TCP Performance over GPRS

Michael Meyer
Ericsson Eurolab Deutschland GmbH
Ericsson Allee 1
52134 Herzogenrath, Germany

Abstract - GPRS is the new packet-oriented data service for GSM. Soon it will be possible to take advantage of the features provided by GPRS for Internet Access like fast connection set-up, volume-based charging and staying on-line for long periods. For non-real-time Internet applications TCP is the applied transport protocol. Often it was suspected that GPRS introduces severe performance degradations for TCP-traffic since both apply their own ARQ mechanism. It is well known that layered protocol interactions can have very negative influences. This paper examines with simulations the performance of TCP over GPRS. The behaviour of the protocols will be analysed under various conditions. It will be shown that both TCP and GPRS are well harmonised.

I. INTRODUCTION

Nowadays, the current growth rates of data traffic in fixed communication environments are extrapolated for wireless environments. It is expected that there will be a significant demand for wireless data services in the near future. It seems that the success of wireless data is at the moment just prohibited because certain requirements for data rates and cost are not fulfilled. But it is not visionary to predict that with enhanced data services e.g. for GSM with HSCSD (High Speed Circuit Switched Data), GPRS (General Packet Radio Service), EDGE (Enhanced Data Rates for GSM Evolution) and ultimately UMTS (Universal Mobile Telecommunication System) this will change.

GSM already supports data users with circuit switched services covering data rates from 1.2 to 64 kbit/s (with HSCSD). Soon GPRS takes off to offer improved services for data users in mobile networks. GPRS is the packet-oriented extension of GSM. It will allow data transmission up to rates of more than 100 kbit/s. Since GPRS operates packet-oriented, many users can share the scarce radio resource. In this way a very flexible access is possible while idle users can still be online always and anywhere. A detailed description of GPRS is given in [1].

Under question is still how Internet applications will behave over GPRS. This paper tries to give some insight into this topic. It focuses on bulk data transmitted using TCP/IP. TCP (Transmission Control Protocol) is the Internet standard protocol for ensuring end-to-end reliability between communicating peer nodes.

In this scenario two worlds of protocols come together. On one hand, the Internet protocols TCP and IP, which are not especially designed for wireless environments. On the other hand, highly sophisticated radio protocols, which optimise data transmission over the problematic radio interface. This link suffers from time-varying characteristics, shadowing, interference and high bit error rates. Traditionally, networking researchers focus on wireline environments and have little experience with radio transmission, while radio engineers concentrated on their specific problems. With GPRS arose for the first time the requirement that this cellular network service should offer optimal support for Internet applications and it was therefore designed in this direction.

Has this design goal been achieved? Since there is currently no system available, simulations have been performed to examine this question. The simulator [2] comprises an event driven model of GPRS- and Internet protocols. It allows studying the performance under various conditions, which are specifiable by simulation parameters.

In brief, TCP offers mechanisms for congestion avoidance ensuring network stability for the Internet. More important for this study is the ARQ (Automatic Repeat Request) functionality. It provides sliding window based ARQ including an adaptive time-out mechanism for ensuring reliable data transmission. The ARQ is additionally supported by a function called Fast Retransmit, which allows the retransmission of single lost packets without waiting for time-outs. For a detailed description of TCP see [3].

Numerous papers describe methods to improve the TCP performance over wireless links [4,5]. This paper however focuses on an analysis of standard TCP to investigate which specific problems occur for GPRS. It will be shown that most of the findings made in the above mentioned studies are not applicable or negligible for GPRS. Therefore most of these proposed modifications are not necessary.

Usually it is assumed that packet losses on the radio link cause severe problems for TCP, since TCP reacts on packet losses with slowing down the transmission leading finally to performance degradation. But differently from studies for WLAN systems like [5] GPRS offers a reliable link with its own ARQ mechanism included in the RLC protocol (Radio Link Control) [6]. This is adapted to the GSM radio specific environment and based on radio blocks much smaller than usual maximum segment sizes for TCP for an optimal trade off between FEC (Forward Error Correction) and ARQ.

In this paper it will be shown which performance can be expected under a number of different scenarios while downloading bulk data, i.e. large files.

