SLM, A Framework for Session Layer Mobility Management

Björn Landfeldt*, Tomas Larsson**, Yuri Ismailov**, Aruna Seneviratne*

*Dept. of Electrical Engineering and Telecommunications
The University of New South Wales
Sydney 2052 Australia

**Ericsson Radio Systems AB
Torshamnsgatan 23
164 80, Kista, Sweden

***Abstract*-This paper describes a novel framework for managing connections to mobile hosts in the Internet. The framework, SLM, integrates the notions of Quality of Service management and mobility management and forms a base for overall session management. We compare SLM with the currently most widely adopted mobility management framework Mobile IP, and show how some of Mobile IP's deficiencies are overcome. The paper further presents some initial experimental results and future research.**

1 INTRODUCTION

The Internet today consists of a heterogeneous set of networks with different characteristics and associated price. The introduction of wireless wide and local area networks has further increased this heterogeneity and the price difference. Furthermore, the introduction of laptops and palmtops has created a large diversity in end system capabilities such as CPU, keypad, display and available energy. This diversity will increase even more with the introduction of newer wireless networks, which will allow access to information from "anywhere at anytime". In parallel, Internet is expanding to incorporate traditional services as well as new services such as video on demand, IP telephony and TV.

Thus, as the Internet becomes a natural part of everyday life there is a need to provide adaptable services via different overlaid network clouds with different addresses and possibly different addressing schemes [10]. These systems must firstly be able to support mobility of both users and computing devices as they move through the different networks. For example, as the user roams through different networks, new IP addresses will need to be acquired dynamically and these addresses will not always be known in advance. Secondly, the system must provide support for adaptation to cope with the diversity of computing devices and network characteristics. For example, the system may need to filter the content using agents or proxies in the data path cope with variation in bandwidth [11, 3].

The MARCH project [7], a collaborative project between Ericsson, the University of New South Wales, (Australia) and the University of Uppsala, (Sweden) is investigating novel methods of providing the services in this type of mobile computing environments. MARCH makes use of proxies for tailoring the communication sessions. In this paper, we report on mobility management within the project.

The IP protocol stack was designed to be used with fixed networks. Consequently, the current IP routing model does not support mobility of computing devices. A number of proposals have been made to overcome this limitation [2, 13]. However, these proposals have a number of limitations, which may lead to poor performance as discussed in section 1. In addition, they are one dimensional in the sense that they only address mobility issues associated with routing. They do not provide the necessary system support for system adaptation using proxies or agents as discussed earlier.

This paper presents "Session Layer Mobility Management", SLM, a novel framework that not only provides support for mobility management in these overlaid networking environments, but also provides support for the use of proxies for system adaptation. The paper is organised as follows. Section 2 gives the background and presents related work. Section 3 outlines the proposed framework. Section 4 presents a test implementation and initial experimental results. Finally in section 5 we conclude and present future work.

2 BACKGROUND AND RELATED WORK

The diversity in end system capabilities and network characteristics, coupled with user mobility, demands special support from the system. We believe that the way users will interact with the communication environment will change drastically when mobile networked computing becomes ubiquitous. As an example, we consider the following scenario, schematically illustrated in Figure 1.

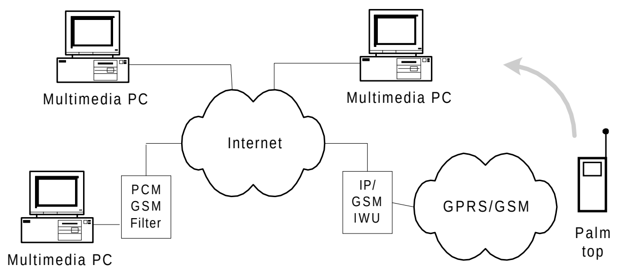

Figure 1. A scenario of future mobile networking

A user is placing a conference (audio+whiteboard) call to two of his colleagues using a multimedia PC. One of the colleagues is running late, and is on her way to the office.

Consequently, she can only be reached on her palmtop, which is connected via a cellular network. Because of the limited network capacity, only the voice is diverted to her palmtop. Furthermore, also because of the limited bandwidth, the audio stream is filtered/transcoded from PCM to GSM encoding. While the session is in progress, the colleague arrives at her work place. When she reaches her desk, she hands over the session to her desktop PC. Once handed over, the original conference call is established with PCM encoded audio and whiteboard.

