End-to-end Reliability in UMTS : TCP over ARQ

A-F. Canton and T. Chahed
Institut National des Telecommunications
9 rue C. Fourier - 91011 EVRY CEDEX - France
Phone +33 1 60 76 46 93 - Fax +33 1 60 76 42 91

{anne-florence.allouche,tijani.chahed}@int-evry.fr

Abstract

The focus of this paper is end-to-end reliable transmission in UMTS environment where TCP, a reliable tranport protocol designed to retransmit information in case of loss, is present at the mobile station as well as the wired portion of the network. The mobile and wireless portion of the network, characterized by significant error figures, relies on ARQ to detect and retransmit corrupted frames. In this work, we propose an analytical model for the operation of TCP over ARQ. We investigate several error patterns corresponding to random, slow and fast fading channels. Our results quantify the enhanced performance of TCP over ARQ in terms of throughput as a function of both loss and error.

Keywords

end-to-end reliability, TCP over ARQ, UMTS

I. INTRODUCTION

TCP is undeniably the transport protocol most commonly used to carry a significant amount of elastic Internet traffic which needs reliability. As such, TCP has been designed, optimized and tuned, in wired networks setting, to react to packet loss, the sole indication and consequence of congestion. As far as wireless and mobile networks are concerned, this assumption does not hold anymore; in those media, error is significant and is more responsible for service degradation than loss. ARQ, a link-level error detection and correction mechanism, is recommended to enforce reliability in forthcoming UMTS standards. For the purpose of end-to-end reliability where TCP is present in mobile and wireless portions of the network, both TCP and ARQ, with quite similar retransmission mechanisms, the former against loss and the latter against error, may indeed enter into conflict and cause service degradation mainly in terms of throughput. This is why, the objective of this work is to analytically quantify the TCP over ARQ protocol stack or an end-to-end reliability of elastic Internet flows.

Several works on the analytical characterization of TCP have recently been carried out [1], [2], [3], and focused mainly on TCP throughput as a function of loss rate in wired networks. In particular, in [1], a simple model for the analytical characterization of the steady-state throughput of a bulk TCP transfer is proposed. TCP retransmission mechanisms are either due to a time-out or triple-duplicate ACKs, which, in turn, are caused by packet loss in a wired environment. Adding a wireless, mobile portion to this setting with its significant error figures, will inevitably trigger TCP retransmission mechanisms against error, and not only loss in this case. Error in this wireless portion follows some generic patterns and is handeled by ARQ. In [5] for instance, a one-step, two-state Markov chain is used to model error on a fading channel. This takes into account the correlation that may exist between different instances of error. The system evaluation considers only the ARQ mechanism, in terms of throughput, and does not include any interactions with higher, nor lower layers. The work in [6] simulates the operation of TCP in the presence of correlated errors; no mention however is made of ARQ. ARQ and its relationship to TCP is investigated in [8] and [7]; again these studies rely on simulations only.

In this paper, we propose an analytical model for the steady-state throughput of a bulk TCP transfer, in a wireless mobile setting where TCP is present at the mobile station and the wired portion of the network and ARQ is active at the wireless, mobile portion of the end-to-end path. Our mathematical model is based on the one developed in [1]; it however takes into account the interference between ARQ at the radio link level and TCP at the higher as well as wired level. In this context, we investigate several scenarios. First, we analyze the end-to-end behavior of TCP in the mobile and wired portions of the network in the absence of ARQ. Next, we consider the TCP over ARQ performance. Herein, we study the different error patterns at the air channel; those patterns can be either independent or lightly or heavily correlated. Our results show that the end-to-end performance of TCP in a mobile, wireless setting is enhanced by the presence of ARQ as the latter takes in charge all the error detection and correction part of the work which permits TCP to ultimately act transparently on an error-free medium. The benefit of ARQ on end-to-end performance is even more significant in the case of heavily correlated errors as is the slow fading channel case.

The remainder of this paper is organised as follows. In Section II, we describe our end-to-end modeling for TCP over ARQ, in a wired network with a wireless, mobile portion. In Section III, we focus on the error schemes present in the air channel for the cases of independent as well as correlated bit errors. In Section IV, we derive an analytical characterisation for TCP throughput as a function of error and loss taking into account the mobile, wireless portion of the network, assuming that ARQ operation does not disturb TCP. In Section V, we study the interaction between TCP and ARQ, calculate the probability that ARQ enters into conflict with TCP for the cases of fast as well as slow fading channels. Numerical applications and discussion of the results are given in Section VI. Section VII eventually concludes the paper.

II. A MODEL FOR TCP CONGESTION CONTROL OVER ARQ

A. End-to-end Model

We consider a TCP connection between two hosts such that the first link on the end-to-end path from sender to receiver is a wireless, radio link. Such a scenario is common in mobile communication and is illustrated in figure 1.

