Optimum Number of RLC Retransmissions for Best TCP Performance in UTRAN

Olivier De Mey, Laurent Schumacher, Xavier Dubois
`{ode,lsc,xdubois}@info.fundp.ac.be`
Computer Science Institute, The University of Namur (FUNDP)
Rue Grandgagnage 21, B-5000 Namur, Belgium

***Abstract*— As far as transfer reliability is concerned, the wireless segment of UTRAN is the weakest part of an Internet session. To overcome this imperfection, the RLC protocol features a link-level retransmission mechanism. In this paper, we show by analytical means and computer simulations that for a given Block Error Rate ($BLER$), the maximum number of RLC transmissions has a very important impact on TCP performance, both Round Trip Time (RTT) and throughput. We extend the link level Automatic Repeat Request model proposed by Peisa et al. (2001) to be able to set a maximum number of RLC transmissions, and with the help of Padhye's TCP throughput model (2000), we determine the corresponding throughput analytically. We also show that, knowing the current $BLER$, a network operator may tune the maximum number of retransmissions either to reduce the RTT or to increase the throughput.**

I. INTRODUCTION

RLC recovery is a local retransmission mechanism designed to partially overcome the imperfection of the wireless segment in the UTRAN [1]. It is based on the buffering of the Protocol Data Units (PDU) at the transmit end of the wireless link, coupled with a local retransmission mechanism between the User Equipment (UE) and the Radio Network Controller (RNC). Using the RLC Acknowledged Mode (AM), this mechanism lets the RNC know whether a RLC PDU has been received correctly or not. In case of loss or erroneous transmission of a PDU, the RNC has the ability to retransmit it locally, making TCP unaware of the loss from an end-to-end perspective.

The parameters of the RLC (buffer size, number of retransmissions) are of uttermost importance to achieve good performance. Usually, these are operator-specific. In [2], the impact of the buffer size has been investigated. It has been shown that a small RLC buffer size has negative effects on the TCP throughput, though the simulation parameters supposed an unlimited number of RLC retransmissions.

The purpose of the current paper is to address the setting of the number of local retransmissions. Indeed, a limited number of retransmissions is likely to lead to a TCP packet loss at the receiver's side due to the unability to recover from the PDU loss. As a result, the TCP sender would reduce its congestion window, and the transmission throughput would decrease, despite the loss could possibly have been recovered with a single additional PDU retransmission. On the other hand, too many retransmissions would lead to a TCP timeout, and therefore the retransmission of PDUs from a duplicated TCP packet. Hence, determining the optimum number of local retransmissions from the circumstances, e.g. channel quality as measured by the PDU error rate, is a meaningful task.

In this paper, we will show to which extend the maximum number of RLC retransmissions is determining the TCP performance in terms of RTT and throughput. The study will be performed by means of analytical developments, validated by computer simulations.

II. DESCRIPTION OF THE INVESTIGATED SCENARIO

We are considering the scenario illustrated on Fig. 1: one user is downloading a file using FTP from his/her UE, where the wireless link is affected by errors measured by the RLC PDU Block Error Rate ($BLER$). Errors on the wireless link are assumed to be uncorrelated. On the other hand, the rest of the connection, from the server down to the RNC, is regarded as error-free.

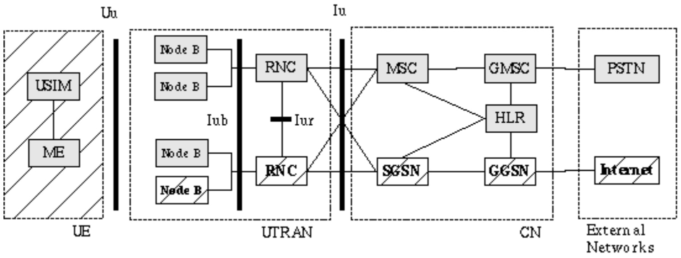

Fig. 1. Scenario

We assume that every TCP segment is split into N RLC PDUs,

$$N = \left\lceil \frac{\text{TCP Segment size}}{\text{RLC PDU size}} \right\rceil , \qquad (1)$$

with $\lceil x \rceil$ standing for the smallest integer greater than or equal to x. Moreover, as in [3], it is further assumed that a simple TCP segment completely fills in the transmitter buffer with its N RLC PDUs.