II. GPRS

GPRS is the packet-oriented extension of GSM. This extension relies on the re-use of the radio infrastructure of GSM while introducing new network nodes in the core network providing the required packet switching functionality. GPRS is mainly intended to provide better service for Internet applications compared to existing circuit switched services of GSM. Certain timeslots of the TDMA frame can be statically or dynamically allocated to GPRS. These are denoted as Packet Data Channels (PDCH). GPRS provides mechanisms for efficient sharing of these radio resources by multiplexing several users over one PDCH as well as the possibility that one user transmits over several PDCH in parallel if the terminal equipment supports this. For a more detailed description of the concept of GPRS see [1].

GPRS provides besides other modes of operation a reliable RLC mode, which is favorable for non-real-time applications. Based on FEC ensuring reasonable block error rates, the RLC ARQ mechanism recovers received erroneous packets.

One RLC block, which forms the retransmission unit for GPRS, is transmitted in four consecutive bursts according to the TDMA concept of GSM. In order to minimize delays, interleaving is also performed within these 4 bursts (compared to 22 bursts for circuit-switched data). To cope with a wide variety of channel conditions 4 different coding schemes were standardized. They are summarized in Table 1. The values are related to a single timeslot. If a user is able to transmit over several PDCH in parallel the data rates have to be multiplied by the given number of timeslots.

Table 1: GPRS Coding Schemes

Coding Scheme	Code Rate	Payload Bits per RLC Block	Data Rate
CS-1	1/2	160	8.0 kbit/s
CS-2	~2/3	240	12.0 kbit/s
CS-3	~3/4	288	14.4 kbit/s
CS-4	1	400	20.0 kbit/s

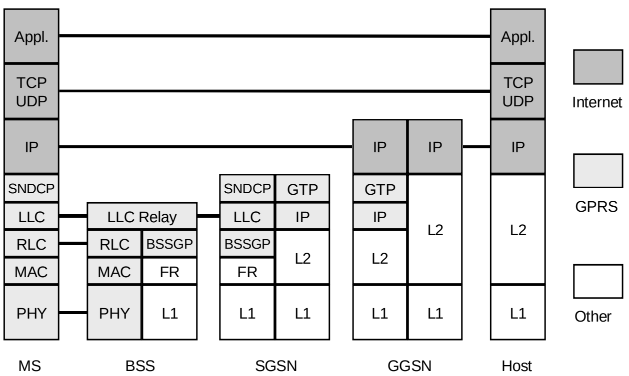

Fig. 1: GPRS Protocol Stack

The transfer of RLC Data Blocks is controlled by a selective ARQ mechanism coupled with the modulo-128 numbering of the RLC Data Blocks within one Temporary Block Flow. The sending side (the MS or the network) transmits blocks within a window of 64 blocks and the receiving side periodically sends ACK/NACK messages. Every such message acknowledges all correctly received RLC Data Blocks up to an indicated block sequence number (BSN), thus “moving” the beginning of the sending window on the sending side. Additionally, a bitmap that starts at the same RLC Data Block is used to selectively request erroneously received RLC Data Blocks for retransmission. The sending side then retransmits the erroneous RLC Data Blocks, eventually resulting in further sliding the sending window.

III. TCP

TCP is the most widely used transport protocol for non-real-time Internet applications like WWW, File Transfer and Email. It provides a connection-oriented end-to-end service ensuring the reliable transfer of data. Besides the data reliability and in-order delivery TCP is also responsible for flow control in the Internet to avoid congestion. This is achieved by complex mechanisms trying to probe for a data rate as high as possible but backing off as soon as congestion occurs. TCP segment losses are interpreted as congestion signals as they are traditionally caused by buffer overflows in routers. If such an event is detected, TCP adjusts its parameters like window size and retransmission timeout values, thus slowing down.

In wireless environments, poor radio characteristics can lead to packet losses or at least to very long delays leading to TCP timeouts. In these cases the reason for packet losses is not congestion and the basic working assumption for TCP is wrong. Thus its countermeasures are also not optimal. Timeouts should only occur if segments are definitely lost and not only too long delayed. For an in-depth discussion of this problem see also [7].

IV. Simulation Results

In order to be able to study the influences of the involved protocols an event-driven simulator was developed. The main concept and structure of the simulator was already described in other publications, e.g. [2]. In order to spend more effort on the obtained results a thorough description is neglected here.

This paper aims at providing insight into the behavior of TCP over GPRS. The study focuses on a single user. Accordingly, multiplexing or sharing resources with other users is not the topic of this paper. Further research will be devoted to these issues.

