Influence of UMTS Radio Links on TCP: Time Delays and Packet Reordering

Niels Möller, Karl Henrik Johansson, Estelle Jaquot

Abstract— **A lossy radio link may reduce TCP performance considerably. Radio blocks which are damaged and retransmitted by the radio link layer cause random delays on the IP-level. This paper describes an analytic model for a radio link used in UMTS. The delay experienced by IP-packets traversing the link is analyzed, and the deteriorating effects on TCP are described. It is shown that even if these delays and their variations are not extraordinary, they may still trigger fast retransmits or timeouts, and thereby reduce the throughput.**

Keywords— **TCP over UMTS, Congestion control, Wireless communication, Control theory.**

I. Introduction

A fundamental assumption of the TCP protocol is that packet losses indicate congestion on the network [1], [2]. This is a problem when using TCP over wireless links, because a noisy radio transmission may erroneously indicate congestion and thereby reducing the TCP sending rate, e.g., [3], [4], [5], [6]. A partial solution is the introduction of link-level retransmission, such as Automatic Repeat Request (ARQ), which is supposed to hide radio link losses from TCP/IP. The IP packets still, however, experience random delays or even reorderings when they are transmitted across such a link. How to deal with these problems on the TCP level is the topic of intensive research, e.g., [7], [8], [6], [9]. In general, it would be desirable for the end-to-end TCP to distinguish between packets that are delayed due to the retransmissions on the radio link and packets that are lost due to congestion on some wired link. Attempts to robustify TCP include the Eifel algorithm [7]. A TCP modification for wireless links is described in [6]. The deteriorating effect on TCP of channel type switching for resource allocation in WCDMA was studied in [9]. Despite recent progress, the influence of the radio link properties on TCP is far from fully understood.

The main contribution of the paper is to derive an analytic model for a radio link used in UMTS. The IP-packet delays are analyzed. The delays are crucial for the understanding of the behavior of TCP over wireless links, as the TCP algorithms depend on them in several ways: e.g., the delays influence the estimated end-to-end round trip time, which in turn influences other TCP parameters. If a packet is delayed long enough to lag three packets behind in the sequence, it triggers the TCP fast retransmit mechanism. This can reduce the throughput considerably. An alternative is to sort the packets (never passing them on out of sequence), by delaying fully reassembled packets when necessary. We show that out-of-order packets should be avoided and derive the probability for entering fast transmit. This supports conclusions in previous work [8], [9]. Another option is to modify TCP in order to make it more robust to packet reordering [10].

The outline of the paper is as follows. In Section II we analyze the delays caused by the UMTS link layer and derive the distribution of the packet delays. The influence of the radio link on TCP is discussed in Section III. The probability that a packet triggers fast retransmit is estimated and numerical values for the radio link properties are calculated for various packet sizes. After discussion of the results in Section IV, a simulation example is presented in Section V. The final section discusses results and conclusions, and suggests topics for future work.

This work was supported by the Swedish Research Council.

N. Möller, K. H. Johansson, and E. Jaquot are with the Department of Signals, Sensors and Systems, Royal Institute of Technology, Stockholm, Sweden. Email: `{niels,kallej,ejacquot}@s3.kth.se`

II. Radio Link Model

A stochastic radio link model is derived in this section.

A. Radio model

Data is transmitted over the radio link as a sequence of *radio blocks* (RB). One radio block corresponds to a time slot or *transmission time interval* (TTI) of 10 or 20 ms. Depending on the bandwidth (typically from 64 kbit/s to 384 kbit/s) the size of a RB can vary from 160 octets to 960 octets. The radio link round trip time is denoted RTT. A typical value is RTT $= 5$TTI, i.e., 50 or 100 ms.

