# A transport protocol to improve access to data networks from GSM using a proxy

J.M. Rivadeneyra Sicilia[1] and J. Miguel-Alonso[1,2]

[1]Departamento de Arquitectura y Tecnología de Computadores
Universidad del País Vasco / Euskal Herriko Unibertsitatea
P.O. Box 649, 20080 Donostia-San Sebastián, SPAIN

[2]School of Electrical and Computer Engineering
Purdue University
1285 Electrical Engineering Building, West Lafayette, IN 47907, USA

E-mail: {pepe, miguel}@si.ehu.es. Tel. +34 43 018039. Fax +34 43 219306

## Abstract

In order to access TCP/IP networks (mainly the Internet) through GSM links,. we have designed an architecture based in the indirect model of client/server interaction, where an intermediate element acts as a bridge between GSM and the rest of the data network. We have replaced the TCP connection in the client-proxy link with a simpler protocol. This allows maintaining the protocols in server machines, and requires minimal changes in those running in the mobile clients. In this paper we describe STP protocol and present some measurements of the performance obtained with it, compared to conventional TCP/IP architecture and a sort of indirect TCP based architecture. The obtained results show that our approach improves the performance achieved with conventional TCP/IP and indirect TCP, while maintaining a high level of application compatibility.

## 1. Introduction

The convergence of developments in mobile communications and personal computing are opening new frontiers to information access, creating the concept of nomadic user: that one who moves along with a portable computer and requests, wherever he or she happens to be, remote access to information services. Different technologies are being developed to support the new services that will arise in the different scenarios of mobile computing, ranging from a single wireless LAN to a heterogeneous system such as UMTS, which will offer universal connectivity and service integration. Among these scenarios, the one we are interested in consists of three main components: (a) a fixed data communication network, where servers are located (typically the Internet), (b) a mass of portable computers where the clients are located, requesting access to the servers from changing locations, and (c) a wireless access to the fixed network through a cellular telephone system. We will focus on the GSM [1] (Global System for Mobile Communications) technology for the wireless access network, as it currently forms one of the most comprehensive digital mobile phone systems, thus being able to provide data networks access to a wide range of nomadic population. The integration of GSM with other digital technologies such as DECT cordless phone systems and satellite services guarantees that this situation will be maintained in the future.

The main advantages of using cellular phone networks for data access are:

1. Those networks already exist, and are increasingly familiar to the user; data transmission becomes just another facility of the mobile phone.
2. They are able to provide ubiquitous access to information.
3. Mobility issues are managed by the network.

On the other hand, the characteristics of cellular telephone networks are quite different from those of wired links. Cellular links have high latency with long, variable delays, a low throughput, and are prone to sudden disconnection. That is why, in practice, applications used over cellular and wired networks are still different: wireless links create problems for existing applications designed for fast and reliable connections. Success is restricted to off-line applications requiring a very reduced bandwidth, like electronic mail or SMS (short messages). These are, though, just 'minimal services', and not what the user would expect from the network.

In this paper we present the design of a framework to provide the functionality required to run standard Internet applications using a cellular system (GSM) as the access medium, with an acceptable performance level. The milestone of the design is STP, a transport-level protocol to be used in the wireless link. The rest of the paper is organized as follows. First, we introduce the current and future use of the pan-European GSM system for data transmission, and the main problems that arise when using it as a platform for TCP/IP-based applications. Next, we give an overview of the designed communications architecture (section 3) and describe STP (section 4). In Section 5 the experimental measurements obtained with an implemented prototype are presented. Finally, Section 6 contains concluding remarks and some guidelines for future work.

## 2. GSM data service

### 2.1 Internet access using GSM

Nowadays, GSM supports circuit-switched connections, with a single error-correcting link per handset. The nominal

