Mobile access to the Internet: a mediator-based solution

Timo Alanko
Markku Kojo
Mika Liljeberg and
Kimmo Raatikainen

The authors

Timo Alanko is Professor**, Markku Kojo** is Assistant Professor**, Mika Liljeberg** is Researcher and **Kimmo Raatikainen** is Professor, all in the Department of Computer Science, at the University of Helsinki, Helsinki, Finland.
E-mail: {timo.alanko, Markku.kojo, mika.liljeberg, kimmo.raatikainen}@cs.Helsinki.fi

Keywords

Cellular communications, Computer architectures, Distributed data processing, Internet, Mobile communications

Abstract

Nomadicity is a new challenge for computing and communication technologies. Modern cellular telephone systems extend the usability of portable personal computers enormously. A nomadic user can be given ubiquitous access to remote information stores and computing services. However, the behaviour of wireless links creates severe inconveniences within the traditional data communication paradigm. In this paper we give an overview of the problems related to wireless mobility. We also present a new software architecture for mastering the problems and discuss a new paradigm for designing mobile distributed applications. The key idea in the architecture is to place a mediator, a distributed intelligent agent, between the mobile node and the wireline network.

Internet Research: Electronic Networking Applications and Policy
Volume 9 · Number 1 · 1999 · pp. 58–65
© MCB University Press · ISSN 1066-2243

Introduction

Nomadicity (Kleinrock, 1997) has created a new epoch in computing through new challenges to client-server computing (Forman and Zahorjan, 1994). Traditional TCP/IP has been reported to be insufficient and vulnerable when the communication path involves a (slow) wireless link (Alanko *et al.*, 1994; Cáceres and Iftode, 1995; Kojo et al., 1997). Mediator-based solutions have been proposed to improve performance and to increase reliability (Bakre and Badrinath, 1995; Balakrishnan *et al.*, 1995; Kojo *et al.*, 1994; Yavatkar and Bhagawat, 1994). It is also recognised that RPC needs support in order to be useful in wireless networking environments (Bakre and Badrinath, 1997; Davies *et al.* 1997). Furthermore, mediators are found to be useful also in CORBA environments (Liljeberg *et al.*, 1997, 1998; Raatikainen, 1997).

In this paper we examine one mediator-based solution, the Mowgli approach developed at the Department of Computer Science in the University of Helsinki. We start by briefly discussing the problems of TCP/IP in mobile environments. In Section 3 we outline the Mowgli approach.

Problems in TCP/IP over wireless links

Under favourable radio conditions a wireless telephone link behaves almost like a normal PSTN link. Therefore, it can be expected that existing Internet applications still work when conditions are good. However, differences between wired and wireless links cause complications. The common data communication protocols, like the TCP/IP Internet protocols, do not expect problems like those occurring in a cellular telephone network.

The TCP/IP protocols are designed for wired networks and recent research shows performance problems due to inappropriate operation of current TCP/IP when wireless links are involved (Alanko *et al.*, 1994; Cáceres and Iftode, 1995; Kojo *et al.*, 1997). The experimental results indicate that even minor disturbances in connections may lead to sub-optimal performance – sometimes to delays which irritate users – and failures at the application level.

To improve the user-visible performance several enhancements are needed:
(1) a link-level protocol tuned for a slow wireless link;
(2) facilities for parallel data transfer operations some of which are to be executed in the background; and
(3) an elaborate multiplexing system giving higher priorities for the more urgent tasks (especially for interactive tasks with a waiting user).

The transmission control mechanisms of TCP are based on one basic performance metric: the round-trip time. An abruptly increased round-trip time may cause the TCP retransmission timer to activate, which in turn is interpreted as a sign of insufficient capacity somewhere in the network, and the recovery mechanism is to decrease the sending rate. For wireless connections, like the GSM non-transparent data service, highly variable transmission delays are typical, but they have totally different origins: link-level re-transmissions generated by bursts of errors on the radio link. Hence, the reaction of TCP is incorrect. What is needed is a separation of control: the wireless side and the wired side should be controlled independently of each other.

The wireless link is vulnerable. The mobile workstation may move through an uncovered area, or the radio conditions may temporarily deteriorate. In both cases the link becomes inaccessible for some time. Neither the TCP/IP protocols nor applications are prepared for this – in the wired network a broken link is considered to be a fatal failure, and typically all active operations are terminated. A mediator can obviously be used to mask intermittent breaks.