The simulation results, which will be presented in this section, are based on the following parameter settings of the simulator.

Table 2: Main Simulation Parameters

Max. TCP Send Window	8192 Byte
TCP maximum segment size	512 Byte
TCP/IP Header Compression	Off
LLC Mode	UNACK
Number of PDCH	4
Multislot Class Uplink	1
Multislot Class Downlink	4
C/I Mean	6-30dB
Channel Coding	CS 1- CS4

A few of the parameters need some explanation. It is assumed that 4 PDCH are allocated for GPRS, i.e. that 4 timeslots on one GSM frequency can be used for GPRS. The mobile terminal has the capability to use 1 PDCH in up- and 4 PDCH in downlink direction in parallel for increased data rates.

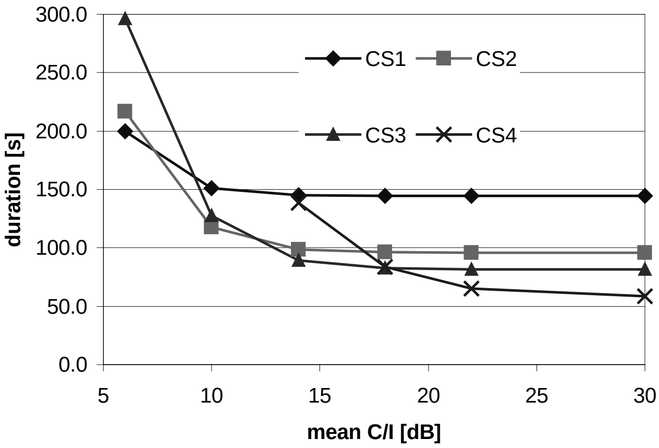

Fig. 2: Download duration for 500 kbyte for different C/I

Fig. 2 shows the download time for 500 kbytes of application data over GPRS. As expected, for different channel scenarios the particular coding scheme has the dominating influence on the download time. Anyway, under the assumption that this single user has not to share the existing resources with other users, 500 kbyte of data can be downloaded within 200 s for CS1 and C/I=6dB (rather tough conditions) and 58 s using CS4 for C/I=30dB. As soon as channel conditions improve it is highly desirable to switch to less robust coding schemes enabling a superior FEC/ARQ trade off, which finally results in higher throughputs. Under mediocre conditions (C/I=14dB) throughputs between 27 kbit/s (CS1) and 44.8 kbit/s (CS3) can be achieved.

Several effects contribute to the fact that the theoretical throughput of the data rate corresponding to Table 1 cannot be provided (e.g. 32 kbit/s for CS1). First of all there are retransmissions of RLC blocks necessary depending on the block error rate of the channel. Furthermore, protocol overhead from layers on top of RLC has to be taken into account. Since no TCP/IP header compression was applied, the overhead per 512 Byte of TCP payload sums up to 51 Byte, i.e. 10 percent. Finally, although the packet data resources are solely dedicated to the considered user, signaling messages (like broadcast messages, RLC acknowledgements for up-link traffic (TCP ACKs)) have to be transmitted. Under optimal channel conditions a throughput of approximately 85% of the value corresponding to Table 1 can be achieved. Example: For CS2 and C/I=30dB the duration of 96s corresponds to a bit rate of 41.6 kbit/s compared to 48 kbit/s according to Table 1 for 4 PDCHs.

The remaining part of the paper is dedicated to show how TCP behaves over GPRS. Fig. 3 shows a trace of TCP segments arriving at the mobile terminal. Within the 30s of simulation the C/I was switched between two values. Initially, the C/I was 13dB. After 10s the C/I was changed to 6dB and after 20s back to 13dB. 13dB corresponds for the applied CS2 to 3.4% block errors and 6 dB to 46.4%.