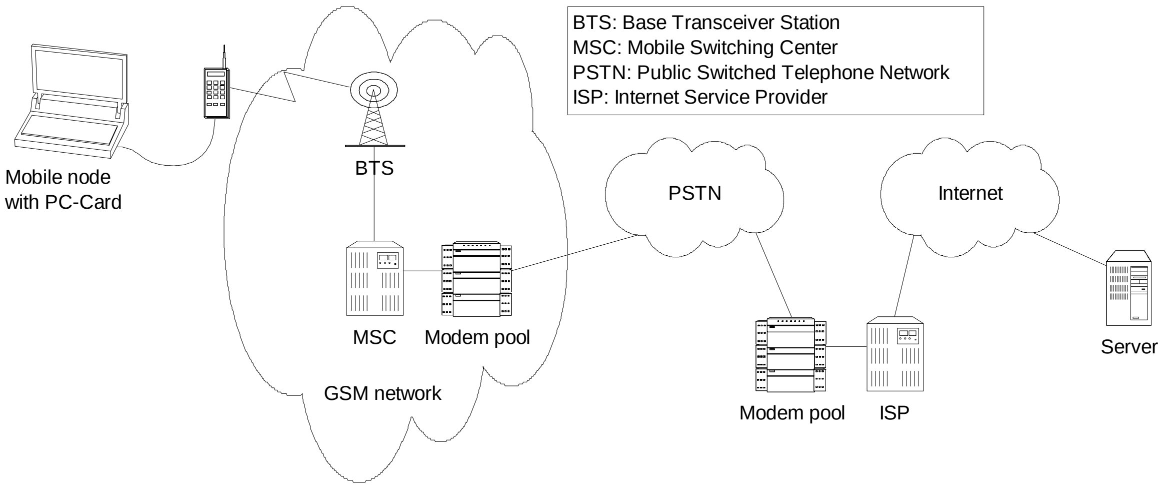



***Fig. 1.*** *Conventional access to Internet through GSM networks*

transfer rate is 9.6 Kb/s, far from the 56 Kb/s of the wired telephone network (in optimal conditions). Besides, actual sustainable transfer rates handled are even lower: between 5 and 7 Kb/s for bulk data transfer over a good quality GSM link [2][3]. The use of all-digital connections from GSM to ISDN and data compression techniques will make that rate to go up, but for some considerable time most traffic will continue to be handled by the PSTN. Figure 1 gives an overview of the use of a GSM network to access the Internet. Three different networks are traversed in the way from mobile client to fixed server (and vice-versa): GSM (digital, circuit switched), PSTN (analog, circuit switched) and Internet (digital, packet switched). GSM specifies two modes of asynchronous bearer service: transparent and non-transparent. In the first one the bit error rate on the radio link, within normal coverage, is assumed not to exceed the level of $10^3$ at the line speed of 9.600 b/s and after mere forward-error correction. The non-transparent mode is implemented using the Radio Link Protocol (RLP), a member of the HDLC family. RLP includes error correction mechanisms, using selective retransmissions, which reduce the average bit error rate down to $10^8$ [4]. The price to pay is an increase in latency, while throughput and transmission delays become variable, remarkably so under poor conditions of the air link.

Figure 2 shows the protocol architecture used of this access. It is assumed, as it is common practice, that the non-transparent GSM data service is chosen.

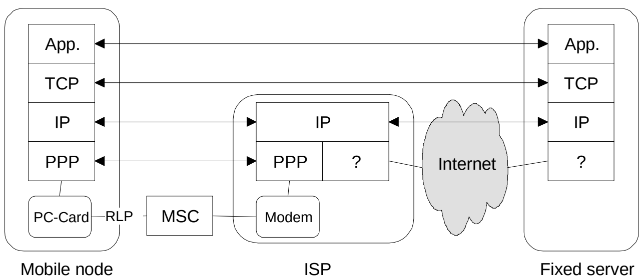



***Fig. 2.*** *Protocol architecture of a conventional connection to Internet through GSM*

## 2.2 Problems for TCP/IP applications

The first problem is the low bandwidth of the GSM link. Current applications need an increasingly available bandwidth in the access and backbone networks, because of the general use of web and multimedia applications. Partial solutions to this problem are described in the next section.

The second source of problems is the inadequacy of some aspects of the TCP protocol for wireless networks in general [5][6][7], and for GSM access in particular [8]. The problem is that TCP behavior is not appropriate when the delay between sending a segment and receiving the corresponding acknowledgement is variable. It was designed for relatively fast and reliable fixed networks where that delay keeps stable, unless a congestion appears. So, in the case of an excessive delay when receiving an ACK, TCP interprets it as a symptom of congestion, and reacts in consequence. When a GSM data link is present, irregular delay in the incoming ACKs is not caused solely by congestion but, mainly, by other sources, inherent to the low bandwidth and error-prone radio link:

1. The high error rate of the radio link hardly enforce any retransmission at transport layer, because of the use of RLP at link layer, which will do those retransmissions in a more efficient way. But RLP retransmissions are reflected in irregular round trip times registered at transport layer for successive segments, often triggering congestion control algorithms.
2. When multiple simultaneous TCP connections are present in a GSM access (the usual case in a web access), all those connections have to share the single dial up access and related buffers. Different connections will mutually interfere in the use of the outgoing buffers in the network interface, causing additional, unpredictable delays. Again, these delays may be misinterpreted by the TCP sender as packets lost in a congestion.

This behavior of TCP will affect negatively the throughput of the connection in different ways:

- It causes unnecessary retransmissions.

- The slow start [9] algorithm is triggered. It has two negative effects in the throughput of the connection: it reduces to one the sending window (specially harmful for short connections, as it neutralizes the benefits of using windows in long latency links, as it is a GSM link), and the unnecessary slow growing of that window, after the retransmission.

A third problem of classic TCP over cellular networks is the large amount of redundant control information in the segment headers, which wastes a good deal of the available bandwidth. Most of the 20 header bytes in a TCP segment are devoted to error and congestion control. The same can be stated about the complex mechanism used to open connections, the so called *three way handshake*. But neither errors (at transport layer) nor congestion are present in a GSM link. Some techniques are already available to reduce the amount of control information in TCP headers sent through SLIP (CSLIP) or PPP.

### 2.3 Future data services in GSM

In order to achieve higher speeds, new parts of the GSM standard has been developed. HSCSD (High Speed Circuit-Switched Data) boost user capacity from 9.6 Kb/s to 14.4 Kb/s in a single data channel. Besides, it allows to combine some TDMA time slots of the 200 kHz GSM carrier to provide a single data circuit [10]. At most 4 slots can be combined, getting up to 57.6 Kb/s. The service can be asymmetric, offering a different transmission capacity for downlink and uplink. For example, a 3+1 service can be very adequate for applications where a higher information volume flows downlink, as it is the case of web access. HSCSD does not suppose important changes in the network infrastructure for telephone carriers, so it is being increasingly offered. In the user side, it is necessary to use new terminals, able to use a different code scheme and to use more than one time slot per channel.

