THE IMPACT OF PREEMPTIVE PRIORITY IN GPRS ON TCP PERFORMANCE: A MEASUREMENT STUDY

Annika Wennström, Anna Brunstrom
Dept. of Computer Science, Karlstad University
SE-651 88 Karlstad, Sweden

Juan Rendón
Technology Dept., Pompeu Fabra University
Passeig de Circumvalació 8, 08003 Barcelona, Spain

Jan H. Gustafsson
Telia Mobile AB
Lagergrens gata 7, SE-651 15 Karlstad, Sweden

Keywords: Wireless Internet, TCP, GPRS, preemption.

Abstract: GPRS extends the widely deployed GSM system with a more efficient wireless Internet access. In this paper we investigate how a TCP transmission over GPRS is affected when it loses all its resources due to preemption by circuit-switched calls with higher priority. The results indicate that TCP performance is degraded more than necessary, as buffered data is flushed immediately when the GPRS traffic is preempted. The time required for error recovery is considerable also for very short preemption periods. The situation would improve if data was buffered during preemption and if the data was transmitted immediately as GPRS resources become available again.

1 INTRODUCTION

The General Packet Radio Service (GPRS) (GSM03.60, 2000), (Bettstetter et al., 1999) is a packet-oriented extension to the Global System for Mobile Communications (GSM) that provides its users with an ”always on” wireless access to the Internet. As most applications on the Internet rely on the Transmission Control Protocol (TCP) (Postel, 1981) for transport, it is important that TCP works efficiently in GPRS.

In this paper, we consider TCP over GPRS in the case when the GPRS traffic is preempted by traffic with higher priority, such as circuit-switched calls. The results show that TCP suffers more than necessary due to preemption when GPRS resources are lost even for a very short time. In the testbed and in many commercial networks, the buffer in the Base Station Controller (BSC) is flushed immediately when there are no resources left for GPRS traffic. In many situations, TCP performance would improve if the BSC waited for a timeout interval before it flushed its buffer.

The impact of buffering in intermediate nodes on TCP performance is described in e.g. (Chakravorty et al., 2002), (Gurtov et al., 2002), (Sagfors et al., 2003). In (Sagfors et al., 2003) an active queue management scheme optimized for third generation cellular networks is proposed and evaluated through simulation. In (Chakravorty et al., 2002), (Gurtov et al., 2002) GPRS measurements are presented. The measurements do not focus exclusively on buffering, but as real networks are used aspects of buffering are taken into account. Neither of these studies investigate preemption. The analysis and simulation study in (Chen et al., 2002) suggests buffering of GPRS data as a means to reduce the blocking probability for GPRS when the GPRS traffic is preempted by circuit-switched calls with higher priority. However, TCP is not directly considered in (Chen et al., 2002).

In contrast to related work, we evaluate in detail how TCP is affected by GPRS buffering during preemption. Our work is based on measurements in a GPRS testbed consisting of real network nodes with real protocol implementations and an emulated radio environment.

The rest of the paper is structured as follows. Some background on GPRS and TCP is given in Section 2.

Wennström A., Brunstrom A., Rendón J. and H. Gustafsson J. (2004).
THE IMPACT OF PREEMPTIVE PRIORITY IN GPRS ON TCP PERFORMANCE: A MEASUREMENT STUDY.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 430-436
DOI: 10.5220/0001404804300436

In Section 3, an overview of the test environment is provided. The parameter settings used in the measurements are described in Section 4. In Section 5, the results are presented. The results and their implications are discussed in Section 6. Finally, in Section 7, some conclusions are drawn and plans for future work are presented.

2 BACKGROUND

In this section some background on GPRS and TCP is provided. The main purpose is to provide the information required for the rest of the paper, not to give a complete picture of either GPRS or TCP.

