Bandwidth tradeoff between TCP and link-level FEC *

Chadi Barakat and Eitan Altman
INRIA, 2004 route des Lucioles, 06902 Sophia Antipolis, France
Email : {cbarakat,altman}@sophia.inria.fr

Abstract. FEC is widely used to improve the quality of noisy transmission media as wireless links. This improvement is of importance for a transport protocol as TCP which uses the loss of packets as an indication of network congestion. FEC shields TCP from losses not caused by congestion but it consumes some bandwidth that could be used by TCP. We study in this paper the tradeoff between the bandwidth consumed by FEC and that gained by TCP.

1 Introduction

Forward Error Correction (FEC) is widely used to improve the quality of noisy transmission media as wireless links [2, 4]. This improvement is of importance for a transport protocol as TCP [8, 16] which uses the loss of packets as an indication of network congestion. 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, and hence in a deterioration of the performance of the TCP transfer [2]. In the following, we will only focus on transmission errors on wireless links and we will call the corrupted packets *non-congestion losses* or *link-level losses* since they appear at a level below IP.

The idea behind FEC is to transmit on the wireless link, together with the original data, some redundant information so that a corrupted packet can be reconstructed at the output of the link without the need for any retransmission from TCP [2, 4]. Normally, this should improve the performance of TCP since it shields it from non-congestion losses. But, FEC consumes some bandwidth. Using much FEC may steal some of the bandwidth used by TCP which deteriorates the performance instead of improving it. Clearly, a tradeoff exists between the bandwidth consumed by FEC and that gained by TCP. We analyze this tradeoff in this paper. The question that we asked is, given a certain wireless link with certain characteristics (bandwidth, error rate, burstiness of errors), how to choose the amount of FEC so that to get the maximum gain in TCP performance. A mathematical model and a set of simulations are used for this purpose.

2 Model for non-congestion losses

Consider a long-life TCP connection that crosses a network including a noisy wireless link of rate μ. We suppose that the quality of the noisy link is improved by a certain FEC mechanism that we will model in the next section.

Most of the works on TCP performance [10, 11, 13] make the assumption that the loss process of TCP packets is not correlated. Packets are assumed

* A detailed version of this paper can be obtained upon request from the authors.

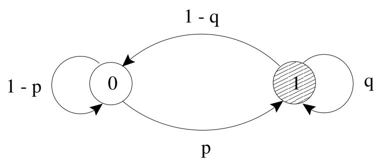


Fig. 1. The Gilbert loss model

to be lost independently with the same probability P. This does not work for wireless links where transmission errors tend to appear in bursts [3–5, 9]. The model often used in the literature to analyze correlated losses on a wireless link is the one introduced by Gilbert [3, 4, 7, 9]. It is a simple ON/OFF model. The noisy link is supposed to be in one of two states: 0 for Good and 1 for Bad. A packet is lost if it leaves the link while it is in the Bad state, otherwise it is supposed to be correctly received. We use such a model in our work. A discrete-time Markov chain (Fig. 1) with two states (Good and Bad) models the dynamics of the wireless link. We focus on the loss process of link-level packets also called *transmission units*. We suppose that a TCP packet is transmitted over the wireless link using multiple small transmission units [3, 4]. A transmission unit can be a bit, a byte, an ATM cell, or any other kind of link-level blocks used for the transmission of TCP/IP packets. The state of the wireless link is observed upon the arrivals of transmission units at its output. We suppose that units cross continuously the link. If no real units exist, *fictive units* are inserted.

Let p denote the probability that the wireless link passes from Good state to Bad state when a transmission unit arrives at its output. Let q denote the probability that the link stays in the Bad state. q represents how much the loss process of transmission units is bursty. The stationary probabilities of the Markov chain associated to the wireless link are equal to: $\pi_B = p/(1-q+p)$ and $\pi_G = (1-q)/(1-q+p)$. Denote by L_B and L_G the average lengths of Bad and Good periods in terms of transmission units. A simple calculation shows that,

$$L_B = 1/(1-q), \qquad L_G = 1/p. \tag{1}$$

The average loss rate, denoted by L, is equal to

$$L = L_B/(L_B + L_G) = p/(1-q+p) = \pi_B. \tag{2}$$

3 Model for FEC

The most common code used for error correction is the *block* code [14, 15]. Suppose that data is transmitted in units as in our model for the noisy link. Block FEC consists in grouping the units in blocks of K units each. A codec then adds to every block a group of R redundant units calculated from the K original units. The result is the transmission of blocks of total size $N = K + R$ units. At the receiver, the original K units of a block are reconstructed if at least K of the total N units it carries are correctly received. This improves the quality of the transmission since a block can now resist to R losses without being discarded.