Another new possibility for GSM data transmission, specially useful for high-bandwidth traffic, is GPRS (General Packet Radio Services) [11], a packet-switching technology well suited to the highly bursty nature of most data applications. It will also use multislot techniques in the radio access, handling variable peak data rates and higher access data rates than HSCSD. It introduces important changes in the infrastructure of backbone network, that will be an IP network, and in the link that connects base stations and MSC (SGSN in GPRS), a Frame Relay circuit. The access from GPRS to other packet-switched networks, like the Internet, is made in the usual way through gateways (GGSN in GPRS naming). A special set of gateways, called Border Gateways, connects interconnect separate GPRS networks. Fixed IP addresses can be used by a computer connected to GPRS, making possible, in combination with the higher bandwidth available, to connect a mobile server to the network.

We still do not know how TCP/IP applications will behave in these new environments. It can be supposed that the problems inherent to the high BER of the air link will remain, but experimental measurements are needed.

### 2.4 WAP architecture

In December 1997 the Wireless Application Protocol (WAP) Forum was legally constituted [12], with the participation of important companies from the telecommunications and computer industries. Their objective was to develop a worldwide standard for providing Internet communications and advanced telephony services on digital mobile phones, pagers, personal digital assistants, and other wireless terminals. It is not focused on GSM bearer services, but this technology is considered as one of the options to put on it the WAP protocol stage. It is mainly thought for low-capacity terminals, but the designed architecture [13] can be exploited to provide remote access to Internet for portable computers as well.

The WAP architecture can be an indirect architecture, when a proxy is located between the mobile client and the fixed server (a proxy WAP). That is the case when the server doesn’t support WAP protocols. The proxy is not required if there is a WAP server. At the transport layer, WAP provides a connectionless, unreliable datagram service (WDP – Wireless Datagram Protocol), replaced by UDP when used over an IP network layer. Over that transport protocol, WAP defines a Transaction Layer (WTP), that provides reliable data transfer based on the request/reply paradigm. This approach is similar to the architecture presented in this paper, because it is also an indirect design. But the WAP architecture is much more general and, therefore, complex. The number of layers is larger (it has four layers above the transport layer) and the functionality of them is different. When used over TCP/IP networks (the usual case) the transport protocol used is UDP, leaving error control to upper layers. In our design we propose a new transport service over IP and down the applications, a reliable service but not as complex as the one offered by TCP.

## 3. Proposed architecture

### 1.1 Objectives

Our target is to design an architecture that provides a better interaction between mobile clients in a GSM network and fixed servers in Internet, improving the current situation in many aspects, but mainly in the use of the scarce bandwidth of the radio link. The design must consider these restrictions:

- Compatibility with TCP/IP protocol stack. The interaction with a mobile client must be invisible to the fixed server.
- Improvement in the throughput as perceived by the applications.
- Easy implementation in current and future telecommunication infrastructures, minimizing investment and efforts.

### 3.2 General overview of the architecture

Our proposal (see Figure 3) is an indirect architecture, with client-server communications divided in two parts: the first one for the cellular network, controlled by special network protocols adapted for the wireless environment, and the other one for the rest of the connection, managed by conventional TCP/IP protocols. In the frontier of both sides is the intermediary (from now on just *proxy*). The proxy could be located in several places [3]; here we propose to locate it in the same machine that plays de role of remote access server of an Internet service provider. The user will be connected to it by a circuit switched connection (it will be this way until GPRS becomes available). The rest of the communication, from the proxy to the server, uses packet switching. Therefore, the proxy is located in the border between the GSM network and the Internet. A more detailed description of the architecture can be found in [14][3]. Next we will have a look just to those levels that change when comparing Figures 2 and 3, with special attention to STP, a new transport protocol that plays a fundamental role in our proposal.

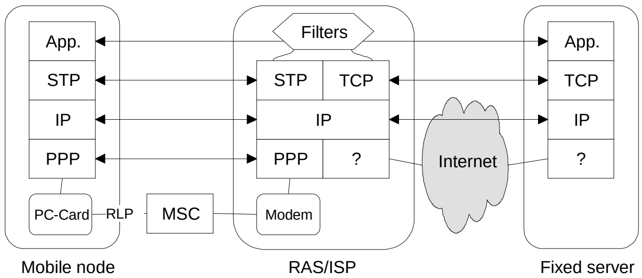



***Fig. 3.*** *Protocol architecture with an intermediary located at the ISP*

*Transport protocol*

In the circuit-switched part of the communication (the one that includes the wireless link), TCP is replaced by a simpler transport protocol, keeping a regular TCP connection for the packet-switched part (from proxy to server). Splitting the TCP connection at the proxy, the source of problems, i.e., the variable-delay wireless link, is removed. Therefore, we can expect "normal" throughput in the TCP connection from proxy to server, avoiding unnecessary retransmissions and their pernicious effects, and an efficient use of the scarce available bandwidth in the GSM link. Here, RLP should provide an error and congestion free data interchange service, so a full-fledged transport protocol like TCP is not required—a simple mechanism to interchange data should be enough. Unfortunately, the BER registered in non transparent GSM links, as perceived by IP layer, can be much higher than the maximum $10^{-8}$ specified by the standard [3], making necessary to add error correction mechanisms. The resulting protocol is STP (Simple Transport Protocol), described in next section.