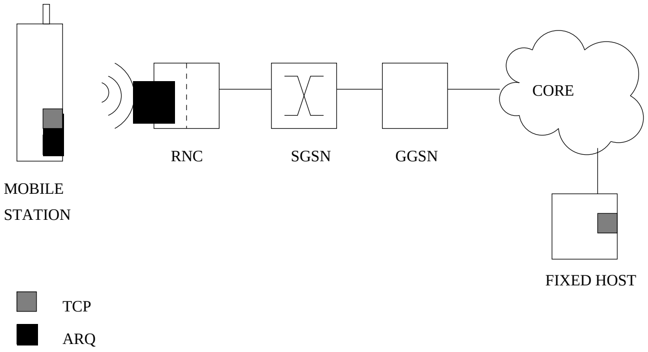

Fig. 1. End-to-end System

As seen from the figure, the model has two portions: a radio, mobile link and a wired, fixed network. At the source, mobile host and at the base station, two error control schemes are mounted independently in different layers : ARQ at the data link layer and TCP at the transport layer. At the fixed, receiver side, a peer TCP entity is present.

For a connection supporting a bulk TCP transfer, i.e., a flow with a large amount of data to send, such as FTP transfers, we focus on the end-to-end reliability as offered by both TCP and ARQ and assume that :

(1) On the wireless link, there is no packet loss but some packets may be corrupted under adverse radio link conditions. In our study, we consider two models for the bit error patterns on the radio link: independent and correlated.

(2) On the wired network, packets may get lost when congestion occurs.

B. ARQ

We consider a classical Go-back-N ARQ scheme as described in [12], with ARQ source at the mobile station and ARQ receiver at the base station. The ARQ sender segments the TCP segment into constant-size ARQ frames and sends them in an ordered fashion, to the limit of N frames at once with no need for an acknowledgment. The receiver sends an ACK_{ARQ} for every correctly received frame which makes N slide on the subsequent frames. If no ACK_{ARQ} concerning frame i is received after the $N-1$ following frames have been sent, the sender times out, goes back to frame i and retransmits all N frames in order from frame i. The receiver does not accept out-of-order frames.

In any case, ARQ tramissions and retransmissions introduce latency in processing frames and sending them through the air channel. ARQ operates smoothly except in the presence of corrupted frames. The underlying assumption is that time for ARQ frames processing is comparable to the Round Trip Time (RTT) for ARQ, RTT_{ARQ}.

Let us note that we assume that no error gets inserted in the ACK_{ARQ}. This can be justified by the fact the up- and downlinks are physically independent; adverse radio conditions may be present at the uplink from mobile station to base station and not at the downlink.

C. TCP

We consider the Reno version of TCP, as described in [11]. Let W denote the TCP congestion window size. TCP performs slow start and congestion avoidance. In order to keep things simple, we do not consider Fast Retransmit and Fast Recovery. Furthermore, as we model the steady state of a bulk TCP transfer, we assume that TCP is always in congestion avoidance. Hence, W is increased by $1/W$ each time an ACK_{TCP} is received. In fact, we assume that TCP sends one cumulative ACK_{TCP} for b consecutive TCP segments, keeping track of the model and terminology adopted in [1]. Packet loss is detected in one of two ways, either upon reception of a triple-duplicate ACK_{TCP} (denoted by TD in [1]), or upon expiration of a Time-Out (denoted by TO). In case of a TD, W is decreased by half, while upon expiration of a TO, W is decreased to 1.

To keep track of the model developed in [1], the behavior of TCP is modeled in terms of rounds: a round starts with the back-to-back transmission of W packets and ends with the reception of the first ACK_{TCP} corresponding to one of the W packets sent. We assume that the processing time for a TCP is negligeable with respect to RTT_{TCP}. Under this assumption, the duration of a round is equal to RTT_{TCP} and is independent of W. Moreover, we assume that the loss behavior is bursty, i.e., packet losses are correlated within a back-to-back transmission. Hence, when a packet is lost, all remaining packets in the same round are lost as well. Furthermore, under the assumption that rounds are separated by one RTT_{TCP}, loss in one round is independent of loss in other rounds. Let T_o denote the TCP time-out and let p denote the loss rate in the wired portion of the network.

The expression in [1] for the steady-state TCP throughput, in a wired context only, as a function of loss probability p is the following:

$$Th(p) = \frac{1}{RTT_{TCP}\sqrt{\frac{2bp}{3}} + T_o min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)} \tag{1}$$

D. TCP over ARQ

So far, we considered the independent operation of both TCP and ARQ. We now turn to the operation of TCP over ARQ as illustrated by the model in Figure 2.

