University of Helsinki
Department of Computer Science
Series of Publications C, No. C-1994-39

Connecting Mobile Workstations to the Internet over a Digital Cellular Telephone Network

Markku Kojo, Kimmo Raatikainen, and Timo Alanko

Helsinki, September 1994

Report C-1994-39

University of Helsinki
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, FINLAND

Connecting Mobile Workstations to the Internet over a Digital Cellular Telephone Network

Markku Kojo, Kimmo Raatikainen, and Timo Alanko
Department of Computer Science, University of Helsinki
Report C-1994-39 (Second corrected print; First printed June 1994)
September 1994
16 pages

Abstract. Modern portable computers and wireless connections using a cellular telephone network have created a new platform for distributed information processing. We present a communication architecture framework which makes it possible to exploit the existing TCP/IP communication architecture but which also takes into account the specific features of wireless links. Our communication architecture is based on the paradigm of indirect interaction. The mediating interceptor, *Mobile-Connection Host*, is the bridge between the worlds of wireless and wireline communication. The interceptor also provides enhanced functionality that improves fault-tolerance and performance for applications aware of mobility.

Key Words: mobility, wireless connections, mobile workstations, cellular telephone, GSM, TCP/IP, mobile IP

CR Classification: C.2.1, C.2.4

Contents

1. Introduction

Recent developments in mobile communication and personal computer technology have created a new platform for information processing. A modern portable computer offers remarkable processing power, always at hand for a nomadic user, and a wireless connection gives to the user access to information stores independent of his or her present location.

The computing environment we are interested in, consists of a fixed data communication network, of portable laptop computers, and of a wireless access to that network through a cellular telephone network. Examples of such cellular networks include the Global System for Mobile Communications (GSM) [Rah93] and the Nordic Mobile Telephone (NMT), a digital system and an analogue system, respectively. Even in this new kind of environment it is still expected that old applications are available. From the software engineering point of view it would be desirable if all software which is sensitive to mobility could be embedded in the data communication subsystem, in other words within the physical, data link and network layers. Such expectations are not completely unjustified. Under favourable conditions a wireless telephone link behaves like a link in a traditional Public Switched Telecommunications Network (PSTN). In addition, the cellular telephone network can – if the user so wishes – hide the user mobility. However, there are situations in which the differences between wireline and wireless links should be taken into account. Unmodified software may behave in a way that the user finds disrupting or even in a way that creates sheer catastrophes.

The realization of a mobile computing environment requires a communication architecture which is not only compatible with the current architectures but which also takes into account the specific features of mobility and wirelessness. Current communication architectures, such as the TCP/IP Internet protocols, are suitable for fast and reliable networks like the megabits per second LAN's. In contrast, the wireless links have a low throughput, only kilobits per second, and a highly variable quality of transmission.

The cellular GSM telephone system offers a communication medium which is rather reliable when used in the so called non-transparent mode: the bit error rate is required to be less than 10^{-8} [ETS90]. However, the price paid for the low bit error rate is the occurence of highly variable transmission delays. Under unfavourable conditions the delays can extend to tens of seconds, or even to minutes. It should also be remembered that the conditions may become so unfavourable that a mobile telephone transmission path will be temporarily broken.

One of the goals of the project Mowgli[1] is to examine the behaviour of data communication over cellular telephone links and to develop architectural solutions to overcome difficulties typical for this kind of mobile wireless environment. In this paper we introduce a communication architecture framework which builds a bridge between the wireline and wireless worlds. The main design goals in our approach are:

1. to retain the TCP/IP-based communication 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,
2. to provide an application programming interface which allows an easy development of new mobile client applications in a way that they are able to co-operate with existing services on fixed hosts, as well as the development of new kinds of client-server applications for mobile computing,
3. to minimize the traffic over the wireless link and to improve fault tolerance and performance, and
4. to be able to conform with the work in progress on the Mobile IP standard [Sim94].

These goals cannot be achieved at any single architectural level. Instead, we must consider several design issues at all architectural levels, from the data link layer up to the application layer. The most vital point of interest is the fixed-network node to which the mobile node is connected during a (GSM-) session. This mediating node has a "Janus-like" functionality. On one side it is vis-a-vis with a fragile, low-performing neighbour. On the other side it is part of a reliable, high-performance world-wide network, which is rigid and well-established in its functionality.

In a distributed application it would be reasonable to implement the communication between the mobile part and the fixed part using two distinct subsystems. For this purpose the mediating node is a natural place to provide enhanced functionality. For example, this node can act as a gateway which connects two different communication protocols, each suited for the underlying communication medium. It can also offer more sophisticated services available on higher layers of the communication system. A similar idea of indirect client-server interaction between a mobile client and a fixed server has recently been presented in [BBIM93].

