

Modeling Interactions between Link Layer and Transport Layer in Wireless Networks

Carla F. Chiasserini, Michela Meo

Dipartimento di Elettronica, Politecnico di Torino

C.so Duca degli Abruzzi 24, 10129 Torino - Italy

Tel.: +39-011-5644183, Fax: +39-011-5644099

Email: {chiasserini,michela}@polito.it

Abstract—Wireless access to Internet requires that information integrity is preserved while transmitting data over the radio channel. ARQ schemes and TCP are often used as error-control techniques at the link layer and at the transport layer, respectively. In this paper, we study the interactions between an ARQ protocol and TCP when a data traffic connection includes both wired and wireless links. By using standard Markovian techniques, we analyze the impact of different parameters setting of the ARQ scheme and of the radio channel conditions on the TCP performance.

I. INTRODUCTION

Wireless access to Internet is experiencing a great deal of popularity; every cellular telephony system is evolving into the wireless Internet, and products providing Internet access to short-range devices, such as Bluetooth terminals, are being developed. As a consequence, new issues need to be solved at the various layers of the protocol stack. In this paper, we deal with the problem of information integrity while packet traffic is transmitted over end-to-end connections, which include both wired and wireless links.

In order to provide a reliable packet transfer, TCP is often used for end-to-end error control at the transport layer. TCP, however, performs poorly over wireless links since it considers packet losses due to the radio channel as congestion losses and invokes congestion control mechanisms that significantly reduce the throughput [1], [2], [3], [4], [5], [6]. To avoid such performance degradation, link layer solutions can be used to obtain local data reliability and make the wireless link appear to TCP as a more reliable link, although with a longer and variable delay.

We consider an ARQ (Automatic Repeat reQuest) scheme at the link layer, and TCP NewReno at the transport layer. We develop detailed analytical models of the two protocols, which are based on standard Markovian techniques. The TCP model includes the description of an *on-off* traffic source generating data segments; segments are fragmented into several data units at the link layer, where the ARQ model handles their transmission over the radio channel. The packet error process over the wireless link is represented by a two-state Markov chain [7]. The models of the ARQ scheme and TCP are solved by means of a fixed-point procedure. This approach al-

lows us to create a flexible and accurate analytical framework for the joint study of the ARQ scheme and TCP, and to get useful insights in the interaction between the two protocols.

While deriving the system performance, we focus on the 3G (Third Generation) network scenario [8], and on data traffic services. Based on the QoS metrics typical of data traffic, we derive results in terms of throughput and traffic loss probability at the transport layer as the radio channel conditions and the ARQ parameters vary.

II. INTERACTION BETWEEN THE ARQ PROTOCOL AND THE TCP

The system under study is sketched in Fig. 1. A single TCP connection is considered where the receiver resides at a mobile terminal. Before reaching the fixed network access point, hereinafter called Base Station (BS), TCP segments traverse the wired network from which they perceive an average delay D and average loss probability L . At the BS they rely on the Link Layer (LL) protocols to get to the TCP receiver through the wireless link.

At the LL, a TCP segment is divided into N LL data units which are stored in a buffer whose capacity is B TCP segments. The transmission of a LL data unit over the wireless channel takes a time interval equal to Δ . Due to error probability of the radio channel, a data unit transmission may fail. In case of failure, the transmission is repeated; a maximum number M_r of attempts per data unit is allowed. After M_r failed transmissions, a LL data unit is discarded together with all the other units belonging to the same TCP segment. Thus, at the LL a segment can be lost with probability P_o because there is not enough room in the buffer to accommodate all the LL data units in which the segment is divided, or with probability P_r if one of the data units fails M_r transmissions.

The average round-trip time perceived by TCP is given by

$$RTT = 2D + Q + \Delta$$

where D is the delay due to the wired network, Q includes the queueing delay at the LL and the transmission time of a TCP segment over the wireless channel, and Δ takes

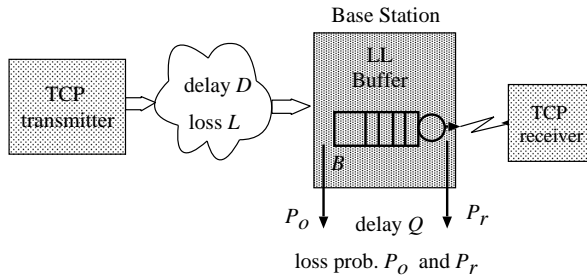


Fig. 1. Sketch of the system under study.

into account the transmission of TCP ACKs over the up-link channel (it is assumed that TCP ACKs do not queue at the LL and are never lost). Moreover, TCP segments are prone, not only to losses in the wired portion of the connection, but also to the two additional causes of losses at the LL. Then, a TCP segment is lost with probability P ,

$$P = 1 - (1 - L)(1 - P_o)(1 - P_r)$$

where L is the probability that the segment is lost in the wired network.