Present cellular telephone systems are circuit switched, and charging is based on the connection time. Thus, the customary habit of working with the link open over the whole session is not reasonable. Clearly, a more elaborate control of the wireless link is needed.

TCP and GSM data

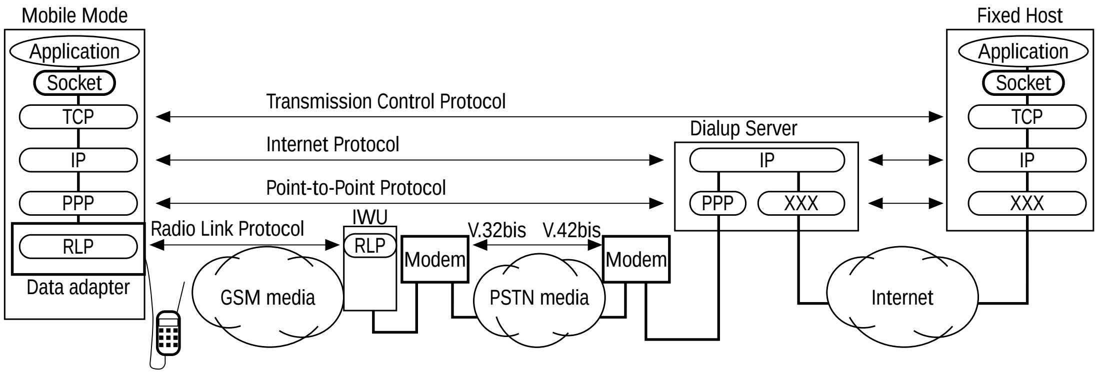
Figure 1 shows how the commercial non-transparent GSM data service can be used to connect a mobile workstation to the fixed Internet using standard Internet protocols. The point-to-point protocol (PPP) (Simpson, 1994), or alternatively the Serial Line Internet Protocol (SLIP) (Romkey, 1988), is used as the framing protocol between the mobile workstation and a dial-up server within the fixed network. The dial-up server functions as a router between the point-to-point link to the mobile workstation and the rest of the Internet. IP datagrams are encapsulated within PPP (or SLIP) frames so that TCP/IP protocols can be used in an end-to-end manner between the mobile workstation and any other Internet host. This is a very typical arrangement that has worked well for connecting modem users to the Internet over PSTN. However, the uncertainty of the wireless medium brings problems not found in the PSTN environment.

Many of the performance problems are related to the slow-start congestion control algorithm (Jacobson, 1988) used by TCP. The slow-start algorithm is employed on two occasions:
(1) at the start of each new TCP connection; and
(2) whenever the network is congested.

The slow-start algorithm is applied at the start of each TCP connection just for the case that the network might be congested at that particular moment. The theory is that this inefficiency at the beginning of the connection is negligible because it only occurs at the start of the connection. This is true for connections that transfer a significant amount of data. However, in very short-lived TCP connections, such as those created by WWW type activity, the inefficiency is significant. In the GSM data environment with its high latency, the slow-start, together with the three-way handshake employed at the start of each TCP connection creates a delay visible to the end-user.

The TCP protocol has been tuned for the relatively reliable fixed networks, in which a packet loss due to transmission errors is rare. Therefore, the protocol assumes that each packet loss is a sign of network congestion and the slow start is triggered whenever a packet is lost. Unfortunately, in a wireless environment, packet losses due to transmission errors are quite common. A TCP connection, spanning both wireless and fixed networks, will have poor performance because TCP misinterprets every lost packet as network congestion and slows

Figure 1 TCP/IP and PPP over GSM data service



down the rate of transmission. The result is that even though new data could immediately be delivered over the wireless link, the sending TCP layer, instead of transmitting packets, waits for a non-existent congestion to disappear.

In the non-transparent GSM data service the Radio Link Protocol (RLP) is used to provide reliable transfer over the wireless link. In order to correct transmission errors the RLP protocol retransmits corrupted frames over the wireless link. The result is that transmission errors corrected by RLP will be perceived as transmission delays. Usually the delays are long enough for TCP to interpret them as lost packets, since acknowledgements do not arrive in time. Again, TCP assumes that there are congestion problems in the network and slows down the rate of transmission.