As an application programming interface (API) we provide a socket interface, which is similar to and highly compatible with the Berkeley sockets [Com91, pp. 335–363] as well

[1] Mobile Office Workstation using GSM Link; participants: University of Helsinki, Digital Equipment Corp., Nokia Mobile Phones, Telecom Finland

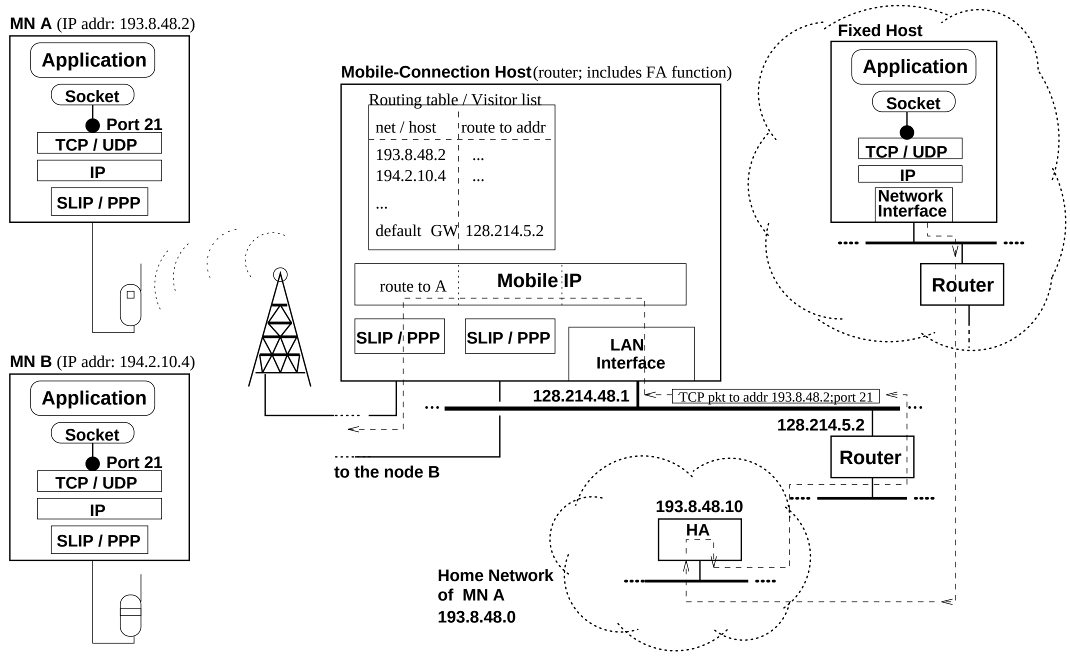

Figure 1: Routing IP datagrams to a Mobile Node

as the Windows Sockets [HTA+93]. Hence, it provides the functionality of reliable stream connections and of connectionless datagram delivery. In addition, a possibility to implement enhanced functionality needed by users who are aware of the impacts of mobility and wireless communication on their computation (mobile-aware users) is provided to application programmers.

In this paper we first discuss the work in progress on mobility support for TCP/IP networking and then we relate the TCP/IP protocols to some problems of wireless links. In Section 3 we outline a socket-level solution based on the use of a mediating node between the wireline world and the wireless world. Section 4 contains some ideas about enhanced functionality that our architecture makes feasible.

2. Mobile Nodes and TCP/IP Protocols

The general structure of a distributed system with Mobile Nodes using GSM-links is the basis for Figure 1. When a *Mobile Node* (MN) wants to be attached to a fixed TCP/IP network, it makes a GSM-call to a fixed node which is able to provide that connection. In our model this node, the mediating dial-in server, is called the *Mobile-Connection Host* (MCH). The task of the MCH is to deliver packets to and from the Mobile Node.

2.1 Mobile IP Routing

Technically, there are two straightforward ways to attach a Mobile Node to the TCP/IP Internet. The cellular telephone network is one way to solve the mobility problem: using a mobile telephone an end user can always make a direct call to an MCH at the home network. This MCH acts as a dial-up IP router. IP datagrams are transmitted over the wireless link by encapsulating them within the Serial Line Internet Protocol (SLIP) [Rom88] or the Point-to-Point Protocol (PPP) [Sim93, McG92].

However, if the user happens to be far away from the home network, it may be economically more attractive to use a variant of this solution. The user chooses the closest available MCH in the fixed data communication network and dials in to this MCH; through this MCH the Mobile Node then has a connection to any desired network. This variant has to make use of the Mobile IP techniques to manage the routing of the IP datagrams to the Mobile Node.