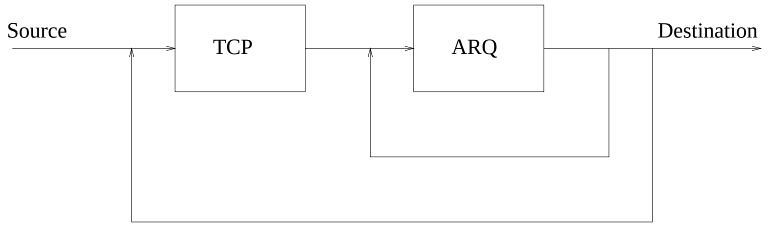

Fig. 2. TCP over ARQ Block Diagram

We assume that the TCP segment size is fixed as is the ARQ frame size. One TCP segment is hence divided into n ARQ frames that are transmitted over the radio link; we assume that n is an integer. RTT_{TCP} is the round trip time seen from the TCP layer and includes the time spent in the ARQ layer. To keep track of the above mentioned result for TCP and particularly Eqn. 1, which is itself based on the model found in [1], we have to first ensure that the so-called model can be kept as is and second we have to derive RTT_{TCP} as a function of the ARQ parameters as well.

The first action is thus to validate the notion of TCP rounds in our present context. Let us recall that the notion of TCP rounds relies on the negligeable effect of TCP processing time compared to RTT_{TCP}. ARQ introduces latency, in terms of frame processing, transmission and if need be, retransmission. This cumulative delay introduced by ARQ is nevertheless part of RTT_{TCP} and shall not introduce any deficiency to the notion of TCP rounds. In other words, when the delay introduced by ARQ grows unboundedly, all it will cause is a premature TO at the TCP module. This in turn will cause W to be set to 1 and the notion of rounds is kept as is. It is important to understand that ARQ and its consequent delay are merely an extra delay element seen from the TCP layer which is the layer under study. From a TCP perspective, [1] only mentions a lower limit on RTT_{TCP} in the definition and validation of the TCP rounds; no upper limit is mentioned. This assumption shall be verified here too. The second action, which is RTT_{TCP} as a function of ARQ parameters as well as the loss probability at the wired portion of the network, is derived in the following analytical sections.

III. ERROR SCHEME IN THE AIR CHANNEL

There are two major error patterns in the air channel. Errors may happen in a completely independent manner or in a correlated fashion; a low degree of correlation corresponds to a fast fading channel whereas a high figure corresponds to a slow fading one.

A. Case of independent bit errors

Let BER (FER) denote the bit (frame) error rate.

$$FER = 1 - (1 - BER)^m$$

m being size of the ARQ frame.

B. Case of correlated bit errors

According to [5], the pattern of errors follows a first-order Markov model with transition matrix:

$$\mathbf{M} = \begin{bmatrix} c & q \\ r & s \end{bmatrix}$$

where c is the probability of correlated success, s probability of correlated error, q probability of error after success and r probability of success after error. $1/r$ is the average length of one burst of errors. r and q take larger values for fast fading than for slow fading channels.

Rayleigh fading is due to multi-path interference. Mobiles moving at high speed experience fast fading, whereas slow fading is typical at low speed. Modulation and coding techniques, interleaving schemes and correlation receivers are means used to reduce fading. In UMTS setting, the selection of W-CDMA for the air interface allows further the use of fast power control to reduce fading effects. The matrix parameters depend essentially on the following factors: i. mobile velocity, ii. carrier wavelength, iii. bit processing rate, iv. frame length and v. efficiency of correction techniques.

IV. SILENT ARQ

We now consider the case where ARQ does not disturb TCP, i.e., does not cause a premature TO nor TD; it only introduces latency. We consider b-tuplets of TCP packets and their bn constituing ARQ frames. Each TCP round ends upon reception of one cumulative ACK_{TCP} referring to the first b-tuplet TCP packets in the congestion window of size W.

A. Case of independent bit errors

The case of independent and identically distributed bit errors is a subset of the correlated error case with transition matrix:

$$\mathbf{M} = \begin{bmatrix} 1 - FER & FER \\ 1 - FER & FER \end{bmatrix}$$

We hence turn directly to the case of correlated error.

B. Case of correlated error

In the mobile, wireless context, RTT_{TCP}, accounting for both air channel and wired network, is a combination of the RTT of the wired network, hereby denoted by RTT_{wire}, and the RTT corresponding to the air channel, hereby denoted by RTT_{ARQ}. Let D_{ARQ} denote the fixed component of the delay at the ARQ entity and which quantifies the time for processing one ARQ frame.

Correlated errors are equivalent to periods of bursty errors with error burst size equal to the number of corrupted frames. To keep track of the model in [1] and particularly the independence of rounds, packets and losses, which mathematically correspond to a renewal process, the error periods should be contained inside one packet. The latter should be delimited by error free frames which is the case because in the errored period, there is no possibility for start or ending of a TCP packet.