In our work, we consider a block FEC implemented on the wireless link in the layer of transmission units. We ignore any FEC that may exist below this layer (e.g., in the physical layer). The input to our study is the loss process of transmission units which is assumed to follow the Gilbert model. In what follows, we will show how much the parameters of the FEC scheme (N, K) impact the performance of TCP transfers.

4 Approximation of TCP throughput

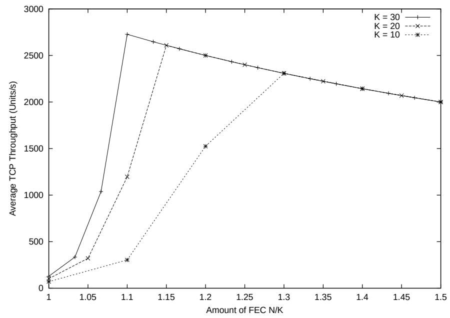
Consider the throughput as the performance measure that indicates how well TCP behaves over the wireless link. Different models exist in the literature for TCP throughput $[1, 10, 11, 13]$. Without loss of generality, we consider the following simple expression for TCP throughput in terms of packets/s: $X = (1/RTT)\sqrt{3T/2} = (1/RTT)\sqrt{3/(2P)}$ [11]. This expression is often called the *square root formula*. $T = 1/P$ denotes the average number of TCP packets correctly received between packet losses. P denotes the probability that a TCP packet is lost in the network. In case of bursty losses, P represents the probability that a TCP packet is the first loss in a burst of packet losses [13]. This is because the new versions of TCP (e.g., SACK [6]) are designed in a way to divide their windows one time by two for a burst of packet losses. RTT is the average round-trip time seen by the connection. Note that it is also possible to use in our analysis other more sophisticated expressions for TCP throughput (e.g., [1]).

Suppose that the wireless link is the bottleneck on the path of the connection. Thus, in the absence of FEC, the throughput of TCP is upper bounded by μ. We write $X = \min\left((1/RTT)\sqrt{3T/2}, \mu\right)$.

Our objective is to express the throughput of TCP as a function of the parameters of the loss process of transmission units (p, q) and the parameters of the FEC scheme (N, K). We already have the expression of the throughput as a function of what happens at the packet level (P). What we still need to do is to relate the loss process of TCP packets to the loss process of transmission units. To simplify the analysis, we consider the best case when packets are only lost on the wireless interface whenever the wireless bandwidth μ is not fully utilize (possible since the wireless link is assumed to be the bottleneck). In the case when transmission units are lost independently of each other $(p = q)$, $P = 1/T$ is simply equal to the probability that a TCP packet is lost while crossing the wireless link. This case is studied in the next section. In the case when transmission units are lost in bursts, T must be calculated as the average number of TCP packets correctly transmitted between bursts of packet losses. This is done in Section 6 where we study the impact of the correlation of unit losses on the performance of TCP.

Now, even though it increases T, the addition of FEC consumes some bandwidth and decreases the maximum throughput the TCP connection can achieve. Instead of μ, we get $K\mu/N$ as a maximum TCP throughput. If we denote by S the size of a TCP packet in terms of transmission units, the throughput of TCP in presence of a FEC scheme (N, K) and in terms of units/s can be written as,