The RLC protocol is fully described in [1]. Our focus here is on its AM mode. If a RLC PDU has been either lost or corrupted between the Node B and the UE, the receiver signals the incident through a status report. In the absence of such a feedback, the transmitter can poll the receiver at the expiry of a polling timer. The RLC can retransmit a missing PDU ($maxDAT - 1$) times, where $maxDAT$ is the total number of transmissions of a RLC PDU: the first transmission itself, and the maximum number of retransmissions. In case the PDU has not been acknowledged by the UE to the RNC after $maxDAT$ transmissions, the whole TCP packet is then dropped. Already acknowledged PDUs are dropped as well, and a TCP retransmission is triggered.

In the following sections, we will investigate the influence of $BLER$ and $maxDAT$ on TCP performance, first analytically, then with the help of computer simulations.

III. CLOSED-FORM EXPRESSION OF THE THROUGHPUT

A TCP segment is successfully transmitted if and only if each of its RLC PDU reaches the UE correctly. Knowing that a RLC PDU may be transmitted at most $maxDAT$ times, the probability of a successfull RLC PDU transmission is given by

$$\begin{aligned} & Prob\,(\text{Successful PDU transmission}) \\ &= \; 1 - BLER^{(maxDAT)}. \end{aligned} \tag{2}$$

If the TCP segment is split into N RLC PDUs, the probability of a successfull TCP transmission p writes

$$\begin{aligned} p &= Prob\,(\text{Successful TCP transmission}) \\ &= \left[1 - BLER^{(maxDAT)}\right]^N . \end{aligned} \tag{3}$$

In order to compute the throughput of the TCP connection, we need to estimate the Round Trip Time (RTT) of the session. An analytical expression of the RTT applicable to UMTS scenarios has been derived in [3]. The authors split the whole RTT into a contribution due to the RLC, RTT_{RLC}, and the transmission delay in the wired network, RTT_{Wired}, such that the whole RTT writes

$$RTT_{TCP} = RTT_{RLC} + RTT_{Wired}. \tag{4}$$

In the following, RTT_{Wired} values will be collected from the trace files of the computer simulations.

In [3], a closed-form expression of RTT_{RLC} is proposed:

$$\begin{aligned} & RTT_{RLC}\,(BLER) \\ &= \; RTT_{RLC} \sum_i iP_i + \frac{RTT_{RLC}}{2} \sum_i P_i \\ &\quad + T_{Polling} \left(\frac{e}{1-e}\right) \sum_i iP_i \\ &\quad + TTI \sum_i \sum_{j=0}^{i} \left\lfloor \frac{BLER^j N}{r} \right\rfloor P_i \end{aligned} \tag{5}$$

where variables e, P_i, $T_{Polling}$, TTI and r respectively stand for the probability of unsuccessful transmission of a status report, set as

$$e = \left[1 - (1 - BLER)^2\right], \tag{6}$$

the probability of needing exactly i retransmissions, the polling timer delay, the Transmission Time Interval, and the number of PDUs transmitted within one TTI. $\lfloor x \rfloor$ represents the largest integer less than or equal to x.

Relation (5) does unfortunately not take into account the limitation of the maximum number of retransmissions. Writing that the probability of the successfull transmission of N RLC PDUs after i attempts as

$$P_i^N = \left[1 - BLER^{(i+1)}\right]^N - \left[1 - BLER^i\right]^N \tag{7}$$

the expression of RTT_{RLC} becomes

$$\begin{aligned} & RTT_{RLC}\,(BLER, maxDAT) \\ \approx \;& \frac{RTT_{RLC}}{2} \left(2 - e^{maxDAT}\right) \sum_{i=1}^{maxDAT} iP_i^N \\ & + \frac{RTT_{RLC}}{2} \sum_{i=1}^{maxDAT} P_i^N \\ & + \; T_{Polling} \\ & \quad \left\{ \begin{array}{l} \frac{e\left[1-e^{(maxDAT-1)}\right]}{1-e} \\ -(maxDAT-1)\, e^{maxDAT} \end{array} \right\} \\ & \quad \sum_{i=1}^{maxDAT} iP_i^N \\ & + TTI \sum_{i=1}^{maxDAT} \sum_{j=0}^{i} \left\lfloor \frac{BLER^j N}{r} \right\rfloor P_i^N \end{aligned} \tag{8}$$

with $maxDAT$ setting the limit on the number of possible transmissions of a given PDU. As already mentioned, errors affecting successive PDUs are assumed to be uncorrelated.