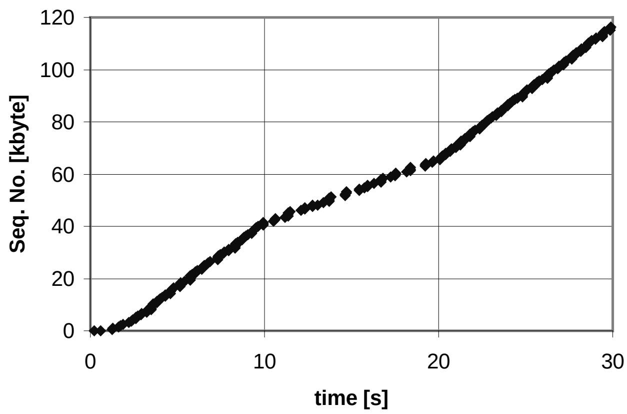

Fig. 3: TCP receiver trace

Fig. 3 shows that under these conditions TCP perceives a constant data flow. In total 118 kbyte are transmitted in 30s. Not a single timeout or retransmission occurred although the channel was rather bad during 10s of this simulation run. Obviously, the RLC ARQ mechanism works sufficiently fast to avoid spurious TCP timeouts (s. also [7]). Problems of the link layer are just recognizable by increased packet interarrival times of TCP segments. The two different slopes of the curve for the good and bad channel indicate this.

Fig. 4 (subset of Fig. 3) shows the transient region around 10s. While in the first half (C/I=13dB) the packets arrive more or less regularly, during the second half the segments arrive with a certain variance in delay.

Fig. 5 depicts the corresponding RLC trace. RLC blocks use a block sequence number (BSN) based on a modulo-128 notation, leading to the cyclic structure of the trace. From this trace Fig. 4 can be explained in more detail.

For example, the reason why three TCP segments arrive at 8s in a bulk is due to several retransmissions on RLC. Since RLC has to deliver packets in order, it can happen that all blocks corresponding to subsequent TCP segments have been already successfully received. They are queued until an in order delivery is possible. Therefore, the RLC receiver releases those packets in a single burst.

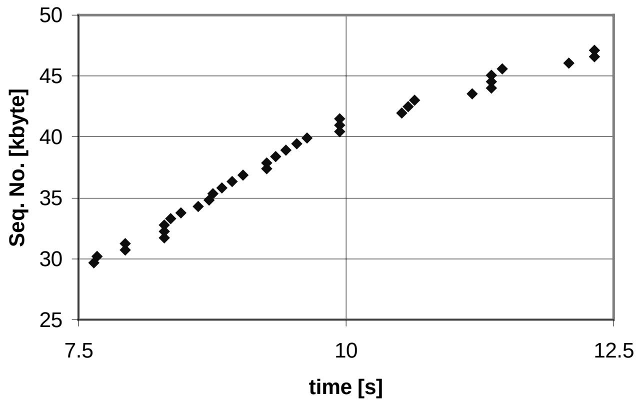

Fig. 4: TCP receiver trace

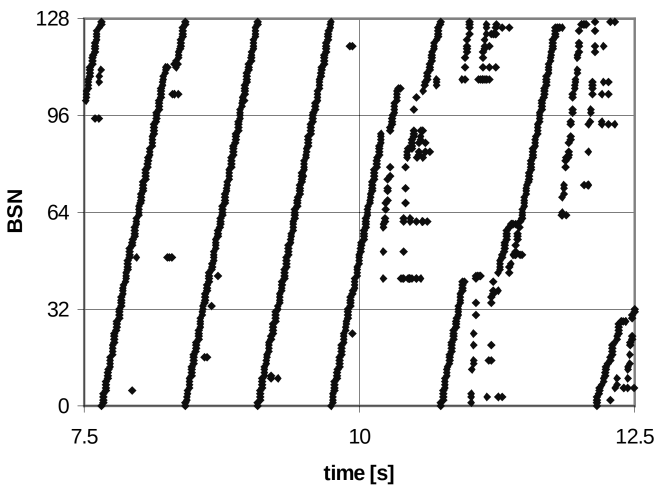

Fig. 5: RLC trace with retransmissions corresponding to Fig. 4

After 10s the radio conditions become worse. Sometimes several attempts are necessary before an RLC block is successfully received. But even under these poor conditions it takes always less than 1.5s until all blocks corresponding to a TCP segment are received without error. Anyway, in these situations the poor channel can lead to stall conditions of the RLC window as for example at ~11s. The window comprises only 64 blocks and these can be transmitted within only 320 ms using 4 timeslots. In such situations, when several retransmissions are required, RLC is still working on the delivery of a certain packet and cannot advance the window.

Nevertheless, it can be concluded that TCP works very stable over GPRS, since GPRS shields TCP from problems of the radio link. But TCP contributes also to this robustness by adapting its protocol parameters like the retransmission timeout value (RTO). This can be explained by examining the timeout behavior of TCP for the considered scenario. In general, the results show the demand for a reliable link layer for wireless links supporting TCP. Similar conclusions are drawn in [8] for Wireless LANs.