$$X_{N,K} = \min\left((S/RTT)\sqrt{3T_{N,K}/2}, K\mu/N\right). \quad (3)$$



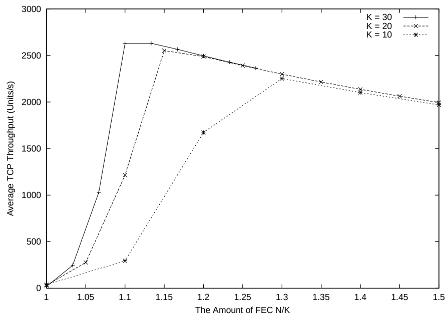


Fig. 2. Model: X vs. N/K and K

Fig. 3. Simulation: X vs. N/K and K

5 The case of non-correlated losses

Consider the case when transmission units are lost independently of each other with probability p ($p = q$). Thus, TCP packets are also lost independently of each other but with probability $P_{N,K} = 1/T_{N,K}$ which is a function of the amount of FEC (N, K). The throughput of TCP can be approximated by using (3).

5.1 The analysis

Suppose that TCP packets are of the size of one link-level block ($S = K$ units). Given a certain block size (K) and a certain amount of FEC (N, K), the choice of the size of the TCP packet in terms of blocks is another problem that we will not address in this paper. A TCP packet is then lost when more than R of its units are lost due to transmission errors. This happens with probability $P_{N,K} = \sum_{i=0}^{K-1} \binom{N}{i} (1-p)^i p^{N-i}$.

It is clear that the addition of FEC at the link level reduces the loss probability of TCP packets. This addition improves the throughout whenever the first term of the minimum function in (3) is smaller than the second term. When these two terms are equal, the quantity of FEC added to the wireless link is sufficient to eliminate the negative effect of non-congestion losses on TCP. We say here that FEC has *cleaned* the link from TCP point of view. Any increase in FEC beyond this point results in a throughput deterioration. There will be more FEC than what is needed to clean the link. Given μ, K, and p, the optimal quantity of FEC from TCP point of view is the solution of the following equation,

$$(N/(KRTT))\sqrt{3/(2P_{N,K})} = \mu. \tag{4}$$

5.2 Analytical results

We show in Figure 2 how the throughput of TCP varies as a function of the ratio N/K (FEC rate) for different values of K (10, 20, and 30 units). RTT is taken equal to 560 ms and the wireless link bandwidth μ to 3000 units/s. We can see this scenario as the case of a mobile user downloading data from the Internet through a satellite link. This value of μ is approximately equal to the maximum ATM cell rate on a T1 link (1.5 Mbps). p is set to 0.01.

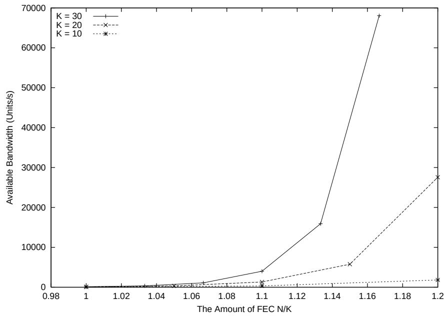


Fig. 4. Model: Optimal FEC vs. μ

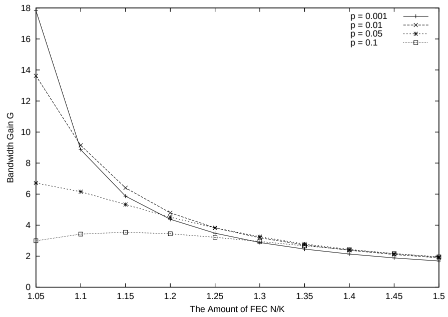


Fig. 5. Model: G vs. N/K and p

It is clear that the performance improves considerably when FEC is added and this improvement continues until the optimum point given by (4) is reached. Beyond this point, any increase in FEC deteriorates the throughput. Also, we notice that for a certain quantity of FEC, an increase in K improves the performance. An increase in K results in a faster window growth. TCP window is increased in terms of packets rather than bytes [16]. The TCP source then returns faster to its rate prior to the detection of a non-congestion loss.

In Figure 4, we plot the left-hand term of (4) as a function of N/K for the same three values of K. These curves provide us with the optimal amount of FEC for given μ, p, and K. We see well how the increase in K reduces considerably the amount of FEC needed to clean the wireless link from TCP point of view. Given μ, a compromise between K and FEC rate must be done. First, we choose the largest possible K, then we choose the appropriate amount of FEC.

