TCP/IP Performance over EGPRS network

Rafael Sánchez, Julia Martinez, Javier Romero, Rauli Järvelä
Nokia Networks, C/ Severo Ochoa, s/n Edif. de Inst. Universitarios, Pl 3.
Parque Tecnológico de Andalucía – 29590 Campanillas (Málaga) Spain
E-mail: ext-rafael.m.sanchez@nokia.com, julia.martinez@nokia.com, javier.romero@nokia.com , rauli.jarvela@nokia.com

***Abstract*— The Performance of TCP/IP based applications over 2.5 and 3G networks presents problems due to long bandwidth-delay products, link errors and long delay variance. Specific characteristics of underlying technology like link level error recovery algorithms or quality of service procedures have also important influence on the behaviour of TCP layer and affect to the end to end performance. In this paper simulation results for EGPRS network are presented from an end-to-end dynamic simulator that models TCP Reno with some additional features, core network delays and buffering, GPRS RLC acknowledged mode and link level interference calculations.**

Keywords—TCP/IP; GPRS; EGPRS; Performance;

I. INTRODUCTION

It is clear nowadays that the future of the telecom companies, operators and manufacturers, is tightly joined to the explosion of wireless data services and mobile Internet.

TCP/IP is the most extended transport layer protocol over Internet, and most of the non real time services such as Web Browsing, FTP, or E-mail make use of it. Since the Transmission Control Protocol was officially adopted as Internet standard in 1983, this protocol has been developed in order to optimize its performance over fast and reliable wired networks. The problem arises when wireless networks appear. Some of the assumption that were considered when designing the protocol procedures, very low packet loss probability and slow variance of the delays, are not valid anymore.

In the case of EGPRS, reliable link level protocol (RLC) takes care of doing retransmissions and the problem of data losses is translated into variable delays.

There are several studies that investigate the effect of variable delays over TCP retransmission procedures and propose ways to improve the protocol performance, (see [1], [2]). Improvement of the protocol starts with parameters optimization and continues with new features added to the standard or the inclusion of intelligent elements inside the operator's core network such as Performance Enhanced Proxies (PEPs).

This paper focuses on TCP parameters at both sides of the connection, and their optimum values in a EGPRS scenario. Throughput and delay curves are presented for different TCP parameter configurations. The simulator used for the study provides a multi-cell environment where the link level characteristics of the system are modeled in detail and also mobility of the terminals is considered. TCP Reno type is modeled and application layer protocol used are FTP or Web browsing.

The rest of this paper is organized as follows. In section II, the TCP protocol considered in the model is described. In section III, some concepts of interest for the study are presented. Section IV describes the whole simulation model organized in different layers. In sections V and VI simulation results for FTP and Web browsing are commented. Finally, in section VII results and main conclusions are presented.

II. TCP DESCRIPTION AND PARAMETERS

TCP is a connection oriented transport protocol that provides reliable end-to-end communication on top of a datagram network. In this section the main procedures and parameters included in the model are presented.

The TCP sender divides the data into *segments* that are transmitted and later acknowledged by the receiver. The standard defines different mechanisms that allow to achieve reliable transmission .and avoid overflow of the network capacity. The TCP-Reno includes slow start, congestion avoidance, delayed acknowledgement, fast retransmit and fast recovery algorithms [4]. In table I some basic definitions of TCP parameters used in the different algorithms are presented.

TABLE I.TCP PARAMETERS

Name	Description
MSS	Maximum Segment Size for a connection. Some typical values are 536 and 1460 Bytes.
Congestion Window (cwnd)	Variable that limits the amount of data that a TCP connection can send. Congestion control algorithms will change this value during the TCP connection.
Advertised Window (awnd)	Maximum value of the transmission window indicated by the receiver.
Slow Start Threshold (ssthresh)	Threshold value that sets whether to use *slow start* or *congestion avoidance* as congestion control algorithm.
RTT	Round Trip Time. Time elapsed from segment transmission to acknowledgement.
RTO	Retransmission Time Out. Time that TCP layer waits before starting retransmission procedure.