This example highlights some of the future requirements of emerging mobile computing environments. Firstly, there is a need for locating the colleague and determining the capabilities of the end devices being used. Secondly, there is a need for address translation, to allow the diversion of calls between different types of networks. Thirdly, it may be necessary to perform some data filtering to suit the operating environment. Finally, mechanisms are needed to hand over the active session to a different address and possibly to a different computing device. Below we examine some of the limitations of the current IP model, which make it unsuitable for this environment, and discuss previous work.

Firstly, the Domain Name Service (DNS) [1] statically maps machine names to addresses. Thus, if a host changes its address, it cannot be found by the DNS. Furthermore, the DNS only maps machine names, not user names and therefore, a user cannot be located if s/he is not at a specific host known in advance. A recent proposal, called Session Initiation Protocol, SIP [8], overcomes this limitation of the DNS. SIP provides a directory lookup of user location; however, it does not provide methods for finding devices when they change address.

Secondly, the IP routing model does not allow machines to dynamically change their address. Open sockets do not allow a change in address. So if a host were to change its address during a session, the connection would have to be torn down and restarted with the new address. In addition, IP does not allow open connections to change physical location. If a host were to keep its address and move to a different location, IP routing would fail and the connection would be lost. Together this means that data in an open connection has to be routed to the same address at the same end node.

To date the issue of mobility in IP based networks have primarily been achieved through an extension to IP called MobileIP [2]. MobileIP addresses the issues of finding a roaming mobile host and maintaining established connections when roaming. It shows some nice properties, such as transparency to existing IP implementations and equipment. MobileIP has therefore been adopted by the IETF. However, MobileIP has a number of limitations that can have a severe impact on system performance.

In MobileIP, all IP traffic to a mobile host (MH) is sent to the host's home network where it is encapsulated in another IP stream and retransmitted to the MH's current location. This behaviour creates unnecessary and unwanted traffic in the home network. Furthermore, many applications require predictable behaviour from the network in order to operate correctly. A way of ensuring this is by reserving resources such as bandwidth and delay boundaries. Today, two models are commonly considered for making reservations, IntServ [4] and DiffServ [6]. DiffServ uses in-band signalling and so the IP packets are marked with priority information, which intermediate routers use for the traffic shaping. IntServ uses an external end-to-end signalling protocol such as RSVP [5], which is used by every intermediate router to set up the reservations. In both cases, the tunnelling breaches the end-to-end semantic and makes reservation, if not impossible so at least cumbersome.

Another weakness of Mobile IP is the lack of support for multiple network interfaces at the MH. In future networking environments, it is very likely that a MH will have access to more than one network simultaneously. It is also very likely that network availability will be dynamic so that during a session another network type might become available. All different networks will have different characteristics and associated price. This in turn will lead to users wanting to dynamically select different network connections for different traffic flows. In Mobile IP all traffic to a MH's network interface is aggregated and tunnelled to the same address, so MobileIP makes no distinction between different flows. Thus if a user wants one of the open flows to use another interface, all flows must be redirected to use that interface.

Recently a new way of looking at mobility management was introduced in a framework called Transport Layer Mobility Management, TLM [13]. In this framework, the mobility is managed at the transport layer rather than the network layer, as is the case in MobileIP. TLM manages mobility by introducing a proxy in-between the client and server. The data stream is intercepted at the client and a connection is made to the proxy rather than the server. The proxy then opens a new socket to the server and connects the two sockets. The intermediate proxy does not copy the data in user space, but splices the two TCP sockets in the kernel. This way the end to end semantics of TCP is kept intact, i.e. the intermediate sockets do not acknowledge any data. The proxy has a fix location at a predefined point in the network, typically at the gateway out from the network. We believe that this method of managing mobility is preferable to that of MobileIP. However, TLM shows some limitations, which make it unsuitable for a heterogeneous networking environment.

TLM was designed to deal with connections only within a single network segment. If the mobility proxy is fixed and located at an intermediate host and a host changes network, all data will have to be routed through the proxy for the lifetime of the connection. This will lead to triangular routing and exhibit the same undesirable features of triangular routing in MobileIP. We believe that open connections will migrate over network boundaries, so support for vertical hand off is necessary.