For $\mu = 3000$ units/s and $K = 20$, we show in Figure 6 how the throughput of TCP varies as a function of p for different values of N. It is clear that adding just one redundant unit to every FEC block results in a considerable gain in performance especially at small p. Adding more redundancy at small p deteriorates slightly the performance since the link is already clean and the additional redundancy steals some of the bandwidth used by TCP. This is not the case at high p where much redundancy needs to be used in order to get good performance. Note that even though an excess of FEC reduces the performance of TCP when losses are rare, the reduction is negligible in front of the gain in performance we obtain when losses become frequent. When the link is heavily lossy ($\log(p) > -1.7$), the three amounts of FEC plotted in the figure become insufficient and all the curves converge to the same point.

5.3 Simulation results

Using the `ns` simulator [12], we simulate a simple scenario where a TCP source is connected to a router via a high speed terrestrial link and where the router is connected to the TCP receiver via a noisy wireless link. The Reno version of TCP [6] is used. The TCP source is fed by an FTP application with an infinite amount of data to send. We add our FEC model to the simulator. The transmission units on the wireless link are supposed to be ATM cells of size

53 bytes. We choose the bandwidth of the wireless link in a way to get a μ equal to 3000 cells/s. RTT is taken equal to 560 ms and the buffer size in the middle router is set to 100 packets. This guarantees that no losses occur in the middle router before the full utilization of μ.

Figures 3 and 7 show the variation of the simulated throughput as a function of the amount of FEC (N/K) and the unit loss probability p respectively. In the first figure, p is set to 0.01. We clearly notice the good match between these results and the analytical ones. The small difference is due to the fact that the expression of the throughput we used does not consider the possibility of a timeout when multiple packet losses appear in the same TCP window [6]. Also, in our analysis, we considered that RTT is always constant which does not hold when the throughput of TCP approaches the available bandwidth.

5.4 The tradeoff between TCP throughput and FEC cost

We compare in this section the bandwidth gained by TCP to that consumed by FEC. Let G be the ratio of these two bandwidths,

$$G = (X_{N,K} - X_{K,K})/(X_{N,K}(N-K)/K) = (1 - X_{K,K}/X_{N,K}) \times (K/(N-K)). \quad (5)$$

This ratio indicates how much beneficial is the addition of FEC. It can be seen as a measure of the overall performance of the system TCP-FEC. A value close to one of G means that we pay for FEC as much as we gain in TCP throughput. A negative value means that the addition of FEC has reduced the performance of TCP instead of improving it.

In Figure 5 we plot G as a function of the amount of FEC for different unit loss probabilities. Again, we take $\mu = 3000$ units/s and $K = 20$. This figure shows that the gain in overall performance is important when the loss probability and the amount of FEC are small. Moreover, with small amounts of FEC, the gain decreases considerably when the loss rate ($L = p$) increases. Now, when the FEC rate increases, the curves converge approximately to the same point with a slightly better gain this time for higher loss probabilities.

5.5 Number of connections and the gain in performance

We notice in Figure 5 that using a small amount of FEC gives the best gain in overall performance. Thus, in order to maintain a high gain, one can use a small amount of FEC and fully utilize the available bandwidth on the wireless link by opening multiple TCP connections. But, in practice we cannot always guarantee that there are enough TCP connections to fully utilize the available bandwidth. A TCP connection must be able to use alone all the available bandwidth. For this reason, FEC has to be added in large amounts so that to make the noisy link clean from the point of view of a single TCP connection even if the achieved gain is not very important.

6 The case of correlated losses

In this section we study the influence of burstiness of transmission unit losses on the efficiency of a FEC scheme. It is clear that when unit losses tend to appear