Delays due to RLP retransmissions also cause other performance problems. A notable one is that, in addition to triggering congestion control, such a delay may also cause the sending TCP layer to unnecessarily retransmit packets. Naturally, the retransmissions will also go over the wireless link, consuming bandwidth that could be better used for other traffic. Further retransmissions are caused by the fact that the components in the GSM data service can buffer tens of kilobytes of data along the communication path, particularly in modems and in the dial-up server that functions as a router between the fixed and wireless networks. The resulting queuing effect increases the round-trip time causing further retransmissions followed by slow starts. Unnecessary retransmission accumulation is a problem that is more probable with TCP connections where a significant amount of data is transferred, e.g. in file transfers. Problems due to delayed acknowledgements have also been reported in an earlier study (Zhang *et al.*, 1991). The phenomenon was called “ack-compression”.

A particularly difficult case of the unnecessary retransmission syndrome manifests when the TCP send window is fully open (transmission has gone well for a while), most of the segments in the send window have been buffered along the communication path, and a long delay on the wireless link causes all those segments to time out. According to the slow-start algorithm the congestion window is set to one segment and the first timed-out segment is retransmitted. Soon, the sending TCP receives an acknowledgement for the first segment, however, not in response to the retransmission but to the original transmission. Since the sending TCP is receiving acknowledgements, it assumes that everything is all right, increases the congestion window and retransmits some more segments in accordance with the slow-start algorithm. Thus, retransmitted TCP segments keep piling into the buffers along the communication path. The syndrome may repeat for a while but eventually it will die out, because at each repetition the congestion window is allowed to increase at a much slower rate.

It should be noted, that TCP implementations differ in the way they handle retransmissions. Some of them adhere to the TCP specification more strictly than others. In addition, the specification is not entirely clear in how to deal with multiple packets “timeouting” simultaneously.

In TCP the retransmission time-out value (RTO) is constantly adjusted according to measured round-trip time estimates (Jacobson, 1988). In our experience, the algorithm does not adapt well to the highly variable delays of a wireless environment, causing either numerous unnecessary retransmissions or too slow retransmissions when packets are actually being lost. In addition, some TCP implementations, which are optimised for fixed networks, incorrectly use a quite short initial retransmission time-out at the start of a TCP connection.

The low initial value of RTO causes unnecessary retransmissions before the algorithm adjusts to the high-latency link. This is not a serious problem in long-lived TCP connections, in which a lot of data are transferred, but in short-lived TCP connections a significant fraction of the transmitted data segments can in fact be retransmissions. Part of the problem is that adjusting the RTO is slow, because the algorithm attempts to smooth variations in measured round-trips. The main reason is that round-trip times cannot be reliably measured from retransmitted segments (Karn and Partridge, 1991). Thus, the unnecessary retransmissions in the beginning of the connection slow down the RTO adjustment causing more retransmissions.

A further issue with multiple TCP connections over a low-bandwidth link is that they can interfere with each other. Even a single bulk transfer over the wireless link causes the round-trip times observed by other connections to reach several (5-10) seconds (Kojo *et al.*, 1997). Again the reason is that data are buffered along the data communication path below the TCP layer. Especially new connections have difficulties:

- the retransmission timer takes a very long time to adjust upwards to the relatively high observed round-trip time;
- the too short initial RTO causes numerous unnecessary retransmissions and each time the sender applies the slow-start algorithm, making the adjustment very slow indeed.

In our experience, it may be almost impossible to start an interactive connection when there are ongoing bulk data transfers.

A small additional problem with TCP/IP performance over a low-bandwidth link is that the combined IP headers and TCP headers are quite verbose (typically 40 bytes). While van Jacobson’s header compression (Jacobson, 1990) is typically used with PPP framing, the overhead still shaves off a few percentage points of the available bandwidth.

To summarise, there are a number of subtle performance problems in using the TCP protocol over the non-transparent GSM data service. Because of the relatively high latency of the wireless link, short-lived TCP connections are slowed down by the initial three-way handshake, TCP slow-start, and the unnecessary retransmissions caused by a too tight initial retransmission time-out. Long-lived TCP connections, however, suffer from an excessive queuing effect on the wireless path, coupled with subtle interactions between congestion control and the round-trip time estimation algorithm used to calculate the retransmission time-out. Furthermore, parallel TCP connections competing for the same constrained link interfere with each other in a way that causes additional retransmissions.