The model we propose is based on the idea of decomposition. The system is divided in two parts which are studied in isolation approximating their reciprocal interaction.

On the one hand, the TCP transmitter is modeled by a continuous-time Markov chain (CTMC); the interaction with the rest of the system is described by means of the segment loss probability and the average round-trip time. These two account for losses and delays introduced by the LL protocols. On the other hand, the LL is studied by developing a discrete-time Markov chain (DTMC); the input traffic is derived from the TCP transmitter model and it is assumed that segments arrive at the LL buffer according to a Poisson process.

Repeatedly solving the two models, a fixed-point procedure provides the steady-state behavior of the system. Similar approaches based on fixed-point approximations were already employed for the analysis of TCP in [9], [10], [11].

The main approximations introduced in the development of the decompositional approach concern the segment arrival process at the LL buffer, and the loss process and delay distribution at the TCP layer. Even if the segment generation process at the transmitter is far from being Poisson, when the network is large the randomness introduced by the network makes the arrival process at the LL buffer closer to Poisson. Correlation between segment losses and the moments of the round-trip time distribution higher than 1 are not taken into account in the TCP model. Instead, mean values only are considered.

A. Model of the TCP transmitter

For the CTMC model, which describes the TCP behavior, we employ the approach proposed in [11].

In order to describe the typical bursty nature of many applications which adopt TCP as transport protocol, we assume that the transmitter has an *on/off* kind of behavior. The time spent in each state is a random variable with negative exponential distribution; the mean values are T_{off} and T_{on} . When in *off* state, the application is idle; while in *on* state the application has data to transmit and sets up a TCP connection. The behavior of an active TCP connection is described quite in detail in our model introducing states in the CTMC which represent those mechanisms of the protocol having a relevant impact on the performance. In the model development, the TCP window size is measured in segments instead of bytes and segments have all the same size, which is equal to the Maximum Segment Size (MSS). The dynamics of the window size growth are observed with a time granularity equal to the average round-trip time.

Let us focus on the CTMC state space. A part from the *off* state of the application, the state space can be partitioned in three sets. The set \mathcal{S}_N collects all the states corresponding to the behavior of the TCP transmitter when no loss occurs,

$$\mathcal{S}_N = \{s = (w, W_t)\}$$

where w denotes the current window size and varies between 1 and the maximum allowed window size W_M . W_t represents the window size growth mode (either slow start or congestion avoidance). $W_t = 1$ stands for congestion avoidance mode; when W_t is larger than 1 the window grows in slow start mode up to the value W_t . For simplicity, in slow start mode we only consider those values of W_t that the window may take in the exponential growth; i.e., the powers of 2.

The set \mathcal{S}_F comprises all the states corresponding to the fast recovery and fast retransmit mechanisms which TCP implements in order to recover a loss,

$$\mathcal{S}_F = \{s = (w)\}$$

where w denotes the window size during fast retransmit and fast recovery procedures.

Finally, states in the set \mathcal{S}_T describe the protocol waiting for a timeout expiration after a loss occurred,

$$\mathcal{S}_T = \{s = (W_t)\}$$

where W_t denotes the value assumed by the threshold after the timeout expiration. The state space is then $\mathcal{S} = \{off\} \cup \mathcal{S}_N \cup \mathcal{S}_F \cup \mathcal{S}_T$.

For the sake of brevity we do not provide transition rates between all states, we instead show only an example of transition rate between states in \mathcal{S}_N . Consider a state $s = (w, W_t) \in \mathcal{S}_N$, the window size is equal to w and the protocol is in slow start mode. If none of the w segments of the window is lost, the window size doubles after roughly a round-trip time. To represent this behavior, the transition rate out of s is set to the inverse of the average round-trip time so that the average time spent in s is equal to the round-trip time; the destination state is $d = (2w, W_t)$ with probability equal to the probability that no segment is lost. The resulting transition rate from s to d is,

$$s = (w, W_t) \in \mathcal{S}_N \rightarrow d = (2w, W_t) \in \mathcal{S}_N : \\ \frac{1}{RTT}(1 - P)^w.$$

Similarly, all the other transition rates are computed.