*Proxy level*

The use of the indirect model at the application layer means to interpose application specific programs, called filters, in the middle of client-server connections, running in the proxy. These filters, and its management, is what we call the proxy level.

## 4. The Simple Transport Protocol

Being a very simple protocol, we don't use any formal language to describe it. Instead, we follow the informal style of Internet RFC's. Sometimes we mention implementation aspects to make clear why some decisions on STP definition have been made.

### 4.1 Service description

STP provides a connection oriented, error free service, on top of non-transparent GSM circuit-switched connections. Errors are corrected at the link layer by RLP, so it is not necessary to do it in our transport protocol. However, it incorporates an optional mechanism for error recovering, to be used when the BER of the GSM line is superior to the standard ($10^{-8}$). Anyway, is not recommended to use our protocol in very noisy environments (registered BER $>10^{-5}$).

Connections are identified in the same way as they are in TCP, this is, by the two end points of the communication. An end point is identified by the IP address of the machine and the port number used by the application. A connection is always initiated by the client side, situated in the mobile node. Connections are point-to-point full-duplex.

The protocol data unit is the packet. Its size is fixed, not negotiated in a per-connection basis. There is no limit for that size, but in practice it should not be greater than the maximum size of an IP datagram.

STP does not make any congestion control, as there are no routers between client and proxy, and, therefore, no risk of congestion. It uses an optional credit based mechanism for flow control.

### 4.2 Packet format

Figure 4 describes the header of STP packets. Two classes of STP packets are defined: packets to open connections (fig 4.a), and packets to send information, acknowledgements, or credits, and to release connections (fig. 4.b). Optional fields (shadowed) are used or not depending of the type of connection negotiated, with or without error control and with or without flow control.

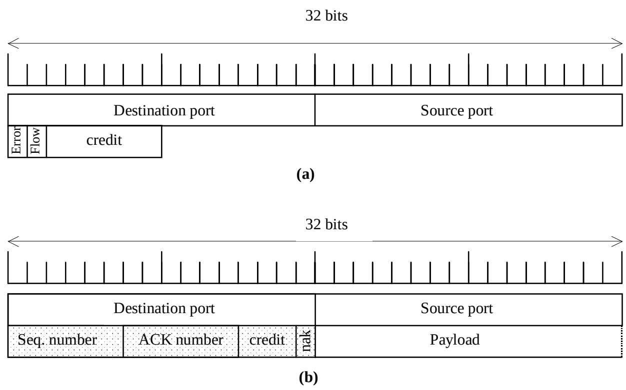



***Fig. 4.*** *STP packet formats*

STP uses selective retransmission: it could manage a sliding window of size 32, but sequence number 0 (zero) is reserved to identify control packets (ACK or credits). Thus, the maximum allowed window size is 31. The value of the credit field is interpreted as an increase in the current allowed credit (in packets), not as the absolute available credit. This means that it is not possible for the sender to reduce already granted credit (the use of negative numbers is not considered by now). Optionally, when used in lines with a higher *delayxbandwith* product, the length of these three fields can be expanded to 12 bits for sequence numbers and ACK, and 7 for credit. At the end of the head there is a bit for NAK, used in error recovery. We will see its use in the subsection devoted to error control.

## 4.3 Connection management

STP connections are established in two steps. First, the client sends a connection request, specifying if error and/or flow control are required. When client asks for flow control, the *Credit* field in the header of the reply packet will be the initial credit accepted by the client. Second, the proxy replies accepting or rejecting the request. If it accepts, *Error* and *Flow* fields are irrelevant in the reply, and *Credit* will be a non-zero value. If the client requested flow control, this field contains the initial credit accepted by the proxy; otherwise its value is irrelevant (but still non-zero).

When the connection request is rejected, *Credit* field values 0, and *Error* and *Flow* bits encode the reason to reject. A timer controls the time a client spends waiting for a reply to a connection request.

The procedure to release a connection is very similar to the one used in TCP. A duplex connection is seen as a pair of simplex connections. Each simplex connection is released independently of its sibling, sending an empty packet. It is a close report (not a request), enough to effectively release the connection when no error control has been negotiated. If error control is enabled, the sender of the close report must wait for an acknowledgment from the other end.

The steps required to establish and release connections can be represented in a finite state machine with the 9 states listed in table 1.

The finite machine itself is shown if figure 5. Three kinds of events appear in the figure: packet reception (labeled with '+'), packet send (labeled with '-'), and timeout. When packets are sent or received, the significant fields of the packet appear, separated by commas. Events and transitions written in *italic* only occur when error control is enabled. The <u>underlined</u> ones are legal only when there is no error control. The only difference between client and proxy machines is in the connection opening process, symmetric to each other.