A. TCP Acknowledgments and Delayed Ack

The reliability of TCP protocol is obtained by an acknowledgment scheme that tries to confirm the reception of every data segment. One acknowledgment message confirms the reception of the last data segment received in sequence. Whenever out of sequence data is received, a duplicated acknowledgment is sent back to the receiver.

The number of Ack packets is reduced with the usage of the delayed Ack procedure, which delays the transmission of the acknowledge information for 200 ms or till two data segments are received.

B. Congestion Control

TCP probes network capacity and avoids congestion by limiting the maximum amount of data that can be under transmission in a certain moment.

After connection establishment, the TCP sender begins the transmission in the *slow start* phase with the initial value of the congestion window set to one MSS. The cwnd is increased by one MSS after reception of every acknowledgement packet, producing an exponential increase of the transmission window that last till congestion is detected or the maximum size, set by the advertised window, is reached.

When congestion is detected, the cwnd and slow start threshold are reduced. When cwnd is higher than the ssthres, the *congestion avoidance* algorithm is used, and the growth rate of the congestion window will continue but with a linear evolution.

$$\mathrm{cwnd} = \mathrm{cwnd} + \frac{\mathrm{MSS} \cdot \mathrm{MSS}}{\mathrm{cwnd}} \qquad (1)$$

C. Retransmissions

TCP tries to recover from data losses with two different mechanisms. First one is a retransmission timeout that is started for the first unacknowledged segment in the transmission window, and reset every time an acknowledgment is received that acknowledges new data. If the retransmission timer expires, all the outstanding data is retransmitted. This event is also interpreted as a congestion signal, so the congestion window is reset to its initial value and TCP goes back to *slow start* phase.

The second one is the Fast Retransmit/Fast recovery algorithm, that tries to detect data losses before the retransmission timer expires. In this case, three duplicated acknowledgments are interpreted as a signal that one segment has been lost, and the retransmission is done without waiting for a time-out. When new data is acknowledged, the congestion window is halved and the transmission continues in *congestion avoidance* phase.

D. Retransmission timer estimation

During data transmission the sender measures the Round Trip Delay of the data segments and estimates the retransmission time-out (RTO) with an adaptive algorithm described in [6]. The initial value of the RTO and the accuracy of the measurements is very much important for the performance of the protocol.

RTT measurements from retransmitted segments are nor reliable, because TCP is not able to differentiate between the acknowledgment for the original segment or the retransmission. For this reason the unreliable RTT samples are skipped and the RTO doubled every time retransmission timer expires (Karn's algorithm).

III. Parameters Affecting TCP Performance in Wireless Networks

In this section three important concepts that are considered when configuring TCP parameters in sections V and VI are presented.

A. Bandwidth Delay Product (BDP)

The BDP is a merit figure that gives information about the minimum amount of data that shall be outstanding in the connection in order to provide optimum utilization of the available throughput.

$$\mathrm{BDP(kbit)} = \mathrm{BW(kbit/s)} \cdot \mathrm{RTT(s)} \qquad (1)$$

The flow control mechanism of TCP layer shall be configured to a value closed to the BDP in order to guarantee that bandwidth is fully used and protect the buffers from overflow. In table II some examples of BDP for EGPRS are shown.

TABLE II. Bandwidth Delay Product for different codings

Coding Scheme	**Nominal BW/TSL (Kbit/s)**	**Nr TSL**	**BDP (KB) RTT=0.5s**	**BDP (KB) RTT=1s**	**BDP (KB) RTT=1.5s**
MCS-9	59.2	1	3.70	7.40	14.80
MCS-9	59.2	3	11.10	22.20	33.30
CS-2	12	1	0.75	1.5	2.25
CS-2	12	3	2.25	4.5	6.75

B. Initial Congestion Window

The default value of the initial congestion window is 1 MSS, but in the case of using delayed ACK it is recommended to use at least 2 MSS in order to compensate the extra 200 ms delay at the beginning of the connection. The standard allows to use up to 4KB, what could even hide the effect of slow start in cases of low BDP (see table II). However this option must be considered with care because of the drawback effect that it can have over a congested network.