2.1 Overview of GPRS

GPRS provides more efficient sharing of radio resources and higher data rates than circuit-switched GSM. The GPRS system requires two new nodes: the Gateway GPRS Support Node (GGSN) and the Serving GPRS Support Node (SGSN). A GSM time slot allocated for GPRS is called a packet data channel (PDCH). The radio resources are more efficiently used than in GSM, since the PDCHs in a cell are shared between the GPRS users, and not, as in GSM, reserved for one user at a time. Access to the PDCHs is controlled by the Radio Link Control/Medium Access Control (RLC/MAC) protocol (GSM04.60, 2000) which provides a link between the mobile station (MS) and the Base Station Subsystem (BSS). The MAC protocol is similar to slotted ALOHA. In addition, the Logical Link Control (LLC) protocol provides a logical link between the MS and its associated SGSN. RLC and LLC support both acknowledged and unacknowledged data transmission. The Base Station Subsystem GPRS Protocol (BSSGP) operates between the SGSN and the BSS. BSSGP is further described in Section 3.

2.2 Overview of TCP

The TCP protocol provides reliable data transport. Error recovery is triggered by a timeout event or by three duplicate acknowledgments. The timeout value is dynamically adjusted based on estimates of the round trip time (RTT). In order to prevent congestion in the network, the amount of data the TCP sender can inject into the network is limited by a congestion window (cwnd) and several algorithms for congestion control are employed (Allman et al., 1999).

After a timeout event, the missing data is retransmitted and the slow start algorithm is used. The cwnd is first set to one segment, and then it is increased with one segment for each incoming acknowledgment which results in an exponential increase in the transmission rate. When a slow start threshold (ssthresh) value is reached, the congestion avoidance algorithm is used instead. Here, one new segment is added to the cwnd each RTT. After retransmission due to three duplicate acknowledgments fast retransmit and fast recovery are used. The cwnd is reduced to half of the value it had before the data loss. After the TCP sender receives an acknowledgment covering all the retransmitted data, the congestion avoidance algorithm is used.

The performance of TCP may be enhanced by the use of optional features. The selective acknowledgments (SACK) option (Mathis et al., 1996) gives the TCP sender precise information about the TCP segments that have arrived at the receiver. The timestamps option (Jacobson et al., 1992) provides an additional means to identify segments and their acknowledgments.

3 TEST ENVIRONMENT

The GPRS testbed used for the measurements is shown in Figure 1. The testbed consists of a client, a GPRS network, and a server. The client uses the GPRS network to access services provided by the server. The client in this testbed is a laptop connected with a serial cable to a GPRS terminal at the R reference point. The GPRS terminal accesses the GPRS network over the radio interface. In order to provide controllable and repeatable radio conditions, an emulated radio environment is used at the radio interface. The emulated radio environment interconnects the client with the BSS which consists of a BSC and at least one Base Transceiver Station (BTS). The GPRS terminal communicates with the BTS in the cell on which it currently camps. The BTS is connected at the Abis interface with a BSC which provides control functions and physical links over the Gb interface to an SGSN. The SGSN is connected to a GGSN over the Gn interface. The GGSN connects the GPRS network to the server, over the Gi interface.

Measurement data can be gathered at several places in the testbed. On the client and the server machines, Ethereal (Ethereal, 2004) captures packets which are further analyzed with Tcptrace (Ostermann, 2004). Between the BTS and the BSC, the NetHawk analyzer (NetHawk, 2004) captures GPRS data and control information, such as RLC/MAC blocks.

Since the server is placed in a wired network which is faster than the GPRS network, data from the server is buffered in the GPRS network before it is transmitted over the radio interface to the client. The maximum time that user data may be buffered before it is transmitted over the radio interface, is determined