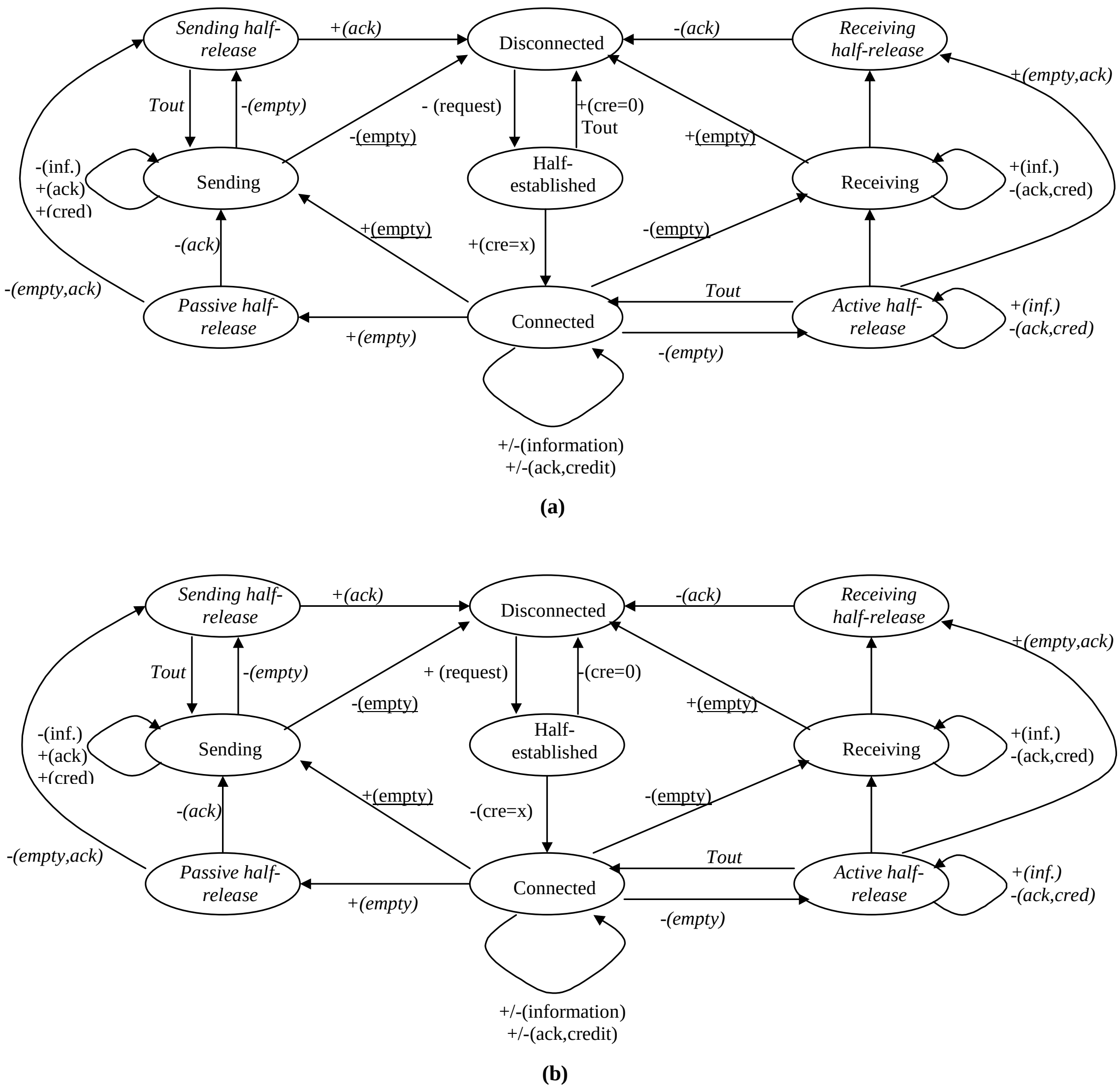



***Fig. 5**. STP connection management finite state machine. (a) Client (b) Proxy*

| State | Description |
|---|---|
| Disconnected | There is no established connection |
| Half-established | A connection request is pending for reply |
| Connected | Connection is opened and information can flow in both directions |
| Passive half-release | The other end has reported the release of the connection, and is waiting for the reception of the corresponding ACK |
| Sending | Connection is established only to send information, not to receive |
| Sending half-release | The current only sender of the connection has sent a release report and is waiting for the corresponding ACK |
| Active half-release | In a full-duplex connection, a release report has been sent and we are waiting for the corresponding ACK |
| Receiving | Connection is established only to receive information, not to send |
| Receiving half-release | The current only receiver of the connection has received a release report from the other end |

***Table 1.*** *The states used in the TCP connection management finite state machine*

## 4.4 Error control

Errors are not detected by error detecting codes added as a trailer to the packet. Instead, gaps in the sequence of received packets are interpreted as an error, taking into account that between client and proxy there is no routers, but just a circuit switched connection. So, any disorder in the sequence has to be the result of a missing packet or of an erroneous packet removed by RLP or PPP.