C. Spurious Retransmissions

Spurious retransmissions, described in [2], are produced because of a bad estimation of the RTO, that makes the retransmission timer to expire and proceed with the unnecessary retransmission of whole window. There are two main reasons for this problem:

1- low number RTT samples due to retransmitted segments that are not valid for RTT samples.

2- Sudden changes in the RTT produced by cell reselections or Quality of Service (QoS) procedures.

For the first problem some solutions are proposed, like ECN (Explicit Congestion Notification) and Eifel algorithm [2], but they are not considered in this model. For the second, conservative values of the RTO can reduce the situations in which spurious retransmissions appear.

IV. SIMULATOR DESCRIPTION

The simulator has been built so end-to-end performance of user connections can be measured at network and application level. The link layer calculations, mobility and cellular network setup is performed by a widely tested dynamic network simulator, that is connected to new layers which introduce the delay modeling of the operator's core network and Internet. Detailed implementation of TCP layer, as described in section II, and application level protocols are added on top of the simulator to complete the model, as described in figure 1.

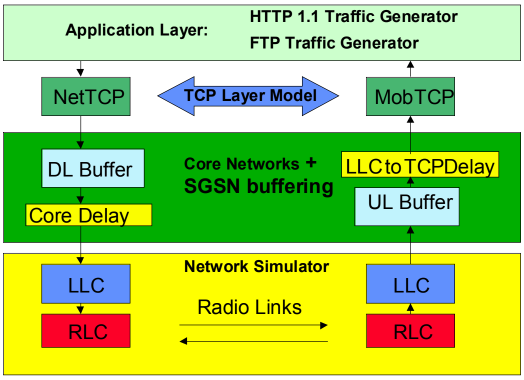

Figure 1. Structure of the simulator.

A. Network Simulator

This entity defines a regular simulation scenario with tri-sectorized cells where a number of mobile terminals are allocated and moving with a defined speed. Each terminal is performing packet switched calls with a certain periodicity.

The main configuration parameters that define the simulation scenario can be found in table III.

TABLE III. EGPRS NETWORK DESCRIPTION

Element	Element Description
Scenario	75 Cells with 500m of cell radius. 3-sector sites
Antennas	Directional (65 degrees radiation pattern)
GPRS Territory	2 TRXs per BTS. *7 TSL* available for GPRS in each BTS. GPRS Territory does not include BCCH.
EGPRS Connections	RLC layer works in Acknowledged mode. Link level using Incremental Reduncancy and Link Adaptation with MCS-1 to MCS-9.
Time-Slot Capacity	Channel allocator assignes *3 Time Slots* in downlink direction and *1 Time Slot* in uplink.
Power Control	Downlink and Uplink power control activated.
Reuse	Tight reuse, 1/3 for TCH. Reuse 5/15 in BCCH.
TCP configuration	MSS = 1460B; AWND = 64 KB; Initial CWND = 2MSS; SGSN Buffer size = 64 KB; delayed ACK = 200ms

The EGPRS network behavior is based on Release '99, with *best effort* management, where all the connections are treated with the same priority.

B. Core Network and SGSN buffering

Core network is normally an IP network that connects different nodes of the operator's system. For packet switched services the basic element is the Serving GPRS Support Node (SGSN), and for connection with public Internet, the Gateway GPRS Support Node (GGSN) is used.

The effect of core network is modeled as an extra delay to the TCP segments. This delay is calculated based on the bandwidth of the links, packet sizes and real measurements. Furthermore, the effect of buffering delay and packet dropping in SGSN is also considered.

C. TCP Layer Description

TCP layer is modeled as described in section II for client and server.

D. Application Layer Description

Two basic services have been modeled at application level for this study: FTP and Web browsing.