The transmission of the radio blocks is lossy. Let p denote the probability that a radio block is damaged. The power of the radio transmitters are controlled, so that the loss probability stays fairly constant. The reference block error rate is a deployment trade-off between channel quality and the number of required base stations. For UMTS the reference block error rate is often chosen to be about

10%, see [11]. In the following, we thus assume $p = 0.1$. For simplicity we also assume that consecutive blocks are dropped independently from each other. Simplistic as it may be, this model should capture at least two important classes of disturbances: errors caused by stationary background noise, with time constants smaller than TTI, and disturbances that the power control is able to compensate for, e.g. slowly moving terminals and obstacles. A model for correlated errors could be incorporated in the derivation below, and is subject for future studies.

B. Link level retransmissions

Radio blocks that are received successfully are acknowledged by the receiver. Damaged radio blocks, for which no acknowledgment is received, are resent. A radio block can be resent over and over again, until it is finally received correctly. Thus, if we treat the radio link and the retransmission mechanism as one unit, we have an essentially loss-less transmission of radio blocks, but with random delays and, possibly, reorderings of the blocks that needed retransmission.

We can calculate the distribution of the number of link level retransmissions that are experienced by a single radio block transmitted across the channel. The probability that there are exactly j retransmissions is $(1-p)p^j$. The expected number of retransmissions is then given by

$$E(\text{retransmissions}) = \frac{p}{1-p} \tag{1}$$

The distribution function is denoted

$$S_j = P(\text{at most } j \text{ retransmissions of a RB}) = 1 - p^{j+1} \tag{2}$$

C. IP packet reassembly

Small IP packets can fit in a single RB (it is even possible that several small IP packets can share a single RB), but larger IP packets have to be split over several radio blocks. The latter radio blocks can be damaged and retransmitted independently of each other. When all the pieces of an IP packet have been received correctly, the packet can be reassembled and passed on. Note that the loss of a single RB can cause a large IP packet to be delayed.

The receiver can handle delayed packets in several ways. If it passes each IP packet on as soon as it has been reassembled, independently of any other packets that are only partially reassembled, the sequence of IP packets is reordered. An alternative is to sort the packets, never passing them on out of sequence, by delaying fully reassembled packets when necessary. In general, out-of-order delivery should be avoided and thus traded for some extra delays, cf., [8], [9].

D. IP packet delay

Let the size of the IP packet be n radio blocks, $n \geq 1$. As an IP packet is usually smaller than 1500 octets (the maximum packet size for IP over Ethernet), and the radio blocks are 160 octets or larger, we will usually have $n \leq 10$. The minimum delay of the packet passing over the radio link, if none of the radio blocks are damaged, is

$$d_{\min} = \text{RTT}/2 + (n-1)\text{TTI}$$

Let the stochastic variable X denote the number of link-level retransmissions that is experienced by an IP packet. Then, the total delay for the packet is given by

$$d = (X + 1/2)\text{RTT} + (n-1)\text{TTI}$$

We start the analysis by considering the number of damaged blocks. So consider an IP packet of n radio blocks, and let k be the number of blocks that are damaged during the initial transmission of the packet. The probability that exactly k blocks are damaged is

$$D_k = P(k \text{ blocks damaged}) = \binom{n}{k} p^k (1-p)^{n-k}$$

If $k = 0$, there are no retransmissions, and there is no additional delay. If $k = 1$, then there is definitely one retransmission plus some number of additional retransmissions given by equation (2).

What about the general case? One could compute properties of the distribution of X by conditionalizing on k, but it turns out that it is simpler to exploit the independence of the radio block errors. We assume that retransmissions for distinct radio blocks take place in parallel, independently of each other. We also ignore any queueing delays. For this model to resemble reality, it is important that blocks that are resent are given a higher priority in the radio block send queue than blocks that are sent for the first time. One should also note that in the real system, the retransmission of an early block that is damaged can start some time before the retransmission of later blocks, resulting in slightly shorter delays than in our model.