The steady-state probabilities $\pi_T(s)$'s of the CTMC are derived by using standard techniques. From the $\pi_T(s)$'s the performance of TCP can be computed. In particular, the average number of segments generated in the time unit, given that the application is *on*, is computed by

$$\lambda = \frac{T_{on}}{T_{on} + T_{off}} \sum_{s \in \mathcal{S}_N \cup \mathcal{S}_F} \frac{w}{RTT} \pi_T(s)$$

where w is the window size in state s . This value is used as input in the LL model.

B. The ARQ scheme model

We develop a discrete-time Markov chain (DTMC) model of the ARQ scheme, in which the time is slotted according to the transmission time of a LL data unit. We assume that the transmitter has knowledge of the transmission outcome of a LL data unit transmission right after its completion; as shown by simulation results (not presented here for the lack of room), this assumption does not have a relevant impact on system performance. We also assume that acknowledgments and negative acknowledgments are always correctly received.

In the development of the model the following aspects have to be considered: the arrival process of data segments from the upper protocol layers to the link layer, the transmission buffer, and the radio channel.

The traffic arrival process of the data segments to the LL is assumed to be Poisson with rate equal to λ . Being MSS the data segment size and L_{LL} the LL data unit size, each data segment is fragmented into N LL data units with $N = \lceil MSS/L_{LL} \rceil$. We represent the segment fragmentation by considering that data are stored in the transmission buffer as segments and the service of each

segment consists of N phases. Each service phase corresponds to the service of one LL data unit; depending on the channel status the service can be repeated for a total number of transmissions not larger than M_r . A data segment is removed from the LL queue either when all N LL data units have been successfully delivered to the receiver (i.e., the N service phases have been successfully completed) or one LL data unit is discarded because the maximum number of transmissions has been reached.

The radio channel is modeled as a Gilbert channel [7], with two states, *good* and *bad*, that represent the state of the channel during the transmission time of one LL data unit. The transition probabilities between the two states depend on the steady-state data unit error rate, denoted by ϵ , and on the normalized Doppler frequency.

In order to model the system behavior, the following dynamics have to be described: i) the buffer occupancy, ii) the channel state, iii) the number of transmissions per LL data unit, iv) the sequence number within the TCP segment of the LL data unit currently under transmission.

Accordingly, let the DTMC state be defined by the vector

$$\bar{s} = (b, c, r, u)$$

where

- b is the transmission buffer occupancy in number of TCP segments; b can assume all integer values between 0 and B ;
- c is the channel state, $c \in \{good, bad\}$;
- r is the number of transmissions associated to the LL data unit that is currently transmitted; r can assume all integer values between 1 and M_r ;
- u is the sequence number within the TCP segment of the LL data unit that is currently transmitted; u can assume all integer values between 1 and N .

Let $P(\bar{s}, \bar{d})$ denote the probability that the chain moves in one-step from source state \bar{s} to destination state \bar{d} . Probabilities $P(\bar{s}, \bar{d})$'s are derived from the behavior of the channel and the traffic arrival process, which are independent of the rest of the system. By employing standard techniques, we compute the steady-state probabilities $\pi_L(\bar{s})$, where $\pi_L(\bar{s})$ denotes the steady-state probability of \bar{s} . From values $\pi_L(\bar{s})$'s, we derive the following performance metrics.

Let P_o be the probability that a data segment is lost at the LL due to buffer overflow, and let P_r be the probability that a data segment is lost because for one of the corresponding LL data units the maximum number of transmissions has been reached. Denoting by ρ the average number of segments arriving at the LL buffer during a time slot, i.e., $\rho = \lambda\Delta$, we have

$$P_o = \frac{1}{\rho} \sum_{\bar{s}} P(\bar{s}, \bar{d}) \pi_L(\bar{s}) \quad (1)$$

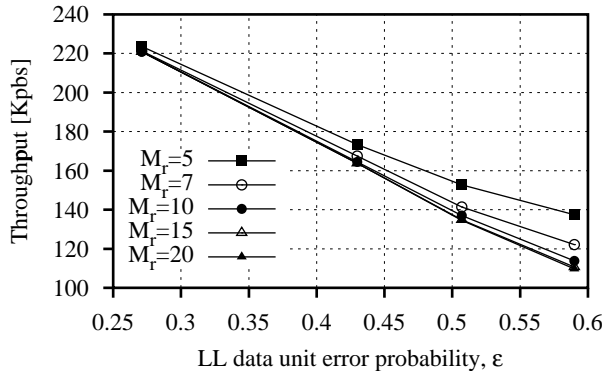


Fig. 2. TCP throughput versus error probability of a LL data unit.

where \mathcal{T}_o is the set of transitions from \bar{s} to \bar{d} , which cause losses due to the buffer overflow; and

$$P_r = \frac{1}{\rho(1 - P_o)} \sum_{\mathcal{T}_r} P(\bar{s}, \bar{d}) \pi_L(\bar{s}) \quad (2)$$

where \mathcal{T}_r is the set of transitions from $\bar{s} = (\cdot, \cdot, M_r, \cdot)$ to $\bar{d} = (\cdot, \cdot, 1, \cdot)$ describing a segment discarding due to the maximum number of allowed transmissions.