Furthermore, if data filtering is required on an intermediate host, it is reasonable to assume that the location of the filtering proxy will coincide with that of the mobility proxy in order to avoid the complexity of multiple concatenated proxies. Since data filtering must take place in user space the TCP splicer cannot be used. Even if the two proxies would not coincide, the connection would have to be broken at the data filtering proxy. Data filtering can be used both for content alterations as well as protocol enhancements. For example, images can be converted from colour to b/w to reduce their size and TCP can be altered to work better over lossy links [10, 11]. It has been shown in [3] that substantial improvements can be made to download times of images through distillation.

3 THE PROPOSED FRAMEWORK, SLM

SLM is based on existing Internet technologies. It extends the current TCP/IP model by introducing a session layer, similar to that of the OSI model. Future communication applications such as video/audio telephones will use managers for session set-up using specialised session protocols such as SIP and H323 [12]. We therefore believe that this extension complies with the future Internet environment.

SLM operates above TCP and switches TCP streams between connections. It consists of two parts. One part, a session management part, operates within the end hosts and supports the dynamic connection establishment and tear down. The other, location management part, is situated within the network and enables locating the called party and does the address resolution. The rest of this section first describes the session management and then the location management parts before concluding by describing how they interact to provide the overall framework.

3.1 THE SESSION MANAGEMENT

As described earlier, the mobility management schemes proposed to date have focused on maintaining connections to a computer. For example, MobileIP operates at the network layer, thus limiting itself to only tracking movements of an end system. Session layer mobility management introduces a new semantic where the management applies to the open data streams rather than the computer. Thus, in SLM, a data stream can move in-between different network addresses, network devices or end hosts. The emphasise is on a data stream changing address, possibly together with a computing device, thereby supporting the techniques described in the operational scenario in section 1.

In SLM, all open connections can either be treated as separate sessions or be aggregated into larger groups and managed accordingly. Assume for instance that a user is downloading a web page with a few concurrent streams for images and text. Each stream is regarded as a session, but since they all belong to the same process and presentation on the screen, they can be aggregated so that the same management rules applies to them all. Equally, SLM allows the separate management of different streams belonging to the same process. For example, during a videoconference with audio and video data streams, generally the audio stream is given a higher priority. Consequently, the user may want to switch the audio stream over to a more reliable (hence more expensive) network if the current network becomes congested. In SLM, this is achieved by each connection being assigned a unique session ID or a group ID if connections are aggregated. Figure 2 shows the layout of the session management hierarchy on the end-hosts.

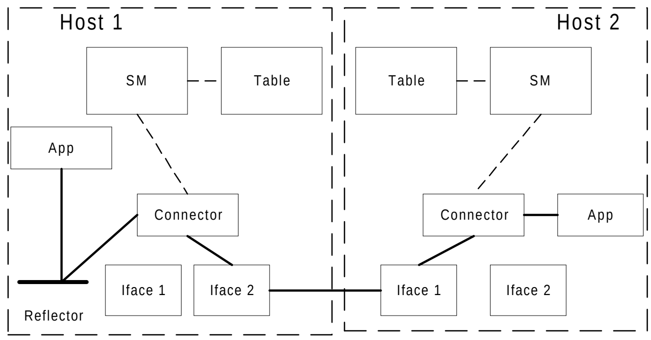

Figure 2. SLM end-host session management hierarchy

The connections within a host are controlled by a session management entity (SM). The SM contains the necessary logic for starting and maintaining sessions. The state of each connection is also maintained by the SM and stored in a session table.

The end-to-end network path for each data stream is broken into three separate paths. One path between the application on host 1 and a socket connector, another path between the application on host 2 and a socket connector and finally one path between the socket connectors of the host1 and host2. The paths between the applications and socket connectors stay constant during the lifetime of the session. As the data is reflected, and the connectors remain constant, the applications are shielded from changes to the connections that occur as a result of mobility. In contrast, the intermediate path will change as the hosts move, or network preferences change.

The only difference between the hosts is that the host that establishes a connection (host1) needs a reflector, which redirects the applications data stream to the connector. This makes the use of SLM transparent. The other host (host2) does not need the reflector as the session manager opens the connection rather than the application.

To illustrate the function of the framework; consider the scenario in section 1. When the caller starts the audio session, the audio channel is intercepted by the reflector and redirected to a connector. The connector then signals the colleague’s palmtop SM, which in turn starts a connector for the channel (how the palmtop is located is described below). Finally, the palmtop connector opens a socket to the audio application.

The divided path is transparent to the application, so the connection looks normal. The two connectors copy the data in-between the two sockets until the session is closed or reconnected (the use of a transcoding proxy is also described below).