Under this model, the number of retransmissions needed for block i is an independent stochastic variable with the distribution given in equation (2). The number of retransmissions needed for an IP-packet is the maximum number of retransmissions that were needed for any of the n corresponding radio blocks. We get

$$\begin{aligned} F_j &= P(X \leq j) \\ &= P(\text{at most } j \text{ retransmissions for all the } n \text{ RBs}) \\ &= P(\text{at most } j \text{ retransmissions for one RB})^n \\ &= S_j^n = (1 - p^{j+1})^n \end{aligned} \tag{3}$$

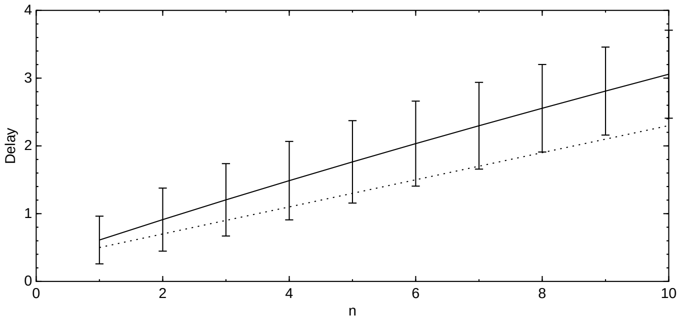

Fig. 1. Packet delay in units of RTT. Minimum delay, $d_{\min}$ (dashed line), and expected delay $E(d)$ (solid). The error bars show the standard deviation $\sigma(d)$.

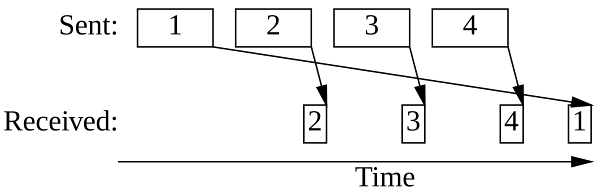

Fig. 2. Reorder triggers fast retransmit

From this expression, we can compute the expected number of retransmissions, as well as the standard deviation. The results for $p = 0.1$, RTT $= 100$ ms and TTI $= 20$ ms are shown in Figure 1.

We see that the expected delay grows almost linearly with n, at least in the range in which we are interested. The standard deviation, on the other hand, grows slower and flattens out at about 0.65RTT for $n > 7$. Due to the parallel retransmissions of damaged radio blocks, one extra round of retransmissions can repair several damages.

III. Influence of Radio Link on TCP

Radio link retransmissions may trigger fast retransmits or timeouts, and thereby reduce the throughput.

A. Fast retransmit

If a packet in a TCP stream is delayed so much that it lags behind three other packets (see Figure 2), the acknowledgements for the next three packets are "duplicated", and the sender will go into fast retransmit/fast recovery mode. That means that from the point of view of TCP, even if the packet is not really dropped, only delayed, it could just as well have been dropped.

From the delay distribution in the previous section, we estimate the probability that a packet triggers the TCP fast retransmit mechanism. Obviously, the sequence of events "packet number i triggers fast retransmit" are not independent, but as an event happens when one packet is late and the next three are early, we can look at the probability that a packet at a fixed position in the stream triggers fast retransmit. If we then use a model that applies the resulting probability independently to all packets, reality should differ from the model only by spreading out the events slightly more.

We treat the number of retransmissions needed for individual packets as independent stochastic variables X_i, with a distribution from equation (3). The delay of the ith packet is denoted

$$d_i = \text{RTT}(1/2 + X_i) + (n-1)\text{TTI}$$

We consider a stream of equal sized packets, so that n is fixed. Packet i is then sent at time in TTI. Fast retransmit is triggered by packet 1 if

$$\begin{cases} d_1 > d_2 + n\ \text{TTI} \\ d_1 > d_3 + 2n\ \text{TTI} \\ d_1 > d_4 + 3n\ \text{TTI} \end{cases}$$

The probabilities depend hence only on $r = n\ \text{TTI}/\text{RTT}$. We get the following inequalities for X_i:

$$\begin{cases} X_1 > X_2 + r \\ X_1 > X_3 + 2r \\ X_1 > X_4 + 3r \end{cases}$$

We conditionalize on X_1, and treat the three inequalities that must be satisfied as independent events. The probability that the first packet triggers fast retransmit can be expressed explicitly using the distribution in equation (3):

$$\begin{aligned} P_{FR} &= \sum_{j=0}^{\infty} P(X_1 = j) \prod_{k=1}^{3} P(X_{k+1} < j - kr) \\ &= \sum_{j=0}^{\infty} (F_j - F_{j-1}) F_{\lfloor j-r \rfloor} F_{\lfloor j-2r \rfloor} F_{\lfloor j-3r \rfloor} \end{aligned}$$