The proposed Mobile IP standard [Sim94] specifies protocol enhancements which allow transparent routing of IP datagrams to Mobile Nodes in the Internet. The key components in the proposed standard are the *Home Agent* (HA) located at the home network of the Mobile Node and the Foreign Agent located at the network to which the Mobile Node is connected. When a Mobile Node attaches to a foreign network, it first finds a locally reachable Foreign Agent. The Mobile Node then registers its current location with the Home Agent. The registration occurs through the Foreign Agent which forwards the registration to the Home Agent. After a successful registration, the Home Agent intercepts all packets addressed to the Mobile Node, encapsulates them, and tunnels them to the Foreign Agent. The Foreign Agent decapsulates the packets and delivers them to the Mobile Node. Outbounding IP datagrams from the Mobile Node are delivered to their destination by using the standard IP routing. Thus, applications on the Mobile Node can communicate with the rest of the Internet as if the Mobile Node were connected to its home network. In our architecture, the MCH to which the Mobile Node dials in provides the Foreign Agent function (see Figure 1). After the dial-up connection to the MCH is set up, the cellular telephone system hides all further mobility of the Mobile Node.

As described above the Mobile IP routing and the cellular telephone system can be used to solve the mobility problem. However, neither of them is able to solve the problems related to the behaviour of the wireless data link: high variability in the transmission delays, which can sometimes be very long, and occasional temporary disruptions of the transmission service.

2.2 Problems in Using TCP/IP over Wireless Links

Due to the different nature of the wireline and wireless behaviour, there are several drawbacks in a straightforward adoption of a cellular telephone system and the Mobile IP routing support. Firstly, there is network traffic which is unnecessary for the Mobile Node but which will be delivered over the slow wireless link if not explicitly filtered by the MCH. There is also unnecessary TCP/IP protocol header information exchanged over this link. Of course, the header information of TCP connections can be reduced by using the technique proposed by Van Jacobson [Jac90]. This technique can be used with the SLIP framing protocol called CSLIP. The new PPP protocol also supports compression of TCP and IP headers. However, reducing the header information of UDP datagrams is not supported.

Secondly, most of the traditional network applications, such as FTP and mail exchange (e.g., SMTP [Pos82]) use TCP, which is an end-to-end protocol. Therefore, all retransmissions, also those due to packets lost on the path between the MCH and the peer host in the fixed net, will pass the slow wireless link. Furthermore, link-level recovery from bursty errors may cause delays that are long enough for the TCP timeouts to trigger a packet retransmission. This can result in two kinds of undesirable behaviour: 1) The slow wireless link has to transmit all the resent packets even though no packets were actually lost but only delayed. 2) The fixed network has the extra burden of transporting all the packets from the fixed host via the Home Agent to the MCH. It should also be noted that the reasonable packet sizes on wireline and wireless links are different.

In addition to excess data delivery, there are performance problems due to the congestion control policy of TCP [Jac88]. If the retransmission timer goes off indicating a packet loss, the TCP sender reacts as if the reason were congestion somewhere on the route. Hence, after each timeout TCP sender slows down its transmission rate through doubling the retransmission timeout value and through decreasing the transmission window. Thus, a temporary disruption on the wireless link rapidly leads to long response times. Furthermore, the slow-start algorithm used by TCP is rather conservative in restoring the efficient operational transmission rate [CI93]. However, after the disappearance of the distortion, the wireless link is immediately capable of working at full speed.

Finally, the wireless link can cause serious problems for existing applications which depend on TCP. These problems arise when the behavior of the wireless link causes a TCP connection to break down: few applications are sophisticated enough to continue the interrupted activity. For example, a file transfer may have to be restarted from the beginning. An unnecessary break-down of TCP connection can occur in a couple of cases. When the wireless link between the Mobile Node and the MCH breaks, many TCP/IP implementations react by terminating the end-to-end TCP connection once and for all. Even if the break-down of the wireless link would not directly cause termination of the TCP connection, resuming the dial-up connection can take very long time causing the TCP sender to terminate the TCP connection after several retries. Exceptionally long delays on wireless links can have the same effect: after several retransmissions the TCP sender believes that a data link is broken and terminates the TCP connection. In principle, it is possible to reconfigure or dynamically adjust the TCP timeout or counter that is used to decide how long to retransmit an unacknowledged packet before terminating the connection. However, hardly any implementation of TCP allows reconfiguration of the TCP timeout for maximum retransmitts, not to mention a dynamic adjustment.