1) FTP: A normal FTP session has one control connection where the commands and responses are sent, and a data connection that is created for the transmission of each file. In the model only the data connection is considered. A file download consists in one TCP connection, where a request is sent from receiver to sender, and whole file is sent to the client.

2) HTTP 1.1: A Web page is described as a main page that contains the URL addresses of a number of embedded objects. Several studies about Internet traffic have built different models for size distributions of these elements. In the simulator we have chosen the one described in [7]. The parameters and distributions for this model are shown in table IV.

Transmission at application layer consists in a first request and download of the main page. Once the main page is received, the client starts requesting the rest of the objects. When HTTP1.1 is used, following mechanisms are included which improve the performance from previous versions of the protocol [8].

a)persistent connection: Instead of establishing a new TCP connection for each object in the web page, one connection is kept alive during the whole download.

b)pipelining: By default request for new objects are not sent till previous object has been received. This adds an extra RTT delay for each object download. When this option is activated, several requests can be sent, so the server is able to start the transmission of next objects without further delays.

TABLE IV. WEB PAGE DESCRIPTION

Element	Distribution	Parameters
Main page size	LogNormal	Mean = 10710 bytes std. dev. = 25032 bytes
Embedded object size	LogNormal	Mean = 7758 bytes std. dev = 126168 bytes

Element	Distribution	Parameters
Number of Embedded objects per page	Truncated Pareto	Mean = 5.5 Max = 55
Reading time	Exponential	Mean = 30 s
Request size	Fixed	250 bytes.

V. FTP Simulation Results

In this section it is tried to determine how different TCP parameter configurations affect to the end to end performance of the users in a EGPRS network. For this purpose an standard configuration has been chosen as described in table III, and the TCP parameters have been tuned to different values.

A. *Effect of Slow Start over different file sizes*

During the *slow start* phase the transmission is limited by the TCP congestion control and total bandwidth of the link is not reached till the transmission pipe is filled. This reduction is specially significative when the size of the file is small because the time spend in slow start has more significative wait over the whole transmission time.

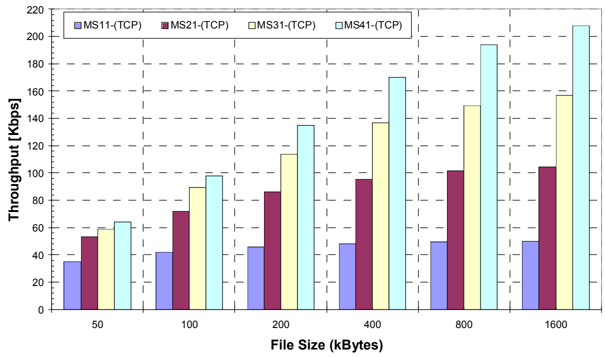

Figure 2. Application Throughput Vs. File size and multi-slot capacity.

In figure 2 it is shown the ideal throughput obtained for one single terminal with different multi-slot capabilities, under ideal radio conditions (CIR = 100 dB) and absence of time-slot multiplexing. For each file size the throughput obtained for mobiles with 1, 2, 3 and 4 time-slots in the downlink is shown. It can be observed that the throughput reduction caused by the slow start is more significative as higher is the available bandwidth. The reason is that when the BDP is high there are needed more cycles to fill the link capacity. For example, in the case of 1 time-slot, the effect of slow start is null because the BDP is already fulfilled by the initial congestion window.

B. *Effect of Maximum Segment Size (MSS)*

Maximum segment size in TCP connections is determined by the size that sender and receiver can manage. This value is indicated in SYN messages during connection establishment phase.

From performance point of view, large MSS reduces packet overhead and allows faster increase of transmission window, what means that full utilization of available throughput is reached earlier. On the other hand, if some router of the path has a MTU smaller than the selected MSS, IP fragmentation would be needed with an important impact over performance.

In real networks, in order to use optimum MSS it is recommended to use the MTU Discovery procedure [RFC1191], that allows the sender to check the maximum packet size without fragmentation. In this model the value of MSS is set as parameter.