TCP updates its RTO value based on its received acknowledgements and the measured round trip time (RTT). Fig. 6 shows the measured RTT for each received acknowledgement and the RTO calculated from these values using a smoothing function and also the variance of RTT. For details see [3]. During the phase with 13dB C/I the RTT fluctuates around 1.8s and the RTO converges to 2.6s. That implies that TCP can have unacknowledged data outstanding for 2.6s before it times out. This gives the RLC of GPRS the chance to perform several retransmissions, if required, before TCP times out. As soon as the channel becomes worse, the RTT increases, since RLC retransmissions contribute to the delay. Accordingly, the RTO is adapted very quickly to larger values. In this example it takes only 3s (starting at time 10s) that the RTO is increased to 5s giving even better chances for RLC to recover.

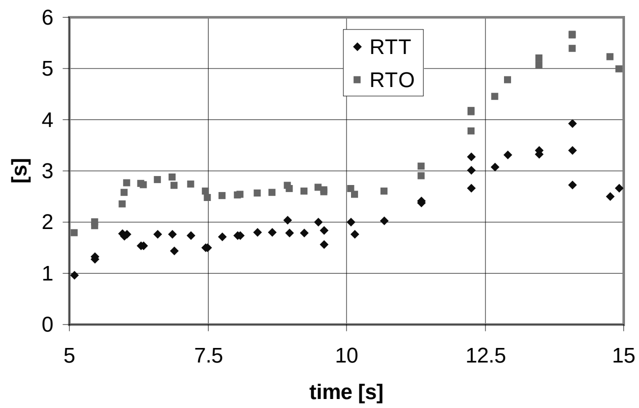

Fig. 6: RTT and RTO trace

Fig. 7 gives the RTT and RTO curves for the complete simulation period of 30s to provide a better overview on RTO dynamics. At the beginning the RTO is initialized with a default value of 4s [3]. As soon as TCP ACKs are received, the RTO is updated and converges to a value of 2s. During this initial phase the RTT increases from values of a few hundred milliseconds to 1.8s due to the fact that the TCP send window opens up (TCP slow start [3]). TCP sends more and more segments until the maximum send window size of 8kbyte of unacknowledged data is reached. Buffering of these segments increases the RTT.

After 5s the send window size has reached its maximum value. Then the RTT and RTO stabilize. After 10s the bad channel conditions lead to increased RTTs and the RTO grows. When the channel is changed back to 13dB after 20s it takes 2s for the RTT to decrease again. Due to the smoothing function involved in the calculation of the RTO, this value needs about 7s to arrive again at the same level as at 10s. As a conclusion it can be stated that TCP maintains an adaptive RTO value which allows under the demonstrated circumstances for enough time for RLC to recover from packet losses. The layered approach of two ARQ schemes, RLC and TCP, proves to be very useful compared to relying on TCP error recovery mechanisms alone [5].

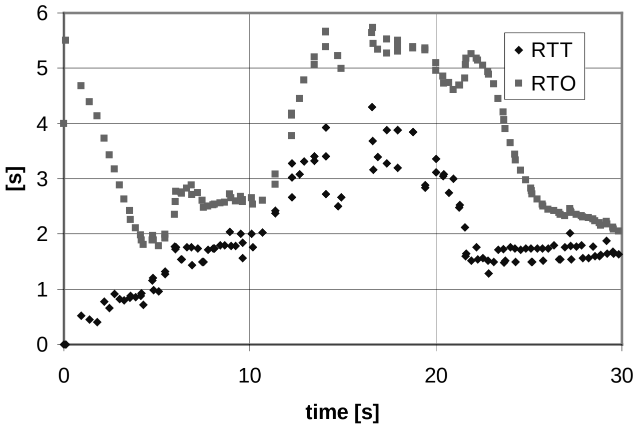

Fig. 7: RTT and RTO trace

Finally, it shall be shown what happens if the radio problems are too severe and TCP times out. Fig. 8 shows the TCP trace for the sender and the receiver. Again, the two state channel was used, but now with 20dB and 8dB respectively. Additionally, the used coding scheme was changed to CS4 resulting in BLERs of 11.8% and 91.6%.