Even if it were possible to readjust the retransmission timeout, it is well known that finding a suitable timeout value is difficult. If the wireless link is broken and the timeout period is long, all pending packets from the fixed host to the Mobile Node will be retransmitted for a long time. This will inevitably cause excess load in the fixed network and in the MCH. However, the most serious aspect is that timeouts should be adjusted at both ends of the connection. This implies that the fixed node would need to know that it is dealing with a Mobile Node. Hence, the TCP protocol software on any fixed node willing to communicate with Mobile Nodes would have to be modified.

3. The Mowgli Communication Architecture

The Mowgli communication architecture is designed to increase the usability, reliability and efficiency of client-server communication between a Mobile Node and a fixed host. In the Mowgli architecture the basic idea is to split the channel with an end-to-end control into two parts with a store-and-forward -type interceptor. This interceptor, the Mobile-Connection Host, allows us to implement two separate but flexibly co-operating data communication subsystems: one wireline oriented, the other wireless oriented. Therefore, we can retain the existing TCP/IP-based communication infrastructure of the fixed network but we can

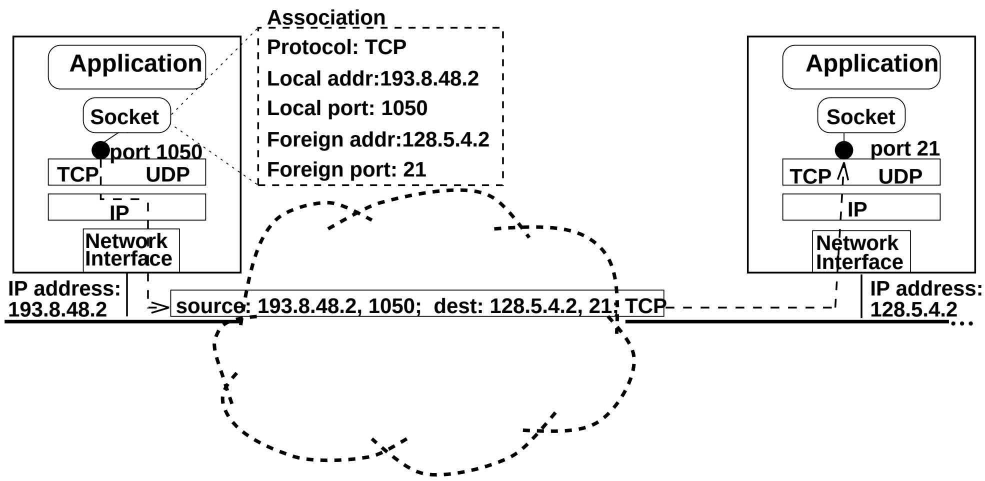

Figure 2: An association assigned to a TCP socket

also implement special purpose protocols for the wireless link. The interceptor also makes implementation of high-level functionality possible.

3.1 The Socket Abstraction and TCP/IP Protocols

A socket interface is widely used as the interface between application programs and communication protocols. Nowadays, most of the traditional TCP/IP Internet applications use TCP/IP protocols through some version of socket API.

When a socket abstraction is used for interprocess communication, data is exchanged between a pair of sockets bound to the underlying protocol. A socket interface defines functions needed to address a peer socket, create and close connections, as well as to send and receive data according the scheme defined by the underlying protocol. TCP/IP protocols define a communication endpoint, a half association, to consist of a 3-tuple: 1) an IP address, 2) a port number, and 3) a protocol (TCP or UDP). In order to accept incoming TCP connection requests or to receive UDP datagrams from a socket, the application program must bind a local communication endpoint to the socket. In order to create a TCP connection or to send a UDP datagram from the socket to a foreign destination, the program has to supply the communication endpoint of the desired destination so that a full association can be assigned to the socket. The full association is defined as a 5-tuple which consists of the half associations of the involved ends (see Figure 2).