Finally, to derive the throughput, we add RTT_{Wired} to RTT_{RLC} in (4) and insert the resulting RTT_{TCP} in the analytical expression from [4] giving the throughput of a TCP Reno connection. Assuming that the TCP window size reaches its maximum value W_{MAX}, the authors of [4] obtain

$$\begin{aligned} & Throughput\,(BLER, maxDAT) \\ = \quad & \frac{\frac{\bar{p}}{p} + \frac{W_{MAX}}{2} + Q\left(p, W_{MAX}\right)}{RTT_{TCP}\left(\frac{W_{MAX}}{4} + \frac{\bar{p}}{pW_{MAX}} + 2\right) + \frac{Q(p,W_{MAX})G(p)T_0}{\bar{p}}} \end{aligned} \tag{9}$$

with the help of (3) and (8). In (9), $\bar{p}$ stands for $1 - p$, T_0 represents the TCP retransmission timer, while

Simulation	1	2	3
Number of simulations	5	5	10
Length [s]	600	600	60
Number of users	1		
Traffic	FTP		
Downlink [kbps]	32	128	384
RTT_{Wired} [s]	5.1966	1.283	0.4321
DCH TTI [ms]	10		
PDU size [Bytes]	40		
W_{Max}	20		
TCP Version	Reno		
$maxDAT$	[1..12]		
BLER	[0..0.4]		

TABLE I

Simulations parameters

$Q(p, W_{MAX})$ and $G(p)$ are functions defined as

$$\begin{aligned} & Q\left(p, W_{MAX}\right) \\ = \quad & \min\left\{1, \frac{\left[1-\bar{p}^{3}\right]\left\{1+\bar{p}^{3}\left[1-\bar{p}^{W_{MAX}-3}\right]\right\}}{1-\bar{p}^{W_{MAX}}}\right\} \end{aligned} \tag{10}$$

$$G(p) = 1 + \sum_{i=1}^{6} 2^{(i-1)}\, p^{i}. \tag{11}$$

IV. Simulation set-up

The scenario described in Section II has been simulated with the help of the Network Simulator ns-2 [5] and its Enhanced UMTS Radio Access Network Extensions (EURANE) [6]. These extensions did not provide a limitation for the number of retransmissions, and therefore we had to implement it ourselves. Only the hatched elements of Fig. 1 were simulated.

The details of the ns-2 simulation set-up were as follows: a single user was downloading a single file using FTP, during either 60 or 600 seconds, with a downlink bandwidth of 32, 128 or 384 kbps. The uplink bandwidth was set to 32 kbps, to avoid uplink packet loss due to congestion. The $BLER$ was set within $[0, 0.4]$, increasing by 0.05 for each simulation. The simulation parameters are summarised in Table I. The TCP performance is measured in terms of RTT and average throughput. The RTT is calculated according to Karn's algorithm [7, p. 301], e.g. ignoring retransmissions. The average throughput is the total amount of transfered data during 5 (32 kbps, 128 kbps) or 10 (384 kbps) simulations, divided by the total simulation length. The RLC buffer size was set large enough to have no impact on TCP performance, as opposed to the work presented in [2].

V. Analysis of Results

Figs. 2-5 show the RTT and the throughput as a function of the PDU error rate $BLER$ and the maximum number of RLC AM transmissions $maxDAT$, for a downlink bandwidth of 128 kbps. $BLER$ value goes from 0 to 0.4, while we limited $maxDAT$ to 12. RTT and throughput values are either measured by post-processing ns-2 trace files (Figs. 2 and 4) or derived from relations (8) and (9), refer to Figs. 3 and 5.

Estimates of the RTT for low $maxDAT$ and high $BLER$ are either unavailable or questionable in Fig. 2. Obviously, the total RTT can not be lower than RTT_{Wired}. However, some estimates seem to be that small. This is actually a statistical artefact, due to the fact that the number of collected samples from ns-2 trace files was too low to enable us to derive a statistically meaningful estimate. The width of this uncertainty area grows with $BLER$. In the corresponding area in Fig. 4, no throughput is observed. We will call the limit beyond which no throughput is observed Ret_A. On the other hand, there is a $maxDAT$ value beyond which the throughput is not growing anymore, that we shall call Ret_B. For a given $BLER$, $[Ret_A, Ret_B]$ bound the $maxDAT$ interval setting the actual TCP throughput, as illustrated in Figs. 6 and 7, to be discussed later on.