In (E)GPRS the maximum value for MSS is 1460 Bytes that added to the 40 B of TCP/IP header, fits in one LLC frame. In figure 3 simulations have been performed with the same amount of input traffic but different values of MSS. In the figure it is possible to see how throughput is reduced 30-40 % with a MSS of 536B and 50-70% for 256B.

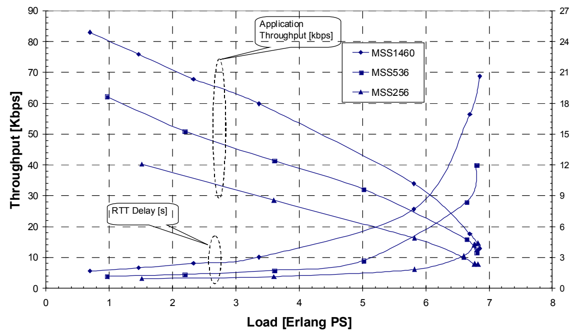

Figure 3. Application Throughput & RTT for different MSS sizes.[1]

C. *Effect of the Advertised window*

The receiver shall configure a suitable value for the advertised window that guarantees that the bandwidth is fully used. For this simulation case, values of 8, 16, 32 and 64 KB are chosen for the advertised window in order to see how it affects to the performance.

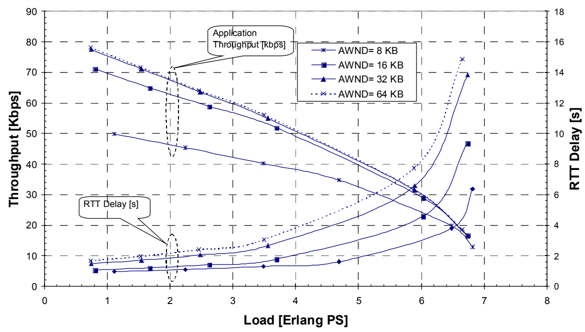

Figure 4. Application Throughput & RTT for different AWND sizes.

Assuming a RTT between 1 and 1.5 seconds, and according to table II, the bandwidth delay product should be between 22.2 and 33.3 KB. Looking into figure 4, curves for 8 and 16 KB show a reduction in the throughput, while for the case of 64KB, there is not better performance than fore 32 KB. In this case the limitation is imposed by the capacity of the link.

---

[1] 1 Erlang PS means one time-slot fully used by PSW traffic.

D. Application throughput Vs. cell reselection delay

In this case the mobile speed was incremented to 50 Km/h and the handover control parameters modified so the number of cell reselections per connection could be increased to 2-3. Then, for the same simulation scenario, the delay produced by cell reselection is changed to 1, 2 and 3 seconds, and the results showed in figure 5.

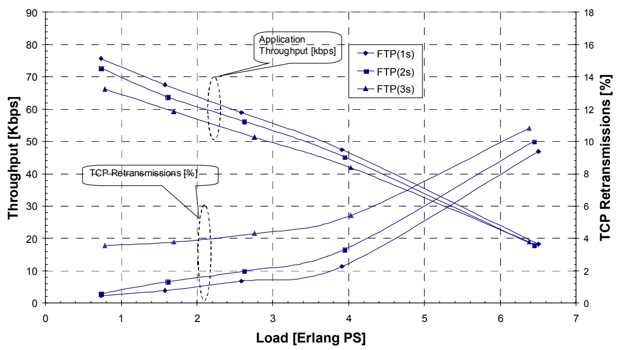

Figure 5. Application Throughput & TCP Rtx. Vs. Cell-reselection delay.

Performance is reduced if cell- reselection delay is longer, but not many spurious retransmissions are observed. Only in case of 3s cell reselection delay the TCP retransmissions are significantly increased. Assuming the absence of lost frames in the model, the main cause of spurious retransmissions is the cell-reselection delay. However, the initial value of 3s of the RTO, and the long RTT values, keeps RTO into a range of values that can support a 2 or 3 seconds stop in the transmission. Also, in the case that timer expires, after first retransmission timeout the RTO is doubled by Karn's algorithm, and further cell reselections are not easily causing more unnecessary retransmissions for that connection.