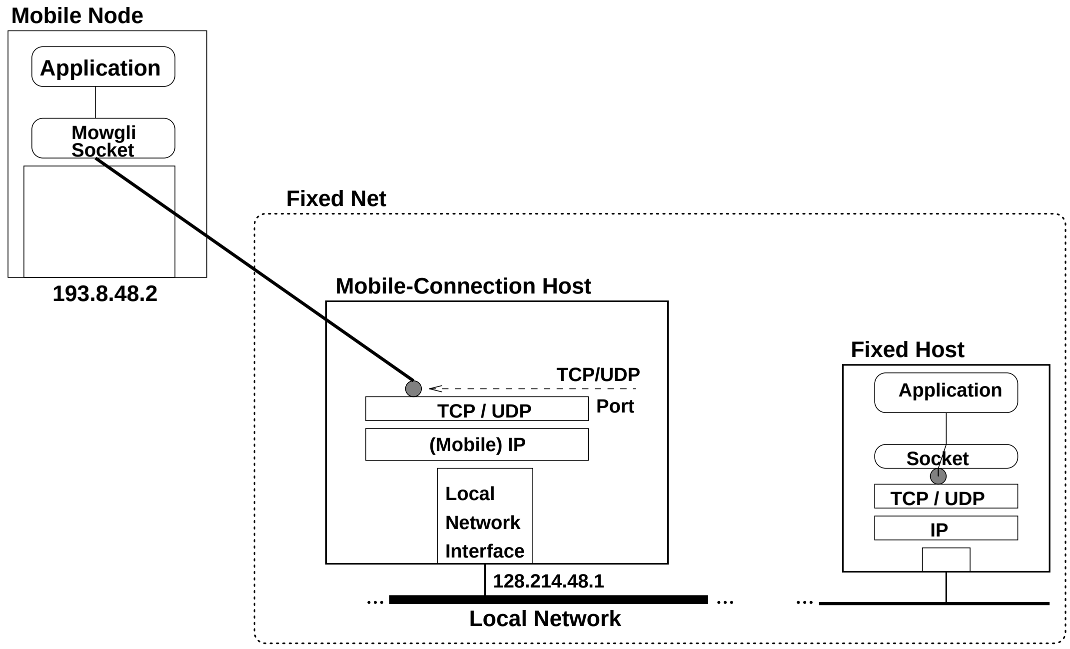

Figure 3: General view of the Mowgli socket approach

3.2 Mowgli Sockets

The application programming interface offered in our communication architecture is based on the socket abstraction. The socket interface, called Mowgli sockets, enables application programs on a Mobile Node to use TCP/IP-based services which are available on fixed hosts although the TCP/IP protocols are not used on the wireless link between the Mobile Node and the MCH. The Mowgli socket interface will be implemented in such a way that it will provide the same functionality as the original stream sockets and datagram sockets, i.e. sockets for TCP and UDP protocols, respectively. Thus, applications on a Mobile Node can use TCP/IP sockets for communication but the TCP/IP stack is actually running on the MCH as exemplified in Figure 3. This is achieved by using the MCH as a socket-level gateway.

As a result, an application program on a Mobile Node can communicate through the Mowgli socket interface with the rest of the TCP/IP Internet by sending and receiving data exactly in the same way as it would do when connected directly to the Internet. The application program at the other end will be unaware of the modifications that have been made to transfer user data over the wireless link. This approach works well when all traffic to and from a Mobile Node traverses through an MCH as it does when wireless point-to-point

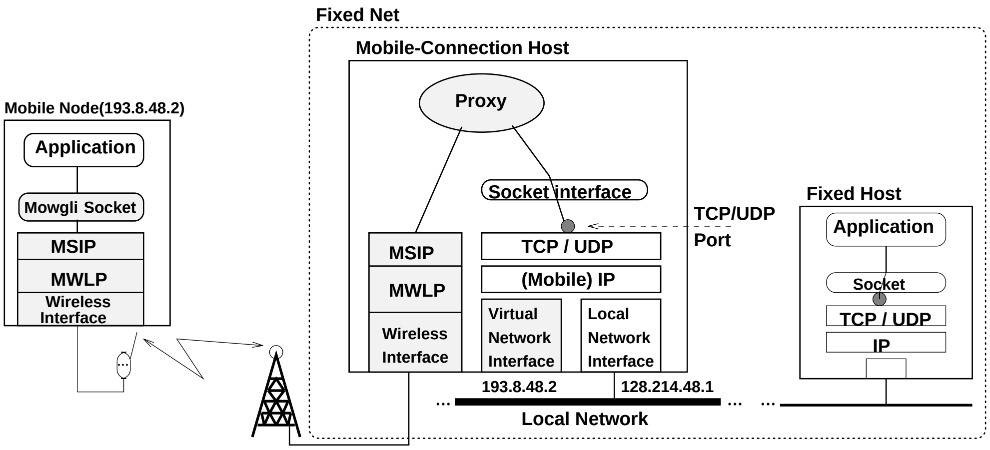

Figure 4: Implementation outline of the Mowgli sockets

link is used to connect the Mobile Node to the Internet.

When TCP is used as the transport level service, a reliable byte stream between the fixed host and the MCH is achieved as a result of standard TCP functionality. The Mowgli socket implementation preserves the reliability between the MCH and the Mobile Node.

3.3 Implementation Overview

An implementation outline of the Mowgli sockets is given in Figure 4. When a Mobile Node dials in to an MCH, the MCH creates a new virtual network interface and assigns the IP address of the Mobile Node to that interface. Then it creates a *proxy* that will act as the mediating agent for the Mobile Node. Adopting the Mobile IP routing support allows a mobile user to dial in to an MCH which is not located at the user's home network. In this case, the MCH registers with the Home Agent of the Mobile Node on its behalf. The Home Agent tunnels IP datagrams destined for the Mobile Node to the Foreign Agent on the MCH as proposed in the Mobile IP draft.