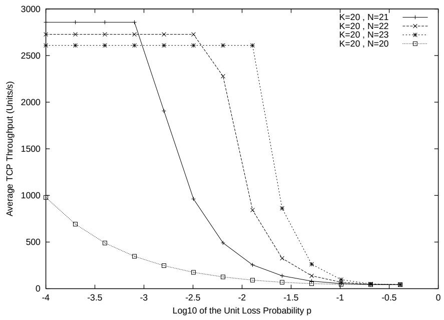


Fig. 6. Model: X vs. p and N/K

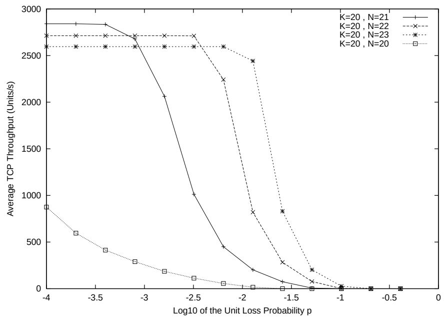


Fig. 7. Simulation: X vs. p and N/K

in bursts, more FEC is needed to clean the link. Packets are hurt by bursts of losses and they require a large number of redundant units per packet (R) to be corrected. But, for the same average loss rate (L), the burstiness of losses reduces the probability that the link passes to the Bad state (p decreases when q increases). This reduces the probability that a TCP packet is hurt by a burst of losses. TCP throughput may then improve and the amount of FEC could be reduced. An analysis is needed to understand these opposite effects of burstiness.

6.1 Performance analysis

Let us calculate T, and hence the throughput of TCP using (3), as a function of the amount of FEC, K, the average loss rate, the burstiness of losses, and μ. Recall that T in this case denotes the average number of TCP packets correctly transmitted between bursts of packet losses.

Let t be the number of good TCP packets between two separate bursts. The minimum value of t is one packet and its expectation is equal to T. Let Y_n be the state of packet n. 0 is the number of the first good TCP packet between the two bursts. Y_n takes two values B (Bad) and G (Good). We have $Y_0 = G$. T can be written as $\sum_{n=0}^{\infty} P(t > n | Y_0 = G) = 1 + \sum_{n=1}^{\infty} P(t > n | Y_0 = G)$.

The computation of T is quite complicated since the TCP packets are not always transmitted back-to-back. Another complication is that $\{Y_n\}$ does not form a Markov chain. Indeed, if we know for example that a packet, say n, is of type B then the probability that packet $n+1$ is of type G also depends on the type of packet $n-1$. If packet $n-1$ were G rather than B, then the *last units of packet* n are more likely to be those that caused its loss. Hence, the probability that packet $n+1$ is B is larger in this case. This motivates us to introduce another random variable which will make the system more "Markovian" and will permit us to write recurrent equations in order to solve for T. We propose to use the state of the last *transmission unit* received, or fictively received, before the nth TCP packet. The knowledge of the state of this unit, denoted by Y_n^{-1} (which may again take the values B and G), fully determines the distribution of the state Y_n of the following TCP packet. We write T as $1 + \alpha P(Y_1^{-1} = G | Y_0 = G) + \beta P(Y_1^{-1} = B | Y_0 = G)$, where $\alpha = \sum_{n=1}^{\infty} P(t > n | Y_0 = G, Y_1^{-1} = G)$ and $\beta = \sum_{n=1}^{\infty} P(t > n | Y_0 = G, Y_1^{-1} = B)$. We shall make the following assumption,

Assumption 1: $P(Y_1^{-1} = G|Y_0 = G) \approx \pi_G$ and $P(Y_1^{-1} = B|Y_0 = G) \approx \pi_B$.

Assumption 1 holds when the time to reach steady state for the Markov chain in the Gilbert model is shorter than the time between the beginning of two consecutive TCP packets (either because the TCP packets are sufficiently large or because they are sufficiently spaced). Assumption 1 also holds when π_B and the loss probability of a whole TCP packet are small. Indeed, we can write, $\pi_G = P(Y_1^{-1} = G|Y_0 = G)P(Y_0 = G) + P(Y_1^{-1} = G|Y_0 = B)P(Y_0 = B) \approx P(Y_1^{-1} = G|Y_0 = G) \cdot 1 + P(Y_1^{-1} = G|Y_0 = B) \cdot 0$.