The infinite sum over j converges very quickly: for $p = 0.1$ only the terms for $j = 1$, 2, and 3 are significant. The explicit expression for the fast retransmit probability P_{FR} will help us understand the TCP behavior. It is presented in Figure 3 as a function of n. The numerical value of P_{FR}, in percent, is also given in the table below. Here, $p = 0.1$, RTT $= 100$ ms and TTI $= 20$ ms. For $n \geq 7$, $P_{FR} < 0.01\%$.

n	1	2	3	4	5	6	7
P_{FR}	7.53	1.58	1.65	0.26	0.29	0.03	0.00

The probability decreases quickly with increasing n, but not quite monotonically. The non-monotonicity seems to be an artifact of the discreteness of the packet delay distribution.

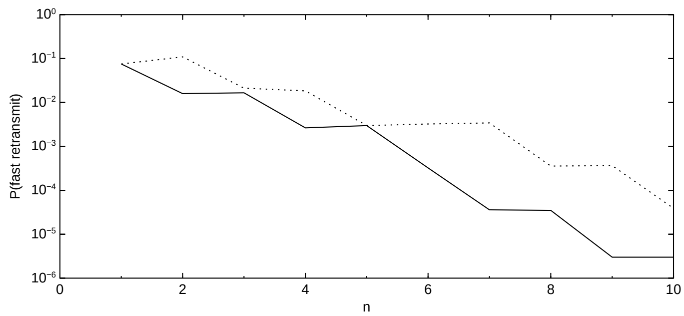

Fig. 3. The fast retransmit probability P_{FR} as a function of n. The solid line corresponds to RTT $= 5$TTI $= 100$ ms. The dashed line corresponds to a slightly larger link round trip time, RTT $= 7$TTI $= 140$ ms.

IV. DISCUSSION

We draw the following conclusions from the results:

- The delays, and in particular the delay variations, are not really extraordinary. They are of the same order of magnitude as the radio link round trip time, and thus not much larger than the end-to-end round trip time. The maximum delays are of the same order of magnitude as the round trip time of an old fashioned modem dial-up. For example, it seems unlikely that it should make the round trip time estimation of TCP break down.
- It seems crucial that retransmitted radio blocks are given a high priority in the radio block send queues. Any extra delays introduced for the retransmitted blocks will make the overall delay, and thus TCP, more sensitive to the block error rate.
- Small packets, fitting in five radio blocks or less, will get reordered (with a probability of 0.3% or higher). For the fastest links, all packets are in this class. If the traffic consists of TCP streams with small packets, one should consider sorting packets to avoid out of order delivery. For TCP, the probabilities in the above table can be thought of as IP packet drop probabilities.
- From the dependence on the ratio r, or from Figure 3, we also see that an increased link round trip time (i.e. slower feedback across the link) implies a higher probability for fast retransmissions, in the same way that decreased packet sizes do. Making the link round trip faster is one way to decrease the fast retransmit probabilities.

V. SIMULATIONS

It has been reported that packet reorderings that trigger TCP fast retransmit are a major performance problem for TCP over UMTS [11]. We confirm this below by simulating a simple example, where the radio link performance is based on the previous derivations.

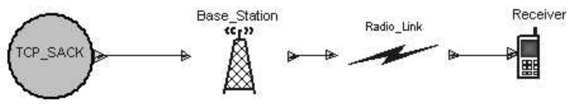

Fig. 4. Modelica example of TCP over a wireless link

A. Model

Consider the simulation model in Figure 4. It is implemented in Modelica [12], which is an object-oriented language for modeling dynamical systems. Modelica was recently shown to provide an efficient tool for simulation of communication networks [13]. The developed library is based on a recent hybrid model [14]. The hybrid model is based on average rates but takes packet drops and rate adjustments due to congestion control into account. The motivation for this model is to capture the network behavior on a time scale in between packet models and flow models. Studies have shown that the hybrid model is able to capture many important network phenomena, see [14], [15], [13].