When a new Mowgli socket is created by an application program on the Mobile Node, the proxy creates a corresponding socket that will be bound to the virtual network interface created for that Mobile Node. When an IP datagram addressed to the Mobile Node arrives at the MCH, the Foreign Agent does not deliver it directly to the Mobile Node as it does in a straightforward Mobile IP implementation (see Figure 1). Instead, in our approach all IP datagrams which arrive at the MCH and contain the address of the the Mobile Node as their destination are passed all the way up through the TCP/IP protocol stack to the corresponding socket.

The proxy is responsible for receiving data from the socket and transmitting the data over the wireless link to the Mowgli socket on the Mobile Node. Special purpose protocols are used to transmit the data over the wireless medium. The application on the Mobile Node receives the data from the Mowgli socket. Similarly, data sent from the Mobile Node through the Mowgli socket is first transferred to the proxy on the MCH. The proxy then sends the data through the "traditional" socket on behalf of the application program on the Mobile Node. The outbound packets which carry user data originating from the Mobile Node will contain the IP address of the Mobile Node as their source address, i.e. the address assigned to the corresponding virtual network interface. Thus, the peer somewhere on the fixed network receives the data as if it were directly sent by the application program on the Mobile Node. Even though the data traverses through the higher layers at the MCH, there will not be any significant performance degradation because the slow wireless link still remains the bottleneck.

The basic responsibility of the proxy is simply to redirect data between the sockets used by applications on a Mobile Node and on a fixed host. It also performs necessary socket operations on behalf of the applications on the Mobile Node. The functions needed on the MCH can be implemented as user-level software together with a suitable configuration of the MCH. In many Unix[2] versions no kernel-level modifications should be needed. However, a more convenient solution can probably be implemented with minor kernel-level modifications in conjunction with a Mobile IP implementation. Any kernel-level modifications that are needed depend on the way the Mobile IP routing is adopted. It should be noted that if a Mobile Node dials in to an MCH located at its home network, Mobile IP routing support is not needed.

We need a protocol to carry the control information concerning different socket operations as well as the user data between a Mowgli socket on the Mobile Node and the proxy on the MCH. The *Mowgli Socket Interface Protocol* (MSIP) is designed for this purpose. The MSIP protocol is also able to carry the control information which the Mobile Node needs in controlling the functions of the proxy.

The *Mowgli Wireless Link Protocol* (MWLP) is a protocol that is specially designed for data transmission over a slow and unreliable wireless point-to-point link. An existing

[2]UNIX is a registered trademark in the U.S. and other countries licensed exclusively through X/Open Company, Ltd.

protocol suitable for wireless links may also be used. The main objective of the protocol is to make it possible for upper protocol layers and application platforms to fully exploit the narrow data channel offered by a cellular telephone system. Yet, it should also be well suited for transmitting data over a wired serial (modem) link. Furthermore, it should be able to support byte stream type of data transfer as well as datagram delivery so that the functionality of the TCP and UDP sockets is easy to implement in the MSIP protocol.

4. Enhanced Functionality for Mobility

In addition to the implementation of basic functionality in the Mowgli sockets, the proxy can perform enhanced operations for the Mobile Node. The enhanced functionality provided by the proxy can be used to control the operations crossing the slow and vulnerable wireless link. Typical examples include fault-tolerant transfer of long files, remote control of complicated data-base operations, and processing of replicated information. However, any action taken by the proxy must be either explicitly or implicitly agreed with the Mobile Node or with the application on the Mobile Node. The enhanced functionality will be implemented using application level parameters for quality of service and by including new operations to the Mowgli socket interface. Furthermore, a new level of abstraction is added above the socket layer to support implementation of client-server communication in applications aware of mobility.

4.1 Improving Quality of Service

In a mobile environment, the most important quality-of-service attributes at the application level are those related to reliability and performance. The functions intended for improving the quality of service can be roughly classified into two groups. The first group belongs to the general infrastructure of platforms aware of mobility. They exist to hide the effects of mobility. In terms of the ODP Reference Model [ISO94], these functions implement the essential distribution transparencies such as the location transparency, the failure transparency, etc. The second group is a collection of functions for supporting mobility. They can be used to design enhanced application-level functionality, to help the user, either explicitly or implicitly, and to cope with the problems of mobility. Using the basic functions the application programmer can create compound functions needed for the specific application.