Throughput at the transport layer can be computed as

$$X = \lambda(1 - L)(1 - P_o)(1 - P_r). \quad (3)$$

By denoting the mean service time of a data segment successfully transferred over the radio channel by $E[T_s]$, the mean delay that a data segment successfully delivered to the receiver experiences at the LL can be written as

$$E[T] = \frac{E[N_q]}{\rho(1 - P_o)} + E[T_s], \quad (4)$$

where $E[N_q]$ is the mean number of segments in the LL buffer. We have

$$E[N_q] = \sum_{\bar{s} \in \mathcal{S}_q} (b - 1) \cdot \pi_L(\bar{s}) \quad (5)$$

with \mathcal{S}_q being the set of states \bar{s} such that $\bar{s} = (b, \cdot, \cdot, \cdot)$ ($1 < b \leq B$).

III. NUMERICAL RESULTS

Results are obtained in the context of 3G wireless systems [8]. Thus, while deriving our results we consider that Δ is equal to 10 ms and the peak data rate is equal to 384 Kbps. We have $L_{LL} = 384 \text{ Kbps} \cdot 0.01 \text{ s} = 480$ bytes. We consider a MSS equal to 1,000 bytes, so that each TCP segment is divided into $N = 3$ LL data units. The TCP maximum window size is assumed to

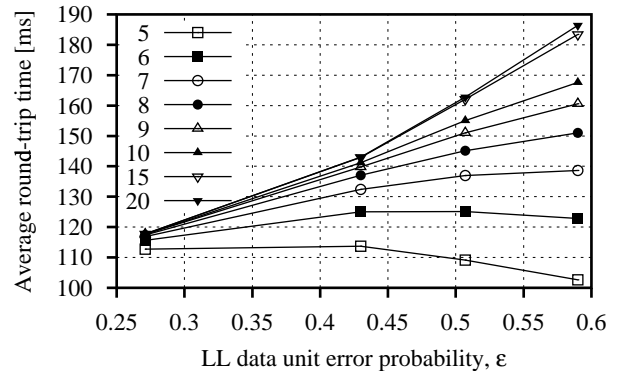


Fig. 3. Average round-trip time versus error probability of a LL data unit.

be equal to 20 segments, and the average *on* and *off* periods of the application to be equal to 500 ms. We assume that the average delay D perceived by segments in traversing the network is equal to 5 ms, while we neglect losses due to the wired network. In showing the system performance as channel conditions change, we consider a constant normalized Doppler frequency equal to 0.2 and a varying steady-state error rate.

In Fig. 2 the TCP throughput is shown versus the probability ϵ that a LL data unit transmission fails, when the buffer capacity is equal to 10 segments and for different values of M_r . Obviously, as the channel conditions worsen, the TCP performance deteriorates. Cases with smaller values of M_r , have a slightly larger throughput. This is due to the smaller queueing delay introduced by the LL buffer when a larger number of segments is discarded because M_r is reached; small delays keep the round-trip time small and make the TCP behavior be more aggressive.

This behavior is confirmed by results in Fig. 3, which shows the average round-trip time at the transport layer under the same system configuration as considered in Fig. 2. Observe how remarkable the difference in the round-trip time is for different values of M_r , when error probability ϵ is large.