In view of Assumption 1, the probability that the unit preceding a TCP packet is lost can be considered as independent of the state of the previous packet. It follows that $T = 1 + \alpha\pi_G + \beta\pi_B$, with

$$\alpha = (1 - P(Y_1 = B|Y_1^{-1} = G))(1 + \alpha\pi_G + \beta\pi_B),$$
$$\beta = (1 - P(Y_1 = B|Y_1^{-1} = B))(1 + \alpha\pi_G + \beta\pi_B).$$

This yields, $1/T = \pi_G P(Y_1 = B|Y_1^{-1} = G) + \pi_B P(Y_1 = B|Y_1^{-1} = B)$. The calculation of T is then simplified to the calculation of the probability that a TCP packet is lost given the state of the unit just preceding it. Again, it is difficult to find an explicit expression for this probability. A TCP packet can be lost by a single long burst of unit losses as well as by multiple separate small bursts. To further facilitate the analysis, we assume that bursts of losses at the unit level are separated so that two bursts rarely appear within the same packet. This holds if,

Assumption 2: $(1 - q) \cdot L \cdot N << 1$.

Indeed, a TCP packet is supposed to be lost if it is hurt by a burst of unit losses larger than R. We don't consider the probability that multiple small and separate bursts at the unit level contribute to the loss of the packet. This is possible when the sum of the average lengths of the Good state (L_G) and the Bad state (L_B) is much larger than the packet length N. Using (1) and (2), we get the condition in Assumption 2. If this condition is not satisfied, many bursts may appear within the same packet leading to a higher loss probability than the one we will find, hence to a lower throughput.

Consider first the case $Y_1^{-1} = B$. In view of Assumption 2, packet 1 is lost if its first $R+1$ units are also lost. Thus, $P(Y_1 = B|Y_1^{-1} = B) = q^{R+1}$. For the case $Y_1^{-1} = G$, packet 1 is lost if a burst of losses of length at least $R+1$ units appears in its middle. Thus,
$P(Y_1 = B|Y_1^{-1} = G) = q^R p\left(1 + (1-p) + \cdots + (1-p)^{N-R-1}\right) \simeq K q^R p$.
We used here the approximation $(1 - (1-p)^{N-R}) \simeq Kp$. Substituting p by its value as a function of q and the loss rate L (Equation (2)), we get

$$1/T = q^{N-K} L \left((1-q)K + q\right). \qquad (6)$$

6.2 Analytical results

Using (3) and (6), we plot in Figure 8 the throughput of TCP as a function of burstiness and this is for different amounts of FEC. The burstiness is varied by