The first group includes basic functions needed when old client-server applications are used in a mobile environment without any modifications. Enhanced functions which are general enough to help any type of communication are needed. For example, the dial-up connection on an idle wireless link can be automatically torn down after a predefined timeout and re-established when new data arrives to be delivered across the link. The link can also be automatically re-established after an abnormal termination. Furthermore, if the wireless connection is temporarily broken, the proxy can receive and store data for delayed transmission to the Mobile Node.

The second group provides functions which are more application oriented. Typical operations may include name service functions accommodated to mobility, fault-tolerant transfer of data, and control functions for remote and group operations. These operations can have attributes related to application-level quality of service.

An interface for setting the quality-of-service parameters will be given for mobile users so that they can tailor the proxy for applications which use the traditional BSD or Windows sockets. The parameters can be set according to the needs of the applications protocols. Particularly, different parameters can be set for well-known ports which applications use.

New client programs are needed in order to take full advantage of the enhanced functionality of a proxy. They must be able to co-operate with a sophisticated proxy when using TCP/IP-based services provided by existing servers. New pieces of functionality will be included in the Mowgli sockets for this purpose.

4.2 Functionality above the Mowgli Socket Layer

The Mowgli communication architecture to be implemented in Mobile Nodes is shown in Figure 5. One of the basic ideas is to use a virtual layer so that it is possible to select between the socket implementations below the virtual layer. In our architecture, the virtual layer is called the *Mowgli Virtual Socket Layer* (MVSL)

The objective of the virtual socket layer is to hide the slight differences between socket interfaces and to bind a virtual socket to the appropriate protocol stack according to currently available communication media. Thus, the MVSL layer is bound to the Mowgli socket service when the Mobile Node is operating in "mobile" mode, and to an appropriate "traditional" socket interface when the Mobile Node is operating in "wired" mode. In this way, the MVSL performs a simple trader function.

For applications aware of mobility, a sophisticated communication protocol may be appropriate. The *Mowgli Wireless Message Protocol* (MWMP) is a special protocol which can be used by these applications. It can be used to define interactions between a mobile

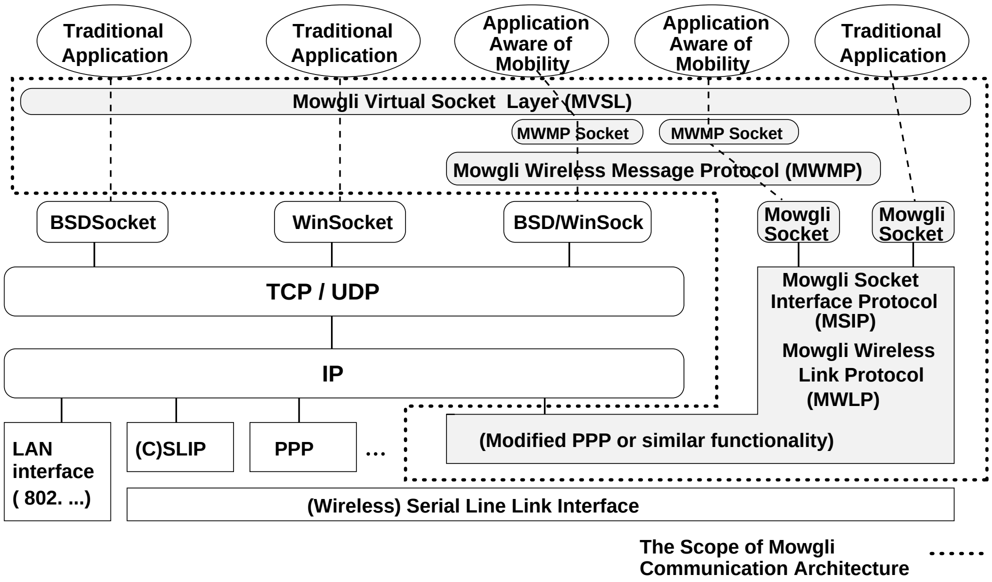

Figure 5: An overview of the implementation on a Mobile Node

client and a mediating proxy needed to use some existing fixed server. Enhanced operations specified in the MWMP protocol are intended for creating new kinds of "compound" client programs in which the client on the Mobile Node and the specialized proxy on the MCH are combined closely together to act as a client for an existing fixed server. Thus, most of the communication with the server can be done by the proxy on behalf of the mobile client.

As an example we can consider distributed transaction processing with a client on a Mobile Node. The mobile-node resident part of the application specifies a particular transaction. This specification is then sent to the MCH-proxy of the application. The proxy takes over the responsibility for the transaction while the connection between the Mobile Node and the MCH, an expensive GSM call, can be terminated. The client on the Mobile Node may later ask the proxy about the results of the transaction, or the proxy can call back to inform the client.