During the transmission of bn consecutive frames, we now investigate the probability of having $U = u$ bursty error periods, their length and its impact on ARQ retransmission and latency. For Go-back-N ARQ, when a frame is corrupted (the first of N), the remaining frames are not accepted even though they are error free. Let Z be the number of consecutive errored frames in a bursty errored period. Bursty, consecutive corruption may be distinguished from the case of a single frame error as follows. For a single frame error, the corrupted frame corresponds to the start of a cycle of N frames and results in the drop of next $N-1$ frames and retransmission of all N frames. In the consecutive corruption case, the first frame out of N is indeed corrupted; however, if the next frame is also corrupted, this will not cause twice retransmission of N frames but just one N retransmission of all corrupted and corruption-free frames.

Formally, there exists a K such that $(K-1)N < Z \leq KN$. The next consecutive frames starting from Z are also lost up to KN. Indeed, N frames are retransmitted K times. Let U denote the number of errored periods, of length KN. We model the errored periods as periods of time that get inserted into the back-to-back transmission of ARQ frames, causing error, retransmission and delay.

For the good transmission of b TCP segments, there must be nb good ARQ frames; other frames sent at ARQ level may get corrupted and discarded or discarded in a cycle of N frames. The corrupted frames correspond to the periods of bursty errors. Let us consider two 'consecutive' good frames in the set of correct frames, i.e., bursty errors get inserted between them. The conditionnal probability of having a set of consecutive corrupted frames between these two correct frames is given by q, as stated in Subsection III-B.

Indeed, two consecutive frames in the set of correct frames are separated by a bursty-error period under the following condition: the frame sent just after the first of the two good frames is corrupted; this happens with conditionnal probability q. Given that the two good frames of interest are consecutive in the set of correct frames, this condition is equivalent to the presence of a burst. It is important to note that the periods of bursty error are i.i.d. with respect to the ensemble of correct frames; the correlation lies in the consecutive corrupted frames within a bursty error period. Now if every bursty-error period contains only one errored frame, this brings us back to the i.i.d. error pattern.

There can be at most $bn-1$ periods of bursty errors inserted into the back-to-back transmission of bn correct frames. Hence, the probability of having $U=u$ periods of error inserted in the bn

ARQ frames is given by:

$$P(U=u)=\left(\begin{array}{c} bn-1 \\ u \end{array}\right)(1-q)^{bn-1-u}\,(q)^{u}$$

Next, inside the ith period of error, the event Z of consecutive errors, with $(K-1)N < Z < KN$, is given by:

$$P(K=k)=s^{(k-1)N}r+s^{(k-1)N+1}r+...+s^{kN-1}r=s^{(k-1)N}(1-s^{N})$$

This causes RTT_{TCP} to have the following expression:

$$RTT_{TCP}=RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}\sum_{i=1}^{U}K_{i}$$

The mean RTT_{TCP} is thus given by:

$$E(RTT_{TCP})=RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}E(\sum_{i=1}^{U}K_{i})$$

Since U and K are independent, we have,

$$E(RTT_{TCP})=RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}E(U)E(K)$$

$$E(K)=\sum_{k=1}^{\infty}ks^{(k-1)N}(1-s^{N})=\frac{1}{1-s^{N}}$$

and

$$E(U)=q(nb-1).$$

Thus,

$$E(RTT_{TCP})=RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}\frac{q(nb-1)}{1-s^{N}}$$

Hence, the throughput of TCP as a function of both probability of loss p and correlated error parameters q and s is

$$Th(p,q,s)=\frac{1}{(RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}\frac{q(nb-1)}{1-s^{N}})\sqrt{\frac{2bp}{3}}+T_{O}min(1,3\sqrt{\frac{3bp}{8}})p(1+32p^{2})} \qquad (2)$$

and the expression of the throughput as function of both probability of loss p and i.i.d. error parameter FER is

$$Th(p,FER)=\frac{1}{(RTT_{wire}+nbD_{ARQ}+RTT_{ARQ}+ND_{ARQ}\frac{FER(nb-1)}{1-FER^{N}})\sqrt{\frac{2bp}{3}}+T_{o}min(1,3\sqrt{\frac{3bp}{8}})p(1+32p^{2})} \tag{3}$$

V. INFLUENCE OF ARQ ON TCP

Simple Go-back-N ARQ cannot cause a TD. TD is caused by the absence of a packet at destination, due to loss or error, and the receipt of next packets which themselves cause the sending of duplicate ACKs requesting the lost segment. Go-back-N does not accept out of order packets which consequently cannot cause TD at the TCP destination and hence TD is only caused by loss at the wired network level.

A TO at the TCP sender happens when the timer expires before the receipt of an ACK acknowledging the packet that it sent. The ACK is not sent back by the receiver because it did not receive any packet (implicitly, we assume ACKs are not lost). This means that either a packet is lost or severely delayed. ARQ may play a role in the delay component (we assume implicitly that ARQ corrects all errors). We then investigate the delay component added by ARQ and identify its role in the total RTT_{TCP}, its margin for a silent ARQ and the condition on the triggering of a TO.