The example in Figure 4 consists of a TCP SACK connection over a wireless link. We use a bandwidth of 128 kbit/s over the radio link, a propagation delay of 550 ms in the wired part of the network (to get a larger bandwidth-delay product), and a packet size of 640 octets or 2 radio blocks. The sender's maximum window size was set to 14 packets, or about 9000 octets.

B. Results

Figures 5 and 6 show the window size and accumulated throughput for two cases: with no packet reorderings (corresponding to a radio link layer that delays packets, in order to avoid out-of-order delivery), and with reorderings that trigger TCP fast retransmit with the probability 1.58%, according to the table in Section III.

When fast retransmit is triggered at the TCP sender, it decreases the window size and consequently the sending rate. Note that fast retransmit is triggered several times when packet reordering is accepted. Without reordering the window size is equal to its maximum.

Note that the throughput will *not* be reduced if the maximum window size is sufficiently large, as the buffer before the radio link then smoothes out the sending rate and keeps the radio link saturated at all times. In our simulations, however, the throughput is decreased because we enforce a limit on the window size. In this case, the radio link is not saturated, there's no significant buffering before

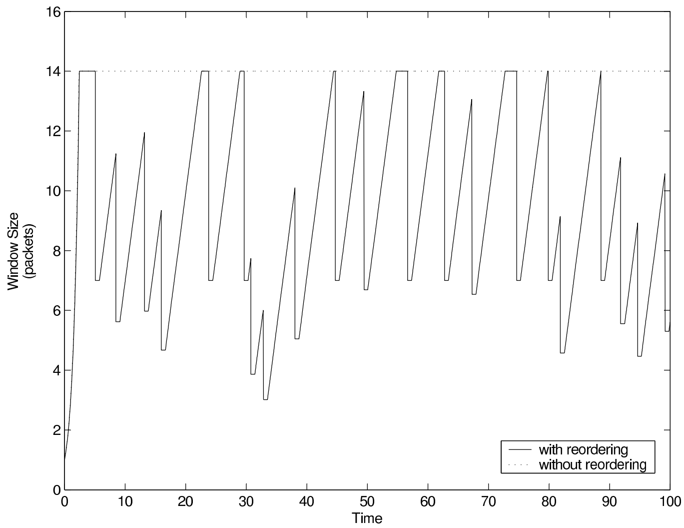

Fig. 5. Window size for the example in Figure 4. Without reordering (dashed line), the window size is fixed at its maximum. If packet reordering is allowed (solid line), fast retransmit is triggered at several instances and consequently the window size evolve according to the TCP fast recovery and congestion avoidance algorithms.

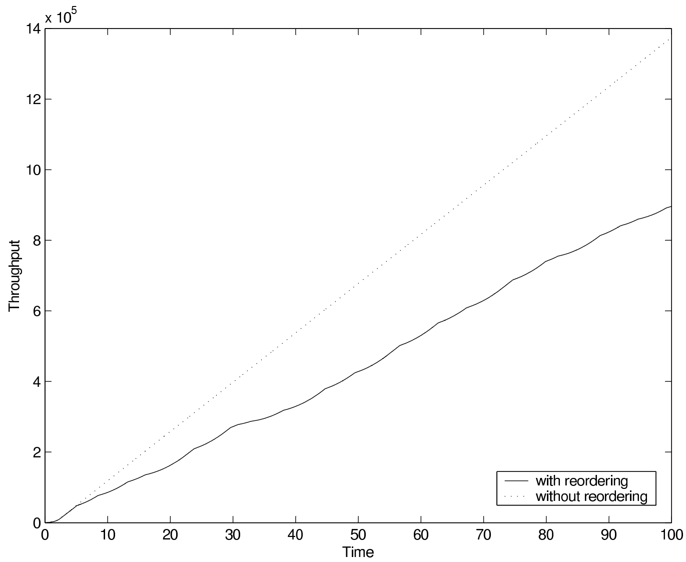

Fig. 6. Accumulated throughput for the example in Figure 4. The reduced window sizes for the packet reordering case lead to reduced throughput.

the link, and therefore the radio link delays yield not only fluctuations in the sending rate, but also significantly decreased throughput.