VI. Web Browsing Simulation Results

With web browsing traffic and the configuration defined in table III the following throughput curves are obtained for different web page sizes and network load.

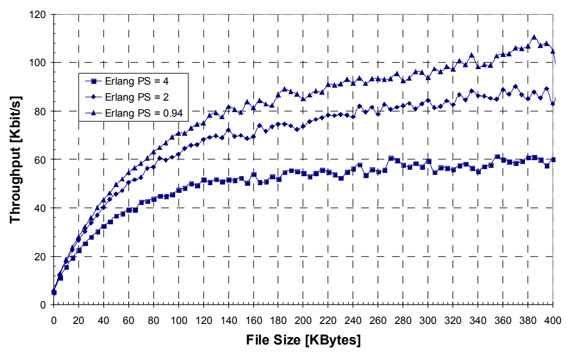

Figure 6.Application Throughput Vs. Web Page Size.

In figure 6, three throughput curves are presented for different loads. For small pages, the throughput depends on the effect of the slow start, which limits the transmission more strictly than link capacity. For longer pages, the throughput tends to be stable as the effect of slow start is less significative and maximum link capacity is reached.

VII. Conclussions and Further Work

In this paper it has been presented end to end performance figures of Internet applications over wireless network.

Slow start is specially noticeable when small files are transmitted because the full link capacity is not used during an important part of the transmission time. The effect is more important the higher it is the value of the bandwidth delay product.

Large MSS allows shorter slow start phase, better bandwidth utilization and reduces the TCP overhead. For example, MSS of 1460 Bytes can increase application throughput in more than 30% compared with an MSS of 536 B. The MTU discovery procedure can be used to obtain the optimum value for the MSS.

An initial congestion window of 2MSS is enough for a slow connections to reduce the effect of slow start (see table II). A 4KB initial window increased the throughput in 5%, but it could have a drawback impact on congested network.

The impact of cell reselections comes from the extra transmission delay and the possibility to produce spurious retransmissions.. RTO has been shown to be conservative enough to avoid these retransmissions, and only in the case of 3 s delay, the statistics showed some retransmission increase.

Further investigations in this subject will consider the effect of packet losses in during cell reselection procedure as a possible cause of spurious retransmissions, and Selective Acknowledgement (SACK) as protocol improvement to deal with multiple losses.

Acknowledgment

This work has been performed as part of the cooperation agreement between Nokia and the University of Malaga. This agreement is partially supported by the Program to promote technical research (Programa de Fomento de la Investigación Técnica. PROFIT) of the Spanish Ministry of Science and Technology.

References

[1] Gurtov Andrei, "Effect of Delays on TCP Performance," *Proceedings of IFIP Personal Wireless Communications '2001, August 2001.*

[2] R. Ludwig, R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," ACM Computer Communication Review, Vol. 30, No. 1, January 2000.

[3] T.Halonen, J. Romero, J.Melero, GSM, GPRS and EDGE Performance, Efolution Towards 3G/UMTS, Wiley, 2002-06-19.

[4] M. Allman, V. Paxon. "TCP Congestion Control". [RFC2581].

[5] R. Braden. “Requirements for Internet Hosts – Communication Layers.”. [RFC1122].

[6] V.Paxon. “Computing TCP’s Retransmission Timer”. [RFC2988]

[7] Choi, H. and Limb, J. A, “Behavioral Model of Web Traffic”, International Conference of Networking Protocol 99’ (ICNP 99’), September 1999.

[8] R. Fielding, J. Gettys, J. Mogul, et. al.“Hypertext Transfer Protocol – HTTP/1.1”. [RFC2616]

[9] Pablo Ameigeiras, Jeroen Wigard, Preben Mogensen, “Impact of TCP Flow Control on the Radio Resource Management of WCDMA Networks” Vehicular Technology Conference. VTC Spring 2002.