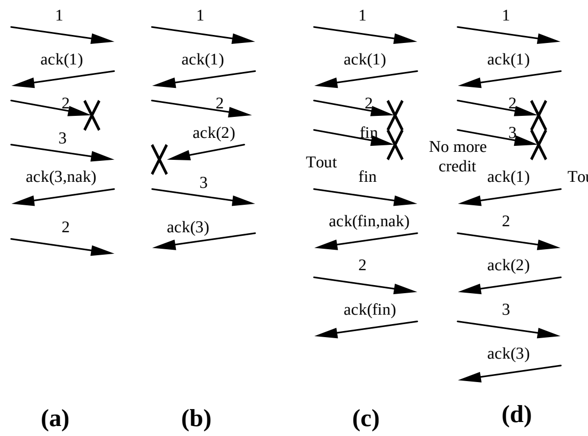



***Fig. 6.*** *Error recovering in STP. (a) Packet loss (b) ACK loss (c) packet burst loss, including close packet (d) deadlock by credit run out*

Selective retransmission of lost packets is used for error recovering, as shown in figure 6. Acknowledgments cannot be grouped: every sent packet has to be independently acknowledged. The *nak* bit is activated only when the receiver detects a gap in the receiving sequence (fig. 6a). An exception can occur when an ACK is lost. In that case, retransmission is not necessary, and should not be made. This way, if the lost packet is an ACK, the sender will receive the ACK corresponding to the next packet with the *nak* bit off, and will interpret it as an ACK for the last two packets. An example of this procedure is showed in figure 6b.

## 4.5 Timers in STP

Note that detecting errors as sequence gaps avoids the use of retransmission timers. It is a target in the design of STP to minimize the use of timers, as it management is the main cause of problems when using TCP in GSM links. Only three are used: one to establish a connection, another one to close it, and an activity timer used when error and flow control are enabled.

The first two are necessary to ensure a correct connection establishment and release, and to avoid situations like the one described in figure 6c. If a timer for the release packet is not provided, and the last two packets are lost (including the release packet) a deadlock occurs.

The third timer is used to prevent situations like the one described in figure 6d. In it, an error burst causes the loss of all the packets sent until the sender spends all the available credit. At this point, a deadlock will occur, with the sender waiting for some ACK with new credit, and the receiver waiting for new data to acknowledge. The solution is the defined activity timer, restarted in the receiver with every new packet arrival. When it times out, the last sent ACK is retransmitted, in the same manner TCP uses *keepalive* packets [15].

# 5. Performance evaluation

The performance of our approach was measured in extensive field trials. The objective was to study how our system behaves under different conditions, and to compare its performance with regular TCP/IP and indirect TCP [16]. The difference between indirect TCP and our approach is in the client-proxy link: we use STP, while in ITCP the standard TCP protocol is used.

## 5.1 Experimental environment

We carried out our study in an everyday working environment of the metropolitan area of Donostia-San Sebastian (Basque Country), accessing a publicly available GSM network from the School of Computer Engineering of The University of the Basque Country.

### *Hardware*

The hardware configuration is shown in figure 7. We will not describe the whole hardware configuration, but only the equipment and options that are relevant to the study. The end-user (client) equipment consisted of a portable PC, a GSM phone, and a GSM PC Card[1] (Mitsubishi MT-20D). The processor is a 133 MHz Pentium MMX, with 32 MB of main memory. The GSM phone is a handheld model (Mitsubishi MT-230), and the PC Card supports the asynchronous, non-transparent 2400-9600 b/s bearer service. The portable PC and phone of the client have been always been plugged to their power supplies (we have not used the batteries). The GSM network is the one built and

[1] GSM phone and PC Card provided by Telefónica Servicios Móviles S.A. to support this work.

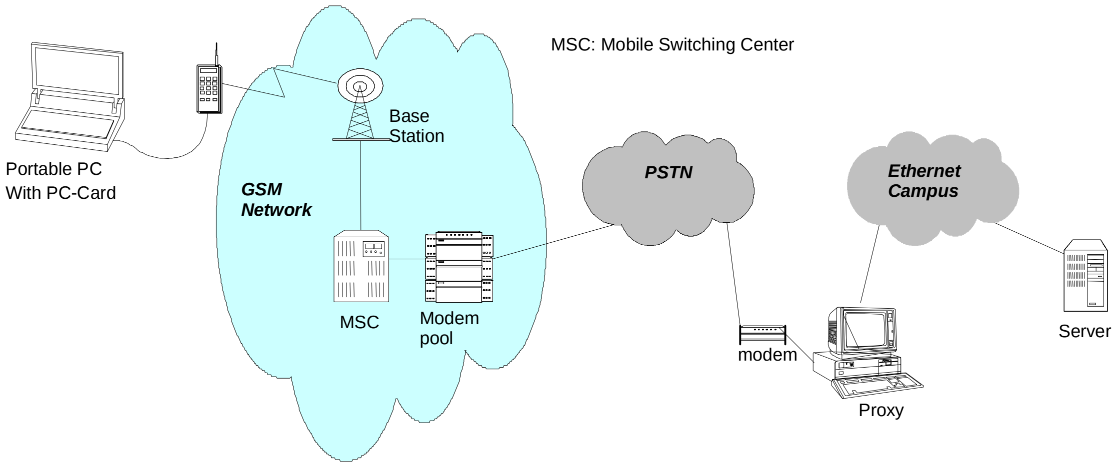



***Fig. 7. Experimental environment***

operated by Telefónica Servicios Móviles S.A.[2] It has a base station at about 400 m from the location of the client. The strength of the received signal in the phone of the client has been 4-6, using a 1-6 scale. All the measurements have been taken form the same point, avoiding any change in the conditions of the system.

