

Suppose that TCP packets arrive at regular instants separated by $1/\lambda$. Let s_n be the time packet n takes to cross the wireless link excluding the waiting time in the resequencing buffer. Let a_n be the arrival time of packet n at the wireless link. The delivery time of packet n, d_n, is then,

$$d_n = \max(a_n + s_n, a_{n-1} + s_{n-1}, \ldots).$$

The round-trip time of packet n is then,

$$RTT_i = T + D + \max(s_n, s_{n-1} - 1/\lambda, s_{n-2} - 2/\lambda, \ldots).$$

Let $Z = \max(s_n, s_{n-1} - 1/\lambda, s_{n-2} - 2/\lambda, \ldots)$. Then, $A_i = \mathbb{E}^0[RTT_i] = T + D + \mathbb{E}^0[Z]$. We focus on the computation of $\mathbb{E}^0[Z]$. We have

$$\mathbb{E}^0[Z] = \int_0^\infty \mathbb{P}^0\{Z > x\}\,\mathrm{d}x,$$

and

$$\mathbb{P}^0\{Z > x\} = 1 - \mathbb{P}^0\{Z \le x\} = 1 - \prod_{i=0}^{\infty} \mathbb{P}^0\{s_n \le x + i/\lambda\}.$$

This holds since all the s_n are iid. If we assume furthermore that the s_n are exponentially distributed with average $1/\mu$, we can write

$$\mathbb{E}^0\left[Z\right] = \int_0^\infty \mathbb{P}^0\left\{Z > x\right\}\mathrm{d}x = \int_0^\infty 1 - \prod_{i=0}^{\infty}(1 - e^{-\mu(x+i/\lambda)})\mathrm{d}x.$$

We combine this expression with that of A_i to get (11).

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 Mechanisims 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, E. Altman, "Bandwidth tradeoff between TCP and link-level FEC", *Computer Networks*, vol. 39, no. 2, pp. 133-150, Jun. 2002.

[7] 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.

[8] 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.

[9] H. Chaskar, T. V. Lakshman, U. Madhow, "On the design of interfaces for TCP/IP over wireless", *IEEE MILCOM*, Oct. 1996.

[10] A. Chockalingam, M. Zorzi, R.R. Rao, "Performance of TCP on Wireless Fading Links with Memory", *IEEE ICC*, Jun. 1998.

[11] 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.

[12] V. Jacobson, "Congestion avoidance and control", *ACM SIGCOMM*, Aug. 1988.

[13] 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.

[14] 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.

[15] 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.

[16] The LBNL Network Simulator, *ns-2*, http://www.isi.edu/nsnam/ns/

[17] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", ACM SIGCOMM, Sep. 1998.

[18] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, Jan. 1997.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot-St-Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399