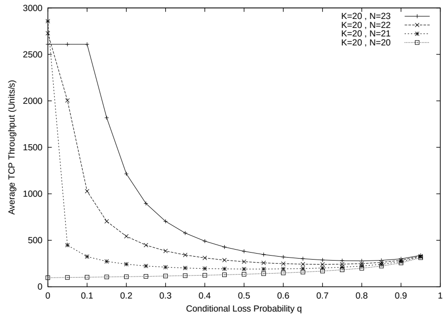


Fig. 8. Model: X vs. q and N/K

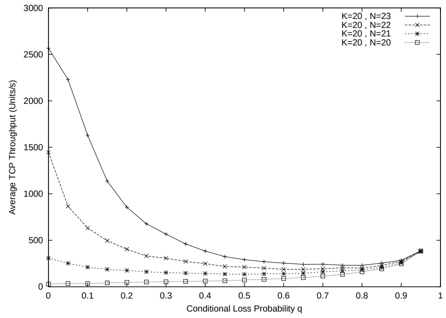


Fig. 9. Simulation: X vs. q and N/K

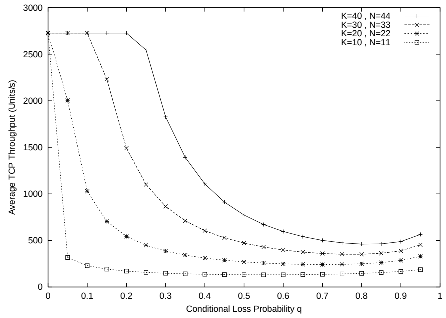


Fig. 10. Model: X vs. q at constant N/K

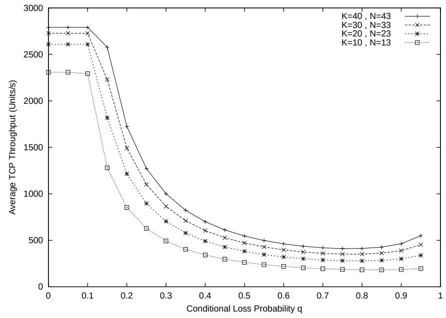


Fig. 11. Model: X vs. q at constant R

varying q which is called the Conditional Loss Probability in the figure. K is set to 20 and the loss rate L to 0.01. The other parameters of the model are taken as in the previous section. We see well that a large amount of FEC gives always the best performance. The difference in performance is important for small bursts (small q). We also see that when burstiness increases, the throughput of TCP decreases drastically for the three FEC schemes we consider in the figure. This is because the length of bursts becomes larger the number of redundant units per packet (R). Here, much FEC must be added to clean the link. But, much FEC reduces the throughput of TCP when burstiness decreases given the bandwidth it consumes. A compromise must be made between much FEC to resist to bursts and a small amount of FEC to give better performance when burstiness decreases. One can think about implementing some kind of adaptive FEC that adjusts the amount of redundancy as a function of the degree of burstiness.

Now, we show in Figure 10 how the block size K can help TCP to resist to burstiness. First, we take the same amount of FEC ($N/K = 11/10$) and we vary K. Increasing K increases the number of redundant units in a TCP packet and thus helps TCP to resist to larger bursts. Better performance is obtained even though the amount of FEC is not changed. The benefit of large packets is also illustrated in Figure 11. In this figure we plot for the same R, the variation of the throughput for different packet sizes. Surprisingly, a large packet size gives better performance than a small one even though the amount of FEC is smaller. From (6), increasing K for the same R decreases T, but this decrease is small compared to the gain we get from the increase in the packet size. In other words, the throughput in terms of packets/s deteriorates when we increase K at a constant R, but it improves in terms of units/s.

6.3 Simulation results

Our intention is to validate by simulation the analytical results we plotted in Figure 8. We consider the same simulation scenario as that in the non-correlation case. The results are plotted in Figure 9. The curves show the same behavior as those in Figure 8. But, we see some mismatch at low burstiness. This is due to our assumption that a packet can only be lost by a single burst not by multiple small and separate bursts of losses at the unit level. As one must expect, the simulation gives a lower throughput in this region given that we are overestimating T.

References

1. E. Altman, K. Avratchenkov, and C. Barakat, "A stochastic model for TCP/IP with stationary random losses", *ACM SIGCOMM*, Sep. 2000.
2. H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. Katz, "A comparison of Mechanisms for Improving TCP Performance over Wireless Links", *ACM SIGCOMM*, Aug. 1996.
3. H. Chaskar, T. V. Lakshman, and U. Madhow, "On the design of interfaces for TCP/IP over wireless", *IEEE MILCOM*, Oct. 1996.
4. A. Chockalingam, M. Zorzi, and R.R. Rao, "Performance of TCP on Wireless Fading Links with Memory", *IEEE ICC*, Jun. 1998.
5. B. R. Elbert, "The Satellite Communication Applications Handbook", *Artech House*, Boston, London, 1997.
6. K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP", *ACM Computer Communication Review*, vol. 26, no. 3, pp. 5-21, Jul. 1996.
7. E.N. Gilbert, "Capacity of a burst-noise channel", *Bell Systems Technical Journal*, Sep. 1960.
8. V. Jacobson, "Congestion avoidance and control", *ACM SIGCOMM*, Aug. 1988.
9. A. Kumar and 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.
10. T.V. Lakshman and 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.
11. M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", *ACM Computer Communication Review*, vol. 27, no. 3, pp. 67-82, Jul. 1997.
12. The LBNL Network Simulator, *ns*, http://www.isi.edu/nsnam/ns/
13. J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", *ACM SIGCOMM*, Sep. 1998.
14. L. Rizzo, "Effective erasure codes for reliable computer communication protocols", *ACM Computer Communication Review*, vol. 27, no. 2, pp. 24-36, Apr. 1997.
15. N. Shacham and P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management", *IEEE INFOCOM*, Jun. 1990.
16. W. Stevens, "TCP Slow-Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", *RFC 2001*, Jan. 1997.