For $maxDAT$ values smaller than RET_A, there is always at least one RLC PDU corrupted or lost per TCP packet, and therefore the whole TCP packet is regarded as lost. Since there is no explicit congestion notification, the next TCP packet is sent only after the TCP timeout has expired. Throughput is therefore very low. For $maxDAT$ values in $[Ret_A, Ret_B]$, some TCP packets are being received correctly, and the throughput is growing very fast with $maxDAT$, reaching its maximum in Ret_B. For $maxDAT$ values beyond Ret_B, no further increase of the throughput is observed. If $maxDAT$ is set higher than Ret_B, then the throughput decreases linearly with $BLER$, even for high RLC PDU loss rates.

Moving to the RTT, one can notice from Fig. 2 that the average RTT grows slightly more than linearly with $BLER$.

Comparing simulated to analytical results, one can notice a pretty good match between absolute figures on the average. Table II presents the average and maximal mismatch percentage Δ between simulation and analytical results for the three investigated bandwidths, for both RTT and throughput. For RTT values, the uncertainty area discussed here above has been disregarded. Whereas a good match is achieved at 128 kbps, the mismatch culminates at 74.43% for the 32 kbps bandwidth. However, one should keep in mind that the authors of [4] acknowledged themselves that their expression of the throughput was suffering from some limitations. It seems that these limitations appear here.

Finally, as already announced, Figs. 6 and 7 show the influence of $maxDAT$ on the RTT and the throughput of a 128 kbps TCP connection at $BLER = 0.15$ and 0.4 respectively. Such figures can help operators to assess the incidence of the choice of their $maxDAT$ value. They

Downlink [kbps]	Δ_{RTT}		$\Delta_{Throughput}$	
	Average	MAX	Average	MAX
32	13.64%	35.64%	22.74%	74.43%
128	6.93%	13.85%	8.90%	34.79%
384	14.33%	35.58%	13.38%	31.31%

TABLE II

Comparison between simulation and analytical results

illustrate in how much they are confronted with conflicting goals, as they would like to achieve the shortest possible RTT, which requires small $maxDAT$, while delivering the highest throughput, typically reached at high $maxDAT$. In the $BLER = 0.15$ case (Fig. 6), a $maxDAT$ value of 6 enables to offer the highest throughput (115 kbps) at the expense of a RTT equal to 1.4 s. However, dividing $maxDAT$ by 2 leads to a division by 8 of the throughput without a similar reduction of the RTT. The situation is even more problematic when $BLER = 0.4$ (Fig. 7) where a $maxDAT$ of 6 would enable to achieve less than one eighth of the measured maximum throughput. Adapting $maxDAT$ to the channel condition might therefore appear as an appealing solution. However, the signalling overhead its implementation would require most likely rules out this idea. Nevertheless, figures such as the ones shown in this paper help to pick an operating $maxDAT$ value.

VI. Conclusion

With the help of existing analytical models that we enhanced on the one hand, and computer simulations on the other hand, we have shown in this paper in how much the number of RLC local retransmission has a direct impact of the TCP performance. The presented results illustrate to which extend the selection of the number of retransmissions based on the quality of the wireless link expressed in terms of error rate can help in achieving a good trade-off between latency and throughput.

Acknowledgements

The first two authors would like to thank their colleague Etienne Cuvelier for his help in mastering the analytical developments of Section III. The authors are also grateful to the anonymous reviewers for their help in improving the quality of the paper.

References

[1] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network, “Radio Link Control (RLC) protocol specification”, Technical Report TS 25.322 v6.2.0, December 2004.

[2] Robert Bestak, Philippe Godlewski and Philippe Martins, “RLC Buffer Occupancy When Using a TCP Connection Over UMTS”, in *Proceedings of 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC 2002, Lisbon, Portugal*, volume 3, pp. 161–165, September 2002.

[3] Janne Peisa and Michael Meyer, “Analytical Model for TCP File Transfers over UMTS”, in *Proceedings of 2001 International Conference on Third Generation Wireless and Beyond, San Francisco, United States of America*, May 2001.

[4] Jitendra Padhye, Victor Firoiu, Donald F. Towsley and James F. Kurose, “Modeling TCP Reno Performance: a Simple Model and its Empirical Validation”, *IEEE/ACM Transactions on Networking*, vol. 8, n. 2, pp. 133–145, April 2000.

[5] Anonymous, “The Network Simulator - ns-2”, [online]. No date [Cited 24 February 2005]. Available from: <http://www.isi.edu/nsnam/ns/>.

[6] Twente Institute for Wireless and Mobile Communications, “EURANE Website”, [online]. 15 February 2005 [Cited 24 February 2005]. Available from: <http://www.ti-wmc.nl/eurane/>.