Mowgli communication architecture

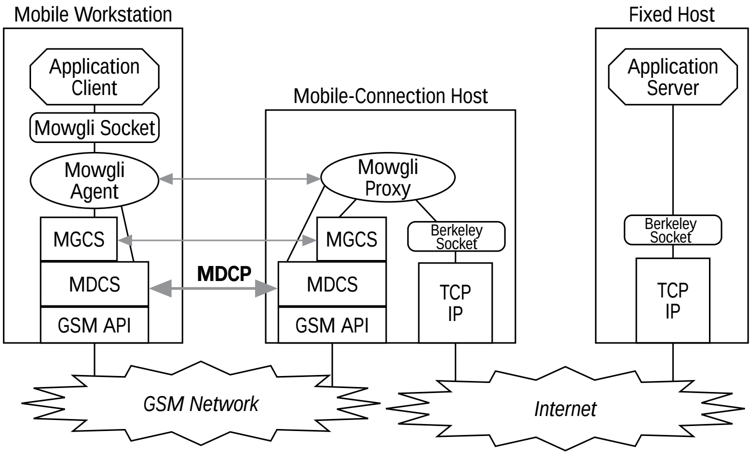
In the Mowgli architecture a mobile node is connected to the fixed network through a wireless telephone link. The key idea is to separate the control of the behaviourally-different wired and wireless worlds. Hence, the node on the border of these two worlds has an important role. This node, called the mobile-connection host (MCH), provides the mobile node with a connection point to the wired Internet. In addition, it offers a platform for intelligent agents, called proxies, which are capable of operating on behalf of the mobile node applications.

The mediator approach allows us to replace the regular TCP/IP protocols with a specialised set of communication services for the slow wireless link. These services are provided at three different layers:

(1) the agent-proxy layer;
(2) the data transfer layer, and
(3) the data transport layer.

Figure 2 depicts the logical organisation of the service architecture.

Figure 2 The Mowgli communication architecture for connecting mobile nodes to fixed networks



Mowgli data channel service

At the data transport layer, the TCP/IP is replaced with the Mowgli Data Channel Protocol (MDCP) (Kojo *et al.*, 1997), which is designed to cope with the special characteristics of slow wireless links. All communication above the data transport layer uses the Mowgli data channel service (MDCS) that implements the MDCP protocol. The transport services of the MDCS through the MDCS socket interface, is similar to the widely used Berkeley socket interface. The communication in the MDCS is based on data channels that provide similar transport services as TCP and UDP. However, each data channel has a priority and a set of additional attributes for controlling the behaviour of the channel in case of exceptional events.

The priority-based multiplexing of data channels significantly improves usability of the slow wireless link. For example, a channel with high priority can be employed to transmit data for an interactive application, thus providing reasonable response times to the end user even though a low-priority channel is simultaneously used for a background file transfer. Another convenient feature of the MDCS is improved fault-tolerance. The MDCS is particularly designed so that recovery from unexpected temporary disconnection of the wireless link is efficient. In order to accomplish this, MDCS maintains the state information of the ongoing transmission over each channel so that an interrupted transmission can be resumed later.

The high performance of the MDCP protocol has been achieved by minimising the amount of protocol overhead, specifically protocol headers and round-trips over the wireless link. In addition, MDCP always tries to transmit data at the best possible speed the wireless link is capable of, and it uses acknowledgements sparingly. Thus, MDCP reaches the full bandwidth much faster than TCP and is capable of retaining the transfer rate rather stable also in case of temporary delays on the wireless link (Kojo *et al.*, 1997).

Mowgli generic communication services

The generic communication service level is a collection of services that can be exploited in various applications. The Mowgli approach is open-ended in the sense that new services can easily be introduced into the generic communication service level. Currently, the following services have been identified:

- Mowgli data transfer service;
- Mowgli security service adaption;
- Mowgli ORB support;
- Mowgli cache service; and
- Mowgli event notification service.

The Mowgli data transfer service (MDTS) is able to take over the responsibility of transferring structured user data over the wireless link. The basic element of information for transfer operations is an information eXchange unit (IXU). In general, an IXU is something that the user considers as an independent unit of information, the transfer of which he/she may want to control. Examples of IXUs include mail messages, files, print jobs, and WWW pages or inline images of WWW pages.

Each IXU can have a set of attributes, which are used in controlling the transfer of the IXU. The MDTS provides an API for creating IXUs to be sent, for receiving IXUs, for managing transfer queues, and for changing attributes of specified IXUs. Owing to the attribute system, the MDTS is able to operate rather independently on various kinds of background transfers; it can make decisions about invoking transfers when conditions are favourable, about postponing transfers when conditions deteriorate, about trying to recover from failures, and about cancelling operations.