In Fig. 4 the segment loss probabilities P_o and P_r due to the link layer are plotted. As expected, P_r decreases with increased M_r while P_o exhibits the opposite behavior. In fact, lower values of P_r imply higher buffer occupancy and, hence, larger buffer overflow probabilities. This plot suggests that, fixed the required loss probability P_r , an optimal value for M_r can be found as the radio channel conditions vary. For instance, fixed the desired P_r to 10^{-3} , a maximum number of transmissions per LL data unit between 7 and 10 can be selected when ϵ is less than 0.35, while for $0.35 \leq \epsilon \leq 0.5$ a value between 10 and 15 should be chosen. Optimizing the maximum

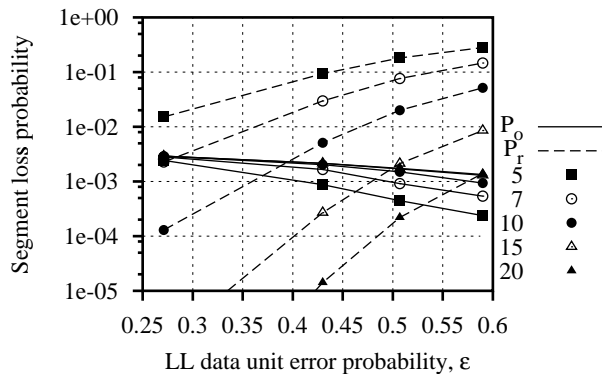


Fig. 4. TCP segment loss probabilities P_o and P_r versus error probability of a LL data unit.

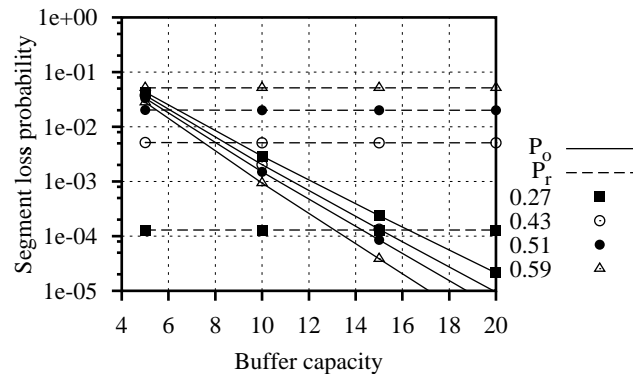


Fig. 5. TCP segment loss probabilities P_o and P_r versus buffer capacity.

number of transmissions per data unit reduces the energy consumption in receiving mode at the mobile terminal, as well as the average round trip time (see Fig. 3).

In Fig. 5, P_o and P_r are shown as functions of the buffer capacity when $M_r = 10$. For low values of P_r , loss probability is dominated by buffer overflow; in these cases, loss probability can be significantly improved by enlarging the buffer capacity.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we considered data traffic transmitted over end-to-end connections including both wired and wireless links. We studied the interactions between an ARQ scheme and TCP by developing analytical models based on standard Markovian techniques. We studied the system performance as the channel conditions varied and for different values of the maximum number of transmissions per data unit at the link layer in the context of a 3G network scenario.

Further research will focus on the study of the TCP performance when users mobility is taken into account and on the use of a scheme that combines FEC and ARQ techniques.

REFERENCES

- [1] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ," in *Proc. of ICC*, 1999, pp. 1212–1216.
- [2] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz, "A comparison of mechanisms for improving TCP performance," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 756–769, December 1997.
- [3] J.W.K. Wong and V.C.M. Leung, "Improving end-to-end performance of TCP using link-layer retransmissions over mobile internetworks," in *Proc. of ICC*, 1999, pp. 324–328.
- [4] Y. Bai, A.T. Ogielski, and G. Wu, "Interactions of TCP and radio link ARQ protocol," in *Proc. of IEEE VTC*, 1999, pp. 1710–1714.
- [5] K.-Y. Wang and S.K. Tripathi, "Mobile-end transport protocol: an alternative to TCP/IP over wireless links," in *Proc. of INFOCOM*, 1998, pp. 1046–1053.

- [6] Chaskar N.M., T.V. Lakshman, and U. Madhow, "TCP over wireless with link level error control: analysis and design methodology," *IEEE/ACM Transactions on Networking*, vol. 7, no. 5, pp. 605–615, October 1999.
- [7] M. Zorzi, R.R. Rao, and L.B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 2, pp. 445–55, May 1997.
- [8] Third Generation Partnership Project (3GPP), "Technical specifications," March 2000, <http://www.3gpp.org>.
- [9] C. Casetti and M. Meo, "A new approach to model the stationary behavior of TCP connections," in *Proc. of INFOCOM 2000*, Tel Aviv, Israel, March 2000.
- [10] T. Bu and D. Towsley, "Fixed point approximation for TCP behavior in an AQM network," in *Proc. of ACM SIGMETRICS 2001*, Cambridge, Massachusetts, USA, June 2001.
- [11] C. Casetti and M. Meo, "Modeling the stationary behavior of TCP Reno connections," in *Proc. of International Workshop on QoS in Multiservice IP Networks*, Rome, Italy, January 2001.