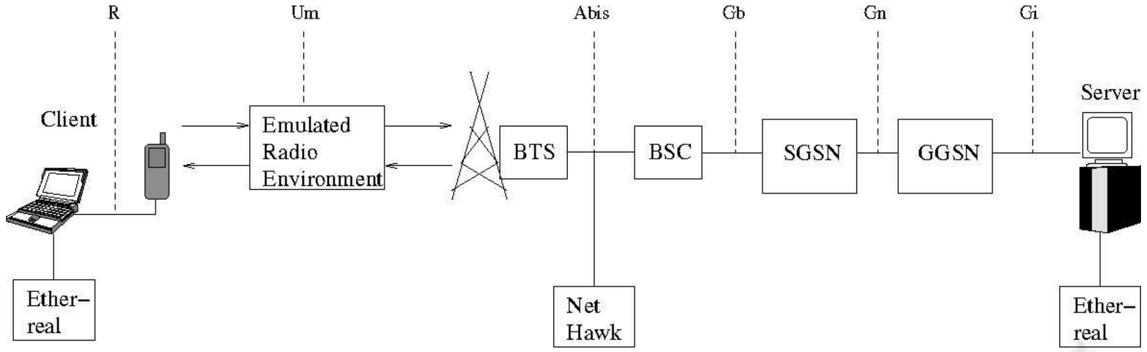


Figure 1: GPRS testbed

by the *maximum holding time* (GSM03.60, 2000). The buffer management in the GPRS network is coordinated between the SGSN and the BSC. The rate of the data transmitted from the SGSN to the BSC in the downlink is limited by BSSGP flow control. The SGSN keeps track of the time that user data is buffered and when user data is transmitted in BSSGP protocol data units (PDUs) to the BSC, the time that is left of the maximum holding time is indicated in the *PDU lifetime* header field (GSM08.18, 2000). After the maximum holding time data is considered old and is discarded from the SGSN or the BSC buffer, depending on where data is resided. It is then up to the end hosts or LLC to retransmit the discarded data if necessary. In the experiments presented, the end hosts use TCP to recover discarded data.

4 PARAMETER SETTINGS

For the presented measurements, the GPRS system is configured as follows. The maximum holding time is set to 63 seconds. The coding scheme is CS-2, LLC is run in unacknowledged mode, and RLC in acknowledged mode. These parameter settings were chosen, since they are commonly used in commercial GPRS networks. Compression is disabled, since compression complicates the analysis of the data traffic, and because header compression may degrade performance in case of data loss (Inamura et al., 2003). The radio quality is nearly optimal with a received signal level of -80dBm and a C/I of 20dB.

The measurements are conducted with only one GPRS client in the system, and the application model used is bulk transfer over one TCP connection. Data is transmitted in the downlink from the server in the wired network to the GPRS client. The end hosts run Linux 2.4.2-2 with the default settings of TCP options and parameters. This implies that SACK and timestamps are used, as recommended in (Inamura et al., 2003). The size of IP packets is 1500 bytes, which, due to enabled options, results in a maximum segment size of 1448 bytes. The receiver window is set to 64KB. The TCP implementation in Linux 2.4 is further described in (Sarolahti and Kuznetsov, 2002).

The GPRS terminal is capable of using three PDCHs in downlink and one in uplink. The raw data rate over three PDCHs is 40.2kbps with CS-2. The maximum data rate achievable on higher protocol layers is much lower due to overhead, such as protocol headers of all underlying protocols. The maximum rate of TCP data in the test environment with the chosen parameter settings is 31kbps, which is consistent with values reported for other test environments (Taferner and Bonek, 2002), (Gurtov et al., 2002).

The purpose of the measurements is to investigate how a GPRS data transfer is affected when it loses all its resources due to preemption by circuit-switched calls with higher priority. In order to preempt the GPRS resources, circuit-switched calls are set up between GSM terminals located in the cell. The GPRS client itself does not take part in any of the circuit-switched calls. We use the term *preemption period* to denote the time interval during which there are no resources available for GPRS traffic, because all resources are occupied by circuit-switched calls. The preemption periods tested are 3, 6, and 15 seconds. Each measurement lasts in total for 40 minutes and during this time the GPRS resources are preempted completely at least 30 times. The time between two consecutive preemption periods is one minute and during this time data is transmitted to the client over three PDCHs.

5 MEASUREMENT RESULTS

In this section, we first give an overview of the measurement results. Then, the results for 3 and 15 seconds preemption periods are described in further de-

Table 1: Overview of results

Preemption period (secs)	**Throughput (kbps)**	**Retransmitted packets (%)**	**RTT avg (ms)**
3	21.64	6.96	2911
6	22.20	6.55	3081
15	21.27	7.03	3316

tail. Finally, buffering and TCP error recovery are discussed.