The colleague arrives at her desk and wants to hand over the session to her desktop PC. The SM looks up its table for the audio application's session key. The SM would then signal the SM located on the desktop PC and ask it to take over the session. The desktop SM would then create a new connector, which would signal the callers SM with the session ID and ask it to hand over the session. Finally, the callers SM would reconnect the application path with the new path established to the desktop connector.

3.2 NETWORK SUPPORT

The network support required by SLM is essentially to locate the called party and provide the necessary address translations. For example, if a MH is away from its "home network", it becomes impossible to locate using its IP address, since the IP address reflects its physical location in the network.

SLM uses a new network entity, the User Location Server, (ULS). The ULS can be queried to acquire MH's current locations as they roam. Thus, the ULS have a functionality similar to the HA in MobileIP, or the HLR in GSM networks. The location of the ULS is typically associated with the "home address" of a MH, the address through which an MH can be located.

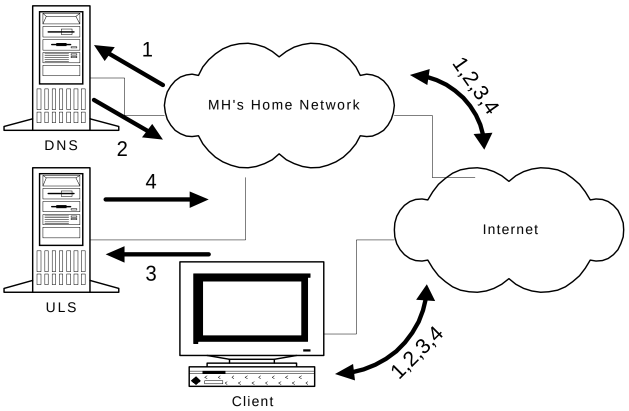

Figure 3. SLM location signalling

The location of the ULS can be determined in two ways. Either by sending its address as a part of the session set-up protocol between two session managers. This procedure allows a MH to specify the ULS to use for the remote host. The second is by querying an extended DNS about its location. This requires an extension of the DNS with a new record type. How this update can be done is described in [9]. Figure 3 shows the sequence of events related to locating a MH through a modified DNS. The client application starts locating the server by querying the extended DNS, using SLM's mobile address resolution API. In response, the extended DNS returns the location of the MH's ULS address. The application in return queries the ULS about the current address of the MH. The ULS looks up its table and returns the address to the application.

The ULS is a natural extension to user location protocols such as SIP. SIP, enables the mapping of user location to a certain host, i.e. looking for user@unsw.edu.au will return the IP address of the host that user is currently using. The ULS will further enhance this function by returning the current IP address of that host if it has moved to a new network address.

3.3 QoS SUPPORT

SLM allows QoS management to be carried out in a number of ways. Firstly, it maintains the normal IP routing semantic between two hosts, so it allows resource reservation using both IntServ [4] and DiffServ [6], without breaking their semantics. When a host changes addresses, the existing reservations can be torn down and a new reservation can be established.

Secondly, SLM allows the placement of intermediate proxy modules for data filtering. Filtering can be used for adapting the data to the hosts hardware limitations as well as adapting to existing network limitations. As an example, consider the operational scenario presented in section 1. The SM will install a proxy on top of the connector in host 1 when the call is diverted to the palmtop. This will convert the audio stream to GSM encoding to suit the limited data rate in the cellular network. When the session has being handed over to the multimedia PC at the remote end, SM will simply remove it.

4 EXPERIMENTAL RESULTS

We have made a test implementation of SLM, in order to verify its feasibility and to make preliminary measurements. To make the framework platform independent, the implementation was done in JAVA. All machines were standard PC's running Linux. We were interested to see how the framework behaved while roaming through different wireless networks. Therefore, we used 2Mbit/s WaveLAN networks in our experiments. Since our primary goal was to investigate mobility issues, we only used session managers on the end systems and no intermediate filtering proxies.

We used a signal daemon for measuring the signal strength of DHCP offers from different WaveLAN base stations. The daemon was tuned to our environment by setting appropriate thresholds for hand over between different subnets. Furthermore, some hysteresis was used to avoid oscillations at network boundaries. When the signal strength reached the hand over threshold, the signal daemon reconfigured the interface before signalling the SM to initiate hand over to the new network. The SM logic selected networks for a stream by using Linux's bindToDevice() function, which allows a socket not only to listen but also transmit on a specified network interface.