Agents and proxies

At the agent-proxy layer reside the agent–proxy teams. A proxy on the MCH and an agent on the mobile node co-operate to act as an intermediary for all data delivered between an application on the mobile node and its peer in the Internet. The proxy has a special mission: it plays the role of the mobile node application while communicating on its behalf with the peer.

We have slightly modified the existing socket interface so that the services of the agent-proxy layer are available to all existing Internet applications using the socket interface to interact with TCP/IP protocols. This interface, called the Mowgli socket interface, is retrospectively compatible with the Berkeley socket interface. Hence, existing applications that use TCP or UDP sockets can be executed on the mobile node without modifications or re-compilation. But, in addition to this the Mowgli sockets provide new features not available with the conventional TCP/IP sockets. These features include assignment of priorities to the data delivered through a socket, control of the wireless link, and ordering an automatic recovery from unexpected link-level disconnection. The new features can be configured individually for each application using an external configuration table in which the configuration is based on the well-known ports that the Internet applications use.

The Mowgli socket interface binds the existing applications to use the services of an appropriate agent–proxy team. The master agent and proxy, together with a generic agent-proxy team, provide the semantics of the TCP and UDP sockets for any application. In order to accomplish this the Mowgli socket layer informs the master agent whenever an application on the mobile node invokes a socket operation. If appropriate, the master agent then sends a message to the master proxy that performs the corresponding operation on behalf of the application. When an application invokes a socket operation to create an end-to-end TCP connection with a peer in the Internet, the masters first open an MDCS data channel with desired priority. Once the TCP connection with the remote peer has been established, the masters create a generic agent-proxy team which will take care of delivering the data for that connection.

The generic agent and proxy can be replaced with a customised agent and proxy which are tailored for a specific application protocol. If such a customised team is configured for an application, the master agent connects the application to the desired customised agent. The agent and proxy are able to decide how communication in that application can be optimised for the constrained wireless link. The Mowgli WWW software (Liljeberg *et al.*, 1995, 1996) is a good example of an agent-proxy team taking advantage of its knowledge about application semantics.

When new applications are implemented for mobile users, the functionality of a customised agent can be integrated into the mobile application software. Such a mobile client application can co-operate directly with the customised proxy on the MCH. This approach amounts to splitting the traditional client program into two parts – one part on each side of the wireless link.

Summary

The conceptual design of the Mowgli system has its origins in the different natures of the wireless and wireline worlds, in some specific features of the wireless link, and in the problems of a nomadic user. The main idea is to develop new functionality using the concept of a mediator that consists of an agent-proxy team and of associated infrastructure. This approach leads to an essentially improved performance when compared to the regular TCP/IP-based solution (Kojo *et al.*, 1997; Liljeberg *et al.*, 1996).

The idea of splitting the end-to-end control turned out to be fruitful. First, it helps to utilise the capacity of the bottleneck device, the wireless link. The proxy on the MCH acts as a high-level buffer balancing variability in arrival rates from the wireless side, which are due to distortions on the radio link, and from the wireline side, which are due to congestion in the Internet. Second, the splitting prevents irregular behaviour on either side from harming the control of the other side: wireless delays do not cause unnecessary end-to-end retransmissions and wireline congestion does not affect operations over the wireless link.

Of more importance for the end users are the qualitative improvements: occasional breaks in the radio link have no visible effects at the user interface level, automatic control of the telephone connection is a blessing for unaccustomed nomadic users, and the possibility of transparent background operations relieves the nervous end user from a lot of idle waiting.

The role of the agent-proxy team is pronounced: together the agent and proxy form a model for a distributed intelligent agent that is “Janus-faced” in two ways. Horizontally it presents the client-interface on the fixed-net side and the server-interface on the mobile node, or vice versa. Vertically, it knows both the application semantics and the problems in wireless communication. Hence, it can take several different roles including an advisor, a filter, a booster, and a representative.

Today the research in mobility is primarily concentrated on the problems of physical mobility: the systems perceive how the nodes move. In the Mowgli environment the cellular telephone system hides certain issues in the terminal mobility like paging, handovers, and location updates. Therefore, we are able to focus on the next set of problems:

- when to connect;
- how to control disconnected agents; and – in general
- how to improve the dependability of applications comprising wireless data communications.

References

Alanko, T., Kojo, M., Laamanen, H., Liljeberg, M., Moilanen, M. and Raatikainen, K. (1994), “Measured performance of data transmission over cellular telephone networks”, *Computer Communications Review*, Vol. 24 No. 5, pp. 24-44.