The proxy equipment consisted of a PC with a 166 MHz Pentium MMX processor and 32 MB of RAM. It was connected to the PSTN by a campus PBX, using a Telebit Wordblazer modem, able to transmit up to 14.400 bps. It also has an Ethernet connection to the university's campus network. The application server to be accessed is in the same network, avoiding this way the effects of the Internet traffic in the experiments.

### *Software*

The operating environment of the portable PC and the proxy was RedHat Linux 5.2 (Kernel 2.0.36). In the proxy we have installed a RAS (Remote Access Server) module, and the data link protocol in the telephone connection was PPP. We have based all our measurements in ftp executions. Three different ftp clients have been run in the client machine: a standard one, another one compiled with SOCKS 5 standard client library to create an indirect TCP setup, and a third one compiled with a STP client library.

## 5.2 Measurements

We have measured the time needed to download a file from server to client. Three different parameters can be changed to make a range of measurements:

- Transport connection: it can be direct TCP between client an server, indirect through a standard SOCKS proxy (using TCP between client and proxy), or indirect through our STP proxy.
- Number of simultaneous connections: it can be one, three or four connections at the same time.
- Size of file to be transmitted: different sizes from 3,909 to 138,910 bytes.

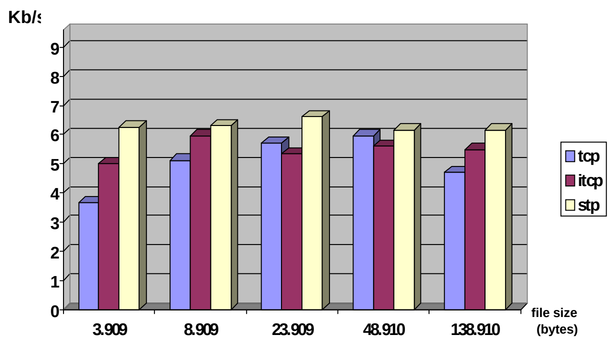
We have measured the time used by transport layer to send the files, capturing packet traces of the transfers in the client machine with the `tcpdump` program. The results of performance are summarized in graphs 1, 2, and 3. As the mean is the most often reported index in performance studies, the mean throughput is the value contained in the graphs.

### *Summary of measured results*

We will summarize all the experiments in two sets: (1) for a single connection between client and server, and (2) for many (three or four) simultaneous connections. In every set we have examined the performance of the three possible transport layer options, and transmitted files of different sizes.

1. One connection

Graph 1 shows that our approach with STP is superior to TCP and to indirect TCP. Typically the throughput with STP is 20-30% higher than with TCP.



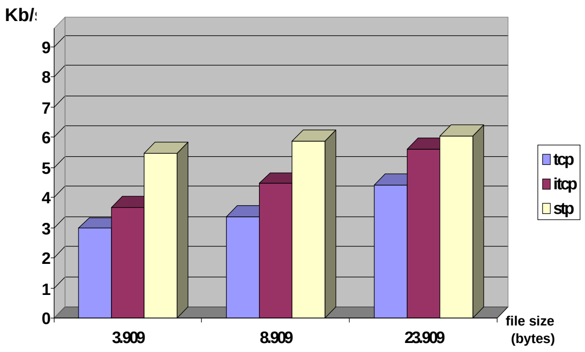

***Graph 1**. Throughput for only one connection*

There are two main reasons to justify the differences. First, the STP protocol uses the full bandwidth from the beginning of the connection, and TCP uses the slow-start congestion control mechanism at the beginning of each new connection. Second, TCP does a number of unnecessary retransmissions before trying to adjust to the latency of the GSM link. It leads TCP to enter the slow-start phase every now and then during the transfer, because of the variability of the latency in the radio link.

[2] Telefónica Servicios Móviles S.A. has provided free use of its network for this study.

2. Simultaneous connections.

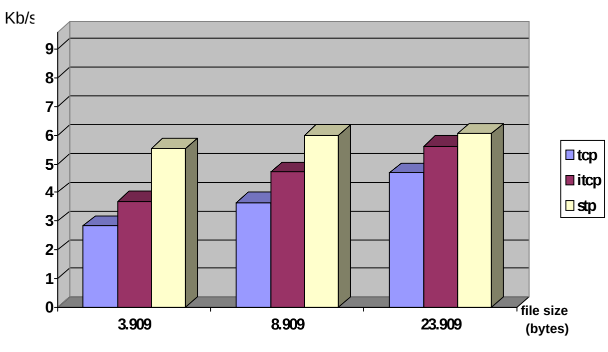
The measured throughput is shown in graph 2 (for three connections) and 3 (for four connections). With more than one simultaneous connection, we have reduced the size of the larger file sent to 23,909 bytes.




***Graph 2.*** *Throughput for two simultaneous connections*

These graphs show again the improvement of the throughput using STP, specially with respect to direct TCP. But there are two points worth mentioning. First, that the improvement is even greater than that one registered for a single connection. Second, the time needed to transmit 3 or 4 simultaneous files with STP is just slightly higher than the time needed to transmit them sequentially (the amount of transferred data being the same in both cases); however, with TCP this ratio do not hold: simultaneous transmission takes much longer.