5.1 Overview of Results

Table 1 gives an overview of the performance of TCP. The table shows throughput, percentage of retransmissions, and average round trip times for the tested preemption periods. In contrast to our expectations, the performance of TCP is similar for preemption periods of 3, 6, and 15 seconds. As compared to the maximum achievable throughput, the repeated preemptions lead to a reduction in throughput of roughly 30% even when the preemption period is as short as 3 seconds. The traces of captured traffic indicate that data losses occur every time the GPRS data transfer is preempted. This is due to the BSC buffer being immediately flushed when the GPRS resources are preempted. The data losses are unrelated to the BSSGP buffer setting, since data is not stored long enough in the BSSGP buffer for the maximum holding time to expire. Depending on the length of the preemption periods, buffered data may also be discarded from the GPRS terminal. The transmission is stalled during the preemption period and in order to restart the transmission again after the preemption period, new data transmitted from the server or an acknowledgment from the client is required. This is further explained below.

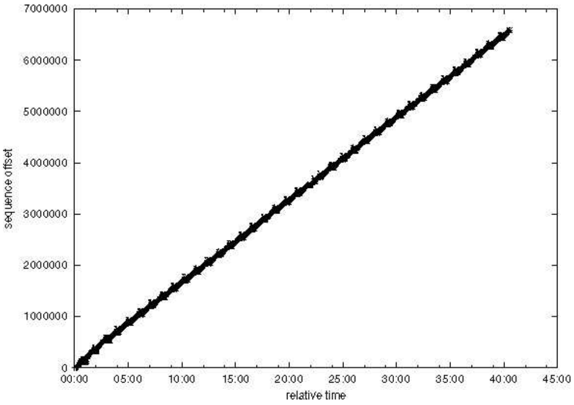
Even if the TCP performance is almost the same for all preemption periods, slightly different scenarios were found in the trace files. The traces for preemption periods of 3 and 6 seconds show a similar pattern of TCP behavior and GPRS signaling. For 15 seconds preemption another series of events are triggered. Other GPRS signaling messages are transmitted and TCP uses another strategy for error recovery. As the results are similar for 3 and 6 seconds, we discuss only the results for 3 and 15 seconds preemption in more detail.

5.2 Preemption Periods of 3 Seconds

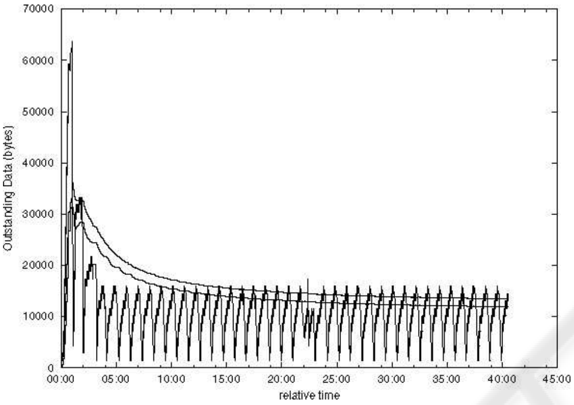
The results for 3 seconds preemption are shown in Figure 2. Figures 2(a), 2(b), and 2(c) show the sequence number evolution, the outstanding data, and the round trip time values, respectively. The time-sequence graph shows all segments captured on the server. The graph confirms that retransmissions occur every minute which is each time the GPRS data transfer is preempted. This is visible, because the sequence number does not increase when a segment is retransmitted which makes the graph look jagged. If no data was retransmitted, we would see a straight line instead. The outstanding data graph shows the amount of unacknowledged data. Before the transfer is preempted for the first time, the outstanding data reaches the capacity of the receiver window of 64KB, and then, after a few minutes, it stabilizes with peaks at 16KB. As shown in the RTT values graph, the round trip time varies between 1 and 4 seconds. A closer examination of the captured TCP data indicates that, in the typical case for preemption periods of 3 seconds the transmission is restarted after a preemption period by an acknowledgment that has been buffered in the GPRS terminal. The NetHawk trace supports this scenario, since it indicates that the first RLC/MAC event after preemption is a request for an uplink channel from the client.