It is necessary to synchronise the streams when using TCP in order to maintain the end-to-end error control. It is impossible to know what data has actually been successfully transferred over the link and what data is being buffered, by looking at the socket. We therefore introduced synchronisation on top of TCP, within the connectors by maintaining counters for sent and received data for each socket in order to obtain the starting point of the new connection.

We played an MPEG video from the server at a rate of 500 kbit/s over the WaveLAN segment, which had a data rate of approximately 1.6 Mbit/s. When we reconnected, the total time for sending TCP's three way handshake together with the reconnection protocol exchange came to approximately 25 ms. This value will change proportionally to the link delay, so if the path consists of more hops the reconnection time will increase. However, the reconnection protocol was a string-based protocol and not optimised for speed, so the time can be further reduced.

Figure 6 shows a plot of transmitted data versus time. Socket 2 is shifted up to the finish of socket 1 to illustrate the continuing transmission as one series. It can be seen in the figure that the derivative of the plot, i.e. the data rate stays stable during the reconnection. Thus, we have confirmed that SLM is feasible to implement on any platform and behaves well while reconnecting data streams.

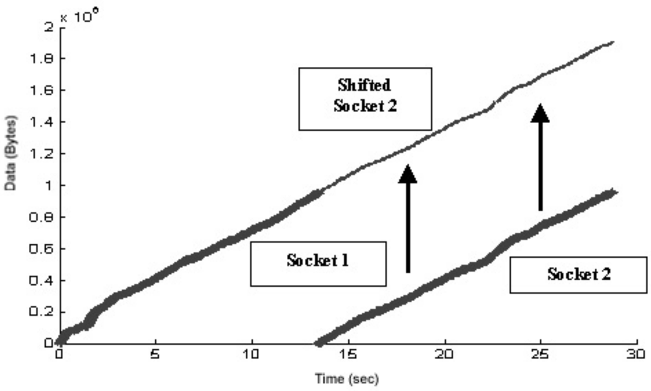

Figure 6. Reconnecting while roaming

5 CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a framework for mobility management of IP networked devices. The framework was designed with support for QoS management in mind and introduced a new semantic where open data streams are treated as being separate sessions. With this semantic, we can focus on providing the necessary system support for applications that demand a specific operational environment.

In the future, we intend to investigate how hand over between different hosts should be carried out within this framework. There are a number of interesting research issues related to this. There is a need to look at the signalling between the hosts as well as the different security issues that arise. For example, it is necessary to provide both authentication of users wanting to hand over sessions as well as intermediate hosts breaking the data stream.

We also intend to look at the demands of applications taking over migrating sessions. It is not trivial for applications to being able to resume an open session. There is a need for investigating how application states can be transferred with the session in order to initialise and synchronise the new "host application".

ACKNOWLEDGMENTS:

Björn Landfeldt and Aruna Seneviratne would like to thank Ericsson Australia Pty Ltd for their financial support.

REFERENCES:

[1] W. Richard Stevens, "TCP/IP Illustrated vol. 1, Addison & Wesley

[2] IETF RFC 2002

[3] A. Fox and E. Brewer, "Reducing World Wide Web Latency and Bandwidth Requirements by Real Time Distillation", Proc. Fifth International Conference on WWW, France 1996.

[4] R. Braden et al. "Integrated Services in the Internet Architecture: an Overview", RFC 1633, ISI, MIT and PARC, June 1994.

[5] R. Braden et al. "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification ", IETF RFC 2205)

[6] K. Nichols, V. Jacobson and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", IETF Internet draft November 1997.

[7] University of New South Wales and University of Uppsala, Technical Report 1999.

[8] IETF Internet draft, "draft-ietf-mmusic-sip-12.txt"

[9] IETF RFC 2052

[10] R. Katz and E. Brewer, "The Case for Wireless Overlaid Networks", SPIE Multimedia and Networking Conference (MMNC'96) San Jose January 1996.

[11] E. Amir, S. McCanne and R. Katz, "An Active Service Framework and its Application to Real Time Multimedia Transcoding", SIGCOMM'98 Boston USA, August 1998.

[12] http://www.itu.int//itudoc/itu-t/rec/h/s_h323.html

[13] MSOCKS: An Architecture for Transport Layer Mobility IEEE INFOCOM'98: Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies pp.1037-1045