We have found that both phenomena have the same cause: the very slow GSM link is the bottleneck of the transmission path, and traffic from different connections fills rapidly the buffers of the PPP interface. Thus, the simultaneous TCP connections start to mutually interfere in the access to the PPP link, causing additional unpredictable delays. With STP, the same situation occurs, but varying delays is not a problem for this protocol.




***Graph 3****. Throughput for three simultaneous connections*

## 6. Conclusions and future work

We have implemented a communication architecture that solves some performance problems that appear when accessing the Internet through a wireless GSM link. It retains the TCP/IP architecture unmodified at existing hosts of the fixed network, so that neither the existing network applications nor the protocol software on fixed hosts need to be modified. This is possible because we implement the indirect model of client/server interaction, which breaks the client/server communication in two separate parts: client-intermediate and intermediate-server. That intermediate element has been set in the frontier between GSM and the Internet.

Special attention has been paid to the performance of standard TCP/IP applications, which dramatically decreases when the access is made through a wireless network. To improve it we have replaced the standard TCP protocol by STP in the mobile-intermediate part of the connection, minimizing the traffic over that link and avoiding the performance problems of a TCP connection traversing a GSM link. We have found that interference between traffic from simultaneous, different TCP connections is as pernicious to TCP performance as link layer retransmissions made by RLP.

This work has been done for the current GSM data connection. New data services will soon be offered by GSM operators. As the basic characteristics of the radio link will not change (in terms of error rate and reduced bandwidth when compared to fixed networks), we think that most of our proposals will be still valid for the new environments. New performance studies will be done when these services become available.

Once demonstrated the interest of using an indirect approach to access Internet from GSM, and substituting TCP by a simpler protocol in the client-proxy link, its time to improve the implementation, going beyond the current prototype. These new and more complete implementations should be located directly on top of the network (IP) layer, and not, as it is now in the prototype, at the application layer.

## 7. Acknowledgements


This work has been supported by the Gipuzkoako Foru Aldundia, Basque Country (OF 325/99), by the Comisión Interministerial de Ciencia y Tecnología, Spain (TIC98-1162-C02-02), and by Telefonica Servicios Moviles S.A. Dr. Miguel-Alonso's stay at Purdue University is supported by the Secretaría de Estado de Universidades, Investigación y Desarrollo, Spain.


## References


[1] M. Rahnema. *Overview of the GSM System and Protocol Architecture*. IEEE Communications Magazine, 31(4) pp. 92-100, April 1993.

[2] T. Alanko, M. Kojo, H. Laamanen, M. Liljeberg, M. Moilanen, K. Raatikainen. *Measured Performance of data transmission over cellular telephone networks.* Computer Communications Review, 24(1994)5.

[3] José María Rivadeneyra. *Una arquitectura indirecta para el acceso a redes de datos desde GSM*. Ph.D. dissertation. Department of Computer Architecture and Technology, The University of the Basque Country, October 1999.

[4] ETSI. *Quality of service*, GSM specification 02.08, Version 3.0.0., March 1990.

[5] R.Cáceres, L. Itfode. *Improving the performance or reliable transport protocols in mobile computing*

*environments.* IEEE JSAC Special Issue on Mobile Computing Networks, 1994.

[6] H. Balakrishnan, V.N. Padmanabhan, R.H. Katz. *The effects of asymmetry on TCP performance*. MOBICOM97, Budapest, Hungary. September 1997.

[7] P. Bhagwat, P. Bhattacharya, A. Krishna, S. K. Tripathi. *Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs*. ACM Wireless Networks, March 1995.

[8] M. Kojo, K. Raatikainen, M. Liljeberg, J. Kiiskinen, T. Alanko. *An efficient transport service for slow wireless telephone links*. IEEE Journal on Selected Areas in Communications, Vol 15, nº 7, September 1997.

[9] V. Jacobson, T. Braden. *TCP extensions for long delay paths*. RFC 1072, Octubre 1988.

[10] ETSI. *High Speed Circuit Switched Data (HSCSD) - Stage 1*, GSM specification 02.34, April 1997.

[11] ETSI. *GPRS; Service description*, GSM specification 02.60, Version 5.1.0. January 1998.

[12] C. Erlandson, S. Jhawar, M. Terho, C. Parrish. *WAP Forum*. GSM World Congress, Cannes, France. February 1998. Available in http://www.wapforum.org/what/WapCanne.pdf.

[13] *Wireless Application Protocol. Architecture Specification*. Version 30/4/98. Available in http://www.wapforum.org/what/technical/arch-30-apr-98.pdf.

[14] J.M. Rivadeneyra, J. Miguel-Alonso. *A communication architecture to access data services through GSM*. In 7th IFIP/ICCC Conference on Information Networks and Data Communications, Aveiro (Portugal), June 1998.

[15] W.R. Stevens, *TCP/IP Illustrated*, *Volume I: The protocols*, Addison-Wesley, 1994.

[16] A. Bakre, B.R. Badrinath, *I-TCP: Indirect TCP for mobile hosts*. Rutgers University, DCS-TR-314, 1994.