Bakre, A. and Badrinath, B.R. (1995), “I-TCP: indirect TCP for mobile hosts”, *Proceedings of the IEEE 15th International Conference on Distributed Computer Systems*, pp. 136-43.

Bakre, A. and Badrinath, B.R. (1997), ”Reworking the RPC paradigm for mobile clients”, *Mobile Networks and Applications*, Vol. 1 No. 4 , pp. 371-85.

Balakrishnan, H., Seshan, S., Amir, E. and Katz, R. (1995), “Improving TCP/IP performance over wireless networks”, *Proceedings of the 1st ACM International Conference on Mobile Computing and Networking*, pp. 2-11.

Cáceres, R. and Iftode, L. (1995), “Improving the performance of reliable transport protocols in mobile computing environments”, *IEEE Journal on Selected Areas in Communications*, Vol. 13 No. 5, pp. 850-7.

Davies, N., Friday, A., Blair, G.S. and Cheverst, K. (1997), “Distributed systems support for adaptive mobile applications”, *Mobile Networks and Applications*, Vol. 1 No. 4, pp. 399-408.

Forman, G.H. and Zahorjan, J. (1994), “The challenges of mobile computing”, *IEEE Computer Magazine*, Vol. 27 No. 4, pp. 38-47.

Jacobson, V. (1988), “Congestion avoidance and control”, *Proceedings of the ACM SIGCOMM ‘88 Symposium on Communications Architectures and Protocols*, pp. 314-29.

Jacobson, V. (1990), *Compressing TCP/IP Headers for Low-Speed Serial Links*, Request for Comments 1144, Network Information Centre.

Karn, P. and Partridge, C. (1991), “Improving round-trip time estimates in reliable transport protocols”, *ACM Transactions on Computer Systems*, Vol. 6 No. 4, pp. 364-73.

Kleinrock, L. (1997), “Nomadicity: anytime, anywhere in a disconnected world”, *Mobile Networks and Applications*, Vol. 1 No. 4, pp. 351-7.

Kojo, M., Raatikainen, K. and Alanko, T. (1994), “Connecting mobile workstations to the Internet over a digital cellular telephone network”, *Proceedings of MOBIDATA Workshop*. An updated version in *Mobile Computing*, Kluwer, 1996, pp. 253-70.

Kojo, M., Raatikainen, K., Liljeberg, M., Kiiskinen, J. and Alanko, T. (1997), “An efficient transport service for slow wireless telephone links”, *IEEE Journal on Selected Areas in Communications*, Vol. 15 No. 7, pp. 1337-48.

Liljeberg, M., Alanko, T, Kojo, M., Laamanen, H. and Raatikainen, K. (1995), “Optimizing World-Wide Web for weakly-connected mobile workstations: an indirect approach", *Proceedings of the IEEE 2nd International Workshop on Services in Distributed and Networked Environments*, pp. 132-9.

Liljeberg, M., Helin, H., Kojo, M. and Raatikainen, K. (1996) "Mowgli WWW software: improved usability of WWW in mobile WAN Environments", *Proceedings of the IEEE Global Internet 1996 Conference*, pp. 33-7.

Liljeberg, M., Maumon, N., Sevanto, J. and Raatikainen, K. (1998), "CORBA in wireless and mobile networks", *Proceedings of the ACTS Mobile Communications Summit '98*, Vol. 1, pp. 361-6.

Liljeberg, M., Raatikainen, K., Evans, M., Furnell, S., Maumon, N., Veltkamp, E., Wind, B. and Trigila, S. (1997), "Using CORBA to support terminal mobility", *Proceedings of the TINA'97 Conference (Global Convergence of Telecommunications and Distributed Object Computing*, IEEE Computer Society Press, pp. 59-67.

Raatikainen, K. (1997), *Bridging and Wireless Access for Terminal Mobility in CORBA*, DOLMEN public document LK-OMG01, http://fub.it/dolmen.

Romkey, J. (1998), *A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP, Request for Comments 1055*, Network Information Centre.

Simpson, W. (1994), *The Point-to-Point Protocol (PPP)*, request for comments 1,661, Network Information Centre.

Yavatkar, R. and Bhagawat, N. (1994), "Improving end-to-end performance of TCP over mobile Internetworks", *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications,* pp. 146-57.

Zhang, L., Shenker, S. and Clark, D.D. (1991), "Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic", *Proceedings of the ACM SIGCOMM '91 Symposium on Communications Architectures and Protocols*, pp. 133-47.