5.3 Preemption Periods of 15 Seconds

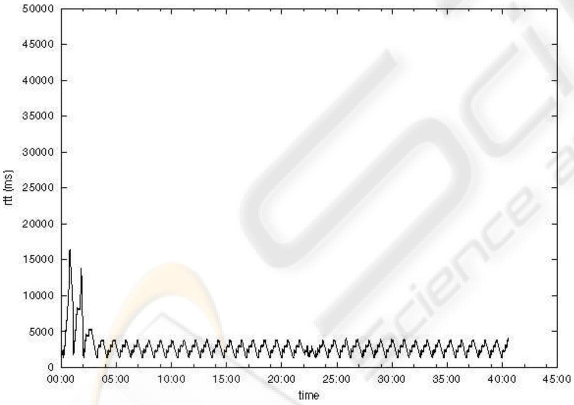
Figure 3 shows the detailed results for 15 seconds preemption. The results are similar to those for 3 seconds preemption. Packet losses occur at regular intervals that coincide with the preemption periods. After the initial slow start phase the transfer stabilizes and the outstanding data reaches levels of about 16KB, and the round trip time peaks at 4 seconds. However, in contrast to the typical scenario for the shorter preemption periods, a closer examination of the trace files indicate that the first event after a 15 seconds preemption period is a retransmission from the server, not an acknowledgment from the client. A preemption period of 15 seconds is long enough for the GPRS terminal to empty its buffer, and therefore there are no buffered acknowledgments to transmit after the preemption period. The first RLC/MAC event captured in the NetHawk trace after the preemption period is a request for a downlink channel from the server side.



(a) Time sequence graph

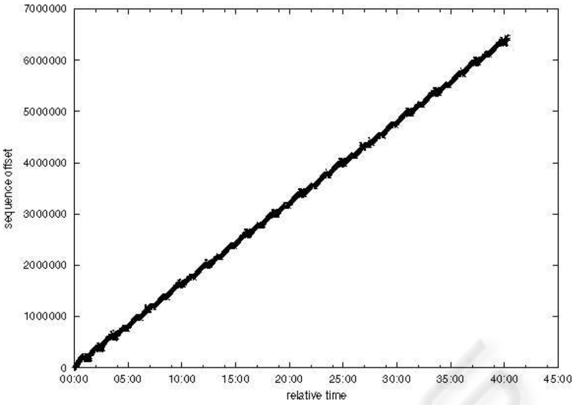


(b) Outstanding data

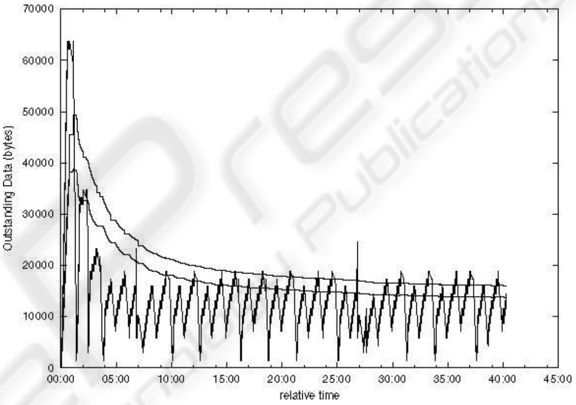


(c) RTT values

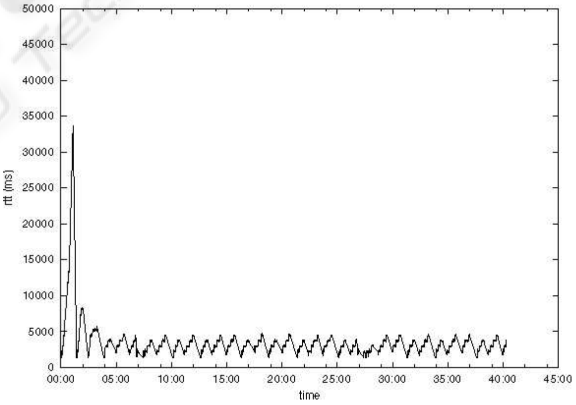
Figure 2: Preemption periods of 3 seconds