Let M be the total number of ARQ frames needed to get an error-free TCP segment, including the original n frames plus the retransmitted ones because of error. The total delay caused by the transmission of M ARQ frames is :

$$RTT_{TCP} = MD_{ARQ} + RTT_{ARQ} + RTT_{wire}$$

This delay component influences TCP when it becomes larger than or equal to T_o. Let M^* be the critical value of M which satisfies the equality $RTT_{TCP} = T_o$ and is equal to:

$$M^* = \frac{T_o - RTT_{ARQ} - RTT_{wire}}{D_{ARQ}}$$

At this point, we investigate the profile of M in order to establish the likelihood that ARQ triggers a TO at TCP, i.e., the probability of M being equal to M^*. Let us recall that the value of T_o is well tuned to match the behavior and thus the value of RTT_{TCP} in the case of wired network. Let V denote the number of error periods that get inserted within the transmission of one TCP segment, i.e., within a set of n correct ARQ frames. Following the same reasoning as in Section IV, for $V = v$ periods of bursty errors within the transmission of one TCP segment, with ith bursty error of length $K_i = k_i$, for $i = 1, ..., v$, this adds a delay component corresponding to the transmission of $A = N \sum_{i=1}^{v} k_i$ additionnal frames. A TO is then triggered off when A reaches a critical value $A^* = M^* - n$.

Recall that the distribution of K is given by $P(K = k) = s^{(k-1)N}(1 - s^N)$. This is a decreasing function of k, with limit value 0, as shown in Figure 3. This function can be heavy tailed corresponding to slow fading channels; all the more true for large values of s. Moreover, the probability of having an error period, given by q, is large for fast fading and small for low fading channels.

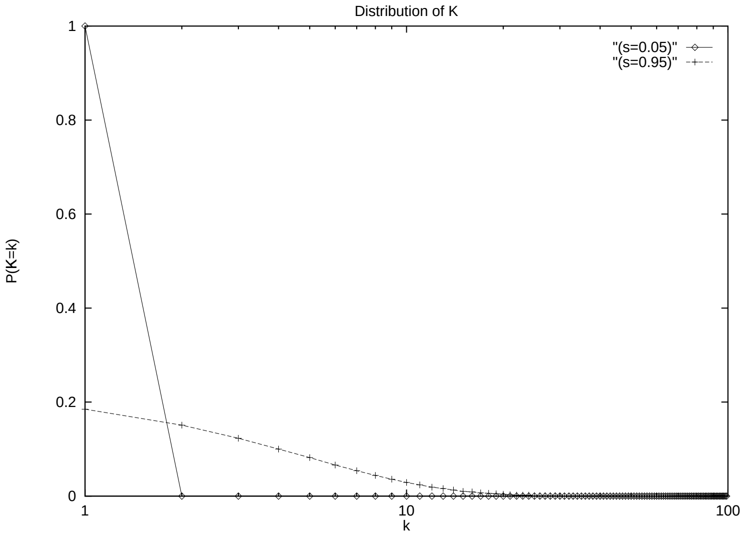

Fig. 3. Distribution of K for (N=4, s=0.05) and (N=4, s=0.95)

Hence, a fast fading channel shall experience many error periods with a short error burst size whereas slow fading channels are characterized by rare error periods with long error burst size. Hence, in case of fast fading, the distribution of the error periods and their length appears as relatively uniform and regular over time whereas, at the same time scales, in case of slow fading, the error periods seem more random. Thus, the triggering of a TCP TO by ARQ can be neglected in the case of fast fading channels. On the contrary, slow fading channels contribute to this event as shown next.

Let us assume that a TO is triggered off when a bursty error period becomes very large compared to the mean value of K, and let us consider that the other bursty error periods belonging to the transmission of the targetted TCP segment just add a delay component assumed to be constant and equal to the mean value for A, denoted by $\bar{A}$. Since we are in a slow fading case, q is small and the presence of an error period becomes itself a rare event as q decreases, which justifies our assumption. A TO is triggered off if the length of the large bursty error period reaches the critical value $A^* - \bar{A}$, corresponding to the critical value

$$K^* = \lfloor \frac{A^* - \bar{A}}{N} \rfloor = \lfloor \frac{M^* - n - \bar{A}}{N} \rfloor$$

Under this assumption, within the transmission of one TCP segment, the event 'a TO is triggered off by ARQ' is equivalent to the event 'a bursty error period of length larger than K^* gets inserted between any two consecutive frames of the set of correct frames'. Hence, we can approximate the probability of having a TO triggered off by ARQ by:

$$\text{Prob (ARQ triggers off a TO)} \approx \frac{qP(K > K^*)}{n-1}$$

where

$$P(K > K^*) = 1 - P(K < K^*) = 1 - \sum_{k=0}^{K^*} s^{(k-1)N}(1 - s^N) = s^{NK^*}$$

and hence,

$$\text{Prob (ARQ triggers off a TO)} \approx \frac{q}{n-1} s^{NK^*}$$

VI. NUMERICAL APPLICATION

The TCP parameters for the fixed portion of the network are inspired from [1] where RTT$_{wire} = 0.2s$, $b = 10$ and $T_o = 0.4$. The ARQ paramaters are $n = 10$, $N = 10$, $RTT_{ARQ} = 0.005$ and $D_{ARQ} = 0.001$ as inspired from [9], [5] and [13] for an air channel with an average FER equal to 0.01.