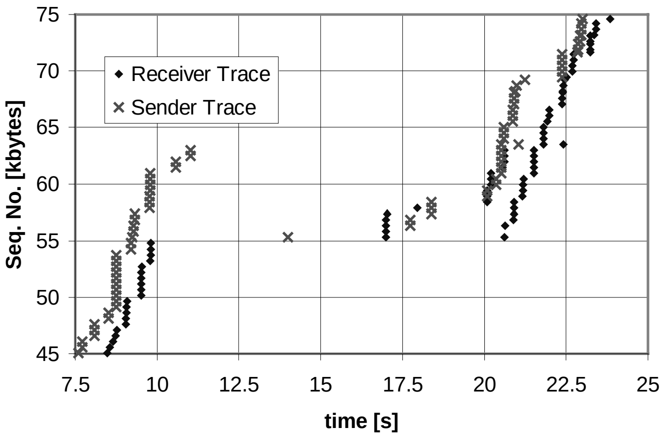

Fig. 8: Sender and Receiver TCP Trace

At 10s the RTO had converged to 2s. Between 10s and 11s two ACKs are received, enabling TCP twice to send two segments. The RTO at that time is 3s. For several seconds no segment is received, thus also no ACK is triggered. This leads to a timeout at 14s triggering the retransmission of the segment with Seq. No. 55296. RLC in this simulation is configured to be fully reliable. Thus, the original packet and the retransmission are in the transmission pipe. The original packet arrives after 17s together with 4 other segments. The corresponding ACKs trigger the transmission of new segments. The sending TCP assumes that all sent segments have been lost and therefore starts to retransmit all of them. Moreover, all incoming duplicate packets arriving after 20.7s trigger duplicate ACKs finally leading to the fast retransmission of segment 63488. In total 16 duplicate segments were received totaling to 8192 byte, thus wasting radio resources for 1.2s. This example shows how disastrous concurrent ARQ mechanisms could interact.

To be fair, it has to be mentioned that GPRS provides the possibility to adapt the coding scheme as soon as problems with a weak coding scheme are detected (CS1 works well with 8 dB C/I as shown before). Additionally, it cannot be expected that protocols work perfectly, if there is no radio coverage available.

V. CONCLUSIONS

The paper has investigated performance issues of TCP when operated over GPRS. Several traces for TCP and RLC were discussed to highlight the characteristics, if GPRS is applied as access for bulk data download.

Download time measurements have shown that GPRS can operate effectively over a wide range of channel conditions thanks to four different coding schemes. Further results have revealed that GPRS provides a sufficiently fast working ARQ mechanism based on the RLC protocol, which allows typically several retransmissions before TCP times out. The RLC ARQ scheme is appropriately designed to ensure that TCP observes just packet delays rather than losses. If additionally these delays could be bounded in such a way that no TCP time-out occurs, TCP would not notice packet errors on the wireless link at all.

REFRENCES

[1] Brasche, G., Walke, B.: Concepts, Services, and Protocols of the new GSM Phase 2+ General Packet Radio Service, IEEE Communications Magazine, pp. 94-104, August 1997.

[2] Hoff, S., Meyer, M., Sachs, J.: Analysis of the General Packet Radio Service (GPRS) of GSM as Access to the Internet, ICUPC'98, Florence, 5.-9., October 1998.

[3] Stevens, W.R.: TCP/IP illustrated, Vol.1, Addison Wesley, 1994.

[4] Caceres, R., Itofde, L.: Improving the performance of Reliable Transport Protocols in Mobile Computing Environments, IEEE JSAC, Vol. 13, No. 5, pp. 850-857, June 1995.

[5] Balakrishnan, H., Seshan, S., Katz, R.: Improving reliable transport and handoff performance in cellular wireless networks, Wireless Networks, Vol. 1, No. 1, pp. 469-481, February 1995.

[6] Digital cellular telecommunications system (Phase 2+); General Packet Radio Service (GPRS); MS-BSS Interface; RLC/MAC; (GSM 04.60 version 6.1.0)

[7] Ludwig, R., Rathonyi, B., Konrad, A., Oden, K., Joseph, A., Multi-Layer Tracing of TCP over a Reliable Wireless Link, SIGMETRICS 99.

[8] Eckhardt, D. A., Steenkiste, P., Improving Wireless LAN Performance via Adaptive Local Error Control, ICNP 98.