(a) Time sequence graph



(b) Outstanding data



(c) RTT values

Figure 3: Preemption periods of 15 seconds

5.4 Loss of Buffered Data

Only data buffered in the BSC is lost due to preemption, which is clearly illustrated in Figure 4. The detailed time-sequence graph, taken from a client trace, shows the first preemption period of 3 seconds. Transmitted segments are represented by diamonds, the advertised receiver window by the upper stair shaped line, and cumulative acknowledgments by the stair shaped line below the segments. Selective acknowledgments are marked with an *S* and out of order segments with an *O*. As seen in the figure, after 41 seconds, there is a gap in the sequence of incoming segments. The missing segments are discarded from the

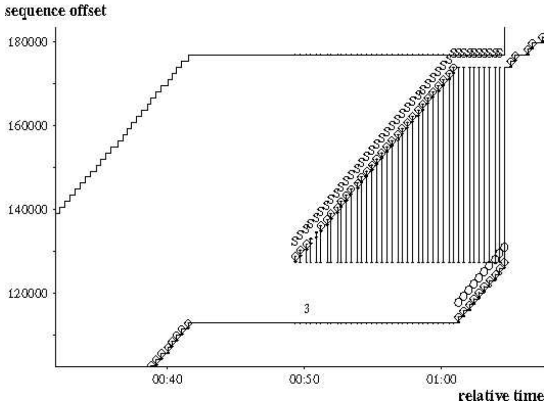


Figure 4: The first preemption period of 3 seconds

BSC buffer when the transfer is preempted. The segments received after the gap are buffered in the SGSN until the transfer is resumed after the preemption period. In this case, even though a large amount of data is buffered in the SGSN, the transfer is not restarted until the client transmits a buffered acknowledgment. The NetHawk trace indicates that the client requests an uplink channel after the preemption period.

5.5 TCP Error Recovery

A comparison of the results indicates that TCP performance is similar for the tested preemption periods. It takes 15 to 20 seconds for TCP to recover all the lost data, even for preemption periods of 3 and 6 seconds. The reason is that TCP uses different algorithms for congestion control when recovering from the losses. Preemption periods of 15 seconds result in slow start in most cases. After the shorter preemption periods, on the other hand, fast retransmit and fast recovery are used instead. Almost ten consecutive segments are lost each time preemption occurs for all settings tested (3, 6 and 15 seconds). Slow start leads to quicker recovery in this case which compensates for the longer preemption periods. Note that this is not an indication that performance in general is independent of the length of the preemption period. Even longer preemption periods than 15 seconds would of course lead to worse performance.

6 DISCUSSION

In the measurements the BSC discards buffered data immediately as the GPRS transfer is preempted, and then new data is required in order to start the GPRS transfer again after the preemption period. Since several TCP segments get lost when the BSC buffer is flushed it takes a substantial amount of time to recover even for short preemption periods. This problem could potentially be avoided by applying a timeout period before flushing data from the BSC buffer when the GPRS resources are preempted. Another alternative would be to leave the data in the buffer until resources become available again. Data buffered in the SGSN buffer should also be immediately transmitted when resources for GPRS become available again.

The TCP implementation in Linux 2.4 has many new features which have not been widely deployed yet. Measurements for other TCP implementations with less advanced congestion control would probably yield slightly different results. However, the main problem that a whole window of data is lost would still remain and lead to performance degradation.

A GPRS measurement testbed that combines the use of real network equipment and protocol implementations with a precise control over radio channel conditions is used for the experiments. Unlike simulations and live measurements, the testbed supports TCP performance measurements over a real GPRS network, yet it provides a repeatable environment in which a wide range of parameter settings can be explored. At the same time, we must be aware that the presented results are not directly applicable for all GPRS systems. The GPRS specifications do not always put strict requirements on buffering, which implies that the handling of buffering is partly system dependent. A testbed with a different hardware and software configuration may therefore produce different results. Still, we feel that real measurements in a controllable environment are invaluable for the understanding and enhancement of TCP performance in the GPRS system.