For a stationary process, FER and the matrix parameters are related as follows : $FER = \frac{q}{q+1-s}$. As $FER < 1, s < 1,$ and $q < 1$, the former equality mathematically imposes that q

ranges from 0 to $\frac{FER}{1-FER}$ and that s ranges from 0 to 1. We focus on the pairs of (q, s) values shown in Table I., corresponding to $FER = 1.10^{-2}$ (where Case (5) models a random channel) .

	(1)	(2)	(3)	(4)	(5)
q	5.10^{-4}	2.10^{-3}	7.10^{-3}	$9.5.10^{-3}$	1.10^{-2}
s	$9.5.10^{-1}$	8.10^{-1}	3.10^{-1}	5.10^{-2}	1.10^{-2}

TABLE I

A. *Effect of the air channel*

We first suppose that no ARQ entities are mounted and we examine the degradation of TCP throughput on the end-to-end path, in the presence of the air channel. On the air channel the average packet error rate PER seen by TCP is: $PER = 1 - (1 - FER)^n$. On the end-to-end path from sender to receiver, TCP sees a global average packet loss rate equal to $1 - (1 - PER)(1 - p) = p + PER - p * PER$. TCP throughput is as given in Eqn. 1. if the wireless portion of the path did not exist.

Figure 4 shows the TCP throughput versus p for both cases.

TCP throughput is considerably decreased by the introduction of the air component because of a much higher loss rate due to loss in the wired portion as well as error in the air channel. Indeed, each corrupted segment is understood at the TCP layer as a congestion indication which triggers off congestion avoidance mechanism; thus, each time a corrupted segment is detected at the receiver, the sender reduces the size of congestion window W by half and the throughput thus decreases.

TCP throughput as a function of both probability of loss p and probability of packet error PER is

$$Th(p,PER)=\frac{1}{RTT_{TCP}\sqrt{\frac{2b(p+PER-p*PER)}{3}}+T_o min(1,3\sqrt{\frac{3b(p+PER-p*PER)}{8}})(p+PER-p*PER)(1+32(p+PER-p*PER)^2)} \quad (4)$$

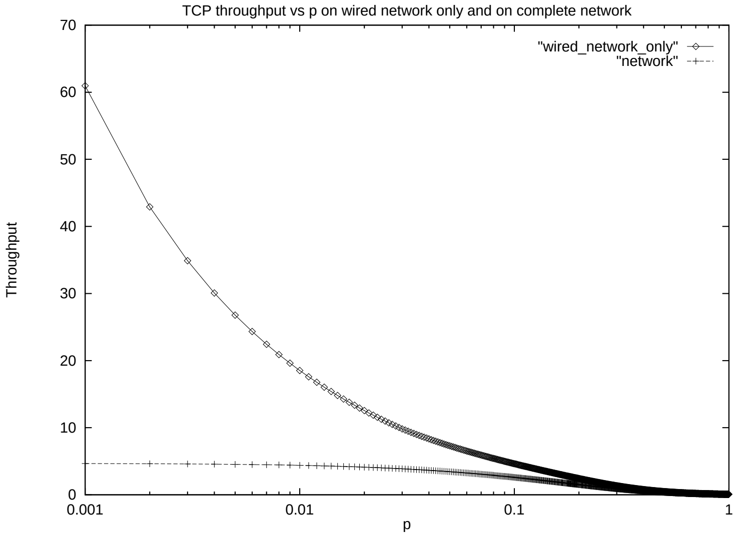

Fig. 4. Effect of the air channel

B. TCP only versus TCP over ARQ

We examine the role of the ARQ layer in comparing the throughput with no ARQ layer in a network with packet loss rate $p + PER - p * PER$ and the throughput with ARQ layer in a network characterized by FER on the air channel and p on the wired part.

Figure 5 shows the throughput versus p for both cases.