The simulation supports the conclusion that the packet reordering should be avoided. It is in most cases better to get an extra delay at the radio link by waiting until dropped radio blocks are retransmitted, than to allow the link to reorder packets without delay. The extra delay can either be handled by existing TCP implementations, or by proposed alternatives in the literature, cf., [6], [7].

VI. CONCLUSIONS

Lossy radio links reduce TCP performance considerably. Using a stochastic model for the radio link, we have quantified properties such as the link delay distribution and the probability that the TCP fast retransmit mechanism is triggered. These help us to both understand and simulate the behavior of networking protocols over wireless links. Our results can be used to guide the design of link-layers for wireless links, as well as for evaluating proposed changes to TCP.

Ongoing and future work includes `ns-2` validation and a more detailed error model for the radio link. Another interesting area is to investigate how the RTT estimation in TCP interacts with the random delays introduced by the radio link.

Acknowledgment

We would like to thank Anders Dahlén, Håkan Hjalmarsson, and Gunnar Karlsson for helpful discussions.

REFERENCES

[1] V. Jacobson, "Congestion avoidance and control," in *Proc. of SIGCOMM*, 1988, vol. 18.4, pp. 314–329.

[2] S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review*, vol. 24, no. 5, pp. 10–23, 1994.

[3] A. DeSimone, M. Chuah, and O. Yue, "Throughput performance of transport-layer protocols over wireless lans," in *IEEE Globecom'93*, 1993, pp. 542–549.

[4] G. Xylomenos and G. C. Polyzos, "Internet protocol performance over networks with wireless links," *IEEE Network*, vol. 13, no. 4, pp. 55–63, 1999.

[5] R. Zorzi, A. Chockalingam, and R. R. Rao, "Throughput analysis of tcp on channels with memory," *IEEE J-SAC*, vol. 18, no. 7, pp. 1289–1300, 2000.

[6] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: bandwidth estimation for enhanced transport over wireless links," in *MobiCom*, Rome, Italy, 2001.

[7] R. Ludwig and R. H. Katz, "The eifel algorithm: Making tcp robust against spurious retransmissions," *ACM Computer Communication Review*, vol. 30, no. 1, January 2000.

[8] A. Canton and T. Chahed, "End-to-end reliability in umts: Tcp over arq," in *Globecom 2001*, 2001.

[9] P. M. Garrosa, "Interactions between tcp and channel type switching in wcdma," M.S. thesis, Chalmers University of Technology, 2002.

[10] Ethan Blanton and Mark Allman, "On making tcp more robust to packet reordering,".

[11] A. Dahlén and P. Ernström, "Tcp over umts," in *Radiovetenskap och Kommunikation 02*, 2002, RVK.

[12] *Modelica*, `http://www.modelica.org`.

[13] D. Färnqvist, K. Strandemar, K. H. Johansson, and J. P. Hespanha, "Hybrid modeling of communication networks using Modelica,"

in *2nd International Modelica Conference*, Munich, Germany, 2002.

[14] J. P. Hespanha, S. Bohacek, K. Obraczka, and J. Lee, "Hybrid modeling of TCP congestion control," in *Hybrid Systems: Computation and Control*, M. Di Benedetto and A. Sangiovanni-Vincentelli, Eds., vol. 2034 of *Lecture Notes in Computer Science*, pp. 291–304. Springer-Verlag, Berlin, Germany, 2001.

[15] S. Bohacek, J. P. Hespanha, J. Lee, and K. Obraczka, "Analysis of a TCP hybrid model," in *Proc. of the 39th Annual Allerton Conference on Communication, Control, and Computing*, 2001.