For new applications intended to run on Mobile Nodes and fixed hosts aware of mobility, the MWMP protocol can be augmented to a sophisticated end-to-end communication protocol which accommodates the indirect interaction paradigm of communication. The functions of the MWMP will be specially designed for the needs of the mobile computing. A programming interface to the MWMP is presented to application programmers as a new type of socket. The MWMP will be designed so that it can be implemented above any of the several socket interfaces and transport services as shown in Figure 5. Binding between an MWMP socket and an appropriate transport level service is provided by the MVSL layer.

5. Discussion

We have presented a communication architecture which builds a bridge between the worlds of wireless and wireline communication. Our architecture retains the TCP/IP-based communication 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. In addition, our approach provides an application programming interface which allows an easy development of new mobile (client) applications which co-operate with existing services on fixed hosts. The application programming interface also allows an easy development of mobile-aware client-server applications of a new kind.

We have paid particular attention to fault-tolerance and performance. Our architecture will minimize the traffic over the wireless link and improve quality of service at the application level, especially fault-tolerance and performance. The bottom line in our architecture is to split the channel with an end-to-end control into two parts with a store-and-forward -type interceptor. Communication within nodes which are aware of mobility is based on a higher level communication model of indirect client-server interaction. The indirect model of client-server interaction between a mobile client and a fixed server involves two separate interactions which can be handled differently, and a mediating agent between these two hops.

The basic functionality of the architecture presented is relatively straightforward to implement in a Unix-environment. The layered structure is flexible so that future developments can easily be adapted. The requirement analysis and the initial specification of the core architecture are ready. Currently we are working on the detailed specification and on the software design. Our research activities include the following three topics:

1. classification of different types of communication in application protocols, quality-of-service parameters and their relationship to the functionality of the proxy,
2. specification of enhanced services needed by applications aware of mobility, and
3. security.

We also follow the work in progress on the Mobile IP standard so that our solutions will be compatible.

Acknowledgments

The authors are grateful for the fruitful discussions with Heimo Laamanen, Marko Moilanen, and Henry Tirri. The Mowgli research project is supervised by professor Martti Tienari and funded by Digital Equipment Corporation, Nokia Mobile Phones, Telecom Finland, and the Finnish Ministry of Education.

References

[BBIM93] B. R. Badrinath, A. Bakre, T. Imielinski, and R. Marantz. Handling Mobile Clients: A Case for Indirect Interaction. In *Proc. IEEE 4th Workshop on Workstation Operating Systems (WWOS-IV)*, IEEE Computer Society (TCOS), Napa, CA, U.S.A., October 1993.

[CI93] R. Caceres and L. Iftode. The Effects of Mobility on Reliable Transport Protocols. Mitl-tr-73-93, Matsushita Information Technology Laboratory, Princeton, NJ, U.S.A., November 1993.

[Com91] D.E. Comer. *Internetworking With TCP/IP: Principles, Protocols, and Architecture.* Prentice-Hall, second edition, 1991.

[ETS90] ETSI. Quality of Service. GSM specification 02.08, version 3.0.0, ETSI/TC GSM, March 1990.

[HTA+93] M. Hall, M. Towfig, G. Arnold, D. Treawell, and H. Sanders. Windows Sockets: An Open Interface for Network Programming under Microsoft Windows. Version 1.1, January 1993.

[ISO94] ISO. Draft Recommendation X.903: Basic Reference Model of Open Distributed Processing - Part 3: Prescriptive Model. Draft International Standard, ISO/IEC DIS 10746-3.1 (Draft ITU-T Recommendation), ISO/IEC JTC1/SC21/WG7, April 1994.

[Jac88] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM '88 Symposium on Communications Architectures and Protocols*, pages 314–329, Stanford, CA, U.S.A., August 1988.

[Jac90] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. Request for Comments 1144, Network Information Center, February 1990.

[McG92] G. McGregor. The PPP Internet Protocol Control Protocol (IPCP). Request for Comments 1332, Network Information Center, May 1992.

[Pos82] J. Postel. Simple Mail Transfer Protocol. Request for Comments 821, Network Information Center, August 1982.

[Rah93] M. Rahnema. Overview of the GSM System and Protocol Architecture. *IEEE Commucation Magazine*, 31(4):92–100, April 1993.

[Rom88] J. Romkey. A Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP. Request for Comments 1055, Network Information Center, June 1988.

[Sim93] W. Simpson. The Point-to-Point Protocol (PPP). Request for Comments 1548, Network Information Center, December 1993.

[Sim94] W. A. Simpson(ed.). IP Mobility Support. Internet draft, IETF, Network Working Group, September 1994.

Helsinki 1994
Yliopistopaino