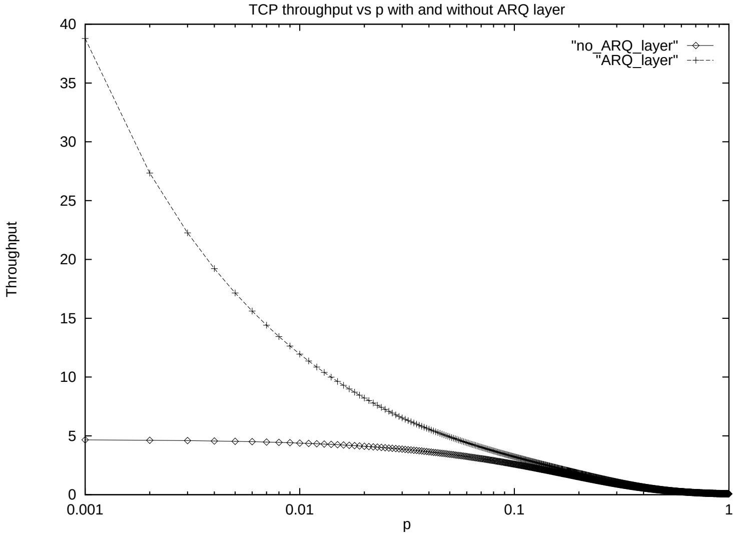

Fig. 5. Effect of the ARQ layer

The presence of ARQ layer considerably improves TCP throughput. The ARQ layer reduces the loss rate seen by TCP layer to p. The price for this decreased loss rate is that ARQ adds latency in processing ARQ frames so that RTT_{TCP} is higher; but it is worth spending more time at the wired portion because, as TCP does not have to deal with corrupted segments, W does get reduced by half only when congestion really occurs on the wired part of network and thus the throughput is not wasted.

C. Effect of the error pattern

We examine the effect of the error pattern and compare the results for cases (1), (2), (3) and (4).

We first have to check the validity of the assumption of "silent ARQ", i.e. ARQ doesn't disturb TCP, it only introduces latency. We compute the values of M^*, $\bar{A}$, K^* and Prob(ARQ triggers off a TO), hereby denoted Prob(TO), for cases (1) and (2), shown in Table II.

error pattern	$\bar{A}$	K^*	Prob(TO)
(1)	0.4	4	3.10^{-7}
(2)	0.6	4	10^{-12}

TABLE II

Note that since $A = N\sum_{i=1}^{v} k_i$, $\bar{A} = NE[K]E[V] = \frac{1}{1-s^N} qN(n-1)$. $M^* = 160$ and ARQ is well silent.

C.1 Effect of the correlation

We compare the performance of a random channel to the performance of a correlated-corruption one. Figure 6 shows the throughput for cases (2) and (5).

We observe that the throughput is higher in case of correlated errors; this may be due to Go-back-N ARQ, i.e. at least N frames are retransmitted when error occurs, what must be more accurate if errors happen in a correlated manner.

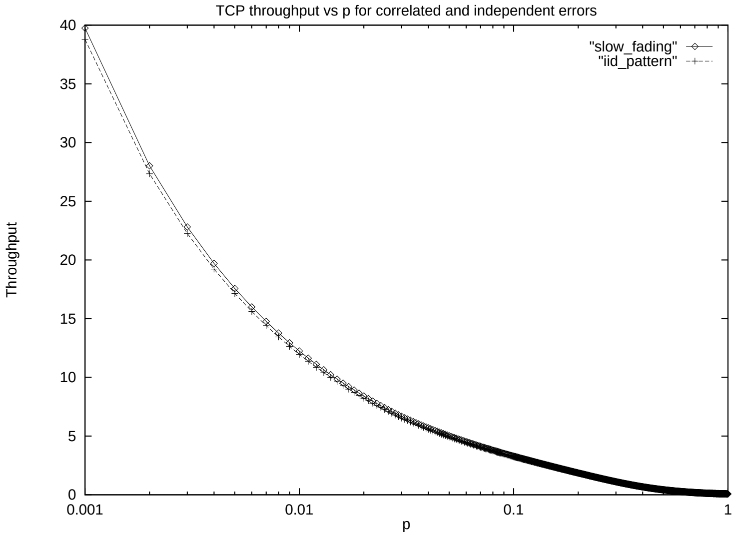

Fig. 6. Independent and correlated errors

C.2 Effect of the matrix parameters

Figure 7 shows the throughput for cases (1), (2), (3), (4).