[7] W. Richard Stevens, *TCP/IP Illustrated, Volume 1 - The Protocols*, Addison-Wesley, 1994.

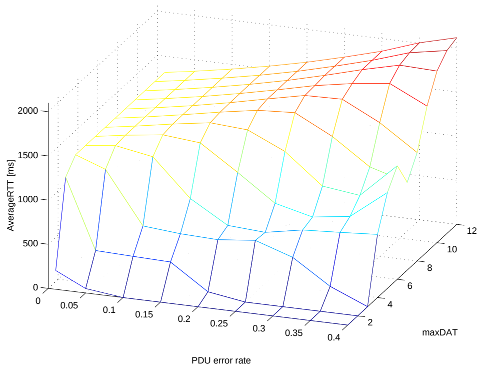

Fig. 2. Average RTT for 128 kbps - ns-2/EURANE results

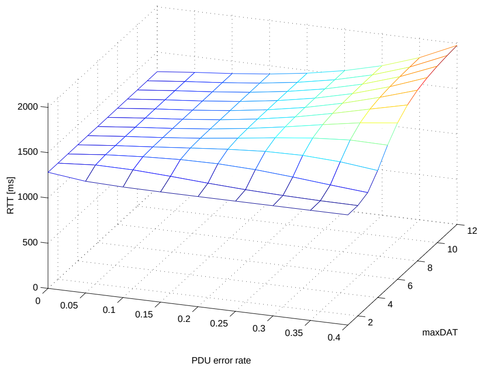

Fig. 3. RTT for 128 kbps - Analytical results

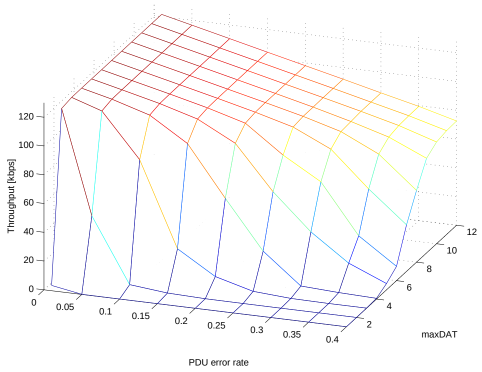

Fig. 4. Average throughput for 128 kbps - ns-2/EURANE results

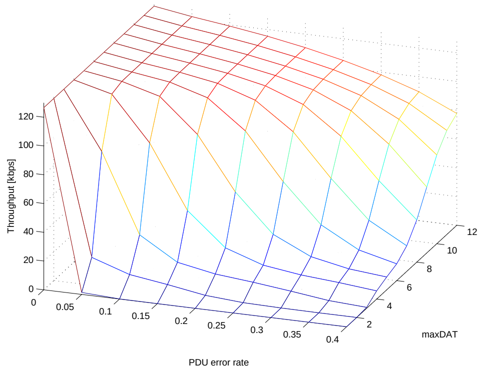

Fig. 5. Throughput for 128 kbps - Analytical results

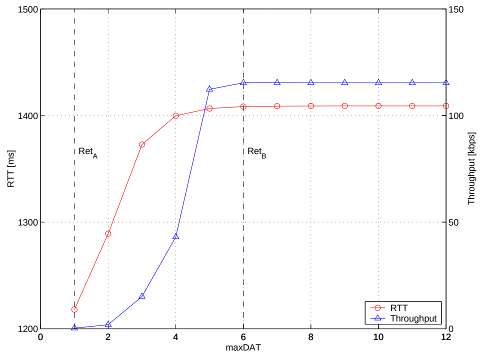

Fig. 6. RTT and throughput vs. $maxDAT$ at $BLER = 0.15$

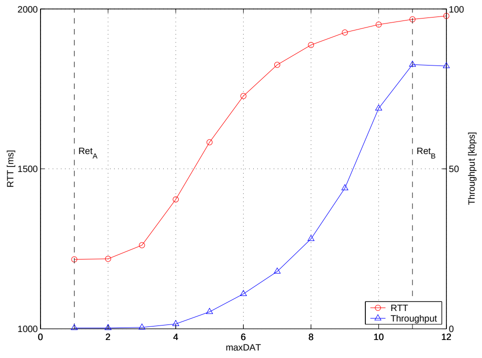

Fig. 7. RTT and throughput vs. $maxDAT$ at $BLER = 0.4$