7 CONCLUSIONS AND FUTURE WORK

In this paper we present measurements of TCP over GPRS when GPRS traffic is preempted by circuit-switched calls with higher priority. In this situation data that is buffered in the BSC is immediately discarded when the GPRS resources are lost. New data, either in the form of a retransmitted segment or in the form of an acknowledgment buffered in the GPRS terminal, is needed to start up the transfer again after the preemption period ends. Since several TCP segments get lost when the BSC buffer is flushed it takes a substantial amount of time to recover even for short preemption periods. The TCP performance could potentially be improved if the BSC waited for a short time before it flushes its buffer, and if data buffered in the SGSN was transmitted as soon as GPRS resources became available again after a preemption period.

Our plans for future work include further experiments on the impact of GPRS buffering and how it affects various application models. Measurements of mobility would allow for interesting comparison with the results from the preemption measurements.

ACKNOWLEDGMENTS

The authors would like to thank Bojne Svensson and Jonas Eriksson at Telia Mobile AB. This work has been partly funded by the Swedish graduate school PCC++ and the research program CMIT.

REFERENCES

Allman, M., Paxson, V., and Stevens, W. (1999). Rfc 2581:tcp congestion control.

Bettstetter, C., Vgel, H.-J., and Eberspcher, J. (1999). Gsm phase 2+ general packet radio service gprs: Architecture, protocols, and air interface. *IEEE Communication Surveys*, 2(3).

Chakravorty, R., Cartwright, J., and Pratt, I. (2002). Practical experience with tcp over gprs. In *In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '02)*, pages 1678–1682, Taipei,Taiwan.

Chen, W.-Y., Wu, J.-L. C., and Liu, H.-H. (2002). Performance analysis of radio resource allocation in gsm/gprs networks. In *In Proceedings of IEEE Vehicular Technology Conference (VTC-02 Fall)*, pages 1461–1465, Vancouver, Canada.

Ethereal (2004). A network protocol analyzer. Retrieved April 26, 2004, from http://www.ethereal.com/.

GSM03.60 (2000). Digital cellular telecommunications system (phase 2+). General Packet Radio Service(GPRS), Service description, Stage 2 (GSM 03.60 version 6.7.0 Release 1997).

GSM04.60 (2000). Digital cellular telecommunications system (phase 2+). General Packet Radio Service (GPRS), Mobile Station (MS) - Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol (GSM 04.60 version 6.9.0 Release 1997).

GSM08.18 (2000). Digital cellular telecommunications system (phase 2+). General Packet Radio Service (GPRS), Base Station System (BSS) - Serving GPRS Support Node (SGSN); BSS GPRS Protocol (BSSGP) (GSM 08.18 version 6.7.1 Release 1997).

Gurtov, A., Passoja, M., Aalto, O., and Raitola, M. (2002). Multi-layer protocol tracing in a gprs network. In *In Proceedings of IEEE Vehicular Technology Conference (VTC-02 Fall)*, pages 1612–1616, Vancouver, Canada.

Inamura, H., Montenegro, G., Ludwig, R., Gurtov, A., and Khafizov, F. (2003). Rfc 3481: Tcp over second (2.5g) and third (3g) generation wireless networks.

Jacobson, V., Braden, R., and Borman, D. (1992). Rfc 1323: Tcp extensions for high performance.

Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A. (1996). Rfc 2018: Tcp selective acknowledgment options.

NetHawk (2004). Nethawk gsm analyser. Retrieved April 26, 2004, from http://www.nethawk.fi/.

Ostermann, S. (2004). Tcptrace. Retrieved April 26, 2004, from http://www.tcptrace.org/.

Postel, J. (1981). Rfc 793: Transmission control protocol.

Sagfors, M., Ludwig, R., Meyer, M., and Peisa, J. (2003). Queue management for tcp traffic over 3g links. In *In Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '03)*, pages 1663–1668, New Orleans, Louisiana, USA.

Sarolahti, P. and Kuznetsov, A. (2002). Congestion control in linux tcp. In *In USENIX Annual Technical Conference, FREENIX Track*, pages 49–62, Monterey, California, USA.

Taferner, M. and Bonek, E. (2002). *Wireless Internet Access over GSM and UMTS*. Springer-Verlag.