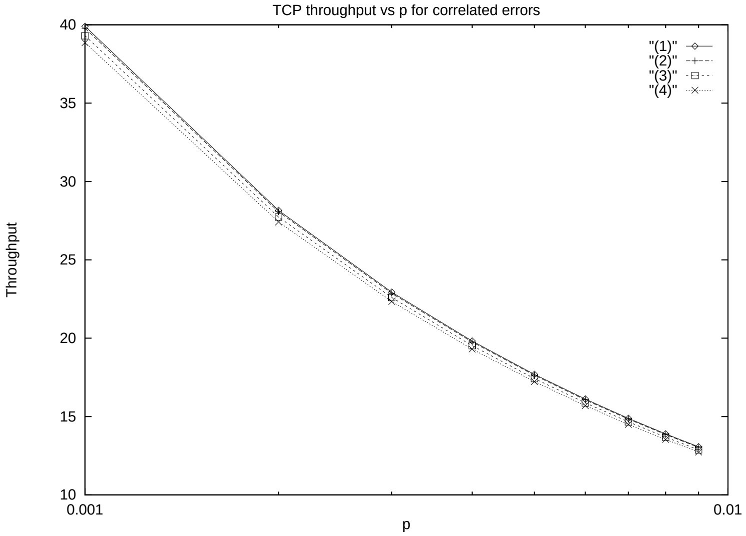

Fig. 7. Fading channels

TCP throughput is all the more increased as fading gets slower. Recall that the average length of bursty error periods, measured in number of errored frames, is given by $\frac{1}{1-s}$. Thus, if $s = 8.10^{-1}$ (respectively $s = 5.10^{-2}$), this average length of bursty error periods is equal to 5 (respectively 1). Hence, as we consider GO-BACK-10 ARQ, a bursty error period in case (2) will have the same effect on the number of retransmitted frames as a single frame in error in (4), i.e. there will be 10 retransmitted frames in both cases. As error periods happen more often in case (4), the throughput is decreased in this case.

Now, case (1) ($s = 9.5.10^{-1}$) gives an average length of error periods equal to 20 and GO-BABK-10 ARQ performs even at best in this case. This tells us that the smaller number of error periods is more significant than their increased length.

VII. CONCLUSION

In this paper, we presented an analytical model for the steady-state throughput of TCP on a heterogeneous end-to-end path, composed of both wireless and wired portions, ARQ being active at the former and TCP at the latter as well as the mobile station, as shall be the case in forthcoming UMTS standards. Our TCP model refers to the one developped in [1], and captures the effect of the presence of ARQ under TCP on the wireless portion of the network and leads to an expression of TCP throughput as a function of both loss rate at the wired portion and error parameters at the air channel.

We have first shown that TCP throughput is highly degraded by the addition of a wireless link characterized by significant error rate. Then, we examined the benefit brought by the mounting of ARQ entities on the wireless part of the path and we found that the action of ARQ considerably enhances TCP throughput; the reason behind this is that detecting and correcting errors at the link level prevents them from uselessly triggering TCP congestion avoidance mechanism. In the presence of ARQ, TCP works transparently on an error-free wireless medium and the correction of errors at the air channel just adds a delay component in the round trip time seen from TCP layer. Eventulally, we compared different levels of correlation for the error pattern at the air channel and we found that the benefit of ARQ is all the more significant than the correlation degree is higher; the reason may be that we have considered GO-BACK-N ARQ which should be more efficient in error correction when errors are gathered in bursts.

Future work remains. First, we have assumed that ARQ does correct all errors. In fact, ARQ does not correct all errors; TCP has yet to deal with a small percentage of non-corrected errors on the air channel. Second, in this work, we focused on Go-Back-N ARQ. It might as well be completed by a further work on Selective Repeat ARQ instead where out-of sequence correct ARQ frames are not rejected. Furthermore, ARQ has a timer that expires upon a maximal number of unsuccessful retransmissions, leading to the loss of those frames. This introduces loss on the air channel which may be interesting to investigate in a future work in connection to loss at the TCP/IP layers.

REFERENCES

[1] J. Padhye, V. Firoiu, D. Towsley, J. Kurose. Modeling TCP Throughput: A simple Model and its Empirical Validation. In ACM SIGCOMM’98.

[2] J. Padhye, V. Firoiu, D. Towsley. A Stochastic Model of TCP Reno Congestion Avoidance and Control. CMPSCI Technical Report 99-02.

[3] M. Mathis, J. Semske, J. Mahdavi, T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review, 27(3), July 1997.*

[4] P. Karn, C. Partridge. Improving Round-Trip time estimates in reliable transport protocols. *Computer Communication Review, 17(5), August 1987.*

[5] M. Zorzi, R.R. Rao, L.B.Milstein. Error statistics in data transmission over fading channels. Accepted for publication in the IEEE Transactions on Communications.

[6] A. Chockalingam, M. Zorzi, R.R. Rao. Performance of TCP on Wireless Fading Links with Memory. In ICC’98.

[7] A. Chockalingam, M. Zorzi, V. Tralli. Wireless TCP Performance with Link Layer FEC/ARQ. In ICC’99.

[8] Y. Bai, P. Zhu, A. Rudrapatna, A.T. Ogielski. Performance of TCP/IP over IS-2000 Based CDMA Radio Links.

[9] K. Akabane, H. Nakase, E. Kudoh. Error Control Scheme for TCP Packet Radio Systems. In IEEE VTC’99.

[10] W. Stevens. *TCP/IP Illustrated, Vol.1 The Protocols.* Addison-Wesley, 1994.

[11] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC2001, Jan 1997.

[12] D. Bertsekas, R. Gallager. *Data Networks, Second Edition.* Prentice Hall, 1992.

[13] H. Holma, A. Toskala. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications.* John Wiley and Sons, 2000.